

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

О. Є. Поморцева

ОСНОВИ ГЕОІНФОРМАЦІЙНИХ
СИСТЕМ І БАЗИ ДАНИХ
ПІДРУЧНИК

Харків
ХНУМГ ім. О. М. Бекетова
2022

УДК 910.27:004.65](075.8)

П55

Автор

Поморцева Олена Євгенівна, кандидат технічних наук, доцент, доцент кафедри земельного адміністрування та геоінформаційних систем

Рецензенти:

К. А. Мамонов, доктор економічних наук, професор кафедри земельного адміністрування та ГІС ХНУМГ ім. О. М. Бекетова;

М. В. Новожилова, доктор фізико-математичних наук, професор кафедри комп'ютерних наук та інформаційних технологій ХНУМГ ім. О. М. Бекетова;

С. Г. Удовенко, доктор технічних наук, професор кафедри інформатики та комп'ютерної техніки Харківського національного економічного університету ім. С. Кузнеця

*Рекомендовано до друку
Вченою радою ХНУМГ ім. О. М. Бекетова,
протокол № 1 від 1 жовтня 2021 р.*

Поморцева О. Є.

П55 Основи геоінформаційних систем і бази даних : підручник / О. Є. Поморцева ; Харків. нац. ун-т міськ. гос-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2022. – 346 с.

ISBN 978-966-695-568-8

У підручнику розглянуто загальні відомості щодо баз даних і системи керування базами даних, основні теоретичні поняття, що розкривають основи та принципи проектування реляційних баз даних, а також геоінформаційні системи як редактори векторної та растрової графіки, що включають просторові бази даних, які можна використовувати для вирішення наукових і прикладних завдань. Отже, розглядається технологічний комплекс, що інтегрує та поєднує багато інформаційних технологій, які входять до геоінформаційних систем.

Досягнутий рівень компетентності дозволить ефективно використовувати можливості геоінформаційних систем і просторових баз даних під час виконання завдань у професійній діяльності студентами освітньої програми «Геодезія, картографія та землеустрій», створить основу для самостійного освоєння нових версій геоінформаційних систем від різних розробників.

УДК 910.27:004.65](075.8)

ISBN 978-966-695-568-8

© О. Є. Поморцева, 2022

© ХНУМГ ім. О. М. Бекетова, 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	4
ВСТУП	5
РОЗДІЛ 1 ПРОЄКТУВАННЯ ТА СТВОРЕННЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ.....	9
1.1 Концепція побудови баз даних	9
1.2 Принципи та теоретичні основи проєктування реляційних баз даних.....	63
1.3 Аналіз даних за допомогою запитів. SQL запити.....	89
1.4 Ведення баз даних за допомогою форм	107
РОЗДІЛ 2 БУДОВА ТА ФУНКЦІЇ ГЕОІНФОРМАЦІЙНИХ СИСТЕМ	112
2.1 Класифікація ГІС. Моделювання навколишнього світу	112
2.2 Введення даних. Цифрування вихідної інформації.....	118
РОЗДІЛ 3 ПРИНЦИПИ ОРГАНІЗАЦІЇ ТА ЗБЕРІГАННЯ ДАНИХ У ГЕОІНФОРМАЦІЙНИХ СИСТЕМАХ	129
3.1 ГІС і просторовий аналіз	135
3.2 Застосування геоінформаційних систем для виконання завдань народного господарства	162
РОЗДІЛ 4 ЗАВДАННЯ НА ПРАКТИЧНІ ЗАНЯТТЯ ТА ІНДИВІДУАЛЬНЕ ЗАВДАННЯ	184
4.1 Практичне заняття. Створення та заповнення бази даних	184
4.2 Практичне заняття. Операції з даними. Захист від помилки введення	196
4.3 Практичне заняття. Операції з даними. Умова на значення. Використання полів зі списком	202
4.4 Практичне заняття. Створення запитів. Запити на вибірку	215
4.5 Практичне заняття. Створення нових форм у СКБД ACCESS. Робота з комбінованими формами	233
4.6 Практичне заняття. Знайомство з ArcMap	271
4.7 Практичне заняття. Знайомство з ARCGIS. Формати просторових даних	281
4.8 Практичне заняття. Конвертація даних в ArcGIS	295
4.9 Практичне заняття. Географічна прив'язка зображень з використанням векторних шарів карти і за координатами. Створення мозаїки растрів	299
4.10 Практичне заняття. Створення цифрової моделі карти	311
4.11 Індивідуальне завдання	322
ГЛОСАРІЙ	337
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	345

ПЕРЕЛІК СКОРОЧЕНЬ

- ESRI – американська компанія, розробник найпоширенішої ГІС
- OLE – технологія зв'язування та впровадження об'єктів в інші документи та об'єкти (Object Linking and Embedding)
- SQL – структурована мова запитів (Structured query language)
- XML – розширювана мова розмітки (Extensible markup language)
- АСУТ – Автоматизовані системи управління транспортом
- БГД – база геоданих
- БД – база даних
- ГІС – географічна інформаційна система
- ДЗ – дистанційне зондування Землі
- ЕОМ – електронно-обчислювальна машина
- ІС – інформаційна система
- ЛКМ – ліва кнопка маніпулятора миша
- ПК – персональний комп'ютер
- ПКМ – права кнопка маніпулятора миша
- СКБД – система керування базами даних

ВСТУП

Навчальний посібник підготовлений згідно з програмою обов'язкової навчальної дисципліни «**Основи геоінформаційних систем і бази даних**», яка має професійне спрямування підготовки бакалаврів за освітньою програмою «Геодезія, картографія та землеустрій».

Основні дисципліни, що забезпечують вивчення дисципліни «Основи геоінформаційних систем і бази даних», і дисципліни, що забезпечуються вивченням цієї дисципліни, подані на рисунку 1.

У посібник включено навчальний матеріал, який необхідний студентам для якісного вивчення таких наукоємних дисциплін: «Проектування баз геоданих», «Програмування геоінформаційних задач», «ГІС аналіз», «Геоінформаційні технології», «Дистанційне зондування землі».

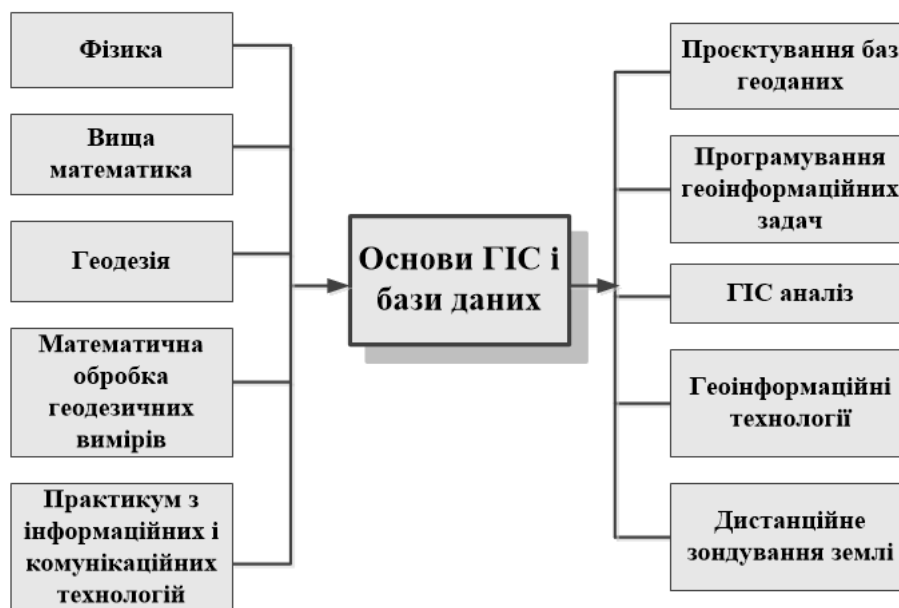


Рисунок 1 – Місце дисципліни «Основи геоінформаційних систем і бази даних» у навчальному процесі

Підручник містить чотири розділи:

1. Проектування та створення реляційних баз даних.
2. Будова та функції геоінформаційних систем.
3. Принципи організації та зберігання даних у геоінформаційних системах.
4. Завдання на практичні заняття та індивідуальне завдання.

У перший розділ включено загальні відомості про бази даних і

системи керування базами даних (СКБД), основні теоретичні поняття й терміни баз даних, розглянуто склад реляційної та інфологічної моделей даних. Теоретичний матеріал та основні поняття баз даних подано з використанням простих і наочних прикладів, які містять інформацію про перебіг навчального процесу у вищому навчальному закладі. Ці приклади добре розуміє кожен студент.

У другому розділі розглянуто геоінформаційні системи як технологічний комплекс, що інтегрує та поєднує багато інформаційних технологій, зокрема просторові бази даних під керуванням універсальних СКБД.

У третьому розділі розглянуто, в який спосіб дані реального світу, які відображаються в ГІС, тобто просторові, часові та тематичні найкраще чином зберігати у реляційних базах даних. Подано основні підходи до того, як враховувати просторовий аспект, який пов'язаний з визначенням місця розташування об'єкта, часовий та тематичний аспекти, які обумовлені виділенням одних ознак об'єкта й виключенням із розгляду інших.

Четвертий розділ присвячено виконанню практичних занять та індивідуальному навчально-дослідному завданню студента. Подано варіанти завдань, основні вимоги до створюваних баз даних та наведено приклад виконання індивідуального завдання.

Професійні компетентності, що формуються під час вивчення навчальної дисципліни «Бази даних».

У процесі навчання студенти отримують необхідні знання під час лекційних занять, закріплюють і поглиблюють знання, набувають практичних навичок та вмінь у процесі виконання практичних робіт. Особливе значення має самостійна робота студентів, під час виконання якої вони самостійно розробляють індивідуальний проєкт прикладної бази даних. У процесі роботи над проєктом студенти набувають навичок роботи з науково-технічною літературою, навчаються самостійно приймати рішення та робити висновки. У результаті засвоєння матеріалу підручника у студента повинні сформуватися такі компетентності.

Проєктні, що пов'язані з використанням основних принципів побудови реляційних баз даних, розробки та створення форм для введення даних і їх аналізу, розробки та створення запитів, тобто вміти створювати базу даних ГІС-проєкту.

Аналітичні, що пов'язані з використанням основних принципів аналізу отриманих на певних стадіях ГІС-проєкту даних, з використанням прикладних пакетів для аналізу одержаних відомостей засобами персональних комп'ютерів, застосуванням та самостійним вибором і освоєння нових програмних продуктів.

Управлінські, що пов'язані із застосуванням засобів сполучення різних прикладних пакетів для комплексної обробки геоінформаційних даних, зокрема в разі віддаленого доступу до бази даних, та створенням супровідної документації.

Принципи, які покладено в основу побудови навчального посібника.

В основу побудови посібника покладено принципи системності та практичної спрямованості з використанням сучасного програмного забезпечення.

Згідно з принципом системності практичні заняття надано в такому порядку: формулюється проблема, задачі, методи та засоби їх розв'язання. Матеріал викладається за принципом «від простого – до складного». Такий підхід сприяє формуванню у студентів системних знань і розвитку у них творчого мислення.

Принцип практичної спрямованості передбачає фундаментальну наукову підготовку й активне практичне навчання студентів. Підручник містить як основні наукові уявлення, так і створює умови для формування навичок і вмінь за допомогою практичної форми навчання. У підручник включено теоретичні основи розробки інфологічних і реляційних моделей даних, методи проєктування баз даних та практичні підходи до створення реляційних баз даних із використанням засобів СКБД **MS Access**. У роботі розглядається російськомовна версія **MS Access**, тому назви елементів управління, меню, команд і відповідних їм діалогових вікон наводяться російською мовою. Практичні заняття з геоінформаційних систем надано на базі однієї з найпоширеніших ГІС у світі – **ArcGis** від американської компанії **Esri**. Тому інтерфейс цього програмного засобу використовується англійський.

У підручнику зміст практичного навчання реалізовано на базі методу активної рефлексії. Застосування цього методу припускає вивчення й обмірковане повторення студентами операцій, які необхідні для досягнення поставленої мети. При викладанні матеріалу дається короткий

опис теоретичних основ, аналіз способів застосування отриманого знання та детально розглядається самостійне виконання. Завдяки цьому студенти навчаються виконувати завдання самостійно з використанням матеріалу підручника.

Важливим є те, що в практичній частині підручника використовується єдина задача, яка має практичну спрямованість за напрямом підготовки. Студенти мають можливість прослідкувати процес створення проєкту від постановки задачі до її розв'язання. Формування навичок може здійснюватися як під керівництвом викладача в аудиторії, так і вдома шляхом самостійного створення індивідуальних проєктів на основі розглянутої методики розробки проєкту.

При відборі матеріалу враховувалися обмеження, які накладає на навчальний процес час навчання. У підручник включено найважливіші аспекти наукового та технічного знання. У виданні подано глосарій зі списком термінів, список рекомендованої літератури, а також теми індивідуальних науково-дослідних завдань і розглянуто методику їх виконання на прикладі туристичного бізнесу. За межами посібника залишився матеріал, який студенти можуть самостійно освоїти в процесі практичної роботи з рекомендованою літературою та довідкою програмного забезпечення.

Підручник апробований під час викладення дисципліни «Основи геоінформаційних систем і бази даних» студентам Харківського національного університету міського господарства імені О. М. Бекетова. У процесі викладення матеріалу навчального посібника використовуються **навігаційні підказки** у вигляді позначок, які допоможуть зорієнтуватися у структурі навчального посібника:

– найважливіші теоретичні відомості;

напівжирне написання – терміни програмного забезпечення;

курсивне написання – назви, які вводить студент;

* – завдання підвищеної складності.

Для визначення рівня засвоєння матеріалу кожного підрозділу пропонуються питання для самоконтролю. Повне засвоєння матеріалу цього підручника допоможе розвинути здатність до подальшого навчання, самостійного розвитку та оволодіти інструментальними засобами для виконання прикладних завдань геоінформаційного характеру.

РОЗДІЛ 1

ПРОЄКТУВАННЯ ТА СТВОРЕННЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ

У цьому розділі розглядаються: концепція побудови баз даних, основні теоретичні поняття баз даних, системи керування базами даних (СКБД), вимоги до бази даних та СКБД, види моделей даних та їхнє призначення, склад реляційної та інфологічної моделей даних. Основні прийоми створення та редагування запитів та форм у MS Access.

1.1 Концепція побудови баз даних

Об'єкт, предмет і методологія баз даних

Історія розвитку баз даних

Історія розвитку баз даних нараховує понад 40 років. Можна виокремити такі етапи розвитку баз даних: використання файлів і файлових систем, перехід до концепції баз даних, використання баз даних на персональних комп'ютерах і створення розподілених баз даних.

Файли та файлові системи

В історії обчислювальної техніки можна прослідкувати розвиток двох основних сфер її використання. Перша сфера – застосування електронно-обчислювальних машин (ЕОМ) для виконання чисельних розрахунків. Характерною особливістю цієї сфери використання ЕОМ є наявність складних алгоритмів обробки, які застосовуються до простих за структурою даних, об'єм яких порівняно невеликий. Друга сфера – використання ЕОМ в автоматичних або автоматизованих інформаційних системах. **Інформаційна система** є програмно-апаратним комплексом, що забезпечує виконання таких функцій:

- надійне зберігання інформації в пам'яті комп'ютера;
- виконання специфічних для конкретного застосування перетворень інформації та обчислень;
- надання користувачам зручного й легко освоюваного інтерфейсу.

Інформаційні системи мають справу з великими обсягами інформації, яка має складну структуру. Прикладами інформаційних систем є банківські системи, автоматизовані системи управління підприємствами, системи резервування авіаційних або залізничних квитків.

У традиційних пакетних системах обробки інформації, які

функціонували донедавна на машинах типу ЄС ЕОМ, дані були організовані у вигляді непов'язаних між собою локальних інформаційних файлів лінійної структури. Суть такого підходу до організації інформаційного забезпечення полягає в тому, що файли проєктуються окремо для кожної конкретної задачі чи для їх комплексів. Такі системи називаються файловими.

Незважаючи на відносну простоту організації, файлові системи мають низку недоліків, головними з яких є такі.

1. Надлишковість даних. Файлові системи мають значну надлишковість, оскільки нерідко для розв'язування різних задач управління використовуються одні й ті самі дані, розміщені в різних файлах. Через дублювання даних у різних файлах зовнішня пам'ять використовується неекономно, інформація одного й того самого об'єкта управління розподіляється між багатьма файлами. При цьому досить важко уявити загальну інформаційну модель предметної області.

2. Неузгодженість даних. Зважаючи на те, що одна й та сама інформація може розміщуватися в різних файлах, технологічно важко простежити за внесенням змін одночасно в усі файли. Через це може виникнути неузгодженість даних, коли одне й те саме поле в різних файлах може мати різні значення.

3. Залежність структур даних і прикладних програм. При файловій організації логічна й фізична структури файлу мають відповідати їхньому опису в прикладній програмі. Прикладна програма повинна бути модифікована при будь-якій зміні логічної чи фізичної структури файлу. Оскільки зміни в одній програмі зазвичай потребують внесення змін до інших інформаційно пов'язаних програм, то інколи простіше створити нову програму, ніж вносити зміни в стару. Тому цей недолік файлових систем призводить до значного збільшення вартості супроводження програмних засобів. Зазначимо, що інколи вартість супроводження програмних засобів може досягати близько 70 % від вартості їх розробки.

Розглянуті недоліки послугували тим поштовхом, який змусив розробників інформаційних систем запропонувати новий підхід до управління інформацією. Таким підходом стала концепція баз даних.

Основні підходи до побудови баз даних

Складниками концепції баз даних є сукупність принципів і методів

проектування та використання баз даних. У межах цієї концепції для зберігання інформації про предметну область застосовуються: база даних – поймає та структурована сукупність взаємопов'язаних даних і нова програмна система – система керування базами даних (СКБД).

У 1963 році С. Бахманом була побудована перша промислова база даних IDS із мережевою моделлю даних, яка все ще характеризувалася надлишковістю даних та їх використанням тільки для однієї програми. Доступ до даних здійснювався за допомогою відповідного програмного забезпечення. У 1969 році сформувалася група, яка створила набір стандартів **CODASYL** для мережевої моделі даних. Фактично почала використовуватися сучасна архітектура баз даних.

Суттєвий стрибок у розвитку технології баз даних дала парадигма реляційної моделі даних, яку запропонував Е. Ф. Кодд у 1970 році. Під **парадигмою** розуміється наукова теорія, втілена в систему понять, що відображають суттєві риси дійсності. Тепер логічні структури могли бути отримані з одних і тих самих фізичних даних, тобто доступ до одних і тих самих фізичних даних міг здійснюватися різними додатками та різними шляхами. Стало можливим забезпечення цілісності й незалежності даних.

Бази даних на персональних комп'ютерах

Персональні комп'ютери (ПК) нестримно увірвалися в наше життя та буквально перевернули наше уявлення про місце та роль обчислювальної техніки в житті суспільства. Комп'ютери стали ближчими кожному користувачеві. Вони мають безліч програм, призначених для роботи непідготовлених користувачів. Програми ПК (редактори текстів, електронні таблиці та інші) стали простими у використанні й інтуїтивно зрозумілими.

З'явилися системи керування базами даних, які дозволяли створювати бази даних для зберігання значних об'ємів інформації. Вони мали зручний інтерфейс для заповнення даних та вбудовані засоби для генерації різних звітів. Ці програми дозволяють автоматизувати багато облікових функцій, які раніше велися вручну. Прикладом використання інформаційних систем може бути інформаційна система будь-якого супермаркету.

Ці бази даних мають такі особливості:

– СКБД призначені для створення баз даних переважно з

монопольним доступом, оскільки комп'ютер не був приєднаний до мережі й база даних на ньому створювалася для роботи одного користувача.

– Більшість СКБД мають розвинений і зручний для користувача інтерфейс, а також інструментарій для розробки готових додатків без програмування. Інструментарій включає готові елементи додатка у вигляді шаблонів екранних форм, звітів та графічних конструкторів запитів.

– У настільних СКБД підтримувалися як високорівневі мови маніпулювання даними типу SQL, так і низькорівневі мови маніпулювання даними на рівні окремих рядків таблиць.

Розподілені бази даних

Відомо, що історія розвивається за спіраллю. Після процесу «персоналізації» почався зворотний процес – інтеграція. Створюються локальні мережі, і все більше інформації передається між комп'ютерами. Гостро постає завдання узгодженості даних, що зберігаються й обробляються в різних місцях, але логічно пов'язані один з одним. Виникають завдання, пов'язані з паралельною обробкою транзакцій – послідовностей операцій над базою даних, що переводять її з одного несуперечливого стану в інший несуперечливий стан. Вирішення цих завдань приводить до появи розподілених баз даних, які зберігають усі переваги настільних СКБД, дозволяють організувати паралельну обробку інформації й підтримку цілісності баз даних.

Більшість сучасних СКБД можуть працювати на комп'ютерах із різною архітектурою та під різними операційними системами. При цьому для користувачів доступ до даних, керованих СКБД на різних платформах, практично невиразний. Необхідність підтримки багатокористувальницької роботи з базою даних і можливість децентралізованого зберігання даних зажадали розвитку засобів адміністрування баз даних із реалізацією загальної концепції засобів захисту даних.

До цього етапу можна також віднести розробку стандартів із мов опису та маніпулювання даними, технологій з обміну даними між різними СКБД, наприклад, протокол **ODBC** (Open DataBase Connectivity) фірми **Microsoft**. Прикладами СКБД, які належать до цього етапу, є **MS Access** і всі сучасні сервери баз даних **Oracle**, **MS SQL**, **SQL Base** та інші сучасні сервери баз даних, яких зараз налічується декілька десятків.

Перспективи розвитку систем керування базами даних

Цей етап характеризується появою нової технології доступу до даних – Інтернету. Для роботи з видаленою базою даних використовується стандартний інтернет-браузер, наприклад **Microsoft Internet Explorer**, і для кінцевого користувача процес звернення до даних відбувається аналогічно пошуку у Всесвітній павутині. При цьому вбудований у завантажувачем HTML-сторінку код, який написаний на мові **Java, Java-Script, Perl** та інших, відстежує всі дії користувача й транслює їх у SQL-запити до бази даних. Зручність цього підходу привела до того, що він почав використовуватися не тільки для видаленого доступу до баз даних, а й для користувачів локальної мережі підприємства. Прості завдання обробки даних, які не пов'язані зі складними алгоритмами, досить просто й ефективно можуть бути побудовані за цією архітектурою. Для підключення нового користувача до можливості використання конкретного завдання не потрібна установка додаткового клієнтського програмного забезпечення. Проте алгоритмічно складні завдання рекомендується реалізовувати в архітектурі «клієнт – сервер» із розробкою спеціального клієнтського програмного забезпечення.

У кожного з перелічених вище підходів до роботи з даними є свої переваги і недоліки, які й визначають сферу застосування того або іншого методу. На сьогодні всі підходи досить широко використовуються.

Отже, використання концепції баз даних дозволяє забезпечити: надійність, простоту й легкість використання та збереження даних, стандартизувати дані в межах однієї предметної області, а також скоротити дублювання інформації шляхом структуризації даних.

Основні поняття баз даних, терміни та визначення

Визначимо основні поняття баз даних, а саме: інформаційна система, об'єкт, предмет, предметна область, дані, модель даних, база даних, система керування базами даних. Інші поняття розглянемо в процесі викладення навчального матеріалу.

Інформаційна система – це програмно-апаратний комплекс, функції якого полягають у надійному збереженні інформації, наданні користувачу зручного інтерфейсу та виконанні специфічних операцій з перетворення та пошуку необхідної інформації.

Об'єкт – це явище зовнішнього світу. Це або щось реально

існуюче – людина, товар, виріб, або процес – облік народжуваності, отримання товарів, випуск виробів.

Властивість об'єкта. Кожен об'єкт має певні властивості. Наприклад, об'єкт «Людина» володіє такими властивостями: зріст, ім'я, дата народження, а об'єкт «Виріб» – такими властивостями: якість, дата виготовлення, зовнішній вигляд.

Зв'язки між об'єктами. Існує безліч зв'язків між об'єктами. Наприклад: **людина** купує, продає, створює **виріб**; **виріб** створюється, купується, продається **людиною**. Серед усієї множини властивостей і взаємозв'язків між об'єктами має сенс виділяти лише істотні, важливі з погляду споживача інформації.

Предмет – це модель реального об'єкта, в якому зафіксовані лише виділені для інформаційної системи властивості та зв'язки. Сукупність відібраних предметів утворює об'єктне ядро предметної області, а сукупність їхніх взаємозв'язків – структуру фрагмента дійсності.

Предметна область є абстрактною картиною реальної дійсності, певна частина якої фіксується як модель фрагмента дійсності. Поняття «Предметна область» відповідає точці зору споживача на об'єктне ядро. Виділяються тільки ті об'єкти, властивості об'єктів та зв'язки між об'єктами, які мають певну цінність і повинні бути збережені в базі даних. У кожен момент часу предметна область знаходиться в одному зі станів, який характеризується сукупністю об'єктів і їхніх взаємозв'язків. Із часом одні об'єкти зникають, інші з'являються, змінюються властивості та взаємозв'язки. Проте нові стани, що виникають, вважаються станами тієї самої предметної області. Отже, предметну область доцільно розглядати як систему, що складається з певної послідовності станів.

Дані – це інформація, подана у формалізованому вигляді, придатному для пересилання, інтерпретування чи оброблення за участю людини або автоматичними засобами.

Модель даних – це інтегрований набір понять для опису даних, зв'язків між ними й обмежень, що накладаються на дані. На основі моделі даних будується база даних.

База даних – це поійменована і структурована сукупність взаємопов'язаних даних, що відображає стан об'єктів та їхніх відносин у певній предметній області. База даних становить інтегроване сховище даних, яке використовується багатьма споживачами та забезпечує

незалежність даних від прикладних програм.

Архітектура бази даних – це різновид (узагальнення) структури, в якій будь-який елемент може бути замінений на інший елемент, характеристики входів і виходів якого ідентичні першому елементу.

Система керування базами даних (СКБД) – це комплекс програмних і мовних засобів загального та спеціального призначення, які необхідні для створення бази даних, підтримування її в актуальному стані, маніпулювання даними й організації доступу до них різних користувачів в умовах прийнятої технології обробки даних. Система керування базами даних виступає посередником між користувачем і базою даних.

Принцип організації зв'язку кінцевих користувачів і прикладних програм із базою даних показано на рисунку 1.1.

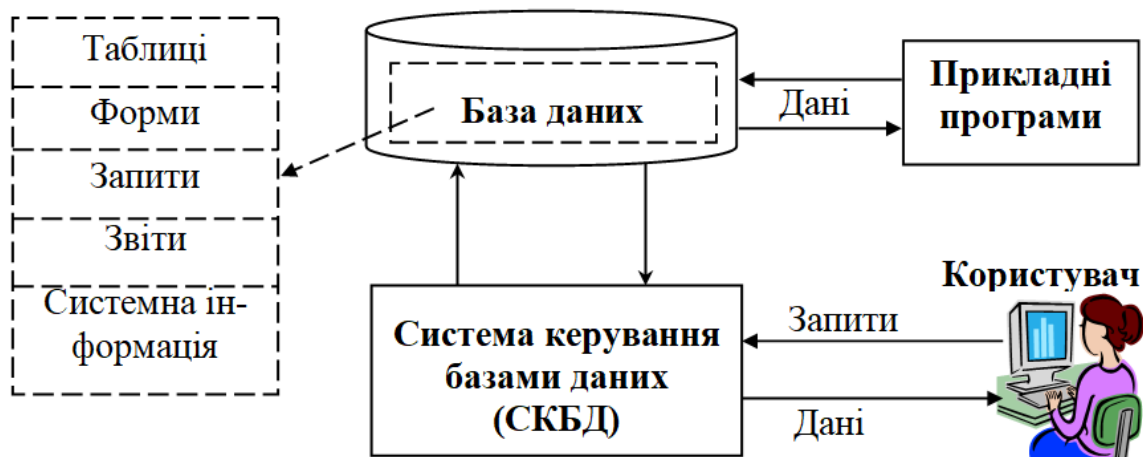


Рисунок 1.1 – Взаємодія користувача та прикладних програм із базою даних

Виконання інформаційних завдань зводиться переважно до пошуку необхідної інформації в базі даних за запитам користувачів. У запитах зазначаються умови пошуку та відображення інформації про конкретні об'єкти, наприклад, про наявність залишків товару на кінець залікового періоду роботи магазину в разі аналізу господарської діяльності.

Методологія та архітектура баз даних

Методологія – це сукупність методів вирішення проблеми. Методологія бази даних визначається в процедурі проектування, але проявляється і в процедурі використання.

Методологія проектування баз даних – це єдина сукупність правил, засобів і процедур, які дають можливість скоротити час і поліпшити якість проектування. Існує багато різновидів методології розгляду баз даних у класичному підході. Однак найчастіше дотримуються методології ANSI/SPARC, згідно з якою архітектура бази даних має три рівні. На рисунку 1.2 показана трирівнева архітектура та етапи проектування бази даних.

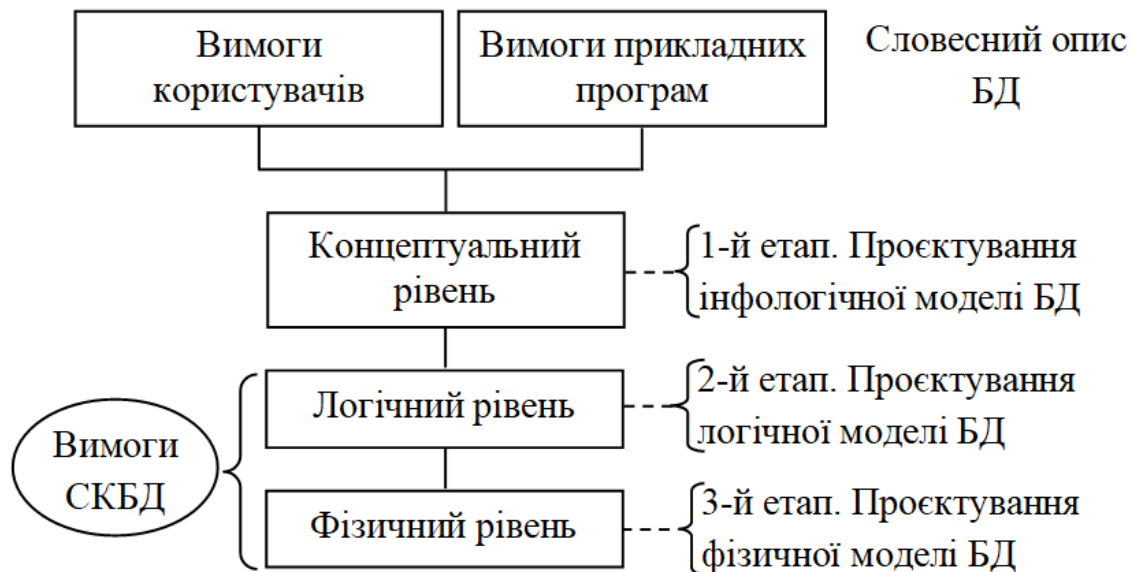


Рисунок 1.2 – Трирівнева архітектура баз даних

На етапі словесного опису бази даних (БД) здійснюється формулювання й аналіз вимог, встановлюються цілі організації та визначаються вимоги до бази даних. Вони складаються з вимог користувачів і специфічних вимог прикладних програм. Усі вимоги документуються у формі, доступній кінцевому користувачу та проектувальнику бази даних.

Перший етап проектування відповідає концептуальному рівню бази даних. На цьому етапі формується інформаційно-логічна (інфологічна) модель бази даних, у якій подається семантика предметної області. Результатом цього етапу є загальне та наочне подання інформаційних вимог користувачів.

Другий етап проектування відповідає логічному рівню бази даних. У процесі логічного проектування здійснюється перетворення інфологічної моделі бази даних у логічну модель, яка враховує особливості

використовуваної СКБД. Основною метою етапу є усунення надмірності даних із застосуванням спеціальних правил нормалізації. Мета нормалізації – мінімізувати повторення даних і можливі структурні зміни бази даних при процедурах оновлення. Отримана логічна структура бази даних може бути оцінена кількісно за допомогою різних характеристик (кількість звернень до логічних записів, обсяг даних у кожному додатку, загальний обсяг даних). На основі цих оцінок логічна структура може бути вдосконалена з метою досягнення більшої ефективності.

На етапі фізичного проектування вирішуються питання, що пов'язані з продуктивністю системи, визначаються структури зберігання даних та методи доступу. На цьому етапі здійснюється тестування бази даних відповідно до вимог кінцевих користувачів.

Процедура управління базою даних є найпростішою в однокористувальницькому режимі. У багатокористувальницькому режимі та в розподілених базах даних процедура значно ускладнюється. При одночасному доступі декількох користувачів без застосування спеціальних заходів можливе порушення цілісності. Для усунення цього явища використовують систему транзакцій і режим блокування таблиць або окремих записів. **Транзакція** – це процес зміни файлу, запису або бази даних, викликаний передачею одного вхідного повідомлення.

Процес проектування є тривалим і трудомістким. Основними ресурсами проектувальника бази даних є його власна інтуїція та досвід, тому якість рішення може виявитися низькою. Основними причинами низької ефективності проєктованих баз даних можуть бути:

- недостатньо глибокий аналіз вимог (початковий етап проєктування), зокрема їхньої семантики та взаємозв'язку даних;
- велика тривалість процесу структуризації, що робить цей процес утомливим і важко виконуваним при ручній обробці.

У цих умовах важливого значення набувають питання автоматизації розробки.

Методологія використання баз даних

База даних використовується зазвичай не самотійно, а є складником різних інформаційних систем: банків даних, інформаційно-пошукових і експертних систем, систем автоматизованого проєктування, автоматизованих робочих місць, автоматизованих систем управління.

У базі даних є три рівні подання бази даних (див. рис. 1.2): концептуальний, логічний і фізичний. У процедурі використання найчастіше мають справу з логічною моделлю та значно рідше з концептуальною та фізичною моделями.

Узагальнене подання всіх трьох рівнів містить словник даних. Словник даних є як би внутрішньою базою даних, що містить централізовані відомості про всі типи даних, їхні імена, структуру, а також інформацію про їх використання. Перевагою словника даних є ефективне накопичення та управління інформаційними ресурсами предметної області. Його застосування дозволяє зменшити надмірність і суперечність даних при їх введенні, здійснити просте та ефективне управління при їх модифікації, спростити процедуру проєктування бази даних за рахунок централізації управління даними, встановити зв'язки з іншим користувачами.

У процесі використання баз даних є операції оновлення (запис, видалення, модифікація даних) і запит-відповідь (читання).

У загальному випадку процес запиту складається з низки етапів. Користувач повинен знати структуру бази даних або звернутися до адміністратора бази даних. На першому етапі користувач має з'ясувати, які форми документів йому потрібні. Це можуть бути не тільки логічні моделі користувача, але й різні їхні модифікації при різних поєднаннях полів. Оскільки логічні, а тим більше модифіковані логічні, моделі можуть відрізнитися від логічної моделі бази даних, потрібно визначити, які поєднання полів необхідні для машинних документів, що виводяться. Ці поєднання утворюються за допомогою елементарних правил, що вивчаються реляційною алгеброю та реляційним численням. Далі правила потрібно трансформувати у відповідні варіанти звернення до СКБД через її інтерфейс. Це можуть бути меню, екранні форми, мова запитів, запит за прикладом, режим перегляду таблиць бази даних. Результат може бути поданий у вигляді таблиць або звітів.

У процесі експлуатації бази даних використовують дві специфічні операції: навігацію та специфікацію.

Навігація – це операція, результат якої представлений єдиним об'єктом, отриманим при проходженні шляху по логічній структурі (сукупності таблиць, зв'язок між якими здійснюється за ключами) бази даних.

Специфікація – це операція, результатом якої є нова структура (таблиця), побудована на основі структур таблиць бази даних. Вона має назву «вигляд».

Поняття схеми бази даних, стандарт SPARC

Схема бази даних – це загальний опис бази даних. Існують різні типи схем бази даних, які визначаються відповідно до рівнів абстракції багаторівневої архітектури, показаної на рисунку 1.2. На кожному рівні використовується своя мова опису схеми бази даних.

Зовнішній рівень є словесним описом вхідних і вихідних повідомлень, а також даних, які доцільно зберігати в базі даних. При проектуванні бази даних необхідно на зовнішньому рівні вивчити функціонування об'єкта керування, для якого проектується база даних, усю первинну та вихідну документацію для визначення даних, які потрібно зберігати в базі даних, а також вимоги користувачів і прикладних програмістів. На цьому рівні не виключена наявність елементів дублювання, надлишковості та неузгодженості даних. Для усунення цих аномалій і протиріч зовнішнього опису даних виконується інфологічне та логічне проектування.

Інфологічний рівень становить інформаційно-логічну модель предметної області, в якій виключено надлишковість даних і відображено інформаційні особливості об'єкта керування без урахування особливостей і специфіки конкретної СКБД. Інфологічне подання даних орієнтоване переважно на людину, яка проектує чи використовує базу даних. На етапі проектування інфологічної моделі будується концептуальна схема, яка є структурованою моделлю бази даних без урахування особливостей СКБД. Для розробки концептуальної схеми широко застосовується метод «сутність-зв'язок».

Логічний рівень урахує специфіку й особливості конкретної СКБД. Цей рівень подання даних орієнтований більше на комп'ютерну обробку та на програмістів, які займаються її розробкою. Проектування логічної моделі здійснюється шляхом перетворення інфологічної моделі даних у логічну модель. Результат цього етапу – логічна схема бази даних у вигляді взаємопов'язаних таблиць. Основною метою етапу є усунення надмірності даних із використанням спеціальних правил нормалізації та мінімізація можливих структурних змін бази даних.

На фізичному рівні вирішуються питання, що пов'язані з продуктивністю системи, створюються таблиці та інші об'єкти бази даних і здійснюється тестування бази даних відповідно до вимог кінцевих користувачів. Результатом цього етапу є схема бази даних у вигляді взаємопов'язаних таблиць.

Значення баз даних у побудові інформаційних систем

Нині – в епоху інформаційного вибуху – розроблюється та впроваджується велика кількість найрізноманітніших автоматизованих інформаційних систем (АІС) із дуже широким спектром використання.

У науковій літературі існує досить значне розмаїття класифікацій інформаційних систем (ІС). Різні автори залежно від своїх завдань та точок зору виокремлюють ті чи інші критерії та розподіляють пріоритети між ними. Зупинимось на одному з таких підходів, який, на погляд авторів, найбільше узгоджується з іншими темами цієї дисципліни. Отже, ІС класифікуються:

- за призначенням – фактографічні, документальні та змішані;
- за мовами – замкнуті системи, системи з базовою мовою та змішані;
- за локалізацією – локальні й розподілені;
- за структурами даних – ієрархічні, мережного типу та реляційні.

Розглянемо кожний із критеріїв.

Призначення ІС. Документальні системи зорієнтовані на обробку та зберігання документа (порівняно великої за розміром послідовності символів), внутрішню структуру якого система (майже) повністю ігнорує, тобто він неподільний (атомарний) з точки зору системи. Споживачем результатів пошуку є зазвичай, кінцевий користувач.

Фактографічні системи оперують фактами (даними) різних типів, що пов'язані в системі в порівняно складні структури. Дані, що є результатом пошуку, можуть стати складовою частиною звітів або використовуються різноманітними обчислювальними процесами.

Змішані системи включають у себе в тих чи інших пропорціях риси обох названих вище варіантів. Переважну більшість сучасних систем для персональних комп'ютерів варто віднести до категорії змішаних.

Мови ІС. Системи з базовою мовою передбачають взаємодію користувача із СКБД із середовища якоїсь іншої мови програмування, де і

виконується більшість постпошукових перетворень даних. Такий підхід зручний для розробки різних систем як надбудов над СКБД, бо дає можливість створювати високоефективні програми постпошукової обробки даних. Замкнуті системи самостійно забезпечують користувача всіма необхідними засобами як для локалізації даних, так і для їх постпошукової чи передпошукової обробки. Недоліком таких систем є те, що в них відсутні (або малоефективні) засоби для розробки надбудов – проблемно-орієнтованих комплексів.

Змішані системи передбачають наявність обох можливостей двох попередніх підходів і є найпоширенішими на сьогодні.

Локалізація ІС. Локальність передбачає розташування всього програмного забезпечення й даних на одному ізольованому комп'ютері, а розподіленість означає розташування системи на мережі комп'ютерів із певною стратегією рознесення даних.

Структури даних. Структури даних, що підтримуються в системі бази даних, – це важливий фактор, який впливає як на виразову потужність, так і на ефективність функціонування.

Для систем з ієрархічною структурою базовою структурою даних є дерево. Зазвичай, вони мають найвищу ефективність функціонування, але виразові можливості їх порівняно низькі.

Системи зі структурами даних типу «мережа» мають значно кращі виразові можливості, але дещо програють у ефективності функціонування, точніше, від користувача вимагається значно вищий рівень кваліфікації для ефективної експлуатації таких систем.

Останніми десятиріччями найрозповсюдженішими, особливо для персональних комп'ютерів, вважаються СКБД реляційного типу, яким притаманні щонайпростіша структура даних (плоский файл). Але одночасно суттєво підвищується рівень мов маніпулювання даними, що максимально употужнює виразові можливості та знижує ефективність функціонування. Для таких систем потрібні потужні комп'ютери, і вони значно чутливіші (порівняно з попередниками) до зростання обсягів даних.

Система керування базами даних як спеціальний програмний комплекс

Склад і функції систем керування базами даних

Усі функції СКБД можна об'єднати в такі групи:

Керування даними. Завданнями керування даними є підготовка даних та їх контроль, занесення даних до бази, структуризація даних, забезпечення цілісності та безпеки даних.

Доступ до даних. Пошук і селекція даних, перетворення їх у форму, зручну для подальшого використання.

Організація та ведення зв'язку з користувачем. Ведення діалогу, видача діагностичних повідомлень про помилки в роботі з базою даних (рис. 1.3).

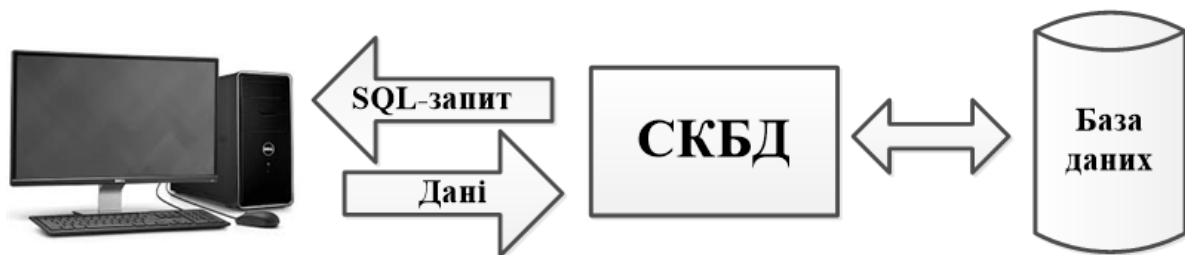


Рисунок 1.3 – Організація та ведення зв'язку з користувачем

Для опису даних, організації спілкування та виконання процедур пошуку й різних перетворень із даними застосовуються мовні засоби СКБД. У склад мовних засобів входять:

- мова опису даних;
- мова маніпулювання даними, що забезпечує виконання основних операцій над даними – введення, модифікацію та вибірку;
- мова структурованих запитів SQL (Structured Query Language), що забезпечує управління структурою бази даних і маніпулювання даними, а також є стандартним засобом доступу до видалених баз даних;
- мова запитів за зразком QBE (Query By Example), що забезпечує візуальне конструювання запитів до бази даних.

Мова опису даних призначена для опису даних на різних рівнях абстракції. Опис усіх імен, типів і розмірів полів зберігається в пам'яті разом із даними.

Мова маніпулювання даними може бути базовою чи автономною. Базова мова – це одна з традиційних мов програмування (БЕЙСІК, С та ін.). Системи, що використовують базову мову, називають відкритими. Автономна мова маніпулювання даними – це власна мова СКБД, яка дає змогу виконувати різні операції з даними. У сучасних СКБД для спрощення процедур пошуку даних у базі даних передбачено такі мови

запитів, як SQL і QBE. Мова запитів QBE – це реалізація запитів за зразком у вигляді таблиць. Для визначення запиту до бази даних користувач повинен заповнити надану системою таблицю QBE і визначити в ній критерії пошуку та вибору даних.

Також було розроблено стандартний інтерфейс мови CLI (Common Language Interface) для всіх основних варіантів мови SQL. Цей інтерфейс був формалізований фірмою Microsoft і дістав назву ODBC (Open Database Connectivity – відкритий доступ до даних). ODBC – це інтерфейс доступу до даних засобами СКБД. Він має цілий набір драйверів, за допомогою яких одна СКБД може працювати з даними інших систем.

Види систем керування базами даних

За характером застосування розрізняють персональні та багатокористувальницькі СКБД.

Персональні СКБД забезпечують можливість створення локальних баз даних, що працюють на одному комп'ютері. До персональних СКБД належать MS Access, Paradox, FoxPro та ін. Більшість персональних СКБД мають розвинений і зручний для користувача інтерфейс, а також інструментарій для розробки готових додатків без програмування. Інструментарій включає готові елементи додатка у вигляді шаблонів екранних форм, звітів, графічних конструкторів запитів. У таких СКБД підтримуються як високорівневі мови маніпулювання даними типу SQL, так і низькорівневі мови маніпулювання даними на рівні рядків таблиць.

Багатокористувальницькі СКБД дозволяють створювати інформаційні системи, що функціонують в архітектурі «клієнт – сервер». Найвідомішими СКБД є Oracle, Microsoft SQL Server, InterBase. Багатокористувальницькі СКБД зберігають усі переваги згаданих вище СКБД та дозволяють організувати паралельну обробку інформації. Більшість сучасних СКБД можуть працювати на комп'ютерах із різною архітектурою та під різними операційними системами. Необхідність підтримки багатокористувальницької роботи з базою даних і можливість децентралізованого зберігання даних привели до розвитку засобів адміністрування баз даних із реалізацією загальної концепції засобів захисту даних.

Вимоги до бази даних і СКБД

Проектування баз даних – це ітераційний, багатоетапний процес

прийняття обґрунтованих рішень у процесі аналізу інформаційної моделі предметної області, вимог до даних із боку прикладних програмістів і користувачів, синтезу логічних і фізичних структур даних, аналізу та обґрунтування вибору програмних і апаратних засобів. Етапність проектування даних пов'язана з багаторівневою організацією даних.

Згідно з пропозиціями дослідницької групи із систем керування даними Американського національного інституту стандартів ANSI/SPARC, як правило, виділяється три рівні подання даних: зовнішній (з точки зору кінцевого користувача та прикладного програміста), концептуальний (з точки зору СКБД) та внутрішній (з точки зору системного програміста). Згідно з цією концепцією зовнішній рівень є частиною концептуальної моделі, що необхідна для реалізації якогось запиту чи прикладної програми. Модель ANSI/SPARC не стала стандартом, проте вона все ще є основою для розуміння деяких функціональних особливостей СКБД.

На базі моделі ANSI/SPARC запропоновані й інші варіанти багаторівневого подання даних, у яких під зовнішнім рівнем розуміють загальні поняття, які пов'язані з вивченням і аналізом інформаційних потоків предметної області, а для структуризації даних вводиться допоміжний рівень, який називається інфологічним, або інформаційно-логічним. Зовнішній рівень у цьому випадку є окремим етапом проектування, на якому вивчається все позамашинне інформаційне забезпечення: форми документування та подання даних, а також зовнішнє середовище, в якому функціонуватиме база даних із точки зору методів фіксації, збирання та передавання інформації в базу даних.

Така концепція доцільніша з точки зору розуміння процесу проектування бази даних. Тому будемо дотримуватися багаторівневого подання даних, що включає зовнішній рівень і три рівні подання даних у базі даних на етапі проектування, створення та супроводження бази даних (див. рис. 1.2).

Мета багаторівневої архітектури полягає у відокремленні призначеного для користувача подання бази даних від її фізичного уявлення. Причини, з яких бажано виконувати це розділення, такі:

- кожен користувач повинен мати можливість звертатися до одних і тих самих даних, використовуючи своє власне уявлення про них, а також змінювати своє уявлення про дані, до того ж ця зміна не повинна впливати на інших користувачів;

- користувачі не повинні безпосередньо мати справи з подробицями фізичного зберігання даних у базі;
- адміністратор бази даних повинен мати можливість змінювати структуру зберігання даних у базі, не впливаючи на призначені для користувача представлення.

Види моделей даних та їхнє призначення

Моделі даних визначаються трьома компонентами: організацією даних, обмеженнями цілісності та багатофункціональністю припустимих операцій. У теорії систем керування базами даних виокремлюють моделі трьох основних типів: ієрархічну, мережеву й реляційну.

Ієрархічна модель даних. Термінологічною основою для ієрархічної та мережевої моделей є поняття: атрибут, агрегат та запис. Під атрибутом (елементом даних) розуміється найменша поійменована структурна одиниця даних. Поійменована множина атрибутів може утворювати агрегат даних. У деяких випадках окремо взятий агрегат може складатися з множини екземплярів однотипних даних, або, як ще говорять, бути множинним елементом. Нарешті, записом називають складовий агрегат, який не входить до складу інших агрегатів.

В ієрархічній моделі всі записи, агрегати й атрибути бази даних утворюють ієрархічно організований набір, який зручно зображувати за допомогою деревоподібного графа (рис. 1.4).

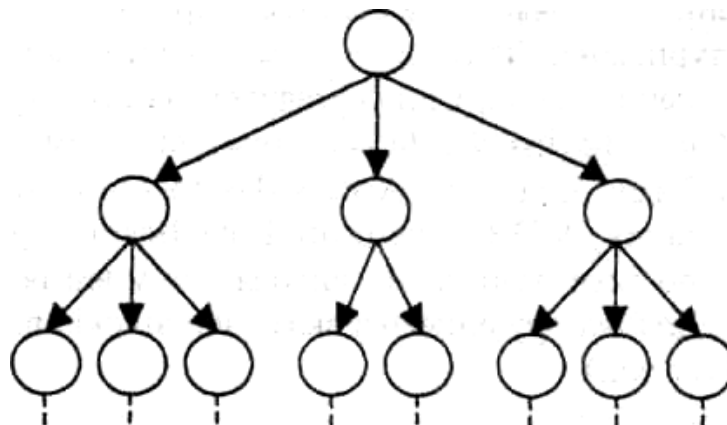


Рисунок 1.4 – Схема ієрархічної моделі даних

В ієрархічній моделі всі елементи пов'язані відношеннями підпорядкованості, і будь-який елемент може підкорятися тільки одному іншому елементу.

Мережева модель даних. Мережевий підхід до організації даних є розширенням ієрархічного. В ієрархічних структурах запис-послідовник повинен мати одного попередника; у мережевій структурі даних послідовник може мати будь-яку кількість попередників (рис. 1.5).

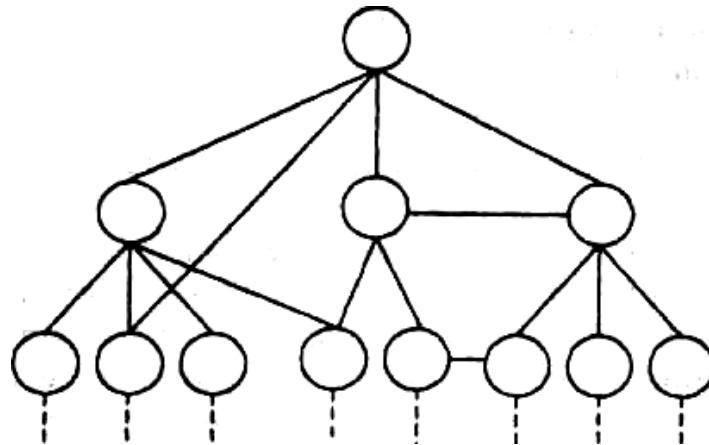


Рисунок 1.5 – Схема мережевої моделі даних

Серед переваг систем керування даними, заснованих на ієрархічній чи мережевій моделях, можуть бути їхня компактність і висока швидкість, а серед недоліків – високий ступінь залежності від конкретних даних.

Реляційна модель даних. Реляційна модель даних уперше була запропонована британським ученим, співробітником компанії ІВМ Едгаром Коддом. Сьогодні ця модель є фактичним стандартом, на який орієнтуються практично всі сучасні СКБД. У реляційній моделі досягається набагато вищий рівень абстракції даних, ніж в ієрархічній або мережевій моделях. Реляційна модель даних ґрунтується на понятті відношення, і становить собою безліч елементів, що називаються кортежами. Наочною формою подання відношення є звична для сприйняття двовимірною таблиця. Таблиця має рядки (записи) і стовпці (колонки). Кожний рядок таблиці має однакову структуру та складається з полів. Рядкам таблиці відповідають кортежі, стовпцям – атрибути відношення.

Створена Коддом теорія нормалізації відношень виявилася, по суті, єдиною формалізованою теорією, за допомогою якої можна спроектувати оптимальну логічну модель даних. Цю теорію можна використовувати не лише при проектуванні баз даних у середовищі реляційних СКБД, а й для СКБД, які підтримують інші моделі даних. Отже, будь-яку логічну модель спочатку проєктують як нормалізовану реляційну модель, а потім

відображують на ту модель, яку підтримує вибрана СКБД. Реляційна модель найширше використовується для створення баз даних різного призначення, зокрема економічного характеру.

Питання для самоконтролю

1. Дайте визначення понять «інформаційна система», «об'єкт», «предмет», «предметна область».
2. Дайте визначення поняттю «база даних».
3. Поясніть підходи до побудови баз даних.
4. Поясніть поняття схеми бази даних і стандарту SPARC.
5. Дайте визначення поняттю «система керування базами даних».
6. Дайте характеристику складу та функцій СКБД.
7. Охарактеризуйте мовні засоби СКБД. Для яких цілей вони використовуються?
8. Які вимоги ставляться до бази даних і СКБД?
9. Які види СКБД існують? Охарактеризуйте їх.
10. Які види моделей даних ви знаєте?
11. Охарактеризуйте основні моделі даних.

Реляційна модель даних

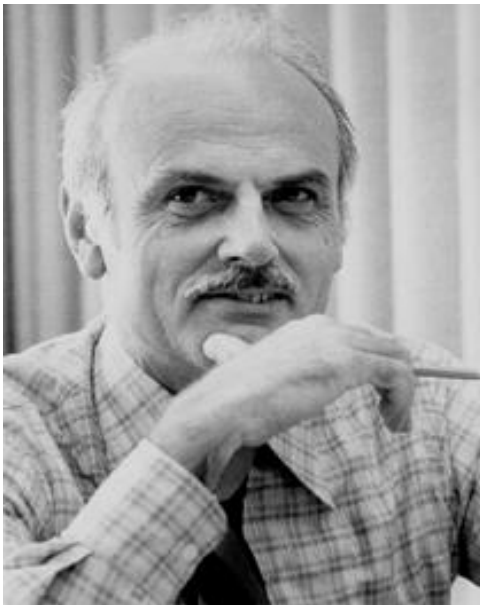


Рисунок 1.6 – Едгар Кодд

Концепцію реляційної моделі даних запропонував американський вчений Кодд Едгар Франк у 1970 році (рис. 1.6)

Едгар Франк Кодд народився 23 серпня 1923 року у м. Портланд (Англія). Освіту з математики та хімії отримав у Оксфордському університеті. З 1948 року працював у США в компанії ІВМ як математик-програміст. Створив теорію нормалізації відношень, а в 1985 році визначив 12 правил, яким повинна відповідати будь-яка база даних, що претендує на тип реляційної. Він зробив суттєвий внесок в інші сфери інформатики.

Помер Е.Ф. Кодд 18 квітня 2003 року.

Виникнення концепції реляційної моделі даних пов'язане з розв'язанням проблеми забезпечення незалежності даних і їхнього опису від прикладних програм. Створена Коддом теорія нормалізації відношень виявилася, по суті, єдиною формалізованою теорією, за допомогою якої можна спроектувати оптимальну логічну модель даних. Цю теорію можна використовувати не лише при проектуванні баз даних у середовищі реляційних СКБД, а й для СКБД, які підтримують інші моделі даних.

Отже, будь-яку логічну модель спочатку проєктують як нормалізовану реляційну модель, а потім відображають на ту модель, яку підтримує вибрана СКБД.

Основні поняття та складові частини реляційної моделі даних

Основні поняття реляційних баз даних: відношення, атрибут, схема відношення, тип даних, домен, кортеж, таблиця, первинний ключ, зв'язок, вторинний ключ, структурна, цілісна та маніпуляційна частини моделі даних.

Значення цих понять розглянемо з використанням прикладу відношення, яке містить інформацію про викладачів кафедри (рис. 1.6).

Відношення є фундаментальним поняттям і засобом структурування даних у реляційній моделі даних. Відношенням називається будь-який зв'язок між об'єктами або їхніми властивостями.

Відношення має ім'я, яке задається іменником. Ім'я повинне бути унікальним у базі даних і пояснювати зміст (семантику) відношення. Засобом подання відношення в реляційній базі даних є іменована таблиця.

Атрибути та схема відношення

Атрибут – це логічно неподільна одиниця даних, яка визначає певну властивість об'єкта або зв'язок. Кожний атрибут має ім'я, певний тип (рядок символів, число та ін.), значення та інші властивості.

Ім'я атрибута задається іменником в однині. Воно повинне бути унікальним у відношенні, простим і зрозумілим.

Значення атрибута – величина, що характеризує деяку властивість об'єкта або зв'язок. Значення має певний тип, формат, довжину та повинно бути атомарним (неподільним) для даної моделі. Приклади атрибутів та їхніх значень наведені на рисунку 1.7 (первинний ключ виділено

напівжирним написанням).

Структура відношення подається у вигляді схеми.

Схемою відношення є кінцева множина впорядкованих імен атрибутів – (A_1, A_2, \dots, A_n) . Опис відношення складається з його ім'я та схеми: $IM'Я(A_1, A_2, \dots, A_n)$, наприклад:

ВИКЛАДАЧ (**Таб_номер**, Прізвище, Зарплата, N_кафедри).

Для однозначної ідентифікації кортежів кожне відношення повинно мати первинний ключ. **Первинний ключ** складається з одного або декількох атрибутів. Наприклад, атрибут «**Таб_номер**» є первинним ключем відношення «ВИКЛАДАЧ» і визначає кортежі цього відношення.

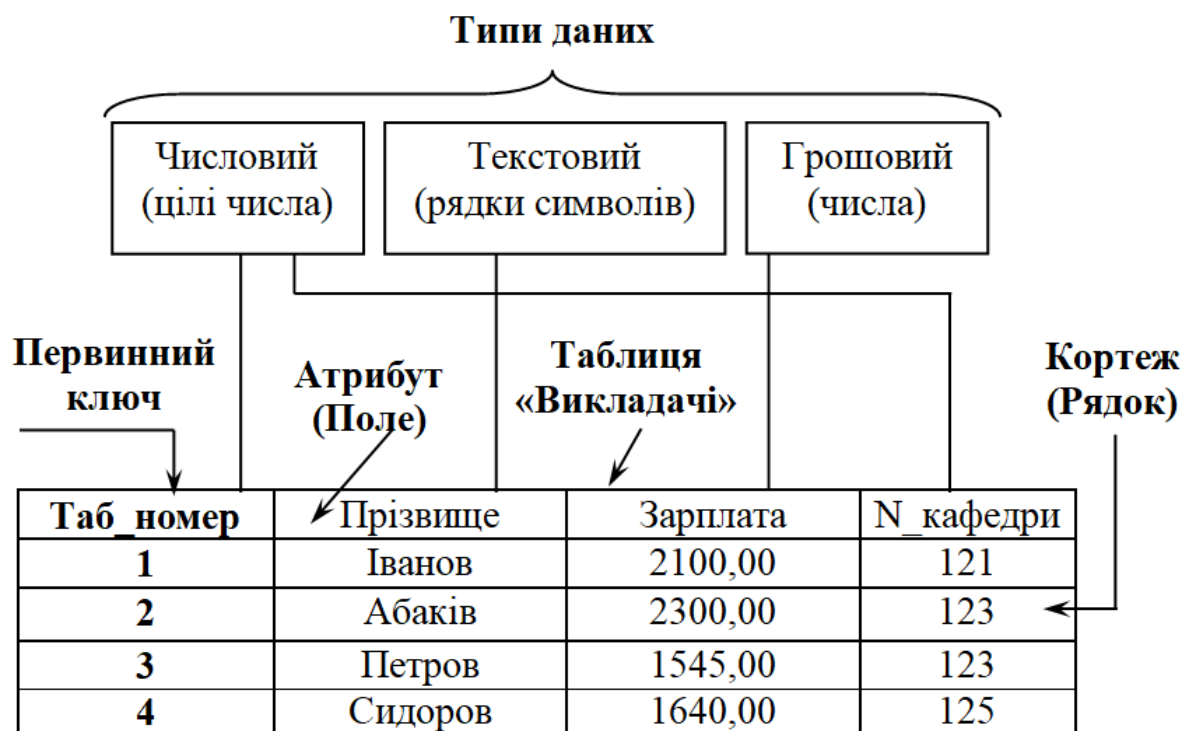


Рисунок 1.7 – Відношення з інформацією про викладачів кафедри

Об'єктні та зв'язкові відношення

Відношення можна поділити на два класи: об'єктні й зв'язкові. Об'єктні відношення зберігають дані про інформаційні об'єкти предметної області. Прикладом об'єктного відношення є відношення «ВИКЛАДАЧ». В об'єктному відношенні не повинно бути рядків з однаковим ключем.

Зв'язкове відношення містить ключі двох або більше об'єктних відношень. За цими ключами зв'язкове відношення встановлює зв'язки між відповідними об'єктними відношеннями. Наприклад, розглянемо два

об'єктних відношення:

СТУДЕНТ(Код_студента, Прізвище);
ДИСЦИПЛІНА(Код_дисципліни, Назва).

Тоді відношення

ЕКЗАМЕН(Код студента, Код дисципліни, Оцінка)

буде зв'язковим між об'єктними відношеннями «СТУДЕНТ» і «ДИСЦИПЛІНА».

У зв'язковому відношенні допускається дублювання значень ключових атрибутів. Крім ключових атрибутів, у зв'язковому відношенні можуть бути й інші атрибути, наприклад, атрибут «Оцінка».

Тип даних, домен і кортеж

Значення даних, що зберігаються в реляційній базі даних, мають певний тип, наприклад, текстовий, числовий, грошовий, логічний, дата/час та ін. Наприклад, у таблиці 1.1 наведені типи даних відношення «ВИКЛАДАЧ».

Таблиця 1.1 – Типи атрибутів відношення «ВИКЛАДАЧ»

Ім'я атрибута	Тип
Таб_номер	Цілий
Прізвище	Текстовий (20)
Зарплата	Грошовий
№_кафедри	Цілий

Тип даних визначає: множину значень, набір операцій, що можуть бути застосовані до значень заданого типу, та спосіб зовнішнього подання значень. Наприклад, числовий тип визначає множину чисел, над якими допустимі арифметичні операції, а зовнішнє подання – числа.

У реляційній моделі використовуються прості (скалярні) дані, значення яких є атомарними (неподільними). Це означає, що в реляційних операціях не повинна враховуватися внутрішня структура даних. Наприклад, простими даними є дані числового типу та рядки символів, оскільки в операціях вони використовуються як єдине ціле.

Якщо розглядати масив як єдине ціле та не використовувати операції над його елементами, то масив також можна вважати простим типом

даних. Більш того, допустимо створити свій, скільки завгодно складний тип даних, описати можливі дії з ним, і якщо в операціях не потрібне знання внутрішньої структури даних, то такий тип даних також буде простим із погляду реляційної теорії. Наприклад, атрибут «Прізвище» (див. рис. 1.6), який має текстовий тип і рядки символів, що виражають прізвища викладачів, є атомарними. Якщо рядок розкласти в масив символів, то при переході на такий низький рівень буде втрачено сенс цього рядка як єдиного цілого, тобто це буде вже не прізвище, наприклад, Іванов.

Доменом є підмножина значень простого типу даних, які мають певний сенс і є допустимими для атрибута, що визначений на домені. Домен має унікальне ім'я в межах бази даних. Зазвичай ім'ям домену є ім'я атрибута, але можливе й інше ім'я (рис. 1.6). Домен може бути визначений як на простому типі даних, так і на іншому домені. Для опису підмножини допустимих даних домену використовуються логічні умови. Наприклад, домен *BB*, що має сенс «вік викладача», можна описати як підмножину *n* множини *N* натуральних чисел: ($BC = n \in N: n > 18 \text{ і } n < 60$). А домен «Прізвища», який задано текстовим типом атрибута «Прізвище», задає обмеження на довжину рядків – 20 символів (див. табл. 1.1).

Основне призначення доменів полягає в тому, що вони обмежують порівняння. Некоректно, з логічної точки зору, порівнювати значення з різних доменів, навіть якщо вони мають однаковий тип. Наприклад, значення різних доменів «Номери перепусток» і «Номери відділів» не є порівнянними, хоча домени визначені одним типом цілих чисел. Отже, поняття домену допомагає правильно моделювати предметну область.

Кортеж становить набір іменованих значень заданого типу. Іменами значень кортежу є імена атрибутів, а значеннями – припустимі значення домену цього атрибута. Відношення складається з множини кортежів.

Таблиці. Первинні ключі таблиць

Таблиці є основною формою подання відношень і основним засобом зберігання даних у реляційній базі даних. Кожному відношенню відповідає таблиця з таким самим ім'ям, але, на відміну від відношення, ім'я таблиці є іменником у множині, наприклад, «Викладачі». У реляційній базі даних імена таблиць повинні бути унікальними.

Таблиця складається із заголовку, стовпців і рядків.

Стовпці таблиці називають також **полями**. На перетині стовпців і рядків містяться значення даних (див. рис. 1.7). Стовпець таблиці відповідає атрибуту відношення, має ім'я та містить дані одного з припустимих типів, наприклад, текстового, числового або ін. В одній таблиці стовпці повинні мати унікальні імена, а в різних таблицях імена стовпців можуть бути однакові, наприклад, кожна з таблиць «Викладачі» і «Студенти» може мати поле з ім'ям «Прізвище». Кількість стовпців у таблиці фіксована, список їхніх імен складає заголовок таблиці. При створенні таблиці її стовпці упорядковуються зліва направо у послідовності введення імен стовпців. Максимально допустима кількість стовпців у таблиці, зазвичай, не вказується. Однак майже у всіх комерційних СКБД ця межа існує та становить приблизно 255 стовпців. У будь-якій таблиці завжди є принаймні, один стовпець.

Рядки таблиці відповідають кортежам відношення. На відміну від стовпців, рядки не мають імен, кількість їх не обмежена, а порядок розміщення довільний. Припустиме існування таблиці з нульовою кількістю рядків, яка називається порожньою. Порожня таблиця зберігає структуру, визначену її стовпцями, але в ній не містяться дані. Стандарти реляційних баз даних не накладають обмежень на кількість рядків у таблиці, і в багатьох СКБД розмір таблиць обмежений лише вільним дисковим простором комп'ютера.

Таблиці розміщуються у файлі бази даних. Кожен рядок таблиці відповідає запису файлу, тому рядки таблиці називають також записами.

Первинні ключі ідентифікують рядки, оскільки рядки таблиці не мають імен. Первинний ключ становить комбінацію атрибутів таблиці, дані яких однозначно визначають кожен рядок таблиці. Якщо ключ складається з одного атрибута, він називається простим ключем, а якщо з декількох – складеним ключем. Атрибути, з яких складається ключ, називають ключовими. Первинний ключ таблиці будемо визначати жирним шрифтом, як у СКБД Access, наприклад, **Таб_номер**.

Логічні зв'язки між таблицями. Вторинні ключі

Реляційна база даних складається з набору взаємопов'язаних таблиць. Зв'язки об'єднують інформацію окремих таблиць в єдину базу даних.

Зв'язком будемо називати спосіб, за допомогою якого дані однієї

таблиці об'єднується з даними другої таблиці. Умовою об'єднання даних є збігання значень певних полів таблиць, що пов'язані між собою. Зв'язок між двома таблицями визначає їхню підпорядкованість – одна з таблиць є головною, а друга – підлеглою. У головній таблиці атрибутом зв'язку є первинний ключ, а в підлеглий таблиці – вторинний, або зовнішній, ключ.

Вторинний ключ – це додаткове поле, що включається в підлеглу таблицю для встановлення зв'язку з головною таблицею. Первинний та вторинний ключі, що пов'язують таблиці, повинні мати однаковий тип. Імена цих ключів можуть бути різними.

Існують такі види зв'язків: «один-до-одного» (1:1), «один-до-багатьох» (1:M), «багато-до-одного» (M:1) та «багато-до-багатьох» (M:M).

Зв'язок «один-до-одного» означає, що одному запису в головній таблиці відповідає один запис у підлеглий таблиці, а кожному запису підлеглої таблиці – рівно один запис у головній. Наприклад, маємо дві таблиці «Викладачі» та «Фотографії». Якщо для кожного викладача таблиці «Викладачі» зберігається одна фотографія в таблиці «Фотографії», то між таблицями встановлюється зв'язок «один-до-одного» (рис. 1.8).

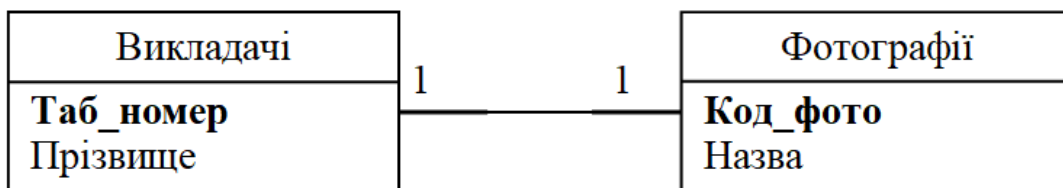


Рисунок 1.8 – Організація зв'язку «один-до-одного»

Зв'язок «один-до-багатьох» означає, що одному рядку головної таблиці може відповідати будь-яка кількість рядків у підлеглий таблиці, зокрема один рядок або жодний із рядків, а кожному рядку підлеглої таблиці – тільки один рядок у головній таблиці. Наприклад, маємо дві таблиці «Викладачі» та «Дисципліни» (рис. 1.9).

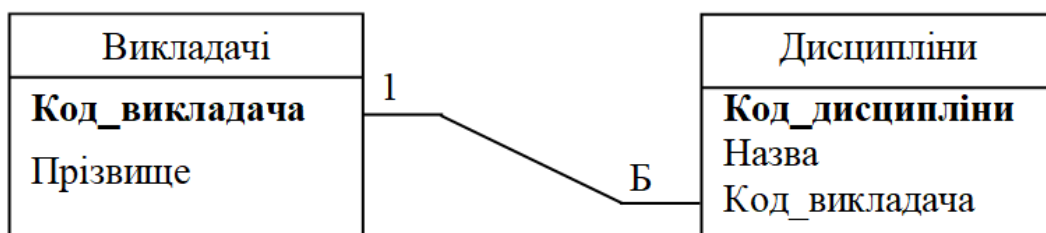


Рисунок 1.9 – Організація зв'язку «один-до-багатьох»

Нехай кожний викладач може читати декілька навчальних дисциплін. Тоді у підлеглу таблицю «Дисципліни» включимо додаткове поле «Код_викладача», яке є вторинним ключем. Зв'язок між таблицями «Викладачі» та «Дисципліни» буде «один-до-багатьох». Цей вид зв'язку зустрічається найчастіше. Зв'язок «багато-до-одного» відрізняється від зв'язку «один-до-багатьох» тільки напрямком. Якщо на зв'язок «багато-до-одного» подивитися з боку підлеглої таблиці, то вона перетворюється в зв'язок «багато-до-одного».

Зв'язок «багато-до-багатьох» спостерігається, коли одному рядку головної таблиці може відповідати кілька рядків підлеглої таблиці, й одночасно одному рядку підлеглої таблиці – кілька рядків головної таблиці. Наприклад, викладач ВНЗ може читати декілька навчальних дисциплін, а одну дисципліну можуть читати декілька викладачів. Тоді між таблицями «Викладачі» та «Дисципліни» потрібно встановити зв'язок «багато-до-багатьох» (рис. 1.10).

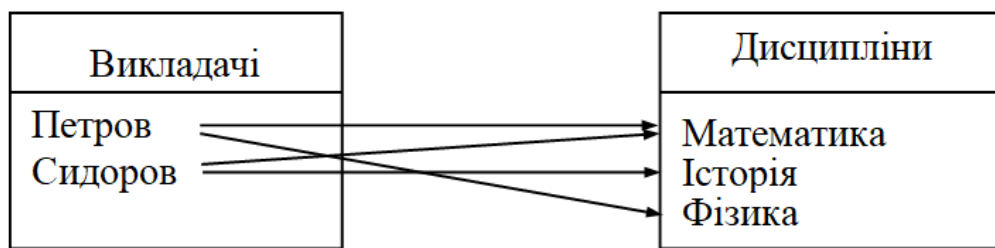


Рисунок 1.10 – Організація зв'язку «багато-до-багатьох»

Однак зв'язок «багато-до-багатьох» безпосередньо не підтримується в реляційних СКБД, наприклад, у СКБД Access. Такий зв'язок потрібно замінити на зв'язки «один-до-багатьох» шляхом використання зв'язкової таблиці (рис. 1.11).

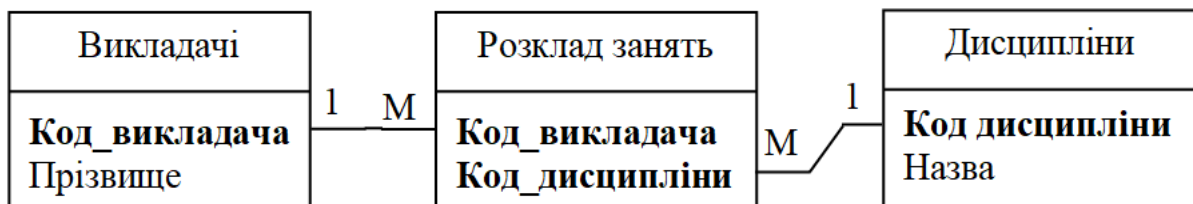


Рисунок 1.11 – Спосіб переходу від зв'язку «багато-до-багатьох» до зв'язків «один-до-багатьох»

Індекси та їхнє призначення

Припустимо, що ми працюємо з кадровою системою. Як знайти інформацію про певного співробітника? У таблиці «Співробітники» записи зазвичай йдуть у довільному порядку. Час пошуку потрібних записів простим перебором збільшується. Щоб скоротити час пошуку, необхідно перед початком пошуку впорядкувати записи таблиці за тими полями, які використовуються для пошуку (наприклад, за полем «Прізвище»), але при цьому всі записи таблиці потрібно буде багаторазово переміщати в пам'яті, що потребує додаткових витрат часу.

Для скорочення часу на впорядкування записів доцільно індексувати поля таблиці, які використовуються для пошуку (наприклад, поле «Прізвище»). Під час сортування змінюється лише порядок розміщення індексів, а записи таблиці не переміщуються в пам'яті. Час на пошук даних значно скорочується.

Індекси створюються автоматично, якщо при формуванні таблиці задати ознаку «Індексоване поле» для тих полів, за якими передбачається пошук. За наявності цієї ознаки СКБД створює внутрішню таблицю для кожного індексу. Таблиця індексу має два стовпці:

значення виразу, що об'єднує всі поля, за якими виконується пошук; місцеположення (адреса) кожного запису в початковій таблиці.

На рисунку 1.12 показано принцип сортування таблиці «Викладачі» за індексованим полем «Прізвище» (другий стовпець таблиці, індекс = 2).

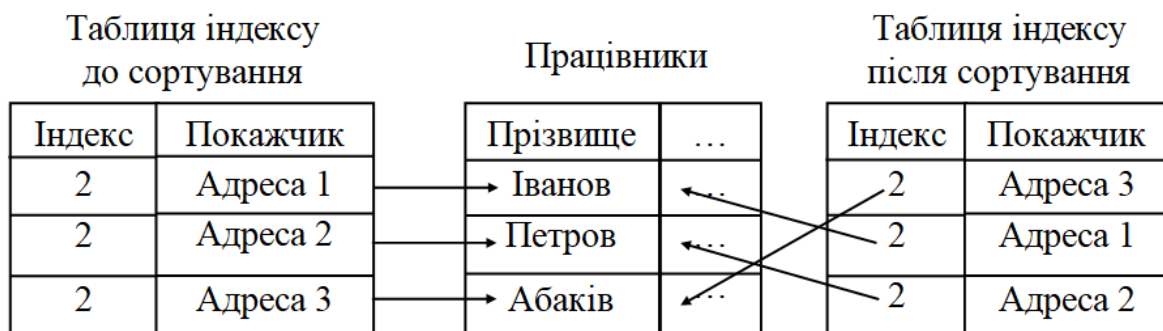


Рисунок 1.12 – Принцип сортування за індексованим полем «Прізвище»

При створенні таблиці індексу в стовпці «Показчик» формуються адреси записів таблиці «Працівники» (див. таблицю індексу до сортування). Сортування таблиці «Викладачі» за алфавітом поля «Прізвище» виконується автоматично. Адреси записів у полі «Показчик» розміщуються відповідно до алфавіту поля «Прізвище» (див. таблицю

індексу після сортування). Наприклад, «Адреса 3» запису «Абаків» розміщується в першому рядку таблиці індексу, «Адреса 1» запису «Іванов» – у другому рядку, а «Адреса 2» запису «Петров» – у третьому рядку. За цими адресами СКБД здійснює швидкий пошук записів у таблиці «Викладачі».

Отже, завжди доцільно індексувати поля таблиць, за якими передбачається пошук інформації у базі даних. Операція створення індексу називається індексуванням. Ключові поля завжди потрібно індексувати.

Цілісність даних у базі даних

Ядром будь-якої бази даних є модель даних. Вона становить множину структур даних, обмежень цілісності й операцій маніпулювання даними. За допомогою моделі даних можуть бути зображені об'єкти предметної області та взаємозв'язки між ними. Найбільш поширене трактування реляційної моделі даних належить К. Дейту. Згідно з трактуванням Дейта у склад реляційної моделі входять три частини: структурна, цілісна та маніпуляційна.

Структурна частина реляційної моделі

Структурна частина описує об'єкти, які розглядаються реляційною моделлю, а єдиною структурою даних, яка використовується в реляційних базах даних, є нормалізоване n -арне відношення.

Нехай задані n множин даних D_1, D_2, \dots, D_n , які є доменами. Тоді відношенням R називається множина n упорядкованих кортежів $\langle d_1, d_2, \dots, d_n \rangle$, де $d_1 \in D_1$ (символ \in означає «елемент з»), $d_2 \in D_2, \dots, d_n \in D_n$. Кортеж становить підмножину пар $\{\langle \text{ім'я атрибута} \rangle, \langle \text{значення} \rangle\}$ і є унікальним у відношенні. У математичному сенсі відношення R є підмножиною декартового добутку доменів $D_1 \times D_2 \times \dots \times D_n$. Відношення R включає заголовок і тіло.

Заголовок (схема) відношення містить фіксовану кількість імен атрибутів відношення. Імена атрибутів мають бути унікальними в межах відношення та можуть збігатися з іменами відповідних доменів. Заголовок не змінюється під час роботи з базою даних. Якщо у відношенні змінені, додані або видалені атрибути, то маємо інше відношення (навіть з колишнім ім'ям).

Тілом відношення є набір кортежів. Воно може бути модифікованим під час роботи з базою даних, тобто кортежі можуть змінюватися,

додаватися та видалятися.

Кількісні характеристики відношення

Ступінь (арність) відношення – це число атрибутів відношення. Відношення ступеня один називають унітарним, ступеня два – бінарним, ступеня три – тернарним, а ступеня n – n -арним (відношення має n атрибутів). Ступінь відношення не змінюється в часі.

Кардинальність (потужність) відношення – це кількість кортежів відношення. Кардинальне число відношення змінюється в часі. Наприклад, відношення «Викладачі» (див. рис. 1.7) є 4-арним, а його кардинальність – 4.

Фундаментальні властивості відношень

Властивості відношень безпосередньо виходять з наведеного вище теоретико-множинного визначення поняття відношення, яке є множиною кортежів. Фундаментальні властивості відношень:

Однотипність кортежів. Усі елементи відношення є однотипними кортежами, що дозволяє вважати кортежі аналогом рядків у таблиці.

Обмеженість відношень. Відношення включає не всі можливі кортежі з декартового добутку доменів, на яких воно визначене. Для кожного відношення є критерій, який дозволяє визначити ті кортежі, що входять у відношення. Цей критерій визначає сенс або семантику відношення, є логічним виразом і називається предикатом відношення R .

Відсутність кортежів-дублікатів і впорядкованості кортежів. У відношенні відсутні однакові кортежі та кортежі не впорядковані зверху вниз, оскільки відношенням є множина кортежів, а згідно з теорією множин кожна множина складається з різних елементів, які не впорядковані. Звідси випливає, що посилання на кортежі у відношенні повинно мати первинний ключ.

Відсутність впорядкованості атрибутів. Атрибути відношень не впорядковані, оскільки за визначенням кортеж становить підмножину пар $\{<ім'я атрибута>, <значення>\}$ і для посилання на значення використовується ім'я атрибута. Ця властивість дозволяє модифікувати схеми існуючих відношень шляхом додавання або видалення атрибутів.

Атомарність значень атрибутів. Значення всіх атрибутів повинні бути атомарними. Це є наслідком визначення домену як потенційної множини значень простого типу, тобто серед значень домену не може бути відношення.

Потенційний, альтернативний і первинний ключі відношення

Потенційний ключ відношення – це мінімальний набір атрибутів, який може бути використаний для однозначної ідентифікації будь-якого кортежу відношення. Ключі відношень бувають атомарними і складеними. Атомарний ключ складається з єдиного атрибута, а **складений ключ** – із набору атрибутів. Складений ключ не повинен мати зайвих атрибутів. У його склад включаються тільки ті атрибути, без яких не можлива ідентифікація кортежів.

Відношення має принаймні один потенційний ключ. Дійсно, якщо ніякий атрибут або група атрибутів не є потенційним ключем, то через унікальність кортежів усі атрибути разом утворюють потенційний ключ.

Відношення може мати декілька потенційних ключів. Один із них оголошується первинним ключем (ПК), а інші – альтернативними ключами.

Поняття потенційного ключа є семантичним і відображає трактування понять предметної області. Наприклад, розглянемо відношення «ВИКЛАДАЧ», яке подане у вигляді таблиці на рисунку 1.7.

На перший погляд може здатися, що в цьому відношенні є три потенційних ключі: «Таб_номер», «Прізвище» і «Зарплата». У кожній колонці таблиці містяться унікальні дані. Проте серед викладачів можуть бути викладачі з однаковою зарплатою та однаковим прізвищем. Табельний номер, однак, є унікальним для кожного викладача, і саме розуміння семантики даних привело до твердження, що цьому відношенні є тільки один потенційний ключ – «Таб_номер».

Отже, потенційні ключі є єдиним засобом адресації кортежів у відношенні. Тільки знання значень потенційного ключа кортежу дозволяє точно визначити первинний ключ відношення.

У практичній реалізації реляційної СКБД відношення є математичним аналогом таблиць. Таблиця є найбільш звичним і зручним вмістищем для зберігання інформації, має фіксовану кількість поіменованих і впорядкованих стовпців та необмежену кількість рядків. У таблиці 1.2 наведені терміни, які вживаються як синоніми.

Таблиця 1.2 – Порівняння термінології

Реляційний термін	Табличний термін у СКБД
Відношення	Таблиця
Заголовок відношення	Заголовок таблиці
Тіло відношення	Тіло таблиці
Атрибут відношення	Найменування стовпця (поля) таблиці
Кортеж відношення	Рядок (запис) таблиці
Ступінь відношення	Кількість стовпців таблиці
Кардинальність відношення	Кількість рядків таблиці
Домен	Тип даних (базовий або користувача)

Цілісна частина реляційної моделі

Під цілісністю даних розуміється забезпечення достовірності й узгодження даних, а також виконання логічних обмежень на дані у базі даних у будь-який момент часу її функціонування.

У цілісній частині моделі даних фіксуються дві базові вимоги цілісності, що повинні підтримуватися в будь-якій реляційній СКБД. Перша вимога називається **вимогою цілісності сутностей**, а друга – **вимогою цілісності за посиланнями, або вимогою цілісності зовнішніх ключів**.

Цілісність сутностей. Під сутністю розуміється деякий об'єкт, що становить інтерес для опису предметної сфери при створенні моделі бази даних. Формою подання сутності є відношення або таблиця.

Правило цілісності сутностей: кожна сутність, що реалізована відношенням, повинна мати первинний ключ, а вхідні до його складу атрибути не можуть приймати NULL-значень. Цей потенційний ключ найкраще оголошувати первинним ключем таблиці даної суті.

Примітка. NULL-значення – це, власне, не значення, а маркер, який показує, що значення невідоме. Проблема використання NULL-значення в теорії реляційних баз даних остаточно не вирішена. Практично всі реалізації сучасних реляційних СКБД дозволяють використовувати NULL-значення, незважаючи на їх недостатню теоретичну обґрунтованість. Безумовно, варто уникати появи NULL-значень, обумовлених некоректним проектуванням бази даних, але в реальній ситуації, коли якісь дані про екземпляр сутності невідомі, без NULL-значень не обійтися.

Цілісність зовнішніх ключів. Цілісність зовнішніх ключів означає підтримку зв'язків між записами пов'язаних таблиць у разі внесення змін у

базу даних. Цілісність даних забезпечується в разі виконання таких умов:

- Пов'язане поле головної таблиці повинно бути ключовим полем.
- Пов'язані ключові поля головної та підлеглої таблиць повинні мати однаковий тип даних, а імена цих полів можуть бути різними.
- Пов'язані таблиці повинні бути таблицями однієї бази даних.

Для забезпечення цілісності зовнішнього ключа необхідно в процесі створення та експлуатації бази даних враховувати такі правила:

- не можна вводити в поле зовнішнього ключа підлеглої таблиці значення, яке відсутнє в ключовому полі головної таблиці. Наприклад, не припустимо вводити в ключове поле «Код_викладача» таблиці «Дисципліни» значення, якого немає в таблиці «Викладачі» (див. рис. 1.8);
- не допускається видалення запису з головної таблиці, якщо існують пов'язані з нею записи в підлеглій таблиці. Наприклад, неможливо видалити запис із таблиці «Викладачі», якщо в таблиці «Фотографії» існують записи, які пов'язані з даним співробітником;
- не можна змінювати значення ключового поля в головній таблиці, якщо існують записи, пов'язані з даним значенням. Наприклад, не можна змінювати код співробітника в таблиці «Викладачі», якщо в таблиці «Фотографії» є записи, що відносяться до цього співробітника.

У СКБД для реалізації цих правил передбачений механізм забезпечення цілісності даних. За умови порушення умов цілісності дія не виконується й на екран виводиться попередження. Крім того, для реалізації обмежень на зміну або видалення пов'язаних записів у СКБД передбачено механізм каскадного оновлення або каскадного видалення. Цілісність даних зберігається завдяки тому, що СКБД при зміні ключового поля головної таблиці автоматично змінює й відповідні значення пов'язаних записів, а при видаленні запису в головній таблиці – видаляє й усі пов'язані записи в підлеглій таблиці.

Маніпуляційна частина реляційної моделі

Важливою частиною реляційної моделі даних є механізм маніпулювання даними, який забезпечує вибірку й оновлення даних. У маніпуляційній частині моделі стверджується два фундаментальних механізми маніпулювання реляційними базами даних: реляційна алгебра та реляційне числення. Перший механізм базується на класичній теорії множин, а другий – на класичному логічному апараті числення предикатів

першого порядку.

Неформально реляційну алгебру можна визначити як високорівневу процедурну мову – мову запитів. За допомогою запитів користувач повідомляє СКБД правила побудови необхідного йому відношення з одного або декількох відношень, що існують у базі. Реляційна алгебра має велику виразну потужність: дуже складні запити до бази даних можуть бути виражені за допомогою одного виразу реляційної алгебри.

Реляційне числення не є процедурною мовою, яку можна використовувати для визначення відношення, що може бути створене на основі одного або декількох інших відношень бази даних. І хоча в реляційній моделі даних присутні обидва механізми, далі розглянемо тільки операції реляційної алгебри.

Склад операцій і замкнутість реляційної алгебри

Реляційна алгебра – це мова операцій, що дозволяють створювати на основі одного або декількох відношень інше відношення без зміни початкових відношень.

Замкнутість реляційних операцій

Реляційні операції є *замкнутими* щодо відношення, оскільки операндами та результатами операцій реляційної алгебри є відношення. Ця властивість реляційних операцій дозволяє створювати вкладені вирази реляційної алгебри (за аналогією з тим, як створюються вкладені арифметичні вирази). При будь-якій глибині вкладеності виразів результатом обчислень завжди є відношення.

Склад операцій реляційної алгебри

Теоретичну основу реляційної алгебри становить теорія множин, оскільки відношення також є множинами. У склад реляційної алгебри, яку запропонував Кодд, включено вісім основних операцій. Вони поділяються на два класи – теоретико-множинні та спеціальні реляційні операції.

Теоретико-множинними є такі операції: об'єднання відношень, пересічення відношень, різниця відношень і декартовий добуток відношень. Спеціальними реляційними операціями є такі операції: вибір (селекція) відношень, проєкція відношень, з'єднання відношень, ділення відношень. Операції вибору та проєкції є **унарними**, а інші – **бінарними**. Унарна операція здійснюється над одним відношенням, а бінарна – над парою відношень.

Крім основних операцій, у склад реляційної алгебри включено операції присвоювання та перейменування.

Операція присвоювання (=) призначена для зберігання результату обчислення реляційного виразу в існуючому відношенні бази даних.

Операція перейменування (RENAME) призначена для зміни імен атрибутів. Результатом цієї операції є нове відношення з новими іменами атрибутів. Синтаксис операції перейменування:

$$R \text{ RENAME } A_1, A_2, \dots, A_k, \text{New}A_1, \text{New}A_2, \dots, \text{New}A_k,$$

де R – відношення;

A_1, A_2, \dots, A_k – початкові імена атрибутів;

$\text{New}A_1, \text{New}A_2, \dots, \text{New}A_k$ – нові імена атрибутів.

Теоретико-множинні операції реляційної алгебри

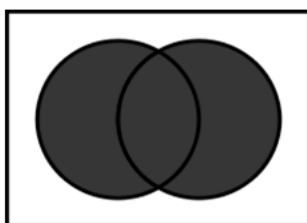
Сенс операцій об'єднання, пересічення та різниці в реляційній алгебрі загалом залишається теоретико-множинним (рис. 1.13):

– об'єднанням двох множин $A\{a\}$ і $B\{b\}$ є така множина $C\{c\}$, що для кожного c або існує такий елемент a множини A , що $c = a$, або існує такий елемент b множини B , що $c = b$ (рис. 1.13, а);

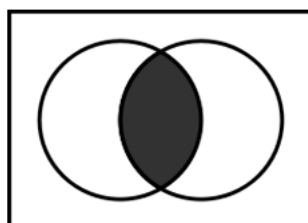
– пересіченням множин $A\{a\}$ і $B\{b\}$ є така множина $C\{c\}$, що для будь-якого c існують такі елементи a і b , що $c = a = b$ (рис. 1.13, б);

– різницею множин $A\{a\}$ і $B\{b\}$ є така множина $C\{c\}$, що для будь-якого c існує такий елемент a , що $c = a$, і не існує такий елемент b , що $c = b$ (рис. 1.13, в).

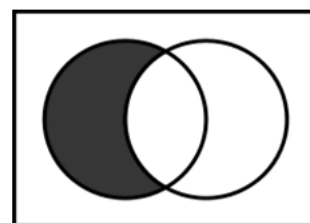
Для операцій об'єднання, пересічення та різниці потрібна сумісність відношень за об'єднанням. Два відношення сумісні за об'єднанням лише в тому випадку, коли мають однакові заголовки й однойменні атрибути визначені на одному й тому самому домені. Інакше кажучи, відношення є сумісними за об'єднанням, якщо мають однакову арність, однакові імена атрибутів і однойменні атрибути мають однаковий тип.



а



б



в

Рисунок 1.13 – Ілюстрація результатів теоретико-множинних операцій:

а – об'єднання; б – пересічення; в – різниця

Друга особливість реляційних операцій полягає в можливості перейменування атрибутів. Якщо два відношення сумісні у всьому, окрім імен атрибутів, то перед виконанням цих операцій необхідно зробити перейменування атрибутів із використанням операції перейменування.

Операції над відношеннями реляційної алгебри

Теоретико-множинні операції реляційної алгебри

Розглянемо правила виконання теоретико-множинних операцій.

Операція об'єднання (UNION). Позначення – $R \cup S$

Об'єднанням відношень R і S називається множина кортежів, які належать або R , або S , або їм обом (рис. 1.14).

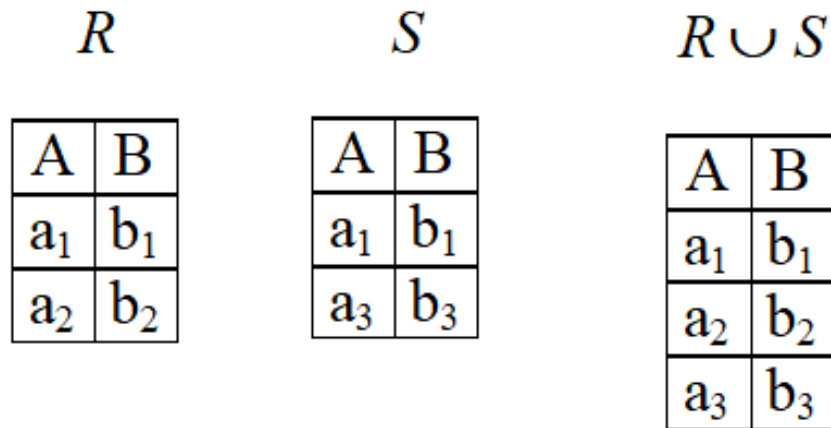
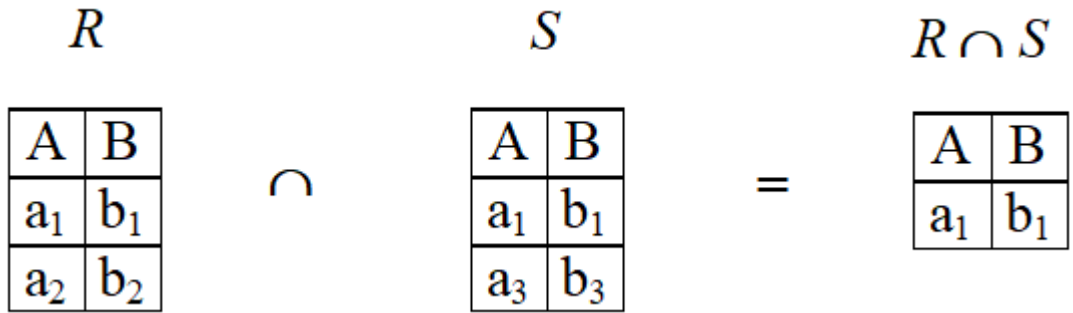


Рисунок 1.14 – Об'єднання відношень $R \cup S$

За допомогою операції об'єднання може бути реалізовано додавання кортежу з наявного відношення. У цьому випадку R – початкове відношення, а S – відношення, що містить один кортеж, який додається.

Операція пересічення (INTERSECTION). Позначення – $R \cap S$

Пересіченням відношень R і S називається множина кортежів, що належать як R , так і S (рис. 1.15).



P

Рисунок 1.15 – Пересічення відношень R і S

Операція різниці (DIFFERENCE). Позначення – $R - S$

Різницею відношень R і S називається множина кортежів, які належать R і не належать S (рис. 1.16). Відношення повинні мати однакову арність.

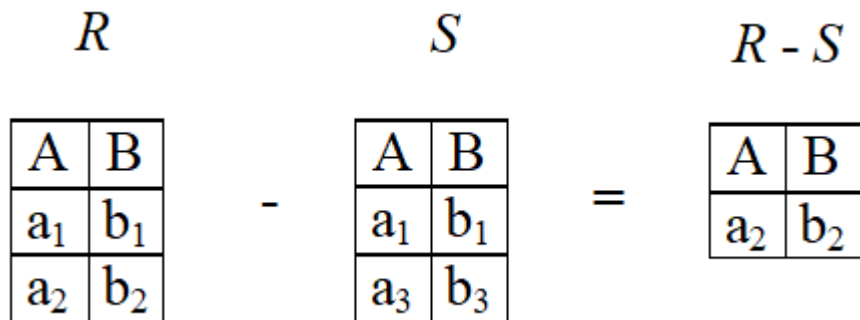


Рисунок 1.16 – Різниця відношень $R - S$

Операція декартового добутку (CROSS-PRODUCT) – $R \times S$

Операція декартового добутку застосовується для множення двох відношень. Множенням двох відношень називається створення іншого відношення, що складається зі всіх можливих пар кортежів обох відношень.

Формально операція розширеного декартового добутку визначається так. Нехай є два відношення $R_1(a_1, a_2, \dots, a_m)$ і $R_2(b_1, b_2, \dots, b_n)$. Результатом цієї операції є відношення $R(a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n)$, у якому кожний кортеж є об'єднанням одного кортежу першого відношення й одного кортежу другого відношення: $(ra_1, ra_2, \dots, ra_m, rb_1, rb_2, \dots, rb_n)$, де $(ra_1, ra_2, \dots, ra_m)$ – кортеж відношення R_1 , а $(rb_1, rb_2, \dots, rb_n)$ – кортеж відношення R_2 . Якщо одне відношення (R_1) має m атрибутів і k кортежів, а інше (R_2) – n атрибутів і p кортежів, то їхній декартовий добуток міститиме $(m + n)$

атрибутів і $(k \times p)$ кортежів.

У цій операції відношення повинні бути сумісними. Два відношення **сумісні за взяттям розширеного декартового добутку**, якщо пересічення множин імен атрибутів, узятих із їхніх схем, буде пустим. Це означає, що імена атрибутів початкових відношень повинні бути унікальними (рис. 1.17).

Для забезпечення цієї вимоги імена атрибутів у відношенні-результаті R доповнюються назвами початкових відношень у вигляді префіксів.

Варто зауважити, що операція взяття декартового добутку не є дуже осмисленою на практиці, оскільки потужність тіла її результату надто велика.

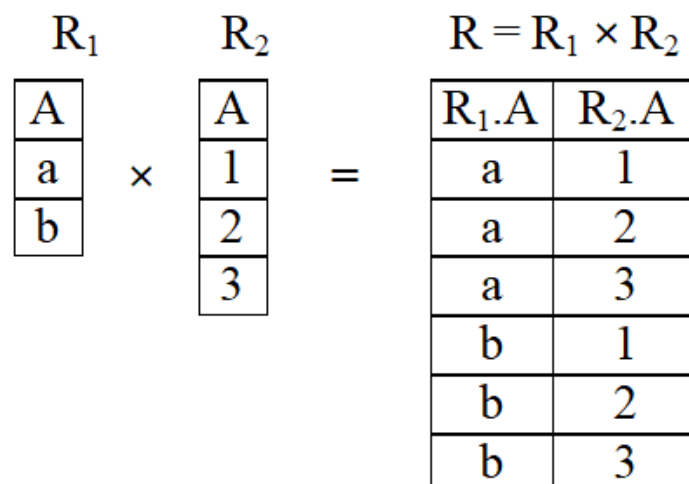


Рисунок 1.17 – Декартовий добуток: $R \times S$

Основний сенс включення операції декартового добутку до складу реляційної алгебри Кодда полягає в тому, що на її основі визначається дійсно корисна операція з'єднання.

Спеціальні операції реляційної алгебри

До складу спеціальних операцій входять: вибір відношень, проєкція відношень, з'єднання відношень та ділення відношень.

Операція вибору (SELECTION). Позначення – $\sigma F(R)$

Нехай R – відношення, а F – предикат (формула), що утворений операндами, які є константами, іменами атрибутів, арифметичними операторами порівняння або логічними операторами (AND – І, OR – АБО,

NOT – III). Тоді вибором (селекцією) σF називається множина кортежів, компоненти якої задовольняють умову, що задана предикатом F . Ця операція будує нове відношення, яке є «горизонтальною» підмножиною початкового відношення. На рисунку 1.18 наведено приклад виконання операції $\sigma_{A=a_3}(R)$ вибору кортежів відношення R . Тут $F: A = a_3$ – вміст стовпця A дорівнює значенню a_3 . Згідно з умовою вибирається третій кортеж. Отримано нове відношення S , у склад якого входить лише один кортеж.

R			
A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2
a_3	b_2	c_3	d_2
a_4	b_4	c_4	d_4

 $\sigma_{A=a_3}(R) =$

S			
A	B	C	D
a_3	b_2	c_3	d_2

Рисунок 1.18 – Виконання операції вибору

Операція проєкції (PROJECT). Позначення – $\pi_{A_1, A_2, \dots, A_m}(R)$

Проекція $\pi_{A_1, A_2, \dots, A_m}(R)$ – це множина кортежів, що складається з кортежів відношення R шляхом вибору стовпців з іменами A_1, A_2, \dots, A_m . Ця операція будує «вертикальну» підмножину (S) шляхом вибору тільки заданих атрибутів і виключення останніх. Кортежі, що повторюються, також видаляються. Результат операції проєкції $\pi_{B,D}(R)$ показано на рисунку 1.19.

R			
A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2
a_3	b_2	c_3	d_2
a_4	b_4	c_4	d_4

 $\pi_{B,D}(R) =$

S	
B	D
b_1	d_1
b_2	d_2
b_4	d_4

Рисунок 1.19 – Проекція: $S = \pi_{B,D}(R)$

Операції з'єднання (JOIN). Позначення – $R \triangleright \triangleleft S$

Як правило, користувачів цікавить лише деяка частина всіх комбінацій кортежів декартового добутку, яка задовольняє задану умову. Тому замість декартового добутку зазвичай використовується одна з найважливіших операцій реляційної алгебри – операція з'єднання. У результаті її виконання на базі двох початкових відношень створюється нове відношення.

На практиці застосовуються три різновиди операції з'єднання:

- умовне з'єднання відношень (тета-з'єднання);
- еквіз'єднання відношень (з'єднання за еквівалентністю);
- природне з'єднання відношень.

Умове (тета-) з'єднання (CONDITIONAL). Позначення – $R \triangleright \triangleleft_a \theta b S$

Операція умовного з'єднання здійснюється над двома відношеннями R і S . Результатом операції є відношення, яке містить кортежі з декартового добутку відношень $R \times S$, що задовольняють предикат F . Предикат F має вигляд $a \theta b$, де a і b – імена атрибутів різних відношень-операндів, θ – арифметичний оператор порівняння ($=, \neq, >, \geq, <, \leq$).

Формальне визначення операції. З'єднання відношень R і S за стовпцями A_i та A_j визначає в декартовому добутку $R \times S$ множину кортежів, у якій i -й компонент R знаходиться у відношенні θ з j -м компонентом S . Операція умовного з'єднання здійснюється за формулою:

$$R \triangleright \triangleleft_{A \theta B} S = \sigma_{i \theta (l+j)}(R \times S),$$

де l – арність відношення R .

Під час реалізації операції з'єднання послідовно виконуються дві операції:

1. операція декартового добутку $R \times S$;
2. операція вибір $\sigma_{i \theta (l+j)}(R \times S)$.

Нехай необхідно виконати з'єднання двох відношень R і S за умови, що значення першого стовпця ($i = 1$) відношення R повинні бути менше значень другого стовпця ($j = 2$) відношення S (рис. 1.20, а). Для наочності стовпці, значення яких порівнюються, виділені напівжирним написанням.

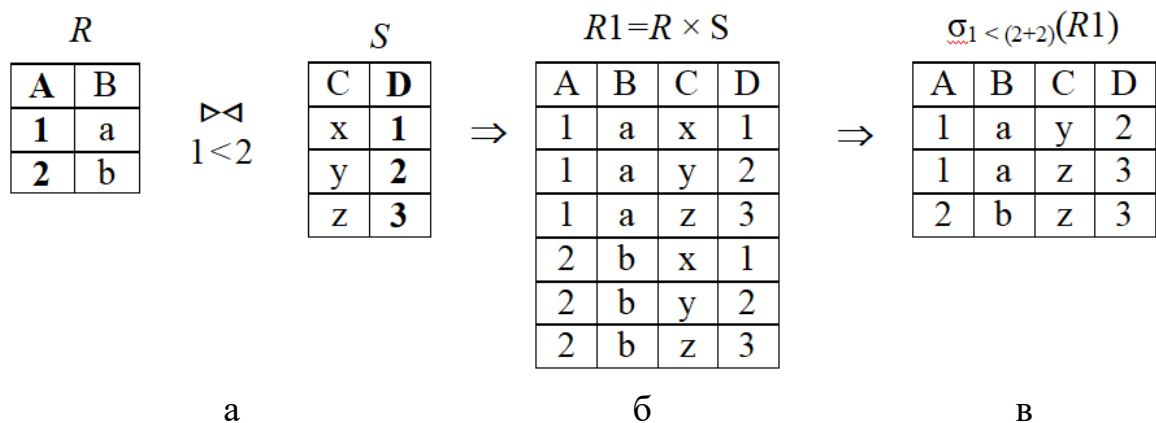


Рисунок 1.20 – Приклад виконання операції з’єднання

Першою виконується операція декартового добутку (рис. 1.20, б), а другою – операція вибору за умовою $1 < (2 + 2)$, яка отримана шляхом підстановки в загальну умову $i \theta (l + j)$ таких значень: $i = 1, l = 2, j = 2, \theta = <$.

Еквіз’єднання (з’єднання за еквівалентністю). Позначення – $R \bowtie_{a=b} S$

Операція називається еквіз’єднанням, якщо предикат F містить тільки операцію порівняння за рівністю ($=$). Приклад виконання операції еквіз’єднання $\sigma_{R.A=S.D}(R \times S)$ наведено на рисунку 1.21.

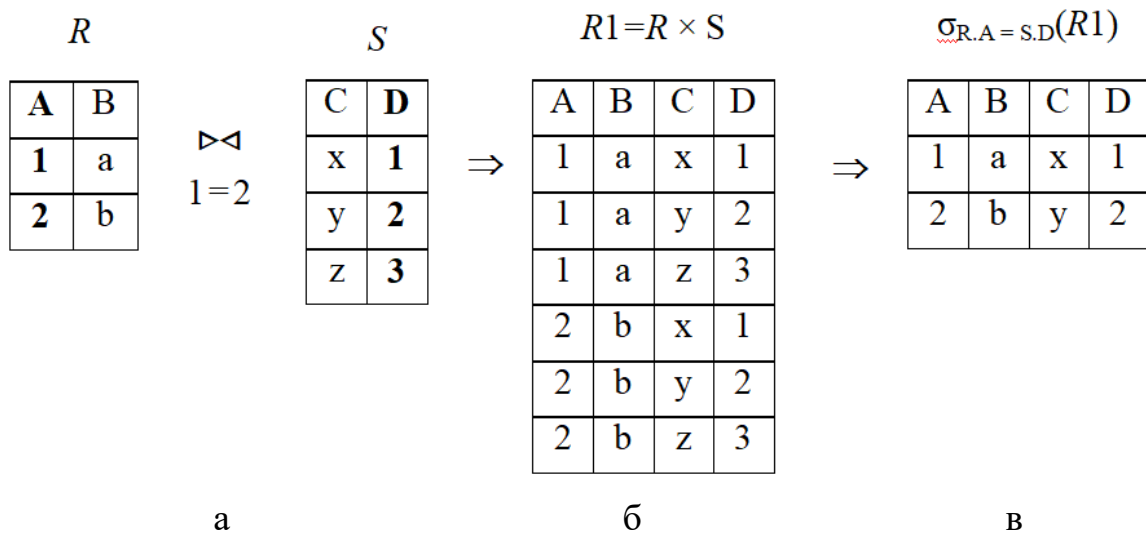


Рисунок 1.21 – Приклад виконання операції еквіз’єднання

Операція еквіз’єднання найчастіше зустрічається на практиці та для неї існують найефективніші алгоритми реалізації.

Природне з’єднання. Позначення – $R \bowtie S$

Ця операція застосовна до пари відношень, у яких збігаються імена деяких атрибутів, наприклад, імена зовнішніх ключів (ключів зв’язку).

Природним з'єднанням називається з'єднання за еквівалентністю двох відношень, що виконане за всіма загальними атрибутами, з результатів якого виключається по одному екземпляру кожного загального атрибута.

Нехай задані два відношення R і S :

$$R = (A_1, \dots, A_k, B_1, \dots, B_n), S = (A_1, \dots, A_k, C_1, \dots, C_m),$$

де імена A_1, \dots, A_k збігаються.

Тоді операція природного з'єднання $R \bowtie S$ визначається так:

$$R \bowtie S = \pi_{A_1, \dots, A_k, B_1, \dots, B_n, C_1, \dots, C_m}(\sigma_{R.A_1 = S.A_1 \& R.A_1 = R.A_2 \& S.A_2, \dots, R.A_k = S.A_k}(R \times S)).$$

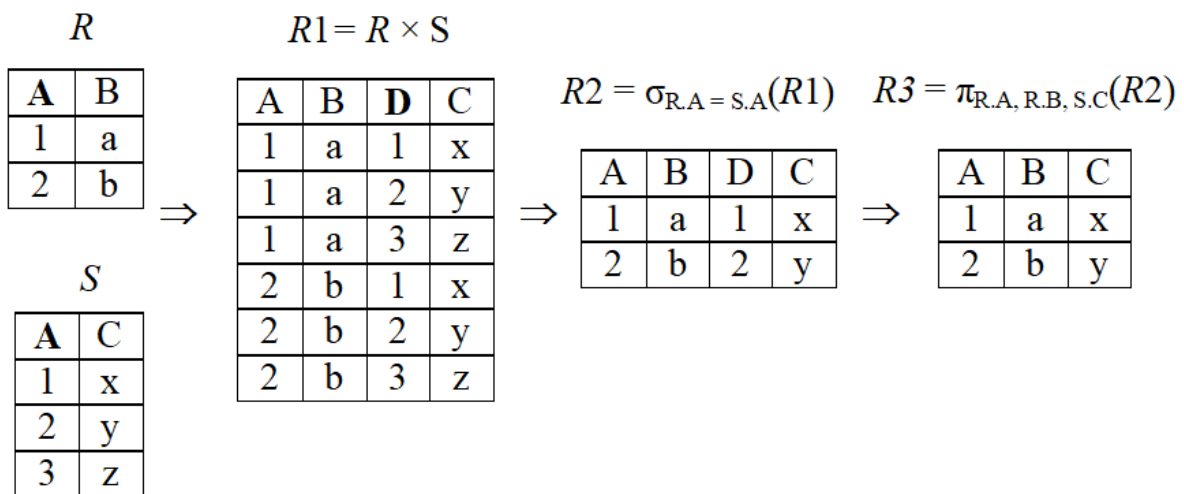
В операції природного з'єднання послідовно виконуються такі операції:

– операція декартового добутку: $R1 = R \times S$ із перейменуванням атрибутів, імена яких збігаються;

операція еквіз'єднання: $R2 = \sigma_{R.A_1 = S.A_1 \& R.A_1 = R.A_2 \& S.A_2, \dots, R.A_k = S.A_k}(R1)$;

операція проєкції $R3 = \pi_{A_1, \dots, A_k, B_1, \dots, B_n, C_1, \dots, C_m}(R2)$.

Приклад виконання операції природного з'єднання $R \bowtie S$ наведено на рисунку 1.22.



$D = \text{RENAME}(S.A, S.D)$ – перейменування атрибута з A на D у відношенні $R1$.

Рисунок 1.22 – Приклад виконання операції природного з'єднання

Приклад операції з'єднання

Нехай у базі даних «Екзамени» зберігаються результати екзаменів студентів групи з математики й інформатики (рис. 1.23).

КодСт	Прізвище
1001	Іванов
1002	Петров

КодДисц	НазваДисц
101	Математика
102	Інформатика

КодСт	КодДисц	Оцінка
1001	101	5
1001	102	5
1002	101	3
1002	102	4

Рисунок 1.23 – Таблиці бази даних «Екзамени»

Необхідно визначити прізвища студентів, які склали всі екзамени на відмінно.

Для розв'язання задачі виконаємо таку послідовність операцій.

1. Операція вибору $R1 = \sigma_{\text{Оцінка} = 5}(\text{Екзамени})$. Отримаємо відношення $R1$, у якому будуть тільки оцінки відмінно (рис. 1.24).

$R1 = \sigma_{\text{Оцінка} = 5}(\text{Екзамени})$

КодСт	КодДисц	Оцінка
1001	101	5
1001	102	5
1003	101	5
1003	102	5

Рисунок 1.24 – Результат операції вибору

2. Операція еквіз'єднання відношень *Студенти* і $R1$ (рис. 1.25)

$R2 = \sigma_{\text{Студенти.КодСт} = \text{Оцінка.КодСт}}(\text{Студенти} \times R1)$.

$R2 = \sigma_{\text{Студенти.КодСт} = \text{Оцінка.КодСт}}(\text{Студенти} \times R1)$

КодСт	Прізвище	КодСт1	КодДисц	Оцінка
1001	Іванов	1001	101	5
1001	Іванов	1001	102	5
1002	Петров	1003	101	5
1002	Петров	1003	102	5

Рисунок 1.25 – Результат операції еквіз’єднання відношень *Студенти* і *R1*

3. Операція проєкції відношення *R2* (рис. 1.26).

$$R3 = \pi_{R2.КодСт, R2.Прізвище, R2.КодДисц, R2.Оцінка}(R2).$$

$$R3 = \pi_{R2.КодСт, R2.Прізвище, R2.КодДисц, R2.Оцінка}(R2)$$

КодСт	Прізвище	КодДисц	Оцінка
1001	Іванов	101	5
1001	Іванов	102	5
1002	Петров	101	5
1002	Петров	102	5

Рисунок 1.26 – Відношення *R3* після операції проєкції відношення *R2*

4. Операція природного з’єднання відношень *R3* і *Дисципліни* (рис. 1.26):

$$R5 = \pi_{R3.КодСт, R3.Прізвище, R3.Оцінка, Дисципліни.Назва}(R4),$$

де $R4 = (\sigma_{R3.КодДисц = Дисципліни.КодДисц}(R3 \times Дисципліни))$.

Отримаємо відношення *R5*, у якому будуть дані про студентів, які отримали оцінки за всіма дисциплінами (рис. 1.27).

$$R5 = \pi_{R3.КодСт, R3.Прізвище, R3.Оцінка, Дисципліни.Назва}(R4)$$

КодСт	Прізвище	Оцінка	НазваДисц
1001	Іванов	5	Математика
1001	Іванов	5	Інформатика
1003	Петров	5	Математика
1003	Петров	5	Інформатика

Рисунок 1.27 – Студенти, які отримали оцінки «відмінно»

Можливі й інші варіанти розв’язання задачі. Спробуйте самостійно знайти інше вирішення поставленої задачі.

Операція з’єднання має велике значення для реляційних баз даних, оскільки при виконанні запитів користувача вона дозволяє з’єднати декілька відношень в одне відношення.

Ділення відношень $R \div S$

Ця операція найменш очевидна з усіх операцій реляційної алгебри. Кода й потребує докладнішого пояснення. Операція виконується над двома відношеннями $R(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)$ і $S(b_1, b_2, \dots, b_m)$, що мають у загальному випадку різні структури та частину однакових атрибутів b_1, b_2, \dots, b_m . Вважатимемо, що атрибути b_i відношень R і S ($i = 1, 2, \dots, m$) мають не тільки однакові імена, але й визначені на одному й тому самому домені, наприклад, це ключові атрибути відношень.

Назвемо множину атрибутів $\{a_j\}$ складеним атрибутом A , а множину атрибутів $\{b_j\}$ – складеним атрибутом B . Після цього розглянемо реляційне ділення «бінарного» відношення $R(A, B)$ на унарне відношення $S(B)$. Результатом ділення відношення $R(A, B)$ на відношення $S(B)$ є «унарне» відношення $C(A)$. Отже, у результаті операції ділення утворюється нове відношення C шляхом виключення з множини атрибутів відношення R множини атрибутів відношення S .

Кортежі відношення C складаються з множини кортежей відношення R , які визначені на атрибуті A та відповідають комбінації всіх кортежей відношенням S (рис. 1.28).

R		S	\div		$=$	$C = R \div S$																
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>A</td><td>B</td></tr> <tr><td>a</td><td>1</td></tr> <tr><td>a</td><td>2</td></tr> <tr><td>b</td><td>1</td></tr> <tr><td>b</td><td>2</td></tr> </table>	A	B	a	1	a	2	b	1	b	2		<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>B</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> </table>	B	1	2				<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>A</td></tr> <tr><td>a</td></tr> <tr><td>b</td></tr> </table>	A	a	b
A	B																					
a	1																					
a	2																					
b	1																					
b	2																					
B																						
1																						
2																						
A																						
a																						
b																						

Рисунок 1.28 – Операція ділення відношень R на S

Розглянуті вище операції реляційної алгебри реалізуються в мові маніпулювання даними СКБД, що забезпечує обробку реляційних таблиць. До таких мов належать, наприклад, мова SQL (Structured Query Language) і мова QBE (Query By Example).

Мова SQL, що реалізована в більшості СКБД, є реляційно повною, оскільки, крім операцій реляційної алгебри, вона містить повний набір операторів над кортежами – *Включити*, *Видалити*, *Змінити*, а також реалізує арифметичні операції та операції порівняння.

Семантика мови SQL здебільшого базується саме на реляційній алгебрі, що спрощує ознайомлення та вивчення цієї мови.

Питання для самоконтролю

1. Дайте характеристику структурної, цілісної та маніпуляційної частин реляційної моделі даних.
2. Дайте загальну характеристику операцій реляційної алгебри.
3. Дайте характеристику різновидам операції з'єднання.
4. Чим відрізняються операції еквіз'єднання та природного з'єднання?
5. Дайте характеристику операції умовного з'єднання. Наведіть приклад виконання операції умовного з'єднання.
6. Дайте характеристику операції проєкції . Наведіть приклад виконання операції проєкції.
7. Яку послідовність операцій необхідно виконати, щоб визначити прізвища студентів, які склали всі экзамени на добре та відмінно (рис. 1.23)?

Інфологічна модель даних

Поняття про інфологічну модель даних

Процес проєктування бази даних тривалий, вимагає обговорень із замовником і фахівцями в предметній області. При розробці корпоративних інформаційних систем проєкт бази даних є тим фундаментом, на якому будується вся система загалом. Питання про можливість кредитування часто вирішується експертами банку на підставі грамотно зробленого інфологічного проєкту бази даних. Отже, інфологічна модель повинна включати такий формалізований опис предметної області, який буде легко «читатися» як фахівцями з баз даних, так і кінцевими користувачами, а також буде придатним для оцінки глибини й коректності опрацьовування проєкту бази даних без прив'язки до конкретної СКБД.

Проблема представлення семантики даних є дуже важливою та актуальною для розробників інформаційних систем. Головним призначенням інфологічного моделювання є забезпечення можливості вираження семантики даних. Потреби проєктувальників баз даних у зручних засобах моделювання предметної області сприяли виникненню напряму інфологічного, або семантичного, моделювання даних. У

70-х роках XX сторіччя було запропоновано декілька моделей даних, які були названі інформаційними або семантичними, моделями.

Однією з найпопулярніших інформаційних моделей даних є модель «сутність–зв’язок», або ER-модель (Entity Relationship). На використанні різновидів ER-моделі заснована більшість сучасних підходів до проектування реляційних баз даних. Модель була запропонована Ченом у 1976 році. На сьогодні модель Чена «сутність – зв’язок» стала фактичним стандартом при інфологічному моделюванні баз даних. Загальноприйнятою стала скорочена назва ER-модель.

Моделювання предметної області базується на використанні графічних діаграм – ER-діаграм (Entity Relationship Diagram), що включають невелику кількість різнорідних компонентів. На основі аналізу предметної області визначаються основні об’єкти предметної області, будується схема бази даних у вигляді ER-діаграм. На схемі в графічній формі відображаються зв’язки між об’єктами та характеристики цих зв’язків. Потім за чіткими правилами здійснюється перехід від ER-діаграм до таблиць бази даних, здійснюється наповнення таблиць атрибутами й перевірка їх на виконання умов нормалізації. Визначаються ключові атрибути таблиць та зв’язки між таблицями. Результатом проектування є схема реляційної бази даних.

У зв’язку з наочністю подання концептуальних схем баз даних ER-моделі широко поширились в CASE-системах, що підтримують автоматизоване проектування реляційних баз даних. Більшість сучасних CASE-систем містять інструментальні засоби для опису даних у формалізмі моделі «сутність – зв’язок» та реалізують методи автоматичного перетворення проекту бази даних із ER-моделі в реляційну модель. Перетворення виконується в логічну модель, яка відповідає конкретній СКБД. Прикладом CASE-системи є проста й універсальна програма ERwin фірми PLATINUM, яка призначена для автоматизованого створення реляційних баз даних.

Усі CASE-системи мають розвинені засоби документування процесу розробки бази даних. Автоматичні генератори звітів дозволяють підготувати звіт про поточний стан проекту бази даних із докладним описом об’єктів бази даних і їхніх відношень як у графічному вигляді, так і у вигляді готових стандартних друкарських звітів, що істотно полегшує ведення проекту.

Зараз не існує єдиної загальноприйнятої системи позначень для ER-моделі, і різні CASE-системи використовують різні графічні нотації, але знання однієї з них, наприклад нотації Чена, дозволяють легко зрозуміти й інші нотації.

Мета інфологічного проектування предметної області

Метою інфологічного моделювання є забезпечення розробника бази даних концептуальною схемою бази даних на рівні уявлень про предметну область.

Як інструмент інфологічного моделювання використовуються різні варіанти ER-діаграм. За їхньою допомогою визначаються важливі для предметної області об'єкти (сутності), їхні властивості (атрибути) і стосунки один з одним (зв'язки). Інфологічне моделювання відображає той факт, що сутності мають зв'язки між собою, а атрибути належать сутностям.

Інфологічна модель даних відображає дані з погляду їхнього сенсу та в контексті користувача. Модель приховує технічні деталі й підкреслює найважливішу суть з погляду предметної області та користувача. Спосіб зображення моделі – діаграми «сутність – зв'язок» (ER-діаграми) будь-якого з наявних типів.

Зазвичай на практиці інфологічне моделювання використовується на стадії інфологічного проектування бази даних. При цьому в термінах моделі проводиться розробка концептуальної схеми бази даних, яка потім уручну перетворюється на реляційну схему. Цей процес виконується з використанням методик, у яких досить чітко обумовлені всі етапи такого перетворення.

Основні терміни та визначення щодо поняття сутності

Модель «сутність-зв'язок» має декілька базових понять, які утворюють початкові цеглинки, з яких будуються вже складніші об'єкти за заздалегідь визначеними правилами. Основні конструктивні елементи моделі «сутність-зв'язок»: сутність, екземпляр сутності, атрибути, зв'язки та асоціативні сутності.

Алгоритм визначення сутностей

Сутність – це клас однотипних об'єктів, інформація про яких має бути врахована в моделі бази даних. Оскільки сутність відповідає деякому

класу однотипних об'єктів, то передбачається, що в системі існує безліч екземплярів даної сутності. Наприклад, сутністю можуть бути люди, об'єкти, події та ролі, які всі вони грають. Ці об'єкти повинні мати екземпляри, що відрізняються один від одного й допускають однозначну ідентифікацію.

Екземпляр сутності – це конкретний представник певної сутності. Кожна сутність повинна мати унікальне в межах конкретної моделі ім'я, яке рекомендується задавати іменником в однині. Наприклад, сутність «ПРЕДМЕТ» («Предмет для вивчення») має екземпляр – «Математика» (конкретний предмет), а для сутності «ГРУПА» – екземпляром є, наприклад, «ГІС 2022» (конкретна студентська група).

Об'єкт, якому відповідає поняття сутності, має свій набір атрибутів – характеристик, що визначають властивості певного представника класу.

Атрибут сутності – це іменована характеристика деякої властивості сутності або зв'язку. Найменування атрибута має бути виражене іменником в однині (можливо, з прикметниками, що характеризують атрибут). Атрибут залежить тільки від сутності та ні від чого іншого. Він може мати тільки одне значення в певний момент часу. Наприклад, у сутності ПРАЦІВНИК може бути такий набір атрибутів: Табельний номер, Прізвище, Ім'я, По батькові, Дата народження, Кількість дітей.

Під час проектування атрибутів корисно ставити такі запитання:

Які дані про сутність ми хочемо зберігати?

Які властивості є в екземплярі цієї сутності?

Чи є в екземплярі сутності тільки один екземпляр цієї сутності?

Чи може змінюватися описана атрибутом характеристика сутності з часом?

Наприклад, у людини завжди є тільки одна стать, у будівлі – тільки одна висота в певний момент часу, у людини завжди є зростання віку.

Імена атрибутів рекомендується робити *унікальними* в межах усієї схеми бази даних. Оскільки атрибут описує тільки одну сутність, він і має бути у всій схемі один (за правилом «один факт – в одному місці»).

Розглянемо деякі правила, за якими корисно визначати атрибути.

Атрибут повинен описувати саме свою сутність. Це просте правило не так просто застосовувати на практиці. Утруднення виникають, коли в наявній частині моделі виявляється новий атрибут, і потрібно знайти для нього відповідну сутність. Якщо сутність відсутня, то модель доведеться

змінити.

Не варто об'єднувати в один атрибут різні дані. Це найчастіше відноситься до спроб з'єднати в одному атрибуті кількість і одиницю вимірювання, наприклад, зберігати в одному атрибуті щось подібне до «20 ящиків» і «100 пляшок» або «100 грн.» і «10 доларів». Потрібно розділяти подібні атрибути на атрибути для зберігання кількості та атрибути для зберігання одиниці вимірювання.

Не рекомендується створювати атрибути, які можуть мати різний сенс залежно від свого конкретного значення. Наприклад, атрибут «клієнт» може містити дані про фізичну або юридичну особу. Зрозуміло, що ці дані виявляться абсолютно різними за своїм сенсом, і без інформації про те, до якого типу відноситься клієнт, трактувати їх надто складно.

Не варто задавати «особливих» діапазонів значень і правил трактування даних. Тобто варто уникати ситуацій, коли, наприклад, атрибут оцінка містить оцінку, отриману студентом на іспиті (число 3, 4 або 5), у тому разі, якщо іспит складений успішно; NULL-значення, якщо термін складання іспиту ще не наступив, і негативне число, яке означає, що іспит здавався кілька разів (модуль значення), але так і не був зданий.

Не варто задавати атрибути, для яких важливо зберігати додаткову інформацію. Тут потрібно застосовувати просте правило: *якщо необхідно зберігати додаткову інформацію про атрибут, отже це і є сутність.* Така сутність зазвичай називається *атрибутивною сутністю*. Приклад атрибутивної сутності: аудиторія, в якій відбуваються заняття, могла б бути атрибутом сутності «Проведення заняття», але якщо потрібно зберігати характеристики аудиторії, такі як місткість, наявність дошки, то доцільно зробити її атрибутивною сутністю.

Зв'язки

Зв'язок – це з'єднання між двома або більше сутностями. Зазвичай зв'язок виражається дієсловом. *Екземпляром зв'язку є конкретний зв'язок між конкретними екземплярами сутностей.*

Приклади опису зв'язків між сутностями (екземплярами сутностей):
студент *входить* до групи (Іванов *належить* до групи ГІС2007);
працівник *може мати* декілька дітей (Петров *має* двох дітей).

Зв'язок між двома сутностями називається *бінарним*. Це основний тип зв'язку, за допомогою якого виражаються зв'язки іншого типу.

Для кожного з двох кінців бінарного зв'язку задаються основні

характеристики зв'язку:

назва, яка описує семантику зв'язку, з погляду однієї із сутностей;

кардинальне число (ступінь, потужність), яке описує кількість можливих зв'язків для кожної із двох сутностей;

ступінь участі сутності у зв'язку (обов'язковість зв'язку, модальність зв'язку, клас належності).

Бінарний зв'язок буває «один-до-одного» (1:1), «один-до-багатьох» (1:N) або «багато-до-багатьох» (N:M). Тип зв'язку задається опцією, яка називається потужністю, або кардинальністю, зв'язку. Крім того, зв'язок характеризується таким параметром, як модальність, який може набувати всього два значення: обов'язковий чи необов'язковий зв'язок.

Визначення первинного та вторинного ключів

Алгоритм визначення ключів відношення

Набір атрибутів сутностей має бути таким, щоб можна було розрізнити конкретні екземпляри сутності.

Ключем сутності є ненадмірний набір атрибутів, значення яких у сукупності є *унікальними* для кожного екземпляра сутності. Ненадмірність полягає в тому, що видалення будь-якого атрибута з ключа порушує його унікальність.

Ключ (можливий ключ, потенційний ключ) сутності – це мінімальний набір атрибутів, який може бути використаний для однозначної ідентифікації будь-якого екземпляра сутності. Ключі сутностей бувають атомарними та складеними. Атомарний ключ складається з єдиного атрибута. Складеним ключем є ненадмірний набір атрибутів (ключ не повинен містити зайвих атрибутів). Це означає, що якщо будь-який атрибут виключити із ключового набору, то атрибутів, що залишилися, буде вже недостатньо для ідентифікації кортежу.

Будь-яка сутність має принаймні один потенційний ключ. Дійсно, якщо ніякий атрибут або група атрибутів не є потенційним ключем, то через унікальність екземплярів сутності всі атрибути разом утворюють потенційний ключ. Набори ключових атрибутів сутності визначаються на ранніх стадіях проектування бази даних.

Наявність *ключа сутності* забезпечує виконання основного правила для сутності: *кожен екземпляр сутності повинен бути унікальним і чітко*

відрізнятися від будь-якого іншого екземпляра цієї сутності й від будь-яких екземплярів інших сутностей. У такий спосіб виконується правило, що кожне явище або об'єкт може бути представлено тільки однією сутністю. Не може бути двох однакових сутностей.

Під алгоритмом визначення ключів будемо розуміти правила, за якими здійснюється вибір ключів відношення.

Первинним ключем сутності може бути призначений один із її потенційних ключів. При цьому варто брати до уваги те, що на фізичному рівні первинний ключ зазвичай реалізується як стовпець або група стовпців із забороненими NULL-значеннями. Згідно з вимогами цілісності сутності для ключів сутності створюються унікальні індекси. Основними правилами, за якими призначаються первинні ключі сутності, є такі:

- первинний ключ повинен однозначно ідентифікувати будь-який екземпляр сутності;
- первинний ключ за можливості має бути найкомпактнішим із усіх потенційних ключів. Як найкращим типом даних для первинного ключа є цілий тип, значеннями якого є множина цілих чисел;
- первинний ключ може бути складеним, але збільшення кількості стовпців, що входять до нього, не відповідає вимозі компактності. Вимогу компактності також не вдається виконати, якщо, наприклад, як первинний ключ вибрати занадто широкий стовпець рядкового типу даних;
- значення первинного ключа не повинні піддаватися частим модифікаціям. Ідеально, якщо бізнес-логіка предметної області така, що ці значення взагалі не передбачається змінювати;
- правила модифікації первинного ключа повинні контролюватися внутрішньою бізнес-логікою предметної області, а не рішеннями, які приймаються над нею. Наприклад, у базі даних, що розробляється для потреб кадрового обліку підприємства, для сутності ПРАЦІВНИК не варто вибрати як первинний ключ серію та номер паспорта працівника. Хоча ці дані й володіють властивостями обов'язковості й унікальності, але їх зміна може бути ініційована працівником, а не адміністрацією підприємства;
- якщо серед інформації, що зібрана про сутність, не вдається виділити дані, що задовольняють перелічені вище вимоги, то рекомендується розглянути можливість створення сурогатного первинного ключа, який, не несучи ніякого семантичного навантаження, просто слугує ідентифікатором конкретного екземпляра сутності. Як сурогатний

первинний ключ вибираються будь-які коди або ідентифікатори. Зазвичай сурогатний первинний ключ прихований від кінцевого користувача. Прикладом сурогатного первинного ключа є табельний номер працівника, який подається у вихідних формах і відомий працівникам.

Зв'язки без права передачі

Екземпляр сутності в якийсь момент часу пов'язаний з одним екземпляром іншої сутності, а потім цей зв'язок може бути розірваний. Після цього зв'язок можливо встановити між іншими екземплярами сутності. Прикладом такої ситуації є перехід працівника в інший відділ або студента в іншу групу. Якщо подібне неприпустимо, то такий зв'язок називають зв'язком без права передачі. Зв'язком без права передачі є, наприклад, зв'язок між сутностями «ОСОБА» і «ПАСПОРТ», оскільки паспорт не можна передати іншій людині, а будь-яка спроба зміни паспорта веде до знищення старого й видачі нового.

Іншим варіантом зв'язку без права передачі є зв'язок між сутністю в тому випадку, коли потрібно зберігати відомості про час створення та розриву зв'язку між конкретними екземплярами сутності. Так, якщо для зв'язку між сутностями «СТУДЕНТ» і «ГРУПА» потрібно зберігати інформацію про момент часу, коли студент змінив групу, то такий зв'язок можна трактувати як зв'язок без права передачі.

При моделюванні бази даних існує низка правил, що стосуються права передачі зв'язку:

- зовнішній ключ, що реалізовує зв'язок із правом передачі, не потрібно робити частиною первинного ключа, оскільки такий ключ буде нестабільним при передачі зв'язку;
- немає сенсу відстежувати історію змін даних, об'єднаних зв'язком без права передачі, оскільки кожна зміна зводиться до знищення старого екземпляра та створення нового;
- дві сутності, що об'єднані зв'язком «один-до-одного» з правом передачі, ніколи не об'єднуються в одну.

Принципи побудови діаграм «сутність – зв'язок»

Як уже визначалось, інфологічну модель реляційної бази даних зазвичай подають у вигляді ER-діаграм у нотації Чена. На ER-діаграмах сутність позначається прямокутником, що містить ім'я сутності, а зв'язок – ромбом, від якого проведені лінії до кожної із сутностей, що бере участь у

зв'язку. Кінці ліній виділяються символами, що задають ступінь й обов'язковість зв'язку.

Розрізняють ER-діаграми для екземплярів сутностей і ER-діаграми для класів сутностей (далі – ER-діаграми).

Діаграми ER-екземплярів є детальною формою зображення зв'язків між сутностями. На цих діаграмах точками або квадратами зображуються всі допустимі значення ключових атрибутів сутності, що вступила у зв'язок, а за допомогою ліній показані можливі зв'язки між точками.

Приклад діаграми ER-екземплярів для бази даних *Викладач читає дисципліну* наведений на рисунку 1.29.

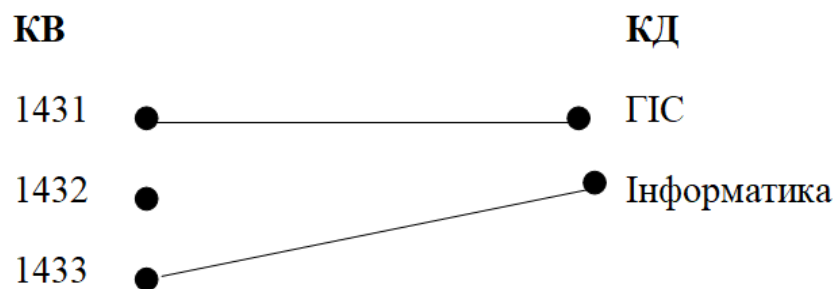


Рисунок 1.29 – Приклад подання діаграми ER-екземплярів

Тут: **КВ** (код викладача), **КД** (код дисципліни) – атрибути, які є первинними ключами сутностей «ВИКЛАДАЧ» і «ДИСЦИПЛІНА». На рисунку 1.28 показано, що один викладач читає одну дисципліну, але є викладачі, які не читають жодної дисципліни.

Діаграми ER-екземплярів є громіздкими, тому вони застосовуються в тих випадках, коли необхідно визначити правила переходу до моделей нижчих рівнів, наприклад до моделей відношень.

На ER-діаграмах сутність позначається прямокутником, що містить ім'я сутності, а зв'язок – ромбом, від якого проведені лінії до кожної із сутностей, що бере участь у зв'язку. Числа над лініями означають кардинальність зв'язку. Клас приналежності (модальність) зв'язку для кожної сутності показується наявністю прямокутної рамки навколо точки. Під прямокутниками вказується ключовий атрибут сутності (рис. 1.30).

Ключ зв'язку <**КВ, КТ**> складений із двох атрибутів. На першому етапі необхідні тільки атрибути, які є ключами сутностей. Інші атрибути та їхні функціональні залежності додаються до відношень на пізніших етапах.



Рисунок 1.30 – Приклад ER-діаграми в нотації Чена

У цьому прикладі показано зв'язок типу «один-до-одного» (один викладач читає тільки одну дисципліну). Для сутності «ВИКЛАДАЧ» зв'язок є необов'язковим (викладач може бути відсутній і не читає жодної дисципліни), а для сутності «ДИСЦИПЛІНА» – обов'язковим (дисципліну викладачі повинні читати обов'язково). Тому на стороні сутності «ДИСЦИПЛІНА» точка виділена прямокутною рамкою.

Питання для самоконтролю

1. Охарактеризуйте поняття «атрибут» реляційної БД.
2. Охарактеризуйте поняття «відношення» реляційної БД. Які види відношень ви знаєте?
3. Охарактеризуйте поняття «тип даних» реляційної БД.
4. Охарактеризуйте поняття «домен» реляційної БД.
5. Охарактеризуйте поняття «кортеж» реляційної БД.
6. Охарактеризуйте поняття «таблиця» реляційної БД.
7. Дайте визначення поняттю «первинний ключ». Наведіть приклади.
8. Дайте визначення поняттю «вторинний ключ». Наведіть приклади.
9. Охарактеризуйте всі види зв'язків між таблицями.

1.2 Принципи та теоретичні основи проєктування реляційних баз даних

Принципи проєктування реляційних баз даних

При проєктуванні бази даних вирішуються дві основні проблеми:

Проблема логічного проєктування баз даних, яка пов'язана з відображенням об'єктів предметної області в абстрактні об'єкти моделі даних у такий спосіб, щоб це відображення не суперечило семантиці предметної області та було за можливістю кращим (ефективним, зручним тощо).

Проблема фізичного проєктування баз даних, яка полягає в тому, щоб забезпечити ефективність виконання запитів до бази даних шляхом створення додаткових структур (наприклад індексів) та використання інших особливостей СКБД.

У цьому розділі обмежимося питаннями логічного проєктування реляційних баз даних, а конкретні рекомендації щодо фізичного проєктування будемо розглядати в методиці створення таблиць засобами СКБД Access.

Основна мета проєктування бази даних – це скорочення надмірності даних, що зберігаються, а отже, економія об'єму використовуваної пам'яті, зменшення витрат на багатократні операції оновлення надмірних копій і усунення можливості виникнення суперечностей через зберігання в різних місцях відомостей про один і той самий об'єкт.

Отже, вирішення проблеми логічного проєктування пов'язано з розв'язанням задач *мінімізації надмірності даних* та виключенням можливих *суперечностей збережених даних*.

Указані завдання є досить складними, але принцип їх виконання формулюється досить просто: *«Кожен факт повинен з'являтися лише в одному місці (відношенні) бази даних»*. Практична реалізація цього принципу базується на методології нормалізації відношень. У результаті такого підходу можна отримати обґрунтоване прийняття рішень про такі питання:

- з яких відношень повинна складатися база даних;
- які атрибути мають бути в цих відношеннях;
- які мають бути зв'язки між відношеннями у схемі бази даних.

При проєктуванні реляційних баз даних широко застосовуються

знайшли два підходи:

- реляційний (класичний) підхід за принципами нормалізації;
- інфологічне моделювання за принципами методології «сутність-зв'язок».

При реляційному підході етап інфологічного проектування не виконується. Проектування проводиться в термінах реляційної моделі даних методом послідовних наближень до задовільного набору схем відношень. Початковою точкою є подання предметної області у вигляді одного або декількох відношень, і на кожному кроці проектування створюється деякий набір схем відношень, що мають кращі властивості. Процесом проектування є процес нормалізації схем відношень, до того ж кожна наступна нормальна форма має кращі властивості, ніж попередня.

Кожній нормальній формі відповідає деякий певний набір обмежень. Відношення знаходиться в деякій нормальній формі, якщо задовольняє певний набір обмежень. Прикладом набору обмежень є обмеження першої нормальної форми – значення всіх атрибутів відношення є атомарними. Оскільки вимога першої нормальної форми є базовою вимогою класичної реляційної моделі даних, вважатимемо, що початковий набір відношень уже відповідає цій вимозі.

Реляційна модель даних достатня для моделювання предметних областей різного призначення. Проте проектування реляційної бази даних із використанням принципів нормалізації відношень є дуже складним і незручним, оскільки проектування виконується вручну.

Реляційна модель даних має обмеженість у таких аспектах:

Модель не надає достатніх засобів для представлення сенсу даних. Семантика реальної предметної області повинна незалежним від моделі способом уявлятися проектувальником. Зокрема, це стосується до проблеми подання обмежень цілісності.

Хоча весь процес проектування відбувається на основі врахування залежностей, реляційна модель не надає засобів для їх подання.

Незважаючи на це, реляційний підхід до проектування бази даних доцільно використовувати в тих випадках, коли необхідно прискорити процес проектування та впровадження системи в експлуатацію. Skorиставшись реляційним підходом, можна спроектувати оптимальну логічну модель бази даних. Така модель бази даних не має аномалій, пов'язаних із модифікацією бази даних, тобто проблем, що можуть

виникнути внаслідок заміन, вставлення і вилучення даних. Під аномаліями розуміють відхилення від норм, які можуть призвести до порушення посилкової цілісності бази даних чи до виникнення протиріч і неузгодженості даних.

Відсутність невинправданої надмірності даних

Формальне визначення терміна «надмірність бази даних» дати неможливо. Але є загально прийняті трактування стандартних ситуацій, які пов'язані з надмірністю даних. Перша типова ситуація пов'язана з визначенням первинних ключів відношень. Так, не раціонально ідентифікувати кортежі відношень первинними ключами з доменів, що містять текстові дані, тобто недоцільно як первинні ключі використовувати текстові дані. Друга типова ситуація виникає, коли одні й ті самі факти можна отримувати з різних об'єктів бази даних.

Наявність надмірності може принести або користь, або бути недоліком із погляду загальної продуктивності системи. Користь від внесення надмірності до бази даних полягає в можливості прискорення виконання запитів, оскільки частина потрібної інформації вже збережена в готовому вигляді, і не потрібно розшукувати її по різних таблицях, здійснюючи досить складні з'єднання таблиць та обчислення. Але за все потрібно платити – надмірність має бути контрольованою. Тобто необхідна програмна реалізація перевірок того, що надмірні й базові дані адекватно узгоджені між собою.

Ще одним прикладом надмірності є наявність у базі даних обчислюваних атрибутів та пов'язаних із ними обмежень. Обчислюваними називаються атрибути, значення яких можуть бути обчислені на підставі значень базових атрибутів, що зберігаються в базі даних.

З погляду способів реалізації важливо, чи обчислюються похідні атрибути за даними одного запису однієї таблиці, або для їх обчислення потрібно використовувати дані з різних записів або таблиць. У першому випадку можна в таблиці створити обчислюваний стовпець, а в другому випадку потрібна спеціальна процедура. Наприклад, у таблиці є стовпець із датою народження. Щоб дізнатися про поточний вік людини (кількість повних років), можна створити обчислюваний стовпець. Якщо в накладній потрібно встановити взаємозв'язок атрибутів «Ціна × Кількість = Сума», то імовірніше знадобиться процедура, оскільки значення ціни навряд чи зберігається в тій самій таблиці, де зберігаються дані про товарні позиції

накладної.

При інфологічному моделюванні за принципами методології «сутність-зв'язок» спочатку виконується проектування інфологічної моделі. У результаті проектування будується концептуальна схема бази даних із використанням ER-діаграм. Далі здійснюється відображення ER-моделі на реляційну модель за формальними правилами.

Теоретичні основи проектування реляційних баз даних

Основні поняття теорії нормалізації

Введемо поняття: нормалізація, функціональна залежність атрибутів, декомпозиція, принцип нормалізації та типи нормальних форм.

Нормалізація відношень – це ітераційний зворотний процес розбиття початкового відношення на кілька простіших відношень меншої розмірності. Під зворотністю процесу розуміють те, що операція об'єднання отриманих нових відношень має дати початкове відношення без втрат інформації.

Функціональна залежність атрибутів. У відношенні $R(A, B)$ атрибут B залежить від атрибута A ($A \rightarrow B$) тоді, коли в кожний момент часу одному й тому самому значенню A відповідає одне значення B .

Декомпозиція та принцип нормалізації. В основу нормалізації покладено принцип декомпозиції. Декомпозиція – це процес розбиття одного відношення на кілька простіших відношень меншої розмірності шляхом застосування до початкового відношення операції проєкції реляційної алгебри. У результаті використання операції проєкції до існуючого відношення отримуємо нове відношення шляхом вибору певних стовпців із поточного відношення. Якщо результат виконання операції проєкції містить кортежі, що повторюються, то залишається тільки один кортеж із групи, що повторюється.

Отриманий у результаті нормалізації склад атрибутів відношень бази даних повинен відповідати таким вимогам: між атрибутами не має бути небажаних функціональних залежностей, групування атрибутів має забезпечувати мінімальне дублювання даних, їх обробку й поновлення без ускладнень і аномалій. Водночас отримані в результаті декомпозиції відношення не повинні втратити функціональних залежностей початкового відношення, бо це може призвести до спотворення семантики цього відношення.

Аномалії ненормалізованого відношення

Структура реляційних відношень у нормалізованій базі даних має бути оптимальною, тобто такою, яка є найстійкішою при внесенні змін у дані та зв'язки між ними. У ненормалізованому відношенні можуть виникнути аномалії, які пов'язані з виконанням операцій підтримки його в актуальному стані. Виявити фактичний прояв тієї чи іншої аномалії можна, лише врахувавши конкретну семантику даних. Проілюструємо це на прикладі конкретного відношення:

КОМПАНІЯ (Код_працівника, НомерВідділу, Керівник, ТипКонтракту).

Аномалія оновлення. Заміна керівника призведе до необхідності внесення змін і модифікацій за кожним працівником конкретного відділу. Для цього у відношенні КОМПАНІЯ потрібно знайти всі записи про керівника, і в кожній з них узгоджено змінити дані про керівника. Отже, для підтримання узгодженості даних необхідно виконати низку змін, що може розглядатись як аномалія, оскільки характер самої зміни повинен стосуватись лише одного певного запису бази даних.

Аномалія поповнення. Приймаючи на роботу нового працівника, необхідно вносити в базу даних відомості не лише про нього, а й про керівника та тип контракту. Це можна розглядати як аномалію, оскільки не завжди можливо внести в базу даних відомості про відділ і контракт, оскільки часто працівників беруть на роботу з випробувальним терміном і лише після його проходження підписують певний вид контракту та визначають відділ, де він працюватиме. Отже, доведеться створювати запис із NULL-значеннями в полях НомерВідділу та ТипКонтракту.

Аномалія видалення. Припустимо, що звільнились усі працівники певного відділу, тоді разом з інформацією про них у базі даних буде втрачено також інформацію про цей відділ. Якщо ж інформацію про відділ необхідно зберігати якийсь час, то це також може розглядатись як аномалія.

Аномалія надлишковості. Відомості про керівника відділу та тип контракту повторюються в ряді кортежів бази даних. Основні проблеми зі зберіганням надлишку інформації пов'язані не лише з неефективним використанням пам'яті, а й із забезпеченням підтримки узгодженості цих даних.

Описані проблеми пов'язані з поганою нормалізацією відношень.

Можливість виникнення аномалій у процесі модифікації ненормалізованих відношень свідчить про неадекватність моделі даних предметній області. Це означає, що логічна модель даних або просто неправильна, або необхідні додаткові зусилля для реалізації всіх обмежень, що визначені в предметній області. Так, для підтримки відношення КОМПАНІЯ у цілісному стані доведеться витратити додаткові зусилля на створення спеціальних процедур, витратити дисковий простір на зберігання надмірних даних та неминуче вводити NULL-значення.

Усі ці проблеми можуть бути вирішені ще на стадії логічного проектування, якщо застосовувати спеціальні методики нормалізації реляційної бази даних. Для усунення аномалій оновлення застосовується метод нормалізації відношень, заснований на поняттях функціональної залежності (ФЗ) і нормальних форм (НФ).

Функціональні залежності атрибутів у відношенні

Кожне відношення БД уміщує як структурну, так і семантичну інформацію. Структурна інформація задається схемою відношення, а семантична виражає функціональні зв'язки між атрибутами.

У теорії нормалізації визначені такі зв'язки між атрибутами: функціональна залежність, тривіальна, нетривіальна та багатозначна залежності, повна функціональна залежність і транзитивна залежність.

Функціональна залежність (ФЗ). Нехай R – відношення. Множина атрибутів Y , яка належить R ($Y \in R$), функціонально залежна від множини атрибутів X ($X \in R$) тоді і лише тоді, коли для будь-якого стану відношення R у всіх кортежах, що мають однакові значення атрибутів X , значення атрибутів Y також збігаються.

Форми наочного подання функціональних залежностей показані на рисунку 1.31. Множина атрибутів X називається аргументом функціональної залежності, а множина атрибутів Y – залежною частиною.

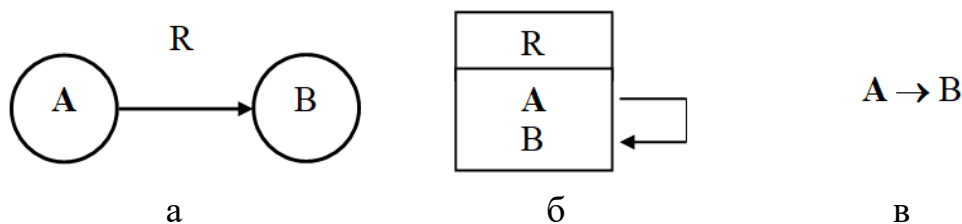


Рисунок 1.31 – Форми подання ФЗ:
а – граф; б – схема; в – символічна

Означення 1. Функціональна залежність – це така залежність між двома атрибутами (множинами атрибутів) A і B , коли у будь-який момент часу для будь-якого значення атрибута A існує одне й лише одне значення атрибута B . Атрибут A може бути як атомарним, так і складеним. Цій залежності відповідає співвідношення 1:1 між атрибутами, наприклад, КодПрацівника \rightarrow Прізвище, КодДисципліни \rightarrow Назва.

Варто зауважити, що ФЗ не можуть бути виведені із зовнішнього вигляду відношення в конкретний момент часу. Вони відображають взаємозв'язки між об'єктами предметної області, а також є додатковими обмеженнями, що визначені предметною областю. Отже, ФЗ – це семантичне поняття. ФЗ виникає, коли за значеннями одних даних у предметній області можна однозначно визначити значення інших даних.

Тривіальна, нетривіальна та багатозначна залежності

Означення 2. Функціональна залежність $X \rightarrow Y$ є *тривіальною*, якщо X також функціонально залежить від Y , тобто $Y \rightarrow X$.

Означення 3. Залежність $X \rightarrow Y$ є *нетривіальною*, якщо X функціонально не залежить від Y .

Багатозначна залежність є різновидом функціональної залежності, їй відповідає співвідношення 1:Б між атрибутами.

Означення 4. У відношенні $R(A, B, C)$ атрибут A багатозначно визначає атрибут B , якщо B залежить лише від A при будь-яких комбінаціях A з іншими атрибутами відношення R . Цю залежність позначають так: $A \twoheadrightarrow B$.

Приклад багатозначної залежності.

Нехай задано відношення

ПРАЦІВНИК (КодПрацівника, Прізвище, ДержавнаНагорода).

У цьому відношенні є багатозначна залежність:

КодПрацівника \twoheadrightarrow Державна нагорода.

Повна функціональна залежність

Якщо така залежність відношення має складені ключі, то залежність неключових атрибутів від такого ключа може бути повною або частковою.

Означення 5. Атрибут перебуває в повній функціональній залежності, якщо він залежить від усього ключа та не залежить від його складових частин.

Приклад. Маємо відношення $R(A, B, C, D)$, ключ якого складається із двох атрибутів A і B , тобто є складеним (ключові атрибути виділені).

Функціональні залежності у відношенні R показані на рисунку 1.32.

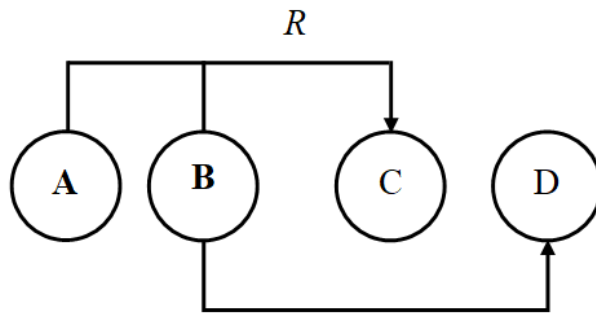


Рисунок 1.32 – Функціональні залежності відношення $R (A, B, C, D)$

Атрибут C перебуває в повній функціональній залежності, оскільки залежить від усього складеного ключа A, B , а атрибут D – у неповній, оскільки залежить лише від його складової частини – атрибута B .

Транзитивна залежність

Транзитивна залежність – це залежність між неключовими атрибутами. Нехай є відношення $R(A, B, C)$, у якому атрибут D безпосередньо не залежить від ключового атрибута A , а залежить від неключового атрибута C , який, зі свого боку, залежить від A (рис. 1.33). Тоді атрибут C транзитивно залежить від A (пунктирна лінія).

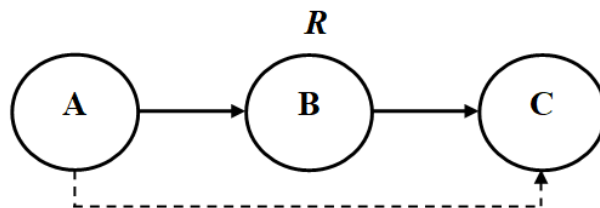


Рисунок 1.33 – Схема транзитивної залежності атрибута C від A

Неповні та транзитивні залежності вилучають за допомогою декомпозиції відношення на два чи більше інших відношень, які не містять цих залежностей і об'єднання яких дасть початкове відношення.

Правила виведення. Мінімальне покриття

Одна із проблем, пов'язана із застосуванням ФЗ, полягає в тому, що зазвичай при використанні формального визначення вдається виявити значно більше ФЗ, ніж потрібно для коректної нормалізації. Із цієї причини залишають мінімально можливий набір ФЗ, який називається мінімальним покриттям.

Мінімальне покриття забезпечує коректне проектування бази даних. Вибір ФЗ для мінімального покриття заснований на використанні такого апарату реляційної алгебри, як правила виведення ФЗ. Розглянемо прості правила виводу, що дозволяють видаляти надлишкові ФЗ:

- Транзитивність ФЗ (рис. 2.3). Якщо $A \rightarrow B, B \rightarrow C$, то $A \rightarrow C$.
- Додавання ФЗ. Якщо $A \rightarrow B$, то $A, C \rightarrow B$.
- Об'єднання ФЗ. Якщо $A \rightarrow B$ і $A \rightarrow C$, то $A \rightarrow B, C$.
- Декомпозиція ФЗ. Якщо $A \rightarrow B, C$, то $A \rightarrow B$ і $A \rightarrow C$.
- Псевдотранзитивність ФЗ. Якщо $A \rightarrow B$ і $B, D \rightarrow C$, то $A, D \rightarrow C$.

Графічно правила виводу 2–5 проілюстровані на рисунку 1.34.

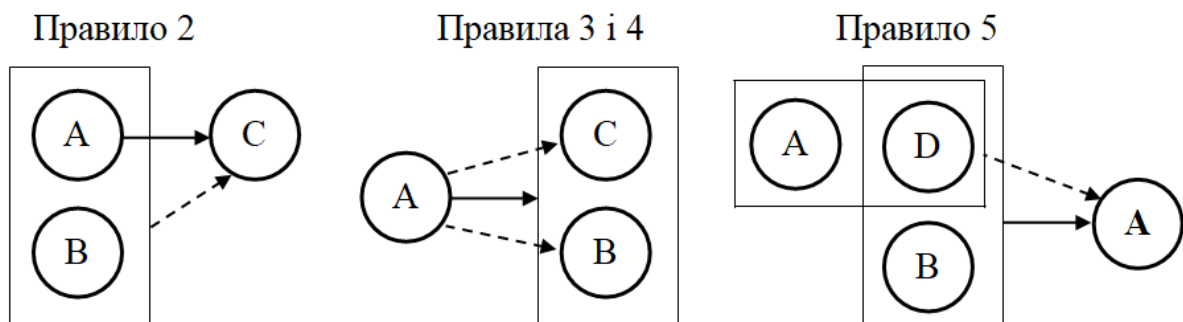


Рисунок 1.34 – Графічне подання правил виводу ФЗ

Для правила 1 (рис. 1.33) та правил 2 і 5 пунктиром показані надмірні ФЗ, що підлягають видаленню. Правила 3 і 4 є взаємно зворотними й дозволяють вибрати одну із комбінацій ФЗ (суцільну стрілку або дві пунктирні).

Нормальні форми відношень

Перша нормальна форма (1НФ)

Першим кроком нормалізації є приведення відношення до першої нормальної форми. Відношення в 1НФ повинно відповідати таким вимогам:

- усі атрибути відношення повинні бути унікальними, тобто не допускається їхнього дублювання, а також атомарними, тобто неподільними;
- усі рядки таблиці повинні мати однакову структуру;
- імена стовпців повинні бути різними, а значення однорідними (однакового типу й формату);
- порядок рядків у таблиці не істотний.

Отже, будь-яка нормалізована таблиця знаходиться в 1НФ.

Розглянемо приклад зведення відношення до 1НФ:

МАТЕРІАЛ (КодМатеріалу, Назва, Характеристика: тип, сорт, розмір).

У цьому відношенні атрибут «Характеристика» є неатомарним, тому потрібно позбутися складеного атрибута й перетворити його в три атомарних атрибути. У результаті зведення до 1НФ відношення матиме вигляд:

МАТЕРІАЛ (КодМатеріалу, Назва, Тип, Сорт, Розмір).

Друга нормальна форма (2НФ)

Відношення знаходиться у другій нормальній формі тоді й тільки тоді, коли відношення перебуває в 1НФ і всі його неключові атрибути функціонально повно залежать від первинного ключа. **Неключовий атрибут** – це атрибут, що не входить до складу первинного, або потенційного, ключа.

Якщо відношення має неповні функціональні залежності, то виконують його декомпозицію на два чи більше відношень, які не мають неповних функціональних залежностей і об'єднання яких дасть початкове відношення. Виконавши декомпозицію відношення R (див. рис. 1.32), отримаємо два відношення $R1$ і $R2$, які будуть перебувати у 2НФ (рис. 1.35).

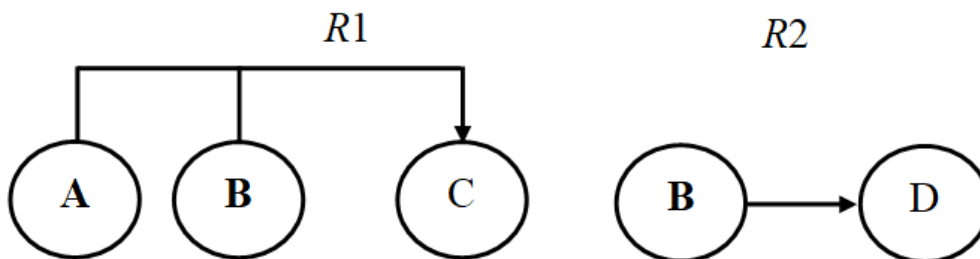


Рисунок 1.35 – Функціональні залежності відношень $R1(A, B, C)$ і $R2(B, D)$

У відношенні $R1$ є тільки один неключовий атрибут C , який функціонально повно залежить від складеного первинного ключа. Відношення $R2$ перебуває у 2НФ із тривіальної причини – воно має атомарний ключ, від якого залежить атрибут D .

Переваги 2НФ: зручність внесення змін у базу даних. 2НФ повністю виключає можливість виникнення протиріччя даних, а також економить пам'ять при зберіганні відношень у пам'яті ЕОМ. Відношення у 2НФ потрібно аналізувати на присутність транзитивних залежностей.

Третя нормальна форма (3НФ)

Відношення знаходиться у третій нормальній формі тоді і тільки тоді, коли воно перебуває у 2НФ і всі його неключові атрибути взаємно функціонально незалежні, тобто кожний неключовий атрибут не має транзитивної залежності від первинного ключа.

Наприклад, відношення

ВИКЛАДАЧ (**КодВикладача**, Прізвище, Посада, Оклад, КодКафедри, Телефон)
перебуває у 2НФ, але вміщує транзитивну залежність (рис. 1.36).



Рисунок 1.36 – Функціональні залежності відношення ВИКЛАДАЧ

У результаті зведення до 3НФ одержуємо два відношення:

ВИКЛАДАЧ (**КодВикладача**, Прізвище, Посада, Оклад, КодКафедри);

КАФЕДРА (**КодКафедри**, телефон кафедри).

Переваги 3НФ: виключається надлишкове дублювання інформації про телефон для викладачів однієї кафедри, спрощується процес внесення змін, оскільки у випадку зміни телефону будь-якої кафедри потрібно внести зміни лише в один відповідний запис, а не по всіх викладачах кафедри. Крім того, якщо викладач буде працювати на іншій кафедрі, необхідно змінювати лише код кафедри й не потрібно змінювати номер телефону, що довелося б робити, якби відношення зберігалось в ненормалізованому виді.

Тому можна зробити ще такий висновок: відношення знаходиться в 3НФ, якщо зміна значення будь-якого його атрибута (крім тих, що входять у первинний ключ) не призведе до необхідності зміни значень інших полів.

Нормальна форма Бойса – Кодда (НФБК)

Нормальна форма Бойса – Кодда – це підсилена ЗНФ, у якій вивчаються залежності ключових атрибутів від неключових.

Відношення перебуває в НФБК, якщо воно перебуває в ЗНФ і в ньому відсутні залежності ключових атрибутів від неключових атрибутів.

Приклад. Нехай задано відношення:

АВТОМОБІЛЬ (**МодельАвтомобіля**, КількістьЦиліндрів, КраїнаВиробник).

Залежності атрибутів у цьому відношенні показані на рисунку 1.37.

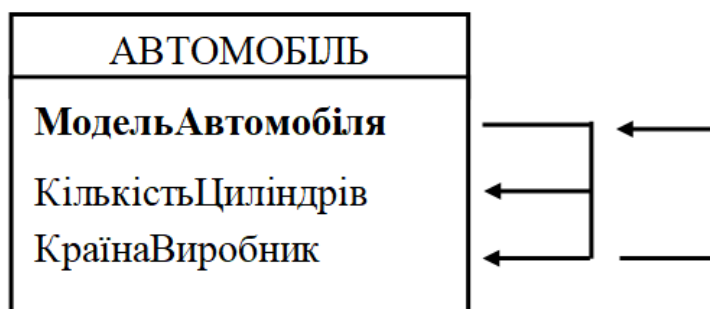


Рисунок 1.37 – Функціональні залежності відношення АВТОМОБІЛЬ

Ключовий атрибут **МодельАвтомобіля** залежить від неключового атрибута **КраїнаВиробник**. Тому при зведенні до БКНФ початкове відношення потрібно розбити на такі два відношення:

АВТОМОБІЛЬ (**МодельАвтомобіля**, КількістьЦиліндрів);

ВИРОБНИК (**КраїнаВиробник**, МодельАвтомобіля).

Зазначимо, що відношення, яке перебуває в НФБК, завжди є відношенням у ЗНФ. Але, навпаки, відношення в ЗНФ не завжди можна привести до нормальної форми Бойса – Кодда, не втративши залежності між його атрибутами.

Приклад. Нехай задано відношення:

ПОШТА (**Місто**, Адреса, Індекс).

Залежності атрибутів у цьому відношенні показані на рисунку 1.38.

Зведення відношення до ЗНФ дасть два відношення:

ПОШТА1 (**Місто**, Адреса),

ПОШТА2 (**Адреса**, Індекс).

Вони перебувають у ЗНФ, але не є відношеннями в НФБК, оскільки у відношенні ПОШТА існує залежність Індекс → Місто, тобто відношення в ЗНФ не завжди є відношенням у НФБК.

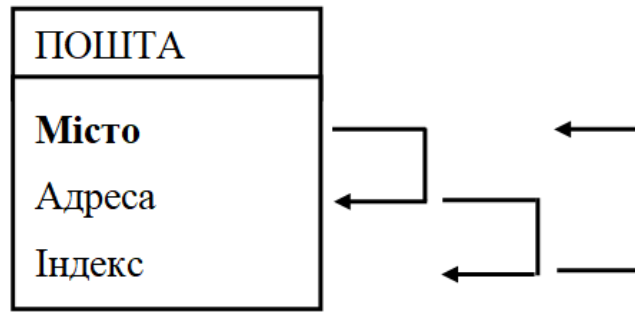


Рисунок 1.38 – Функціональні залежності відношення ПОШТА

Якщо ми зведемо це відношення до НФБК, то отримаємо такі відношення:

ПОШТА1 (Місто, Адреса),

ПОШТА2 (Індекс, Місто).

У цих відношеннях відображені не всі залежності початкового відношення. Так, не відображена залежність Індекс → Адреса, а саме індекс позначає відділення зв'язку, що обслуговує адресатів якоїсь вулиці певного міста.

Отже, перехід до НФБК може призвести до втрати важливих для початкового відношення залежностей, а об'єднання отриманих у результаті такої декомпозиції відношень не дасть початкового відношення. Тому при зведенні до НФБК необхідно ретельно вивчати всі залежності та виконувати його лише тоді, коли виконується вимога «об'єднання без втрат».

Нормальна форма Бойса – Кодда має такі самі переваги, що й ЗНФ, тобто виключає можливість виникнення аномалій, пов'язаних із виконанням операцій поповнення, вилучення та заміни даних, а також усуває надлишковість даних.

Зазвичай у базах даних задовольняються приведенням реляційної бази даних до ЗНФ (НФБК). Проте в теорії існують ще четверта й п'ята нормальні форми, які розглянемо на прикладах без строгих визначень.

Четверта нормальна форма (4НФ)

Якщо відношення має багатозначні залежності між атрибутами, то виконують його декомпозицію й отримують 4НФ.

Четверта нормальна форма, будучи достатньо складною, не вважається за обов'язкову на практиці, тому обійдемося при її розгляді без строгих математичних визначень. Розглянемо такий приклад. Нехай

потрібно враховувати дані про абітурієнтів, що вступають у ВНЗ.

При аналізі предметної області були виділені такі вимоги:

– Кожен абітурієнт має право скласти іспити на декілька факультетів.

– Кожен факультет має свій список предметів, що складають абітурієнти.

– Один і той самий предмет може здаватися на декількох факультетах.

Перезалік оцінок, які отримані за однойменними предметами на різних факультетах, не допускається. Абітурієнт зобов'язаний скласти всі предмети, що вказані для факультету, на який він вступає.

Припустимо, що в одному відношенні *Іспити* потрібно зберігати дані про предмети, які повинні здавати абітурієнти (табл. 1.3).

Таблиця 1.3 – Відношення «Іспити»

Абітурієнт	Факультет	Предмет
Іванов	Математичний	Математика
Іванов	Математичний	Інформатика
Іванов	Фізичний	Математика
Іванов	Фізичний	Фізика
Петров	Математичний	Математика
Петров	Математичний	Інформатика

Ключ відношення *Іспити* є складеним і містить усі три його атрибути. У відношенні зберігається інформація про те, що абітурієнт Іванов вступає на два факультети (математичний і фізичний), а абітурієнт Петров – тільки на математичний факультет. На математичному факультеті потрібно скласти математику та інформатику, а на фізичному – математику і фізику.

У відношенні *Іспити* є аномалії оновлення, вставки й видалення, які полягають у тому, що жодну з цих операцій не можна виконати для єдиного кортежу відношення без урахування інших його кортежів.

Аномалія додавання. При спробі додати в це відношення новий кортеж, наприклад (Котов, Математичний, Математика), необхідно додати й кортеж (Котов, Математичний, Інформатика), оскільки всі абітурієнти математичного факультету зобов'язані скласти однакові предмети.

Аномалія видалення. При спробі видалити кортеж (Іванов, Математичний, Математика) необхідно видалити й кортеж (Іванов, Математичний, Інформатика) з тієї ж самої причини. Якщо видалити кортеж (Іванов, Фізичний, Математика), а також і кортеж (Іванов, Фізичний, Фізика), то буде втрачена інформація про предмети, які повинні здаватися на фізичному факультеті.

Аномалія оновлення. При спробі змінити список іспитів, що здаються на факультеті, потрібно це робити для всіх абітурієнтів.

Отже, вставлення та видалення кортежів не можуть бути виконані незалежно від інших кортежів відношення.

Для нормалізації відношення *Іспити* не можна використовувати нормальні форми, засновані на ФЗ, оскільки в ньому є взаємозв'язок із багатозначною залежністю. На понятті багатозначної залежності засновано визначення 4НФ.

Відношення R знаходиться в 4 НФ тоді і тільки тоді, коли відношення знаходиться в НФБК і не містить нетривіальних багатозначних залежностей, таких, що для багатозначної залежності $X \twoheadrightarrow Y \mid Z$ не існує 3 НФ $X \rightarrow Y$ і $X \rightarrow Z$.

Відношення *Іспити* знаходиться в НФБК, але не в 4НФ. Використовуючи поняття багатозначних залежностей, можна без втрат виконати його декомпозицію на два відношення: ВСТУП (хто куди поступає) і ЕКЗАМЕН (де що здавати). Ці відношення такі:

ВСТУП (Факультет, Абітурієнт) і

ЕКЗАМЕН (Факультет, Предмет).

В отриманих відношеннях усунені аномалії вставлення та видалення, що характерні для відношення *Іспити*. Отримані відношення знаходяться в 4НФ, залишаючись повністю ключовими, і в них як і раніше немає ФЗ.

Відношення з нетривіальними багатозначними залежностями виникають, як правило, в результаті природного з'єднання двох відношень по загальному полю, яке не є ключовим в жодному із відношень. Фактично це приводить до спроби зберігати в одному відношенні інформацію про дві незалежні сутності. У цьому випадку список екзаменів, які здаються на різних факультетах, ніяк не залежить від того, які абітурієнти куди вступають.

Якщо виявиться необхідним зберігати оцінки, отримані під час складання іспитів абітурієнтами, то відношення *Іспити* доведеться

залишити в початковому стані, додавши до нього поле *Оцінка*, оскільки більше цю оцінку розміщувати ніде. Щоб отримати відношення для зберігання оцінок, потрібно виконати природне з'єднання відношень ВСТУП і ЕКЗАМЕН по полю *Факультет*, повернувшись до відношення *Іспити*. Після цього з'явиться можливість заносити оцінки, які функціонально залежать від складеного ключа відношення *Іспити*.

Як ще один приклад 4НФ можна навести ситуацію, коли співробітник може мати багато місць роботи й багато дітей. Зберігання інформації про місця роботи та дітей в одному відношенні приводить до виникнення нетривіальної багатозначної залежності (Працівник → Робота | Діти).

Етапи проєктування реляційних баз даних

Проєктування інфологічної моделі бази даних

На цьому етапі виконується аналіз предметної області, визначаються мета створення та вимоги до бази даних, основні об'єкти та зв'язки між ними, інформаційні потреби користувача (аналіз запитів), а також уточнюється послідовність дій майбутніх користувачів при роботі з базою даних і задачі, що будуть вирішуватися. На основі отриманої інформації розробляється концептуальна схема бази даних із використанням моделі «сутність-зв'язок».

Загальний підхід до побудови бази даних із використанням ER-методу полягає у виконанні таких дій:

Побудова діаграми ER-типу, в яку мають бути включені всі сутності та зв'язки, які є важливими з погляду інтересів організації.

Аналіз зв'язків і визначення їхніх характеристик: ступінь зв'язку та приналежність.

Побудова набору попередніх відношень із вказівкою передбачуваного первинного ключа для кожного відношення.

Підготовка списку всіх атрибутів і призначення кожного із цих атрибутів одному з відношень. Відношення повинні знаходитися в НФБК.

Побудова схеми даних.

Проєктування логічної схеми бази даних

На основі розробленої моделі даних визначається структура первинних таблиць бази даних, типи даних і ключові поля кожної таблиці.

Виконується нормалізація таблиць і будується логічна схема бази даних, у якій визначаються зв'язки між таблицями.

Правила отримання відношень із ER-діаграм залежать від приналежності та ступеня зв'язку сутностей. Поєднання трьох типів зв'язків із двома типами приналежності дають можливість опису множини різних варіантів зв'язків у предметній області. Розглянемо не всі, а тільки найпоширеніші варіанти зв'язків сутностей, відповідні ER-діаграми та правила побудови на основі їхніх відношень. Як приклади будемо використовувати правила взаємин між сутностями **ВИКЛАДАЧ** і **ДИСЦИПЛІНА**, що прийняті в різних навчальних закладах.

Правила для бінарних зв'язків 1:1

Правило 1. Якщо ступінь бінарного зв'язку 1:1 і клас приналежності обох сутностей є обов'язковим, то потрібне тільки одне відношення. Первинним ключем цього відношення може бути ключ будь-якої з двох сутностей, наприклад:

ДИСЦИПЛІНА ВИКЛАДАЧА (**КВ**, Прізвище, Посада, КД, Назва, Час).

Відношення гарантує одноразову появу кожного значення КВ і КД. Воно не буде мати порожніх даних та надмірних даних, що повторюються.

Правило 2. Якщо ступінь бінарного зв'язку 1:1 і приналежність однієї сутності є обов'язковою, а іншої – необов'язковою (рис. 2.10), то необхідна побудова двох відношень. Під кожну сутність виділяється одне відношення. Ключ сутності повинен слугувати первинним ключем для відповідного відношення. У сутність з обов'язковою приналежністю додається ключ сутності, для якої приналежність є необов'язковою (див. рис. 1.29).

ER-діаграма, де сутність **ВИКЛАДАЧ** має обов'язкову приналежність, а сутність **ДИСЦИПЛІНА** – необов'язкову, відображається на два відношення:

ВИКЛАДАЧ (**КВ**, Прізвище, КД);

ДИСЦИПЛІНА (**КД**, Назва, Час).

Відповідні таблиці наведені на рисунку 1.39.

Викладачі		
КВ	Прізвище	КД
V1	Іванов В. А.	Д1
V2	Петров С. М.	Д2

Дисципліни		
КД	Назва	Час
Д1	Математика	54
Д2	Інформатика	72

Рисунок 1.39 – Приклад таблиць для правила 2

У відношення ВИКЛАДАЧ і в таблицю ВИКЛАДАЧІ включено додатковий атрибут КД, який є зовнішнім ключем зв'язку.

Правило 3. Якщо ступінь бінарного зв'язку дорівнює 1:1 і приналежність жодної із сутностей не є обов'язковою (рис. 1.40), то необхідно використовувати три відношення: по одному для кожної сутності й одне відношення для зв'язку. Ключ кожної сутності використовується як первинний ключ відповідного відношення. Відношення зв'язку повинне мати з-поміж своїх атрибутів ключі кожної сутності.



Рисунок 1.40 – Сутності ВИКЛАДАЧ і ДИСЦИПЛІНА мають необов'язкову приналежність

Наведена ER-діаграма відображається на три відношення:

ВИКЛАДАЧ (КВ, Прізвище, КД);

ДИСЦИПЛІНА (КД, Назва, Час);

ДИСЦИПЛІНИ ВИКЛАДАЧА (КД, КВ).

Відповідні таблиці наведені на рисунку 1.41. Таблиця ДИСЦИПЛІНИ ВИКЛАДАЧІВ є зв'язковою таблицею.

Викладачі		
КВ	Прізвище	КД
V1	Іванов В. А.	Д1
V2	Петров С. М.	Д2

Дисципліни		
КД	Назва	Час
Д1	Математика	54
Д2	Інформатика	72

Дисципліни викладачів	
КВ	КД
V1	Д1
V2	Д2

Рисунок 1.41 – Приклад таблиць для правила 3

Правила для бінарних зв'язків 1:М

Для таких зв'язків використовуються два правила, кожне з яких визначається класом приналежності М-зв'язкової сутності. Приналежність 1-зв'язкової сутності на результат не впливає.

Правило 4. Якщо ступінь бінарного зв'язку дорівнює 1:М і приналежність М-зв'язної сутності є обов'язковою (рис. 1.42), то використовуються два відношення, по одному на кожну сутність. Ключ кожної сутності є первинним ключем для відповідного відношення. Відношення М-зв'язкової сутності повинне мати з-поміж своїх атрибутів ключ 1-зв'язкової сутності.



Рисунок 1.42 – Сутність ВИКЛАДАЧ має необов'язкову приналежність, а сутність ДИСЦИПЛІНА – обов'язкову

Наведена ER-діаграма відображається на два відношення:

ВИКЛАДАЧ (КВ, Прізвище, КД);

ДИСЦИПЛІНА (КД, Назва, Час, КВ).

Відповідні таблиці наведені на рисунку 1.43.

Викладачі			Дисципліни			
КВ	Прізвище	КД	КД	Назва	Час	КВ
В1	Іванов В. А.	Д1	Д1	Математика	54	В1
В2	Петров С. М.	Д2	Д2	Інформатика	72	В2

Рисунок 1.43 – Приклад таблиць для правила 4

У відношення та таблицю ДИСЦИПЛІНИ для організації зв'язку додатково включено ключ 1-зв'язкової сутності ВИКЛАДАЧ.

Правило 5. Якщо ступінь бінарного зв'язку рівний 1:М і приналежність М-зв'язкової сутності є необов'язковою (рис. 1.44), то необхідне формування трьох відношень: по одному для кожної сутності й одне відношення для зв'язку. Ключ кожної сутності використовується як первинний ключ відповідного відношення. Відношення зв'язку повинне

мати 3-поміж своїх атрибутів ключі кожної сутності.

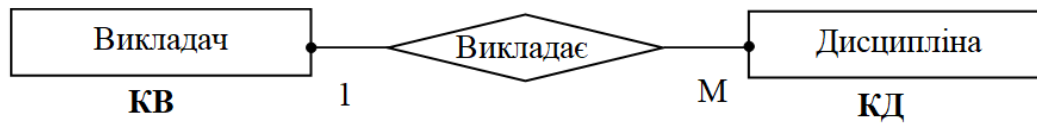


Рисунок 1.44 – Сутності ВИКЛАДАЧ і ДИСЦИПЛІНА мають необов'язкову приналежність

Наведена ER-діаграма відображається на такі самі три відношення та таблиці, як і для правила 3 (див. рис. 1.42).

Правило 6. Якщо ступінь бінарного зв'язку дорівнює М:М, то для зберігання даних необхідно три відношення: по одному для кожної сутності й одне відношення для зв'язку. Ключ кожної сутності використовується як первинний ключ відповідного відношення. Відношення зв'язку повинне мати 3-поміж своїх атрибутів ключі кожної сутності (див. рис. 1.41).

Проектування фізичної моделі бази даних

Починаючи з цього етапу, всі роботи виконуються в середовищі системи керування базами даних MS Access. На цьому етапі виконуються такі роботи.

1. Визначається спосіб створення бази даних – ручний, починаючи з «нуля», або автоматизований – з використанням майстрів MS Access.

2. Створюються структури таблиць: вводяться імена полів, задаються типи даних і властивості полів (розмір поля, формат поля, умови на значення поля та інше), визначаються первинні ключі таблиць.

3. Складається схема даних шляхом встановлення зв'язків між порожніми таблицями. Такий підхід дозволяє спростити внесення змін до структур таблиць у випадку, якщо MS Access виявить порушення цілісності даних. Заповнені таблиці коректувати набагато складніше.

4. Створюються всі об'єкти, які необхідні для роботи з базою даних: запити, форми, звіти, макроси та ін.

5. Виконується тестування бази даних. Перед введенням реальних даних проводиться заповнення таблиць набором спеціально підібраних «правильних» і «помилкових» тестових даних. За допомогою правильних тестових даних перевіряється структура таблиць. Дані з помилками

служать для перевірки реакції бази даних на допущені порушення. У процесі тестування бази даних перевіряється правильність виконання запитів та побудови форм і звітів. За необхідності вносяться зміни до бази даних.

Приклад проєктування бази даних

Як приклад, на якому буде розглянуто тему «Створення бази даних у СКБД Microsoft Access», розглянемо етапи проєктування бази даних типу «Облік нерухомості та мешканців квартир». Вибір прикладу обумовлений тим, що подібні бази даних можуть широко застосовуватися при обліку різноманітних об'єктів нерухомості та їх характеристик, починаючи із фізичних осіб, прописаних у квартирах, і закінчуючи різноманітними документами на права власності. У базі даних будуть враховані найістотніші об'єкти предметної області та характеристики цих об'єктів.

Основне призначення прикладу – показати підхід до проєктування та створення реляційних баз даних засобами СКБД Microsoft Access. Отримані знання будуть корисними при розробці власних баз даних для різних предметних областей.

Призначення додатка. Постановка задачі

Одному з муніципальних відділів поставлено завдання з обліку нерухомості, яка знаходиться на балансі міста.

Потрібно розробити базу даних «**Кадастр**», за допомогою якої буде вестися облік міських будівель, мешканців, нерухомого майна, яким володіють мешканці, та сплати податків. За результатами обліку нерухомого майна працівники муніципальних відділів виконують аналіз змін нерухомості, мешканців та своєчасності сплати податків.

Проєктування інфологічної моделі бази даних

При обліку нерухомості основними сутностями є такі: будівлі, квартири, мешканці та додаткова нерухомість (рис. 1.45).

Концептуальна схема бази даних «Кадастр» обліку нерухомості наведена на рисунку 1.46. Між сутностями існують бінарні зв'язки ступеня 1:М. Перехід від сутностей до відношень будемо здійснювати згідно з правилом 4 для всіх пар, у яких приналежність зв'язку М-зв'язної сутності є обов'язковою, а для сутностей **Нерухомість** – **Мешканець** (приналежність зв'язку сутності **Мешканець** є обов'язковою) – правило 5.

Виходячи з цього, до складу бази даних «Кадастр» включимо такі

відношення:

– Будівля (**Код_будівлі**, Індекс, Місто, Номер_будівлі, Фото_будівлі, Наявність_ліфту, Кількість_поверхів, Номер_ЖЕКу, Кадастровий_код, Рік_будування);

– Квартира (**Код_квартири**, Номер_квартири, Поверх, Кількість_кімнат, Загальна_площа, Житлова_площа, Наявність_балкону, Площа_балкона, Висота_стін, Тип_права, Особовий_рахунок);

Будівля	Квартира	Мешканець	Додаткова нерухомість
Код будівлі	Код квартири	Код мешканця	Код нерухомості
Індекс	Код будівлі	Код квартири	Додаткова нерухомість
Місто	Номер квартири	Прізвище	Житлова площа
Номер будівлі	Поверх	Ім'я	Ціна квадратного метра
Фото будівлі	Кількість кімнат	По батькові	
Наявність ліфту	Загальна площа	Ідентифікаційний код	
Кількість поверхів	Житлова площа	Квартиронаймач	
Номер ЖЕКу	Наявність балкону	Рік народження	
Кадастровий код	Площа балкона	Статус	
Рік будування	Висота стін		
	Тип права		
	Особовий рахунок		

Рисунок 1.45 – Сутності обліку нерухомості та їхні атрибути

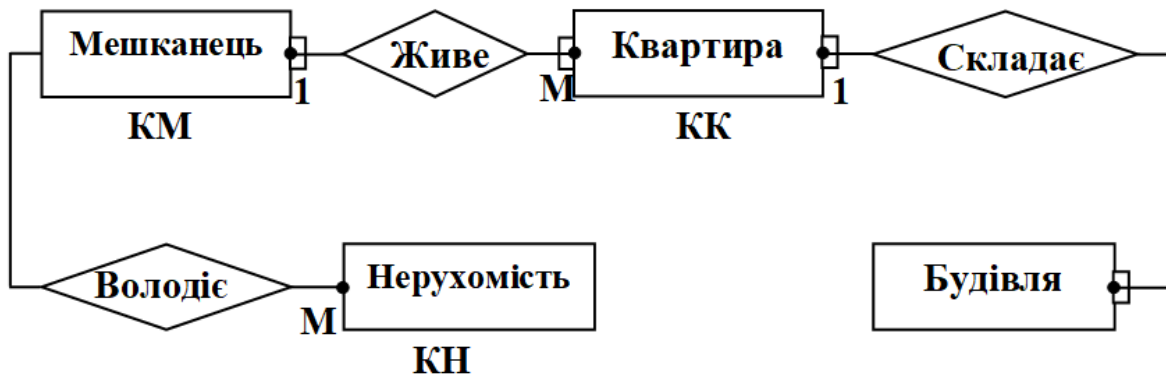


Рисунок 1.46 – ER-діаграма обліку нерухомості

- Мешканець (**Код_мешканця**, Прізвище, Ім'я, По_батькові, Ідентифікаційний_код, Квартиронаймач, Рік_народження, Статус);
- Додаткова нерухомість (**Код_нерухомості**, Назва_нерухомості, Житлова_площа, Ціна_м_кв);
- Податки (**Код_податку**, Сплата_податку, Ставка_податку, Дата_встановлення).

Згідно з правилом 5, крім сутностей **Квартира**, **Мешканець** та **Нерухомість**, створимо відношення, які характеризують будівлі:

- Район (**Код_району**, Назва_району);
- Вулиця (**Код_вулиці**, Тип_вулиці, Назва_вулиці);
- Матеріал (**Код_матеріалу**, Назва_матеріалу);
- Документ (**Код_документа**, Вид_договору, Номер_документа, Дата_реєстрації, Номер_у_книзі_записів);

У наведених відношеннях атрибути мають такі типи:

- Код_будівлі, Індекс, Кількість_поверхів, Номер_ЖЕКу, Рік_будування, Код_квартири, Номер_квартири, Поверх, Кількість_кімнат, Загальна_площа, Житлова_площа, Площа_балкона, Висота_стін, Особовий_рахунок, Код_мешканця, Ідентифікаційний_код, Рік_народження, Код_документа, Номер_у_книзі_записів, Код_вулиці, Код_району, Код_матеріалу, Код_податку, Код_нерухомості, Житлова_площа, Ціна_м_кв – тип **Числової**;

Дата_реєстрації, Дата_встановлення – тип **Дата/Время**;

- Місто, Номер_будівлі, Кадастровий_код, Тип_права, Прізвище, Ім'я, По_батькові, Статус, Вид_договору, Номер_документа, Назва_району, Тип_вулиці, Назва_вулиці, Назва_матеріалу, Додаткова_нерухомість – тип **Текстовий**;

Фото_будівлі – тип **Поле_объекта OLE**;

Наявність_ліфту, Наявність_балкону, Квартиронаймач,
Сплата_податку – тип **Логический**.

Проектування логічної схеми бази даних

Виконаємо нормалізацію відношень і побудуємо логічну схему бази даних. Відношення **Будівля** має атрибути **Район, Вулиця, Матеріал**, які не залежать від первинного ключа **Код_будівлі**. У результаті декомпозиції цього відношення отримаємо чотири відношення: Будівля, Район, Вулиця, Матеріал.

На основі отриманих відношень створені відповідні таблиці. Логічна схема нормалізованої бази даних «**Кадастр**» наведена на рисунку 1.47.

На цьому завершується етап «Розробка логічної схеми бази даних». Результати проектування бази даних «**Кадастр**» будемо використовувати при вивченні способів створення баз даних за допомогою засобів системи керування базами даних Microsoft Access.

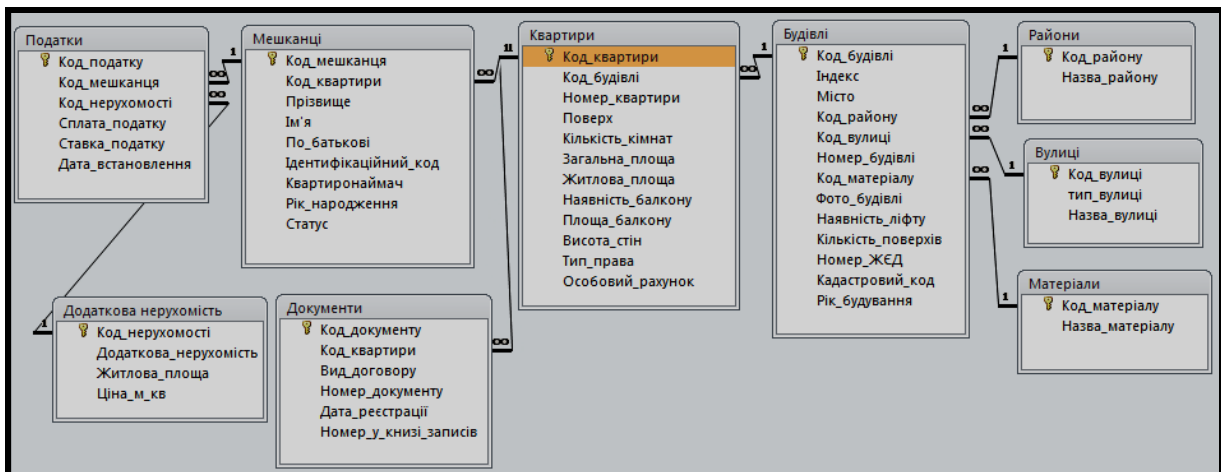


Рисунок 1.47 – Логічна схема бази даних «Кадастр»

Питання для самоконтролю

1. Дайте визначення проблемам і принципам проектування реляційних баз даних.
2. Що таке оптимальна логічна модель даних?
3. Що таке аномалія та якими аномаліями характеризується ненормалізоване відношення?
4. Дайте визначення функціональній залежності.
5. Дайте визначення неповній функціональній залежності.
6. Що таке транзитивна залежність?
7. Дайте визначення багатозначній залежності.

8. У чому полягає суть зведення реляційного відношення до 1НФ?
9. У чому полягає суть зведення реляційного відношення до 2НФ?
10. У чому полягає суть зведення реляційного відношення до 3НФ?
11. У чому полягає суть зведення реляційного відношення до 4НФ?
12. У чому полягає суть зведення реляційного відношення до НФБК?
13. Охарактеризуйте переваги нормалізованої бази даних.
14. Охарактеризуйте етапи проєктування реляційних баз даних.
15. Що виконується на етапі проєктування інфологічної моделі бази даних?
16. Поясніть сутність побудови ER-діаграми за нотацією Чена.
17. Які графічні символи використовуються в нотаціях Чена?
18. У чому полягає алгоритм визначення сутностей?
19. Поясніть сутність проєктування логічної схеми бази даних.
20. Навіщо формулюються основні задачі та вимоги до додатку перед проєктуванням інфологічної моделі?

Практичні завдання для самоконтролю

Завдання 1. У проєкті використовуються деталі, які постачаються кількома постачальниками. Кожний постачальник постачає деталі лише певного виду, тобто неможлива поставка одним постачальником деталей різного виду. Кожний проєкт має свого керівника та виконавців, закріплених за одним проєктом.

Потрібно спроектувати реляційну базу даних, що характеризує описану предметну область, і звести її до 3НФ (4НФ).

Завдання 2. Наведено реляційне відношення, що перебуває в 1НФ:

ПОСТАВКА (код постачальника, назва постачальника, адреса постачальника, телефон, номер розрахункового рахунку постачальника, код товару, назва товару, одиниця вимірювання, оптова ціна за одиницю, номер договору на постачання, дата укладання договору, дата виконання договору, кількість товару згідно з договором).

Потрібно побудувати діаграму функціональних залежностей та звести відношення до 3НФ (4НФ).

Завдання 3. Подано реляційне відношення, що перебуває в 1НФ:

УСПІШНІСТЬ (прізвище, ім'я та по батькові студента, номер залікової книжки, курс, група, код і назва спеціальності, код предмета, назва предмета, табельний номер, прізвище, ім'я та по батькові викладача, кафедра, де працює

викладач, посада викладача, дата здачі екзамену, оцінка).

Потрібно побудувати діаграму функціональних залежностей та звести до ЗНФ (4НФ), урахувавши, що один викладач може читати кілька дисциплін.

Завдання 4. Подано реляційне відношення, що перебуває в 1 НФ:

ДЕТАЛЬ (код деталі, назва деталі, собівартість деталі, код технологічної операції, назва технологічної операції, код і назва обладнання, на якому виконується операція, поопераційна трудомісткість обробки деталі).


Потрібно побудувати діаграму функціональних залежностей та звести відношення до ЗНФ (4НФ).

1.3 Аналіз даних за допомогою запитів. SQL запити

Система керування базами даних Microsoft Access – це програма, призначена для створення та супроводу реляційних баз даних.

База даних MS Access – це сукупність об'єктів (таблиць, форм, запитів та ін.), які використовуються для збереження, перегляду, зміни і виведення даних на друк. MS Access забезпечує введення даних у таблиці бази даних, їх зберігання та супровід, а також отримання із сукупності цієї інформації даних, необхідних для ухвалення важливих рішень.

Усі об'єкти БД зберігаються в одному файлі, який має унікальне ім'я, формат і розширення **.mdb** у версії MS Access 2003 та **.accdb** у версії MS Access 2010. Файл нової БД містить приховані системні таблиці, в які заноситься інформація про всі створювані об'єкти БД. У міру заповнення бази даних розмір її файлу швидко збільшується, оскільки у файл додаються нові об'єкти й дані, а інформація про об'єкти, що видаляються, зберігається.

Запуск MS Access виконується за допомогою команди **Пуск/Все програми/Microsoft Access** головного меню Windows або ярлика , розташованого на робочому столі. Після завантаження MS Access на екрані з'являється його робочий простір (рис. 1.48).

Об'єкти бази даних групуються за типами. У MS Access таких типів п'ять: таблиці, запити, форми, звіти та макроси (рис. 1.49). У БД об'єкти одного типу повинні мати унікальні імена. Імена об'єктів різних категорій можуть збігатися, наприклад: таблиця **Будівлі**, форма **Будівлі**, звіт **Будівлі** та ін. Проте імена таблиць і запитів повинні розрізнятися.

Таблиці складають основу реляційної бази даних і слугують для зберігання даних. Решта об'єктів БД пов'язана з таблицями та залежить від них. Таблиця складається із заголовка й тіла. Заголовок включає імена полів, а тіло – рядки (записи). Кожен запис містить дані про конкретний екземпляр об'єкта, наприклад інформацію про одного мешканця квартири (його прізвище, ідентифікаційний код і рік народження). При відкритті таблиці в режимі перегляду на екрані можна побачити всі поля та всі записи, що зберігаються в таблиці.

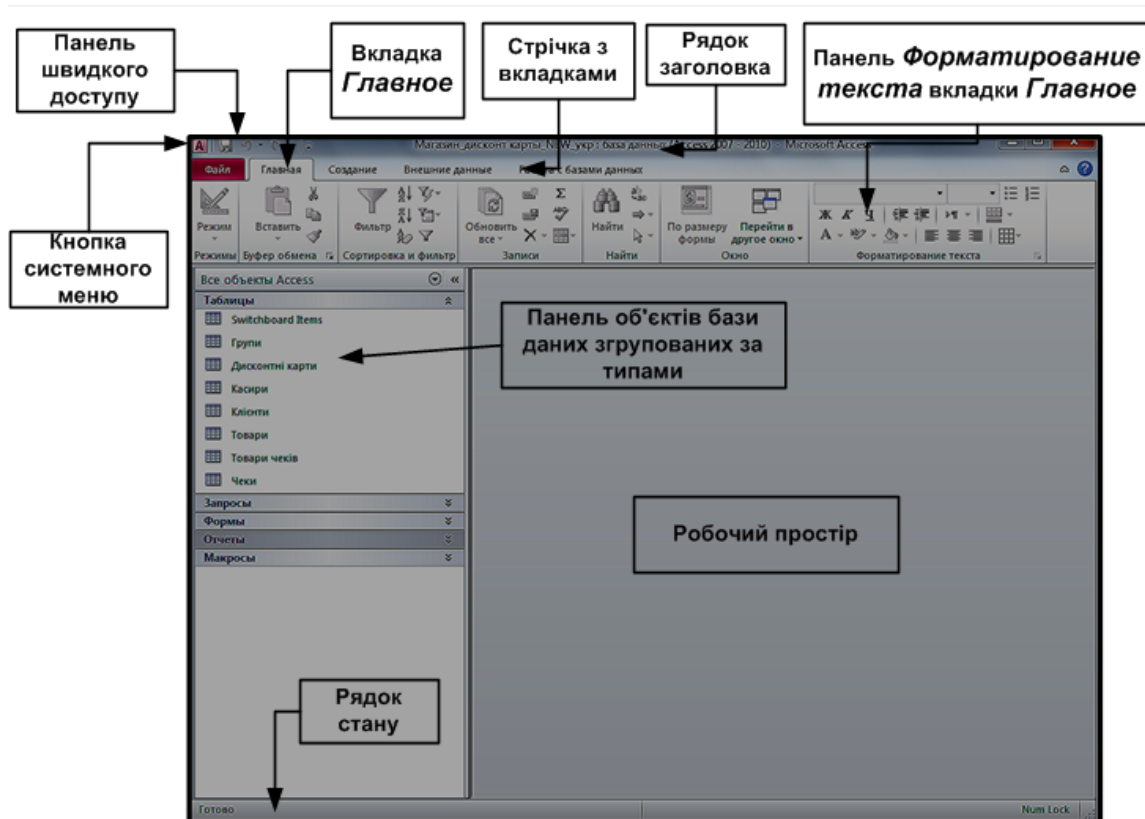


Рисунок 1.48 – Робочий простір MS Access



Рисунок 1.49 – Типи об'єктів бази даних

Запити призначені для управління даними й дозволяють вибирати з таблиць БД відомості, які відповідають певному критерію. За допомогою запитів можна автоматизувати оновлення або видалення записів в одній або декількох таблицях, а також виконувати обчислення на підставі значень, що зберігаються в таблиці. Крім того, можна створювати нові таблиці, використовуючи дані однієї або декількох таблиць, які вже

існують. У MS Access застосовуються запити за зразком QBE і запити, створені користувачем на мові побудови запитів SQL.

Форми використовуються для введення даних, відображення їх на екрані або управління роботою додатка. Вони дозволяють подати дані на екрані в наочній формі, а також є основним інструментом супроводу баз даних і засобом обмеження доступу до них. За допомогою форми можна запустити макрос або процедуру, що виконують обробку даних.

Звіти призначені для створення документів, які надалі можуть бути роздруковані або включені в документ іншого застосування. У звіті можуть бути зазначені результати підсумкових розрахунків та аналізу даних, що наведені у зручному й наочному вигляді, зокрема у вигляді графіків та діаграм.

Макроси – це прості програми, за допомогою яких можна автоматизувати виконання операцій, що часто повторюються, наприклад, відкриття форми, друк певного звіту або вибір команди меню. Використання макросів дозволяє виконувати рутинні процедури простим натисненням кнопок або комбінацій клавіш, а також активізацією спеціальних команд меню.

Модулі – це спеціальні підпрограми, написані на мові Visual Basic для реалізації нестандартних процедур додатка.

У файлі бази даних різні типи об'єктів можуть бути створені за допомогою конструктора або з використанням різних майстрів. Майстри допомагають створювати об'єкти в режимі діалогу, дають підказки користувачеві й пропонують свої рішення.

Дії над об'єктами

У робочому просторі бази даних об'єкти подаються у вигляді піктограм та імен (див. рис. 1.49). Над об'єктами бази даних можуть виконуватися такі дії: відкриття, закриття, модифікація об'єкта та його видалення. Усе це може бути здійснено за допомогою системи керування базою даних (СКБД), наприклад MS Access (рис. 1.50).



Рисунок 1.50 – Можливості СКБД щодо бази даних та її об’єктів

Об’єкти, що були відкриті, відображаються на екрані у вигляді вікна, наприклад, вікна таблиці, вікна форми. Вікна об’єктів використовуються для перегляду, введення або редагування даних. Відкрити об’єкт можна такими способами: двічі клацнути мишею на піктограмі об’єкта або виділити об’єкт і натиснути клавішу **Enter**. При закритті об’єкта вікно закривається.

Правила та прийоми роботи користувача в MS Access

Інтерфейс – це засоби та правила, що забезпечують взаємодію користувача з комп’ютером або додатком Windows. У MS Access реалізований стандартний віконний інтерфейс, прийнятий для додатків Windows. Основу інтерфейсу складають вікна. Крім того, використовуються стандартні діалогові та інші вікна Windows. Робота у вікнах MS Access виконується так само, як і у вікнах інших додатків.

Головне вікно MS Access відкривається при запуску додатка та містить стандартний набір елементів (див. рис. 1.48). Розглянемо їх призначення.

Рядок заголовка. Включає кнопку системного меню, назву додатка та кнопки управління вікном: згорнути, розгорнути й закрити вікно.

Стрічка з вкладками. Складається зі стандартного набору вкладок: **Файл, Главное, Создание, Внешние данные, Работа с базами данных.**

Панелі на вкладках. Містять кнопки для швидкого вибору команд.

Перелік кнопок залежить від призначення конкретної панелі.

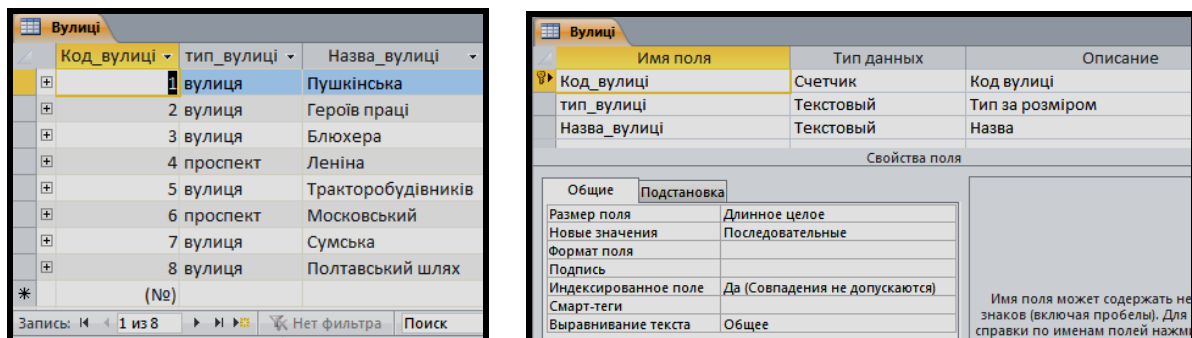
Рядок стану. Відображає інформацію про поточний режим роботи системи (**Готово, Конструктор** та ін.), підказки, підписи полів та ін.

Панель об'єктів бази. Призначена для роботи з об'єктами MS Access у різних режимах. Панель **Все объекты Access** слугує для вибору типів об'єктів за допомогою випадаючих меню.

Вікна об'єктів і конструктора

Залежно від режиму роботи з виділеним об'єктом відкривається або вікно об'єкта, або вікно конструктора (рис. 1.51).


Вікно об'єкта. Містить інформацію про відкритий об'єкт. У вікні об'єкта виконуються операції введення, перегляду або редагування даних. Вікно має стандартну структуру. Рядок заголовка вікна містить піктограму системного меню, ім'я об'єкта та кнопки управління вікном (згортання, розгортання та закриття). У робочій області вікон відображається інформація, а внизу розташовані навігаційні панелі.



а

б

Рисунок 1.51 – Вікна таблиці «Вулиці» в режимі таблиці (а) і в режимі конструктора (б)

Вікно конструктора. Надає можливість створювати новий об'єкт бази даних чи редагувати вже існуючий. Залежно від виду об'єкта, відкритого в режимі конструктора, його зовнішній вигляд може розрізнятися. Основне призначення вікна конструктора – створювати чи редагувати об'єкти бази даних. Незалежно від виду об'єкта перейти у режим конструктора можна за допомогою контекстного меню мишки, вибравши команду **Конструктор** .

Запит – це вимога на отримання даних, виконання дій з ними або те й інше.

Найчастіше використовуються запити на вибірку, де вибираються дані з однієї або декількох таблиць і подаються на екрані в табличній формі.

В Access також є запити на зміну для оновлення, додавання або видалення даних із таблиці та створення нових таблиць.

Для кожного інформаційного завдання створюється запит, а потім він може багато разів виконуватися.

Запит є програмою, що записана мовою **SQL (Structured Query Language)** – структурована мова запитів).

У Access створення запиту полягає у формулюванні постановки завдання в режимі конструктора (задавання імен таблиць даних, полів, операцій із цими полями, умов, які вони задовольняють, критеріїв групування та сортування).

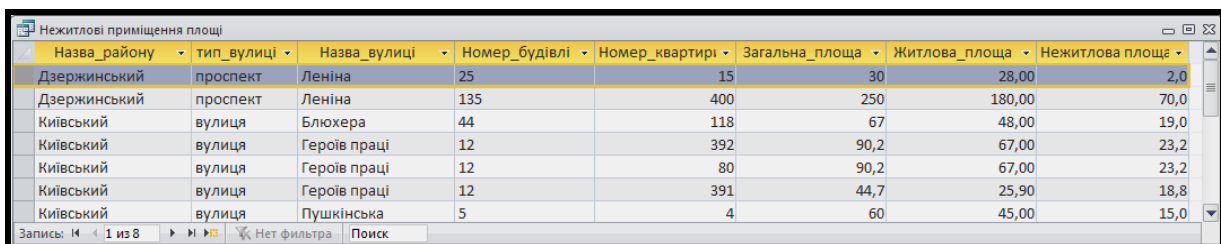
При цьому Access автоматично створює програму мовою SQL. Саме ця програма зберігається як запит у файлі БД Access.

Запити часто слугують основою для побудови нових запитів, джерелом даних для форм, звітів та елементів керування ними.

Виконуючи лабораторну роботу, можна навчитися будувати запити таких видів.

Запити на вибірку

Детальний запит – відображає кожний запис, що задовольняє умови відбору. У детальному запиті можна виконувати обчислення з використанням полів таблиць, що входять до запиту (рис. 1.52).



Назва_району	тип_вулиці	Назва_вулиці	Номер_будівлі	Номер_квартири	Загальна_площа	Житлова_площа	Нежитлова_площа	
Дзержинський	проспект	Леніна	25		15	30	28,00	2,0
Дзержинський	проспект	Леніна	135		400	250	180,00	70,0
Київський	вулиця	Блюхера	44		118	67	48,00	19,0
Київський	вулиця	Героїв праці	12		392	90,2	67,00	23,2
Київський	вулиця	Героїв праці	12		80	90,2	67,00	23,2
Київський	вулиця	Героїв праці	12		391	44,7	25,90	18,8
Київський	вулиця	Пушкінська	5		4	60	45,00	15,0

Рисунок 1.52 – Детальний запит з обчислюваними полями

Підсумковий запит – відображається один запис з кожної групи (рис. 1.53). Групи записів утворюються за ознакою співпадання значень зазначеного поля. Наприклад, якщо в таблиці «Квартири» є по кілька записів про райони, в яких розташовані квартири, то після групування за полем **Назва_району** у підсумковому запиті кожний район

відобразатиметься тільки один раз. Якщо в підсумковому запиті беруть участь ще й інші поля, за ними можна виконувати групування, утворюючи підгрупи або обчислення, визначаючи кількість записів у кожній підгрупі (для текстових полів) або статистичні дані (сума, середнє арифметичне максимальне тощо для числових полів).

Назва_району	Сумарна житлова площа, м2
Дзержинський	208,00
Київський	291,80

Рисунок 1.53 – Підсумковий запит

Перехресний запит – результат відображається у вигляді «шахівниці». Назви її стовпців і рядків є значеннями полів, а в клітинках на їхньому перетині – результат обчислень подібно до підсумкового запиту (рис. 1.54).

Назва_району	Итоговое значение Житлова_площа	Блюхера	Героїв праці	Леніна	Пушкінська
Дзержинський	208,00			208,00	
Київський	291,80	48,00	159,90		83,90

Рисунок 1.54 – Перехресний запит

Запити на зміну

На оновлення – змінює значення декількох полів у відібраних записах.

На додавання – додає декілька записів із запиту в таблицю.

На видалення – видаляє з таблиці декілька вибраних записів.

На створення таблиці – створює нову таблицю і додає до неї записи із запиту.

Редагування запитів

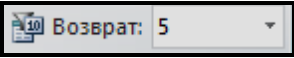
Для редагування запитів потрібно повернутися в режим конструктора та внести зміни.

Для додавання в запит додаткових таблиць або запитів слугує кнопка

Отобразить таблицу  на вкладці **Конструктор** панелі **Настройка**

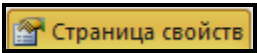
запроса. Для видалення зайвих таблиць і запитів потрібно виділити їх у верхній частині вікна й натиснути клавішу **Delete**.

Для видалення поля з запиту потрібно виділити це поле в бланку (нижній частині вікна) і натиснути клавішу **Delete**. Для виділення поля потрібно навести курсор миші над стовпцем поля так, щоб він набув форми чорної стрілки та клацнути лівою кнопкою. Виділені поля можна переміщати в бланку, перетягуючи їх за верхню сіру смугу.

У випадку якщо потрібно отримати не повний, а частковий список результатів роботи запиту (наприклад, перші 10 записів або 5 % усіх записів), можна на панелі інструментів вказати необхідне значення за допомогою кнопки **Возврат**  на вкладці **Конструктор** панелі **Настройка запроса**. При цьому дозволяється не тільки вибирати значення з випадаючого списку цієї кнопки, але і вводити свої, довільні значення.

Наприклад, можна для запиту встановити у полі **Назва_товару** сортування за убаванням цін і вказати у списку **Возврат** – 1. Це означатиме вимогу вивести один запис результуючого набору значень – тобто найдорожчий товар.

У режимі конструктора запитів можна змінювати імена полів запиту. Щоб перейменувати поле, необхідно встановити курсор у бланку запиту перед першою буквою імені поля та ввести нове ім'я та двокрапку. Нове ім'я поля буде відображатися як заголовок стовпця при прогляданні запиту в режимі таблиці. Крім того, у формах, звітах тощо, заснованих на цьому запиті, також буде використане нове ім'я поля. Ім'я поля базової таблиці при цьому не змінюється.


Для форматування даних якого-небудь поля потрібно викликати вікно властивостей цього поля (команда  на вкладці **Конструктор** панелі **Показать или скрыть**) і заповнити властивість **Формат поля** та **Число десятичных знаков**.

Виконання запитів

Запустити запит на виконання можна різними способами – як явними, так і неявними. До неявних належить використання запитів у формах, інших запитах, звітах, формуванні списків підстановки таблиць та ін.

Для явного запуску запиту у вікні БД на панелі **Все объекты MS**

Access двічі клацнути мишею на назві запиту або виділити його і у контекстному меню вибрати команду **Открыть**.

Для запуску запиту з режиму конструктора потрібно використовувати кнопку  **Режим таблиць** на вкладці **Конструктор** стрічки.

Розширені можливості запитів

Запити, що засновані на даних декількох таблиць

Якщо запит заснований на даних відразу декількох таблиць (або запитів), потрібно додати їх за допомогою вікна вибору таблиць та запитів і прослідкувати, щоб вони були з'єднані. Якщо зв'язки між таблицями в базі даних вже визначені, лінії з'єднання відображаються автоматично. Якщо таблиці, що беруть участь в запиті, не зв'язані між собою безпосередньо, необхідно додати одну або декілька додаткових таблиць, які зв'язують вибрані раніше таблиці.

Так, для побудови запиту, що відображає дані про квартири з їхньою адресою, необхідно включити у запит таблицю «Квартири». Проте ця таблиця безпосередньо не зв'язана з таблицями «Райони», «Вулиці» та «Матеріали» в базі даних. При виконанні такого запиту кожній квартирі будуть приписані дані кожного наявного запису таблиць-довідників. Щоб відобразити реальне положення справ, необхідно за допомогою вікна схеми даних з'ясувати, які таблиці зв'язують таблицю «Квартири» з таблицями «Райони», «Вулиці» та «Матеріали», і додати їх у запит. Для даного прикладу бази даних має бути додатково додана таблиця «Будівлі» (рис. 1.55).

Якщо в запит були додані не зв'язані між собою таблиці, які мають поля з однаковими іменами і узгодженими типами даних, конструктор запитів автоматично встановить зв'язки між цими полями таблиць. Найчастіше це приводить до некоректних результатів запиту. Наприклад, якщо в запит додати дві незв'язані таблиці, що містять поля *Назва* (допустимо, назви вулиць і назви районів міста), MS Access зв'яже таблиці по цих двох полях, що невірно.

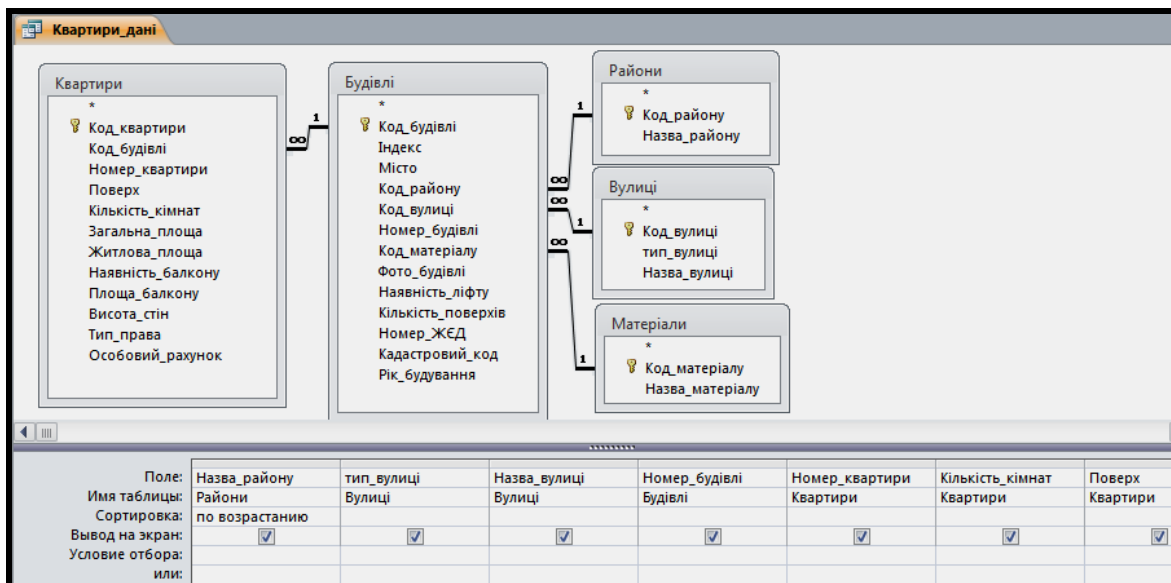


Рисунок 1.55 – Запит заснований на даних декількох таблицях

Тому необхідно перевіряти зв'язки між таблицями у вікні конструктора. Зайві зв'язки потрібно видаляти (виділивши лінію зв'язку й натиснувши клавішу **Delete**).

Параметричні запити

Запит в MS Access зберігається у файлі бази даних і може багато разів повторюватися. Усі створені до цих пір запити містили конкретні значення дат, назв та імен. Умови вибору можна змінити у режимі конструктора. Щоб не виконувати ці операції багато разів, можна створити запит із параметрами. При виконанні такого запиту з'являється діалогове вікно **Ведите значение параметра**, в якому користувач може ввести конкретне значення та отримати потрібний результат.

Допустимо, потрібно вивести дані по квартирах з певною кількістю кімнат, для цього вкажемо замість кількості кімнат параметр. Для цього замінимо умову відбору в полі **Кількість_кімнат** на довільну фразу-запрошення, розташовану у квадратних дужках, наприклад [**Вкажіть кількість кімнат**] (рис. 1.56).

Під час виконання запиту на екран буде виведено вікно, що містить вписану фразу-запрошення та рядок введення (рис. 1.57). Після введення будь-якої кількості кімнат ця кількість буде підставлена в умову вибору замість параметра, і запит буде виконаний для цього значення.

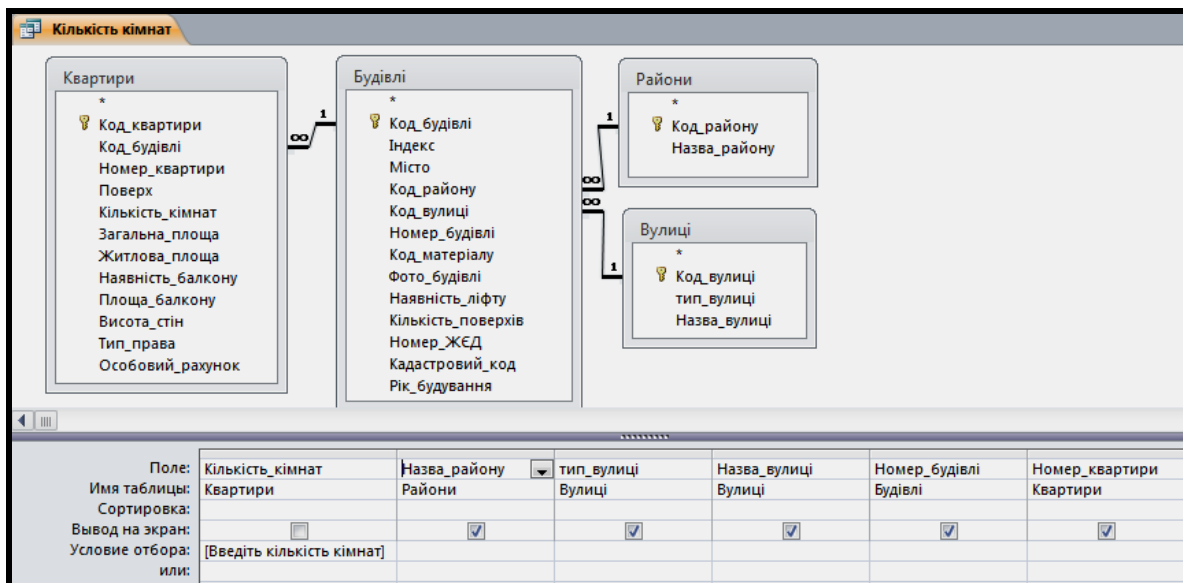


Рисунок 1.56 – Використання параметра в запиті

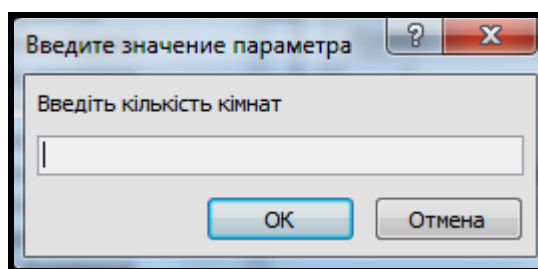


Рисунок 1.57 – Вікно введення значення параметра

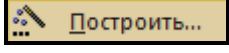
Параметри можуть використовуватися в умовах вибору спільно з логічними або арифметичними операторами, операторами порівняння. Можна також використовувати в одному виразі декілька параметрів.

Обчислення в запитах

Створення обчислюваних полів у запитах

Створення запитів цікаве не тільки тим, що можна відібрати потрібні записи з декількох зв'язаних таблиць і представити дані з них у вигляді однієї таблиці. Можна створювати стовпці в запиті, які є результатом обчислень над значеннями інших полів. Такі поля називаються обчислюваними. Це істотно розширює можливості запитів.

Наприклад, можна проводити розрахунок нежитлової площі квартир (комори, балкони та ін.), визначати житлові площі по окремим будинкам та цілим районам, об'єднувати значення декількох полів в одному полі, обчислювати вартість квартир та ін.

Нескладну формулу можна ввести у вільне поле бланка запиту безпосередньо в рядку **Поле**. Вирази можна вводити в середовищі MS Access не тільки вручну, але й за допомогою інструменту **Построитель выражений**. Для його виклику слугує команда контекстного меню **Построить**  або однойменна команда на панелі **Настройка запроса** вкладки **Конструктор** стрічки.

Верхня частина вікна **Построитель выражений** призначена для введення формули, нижня частина вікна полегшує процес введення формул. Розташовані в центрі вікна кнопки допомагають набирати знаки й оператори за допомогою миші.

Щоб швидко вставити у формулу назву поля іншої таблиці (запиту) або функцію, потрібно розкрити за допомогою кнопки «+» необхідну категорію в лівому стовпці, вибрати назву поля у середньому стовпці (для функцій – у крайньому правому) і здійснити подвійне клацання мишею (рис. 1.58).

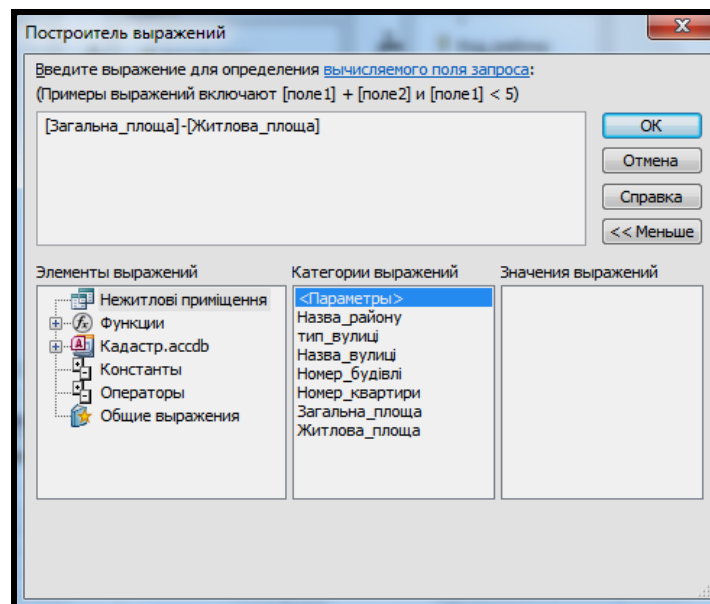


Рисунок 1.58 – Вікно «Построитель выражений» для введення виразів

Створення запитів у режимі SQL Access

SQL (*Structured Query Language* – «мова структурованих запитів») – універсальна комп’ютерна мова, яка використовується для створення, модифікації та управління даними в реляційних базах даних. SQL ґрунтується на обчисленні кортежів.



Мова SQL призначена для створення бази даних і таблиць, а також

для маніпулювання даними за допомогою операцій додавання, видалення та модифікації. Запити SQL створюються в режимі SQL і використовуються для роботи з таблицями бази даних, а також для створення підпорядкованих запитів безпосередньо у рядках **Поле** або **Умовия вибору** в бланку запиту конструктора.

У склад мови SQL входять два основні компоненти:

- мова DDL (Data Definition Language) – використовується для опису структур баз даних і управління доступом до них;
- мова DML (Data Manipulation Language) – використовується для вибірки й оновлення даних.

MS Access не надає користувачеві командного рядка для безпосереднього введення SQL-конструкцій. Для цього використовується вікно конструктора запитів. Алгоритм роботи наступний.

- Відкрийте вкладку **Создание** на стрічці з вкладками MS Access.
- Виберіть панель **Запросы** та команду  **Конструктор запитов.**
- З'явиться вікно **Добавление таблицы**. Закрийте його.
- Виберіть команду  **Режим SQL** панелі **Результаты**.
- Відкриється вікно конструктора в режимі введення кода мови SQL.
- Введіть у вікні текст запиту та збережіть запит.
- Відкрийте запит і перевірте результат його виконання.

Визначення даних за допомогою мови DDL

У склад мови DDL входять чотири оператори:

- **CREATE TABLE** – створення нової таблиці;
- **ALTER TABLE** – зміна структури таблиці;
- **DROP** – видалення таблиці або індексу;
- **CREATE INDEX** – створення індексу.

Вони дозволяють автоматизувати роботу зі створення, модифікації та видалення таблиць без використання конструктора.

Створення таблиць за допомогою мови DDL

Оператор **CREATE TABLE** визначає ім'я таблиці та множину поіменованих стовпців у вказаному порядку. Для кожного стовпця має бути визначений тип і розмір.

Синтаксис оператора:

```
CREATE TABLE <Ім'яТаблиці> (  
  [<Ім'яПоля1> <ТипДаних1> [<Розмір1>] [CONSTRAINT <Індекс1>]  
  [<Ім'яПоля2> <ТипДаних2> [<Розмір2>] [CONSTRAINT <Індекс2>])
```

Тут і далі прописними буквами виділені ключові слова. У кутових дужках курсивом позначені складові елементи оператора, наприклад:

<Розмір1>, <Розмір2> – розмір поля в знаках (тільки для полів з типом даних TEXT, BINARY і STRING);

<Індекс1>, <Індекс2> – індекс за одним полем, який визначає параметр **CONSTRAINT** (обмеження).

У квадратних дужках розміщені необов'язкові елементи.

Типи даних, які можуть бути створені за допомогою оператора **CREATE TABLE** в MS Access 2010, наведені в таблиці 1. 4.

Таблиця 1.4 – Типи даних мови SQL MS Access

Тип даних	Опис
BYTE	Байт
SMALLINT або SHORT	Ціле
INTEGER або LONG	Довге ціле
SINGLE	Число з плаваючою точкою одинарної точності
NUMERIC, FLOAT або DOUBLE	Число з плаваючою точкою подвійної точності
CHAR(N), TEXT(N) або STRING (N)	Текстовий довжиною N символів
BINARY(N)	Дійсне (N розрядів)
DATE або DATETIME	Дата/час
CURRENCY	Грошовий
LOGICAL або BIT	Логічний

Наступні приклади показують створення таблиць «*Будівлі*» і «*Вулиці*» бази даних «*Кадастр*».

Приклад. Запит на створення таблиці «*Вулиці*»

```
CREATE TABLE Вулиці (  
  Код_вулиці LONG CONSTRAINT Код_вулиці PRIMARY KEY,  
  Тип_вулиці TEXT(15),  
  Назва_вулиці TEXT(30) )
```

Другий рядок запиту призначений для створення первинного ключа

(*Код_вулиці* **PRIMARY KEY**). Наступні рядки призначені для створення полів *Тип_вулиці* та *Назва_вулиці*. Вони мають тип текстовий.

Приклад. Запит на створення таблиці «*Будівлі*»

```
CREATE TABLE Будівлі (  
Код_будівлі LONG CONSTRAINT Код_будівлі PRIMARY KEY,  
Індекс INTEGER,  
Місто TEXT(20),  
Код_вулиці LONG CONSTRAINT Код_вулиці REFERENCES Вулиці,  
Номер_будівлі TEXT(5),  
Наявність_ліфту LOGICAL,  
Номер_ЖЄД INTEGER,  
Рік_будування INTEGER,  
Кадастровий_код TEXT(15) )
```

Другий рядок запиту призначений для створення первинного ключа (*Код_будівлі* **PRIMARY KEY**). П'ятий рядок створює стовпець *Код_вулиці*, який призначений зовнішнім ключем (**CONSTRAINT** *Код_вулиці*) для зв'язку (**REFERENCES**) із таблицею «*Вулиці*».

Для створення індексу зовнішнього ключа *Код_клієнта* використаємо оператор **CREATE INDEX**.

Приклад. Запит на створення індексу поля *Код_вулиці* таблиці «*Будівлі*».

```
CREATE INDEX Будівлі  
ON Будівлі (Код_вулиці)
```

Для створення індексу по одному полю зовнішнього ключа **Код_вулиці** в круглих дужках після імені таблиці «*Будівлі*» вказано ім'я цього поля.

Після виконання запитів у базі даних «**Магазин**» створюються таблиці «*Вулиці*», «*Будівлі*» та зв'язки між ними (див. рис. 10.47).

Створення запитів за допомогою мови DML

Оператор **SELECT**

Оператор **SELECT** призначений для вибірки необхідних рядків з бази даних і розміщення їх у динамічному об'єкті наборі записів.

Синтаксис оператора:

```
SELECT <Список_стовпців>  
FROM <Список_таблиць>  
[WHERE <Умова_вибору>]  
[GROUP BY <Список_полів_групи>]  
[HAVING <Умова>]  
[ORDER BY <Список_атрибутів>];
```

Ключове слово **SELECT** (вибір) визначає імена полів таблиці, яку створює запит. Ключове слово **FROM** визначає імена таблиць, які є джерелом записів для створюваного запиту.

Приклад. Для вибірки усіх стовпців таблиці «**Вулиці**» бази даних «**Кадастр**» необхідно написати:

```
SELECT Тип_вулиці, Назва_вулиці  
FROM Вулиці;
```

Точку з комою в кінці останнього рядка MS Access додає сам.

Ключове слово **WHERE**, дозволяє задавати вираз умови, що набуває значення **True** або **False** для значень полів таблиць, до яких звертається оператор **SELECT**. У запит включаються тільки ті рядки, для яких умова, вказана в пропозиції **WHERE**, набуває значення істина.

Текст запиту, що виконує вибірку тих будівель, що розташовані на вулиці «**Пушкінська**» (код 1) із таблиці «**Будівлі**», такий:

```
SELECT Код_вулиці, Номер_будівлі, Рік_будування,  
Кадастровий_код  
FROM Будівлі  
WHERE Код_вулиці = 1  
ORDER BY Номер_будівлі;
```

Ключові слова **ORDER BY** призначені для впорядкування даних. Вони упорядковують виведення запиту згідно значенням в тій або іншій кількості вибраних стовпців. Можна використовувати **ORDER BY** одночасно з будь-якою кількістю стовпців. У всіх випадках стовпці, які упорядковуються, мають бути вказані у ключовому слові **SELECT**. За замовчуванням встановлено зростання. Текстові поля будуть відсортовані в алфавітному порядку. Стовпці типу **Дата/час** – у хронологічному порядку.

Для встановлення сортування за убунням використовується оператор **DESC**, наприклад, оператор **ORDER BY** Номер_будівлі **DESC** виконає сортування поля Номер_будівлі за убунням.

Ключові слова **GROUP BY** призначені для об'єднання записів з однаковими значеннями, що знаходяться у вказаному списку полів, в один запис.

Ключове слово **HAVING** (наявність) визначає згруповані записи, які повинні відображатися в операторі **SELECT** із реченням **GROUP BY**. Після того, як записи будуть згруповані, **HAVING** покаже ті з них, які відповідають його умовам.

Приклад. Для пошуку кількості будівель, що розташовані на вулиці, необхідно написати:

```
SELECT Код_вулиці, COUNT(Номер_будівлі)
AS [Count-Номер_будівлі]
FROM Будівлі
GROUP BY Код_вулиці
HAVING Код_вулиці=4;
```

У запиті використовується функція **Count()**. Вона підраховує кількість будівель на вулиці. Аргументом є строковий вираз. Операндом у виразі може бути ім'я таблиці або функція.

Вибірка даних із декількох таблиць

При роботі з нормалізованою базою даних неодмінно потрібно буде створювати запити, що використовують дані з декількох таблиць. Їх з'єднання обов'язкове. Для з'єднання таблиць використовуються ключові слова **INNER-JOIN-ON** (**JOIN** – операція з'єднання).

Приклад. Для вибору мешканців, що народилися за останній рік з таблиці «Мешканці» бази даних «Кадастр» необхідно написати:

```
SELECT Мешканці.Код_мешканця, Мешканці.Рік_народження
FROM Мешканці
WHERE (((Мешканці.Рік_народження) Between #1/1/2012# And
#31/12/2012#));
```

Тут для задання інтервалу дат використовуються оператори **Between** та оператор **And**.

Приклад. Для вибірки даних щодо розташування будівель по вулицям з таблиць «**Вулиці**» та «**Будівлі**» необхідно написати:

```
SELECT Вулиці.Назва_вулиці, Будівлі.Номер_будівлі  
FROM Вулиці INNER JOIN Будівлі  
ON Вулиці.Код_вулиці = Будівлі.Код_вулиці;
```

Результатом запиту буде таблиця, в якій перелічені вулиці міста та номери будинків, які на них розташовані.

Питання для самоконтролю

1. Наддайте характеристики об'єктам, що входять до СКБД MS Access.
2. Які існують можливості що до побудови запитів у MS Access?
3. Охарактеризуйте мову SQL.
4. У яких цілях використовуються макроси та модулі у базах даних?
5. Які дії можуть виконуватися над об'єктами бази даних? Охарактеризуйте їх.
6. Які види запитів існують? Охарактеризуйте їх і наведіть приклади.
7. Які підходу існують у MS Access до редагування запитів?
8. У яких цілях використовуються параметричні запити? Наведіть приклади.
9. Чи можна створювати обчислювальні поля у запитах? Як саме вони створюються?
10. Охарактеризуйте два основних компоненти мови SQL.
11. Які оператори входять до мови DDL та DML?

1.4 Ведення баз даних за допомогою форм

Інформація до баз даних зазвичай вводиться за допомогою форм. Форми є зручними при введенні інформації через свою схожість на звичайний бланк, у якому містяться підказки. Крім того, форми застосовують для перегляду та коригування даних. Працюючи з формою, користувач може додавати або вилучати записи в таблицях, змінювати окремі значення в полях, здійснювати обчислення. Форми є зручними для здійснення контролю введення даних, установлення окремих шаблонів для значень. Форма це об'єкт системи управління базами даних, призначений для зручності введення інформації в таблиці й відображення її на екрані.

У більшості інформаційних систем форма використовується для збирання та зберігання даних. Форми є альтернативним способом відображення табличних даних. СУБД Access дає змогу створювати форми, які можна використовувати для введення, управління, перегляду та друкування даних.

В СКБД Access форму для таблиці можна створити такими способами:

- у режимі конструктора, який призначений для самостійної розробки форми;
- за допомогою майстра, який створює форму на підставі обраних користувачем полів таблиць і стилю форми;
- у режимі автоформи, який призначений для автоматичного створення форми до обраної таблиці. Зазвичай, форму створюють за допомогою майстра або в режимі автоформи, а потім допрацьовують у режимі конструктора.

Розробка форм

Форми є лише другим способом відображення табличних даних. Першим кроком у створенні форм є створення таблиці або аналіз її структури. Часто через неправильно створену таблицю виникають проблеми в роботі з формами. Пам'ятайте, що властивості полів і властивості таблиць допомагають покращити якість даних. Вони є найкращим захистом від інформаційного «сміття».

Форми дають деякі переваги у відображенні даних порівняно з їх

відображенням у режимі таблиці, а саме:

- можуть відображати тільки один запис на екрані у «вертикальному» форматі (простому форматі);
- можуть відображати поля, які користувач не може редагувати, а також поля, які користувач може редагувати;
- можуть розроблятися так, щоб бути максимально схожими з робочими (паперовими) формами;
- дають змогу змінювати послідовність полів, щоб полегшити введення даних і зробити його зручним;
- можуть містити поля з кількох таблиць. У режимі таблиці на екрані відображаються дані лише з однієї таблиці;
- можуть мати спеціальні поля з функціями, наприклад поле зі списком і поле з виразом;
- можуть містити графіки й діаграми;
- дають змогу автоматизувати задачі введення, а також мати меню користувача.

Добре розроблена форма проста у використанні. Форми потрібно створювати так, щоб полегшити введення даних. Перевантажені форми, які містять багато полів на невеликій ділянці екрана, дратують користувача, що, зі свого боку, призводить до введення даних із помилками. Щоб вирішити цю проблему, продумайте можливість використання кількох різних форм або розбиття даних на кілька сторінок однієї форми.

Необхідно дотримуватися при розробці форм наступних вимог:

- форма має бути проста. Використовуйте шрифти й кольори, які легко читаються. Використовуйте графіки та інші елементи для вдосконалення форми, але не перевантажуйте форму великою кількістю об'єктів і полів;
- якщо форма друкуватиметься на чорно-білому принтері, доберіть кольори й розмітку так, щоб надрукований аркуш мав бездоганний вигляд;
- будьте акуратні, добираючи кольори для екранних форм. Деякі монітори з низькою роздільною здатністю в графічному режимі або з поганим відтворенням кольорів не можуть правильно відобразити вашу форму. Щоб уникнути цього, використовуйте установки для графічної карти й монітора найнижчого рівня;
- будьте послідовні, створюючи одразу кілька форм. Використовуйте один і той самий стиль компоновки для форми введення

даних і для форми, яка відображає замовлення.

Не забувайте що форма – це вікно, яке дозволяє створити зручне середовище для роботи з даними, що зберігаються у базі.

Під час роботи безпосередньо з таблицею відображаються записи тільки однієї таблиці. Якщо виконуються операції із взаємозв'язаними таблицями, бажано спостерігати за відповідними значеннями в інших таблицях.

Наприклад, під час роботи з таблицею «Мешканці» після введення коду квартири доцільно відображати дані про квартиру з таблиці «Квартири».

Форми дозволяють організувати зручну роботу з одним інформаційним об'єктом, розміщуючи дані в логічний і зручний спосіб.

Наприклад, під час введення даних у базу з паперових документів (комунальні платежі, квитанції, ордери тощо) екранна форма може мати вигляд, що повністю повторює паперовий документ. Це спрощує введення даних.

Деякі об'єкти баз даних відтворюються тільки в режимі форми (рисунок, мультиплікація, звуки й відеокліпи).

На формі можуть знаходитися кнопки, за допомогою яких організовують керування роботою застосування, а саме: відкривають нову форму, виконують запит, друкують звіт, закінчують роботу тощо.

Отже, форми забезпечують:

- зручне введення інформації в таблицю (рис. 1.59);

Номер квартири	Поверх	Кількість кімнат	Наявність балкону	Загальна площа	Прізвище квартирономача	Ім'я	По батькові
80	1	4	<input checked="" type="checkbox"/>	90,2	Лібман	Станіслав	Семенович
391	3	2	<input checked="" type="checkbox"/>	44,7	Сидорова	Олена	Миколаївна
392	3	4	<input checked="" type="checkbox"/>	90,2	Плоткін	Валентин	Іпполітович

Рисунок 1.59 – Форма для введення даних

– зручний перегляд даних і їх аналіз (рис. 1.60);

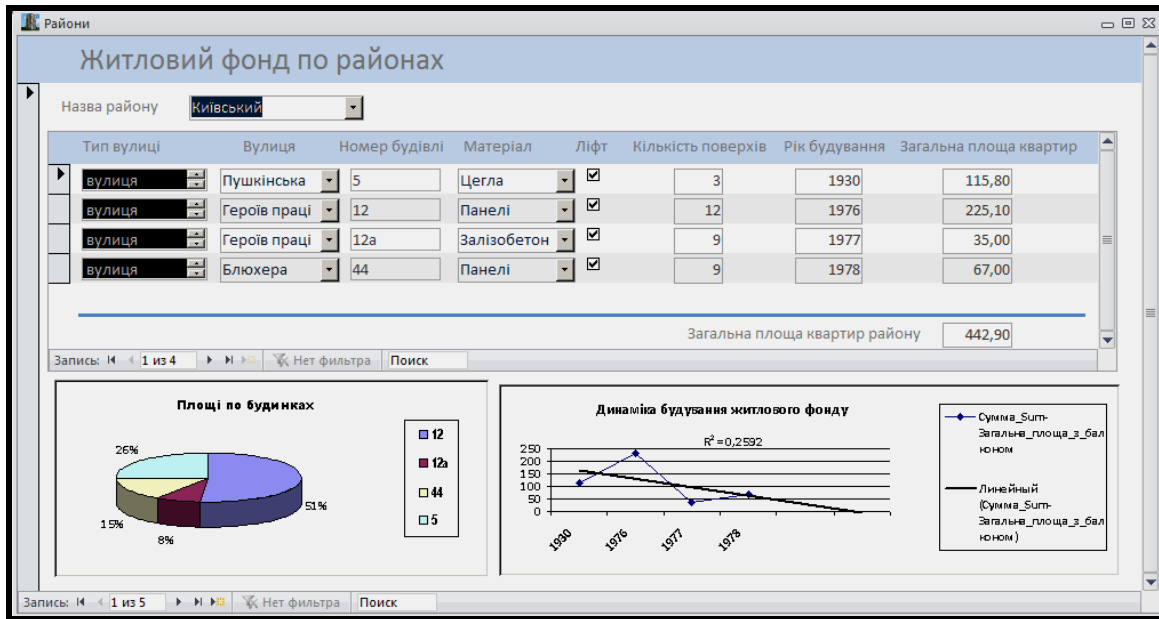


Рисунок 1.60 – Форма для перегляду та аналізу даних

– зручне керування роботою застосування (рис. 1.61);

The screenshot shows a data entry form titled "Будівлі" (Buildings). The form includes fields for "Вулиця" (Street) set to "Пушкінська", "Номер будівлі" (Building number) set to "5", "Кількість поверхів" (Number of floors) set to "3", "Наявність ліфту" (Lift availability) checked, and "Рік будівництва" (Year of construction) set to "1930". To the right of the form is a photograph of a multi-story brick building. The interface also includes a status bar at the bottom with navigation controls and a search button.

Рисунок 1.61 – Форма введення даних по будинках

– перегляд мультимедійних об'єктів (можливий тільки на формах – рисунки, анімація, звук тощо).

Питання для самоконтролю

1. У яких цілях використовуються форми у базах даних, їхнє призначення?
2. Які існують способи створення форм у MS Access?

3. Для яких цілей використовуються форми в MS Access?
4. Як змінити видимість об'єктів форми та їхні властивості?
5. Які переваги мають форми порівняно з таблицями? Наведіть приклади.
6. Яких вимог бажано дотримуватися при розробці форм?
7. Які об'єкти баз даних можуть відтворюються тільки в режимі форми?
8. Що забезпечують форми у СКБД Access?

РОЗДІЛ 2

БУДОВА ТА ФУНКЦІЇ ГЕОІНФОРМАЦІЙНИХ СИСТЕМ

У цьому розділі розглядаються геоінформаційні системи як редактори векторної та растрової графіки, які включають до свого складу просторові бази даних (зокрема під керуванням універсальних СКБД), які можна використовувати для виконання наукових і прикладних завдань інфраструктурного проектування, міського й регіонального планування, просторового аналізу даних; застосовувати у картографії, геології, землеустрої, екології, муніципальному управлінні, транспорті, економіці й багатьох інших сферах. Тобто ГІС розглядаються як технологічний комплекс, що інтегрує та поєднує багато інформаційних технологій.

Функціональні можливості геоінформаційних систем

Класична схема функцій ГІС, запропонована «патріархом» геоінформатики канадцем Роджером Томлінсом, наведена на рисунку 2.1. Відповідно до цих узагальнених функцій у ГІС виокремлюють підсистеми: збору, обробки, аналізу та ін.

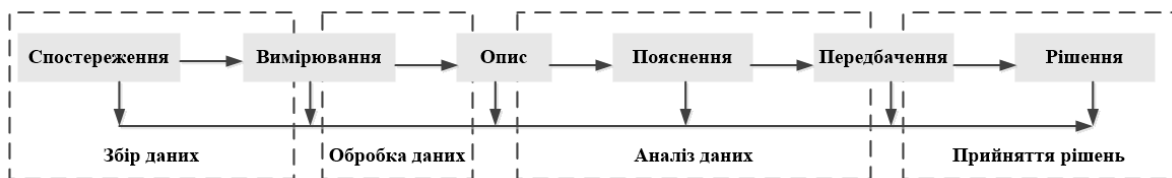


Рисунок 2.1 – Функції геоінформаційної системи

Ця схема відображає функції ГІС з точки зору загальних цілей і технологічного процесу обробки й аналізу просторових даних, проте для звичайного користувача робота в ГІС виглядає трохи інакше.

2.1 Класифікація ГІС. Моделювання навколишнього світу

На сьогодні геоінформаційними системами називають різні системи, за допомогою яких виконують найрізноманітніші завдання. У зв'язку з цим існують декілька класифікацій, які дозволяють повніше зрозуміти сутність ГІС.

Види ГІС за просторовим охопленням

Види ГІС за просторовим охопленням (рис. 2.2):

- глобальні (планетарні);
- субконтинентальні;
- міжнаціональні;
- національні (державні);
- регіональні (обласні, краєві, республіканські);
- субрегіональні (районні або інші регіони);
- локальні (місцеві, муніципальні, міські, селищні);
- ультралокальні (ГІС окремих промислових підприємств і будь-яких обмежених територій).

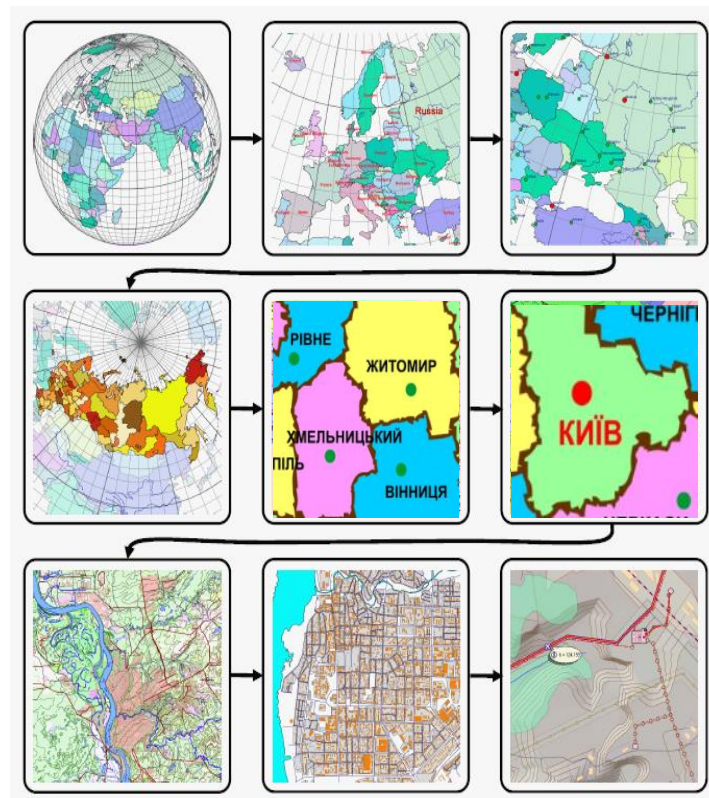


Рисунок 2.2 – Види ГІС за просторовим охопленням

Види ГІС за рівнем управління

Види ГІС за рівнем управління:

- регіональні;
- муніципальні;
- корпоративні.

Види ГІС за сферою діяльності

Види ГІС за сферою діяльності, в якій вона застосовується:

- управління (регіональні, муніципальні, корпоративні; планування

розвитку);

- землекористування (земельні кадастри, інвентаризація земельних ділянок, межування земель);
- управління нерухомістю (кадастр нерухомості);
- містобудування (генеральні плани розвитку, чергові плани, планування розвитку);
- архітектура (проєктування генеральних планів; ландшафтне проєктування);
- бізнес (оцінка інвестицій і планування бізнесу);
- інженерні мережі (управління й експлуатація міських, селищних і корпоративних інженерних мереж: електричних, водопровідних, водовідведення, теплових, газових, телефонних, кабельних, телевізійних, спеціальних та інших);
- інженерно-геодезичні дослідження (введення й обробка даних геодезичних досліджень; зрівнювання геодезичних мереж);
- інженерно-геологічні дослідження (введення й обробка геологічних даних);
- геологія (моделювання геологічних пластів; обробка даних буріння, сейсмозвідки);
- картографія (складання географічних і топографічних карт);
- проєктування та будівництво (проєктування автомобільних і залізничних доріг, генеральних планів, електричних і трубопровідних мереж);
- екстрені служби (диспетчеризація виїздів міліції, пожежної, швидкої допомоги);
- управління інженерним облаштуванням автомобільних доріг: світлофорами, знаками; диспетчеризація виїздів);
- навігація (навігація на місцевості та вибір маршрутів руху);
- транспорт (управління й експлуатація автомобільних і залізничних доріг; управління річковими, морськими та повітряними перевезеннями);
- логістика (планування і управління транспортними перевезеннями);
- оборона (планування військових операцій; тилове забезпечення);
- надзвичайні ситуації (аналіз та прогноз надзвичайних ситуацій; планування і здійснення заходів щодо ліквідації наслідків);
- екологія (оцінка та прогнозування дії на навколишнє

середовище);

- метеорологія (прогноз погоди; управління мережею метеорологічних пунктів і станцій);
- надрокористування (управління родовищами корисних копалин);
- природокористування (управління природними ресурсами);
- нафтогазова галузь (управління нафтогазовидобутком; управління промисловими майданчиками; управління магістральними трубопроводами);
- демографія та статистика (демографічний і статистичний аналіз);
- освіта (навчання, управління набором студентів);
- побутове використання (отримання довідки по об'єктах).

Види ГІС за функціональністю

Повнофункціональні (інструментальні) ГІС. Такі програми володіють максимальною функціональністю, властивою сучасним системам. Ці програми зазвичай забезпечують практично повний цикл роботи з просторовими даними від введення й обробки до аналізу та прийняття рішення. Повнофункціональні ГІС, як правило, дозволяють працювати зі всіма основними моделями даних геоінформатики: векторними, растровими, мережами й моделями поверхонь.

Такі системи, з одного боку, мають велику універсальність і потужність, що перекиває можливе використання. З іншого боку, ця універсальність часто заважає їх застосуванню для виконання конкретних прикладних завдань, оскільки універсальні інструменти не дуже зручні для кінцевих користувачів, а надлишок непотрібних у даному завданні функцій заважає початківцям.

Саме тому для реального застосування інструментальні ГІС володіють багатими засобами налаштування на конкретну область, починаючи з налаштування меню та панелей інструментів і закінчуючи можливістю написання додаткових функцій на вбудованій мові програмування (Microsoft Visual Basic for Applications) або підключення зовнішніх модулів (за технологією ActiveX).

ГІС для перегляду даних (ГІС-в'юери). Функціональність таких систем зазвичай обмежена переглядом і аналізом існуючих наборів просторових даних. Багато ГІС-в'юерів дозволяють взагалі тільки переглядати дані і одержувати коротку інформацію про вибрані на карті

об'єкти. В даний час багато фірм-виробників пропонують свої ГІС-в'юери безкоштовно, заробляючи гроші тільки на продажу більш функціональних ГІС.

ГІС для введення й обробки даних. До цієї категорії належать програми, призначені для підготовки початкових даних для ГІС за допомогою векторизації та обробки даних дистанційного зондування.

Спеціалізовані ГІС. До цієї категорії належать ГІС, призначені для застосування в конкретній галузі.

Види ГІС за моделлю даних, що використовується

Векторні ГІС. Такі системи працюють з топологічними та нетопологічними моделями даних, а також іноді з триангуляційними моделями поверхонь.

Растрові ГІС. Ці системи дозволяють працювати тільки з растровими моделями даних і іноді з регулярними моделями поверхонь.

Гібридні ГІС. Такі системи сполучають у собі можливості векторних і растрових ГІС.

Види ГІС за комп'ютерною платформою

Види ГІС за комп'ютерною платформою, на якій функціонує ГІС:

Настільні ГІС. До цієї категорії належать більшість відомих ГІС. Дані, що в них використовуються як правило, зберігаються у файлах, а комп'ютери як сервери просторових даних якщо й використовуються, то тільки як файл-сервери.

Клієнт-серверні ГІС. У цих системах просторові дані зберігаються повністю в базі даних, яка обслуговується особливою програмою – сервером просторових даних. Цей сервер зазвичай є високорівневою надбудовою над деякою промисловою системою управління базами даних (СКБД типу Microsoft SQL Server, Oracle, DB2, Sybase, Access). Багато сучасних клієнт-серверних ГІС можуть працювати не тільки з серверами, але також і безпосередньо з даними у файлах, тобто ці ГІС також можна віднести до категорії настільних.

Питання для самоконтролю

1. Охарактеризуйте функції та сфери використання ГІС.
2. У який спосіб виконується класифікація ГІС?
3. Наведіть види класифікацій ГІС за різними ознаками.

4. Які види ГІС за функціональною ознакою ви знаєте? Охарактеризуйте їх.

5. Наведіть сфери використання видів ГІС за моделлю даних, що використовується.

6. Наведіть сфери використання видів ГІС за комп'ютерною платформою.

7. Перелічіть види ГІС за сферою діяльності.

2.2 Введення даних. Цифрування вихідної інформації

Існує багато способів введення даних. Одні виглядають примітивними, на зразок накладання прозорої сітки на карту. Інші – більш сучасні. Наприклад, пристрої цифрового введення даних – дигітайзери і сканери.

Перед тим як використовувати структури даних, моделі й системи, необхідно перетворити нашу реальність у форму, що розуміється комп'ютером. Методи, за допомогою яких це буде зроблено, залежать деякою мірою від наявного устаткування та від конкретної системи. По-перше, підсистема введення спроектована для перенесення графічних і атрибутивних даних в комп'ютер. По-друге, вона повинна відповідати хоча б одному з двох фундаментальних методів представлення графічних об'єктів – растровому або векторному. По-третє, вона повинна мати зв'язок із системою зберігання й редагування, щоб гарантувати збереження та можливість вибірки того, що ми введемо, і давати можливість усувати помилки та вносити зміни за необхідністю.

Спочатку необхідно визначити, який тип ГІС, векторний або растровий, буде використовуватися, а також чи буде ГІС здатна перетворювати типи даних один в інший. Деякі програми працюють насамперед на растрових структурах даних, тоді як інші оперують переважно векторною інформацією.

Хоча перетворення між векторною та растровою формами – справа достатньо звичайна, є декілька речей, про які потрібно пам'ятати. Зазвичай при перетворенні векторів у растр результати виходять візуально задовільними, але методи растеризації можуть давати результати, які не задовільні для атрибутів, представляючих кожен клітинку. Це особливо критично уздовж меж областей, де є невизначеність з привласненням клітинкам растра атрибутів з однієї або іншої сторони межі. З іншого боку, перетворюючи растр у вектор, можна зберегти переважну більшість атрибутивних даних, але візуальні результати будуть часто відображати блоковий, ламаний вид осередків растра, з яких було проведено перетворення. Існують алгоритми згладжування цього ефекту, що використовують математичні методи сплайн-інтерполяції. Не вдаючись у подробиці, варто зазначити, що це просто графічний прийом, що згладжує зубчасті лінії.

Способи введення векторних даних

Як було вказано раніше, існує багато інструментів для введення в ГІС векторних даних. Розглянемо дигітайзерне оцифрування, як поширений «класичний» метод. Деякі програми вимагають введення точок у певній послідовності, тоді як інші цього не вимагають. Документація або сама програма повідомить про це. Крім того, програма укаже, які пронумеровані кнопки дигітайзера використовуються для введення конкретних типів об'єктів. Одні кнопки використовуються для вказівки положення точкових об'єктів, інші – для позначення кінців прямих відрізків, треті – для «склеювання» багатокутників (рис. 2.3).



Рисунок 2.3 – Вигляд дигітайзера

Конкретна процедура оцифрування залежить також від структури даних, яка використовується програмою. Одні вимагають вказівки положення вузлів, інші ні. Одні вимагають явного кодування топології під час оцифрування, інші використовують програмні методи побудови топології після того, як база даних заповнена. Правила різні для різних програм, і потрібно завчасно переглянути відповідну документацію для з'ясування цих стратегій. Ця робота може розглядатися як частина процесу підготовки карти, а не самого оцифрування.

Атрибутивні дані у векторних ГІС вводяться зазвичай з використанням клавіатури комп'ютера. Хоча цей спосіб введення даних простий, він вимагає такої самої уваги, як і введення графічних об'єктів. Причини дві. Перша: помилки оператора доволі часто здійснюється. Друга, і можливо, найпроблематичніша: атрибути повинні бути пов'язані з графічними об'єктами. Помилки в такому узгодженні – одні з найважчих

для виявлення помилок, оскільки їх не завжди можна помітити на погляд, і вони не виявляються до початку виконання аналізу. Необхідна перевірка атрибутів у процесі введення.

Способи введення растрових даних

Введення растрових даних виконується згідно з іншою стратегією, ніж введення векторних даних. Растрове введення іноді все ще робиться з використанням накладної сітки, коли атрибути вводяться послідовно, один за одним. Широка доступність сканерів швидко витісняє цей кропіткий метод введення, проте його застосування добре ілюструє різні методи, використовувані програмами оцифрування для введення клітинки растра. У минулому часто використовувався також метод оцифрування растра за допомогою дигітайзера, коли отриманий з дигітайзера контур об'єкта у вигляді векторів потім заповнювався пікселями вже самою програмою оцифрування.

Найперше при використанні цього способу необхідно вирішити, яку площу повинен займати кожний осередок растра. Це необхідно вирішити до початку оцифрування або накладення сітки, щоб повідомити програмі оцифрування розмір клітинки або дати оператору дані про розміри квадратів сітки. Крім того, нам необхідно обрати метод кодування, який міг би скоротити процес. При тому, що методи стиснення даних стають у нагоді для зменшення їхнього обсягу, використання цих методів при введенні може виявитися не менш важливим завдяки скороченню часу введення. Деякі растрові ГІС, не підтримуючі введення з дигітайзера або підтримуючі введення і з клавіатури, і з дигітайзера, мають команди, що дозволяють вводити дані у вигляді ланцюжків або блоків атрибутів. Вибравши метод введення, необхідно вирішити, як кожна клітинка растра буде представляти різні наявні теми. Крім розрішення растра, це найважливіше рішення, яке необхідно прийняти.

Для введення растрових даних найширше застосовуються сканери. Потрібно враховувати, проте, що введені з сканера тематичні дані не стають автоматично тематичними даними в растровій ГІС. Річ у тому, що однорідно закрашені на карті області після прочитування сканером неминуче одержують деякий розкид значень, унаслідок багатьох причин: неоднорідність нанесення фарби на карту, неоднорідність підсвічування в сканері, знос самого носія. Крім того, тематичні карти зазвичай

друкуються офсетним способом, який передбачає утворення всього багатства півтонів і колірних відтінків змішенням невеликої кількості фарб. При скануванні ці непомітні точки, перетворюються на цілком самостійні пікселі на місці зовнішньо однорідної за кольором області. Природно, такі карти не придатні для аналізу. Результат сканування здебільшого залежить від співвідношення розподілення сканера й поліграфічного растра. Саме складність розв'язання цієї проблеми приводить іноді до необхідності введення растрових даних за допомогою векторного оцифрування контурів об'єктів із подальшим перетворенням у растр.

Пристрої введення даних

Різноманітні типи пристроїв використовуються для введення інформації в комп'ютер. Більшість із них використовуються сьогодні для введення в ГІС. Першим підходом до спрощення введення даних було використання прозорого матеріалу з нанесеною сіткою, за допомогою якого дані, клітинка за клітинкою, вводилися вручну в комп'ютер. Зазвичай клітинкам растра привласнювалися числові значення, які вручну, один за одним, вносилися до комп'ютера. Це вимагало застосування правила, яке визначало, де всередині клітинки растра поміщався об'єкт, що оцифровувався. Такою середньою точкою може бути центр клітинки або будь-який з чотирьох її кутів. Знання точного положення точки просторової прив'язки кожного елемента принципово необхідне для векторних систем, важливо визначити це й для растрових даних, які будуть представлятися на комп'ютері клітинками растра. Це досить важливо – наприклад, вимірювання відстані на основі кількості клітинок растра: потрібно знати, від чого ви відраховуєте відстань – від сторін клітинки, або від центру. Усяка клітинка растра займає деяку площу. І чим більше ця площа (тобто чим нижче розподіл), тим більш значущім стає це питання.

Звичайно доводиться працювати з більш сучасним і складним устаткуванням. Для ручного введення просторових даних стандартом є дигітайзер. Він є досконалішим і набагато точнішим родичем найбільш широко використовуваного графічного маніпулятора – миші, яку користувач може вільно переміщати по практично будь-якій поверхні. Усередині миші знаходяться датчики, які реагують на пересування корпусу миші. Для збільшення точності подібного пристрою в дигітайзері

використовується електронна сітка на його столику. До столика приєднаний подібний миші пристрій, званий курсором, який переміщається по столу в різні положення на карті, яка до цього столу прикріплена. Курсор має перехрестя, нанесене на прозору пластинку, яке дозволяє оператору позиціонувати його точно на окремих елементах карти. Крім того, на курсорі розміщені кнопки, які (кількість їх залежить від рівня складності пристрою) дозволяють указувати початок і кінець лінії або межі області, явно визначати ліві та праві сторони області. Використання кнопок визначається переважно специфікою програми введення. Робоча поверхня дигітайзера може бути гнучкою або жорсткою, розмірами від книжкової сторінки до дуже великих форматів для розміщення великих карт (рис. 2.4).



Рисунок 2.4 – Різні види дигітайзерів

Деякі з великоформатних дигітайзерів мають підйимально-поворотну підставку, що дозволяє оператору встановлювати оптимальне для роботи положення. Розмір столу визначається частково розміром документів, що вводяться. Із розширенням використання комп'ютерів росте й автоматизація введення в них інформації. Для автоматизації введення карт використовуються такі пристрої, як автоматизовані дигітайзери та растрові сканери з програмами векторизації або без них.

Автоматизовані дигітайзери, або дигітайзери з відстежуванням ліній, мають пристрій, подібний головці оптичного прочитування програвача компакт-дисків. Він фіксується на вибраній користувачем лінії (як програвач фіксується на дорожці запису) і, самостійно слідуючи уздовж неї, передає координати точок лінії в комп'ютер. Ці пристрої вимагають постійної участі оператора, оскільки їх потрібно вручну встановлювати на кожну нову лінію для продовження процесу сканування. Крім того, вони

легко можуть помилятися на складних картах і картах з низькою контрастністю зображення. Наприклад, коли лінія розщеплюється на дві, цілком звичайна ситуація, коли сканер не знає, куди йти далі. Ця проблема може виявитися ще важчою, коли лінії зображаються пунктиром, який дигітайзер не може прослідити через розриви або через те, що колір світліше й має менший контраст, ніж початкова лінія.

Розповсюдженнішими є растрові сканери. Вони дозволяють вводити растрове зображення карти в комп'ютер без втручання людини. Для введення кольорових карт і знімків потрібно використовувати кольорові сканери, для панхроматичних знімків і топографічних карт достатньо чорно-білих сканерів, які дешевші. Якщо карта повинна зберігатися у векторній моделі даних, то після сканування растрове зображення повинне бути векторизовано. Векторизація в комп'ютері виконується подібно до того, як працює сканер із відстежуванням ліній, але тут вже можлива «розумніша» поведінка алгоритму, який самостійно знаходить і оцифровує лінії. Найвдаліше оцифровуються контрастні карти невисокої складності. Самі растрові сканери поділяються на ручні (а), роликові (з протяжкою листа) (б), планшетні (в) і барабанні (г) (рис. 2.5).

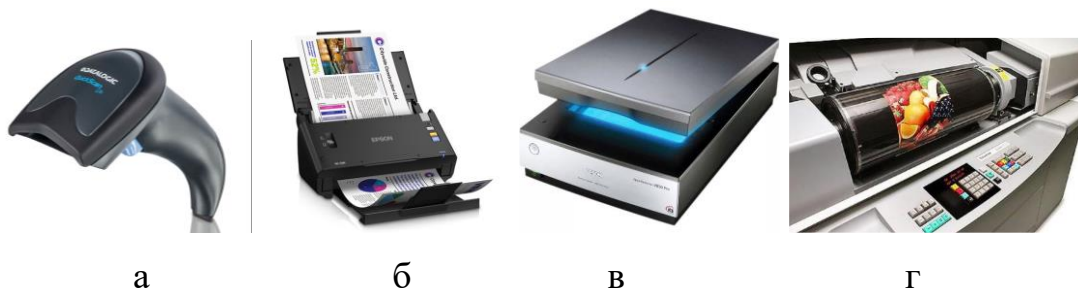


Рисунок 2.5 – Вигляд різних видів сканерів

Планшетні сканери становлять прозоре скло, на яке кладеться оригінал, і під яким переміщається лампа та пристрій оптичного прочитування. Ручний сканер є, по суті, оптичною головкою планшетного сканера, і користувачу доводиться самому рухати її по поверхні оригіналу. Очевидно, що точність сканування ручних сканерів – найнижча, тому пристрої цього вигляду практично не придатні для введення карт. Сканери з протяжкою листа діють подібно до факсного апарата, тобто в них рухається не головка прочитування, а сам оригінал, як у друкарській машинці. Ці пристрої надають точність меншу, ніж планшетні сканери, але зате дозволяють сканувати дуже довгі оригінали. У барабанних сканерах

оригінал закріплюється на круглому барабані, уздовж якого переміщається головка прочитування. Ці пристрої можуть забезпечити високу точність сканування дуже великих оригіналів.

Найдосконаліші (і найдорожчі) сканери утворюють категорію так званих фотограмметричних сканерів. Інший вид сканерів, барабанний, використовує докладніший растровий підхід, який насправді ближче до векторного режиму. Карта прикріплюється до барабана, який обертається, тоді як чутливий датчик приладу переміщається під прямим кутом до напрямку обертання. У такий спосіб, сканується вся карта, лінія за лінією. Записується кожне положення на карті, навіть якщо там немає графічних об'єктів. У результаті створюється докладне растрове зображення всієї карти. Барабанні сканери можуть давати як монохромне, так і кольорове зображення. В останньому випадку кожний з основних кольорів повинен скануватися окремо. Як монохромне, так і кольорове зображення повинно перетворюватися у векторну форму, якщо вона потрібна у використуванні ГІС. Обидві форми створюють дуже великі файли даних.

Спеціалізовані картографічні сканери великого формату дуже дорогі порівняно з дигітайзерами того самого формату. Крім того, векторизація введеного растра може зайняти майже стільки саме часу, скільки й ручне оцифрування, особливо якщо карта виявилася дуже складною. Поза сумнівом, у міру вдосконалення технології обсяг необхідного редагування буде зменшуватися.

На рисунку 2.6 зображена загальна схема роботи сканера. Світло, яке йде від джерела випромінювання, попадає на оригінал (карту) у певній точці. Відбившись від неї, воно попадає на оптичну систему сканера. Вона складається з декількох дзеркал та об'єктива. Оптична система фокусує світло на фотоприймальному елементі, роль якого – перетворення інтенсивності падаючого світла в електронний вигляд. Унаслідок перетворення світла створюється електричний сигнал, який містить інформацію про активність кольору у вихідній точці скануемого зображення. Цей сигнал не є оцифрованим, тому для приведення його у вигляд, зрозумілий для комп'ютера, необхідна конвертація. Цим займається аналогово-цифровий перетворювач. Саме цей сигнал через апаратний інтерфейс сканера йде на комп'ютер, де його отримує та аналізує програма для роботи зі сканером.

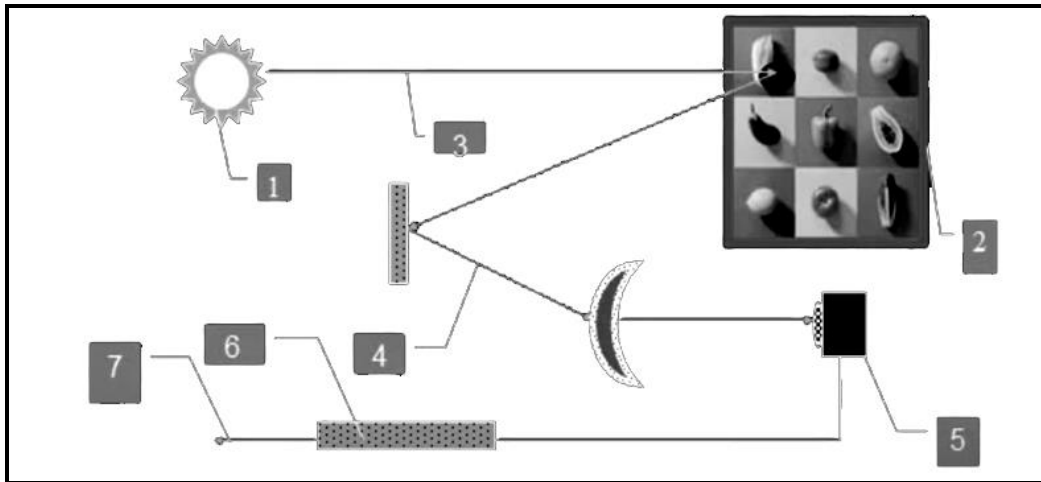


Рисунок 2.6 – Принципова схема роботи сканера:

1 – джерело світла; 2 – скануємий оригінал; 3 – промінь світла; 4 – оптична система сканера; 5 – фотоприймач; 6 – аналогово-цифровий перетворювач; 7 – апаратний інтерфейс.

Основні характеристики сканерів – оптичний розподіл, швидкість сканування та стабільність. Для офісних робіт зазвичай використовуються достатньо швидкі сканери з невисоким розподілом (300 крапок на дюйм). Можливості калібрування у таких сканерів відсутні. Ці пристрої можуть використовуватися для введення карт і знімків дистанційного зондування, коли вимоги точності дозволяють це.

Але не можна вірити заявам, що сканери звільнять людину від процесу введення даних. Пристрої автоматизованого введення та програми векторизації допомагають значно економити час тільки за умови чітких карт з високим контрастом. Зазвичай дорогі сканери використовуються фірмами, що спеціалізуються на послугах оцифрування.

Структура інформаційної системи

Існує безліч видів представлення інформаційних даних. Інформаційні системи є одним із таких видів. Наприклад, «Інформаційна система по природних ресурсах», «Екологічна інформаційна система», «Земельна інформаційна система», «Кадастрова інформаційна система». Хоча ці терміни описують застосування ГІС загалом, вони мало допомагають прояснити дійсну сутність системи. Можливо, тут виявиться корисним більш структурований підхід до класифікації ГІС у формі таксономічного дерева, поданого на рисунку 2.7.



Рисунок 2.7 – Класифікація інформаційних систем

Цей рисунок показує розділення між просторовими та непросторовими інформаційними системами (ІС). Автоматизовані системи управління транспортом (АСУТ) належать до негеографічних просторових ІС.

На гілці географічних інформаційних систем є ще одне розгалуження. ГІС можуть поділятися на земельні й неземельні, або інші інформаційні системи. Хоча таке розбиття дещо штучне, воно іноді корисно, оскільки відділяє застосування ГІС, сфокусовані на власне землю, від тих, де хоча й використовується геокодування, значуща інформація лише впливає на пов'язані із землею чинники або підпадає під вплив із їхнього боку. Прикладом таких систем є демографічні ІС, основною метою яких є населення, житлове будівництво й економічна активність, а не земля, на якій ці люди живуть. Ще одним загальним не пов'язаним із землекористуванням застосуванням ГІС є аналіз ринку, який може включати визначення місткості ринку в заданому радіусі від підприємства. Загалом, неземельні застосування ГІС зазвичай включають соціальні, економічні, транспортні й політичні види діяльності.

Пов'язані із землею види діяльності визначають межі для другого і, можливо найчастіше використовуваного типу ГІС – земельних інформаційних систем (ЗІС). Зазвичай такі системи засновані на володінні, управлінні й аналізі земельних ділянок, переважно з точки зору землеволодіння. Задачі, вирішувані ЗІС, можуть включати відчуження землі для заповідників, спостереження за живою природою, прогноз землетрусів і обвалів, усунення наслідків повеней, оцінку хімічного забруднення, управління лісами й зонами незаселеними дикими тваринами, наукові дослідження.


Як у сферах, пов'язаних із землею, так і в сферах, пов'язаних із


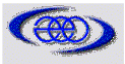
населенням, є багато можливостей застосування геоінформаційних технологій, що мають величезний потенціал, як для простих, так і для складних видів аналізу.


У світі існує величезна кількість різних інформаційних систем, зокрема й географічних. За масштабами застосування їх можна розподілити на глобальні та локальні, спрямовані на розв'язання загальних (багатофункціональні) і приватних, конкретних (однофункціональних) задач. Лідерами в сфері глобальних ГІС на сьогодні є продукти двох фірм – це система **ArcFM** американської фірми **ESRI** і **MapInfo** корпорації **INTERGRAPH**. Крім того, багато фірм, що займаються питаннями, пов'язаними із землеволодінням або землекористуванням створюють свої прикладні ГІС.

Результати опитувань, огляди публікацій та інформація з конфіденційних джерел, близьких до фірм дистриб'юторів дали картину ринку ГІС.

1. Перше місце і 36% ринку займає програмне забезпечення **ESRI Inc** (**ArcInfo, ArcView, ArcCAD**) 

2. Друге місце і 17% ринку належить **MapInfo** 

3. Третє і четверте місця – по 11% поділили між собою **Autodesk**  (**AutoCAD MAP 3D, World, MAPGuide**) і **GeoGraph** (Russia) 

4. П'яте місце – 4% у **Bentley** 

5. Шосте і сьоме місця – по 3% утримують **Ziegler** з **CADDY**  і **ERDAS Inc** 

Питання для самоконтролю

1. Які ви знаєте способи введення векторних даних? Наведіть

приклади.

2. Які ви знаєте способи введення растрових даних? Наведіть приклади.

3. За допомогою яких пристроїв можна вводити дані до бази даних ГІС?

4. Опишіть принцип роботи сканера.

5. Наведіть класифікацію інформаційних систем.

6. Які види ГІС за видом вирішуваних задачі ви знаєте?

7. Назовіть компанії, які спеціалізуються на розробці ГІС.

РОЗДІЛ 3

ПРИНЦИПИ ОРГАНІЗАЦІЇ ТА ЗБЕРІГАННЯ ДАНИХ У ГЕОІНФОРМАЦІЙНИХ СИСТЕМАХ

У цьому розділі розглядається, у який спосіб дані реального світу, які відображаються в ГІС, тобто просторові, часові та тематичні, найкраще зберігати у реляційних базах даних. Необхідно враховувати, що просторовий аспект пов'язаний із визначенням місця розташування об'єкта, часовий – зі змінами об'єкта або процесу з плином часу, тематичний аспект обумовлений виділенням одних ознак об'єкта та виключенням із розгляду інших. У зв'язку з цим у більшості сучасних технологій ГІС використовують два основних класи даних:

- для визначення місця використовують один клас даних – координати;
- для визначення параметрів часу й тематичної спрямованості, інший клас даних – атрибути.

Зберігання бази геоданих у реляційних базах даних

Геоінформаційна система (ГІС) ArcGIS будується по багаторівневій архітектурі додатку за рахунок реалізації розширеної логіки й поведінки, яка знаходиться над рівнем зберігання даних (управління здійснюється в межах різних систем керування базами даних (СКБД), файлів або файлів розширеної мови розмітки (XML) (рис. 3.1). Ця логіка додатку бази геоданих включає підтримку набору основних об'єктів у ГІС і їхньої поведінки, наприклад, класів об'єктів, наборів растрових даних, топологій, мереж.

Ця багаторівнева архітектура бази геоданих іноді називається **об'єктно-реляційною моделлю**.

Ядром бази геоданих є стандартна (тобто не якась спеціалізована) реляційна схема бази даних (набір стандартних для бази даних таблиць, типів полів, індексів і інших об'єктів). Проста фізична модель зберігання працює у взаємодії та контролюється набором об'єктів додатку більш високого порядку, що знаходяться на рівні додатку, яким може бути ArcGIS for Desktop, убудована логіка ядра обробки ArcGIS Engine або ArcGIS for Server.

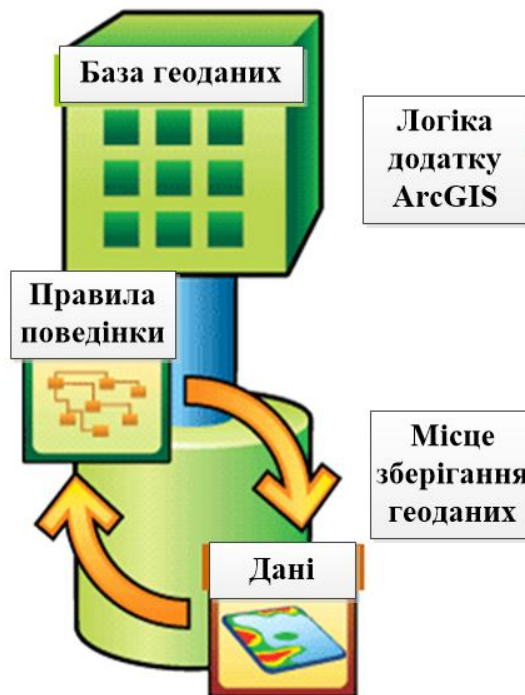


Рисунок 3.1 – Багаторівнева архітектура ГІС

У кожному з цих додатків містяться об'єкти бази геоданих, які визначають типову модель геоінформаційної системи, яка використовується всіма додатками й користувачами ArcGIS. Призначення об'єктів бази геоданих – надати клієнтам інформаційну модель ГІС високого рівня та планомірно реалізувати модель ГІС, що деталізується, за рахунок будь-якої іншої необхідної моделі зберігання. Наприклад, у стандартних таблицях системи керування базами даних, у файлових базах геоданих і у вигляді потоків розширеної мови розмітки (XML-потоків).

База геоданих повинна передбачати як схему даних, набори правил для кожного географічного набору даних, так і просте табличне зберігання просторових і атрибутивних даних.

Схема бази геоданих містить визначення, правила цілісності та поведінку для кожного набору географічних даних. Сюди належать властивості класів просторових об'єктів, топології, мережі, каталоги растрів, відносини, домени. Схема реалізується у вигляді набору мета-таблиць бази геоданих у СКБД, яка визначає цілісність і поведінку географічної інформації.

Просторові подання даних зазвичай зберігаються або як векторні об'єкти, або як растрові набори даних спільно з атрибутами в стандартному табличному форматі. Наприклад, у таблиці СКБД можна

зберегти клас просторових об'єктів, де в кожному рядку таблиці буде поданий просторовий об'єкт. У стовпці *shape* кожному запису співвідноситься геометрія або форма просторового об'єкта.

Однорідна сукупність звичайних просторових об'єктів, кожний з яких має однакове просторове подання, у вигляді точок, ліній або полігонів, і у вигляді загального набору атрибутивних полів має назву класу просторових об'єктів, який зберігається в одній таблиці.

Типи растрових даних і типи даних зображення також можуть бути організовані в таблиці та зберігатися в них. Растрові дані, як правило, набагато більше за розміром, і для їхнього зберігання потрібна допоміжна таблиця. Для зберігання у СКБД і доступу до даних, кожний растр розрізається на невеликі фрагменти, або блоки, які зберігаються в окремих рядках самостійної таблиці. Типи даних стовпців, у яких міститься векторна й растрова геометрія, залежать від конкретної бази даних.

Останнім часом у більшості СКБД додана підтримка розширень для просторового типу, і в базі геоданих можна зі зручністю користуватися цими типами для збереження даних просторової геометрії. Компанія **Esri** брала активну участь у розробці розширень мови структурованих запитів (**SQL**) для просторових типів даних. Вона була основним розробником специфікації **SQL/MM Part 3 Spatial** і специфікацій **SQL** для простих об'єктів **Open Geospatial Consortium, Inc. (OGC) Simple Features SQL**. **Esri** надала особливу увагу підтримці цих типів, а також розробці типів **Oracle Spatial**, **POSTGIS** (на мові **POSTGRESQL**) і просторових типів даних для **Microsoft SQL Server**, у межах роботи з реалізації баз геоданих із використанням стандартів СКБД.

База геоданих порівняно з СКБД

Багато баз геоданих використовують реляційні бази даних, такі як Oracle, IBM DB2 або Microsoft SQL Server для зберігання даних і управління географічною інформацією. Це дозволяє базі геоданих розростатися до величезних розмірів і підтримувати велику кількість користувачів і редакторів. Основним механізмом зберігання баз геоданих є використання таблиць. Таблиці зберігаються або як файли на диску, або всередині вмісту СКБД, як показано нижче (рис. 3.2).

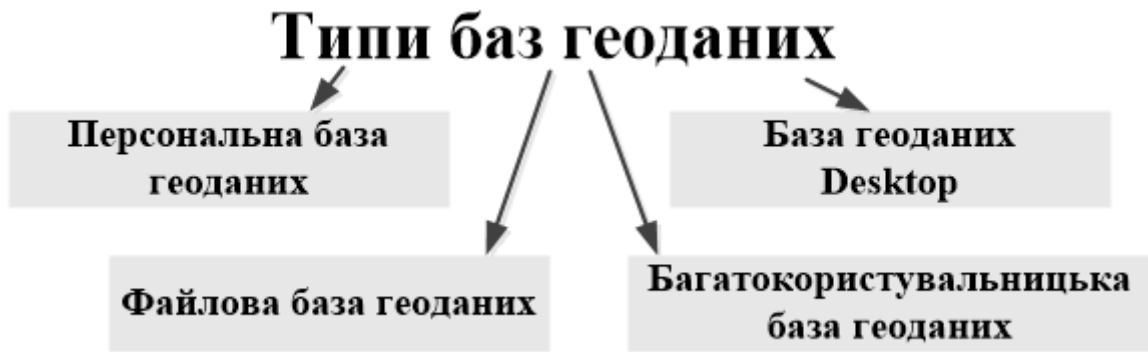


Рисунок 3.2 – Розподіл баз геоданих

Зберігання та управління просторовими даними

Таблиці

Сховище бази геоданих у СКБД містить два набори таблиць – таблиці наборів даних (таблиці, що задаються користувачем) і системні таблиці (рис. 3.3).

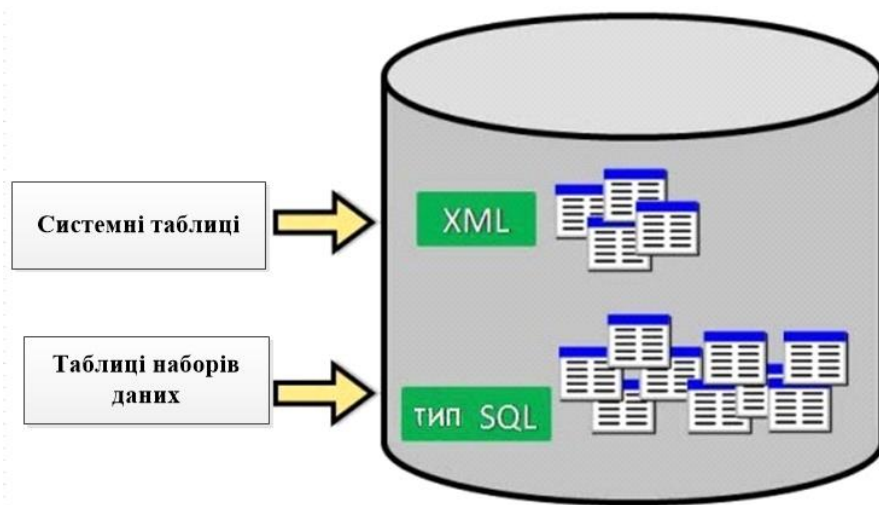


Рисунок 3.3 – Вигляд сховища бази геоданих

Таблиці наборів даних – кожний набір даних у базі геоданих зберігається в одній або декількох таблицях. Таблиці наборів даних для управління даними працюють із системними таблицями.

Системні таблиці – системні таблиці бази геоданих відстежують вміст кожної бази геоданих. По суті, вони описують схему бази геоданих, яка вказує всі визначення, правила та відносини наборів даних.

Ці системні таблиці містять і управляють усіма метаданими,

потрібними для реалізації властивостей бази геоданих, правил перевірки даних і поведінки.

Внутрішня структура цих таблиць була змінена, починаючи з випуску **ArcGIS 10**. Інформація, яка стосується схеми бази геоданих, яка до версії **ArcGIS 10** зберігалася в близько 35 системних таблицях бази геоданих, тепер була консолідована в чотирьох основних таблицях:

GDB Items містить список усіх елементів, які містяться в базі геоданих, таких як класи об'єктів, топології та домени;

GDB ItemTypes містить заздалегідь набутований список розпізнаваних типів елементів, таких як **Table** (таблиця);

GDB ItemRelationships містить схему відносин між елементами, такими як класи об'єктів, що містяться в наборі класів;

GDB ItemRelationshipTypes містить заздалегідь набутований список розпізнаваних типів відносин, наприклад **DatasetInFeatureDataset** (Дані в наборі класів)

Кожна СКБД має невеликі відмінності в наборі таблиць і стовпців, які використовуються для зберігання й управління базою геоданих. Тип СКБД, який буде використовуватися для зберігання бази геоданих, впливає на схему фізичного зберігання.

Додаткові файли бази геоданих також використовують у СКБД тригери, функції, процедури та типи даних, що задаються користувачами для реалізації функціональності й узгодженості підтримки.

Організація даних у геоінформаційних системах

В основі геоінформаційних систем лежить концепція пошарової організації просторових даних, коли однотипні дані на земній поверхні групуються в шари (іноді звані темами). Сукупність всіх шарів у ГІС утворює карту.

Розподіл об'єктів на шари проводиться так, щоб в одному шарі:

- об'єкти були однієї природи походження (дороги, річки, будівлі);
- об'єкти бажано мали однакову топологічну структуру та розмірність (їх можна описати точками, лініями або полігонами).

Водночас надмірно велика кількість шарів небажана. Наприклад, не потрібно створювати окремі шари для автомобілів різних марок, краще зробити один шар з автомобілями, при цьому кожному об'єкту, що описує автомобіль, потрібно зіставити код марки автомобіля.

Наприклад, на рисунку 3.4 наведений фрагмент міської забудови, на якому зображені житлові будинки, фабрики, склади, різні автомобілі, лісопаркова зона, річка, вулиця, міст, тунель. Для геоінформаційної системи створюються п'ять основних шарів.

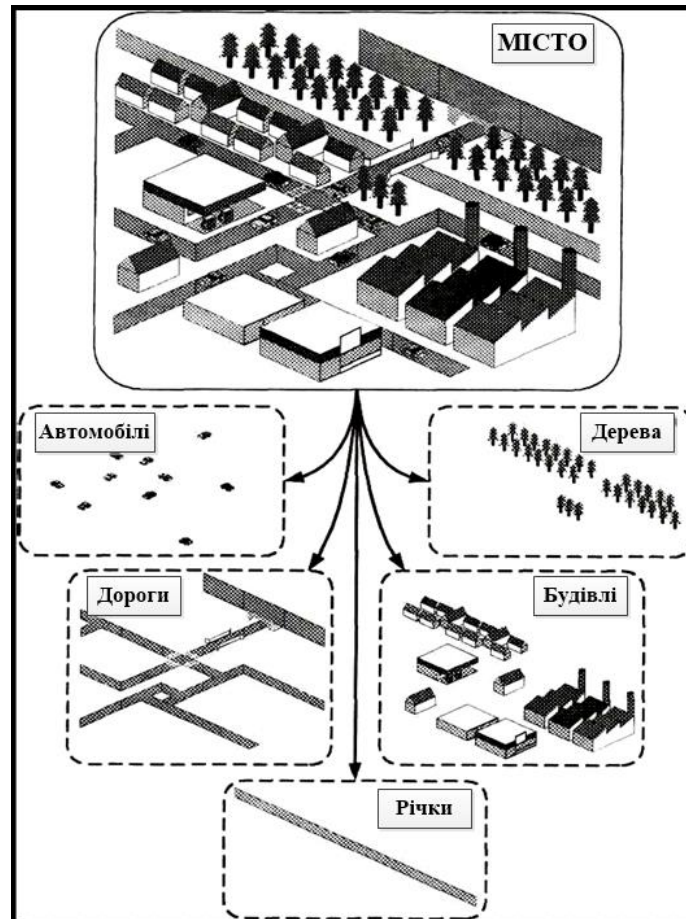


Рисунок 3.4 – Вигляд загальної схеми розподілу просторових даних ГІС на окремі шари

1. Шар автомобілів. Кожний автомобіль повинен бути зображений у вигляді крапки й додатково описуватися параметрами кольору та марки.

2. Шар автомобільних доріг. Залежно від задач, що вирішуються, автомобільні дороги можуть бути зображені у вигляді осьових ліній або у вигляді полігонів, які точно описують проїжджу частину. У деяких випадках має сенс мати два окремих шара для осьових ліній доріг і для проїжджих частин (полігонів).

3. Шар дерев. Дерева можна зобразити в ГІС у вигляді точкових об'єктів, для кожного з яких задані атрибути типу дерева: висота, діаметр, порода. Альтернативно крупні лісопаркові масиви можуть бути описані у

вигляді полігонів, що оздоблюють суцільні зони насаджень. Для кожного такого масиву дерев в атрибутах потрібно указати густину посадки, середню висоту дерев.

4. Шар будівель. Будівлі зображуються у вигляді полігонів, що описують контур будівлі на рівні землі. Для будівель в атрибутах потрібно указати тип будівлі (житлове, промислове, комерційне), адресу, висоту, кількість поверхів.

5. Шар річок. Річки на карті міста зазвичай зображуються у вигляді полігонів. На інших дрібніших картах (картах області або всієї країни) річки часто моделюють за допомогою ламаних ліній.

Розподіл даних на шари дозволяє працювати в ГІС тільки з тими даними, які необхідні для вирішення поставлених задач. У найпростішому випадку можна «виключити» ті шари, які нам не потрібні, і побачити на карті тільки ті, що залишилися.

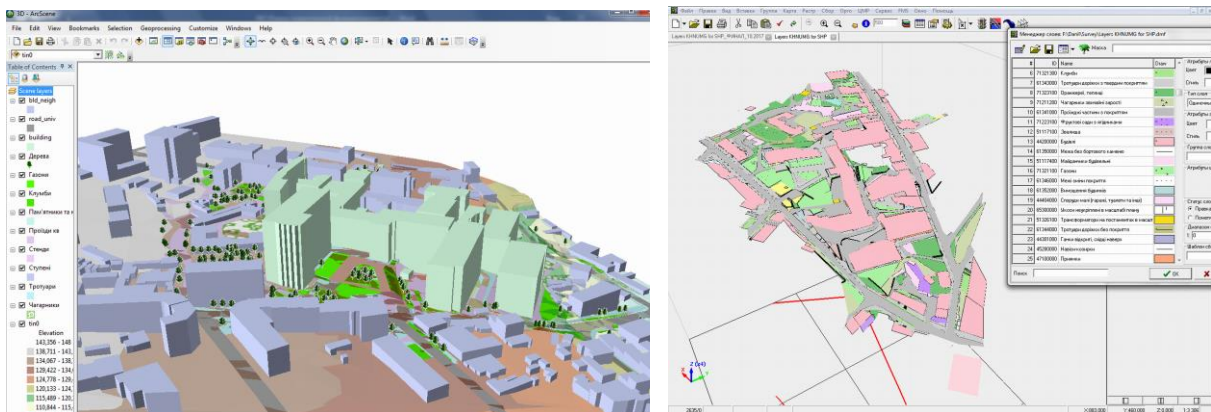


Рисунок 3.5 – Приклад розподілу даних за шарами

На рисунку 3.5 наведено приклад розподілу просторової інформації за шарами на прикладі моделювання території Національного університету міського господарства ім. О. М. Бекетова. На рисунку зліва надано тривимірну модель, виконану у програмному продукті **ArcGis**, а на рисунку справа – двовимірна модель, виконана у програмному продукті **Digitals**.

3.1 ГІС і просторовий аналіз

Вимірювальні операції

До вимірювальних операцій в сучасних ГІС належать функції, що дозволяють визначити різні геометричні характеристики об'єктів, явищ

або простору по карті (в проекції або по поверхні), такі як відстані, площі, азимуту.

Усі вимірювання в ГІС можна зазвичай виконувати двома способами:

- указуючи об'єкти, характеристики яких потрібно отримати;
- інтерактивно, задаючи точки геометричної фігури, що вимірюється.

Розглянемо основні вимірювальні операції, які можна виконувати в геоінформаційних системах.

Визначення координат точки на карті. Щоб скористатися цією функцією в більшості ГІС не вимагається перемикається в якийсь спеціальний режим роботи. При переміщенні мишки по карті в рядку стану зазвичай показуються координати указаной курсором точки. При цьому в рядку стану може відображатися й висотна відмітка цієї крапки, якщо в ГІС деяка поверхня вибрана як поточна (рис. 3.6).

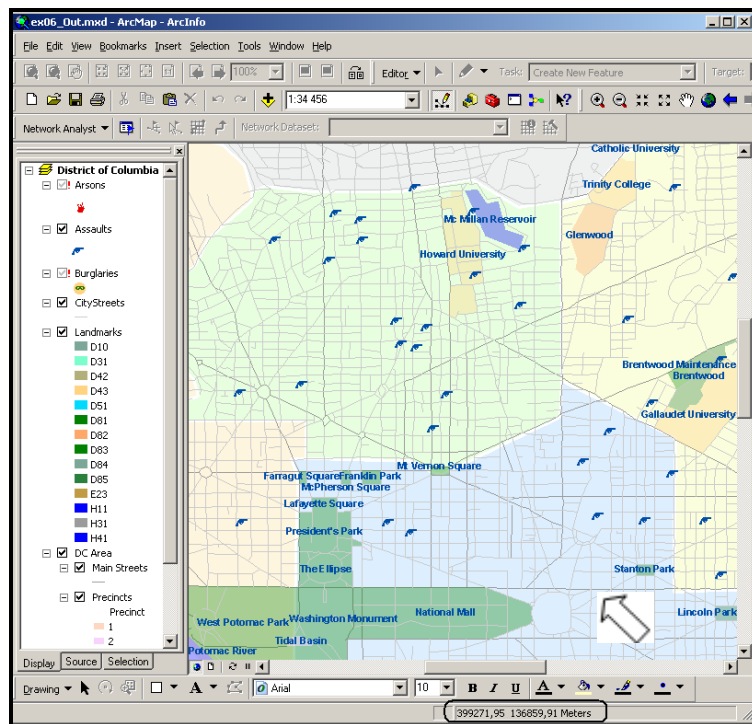


Рисунок 3.6 – Визначення координати точки на місцевості

Координати точок, що відображуються у ГІС, зазвичай показуються в початкових координатах об'єктів карти (наприклад у градусах), проте можливо й у системі координат проекції (наприклад у метрах в проекції Гаусса – Крюгера).

Вимірювання відстаней, азимутів і кутів. Користувач повинен указати на карті в ГІС послідовність точок ламаної, довжину якої потрібно обчислити. ГІС виводить в рядку стану загальну довжину ламаної, а також довжину останнього сегмента ламаної. Крім того, для останнього сегмента може виводитися азимут і кут повороту відносно попереднього сегмента (рис. 3.7).

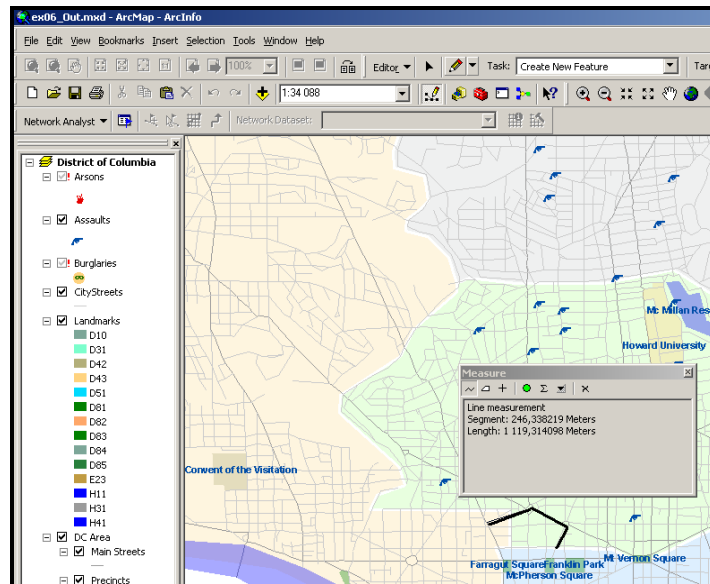


Рисунок 3.7 – Визначення довжини ламаної

Вимірювання площ і периметрів. Як і в попередньому способі, користувач повинен задати в ГІС вершини деякого багатокутника. При цьому в рядку стану ГІС буде виводитися площа та периметр (рис. 3.8).

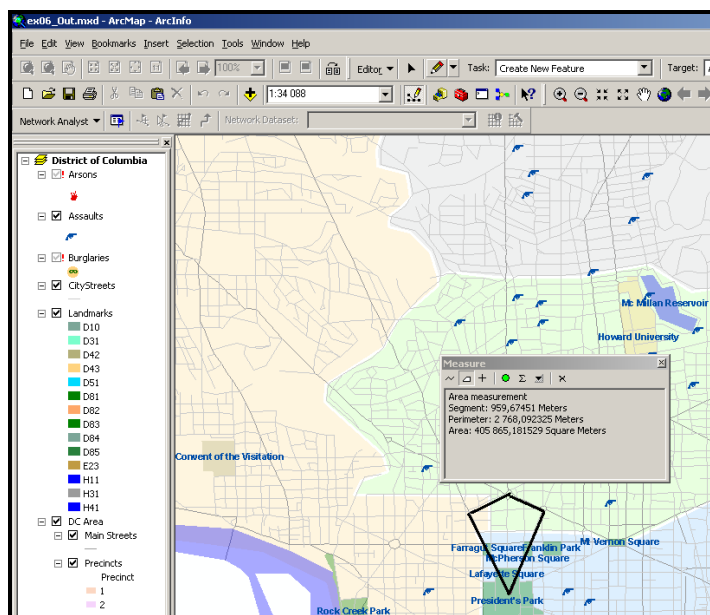


Рисунок 3.8 – Визначення площі та периметра

Вимірювання лінійних координат (пикету та зсуву) точки відносно траси. У цьому режимі користувач повинен указати деяку лінію, відносно якої проводиться визначення лінійних координат. Далі, переміщаючи курсор мишки по карті, в рядку статусу ГІС буде виводитися пикет і зсув указаної точки (рис. 3.9).

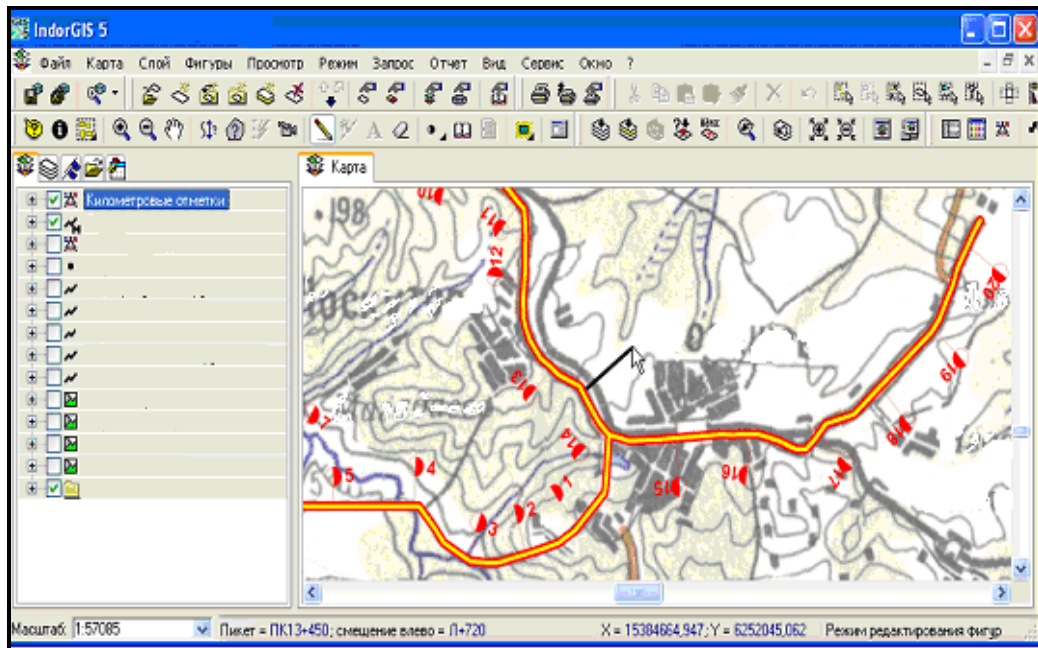


Рисунок 3.9 – Визначення лінійних координат

Деякі геоінформаційні системи надають можливість вимірювання відстаней і площ з урахуванням форми земної поверхні. При цьому обчислення можуть відбуватися або по поверхні референс-еліпсоїда, або по деякій цифровій моделі рельєфу, заданій у вигляді регулярної або нерегулярної мережі.

Мережевий аналіз

На основі транспортних мереж у ГІС можна розв'язувати різноманітні задачі, об'єднані під загальним терміном «мережевий аналіз».

Пошук найкоротшого за часом або відстанню маршруту між двома заданими вузлами транспортної мережі (рис. 3.10). Пошук повинен проводитися з урахуванням часу проходження по дугах, часу виконання поворотів у вузлах, а також з урахуванням допустимого напрямку руху в дугах.



Рисунок 3.10 – Пошук найкоротшого маршруту між заданими пунктами

Пошук найкоротшого обходу заданого набору пунктів (задача комівояжера). У цій задачі спочатку між кожною парою заданих пунктів знаходиться найкоротший маршрут пересування, а потім розв’язується математична задача комівояжера, перебираючи різні варіанти порядку обходу цих пунктів (рис. 3.11).

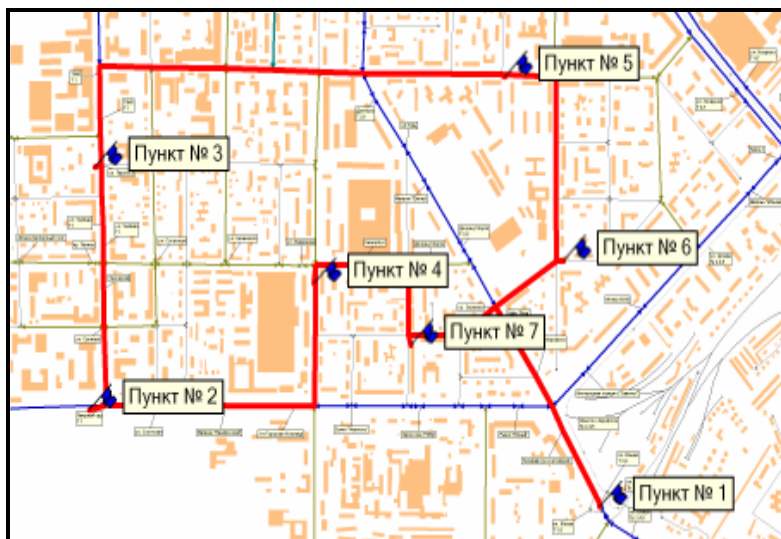


Рисунок 3.11 – Пошук найкоротшого обходу заданого набору пунктів

Пошук найближчих пунктів обслуговування. Передбачається, що на карті заданий точковий шар із деякими пунктами обслуговування, наприклад автозаправними станціями або магазинами. У цій задачі для заданої крапки необхідно знайти декілька найближчих пунктів обслуговування (рис. 3.12).

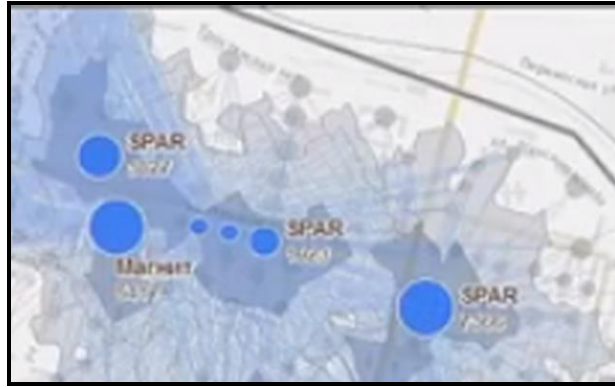


Рисунок 3.12 – Пошук найближчих пунктів обслуговування

Розрахунок зон обслуговування. Так само, як і для попередньої задачі передбачається, що на карті заданий точковий шар із деякими пунктами обслуговування. Необхідно розбити всю карту на непересічні частини, кожна з яких буде відповідати одному пункту обслуговування, який є найближчим для будь-якої крапки всередині отриманої частини (рис. 3.13).

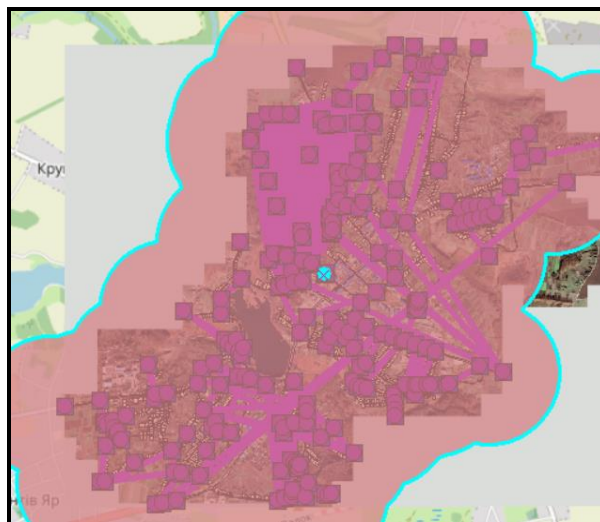


Рисунок 3.13 – Розрахунок зон обслуговування

Розрахунок транспортної доступності. У цій задачі передбачається, що на карті задані два точкових шари: один з аналізованими об'єктами (населеними пунктами на карті області), а інший – з деякими обслуговуючими об'єктами (наприклад районними центрами в областях). Для кожного населеного пункту необхідно визначити мінімальний час, за який з нього можна доїхати до райцентру, або указати, що проїзду немає.

На рисунку 3.14 показаний фрагмент району, в якому практично відсутні автомобільні дороги, але є річкове й авіаційне сполучення.

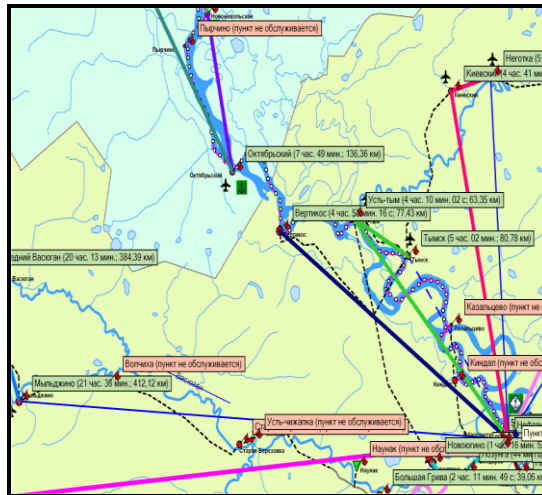


Рисунок 3.14 – Розрахунок транспортної доступності

Розрахунок міжрайонних транспортних зв'язків. Ця задача передбачає, що територія міста або регіону розбита на деякі транспортні райони (групи кварталів у місті або окремі поселення), між якими потрібно встановити величину тяжіння один до одного й рівень транспортної забезпеченості районів відповідно до деякого виду пересування, викликаного цим тяжінням.

Тяжіння між районами визначається на основі їхніх можливостей і привабливості, а також на основі транспортних зв'язків районів.

При розрахунку **переміщень працюючого населення** береться до уваги можливість районів (кількість працездатного населення), а привабливість району, зі свого боку, означає кількість робочих місць.

Наприклад, при розрахунку **культурно-побутових переміщень з будинку** можливостями районів є кількість жителів, а привабливістю – обсяг послуг, що надаються районом за одиницю часу (це охоплює в собі середнє число проданих за місяць квитків у всіх кінотеатрах, театрах, музеях, концертних залах, стадіонах; кількість і розмір різних магазинів, перукарень, аптек, пунктів ремонту).

При розрахунку **культурно-побутових переміщень не з будинку** (з роботи) привабливості беруться ті самі, що і з будинку, а можливостями вважається кількість робочих місць.

Транспортний зв'язок між кожною парою районів визначається по транспортній мережі як мінімальний час, необхідний для переміщення між районами. При цьому в місті розрахунок зазвичай ведеться по мережі маршрутного транспорту, а при розрахунку на рівні регіонів – по загальній

мережі автомобільних доріг.

Розрахунок транспортних зв'язків між районами ведеться на основі **моделі кореспонденцій**, що передбачає, що чим більше часу потрібно людині, щоб потрапити з одного району в інший, тим менш привабливим є цей район. Наприклад, тому більшість людей на роботу намагаються їздити недалеко від місця свого проживання і менша кількість – далеко.

У результаті розрахунку транспортних зв'язків виходить матриця, що показує загальну кількість людей, які в середньому переміщуються з одного району в інший відповідно до заданого виду пересування (кількість людей, які їздять на роботу в певний район). Крім того, для кожного району визначається, скільки в середньому витрачає один житель району на ці поїздки. Такі середні величини характеризують якість транспортної забезпеченості району.

На рисунку 3.15 наведені результати розрахунку транспортних зв'язків між окремими районами міста. На рисунку у вигляді товстих ліній показані об'єднані результати, згруповані по чотирьох адміністративних районах. Товщина лінії між адміністративними районами характеризує кількість жителів, які щодня ідуть на роботу з одного адміністративного району в інший. При цьому частина жителів працює в цьому самому районі, і ця кількість показується кругами відповідного радіуса в центрі районів. Додатково на рисунку в дужках після назви кожного адміністративного району вказано середній час, затрачуваний працюючими жителями цих районів на щоденні пересування на роботу.



Рисунок 3.15 – Розрахунок транспортних зв'язків між адміністративними районами

Аналогічно на рисунку 3.16 у дужках після назви кожного транспортного району наведено середній час, затрачуваний працюючими жителями цих районів на щоденні пересування на роботу.

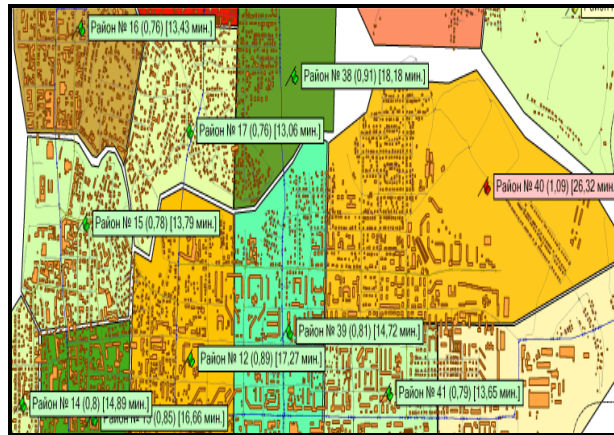


Рисунок 3.16 – Розрахунку транспортної забезпеченості районів

Методом кореспонденцій можна також розрахувати обсяг вантажних перевезень.

Розрахунок транспортних потоків. У цій задачі передбачається що відомі транспортні зв'язки між районами (попередньо обчислені або зміряні) і вимагається визначити, по яких конкретних дорогах будуть пересуватися автомобілі й пасажери. Для цього на основі відомого обсягу перевезень між транспортними районами визначаються декілька найкоротших маршрутів, а також кількості машин, яка поїде по конкретній ділянці дороги. Результат зазвичай показується на карті у вигляді картограм потоків (рис. 3.17).



Рисунок 3.17 – Картограма транспортних потоків

Аналіз поверхонь

До цього виду аналізу належить набір операцій, що дозволяють

виконувати аналіз поверхонь, використовуваних у ГІС як моделі рельєфу та представлених у вигляді регулярної або триангуляційної мережі.

Інтерполяція висот. Ця операція дозволяє обчислити значення висоти поверхні для будь-якої заданої планової точки.

Побудова профілів. Ця операція будує подовжній вертикальний розріз уздовж деякої заданої лінії (рис. 3.18). Додатково при відображенні профілю можна задати ступінь його розтягування за вертикаллю.

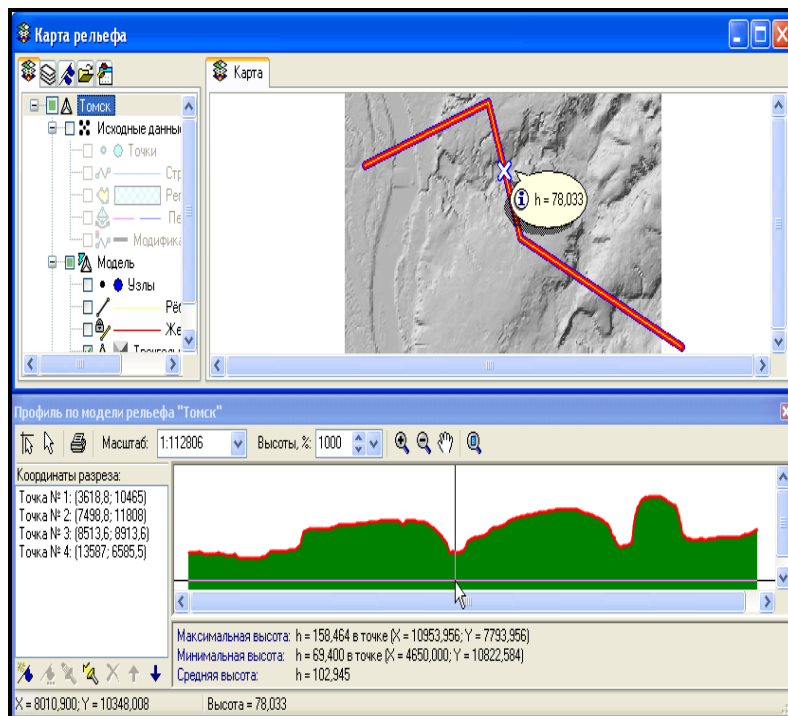


Рисунок 3.18 – Побудова подовжніх розрізів поверхні

Побудова горизонталей (ізоліній). Ця операція будує ізолінії – лінії однакового рівня (рис. 3.19, а). Результат зберігається у векторній моделі даних у вигляді поліліній. Ізолінії можна також трактувати як розріз поверхні горизонтальною площиною, розташованою на заданій висоті.

Побудова ізоконтурів. Ця операція будує ізоконтури – області між ізолініями суміжного рівня (рис. 3.19, б). Результат зберігається у векторній моделі даних у вигляді полігонів.

Побудова ізоклінів. Ця операція будує ізокліни – лінії однакового ухилу. Результат зберігається у векторній моделі даних у вигляді поліліній. Варіантом цієї операції є побудова ізоклін у вигляді контурів, які утворюються як області між суміжними ізоклінами, які побудовані з певним кроком. Ухил в даній точці поверхні вимірюється як відхилення

нормалі до поверхні в цій точці від вертикалі. Ухил може вимірюватися в градусах, проте найчастіше він вимірюється у відсотках або проміле.

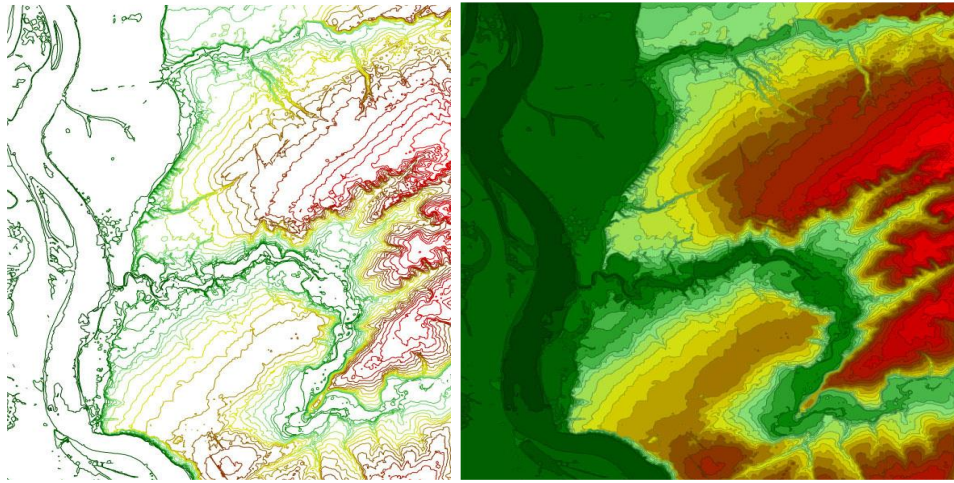


Рисунок 3.19 – Відображення моделі рельєфу ізолініями (а) і ізоконтурями (б)

Величина ухилу поверхні у відсотках указує, на скільки метрів зміниться висотна відмітка на поверхні при переміщенні уздовж поверхні на 100 м. Наприклад, відсотки використовуються для позначення ухилу на дорожніх знаках. Так, значення ухилу 5 % означає перепад висот в 5 м на 100 м дороги.

Розрахунок експозицій схилів. Ця операція обчислює нормалі до кожного елемента поверхні і визначає, в яку сторону світла повернута нормаль. Результат відображується на карті різними кольорами (рис. 3.20, а).

Відображення експозицій схилів. Даний спосіб дозволяє візуально визначити, в яку сторону світу нахилена поверхня. Звичайно всі сторони світа ділять на 8 частин секторами по 45° (північ, південь, захід, схід, північний захід, північний схід, південний захід і південний схід), а потім для кожного осередку моделі поверхні визначають напрям ухилу поверхні і вибирають один з восьми кольорів відображення осередку (рис. 3.20, б).

Результати обчислень відображують на карті у вигляді окремих областей, що показують, де вимагається виконати зрізання рельєфу, а де вимагається зробити насип (рис. 3.21).

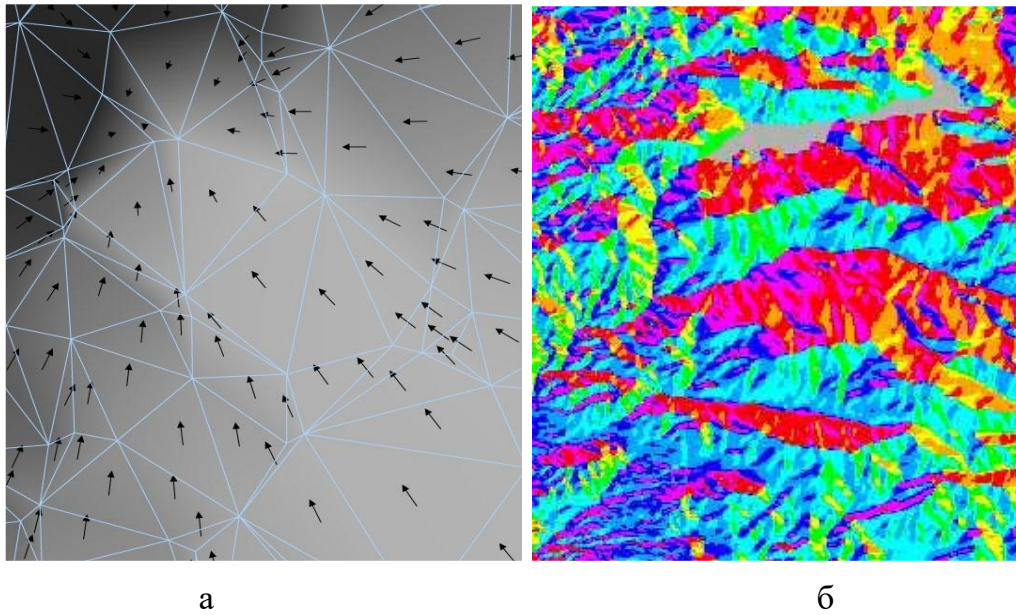


Рисунок 3.20 – Відображення моделі рельєфу векторами ухилів (а) та експозиціями ухилів (б)

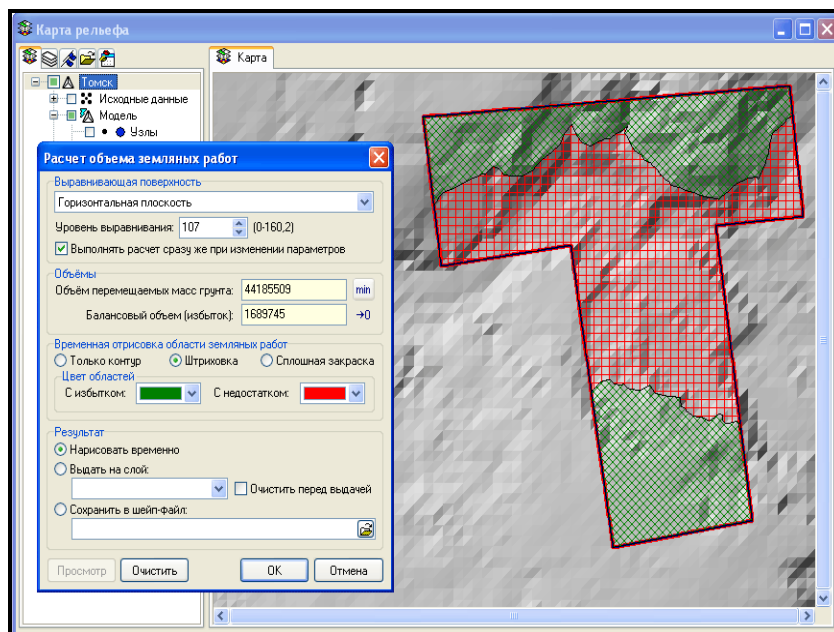
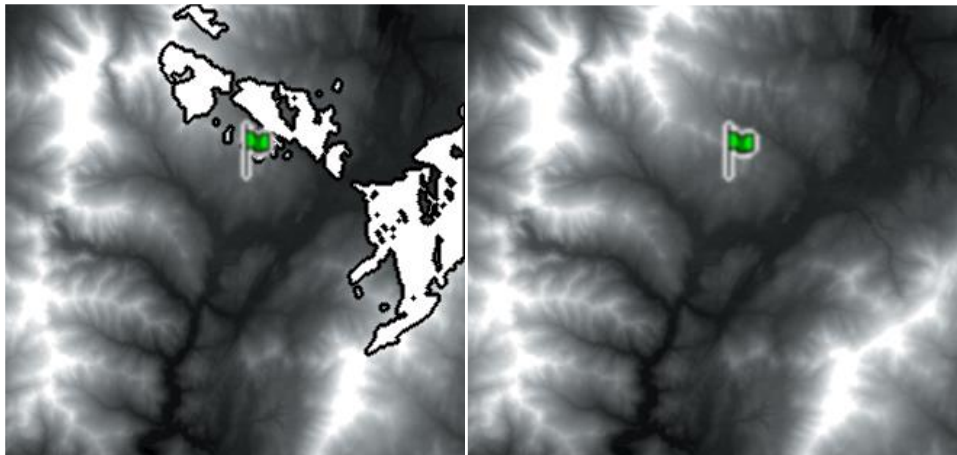


Рисунок 3.21 – Розрахунок обсягів земляних робіт

Аналіз видимості. Ця операція визначає, які області на карті видно із заданої в тривимірному просторі точки. Результат представляється на карті у вигляді областей, з яких буде видно указану точку. На рисунку 3.22 показаний приклад розрахунку зон видимості.



а

б

Рисунок 3.22 – Розрахунок зон видимості: а – зона видимості показана білою областю з чорним контуром, б – вихідна модель і точка спостереження на поверхні

У деяких випадках ця задача розв’язується тільки уздовж деякої одній окремо взятої прямої. Наприклад, так розраховується видимість автомобільної дороги з автомобіля. Як точку зору береться крапка над дорогою на висоті 1,20 м. Результатом розрахунку є відстань, починаючи з якої видимість дороги пропадає.

Побудова мережі тальвегів і водорозділів. Ця операція призначена для аналізу форми рельєфу місцевості (рис. 3.23) і виділення таких структурних елементів, як лінії тальвегів і вододілів, а також особливих елементів (вершин, сідловин, хребтів, ярів). Ця операція аналізує кожний осередок моделі поверхні та визначає напрям, куди буде текти вода з цього осередку. Потім визначаються тальвеги – лінії, що з’єднують найнижчі точки дна річкової долини, яру, промоїни. Оконтуривши всі осередки, які створюють водозбір для кожної окремо взятої річки та водоймища, буде одержано лінії водорозділів. Ця операція використовується зазвичай в гідрології при аналізі рельєфу на маломасштабних картах.

Розрахунок об’ємів земляних робіт. Ця операція передбачає, що є модель існуючого рельєфу та задано, яку форму повинен прийняти цей рельєф. Вимагається визначити обсяги ґрунту, які потрібно перемістити для отримання проєктованого рельєфу.

У найпростішій постановці цієї задачі вимагається вирівняти дно котловану, заданого у вигляді багатокутника, до заданої висотної відмітки. В іншій (складнішій) постановці чітко створюється векторна модель

рельєфу. При цьому вимагається обчислити різницю між існуючою та проєктованою поверхнями. Результатом обчислень є деяка нова поверхня (яку вже не можна інтерпретувати як рельєф).

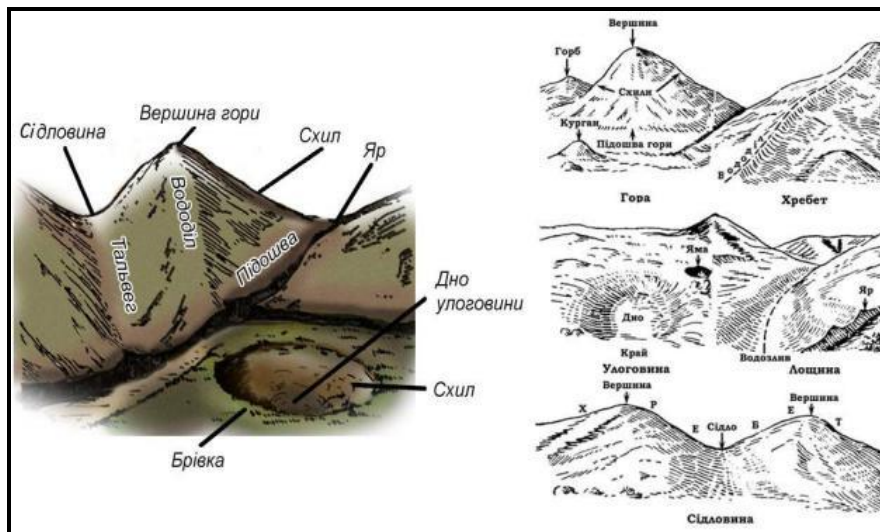


Рисунок 3.23 – Рельєф місцевості й різні його елементи

Аналіз водостоків. Ця операція призначена для аналізу рельєфу місцевості у великому масштабі та визначення місць скупчення води. Для цього спочатку на рельєфі знаходяться всі точки локального мінімуму, а потім ці місця поступово наповнюються водою доти, доки калюжі не переповняться й не стануть переливатися через край (рис. 3.24).

Це використовується, наприклад, під час проєктування автомобільних доріг і генеральних планів для виявлення потенційних місць утворення калюж. Для виключення можливості утворення калюж потрібно перепроєктувати рельєф або встановити в центрі цих калюж зливні колодязі.

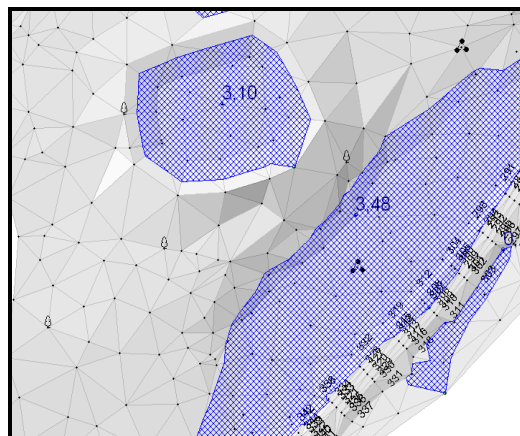


Рисунок 3.24 – Рельєф місцевості з визначеними місцями скупчення води

Дистанційне зондування землі

Ефективну роботу сучасних ГІС важко уявити без методів супутникового дослідження територій нашої планети. Дистанційне зондування з супутника широко застосовується в геоінформаційних технологіях як у зв'язку з швидким розвитком і вдосконаленням космічної техніки, так і з згортанням авіаційних і наземних методів моніторингу.

Дистанційне зондування (ДЗ) – науковий напрям, заснований на зборі інформації про поверхню Землі без фактичного контакту з нею. Процес отримання даних про поверхню включає зондування й запис інформації про відображену або випускаєму об'єктами енергію з метою подальшої обробки, аналізу та практичного використання.

Виділяють такі сфери застосування дистанційного зондування:

- отримання інформації про стан навколишнього середовища та землекористування;
- оцінка урожаю сільськогосподарських угідь;
- вивчення флори й фауни;
- оцінка наслідків стихійних лих (землетруси, повені, пожежі, епідемії, виверження вулканів);
- оцінка збитку при забрудненні суші та водоймищ;
- океанологія.

Засоби ДЗ дозволяють одержувати відомості про стан атмосфери не тільки в локальному, але і в глобальному масштабі. Дані зондування надходять у вигляді зображень, як правило, в цифровій формі. Подальша обробка здійснюється за допомогою комп'ютера. Тому проблематика ДЗ тісно пов'язана із задачами цифрової обробки зображень.

Для спостереження нашої планети з космосу використовують дистанційні методи, за допомогою яких дослідник має можливість на відстані одержувати інформацію про об'єкт, що вивчається (рис. 3.25). Дистанційні методи зондування, як правило, є непрямыми, тобто за їхньою допомогою виміряють не ті параметри, що цікавлять спостерігача, а деякі пов'язані з ними величини.

Методи вивчення Землі з космосу не випадково належать до високотехнологічних. Це зв'язано не тільки з використанням ракетної техніки, складних оптико-електронних приладів, комп'ютерів, швидкісних інформаційних мереж, але і з новим підходом до отримання та

інтерпретації результатів вимірювань.

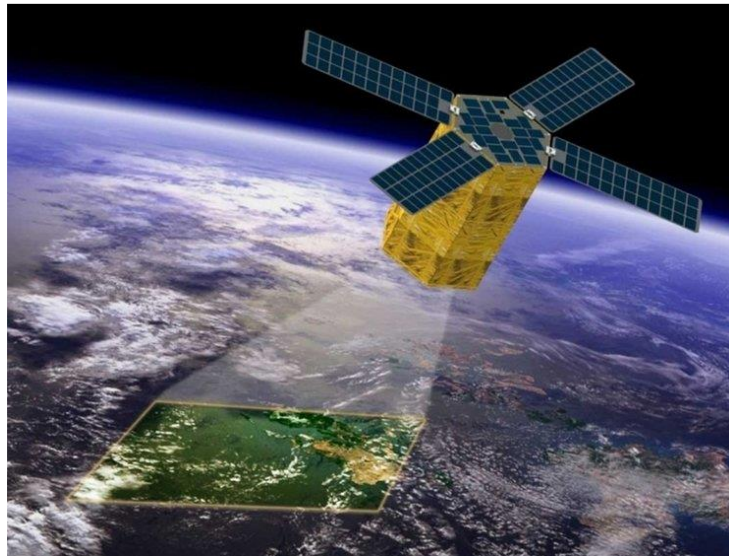


Рисунок 3.25 – Виконання дистанційного зондування Землі супутником

Дослідження супутників проводяться на невеликій площі, але вони дають можливість узагальнювати дані на величезні простори й навіть на всю земну кулю. Методи супутників, як правило, дозволяють одержувати результат за порівняно короткий інтервал часу.

Серед особливостей дистанційних методів виокремлюють вплив середовища (атмосфери), через яку проходить сигнал із супутника. Інформація про Землю та її об'єкти надходить з супутників у цифровому вигляді. Наземна цифрова обробка зображень проводиться за допомогою комп'ютерів. Сучасні методи супутників дозволяють не тільки одержувати зображення Землі. Використовуючи чутливі прилади, вдається виміряти концентрацію атмосферних газів, зокрема спричинених парниковим ефектом.

Дистанційні методи діляться на пасивні (б) і активні (а) (рис. 3.26). При використанні активних методів супутник посилає на Землю сигнал власного джерела енергії (лазера, передавача радіолокації) та реєструє його віддзеркалення. Пасивні методи передбачають реєстрацію відображеної від поверхні об'єктів сонячної енергії або теплового випромінювання Землі.

При дистанційному зондуванні Землі з космосу використовують оптичний діапазон електромагнітних хвиль і мікрохвильову ділянку радіодіапазону.

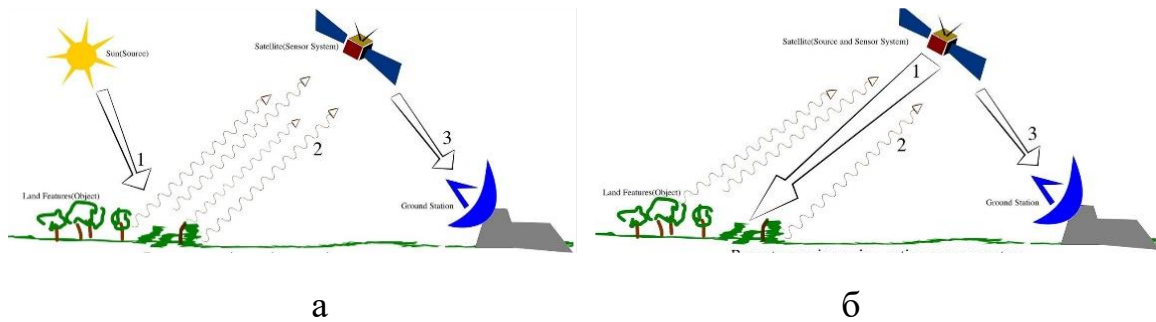


Рисунок 3.26 – Принцип пасивних (а) і активних (б) методів дистанційного зондування

При пасивних методах зондування в оптичному діапазоні джерелами електромагнітної енергії є розігріті до достатньо високої температури тверді, рідкі, газоподібні тіла.

Принципи дистанційного зондування

Суть методу полягає в інтерпретації результатів вимірювання електромагнітного випромінювання, яке відображається або випромінюється об'єктом і реєструється в деякій видаленій від нього точці простору.

За допомогою дистанційного зондування вивчають фізичні та хімічні властивості об'єктів. Прикладами природних форм ДЗ є зір, нюх і слух людини. До методів дистанційного зондування належить і фотографічне знімання, істотним обмеженням якого є те, що емульсивний шар фотоплівки чутливий тільки до випромінювання у видимій або близькій до неї частині електромагнітного спектра.

Ці методи ДЗ засновані на використанні сенсорів, які розміщуються на космічних апаратах і реєструють електромагнітне випромінювання у форматах, істотно більше пристосованих для цифрової обробки, і в істотно ширшому діапазоні електромагнітного спектра (рис. 3.27). У більшості методів ДЗ використовують інфрачервоний діапазон відображеного випромінювання, тепловий інфрачервоний і радіодіапазон електромагнітного спектра.

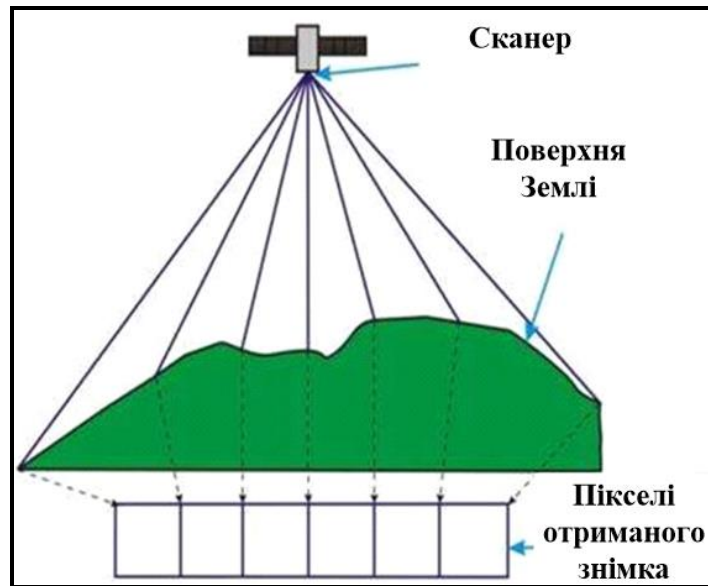


Рисунок 3.27 – Процес отримання даних при дистанційному зондуванні

Процес збору даних дистанційного зондування та їхнє використання в географічних інформаційних системах схематично подано на рисунку 3.28.

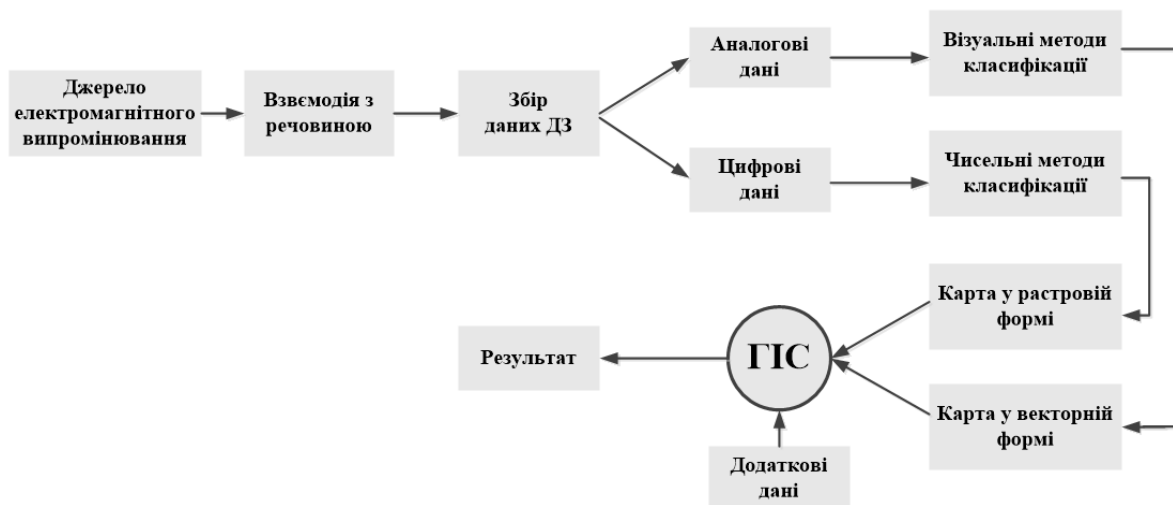


Рисунок 3.28 – Процес збору даних дистанційного зондування та їхнє використання в ГІС

Складовими процесу збору даних є (рис. 3.29):

- джерело електромагнітного випромінювання;
- процес розповсюдження випромінювання;
- взаємодія випромінювання з речовиною об'єкта;
- сигнал у відповідь;
- реєстрація даних і надання їх споживачеві.

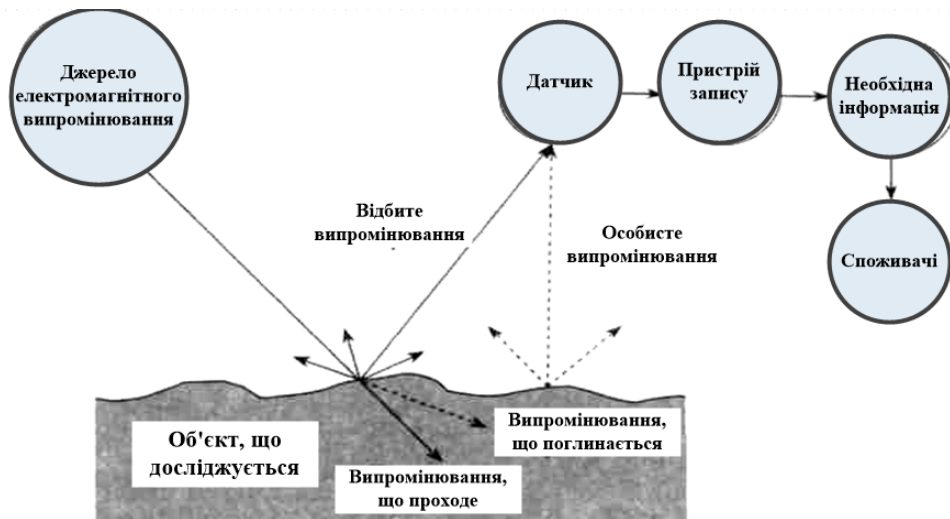


Рисунок 3.29 – Схема дистанційного зондування

Процес збору даних полягає з таких послідовних операцій.

1. Джерело генерує електромагнітне випромінювання:

- з високим рівнем енергії у всьому діапазоні довжин хвиль;
- інтенсивність випромінювання є відомою величиною, яка не залежить від довжини хвилі;
- випромінювання не взаємодіє з атмосферою і розповсюджується через неї без втрати енергії;
- падаюче випромінювання взаємодіє з речовиною об'єкта, внаслідок чого виникає відображене або власне повторне випромінювання, однорідне у всьому діапазоні довжин хвиль.

2. Випромінювання від об'єкта попадає на сенсор, який реєструє просторову інформацію.

Сенсор має достатньо просту й компактну конструкцію, споживає для своєї роботи мінімум енергії та повинен бути високоточним.

3. Дані, зареєстровані сенсором, передаються на комп'ютер, де миттєво перетворюються. Після перетворення форма інформації дозволяє ідентифікувати всі частини об'єкта, що вивчається за їхніми властивостям (фізичними, хімічними і біологічними).

4. Перетворені дані надаються споживачам, від яких потрібна наявність великого досвіду використанні матеріалів ДЗ в конкретних научних областях.

Основні чинники, що знижують надійність даних ДЗ

- Нездатність реальних джерел забезпечити однорідність потоку

випромінювання, як у просторі, так і в часі.

- Зміна інтенсивності випромінювання та його спектру через взаємодію випромінювання з газами атмосфери, молекулами водяного пару й атмосферними частинками.

- Зміна спектральної чутливості однієї і тієї самої речовини за різних умов.

- Можливість збігу спектральної чутливості різних речовин.

- Неможливість створення ідеального сенсора, здатного реєструвати всі довжини хвиль електромагнітного спектра.

- Передача даних із затримкою у часі по технічним обмеженням.

- Передача даних із затримкою у часі через відмінності формату передаваних даних від необхідного споживачу.

- Відсутність у споживача необхідної інформації про параметри збору даних ДЗ і недолік досвіду для їхнього аналізу й дешифрування.

Етапи дистанційного зондування й аналізу даних

Дистанційне зондування можна розглядати як складову частину інформаційної системи. У багатьох областях дані ДЗ є ключовим компонентом у процесі ухвалення рішень.

Принципова схема такого процесу показана на рисунку 3.30.



Рисунок 3.30 – Процес отримання та аналізу даних дистанційного зондування

Споживач та його потреби – головна ланка будь-якої системи управління інформацією. Інформаційні запити логічно пов'язані з вимогами замовників і споживачів продукції до матеріалів ДЗ.

Оптимальний спосіб використання даних ДЗ полягає в їхньому аналізі спільно з інформацією з інших джерел. Тоді вони стають необхідним складником процесу ухвалення розв'язань і моделювання в будь-якій наочній області. Важливим принципом дистанційного зондування є багатокomпонентність. Це реалізується у вигляді різних методів отримання і аналізу даних.

Основні способи отримання інформації при дистанційному зондуванні

Аерокосмічне стереознімання. Отримання зображень із перекриттям з декількох послідовних точок повітряного або космічного простору дозволяє отримати уявлення про тривимірні об'єкти з необхідною точністю та підвищити відношення сигнал/шум.

Багатозональне знімання. Використання багатозональних знімків засновано на унікальності тонових характеристик різних об'єктів. Об'єднання даних яскравості із знімків у різних спектральних діапазонах дозволяє безпомилково виділяти певні просторові структури.

Багаточасове знімання. Планове знімання в наперед певні дати дозволяє виконувати порівняльний аналіз знімків тих об'єктів, характеристики яких змінюються в часі.

Багаторівневе знімання. Знімання з різними рівнями дискретизації використовують для отримання докладнішої інформації про територію, що вивчається. Як правило, весь процес збору даних підрозділяють на три рівні: космічне знімання, аерознімання та наземні дослідження.

Багатополяризаційне знімання. Знімки, отримані цим методом, використовують для проведення меж між об'єктами на основі відмінностей в поляризаційних властивостях відображеного випромінювання. Наприклад, відображене випромінювання від водної поверхні зазвичай сильніше поляризоване, ніж відображене від рослинного покриву.

Комбінований метод. Полягає у використанні багаточасового, багатозонального та багатополяризаційного знімань.

У разі аналізу даних при дистанційному зондуванні на сьогодні все більше використовують **міждисциплінарний аналіз**. Обробку й дешифрування даних у такому разі виконують декілька фахівців із різних наочних областей. Це дозволяє отримати повнішу й достовірнішу інформацію про стан досліджуваного об'єкта. Результати такого аналізу зазвичай представляють у вигляді набору тематичних карт. Переважна більшість даних дистанційного зондування має географічну прив'язку. Дані ДЗ вивчають у взаємозв'язку один з одним. Тому для щоденної роботи й ухвалення розв'язань використовують найефективніший засіб маніпулювання даними, інструмент єдиного підходу до управління і обробки просторової інформації (зокрема й матеріали дистанційного зондування) – географічну інформаційну систему.

Переваги та недоліки даних дистанційного зондування

Основні переваги даних дистанційного зондування

- Містять корисну інформацію, отриману в різних спектральних діапазонах.
- Зберігаються в цифровому вигляді.
- Космічні знімки охоплюють великі області, їх можна використовувати для тематичних регіональних досліджень і ідентифікації великих просторових об'єктів, зокрема структур рельєфу.
- Регулярне знімання території дозволяє проводити моніторинг об'єктів, явищ і процесів, які змінюються під впливом природних і антропогенних чинників (водні ресурси, стан сільськогосподарських культур, ґрунтів, інфраструктури міст).
- Можливість достатньо просто отримати дані про труднодоступні області.
- Можливість отримання знімків різного розподілу для вирішення різноманітних задач у різних наочних областях.
- Виконання аналізу матеріалів ДЗ камерально, що зменшує обсяг польових досліджень, витрати на устаткування та скорочує терміни його окупності.
- Економічно ефективним є застосування космічних знімків для оперативного оновлення середне- та дрібномасштабних карт.
- Кольорові зображення, формовані на основі даних ДЗ в трьох

спектральних каналах, несуть більше інформації, ніж окремі наземні або аерофотознімки.

– Стереопари знімків дають можливість проводити тривимірний аналіз просторових об'єктів.

– Цифровий формат матеріалів ДЗ і використання комп'ютерів для їхньої обробки й аналізу забезпечують швидке отримання результатів.

Основні недоліки даних дистанційного зондування

– Для обробки й аналізу даних потрібна дуже висока кваліфікація й великий практичний досвід.

– При одиничних дослідженнях невеликих територій використання даних стає економічно неефективним.

– На сьогодні космічні знімки ще не можна використовувати для створення планів інженерного призначення.

– Висока вартість програмного забезпечення для обробки цифрових знімків.

– Результати дешифрування матеріалів ДЗ, не підтверджені польовими дослідженнями, не завжди мають достатню надійність.

Технології обробки даних дистанційного зондування.

Пропоновані розв'язання в сфері фотограмметрії та ДЗ

Компанія **ERDAS (Earth Resources Data Analysis System)** була заснована 21 грудня 1978 року й на початку 1979 року була створена перша версія програмного забезпечення для комп'ютерів. Ця програма виявилася першим, і довгий час залишалася єдиним недорогим пакетом для роботи з даними дистанційного зондування в ГІС. Досить швидко від використання програмного забезпечення тільки в своїх роботах, компанія перейшла до широкого продажу цього програмного продукту. Компанія **ERDAS** надавала комерційним організаціям, уряду, армії, учбовим закладам і науково-дослідним організаціям необхідне програмне забезпечення для перегляду й аналізу географічної інформації та перетворення цієї інформації в могутній інструмент для вироблення й ухвалення розв'язань – виявлення та оцінення природних ресурсів, захисту приватної власності. **ERDAS** довгий час залишався світовим лідером у розробці програмного забезпечення для створення додатків із використанням даних дистанційного зондування.

Технологія, розроблена **ERDAS – Geographic Imaging**, визначається як

технологія для тих, кому необхідно створювати, візуалізувати, аналізувати й використовувати географічну інформацію, яка включає зображення в будь-якій формі. Основна технологія, яку розробляла компанія **ERDAS**, називалася «Обробка зображень». По суті, це охоплювало широкий спектр різних сфер застосування, зокрема медичні зображення, металургію і питання технічного спостереження (спостереження роботів). Проте, з розвитком нових знімальних систем, таких як літакові цифрові та космічні сенсори високого й надвисокого розподілу, саме дистанційне зондування стало провідною сферою розробки програмного забезпечення компанії **ERDAS**.

Дистанційне зондування – це технологія, яка вимагає специфічних можливостей обробки, таких як багатоспектральна класифікація, геометрична трансформація та географічна прив'язка зображень. До 1992 року технологія більше розглядалася як науково-дослідна й тому була віддалена від ринку ГІС-технологій. Один елемент є загальним для трьох сфер: ДЗ, ГІС і фотограмметрії – це обов'язкова вимога мати географічно прив'язану інформацію.

До 1994 року для того, щоб мати всі ці функції необхідно було замовляти програмне забезпечення у різних постачальників. З реалізацією версії **ERDAS IMAGINEO 8.1**, перша повнофункціональна ГІС з використанням даних дистанційного зондування стала реальністю.

У квітні 2001 р. компанія **Leica Geosystems**, урахувавши потреби ринку в просторовій інформації та засобах її обробки, придбала права володіння компаніями **ERDAS** і **LH Systems**. Саме вони стали ядром нового відділення ГІС і картографування (**GIS and Mapping Division**) зі штаб-квартирою в Атланті – Джорджія, США. До цього підрозділу увійшли також робочі групи **GPS/GIS** (збір і обробка даних – системи супутникової прив'язки) і **LIS** (геодезичне знімання і земельний кадастр).

Leica Geosystems має широкий спектр сучасних технологій і розв'язань для інтеграції комплексу засобів просторових вимірювань, дистанційного зондування та картографування з можливостями ГІС. В основі програмного забезпечення компанії **Leica Geosystems**, лежить ланцюг обробки даних «від знімка до карти» (**Geospatial Imaging Chain™**), символізуюча весь процес обробки географічних зображень, що полягає із стадій отримання, прив'язки, вимірювань, аналізу та представлення підсумкового результату в необхідному форматі. Ця послідовність перетворює дані на інформацію, а інформацію в знання.

Програмне забезпечення **ERDAS IMAGINE**, компанії **Leica Geosystems**, органічно сполучає цифрову фотограмметрію, дистанційне зондування, просторовий аналіз і картографування у векторному форматі в одному пакеті, який працює як на персональних комп'ютерах, так і на робочих станціях **UNIX**.

Використання даних дистанційного зондування

Області застосування космічних знімків можна розподілити на п'ять загальних категорій.

1. Використання знімка як найпростішої карти або основи, на яку можна наносити дані з інших джерел за відсутності точніших карт, які відображали б сучасну ситуацію.

2. Визначення просторових меж і структури об'єктів для визначення їхніх розмірів і вимірювання відповідних площ. При цьому попередньо проводять геометричну корекцію знімка.

3. Інвентаризація просторових об'єктів на певній території.

4. Оцінка стану території.

5. Кількісна оцінка деяких властивостей земної поверхні.

Дистанційне зондування є перспективним методом формування баз даних, просторове, спектральне і часовий дозвіл яких буде достатнім для виконання завдань раціонального використання природних ресурсів. Ресурси Землі не нескінченні, і оскільки їх експлуатація збільшується в міру зростання кількості та рівня життя населення, все нагальнішим стає завдання їх розумного й ощадливого використання. У цьому плані дистанційне зондування є ефективним методом інвентаризації природних ресурсів та моніторингу їхнього стану. Оскільки дистанційне зондування дозволяє отримувати інформацію про будь-які області Землі, зокрема поверхню морів і океанів, сфери застосування цього методу дійсно безмежні. Сільське й лісове господарство, географія, геологія, археологія, метеорологія та кліматологія, дослідження морського середовища, оцінка стану й використання водних ресурсів – ось неповний перелік сфер застосування даних ДЗ. Основою для експлуатації природних ресурсів слугує аналіз інформації про землекористування та стан земних покривів. Крім збору такої інформації, дистанційне зондування використовують також для вивчення таких природних катастроф, як землетруси, повені, обвали й осідання ґрунту. Сфери застосування даних ДЗ коротко подано в таблиці 3.1.

Таблиця 3.1 – Сфери застосування даних дистанційного зондування

Сільське та лісове господарство	Землекористування	Геологія	Водні ресурси	Океанографія	Вивчення навколишнього середовища
1	2	3	4	5	6
Визначення типів рослинності	Класифікація землекористування	Визначення типу гірської породи	Картографування меж водних поверхонь	Дослідження живих організмів	Моніторинг місць розробки родовищ
Прогноз врожайності та контроль стану посівів	Картографування й оновлення карт	Картографування основних геологічних структур	Картографування місць затоплення	Дослідження течій та мутності води	Картографування та моніторинг забруднення водного середовища
Інвентаризація лісів	Оцінка придатності земель	Оновлення геологічних карт	Визначення меж і товщини снігового покриву	Картографування змін берегової лінії	Моніторинг забруднення повітряного середовища
Оцінка стану пасовищ і визначення рослинної біомаси	Поділ міських і сільських територій	Визначення меж пухких відкладень	Гляціологія	Картографування рельєфу мілководного шельфу	Визначення наслідків природних катастроф
Дослідження стану рослинного покриву	Регіональне планування	Картографування вулканічних інтрузій	Вивчення «цвітіння води», перенесення та осадження наносів	Стеження за льодовою ситуацією	Моніторинг антропогенного впливу на навколишнє середовище
Визначення навантаження на рослинний покрив	Експлуатація транспортних мереж	Картографування новітніх вулканогенних відкладень	Інвентаризація озер	Вивчення хвиль і океанічних вихорів	

Продовження таблиці 3.1

1	2	3	4	5	6
Визначення стану ґрунтового покриву	Картографування кордонів водних об'єктів	Картографування форм рельєфу	Визначення меж іригаційних полів		
Визначення складу ґрунтів		Виявлення регіональних структур			
Оцінка поширення лісових пожеж		Виявлення лінеаментів			

Питання для самоконтролю

1. Перелічіть вимірювальні операції, які здійснюються в ГІС.
2. Які основні операції мережевого аналізу існують в ГІС?
3. Які операції аналізу поверхонь є в ГІС?
4. Структура системи ДЗ. Наземний і орбітальний сегменти. Способи передачі даних ДЗ.
5. Якими способами зі знімків вилучається необхідна для досліджень інформація?
6. Які існують методи дешифрування?
7. Перелічіть переваги автоматичного дешифрування знімків перед візуальними.
8. Які види аерокосмічних знімків існують?
9. Перелічіть основні характеристики даних ДЗ.
10. Перелічіть вимоги, що висуваються до даних ДЗ під час виконання різних прикладних завдань.

3.2 Застосування геоінформаційних систем для виконання завдань народного господарства

Застосування геоінформаційних систем у бізнесі

Технологія геоінформаційних систем швидко проникає в сферу бізнесу. Згідно з **GIS STRATEGIES**, щоквартальному огляду світового ринку ГІС американської компанії **Dataquest** і журналу **GIS World**, проникнення ГІС у бізнес відбувається швидше, ніж у більшість інших сфер їхнього застосування.

Про підвищений інтерес бізнесменів до цієї могутньої сучасної технології свідчить регулярне проведення низки конференцій і виставок. У США виходить спеціальний журнал «**Business Geographies**», додаток до **GIS World**, присвячений опису базових принципів ГІС і використанню цієї технології в бізнесі. Усе частіше зустрічається поняття **геомаркетинг (geomarketing)**, що зв'язує в нерозривне ціле бізнес і геоінформаційні технології.

Перелік основних переваг ГІС перед іншими інформаційними технологіями наведено нижче.

- Наявність набору засобів створення й об'єднання баз даних.
- Забезпечення географічного аналізу та наочної візуалізації БД у вигляді різних карт, графіків, діаграм.
- Можливість безпосередньої прив'язки один до одного в режимі **Hot Link** усіх атрибутивних і графічних даних.

Зі свого боку, сфера застосування ГІС у бізнесі охоплює таке:

- аналіз і відстеження поточного стану та тенденцій зміни ринку;
- планування ділової активності;
- оптимальний вибір місця розташування нових філіалів фірми або банку, торгових точок, складів, виробничих потужностей;
- підтримка ухвалення розв'язань;
- вибір найкоротших або найбезпечніших маршрутів перевезень і шляхів розподілу продукції;
- аналіз ризику матеріальних вкладень і врегулювання розбіжностей;

– демографічні дослідження, що проводяться з метою визначення попиту на продукцію;

– географічна прив'язка баз даних про земле- й домоволодіння.

Комерційні фірми використовують ГІС для вибору місця розташування нових супермаркетів: розташування складу й зони обслуговування визначають за допомогою моделювання доставки та впливу конкурентів.

У різних країнах ГІС дедалі більше використовуються в бізнесі. За допомогою ГІС, наприклад, бізнесмени **Великобританії** відкривають нові супермаркети, бензоколонки та станції техобслуговування автомобілів, а також на основі демографічного аналізу й моделювання розвивають мережу популярних у цій країні пивних барів-пабів.

У **Південній Африці** ГІС застосовуються для такого: оптового й роздрібного продажу автомобілів; розсилки та доставки пошти й іншої кореспонденції, зокрема рекламної відповідно до індивідуальних потреб, професійних інтересів і прибутків кожного занесеного в базу даних жителя; оптових поставках бакалійних товарів; створення інформаційної системи з адресною та картографічною прив'язкою по комерційних компаніях і фірмах з обсягами продажів понад 50 тис. доларів.

В **Іспанії** ГІС використовуються великими банками для розробки планів розвитку й координації діяльності регіональних центрів з обслуговування вкладників.

У **Франції** користувачами ГІС є, наприклад, автомобільні компанії **Citroen, Renault і Peugeot**, які активно впроваджують картографію в свою повсякденну діяльність. Регіональний центр **IBM-France** зміг збільшити обсяг продажів апаратних і програмних засобів, прискорив сервісне обслуговування клієнтів за допомогою підвищення ефективності взаємодії з 1 200 бізнес-партнерами, нових можливостей оперативного аналізу результатів роботи своїх підрозділів і численних дилерів унаслідок сумісного використання потенціалу аналітичних засобів ГІС і власної бази даних «Траєкторія» (**Trajectoire**). Досвід використання подібної зв'язки виявився таким успішним, що й сама універсальна аналітична бізнес-система стала ринковим товаром, зацікавленість у придбанні якого виявила низка фірм, безпосередньо не пов'язаних із комп'ютерним бізнесом.

У **Новій Зеландії** компанія **Eagle Technology** на базі пакету **Arc View** розробила власний додаток **View/NZ** – багатофункціональний засіб

аналізу зведених табличних, текстових і картографічних бізнес-даних, демографічних, статистичних, земельних, муніципальних, адресної та іншої інформації. Використання цього додатка допомагає переорієнтувати головну мету маркетингових зусиль із задоволення усереднених потреб населення міста або району на оперативну реакцію на запити кожної людини, що живе або працює в зоні реалізації товарів фірми. Принципово новий рівень сервісу, що досягається при такому підході, отримав найменування персоніфікованого маркетингу (**personal marketing**).

У США, розташованою в місті Сан-Дієго (штат Каліфорнія) компанією **Equifax National Decision Systems** розроблена ГІС-система **Infomark-GIS**, спеціально призначена для маркетингових додатків і забезпечення процесу ухвалення розв'язань. Система легко інтегрується з понад 60 національними базами бізнес-даних, може бути без великих додаткових зусиль локалізована під спеціальні завдання, типові для операцій із нерухомістю, ресторанного бізнесу, продажу товарів повсякденного попиту, діяльності комунальних служб, банківсько-фінансової індустрії.

Одними з перших користувачів системи стали компанії **Levi Straus & Co., Tennessee Valley Authority, Boston Chicken і Friday's** (колишня **TG1 Fridays**). Система об'єднує засоби пакетів **ArcInfo, Arc View** і власного продукту компанії-розробника **Infomark**.

Останнім десятиріччям у США й інших країнах з'явилася численна група компаній, що спеціалізуються на консультативному обслуговуванні бізнесу, які проводять на замовлення аналітичні маркетингові дослідження на базі ГІС.

Наприклад, фахівці компанії **Castillo Company, Inc.**, Фенікс (штат Арізона), застосовуючи пакет **ArcInfo**, можуть на територіях понад 50 країн світу виявити райони з певним складом населення, розташовані на заданій відстані від аеропорту, з будинками, що мають певну середню вартість, або відповідаючі деяким іншим критеріям.

Замовниками маркетингових, демографічних, соціологічних, політологічних і багатьох комплексних міждисциплінарних завдань є як приватні фірми, так і державні організації, наприклад Геологічна служба США.

Результати, що надаються внаслідок аналізу, сприяють вибору оптимальних, найвигідніших стратегій і тактик дій, швидкій реакції на

умови ринку, що змінюються, за необхідності – переорієнтації профілю діяльності комерційних фірм. Останні домінують серед клієнтів **Castillo Company, Inc**, із поміж них такі гіганти комп'ютерного бізнесу, як **Motorola і Intel**.

Не обходиться без ГІС і така специфічна сфера бізнесу, як швидка доставка кореспонденції. Більше 25 років приватна компанія **Federal Express** займається розсилкою поштових відправлень по всьому світу. У цій роботі, що вимагає особливої ретельності, останніми роками задіяні засоби геокодування пакету **ArcInfo**. В його базі даних зберігаються адреси, поштові індекси, назви, імена та прізвища мільйонів жителів і організацій різних країн. До відповідних карт прив'язано їхнє місце розташування, маршрути й розклади авіарейсів, межі адміністративних районів, інша корисна для успішної роботи інформація.

Перспективи використання ГІС у бізнесі

На думку багатьох бізнесменів і аналітиків, сфера додатків ГІС-технологій безмежна. Вони входять у світ бізнесу, перевернувши всі уявлення про призначення й економічну ефективність географічних методів візуалізації та аналізу інформації. ГІС може перетворити цю інформацію в нове, унікальне за своєю прикладною цінністю знання.

Особливо успішно й вигідно використання ГІС-технологій при масових перевезеннях вантажів і людей, при створенні мереж оптимального розміщення торгових точок, аналізі існуючих і потенційних ринків і районів збуту продукції, в нафтових, газових і електричних компаніях, а також у комерційних фірмах, що займаються операціями з нерухомістю, для обґрунтування, розширення та підтримки банківських операцій, у роботі авіакомпаній і телекомунікаційних корпорацій, низці інших сфер ділової активності.

Кінцева мета використання ГІС – якнайкраще задоволення потреб і запитів покупців і клієнтів, до того ж як сьогодні, так і в майбутньому і, як наслідок, процвітання та стабільно висока конкурентоспроможність фірми.

Спектр пропонованих **ESRI** програмних ГІС-продуктів найширший порівняно з конкурентами на ринку геоінформаційних технологій. Він включає прості засоби кінцевого користувача **Arc View** та повнофункціональні ГІС – **PC ArcInfo** і **ArcCAD**, які працюють на персональних комп'ютерах, а також наймогутніший за функціональними

можливостями програмний пакет **ArcInfo**.

Важливою перевагою є **комплементарність** (повна сумісність) усіх продуктів різних рівнів **ESRI**. Результати роботи з одним пакетом не пропадуть, якщо користувач вирішить доцільним замінити його або додатково використовувати будь-який інший програмний продукт із сімейства **ESRI**. Усі вони працюють в єдиному інформаційному середовищі **ArcInfo**. У цьому самому середовищі написані сотні спеціалізованих програмних продуктів (додатків) для багатьох галузей науки, техніки та бізнесу. При розробці останніх версій програмних продуктів **ESRI** передбачені можливості використання найпоширеніших типів зовнішніх реляційних баз даних і конвертерів файлів даних популярних форматів.

Як показує практика, для великих компаній, що мають розгалужену структуру філіалів, виробництв і магазинів та ведуть свої комерційні операції на великій території, доцільно задіяти пакет **ArcInfo**, краще в поєднанні з **ArcGis**.

ArcGis надає прості й ефективні засоби для візуалізації та аналізу будь-яких даних про об'єкти та явища, довільно розподілених по території (геоданих). Він може використовуватися менеджерами, планувальниками, аналітиками та вченими для повнішого розуміння сутності виникаючих перед ними проблем реального життя, пов'язаних із діяльністю їхніх компаній, адресно-географічної прив'язки корисної для вирішення цих проблем інформації.

Наприклад, при виборі місця для нового магазину по карті в **ArcGis** користувач має можливість спочатку переглянути дані про продажі товару за попередні місяці в інших торгових точках, про демографічний і соціальний склад потенційних покупців (жителів довколишніх будинків або мікрорайонів), додатково вивести на екран монітора фотографію або по поверховий план будівлі, де передбачається відкрити магазин, підсвітити на карті, виведеній як частина того самого зображення, магазини конкурентів (рис. 3.31).

Подібні операції зазвичай займають мінімум часу – весь аналіз виконується за один сеанс роботи. Достатньо швидко можна перебрати низку варіантів передбачуваних маркетингових операцій. Після такого аналізу доцільність відкриття магазину може стати очевидною, або стає зрозуміло, що потрібно добирати інше місце або змінити асортимент

товарів і обсяг товарообігу.

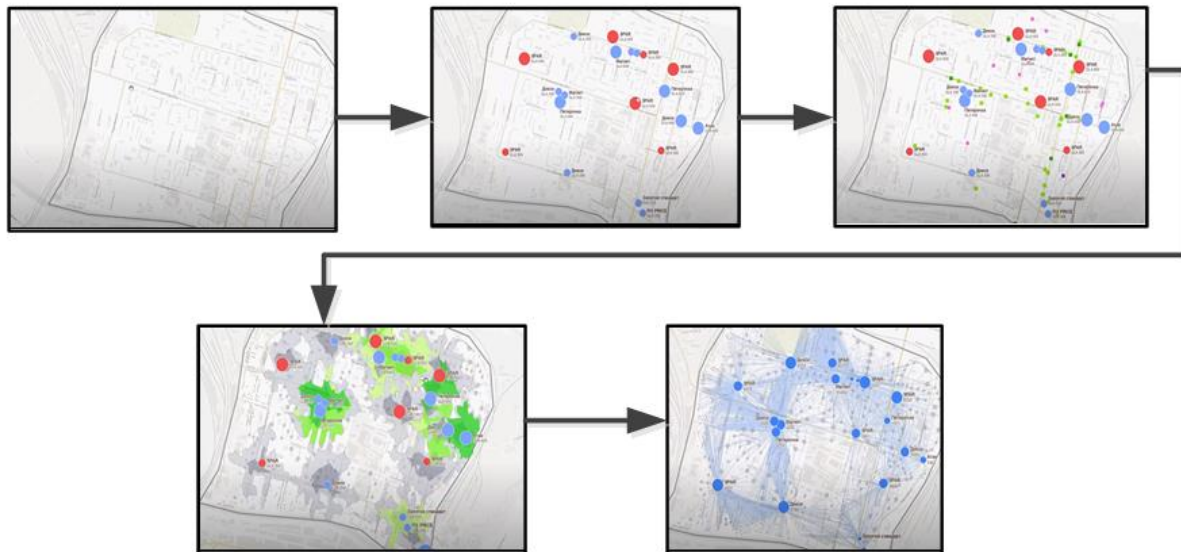


Рисунок 3.31 – Приклад пошуку місця розташування для нового магазину по карті в **ArcGis**

Пакет **ArcGis** надає алгоритм процедури ухвалення рішень, важливих для функціонування компанії. У процесі планування розвитку центрів економічного (комерційного) управління виникає потреба в оптимальному управлінні сукупністю комерційних організацій, транспортними потоками, мережами комунікацій тощо.

Виконання подібних завдань здійснюється методами ГІС. На початку створюється інтегрована інформаційна основа. Потім геокодована інформація про об'єкти завантажується в базу даних ГІС, яка вже має картографічну інформацію в потрібних масштабах. Шляхом використання методів теоретичного й чисельного аналізу, лінійного програмування розв'язуються задачі оптимізації. У результаті виконання подібних завдань здійснюється вибір оптимального місцеположення комерційних центрів, вибір областей (сфер) впливу цих центрів, оптимізація транспортних потоків, оптимізація інформаційного (зокрема рекламного) забезпечення.

Схема застосування геоінформаційної технології у сфері бізнесу досить проста. Будь-яка компанія веде зазвичай декілька баз даних. Навіть якщо це відсутнє, в будь-якій компанії використовується довідник телефонів, довідник адрес клієнтів або партнерів, довідник можливостей і послуг. Ці дані необхідно систематизувати й наочно представити, щоб підвищити ефективність роботи з ними. Для цих цілей необхідно придбати настільну (**desktop**) ГІС і набір цифрових карт відповідної тематики.

Каталог адрес переводиться в базу даних ГІС і стає атрибутивною характеристикою карти, створюючи на ній відповідне навантаження. Потім заповнюються даними інші бази: про клієнтів, постачальників, замовників. Додатково завантажується інформація про терміни транспортування.

Унаслідок таких дій ГІС дозволить:

- швидко виявити по карті, де «ховаються» покупці та конкуренти;
- визначити найвигідніше для бізнесу місце розташування нових виробничих потужностей, філіалів і торгових точок;
- скласти зведені діаграми об'ємів продажів за місяць або рік по торгових підприємствах, що цікавлять;
- прив'язати діаграми до відповідних місць на картах;
- візуально оцінити й отримати повноцінне статистичне зведення за динамікою попиту та пропозиції в будь-якій сфері ринку, наприклад в операціях із нерухомістю (рис. 3.32);

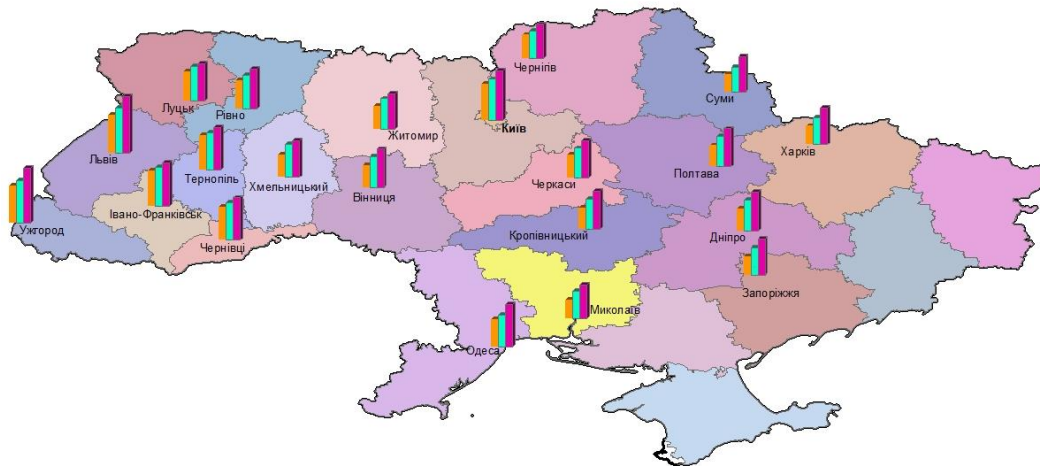


Рисунок 3.32 – Приклад аналізу в операціях із нерухомістю за допомогою атрибутивної інформації по карті в **ArcGis**

- виділити маркетингові території та проаналізувати наявну по них інформацію;
- візуально по карті й на основі супутньої цифрової та текстової інформації провести порівняння демографічних характеристик по різних країнах, областях і районах;
- виявити й оконтурити неблагополучні за екологічними ознаками райони або зони підвищеної чутливості природного середовища до антропогенних дій (рис. 3.33);

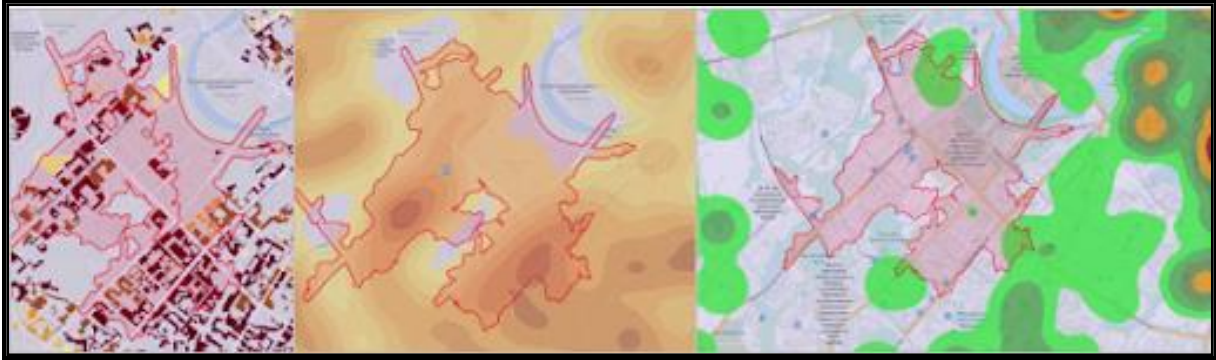


Рисунок 3.33 – Приклад виділення на карті зон, що цікавлять

- нанести на карту, виділити й доповнити супутньою інформацією зони виробництва, зберігання, скидання та накопичення шкідливих для людей і живих організмів речовин і матеріалів;
- вивчити взаємозв'язки між різними чинниками, наприклад між повторюваністю стихійних природних явищ і вартістю нерухомості по будь-якій території;
- визначити ступінь відповідності забрудненої території внаслідок роботи підприємства природоохоронному законодавству, що діє в країні;
- включити в базу даних і вивести на екран тексти відповідних законодавчих актів при натисканні мишею по об'єкту на карті;
- ухвалювати обґрунтовані рішення на підставі всебічного аналізу інформації, що є у розпорядженні;
- виявити сфери дії компанії та конкурентів для вибору оптимальної стратегії;
- на основі внутрішніх ресурсів ГІС оптимізувати задачі (поставка в найкоротші терміни з мінімальними витратами, облік ресурсів або взаємодії учасників операції).

Геомаркетингові дослідження

Маркетинг – це поняття, що об'єднує в собі комплекс інструментів і методів збору, обробки, аналізу та візуалізації просторової інформації для оперативних і стратегічних завдань компанії.

Геомаркетинг виник на стику двох наук: маркетинг і геоінформатика, і став новим напрямом маркетингового аналізу з використанням методів геоінформатики, реалізованих у ГІС (рис. 3.34).



Рисунок 3.34 – Схематичне представлення поняття «Геомаркетинг»

Виокремлюють два визначення геомаркетингу:

Геомаркетинг (як концепція) – ринкова концепція управління сучасним виробництвом на основі геоінформаційних технологій, яка передбачає використання просторово локалізованої інформації для підтримки ухвалення розв’язань.

Геомаркетинг (як технологія) – процес планування, ухвалення розв’язань, ціноутворення, просування та реалізації ідей, товарів і послуг (зокрема інформаційні й політичні) за допомогою обміну, що задовольняє цілі окремих осіб і організацій, за допомогою геоінформаційних технологій.

Мета геомаркетингу

Основна мета – підвищення ефективності ведення бізнесу на основі просторово-часового вивчення реальних різномірних даних – моніторинг, прогноз, управління просторово-розподіленими бізнес-структурами завдяки:

- виявленню прихованих закономірностей поведінки споживача, зміни попиту на продукцію в просторово-часовому розрізі (модель Хаффа);
- ухваленню розв’язань, спрямованих на ведення ефективного бізнесу в умовах швидко змінного соціально-економічного середовища.

Завдання, які виконує геомаркетинг

Основні завдання для бізнесу та торгово-роздрібних мереж і територіально-розподіленого сервісу:

- оптимальное планування мережі торгових точок й сервісу (рітейлу);
- відкриття торгової точки в оптимальному місці (з огляду на критерії доступності, максимальне охоплення потенційних споживачів, їхнє мешкання та потоки, конкурентів, розвиток територій);
- позиціонування торгової точки (формат, площа, асортимент);
- управління асортиментом товарів і просування торгової точки;
- оперативний збір, оновлення й візуалізація інформації про ринки;
- бенчмаркетинг (конкурентний аналіз мережевих магазинів);
- побудова торгової зони й аналіз оточення навкруги наявної адреси: споживач, конкуренція, розвиток;
- прогноз виручки (або товарообігу) та інших показників економічної ефективності присутності торгової точки в певному конкретному місці.

Основні завдання для консалтингових компаній, риелтерів і девелоперских компаній, що аналізують великі об'єкти комерційної нерухомості (торгові комплекси різних форматів, бізнес-центри):

- побудова торгової зони (ресурсної зони, зони транспортної доступності, зони обслуговування, зони охоплення);
 - оцінка споживача в зоні транспортної та пішохідної доступності навкруги об'єкта (пішохідні потоки, розрахунок населення, що проживає, транспортні потоки);
 - прогноз споживача (пішохідних потоків, населення) в зоні доступності через певний відрізок часу;
 - конкурентна ситуація навкруги об'єкта зараз і прогноз будівництва залежно від функціонального призначення через певний відрізок часу;
 - візуалізація портрета споживача в зоні доступності;
 - планування розміщення площ для орендарів, а також профіль самих орендарів, залежно від розподілу геомаркетингових чинників у зоні доступності для досягнення максимальної вигоди власника об'єкта;
 - аналіз ринку нерухомості (вивчення розподілу попиту та пропозиції за типами нерухомості, вивчення інтенсивності будівництва житлових і нежилых об'єктів);
 - планування транспортних вузлів для організації руху транспорту.
- Геомаркетингові продукти (геомаркетингові звіти, геомаркетингові

інформаційні системи) допомагають інвесторам, керівникам, менеджерам середньої ланки, маркетологам, аналітикам, оцінювачам, консультантам сприймати інформацію та ухвалювати ефективні рішення. Вони наочно допомагають обґрунтовувати бізнес-план торгової мережі, концепцію будівництва великих комерційних об'єктів і ділових центрів, спираючись на аналіз ринку загалом по місту і, зокрема, в безпосередній близькості від об'єкта.

Методи в геомаркетинговому дослідженні спрямовані на застосування двох підходів:

1. Обґрунтування відкриття торгової точки, коли підбираються нові місця для планування мережі «з нуля» або розширення мережі з точністю до кварталу.

2. Геомаркетинговий аналіз територій, коли відома ділянка (або об'єкт нерухомості) і необхідно проаналізувати оточення для уточнення ряду характеристик майбутнього об'єкта, а також спрогнозувати показники стосовно цього потенційного місця.

Геомаркетингова система на базі геоінформаційної

Геомаркетингова інформаційна система (або корпоративна ГІС) – програмно-апаратний комплекс, який складається з програмного забезпечення, інформаційних ресурсів і баз геоданих, який дозволяє здійснювати збір, зберігання, обробку, аналіз, моделювання та прогноз просторової інформації про ринки й споживачів для досягнення оперативних і стратегічних цілей компанії у сфері нерухомості та торгівлі.

Мета

Головна мета – це безперервний внутрішній і зовнішній моніторинг різномірної географічно прив'язаної інформації про ринок і споживача, її візуалізація на єдиній картографічній основі для виконання оперативних завдань компанії.

Внутрішній моніторинг – збір і аналіз інформаційних потоків компанії.

Зовнішній моніторинг – збір і аналіз інформаційних потоків про ринок, демографічну ситуацію, конкурентне середовище та інші чинники.

Сфери застосування

– Фаст-фуд ритейл.

- Банківський ритейл.
- Фармацевтичний ритейл.
- Медичний ритейл.
- Фітнес-ритейл.
- Інші сфери ритейлу.
- Девелопмент і інвестиції в торгівельні центри, торгівельно-розважальні центри.
- Будівництво житла та житлових комплексів.
- Дротовий і бездротовий зв'язки.
- Поштова та кур'єрська доставка.
- Виробництво й поставка товарів.
- Транспорт.

Властивості

- Повністю інтегровані в сучасні інформаційні системи.
- Дозволяють виконувати повний спектр завдань компанії: можливість нанесення власних даних, робота з великим набором шарів, отримання інформації про територію, збереження звіту або графіки, отримання інформації про статистичні показники стосовно будь-якої ділянки території.

Програмні рішення

– Настільне (**Desktop**) програмне забезпечення геомаркетингових систем на базі ГІС поставляється як дистрибутив – так званий «товстий клієнт».

– Може бути й серверна реалізація, коли дистрибутиви ставляться на сервер, а робочі місця набуваються залежно від задач (введення даних, аналіз). «Тонкі» клієнти дозволяють заощадити на кількості ліцензій, оскільки набуваються робочі місця під стандартні інтернет-браузери.

– Програмне забезпечення може базуватися на існуючому програмному забезпеченні (**ESRI, Map Info**). Необхідно пам'ятати, що програмне забезпечення геомаркетингових систем на базі ГІС поставляється зазвичай без карт (базового набору просторових даних) і без геоданих для аналізу.

Переваги

- Інтеграція різномірної інформації про територію між різними філіалами компанії на одній платформі.
- Інтеграція різномірної інформації про територію між різними філіалами компанії та іншими структурами на одній платформі.
- Управління ефективним доступом до єдиних баз даних для знаходження певних запитів за заданими критеріями як в Інtranеті, так і в Інтернеті.
- Можливість доступу до повного набору інструментів для обробки даних, редагування, створення власних даних.
- Можливість побудови власних математичних моделей за даними.
- Можливість створення власних різних картографічних звітів.
- Можливість створення додаткових надбудов для вирішення вузьких задач.

Застосування геоінформаційних систем в екологічних дослідженнях

Сьогодні сучасні методи й інструменти здатні фіксувати досить достовірну інформацію про забруднення навколишнього середовища в межах міста або іншого територіального суб'єкта. На сьогодні у будь-якій державі діють різні державні та комерційні системи моніторингу, приналежні до різних відомств. Завдання ГІС – інтегрувати географічно прив'язану екологічну інформацію в одному місці, а також показати причини забруднення: звідки йде хмара, яке джерело забруднення, де воно знаходиться. Потрібно відзначити зону потенційного ризику, побудувати прогнози розповсюдження забруднення, зрозуміти стійкі області забруднення в часі.

Тобто завдання ГІС – надати довідкову інформацію про розповсюдження тієї або іншої речовини за даними вимірів в атмосфері, ґрунті й воді в межах тієї або іншої території та дати аналітичні розрахунки або зробити прогноз ситуації. Незалежна експертиза зазвичай необхідна в судових розглядах, екологічних обґрунтовуваннях розміщення будівельного об'єкта в конкретному місці, оцінках дії на навколишнє середовище.

Особливості ГІС, які використовують у екології

Комплексність (системність) – ГІС, це не тільки дані та не тільки програмне забезпечення. ГІС – це комплекс програмного, апаратного, інформаційного забезпечення, керований спеціальним персоналом.

Просторовість – інструмент, що працює з будь-якими даними, розподіленими в просторі й маючими свою систему координат, починаючи від колоній мікроорганізмів і закінчуючи цілком планетою Земля. ГІС також дозволяє здійснювати операції з даними, що не мають просторову прив'язку, але основна функціональність ГІС орієнтована саме на роботу з просторовими даними.

Зв'язаність – наявність тісного взаємозв'язку між просторовою та атрибутивною інформацією. У межах ГІС вперше ці два типи інформації були тісно поєднані, багато в чому це визначило появу ГІС у вигляді окремої сфери програмного забезпечення. Часто ГІС називають системою керування базами даних (СКБД) із можливістю створення карт або системою цифрової картографії з розширеною підтримкою баз даних.

Функції ГІС, які використовують у екології

Візуалізація – ГІС могутній засіб представлення даних. Ця якість ГІС зазвичай використовується першою, за допомогою ГІС створюються наочні ілюстративні карти та схеми.

Сучасні ГІС надають багато уваги **легкості та продуктивності**. Саме через ці функції вони часто розглядаються лише як засіб створення карт. Ця функція ГІС успадковувана від картографії, яку цілком можна вважати родоначальницею ГІС і вона донині є основною функцією.

Організація інформації – одна з основних функцій ГІС. Сучасні ГІС – зручний інструмент, який допомагає управляти інформацією, використовуючи просторовий принцип.

Обробка й аналіз – функції ГІС, що перетворюють її з інструмента для роботи з готовими даними (візуалізатора) в інструмент для створення нових даних на їхній основі, моделювання та прогнозування.

Використання геоінформаційних систем у новій галузі – екології, дозволяє оперативно одержувати інформацію за запитом і відображати її на картографічній основі, оцінювати стан екосистеми й прогнозувати її розвиток.

Можливості ГІС, які застосовуються в екології

- Введення, накопичення, зберігання й обробка цифрової картографічної та екологічної інформації.
- Побудова на підставі отриманих даних тематичних карт, що відображають поточний стан екосистеми (рис. 3.35).

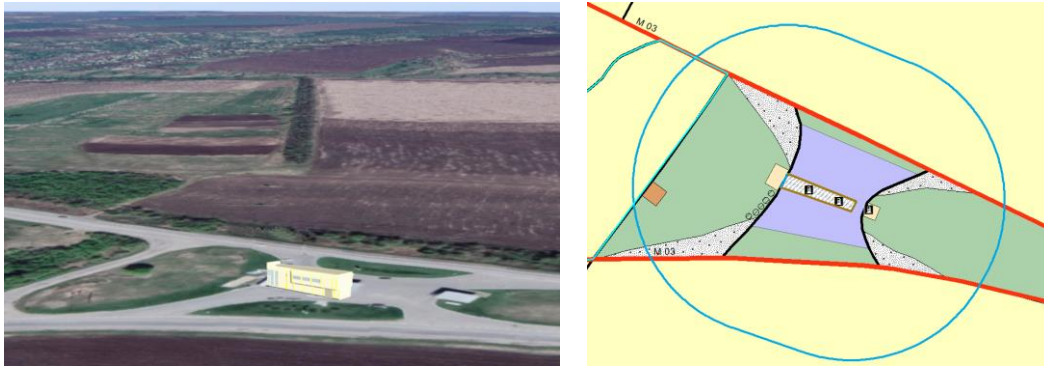


Рисунок 3.35 – Моделювання буферної зони навколо екологонебезпечного об'єкта

- Дослідження динаміки зміни екологічної ситуації в просторі та часі.
- Побудова графіків, таблиць, діаграм – моделювання розвитку екологічної ситуації в різних середовищах і дослідженнях.
- Залежності стану екосистеми від метеоумов, характеристик джерел забруднень, значень фонових концентрацій.
- Отримання комплексних оцінок стану об'єктів навколишнього природного середовища на основі різнорідних даних.

Екологічні проблеми часто вимагають негайних і адекватних дій, ефективність яких безпосередньо пов'язана з оперативністю обробки й подання інформації. При комплексному підході, властивому екології, доводиться спиратися на узагальнювальні характеристики навколишнього середовища, унаслідок чого обсяги навіть мінімально достатньої початкової інформації, поза сумнівом, повинні бути великими. Інакше обґрунтованість дій і рішень навряд чи може бути досягнута. Проте простого накопичення даних також, на жаль, недостатньо. Ці дані повинні бути легко доступні, незалежно від їхнього обсягу систематизовані відповідно до потреб. Добре, якщо є можливість пов'язати різнорідні дані один з одним, порівняти, проаналізувати, просто переглянути їх у зручному та наочному вигляді, наприклад, створивши на їхній основі

необхідну таблицю, схему, креслення, карту, діаграму. Групування даних у потрібному вигляді, їхнє належне зображення, зіставлення й аналіз цілком залежать від кваліфікації та ерудованості дослідника, вибраного їм підходу інтерпретації накопиченої інформації. На етапі обробки й аналізу зібраних даних істотно, але ні в якому разі не перше, місце посідає технічна оснащеність дослідника, що включає відповідні для виконання поставленого завдання апаратні засоби та програмне забезпечення. Функції такого програмного забезпечення все частіше виконує технологія географічних інформаційних систем.

Доцільність використання ГІС при вирішенні проблем екології

У традиційному уявленні можливі межі інтеграції різнорідних даних штучно обмежуються. Наприклад, ГІС надають можливість створення карти врожайності полів шляхом об'єднання даних про ґрунти, клімат і рослинність. ГІС дозволяє піти значно далі. До наведеного вище набору даних можна додати демографічну інформацію, відомості про земельну власність, добробут і доходи населення, обсягах капітальних вкладень і інвестицій, зонування території, стан хлібного ринку. Унаслідок цього з'являється можливість безпосередньо визначити ефективність запланованих або передбачуваних заходів, що вживаються з метою збереження природи, їхній вплив на життя людей і економіку сільського господарства. Можна піти ще далі і, додавши дані про поширення захворювань і епідемій, встановити, чи є взаємозв'язок між темпами деградації природи та здоров'ям людей, визначити можливість виникнення й поширення нових захворювань (рис. 3.36).

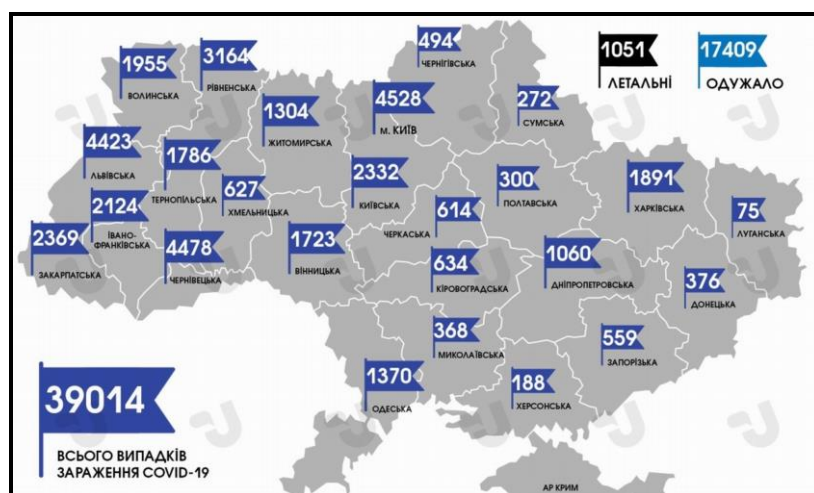


Рисунок 3.36 – Відображення на карті епідеміологічної ситуації в країні

У кінцевому підсумку, вдається досить точно оцінити всі соціально-економічні аспекти будь-якого процесу, наприклад скорочення площі лісових угідь або деградації ґрунтів.

Роль і місце ГІС у природоохоронних заходах

Деградація середовища

ГІС з успіхом використовується для створення карт основних параметрів навколишнього середовища. Надалі, після отримання нових даних, ці карти використовуються для виявлення масштабів і темпів деградації флори та фауни. При введенні даних, отриманих дистанційно, зокрема з супутників і звичайних польових спостережень, можна здійснювати моніторинг місцевих і широкомасштабних антропогенних дій. Дані про антропогенні навантаження доцільно накласти на карти зонування території з виділеними областями, що становлять особливий інтерес із природоохоронної точки зору, наприклад із парками, заповідниками й заказниками. Оцінку стану та темпів деградації природного середовища можна проводити й по виділених на всіх шарах карти тестовим ділянкам.

Забруднення

За допомогою ГІС зручно моделювати вплив і розповсюдження забруднення від точкового та просторового джерела на місцевості, в атмосфері й по гідрологічній мережі. Результати модельних розрахунків можна накласти на природні карти, наприклад карти рослинності, або ж на карти житлових масивів у конкретному районі. У результаті можна оперативно оцінити найближчі і майбутні наслідки таких екстремальних ситуацій, як розлив нафти й інших шкідливих речовин, а також вплив постійно діючих точкових і просторих забруднювачів.

Землеволодіння

ГІС широко застосовуються для складання та ведення різноманітних, зокрема земельних, кадастрів. За їхньою допомогою зручно створювати бази даних і карти по земельній власності, об'єднувати їх із базами даних за будь-якими природними й соціально-економічними показниками, накладати відповідні карти одна на одну та створювати комплексні (наприклад ресурсні) карти (рис. 3.37), будувати графіки й різноманітні

діаграми.

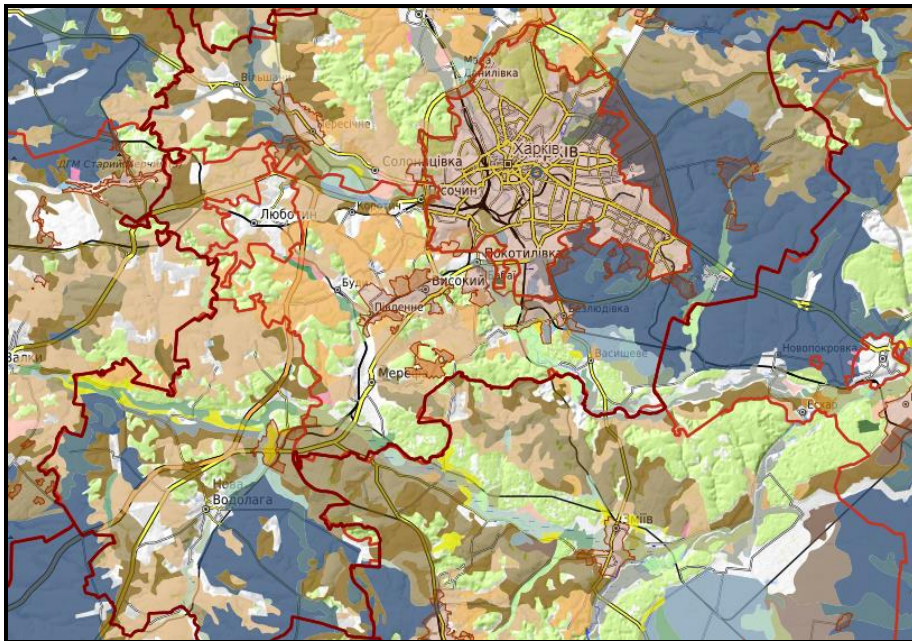


Рисунок 3.37 – Фрагмент публічної кадастрової карти

Території, що охороняються

Ще одна поширена сфера застосування ГІС – збір і управління даними по територіях, що охороняються, таким як заповідники, заповідники та національні парки. У межах районів, що охороняються, можна проводити повноцінний просторовий моніторинг рослинності, цінних і рідких видів тварин, визначати вплив антропогенних втручань, таких як туризм, прокладання доріг або ліній електронапруги, планувати й доводити до реалізації природоохоронні заходи. Можливе виконання завдань, розрахованих на багато користувачів, таких як регулювання випасу худоби та прогнозування продуктивності земельних угідь. Такі завдання ГІС виконує на науковій основі, тобто вибираються рішення, що забезпечують мінімальне навантаження на дику природу, збереження на необхідному рівні чистоти повітря, водних об'єктів і ґрунтів, особливо в часто відвідуваних туристами районах.

Території, що не охороняються

Регіональні та місцеві керівні структури широко застосовують можливості ГІС для отримання оптимальних рішень проблем, пов'язаних із розподілом і контрольованим використанням земельних ресурсів, узгодженням конфліктних ситуацій між власником і орендарями земель.

Корисним і часто необхідним буває порівняння поточних меж ділянок землекористування із зонуванням земель і перспективними планами їхнього використання. ГІС забезпечує також можливість зіставлення меж землекористування з вимогами дикої природи. Наприклад, у деяких випадках буває необхідним зарезервувати коридори міграції диких тварин через освоєні території між заповідниками або національними парками. Постійний збір і оновлення даних про межі землекористування може надати велику допомогу при розробці природоохоронних, зокрема адміністративних і законодавчих заходів, відстежувати їхнє виконання, своєчасно вносити зміни та доповнення в наявні закони й ухвали на основі базових наукових екологічних принципів і концепцій.

Відновлення середовища

ГІС є ефективним засобом для вивчення незаселеного середовища загалом, окремих видів рослинного та тваринного світу в просторовому й часовому аспектах. Якщо встановлені конкретні параметри навколишнього середовища, необхідні, наприклад для існування якого-небудь виду тварин, зокрема наявність пасовищ і місць для розмноження, відповідні типи й запаси кормових ресурсів, джерела води, вимоги до чистоти природного середовища, то ГІС допоможе швидко підшукати райони з відповідною комбінацією параметрів, у межах яких умови існування або відновлення чисельності цього виду будуть близькі до оптимальних. На стадії пристосовування переселеного виду тварин до нової місцевості, ГІС ефективна для моніторингу найближчих і віддалених наслідків виконаних заходів, оцінки їхньої успішності, виявлення проблем і пошуку шляхів їхнього подолання.

Наукові дослідження та технічна підтримка

Функціональні інтегральні можливості ГІС сприяють успішному проведенню сумісних міждисциплінарних досліджень, вони забезпечують об'єднання та накладання один на одного будь-яких типів даних, лише б їх можна було відображувати на карті. До подібних досліджень належать такі: аналіз взаємозв'язків між здоров'ям населення та різноманітними (природними, демографічними, економічними) чинниками; кількісна оцінка впливу параметрів навколишнього середовища на стан локальних і регіональних екосистем і їхніх складників; визначення прибутків землевласників залежно від переважаючих типів ґрунтів, кліматичних

умов, віддаленості від міст; виявлення чисельності та густини ареалів розповсюдження рідкісних та зникаючих видів рослин залежно від висоти місцевості, куту нахилу й експозиції схилів.

Публікація матеріалів

ГІС значно спрощує процедуру публікації будь-яких видів картографічної продукції. За допомогою вбудованої мови програмного забезпечення (VBA) можна написати програми автоматичного створення будь-яких типів друкарських карт, графіків, діаграм і таблиць. Крім того, програмні продукти дозволяють переглядати й безпосередньо оперувати з даними, що містяться в базі даних ГІС будь-якому, навіть малодосвідченому користувачу.

Екологічна освіта

Оскільки створення паперових карт за допомогою ГІС значно спрощується та здешевлюється, з'являється можливість отримання великої кількості різноманітних природних карт, що розширює можливості й широту охоплення програм і курсів екологічної освіти. Зважаючи на простоту копіювання та виробництва картографічної продукції, її може використовувати практично будь-який вчений, викладач або студент. Крім того, стандартизація формату та компоновання базових карт слугує основою для збору й демонстрації даних, які одержують учні та студенти, обміну даними між навчальними закладами та створення єдиної бази по регіонах і в національному масштабі. Можна підготувати спеціальні карти для землевласників із метою ознайомлення їх із планованими природоохоронними заходами, схемами буферних зон і екологічних коридорів, які створюються в конкретному районі й можуть торкнутися їхніх земельних ділянок.

Екотуризм

Можливість швидкого створення привабливих і водночас, якісних професійно складених карт робить ГІС ідеальним засобом створення рекламних і оглядових матеріалів для залучення публіки в сферу екотуризму, що швидко розвивається. Характерною рисою так званих «екотуристів» є глибока зацікавленість в докладній інформації про природні особливості конкретної місцевості або країни, про процеси, що відбуваються в природі. Серед цієї достатньо численної групи людей

особливо популярними є створені за допомогою ГІС науково-освітні карти, що відображують розповсюдження рослинних співтовариств, окремих видів тварин і птахів, області ендеміків. Подібна інформація може виявитися корисною для цілей екологічної освіти або для туристичних агентств, для отримання додаткових коштів із фондаций проектів і національних програм, що заохочують розвиток подорожей і екскурсій.

Моніторинг

У міру розширення та поглиблення природоохоронних заходів однієї з основних сфер застосування ГІС стає спостереження за наслідками дій на локальному й регіональному рівнях. Моніторинг зон стихійних лих і організація рятувальних і відновлювальних робіт – найважливіші завдання, які оперативно дозволяють виконувати саме ГІС (рис. 3.38). Дані ДЗ оптимально підходять для цих завдань управління в районах надзвичайних ситуацій.

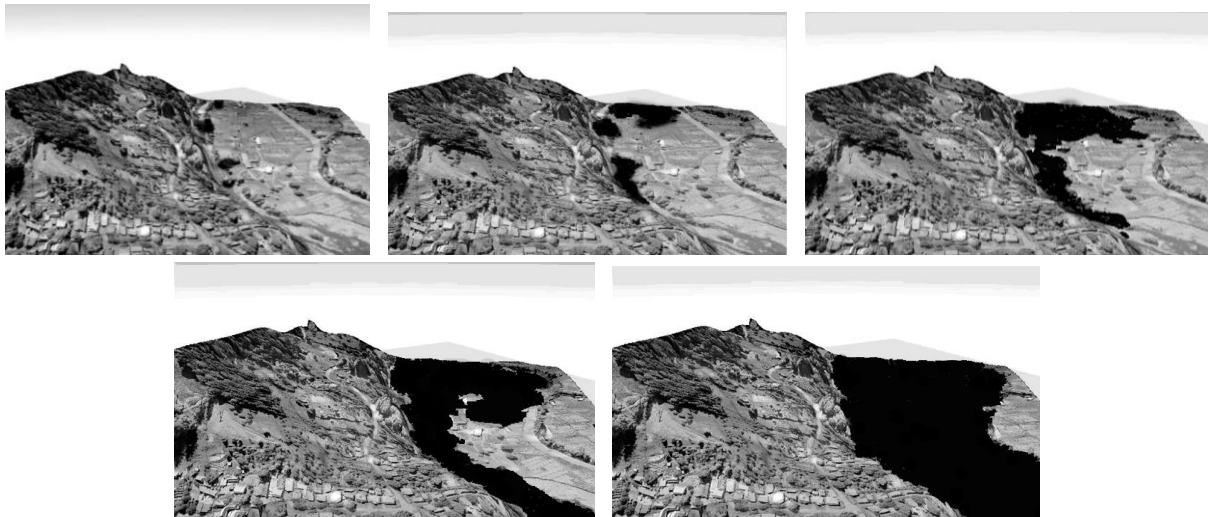


Рисунок 3.38 – Приклад прогнозування повені

Джерелами інформації, що обновлюється, можуть бути результати наземних знімачів або дистанційних спостережень з повітряного транспорту та з космосу. Використання ГІС ефективно й для моніторингу умов життєдіяльності місцевих видів тварин та комах, виявлення причинно-наслідкових ланцюжків і взаємозв'язків, оцінки сприятливих і несприятливих наслідків природоохоронних заходів на екосистему загалом і окремі її компоненти, ухвалення оперативних розв'язань щодо їхнього коригування залежно від змінних зовнішніх умов.

Питання для самоконтролю

1. Де саме в бізнесі використовуються можливості ГІС? Наведіть приклади.
2. Наведіть основні переваги ГІС перед іншими інформаційними технологіями.
3. Як використовують можливості ГІС у різних країнах?
4. Охарактеризуйте перспективи використання ГІС в бізнесі.
5. Що ви розумієте під комплементарністю продуктів різних рівнів у ГІС?
6. Наведіть приклади використання ГІС у геомаркетингу.
7. Охарактеризуйте завдання, які виконує геомаркетинг.
8. Охарактеризуйте основні переваги геомаркетингових ГІС.
9. Наведіть приклади використання геокодованих просторових даних в екологічному моніторингу.
10. Перелічіть особливості ГІС, які використовують в екології.
11. Наведіть приклади, використання ГІС у моніторингу за станом довкілля.
12. Охарактеризуйте роль і місце ГІС у природоохоронних заходах.

РОЗДІЛ 4

ЗАВДАННЯ НА ПРАКТИЧНІ ЗАНЯТТЯ ТА ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

У цьому розділі розглядаються: практичні заняття з дисципліни та кроки виконання індивідуального завдання студента. Подані варіанти завдань, основні вимоги до створюваних баз даних та наведено приклад виконання індивідуального завдання.

4.1 Практичне заняття. Створення та заповнення бази даних

Мета – придбати вміння та навички створення структури таблиць бази даних (БД), створення зв'язків між ними й заповнення таблиць даними.

Призначення – виконавши роботу, можна навчитися:

- створювати базу даних «із нуля»;
- створювати структури таблиць і встановлювати зв'язки між ними;
- вводити та змінювати дані;

Отримані вміння й навички допоможуть виконувати інформаційні завдання у вашій професійній діяльності, створять основу для оволодіння промисловими базами даних, зокрема геоінформаційними.

Призначення додатка. Постановка завдання

Одному з муніципальних відділів поставлено завдання з обліку нерухомості, яка знаходиться на балансі міста.

Потрібно розробити базу даних «*Кадастр*», за допомогою якої буде вестися облік міських будинків, мешканців, нерухомого майна, яким володіють мешканці, та сплати податків. За результатами обліку нерухомого майна працівники муніципальних відділів виконують аналіз змін щодо нерухомості, мешканців та своєчасності сплати податків.

Основні завдання, які будуть розв'язуватися в додатку

Група 1. Завдання обліку міської нерухомості:

- облік міських будинків;
- облік квартир у цих будинках;
- облік мешканців квартир.

Група 2. Завдання обліку нерухомого майна:

- облік нерухомого майна;
- облік сплати податків за нерухоме майно;
- облік боржників.

Група 3. Завдання аналізу змін щодо нерухомості та мешканців:

- облік квартиронаймачів;
- облік правоустановчих документів на квартири;
- динаміка забудови міста.

Група 4. Для забезпечення завдань обліку й аналізу необхідно передбачити виконання допоміжного завдання з ведення такої довідкової інформації:

- за районами міста;
- за вулицями міста;
- за матеріалами, з яких споруджено будинки;
- за змінами податків на нерухомість.

Основні вимоги до бази даних

База даних «*Кадастр*» повинна містити набір даних, який є необхідним для виконання перелічених завдань.

Система керування повинна забезпечувати:

- введення, редагування та перегляд списків районів міста, вулиць міста, матеріалів, із яких споруджено будинки;
- сортування та вибір необхідної інформації за допомогою запитів;
- обчислення кількості будинків по районах та вулицях, а також кількості їхніх мешканців і відображення узагальненої інформації щодо нерухомості за допомогою запитів;
- обчислення податків на нерухоме майно, а також відображення узагальненої інформації щодо податків за допомогою запитів.

Результати всіх запитів мають бути оформлені у вигляді екранних форм для полегшення ведення БД та аналізу результатів господарської діяльності. Для створення повноцінного додатка та полегшення роботи оператора має бути розроблений інтерфейс користувача за допомогою головної кнопкової форми.

ХІД РОБОТИ

Створення порожньої бази даних

Першими в базі даних створюються таблиці. Розглянемо створення таблиць бази даних «*Кадастр*» за допомогою конструктора.

1. Запустіть Microsoft Access за допомогою команди **Пуск – Все Програми – Microsoft Office – Microsoft Access**.

2. На вкладці **Файл Microsoft Access** виберіть команду **Создать** (рис. 4.1).

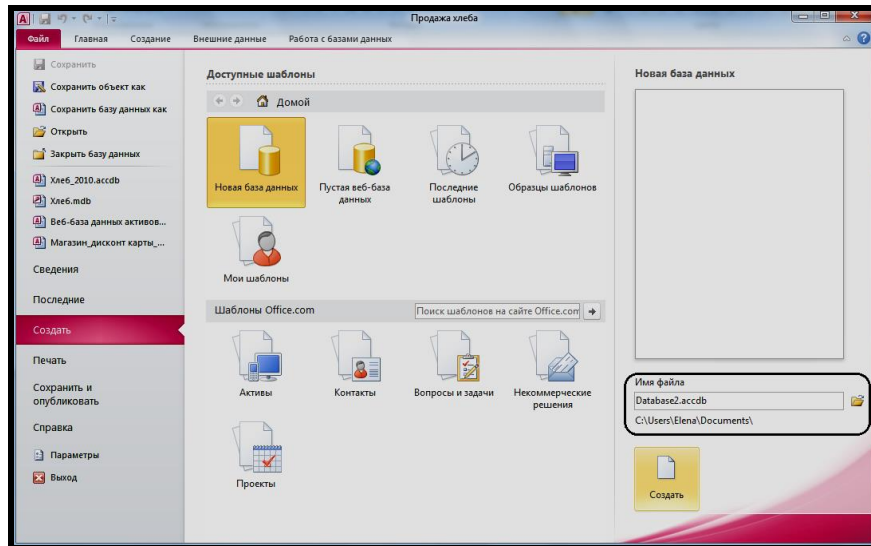


Рисунок 4.1 – Створення нової БД

Виберіть команду **Новая база данных** в області доступних шаблонів.

3. Задайте місце зберігання файла бази даних та її ім'я, для чого:

– знайдіть свою робочу папку, в ній створіть нову папку з ім'ям **БД** і відкрийте її;

– у полі **Имя файла** введіть назву бази даних «*Кадастр*»;

– натисніть кнопку **Создать**.

У результаті виконаних дій з'явиться вікно порожньої бази даних «*Кадастр*».

Таблиці бази даних

Для роботи з таблицею (створення, зміни та заповнення даними) необхідно перейти в режим відображення списку таблиць у робочій області вікна БД, вибравши на панелі **Все объекты Access** тип об'єктів **Таблицы** (рис. 4.2).

Відразу почнемо створювати таблиці. Спочатку будемо створювати батьківські (головні) таблиці, оскільки після задавання структури їх можна заповнювати даними, а потім дочірні (підпорядковані) таблиці. Після

задавання їхньої структури потрібно встановити зв'язки між таблицями, щоб при подальшому заповненні даними забезпечити цілісність БД.

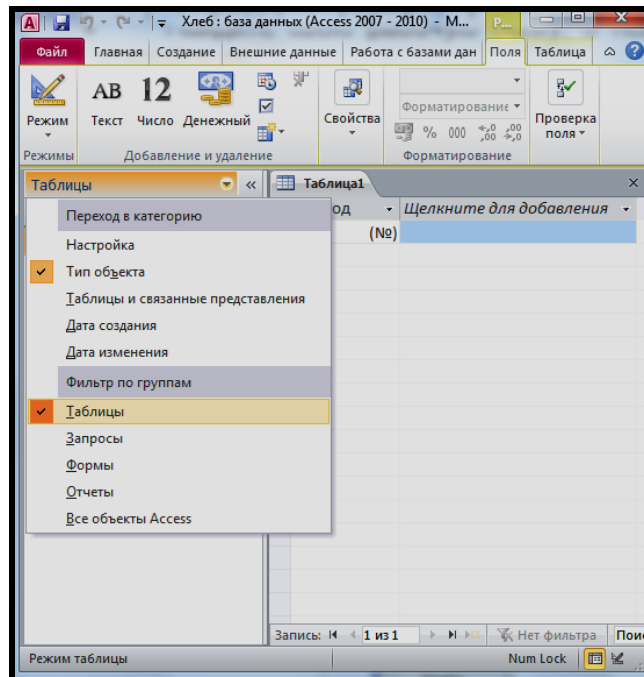


Рисунок 4.2 – Робоча область вікна БД

Отже, спочатку створіть батьківську таблицю **Будівлі**, а потім підпорядковану – **Квартири**. Якщо в БД враховується допоміжна інформація щодо розташування будинків, а також матеріалу, з якого виготовлено будівлі, то створіть також батьківські таблиці **Райони**, **Вулиці** і **Матеріали**.

Таблиця Будівлі

Створення та заповнення таблиці Будівлі

Відповідно до проекту БД «Кадастр» створіть таблицю **Будівлі**, структура якої наведена в таблиці 4.1.

Таблиця 4.1 – Структура таблиці **Будівлі**

Ім'я поля в БД	Тип даних (властивість)
Код_будівлі	Счетчик
Індекс	Короткий текст (10 символів)
Місто	Короткий текст (20 символів)
Код_району	Числовой
Код_вулиці	Числовой

Продовження таблиці 4.1

Номер_будівлі	Короткий текст (8 символів)
Код_матеріалу	Числовой
Фото_будівлі	Поле объекта OLE
Наявність_ліфта	Логический
Кількість_поверхів	Числовой
Номер_ЖЕД	Числовой
Кадастровий_код	Короткий текст (20 символів)
Рік_будівництва	Числовой

Ключ: Код_будівлі

Перехід у режим конструктора

На вкладці **Создание** стрічки MS Access виберіть панель **Таблицы** і клацніть на кнопці **Конструктор таблиц** (рис. 4.3).

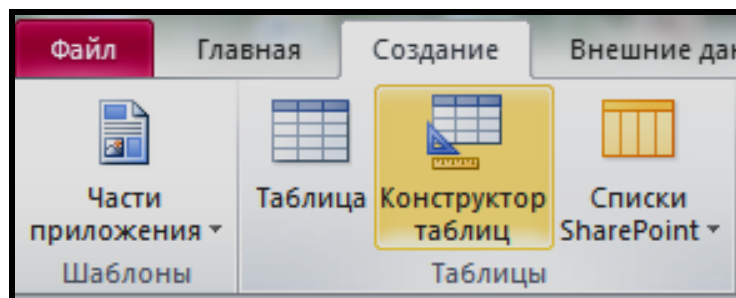


Рисунок 4.3 – Панель **Таблицы**

Створення полів

У першому рядку бланка конструктора таблиць задайте інформацію про поле **Код_будівлі**, для чого:

- У перший стовпець **Имя поля** введіть текст **Код_будівлі**.
- Перейдіть у другий стовпець **Тип данных** і виберіть зі списку, що розкривається, елемент **Счетчик**.

– У другому рядку бланка задайте інформацію про поле **Индекс** (поштовий):

Имя поля: **Индекс**;

Тип данных: **Короткий текст**.

- У третьому рядку бланку задайте інформацію про поле **Місто**.

На вкладці **Общие** конструктора таблиць для поля **Місто** введіть значення

властивості, яка визначає максимальну кількість символів у назві міста:
Размер поля: 20 (рис. 4.4).

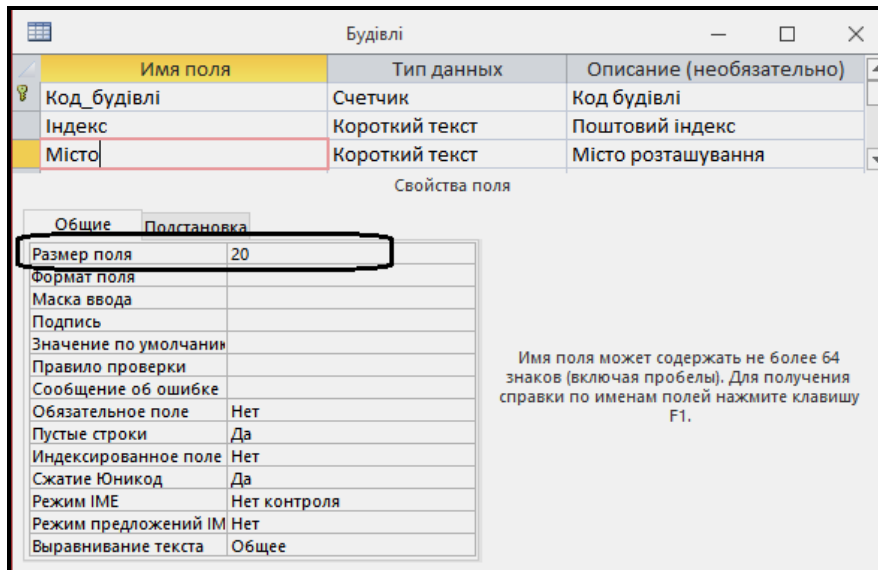


Рисунок 4.4 – Таблица в режиме конструктора

Завдання ключа та збереження таблиці

Задайте первинний ключ таблиці, для чого:

- Клацніть на імені **Код_будівлі**.
- На панелі **Сервис** вкладки **Конструктор** вікна Access клацніть на

кнопці **Ключевое поле** .

-
- Збережіть таблицю на диск, для чого:

- На панелі інструментів клацніть на кнопці **Сохранить** .

– У вікні **Сохранение**, що з'явилось, введіть текст **Будівлі** в полі **Имя таблицы** і клацніть на кнопці **ОК**.

ПРИМІТКА. Зверніть увагу, що після задавання ключового поля його властивість **Индексированное поле** автоматично отримало значення **Да** (**Совпадения не допускаются**).

У результаті виконання вказаних дій створена порожня таблиця **Будівлі**.

Значення за замовчуванням

Для полів **Індекс**, **Код_району**, **Код_вулиці**, **Кількість_поверхів** і **Номер_ЖЕД** задайте порожнє значення властивості **Значение по**

умолчанию в бланку запиту на вкладці *Общие*.

Для поля *Код_материалу* за допомогою будівника виразів задайте значення властивості **Значение по умолчанию** у вигляді виразу, відповідно до якого матеріал, із якого найчастіше споруджуються будинки – залізобетон (код матеріалу 4). Для цього необхідно в правій частині рядка цієї властивості написати необхідний вираз (рис. 4.5).

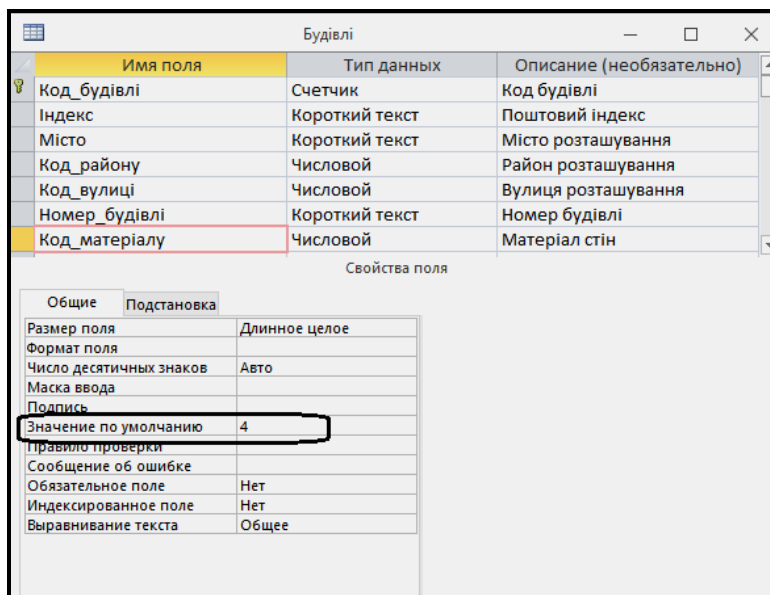



Рисунок 4.5 – Задавання значення за замовчуванням

Введення даних

Щоб заповнити даними таблицю *Будівлі* (табл. 4.2), перейдіть із режиму конструктора в режим таблиці, натиснувши на панелі **Режими** вкладки **Главное** стрічки Access кнопку  **Режим таблицы**.

Таблиця 4.2 – Дані таблиці *Будівлі*

Код_будівлі	1	2	3
1	2	3	4
Індекс	61001	61168	61150
Місто	Харків	Харків	Харків
Код_району	1	1	5
Код_вулиці	1	2	4
Номер_будівлі	5	12	25
Код_матеріалу	1	2	1

Продовження таблиці 4.2

1	2	3	4
Фото_будівлі			
Наявність_ліфта	Так	Так	Ні
Кількість_поверхів	3	12	3
Номер_ЖЕД	2	12	56
Кадастровий_код	12356	13568	14687
Рік_будування	1930	1976	1950

У таблицю, що з'явилася, введіть три записи з даними про будівлі (див. табл. нижче). Після введення кожного значення натискайте клавішу **Tab** або **Enter**. Поле **Код_будівлі** пропускається, оскільки воно має тип **Счетчик** і заповнюється автоматично.

Для додавання фотографії будівлі у поле **Фото_будівлі** потрібно за допомогою контекстного меню у відповідній клітинці запису вибрати команду **Вставити об'єкт**. Потім вибрати перемикач **Создать из файла** і за допомогою кнопки **Обзор** вказати місце розташування фотографії будинку (файли фотографій повинні мати розширення .bmp або .jpg), натиснути кнопку **ОК** (рис. 4.6).

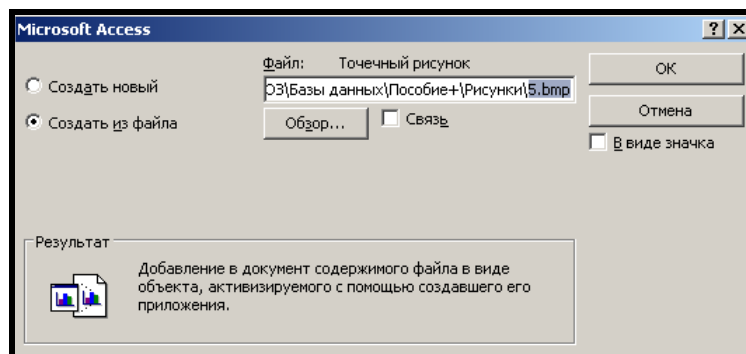


Рисунок 4.6 – Додавання зображень

Закрийте вікно таблиці **Будівлі**, натиснувши відповідну кнопку в його заголовку. Приблизний вигляд таблиці наведено на рисунку 4.7.

Код_будівлі	Индекс	Місто	Код_району	Код_вулиці	Номер_буд	Код_матеріалу	Наявність_ліфту	Кількість_поверхів	Номер_ЖЕД	Кадастровий_код	Рік_будування
1	61001	Харків	1	1	5		<input checked="" type="checkbox"/>	3	2	12356	1930
2	61168	Харків	1	2	12		<input checked="" type="checkbox"/>	12	12	13568	1976
3	61168	Харків	1	3	44		<input checked="" type="checkbox"/>	9	11	13569	1978
4	60001	Харків	5	4	135		<input checked="" type="checkbox"/>	24	158	19847	2010
5	61150	Харків	5	4	25		<input type="checkbox"/>	3	56	14687	1950

Рисунок 4.7 – Приклад заповненої таблиці **Будівлі**

Таблиця Квартири

Створення та заповнення таблиці Квартири

Створіть структуру таблиці *Квартири* (табл. 4.3) відповідно до проекту подібно до того, як ви створювали структуру таблиці *Будівлі*.

Таблиця 4.3 – Структура таблиці *Квартири*

Ім'я поля в БД	Тип даних (властивість)
Код_квартири	Счетчик
Код_будівлі	Числовой
Номер_квартири	Числовой
Поверх	Числовой
Кількість_кімнат	Числовой
Загальна_площа	Числовой
Житлова_площа	Числовой
Наявність_балкона	Логический
Площа_балкона	Числовой
Висота_стін	Числовой
Тип_права	Короткий текст (30 символів)
Особовий_рахунок	Числовой

Ключ: **Код_квартири**. Після її збереження перейдіть у режим таблиці та введіть дані, які подані в таблиці 4.4.

Таблиця 4.4 – Дані таблиці *Квартири*

Код_квартири	1	2	2	4
Код_будівлі	1	1	2	3
Номер_квартири	25	4	391	118
Поверх	1	1	3	4
Кількість_кімнат	2	3	2	3
Загальна_площа	55,8	60	44,7	67
Житлова_площа	38,9	45	25,9	48
Наявність_балкона	Ні	Ні	Так	Так
Площа_балкона	0	0	2,4	5
Висота_стін	3,7	3,7	2,6	2,6
Тип_права	Приватизована	Приватизована	Приватизована	Державна
Особовий_рахунок	112284844	112284800	109680708	254689333

Таблиця Мешканці

Створення структури таблиці Мешканці

Структуру таблиці *Мешканці* (табл. 4.5) створіть відповідно до наступної таблиці.

Таблиця 4.5 – Структура таблиці *Мешканці*

Ім'я поля в БД	Тип даних (властивість)
Код_мешканця	Счетчик
Код_квартири	Числовой (Длинное целое)
Прізвище	Короткий текст (30 символів)
Ім'я	Короткий текст (20 символів)
По_батькові	Короткий текст (20 символів)
Ідентифікаційний_код	Числовой
Квартиронаймач	Логический
Дата_народження	Дата/время
Статус	Короткий текст (20 символів)

Ключ: Код_мешканця

Після визначення полів таблиці, задайте ключове поле та збережіть її структуру на диску під ім'ям *Мешканці*.

Зв'язки між таблицями

Встановлення зв'язків між таблицями

Після створення структури дочірньої таблиці *Квартири* відразу заповнювати її не бажано. Спочатку необхідно набудувати базу даних, щоб MS Access стежив за її цілісністю. Таке налаштування проводиться при встановленні зв'язків між таблицями в такий спосіб.

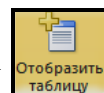
Простежте за тим, щоб були закриті вікна всіх таблиць.

Відкрийте вікно **Схема даних**, натиснувши на вкладці **Работа с базами даних** панелі **Отношения** стрічки Access кнопку **Схема даних**



У вікні **Добавление таблицы**, що з'явилося, виберіть усі таблиці БД і закрийте це вікно. У разі його відсутності клацніть на панелі **Связи**

вкладки **Конструктор** кнопку **Отобразить таблицу**



Встановіть зв'язки між таблицями у вікні **Схема даних**.

З таблиці **Будівлі** перетягніть мишкою поле **Код_будівлі** на однойменне поле в таблиці **Квартири**.

У вікні **Изменение связей** задайте властивості цілісності даних, включивши прапорці:

- **Обеспечение целостности данных;**
- **Каскадное обновление связанных полей;**
- **Каскадное удаление связанных записей,**

а потім натисніть кнопку **ОК** (рис. 4.8).

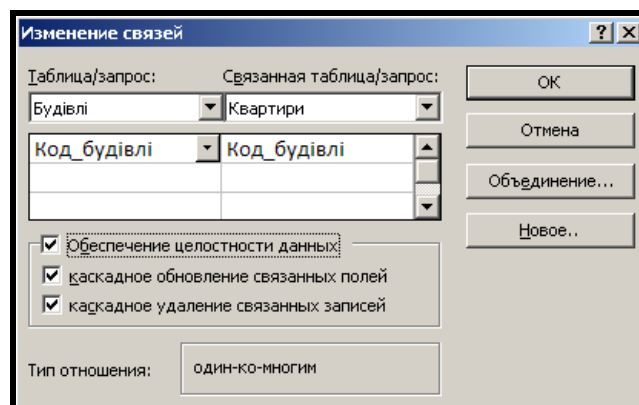


Рисунок 4.8 – Вікно **Изменение связей**

Аналогічно встановіть зв'язок між таблицями **Квартири** та **Мешканці** по полю **Код_квартири**.

Збережіть схему даних на диску, натиснувши на панелі інструментів кнопку **Сохранить**, та закрийте вікно **Схема даних** (рис. 4.9).

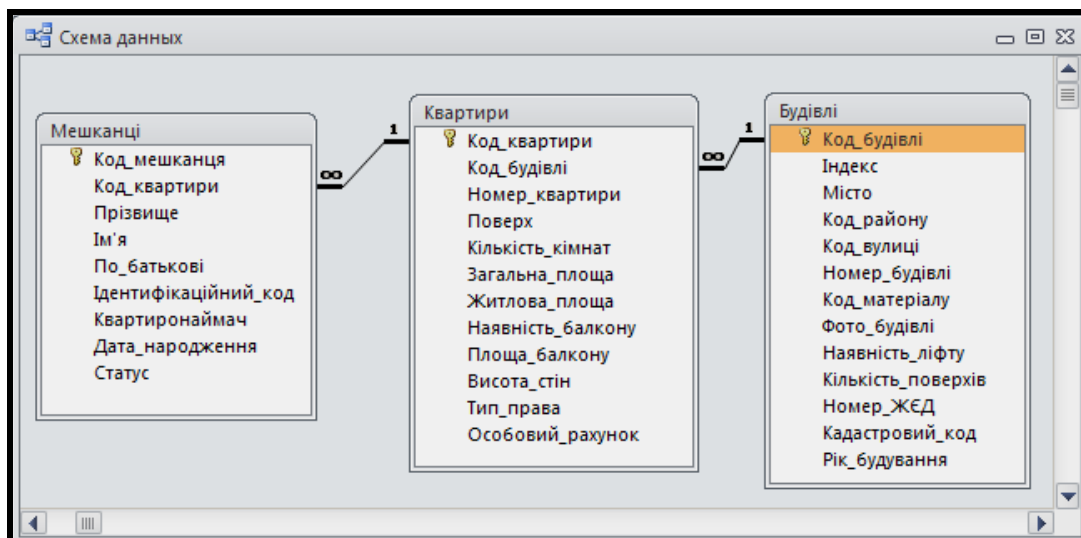


Рисунок 4.9 – Схема даних БД

Заповнення таблиці *Мешканці*

Відкрийте таблицю *Мешканці* в режимі таблиці, виділивши її значок на панелі **Типов об'єктів** Access і виконавши подвійне клацання лівою кнопкою мишки (ЛКМ) по зображенню таблиці. Введіть дані (не менше 20 записів), що показують інформацію про мешканців, які проживають у різних квартирах. Після введення даних закрийте вікно таблиці *Мешканці*.

Завдання для самостійного виконання

1. Заповніть свою БД даними мінімум по 16 квартирам з різною кількістю кімнат, які знаходяться у різних будинках (мінімум чотири будинки) та подайте інформацію про мешканців цих квартири.

2. Доповніть свою БД даними про сплату податків за нерухоме майно мешканцями квартир, щоб вона визначалася схемою даних, поданою на рисунку 4.10. Потім заповніть таблиці *Податки* та *Додаткова нерухомість* даними, які відображають додаткову нерухомість, якою володіють мешканці.

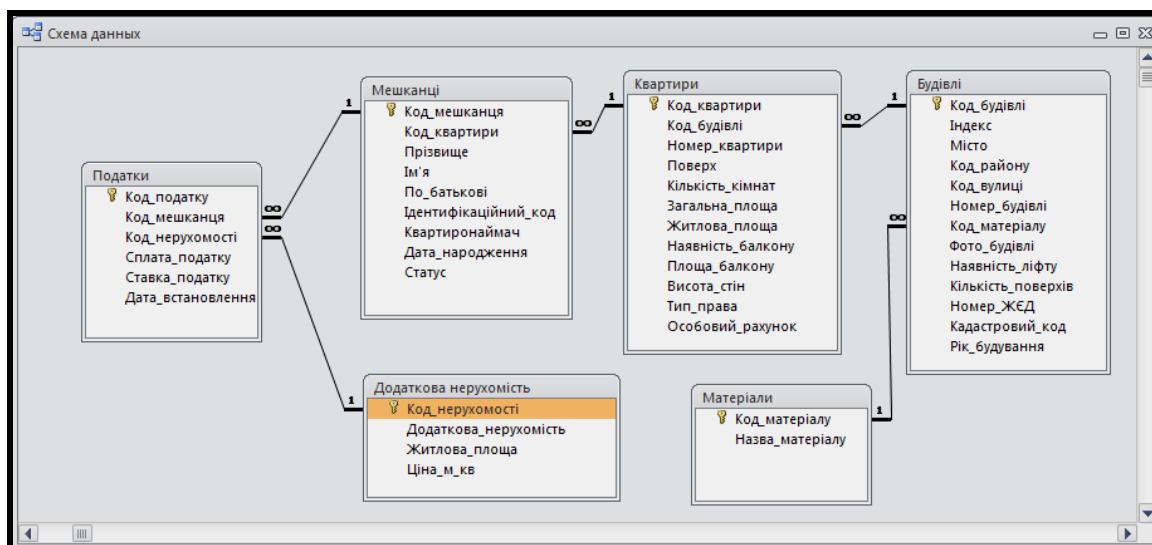


Рисунок 4.10 – Схема бази даних «Кадастр»

3. Доповніть свою БД даними про райони та вулиці, на яких розташовано будівлі, внесіть відповідні зміни до схеми БД.

4.2 Практичне заняття. Операції з даними. Захист від помилок введення

Мета – придбати вміння та навички створення структури таблиць бази даних (БД), створення зв'язків між ними й заповнення таблиць даними.

Призначення – виконавши роботу, можна навчитися:

- сортувати і фільтрувати записи в таблиці;
- захищати табличні дані від помилок введення;


Отримані вміння й навички допоможуть розв'язувати інформаційні завдання у вашій професійній діяльності, створять основу для оволодіння промисловими базами даних, зокрема геоінформаційними.

ХІД РОБОТИ

Сортування записів

Потрібно подати дані таблиці *Мешканці* в алфавітному порядку назв прізвищ. Для цього:

- Відкрийте таблицю *Мешканці*.
- Клацніть на якому-небудь значенні поля *Прізвище*.
- Перейдіть на вкладку **Главное** стрічки Access на панелі

Сортировка и фильтр клацніть на кнопці **Сортировка по возрастанию** .

- Ознайомтеся з отриманими результатами та закрийте вікно таблиці *Мешканці* без збереження змін її структури.

Копіювання таблиць

Оскільки дані таблиці *Мешканці* будуть використовуватися у всіх подальших роботах, бажано мати її копію, а також навчитися відновлювати дані з таблиці-копії в початкову таблицю.

Завдання 1

Створити таблицю *МешканціКопія*, в якій зберігаються ті самі дані, що й у таблиці *Мешканці*.

Виконання

- На панелі типів об'єктів вікна Access виділіть таблицю *Мешканці*, клацнувши на її значку.
- Клацніть на кнопці **Копировать**, а потім – на кнопці **Вставить** на панелі **Буфер обмена** вкладки **Главное** стрічки Access (рис. 4.11).

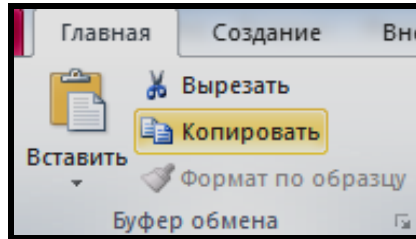


Рисунок 4.11 – Панель **Буфер обмена**

- У вікні, що з'явилося, введіть ім'я нової таблиці *МешканціКопія* (рис. 4.12).

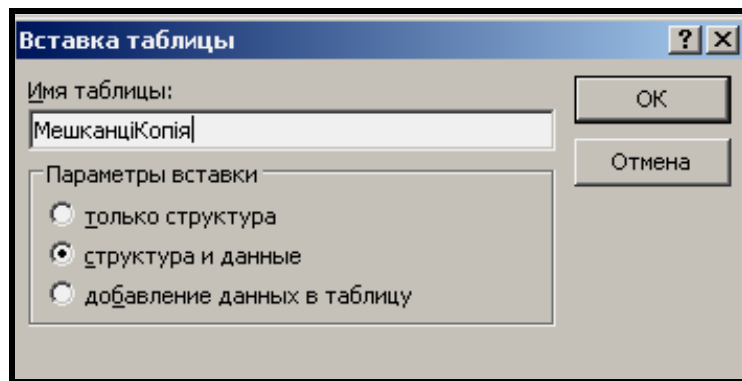


Рисунок 4.12 – Завдання назви таблиці

- Відкрийте таблицю *МешканціКопія* та переконайтеся, що в ній є ті самі записи, що й у таблиці *Мешканці*.
- Закрийте таблицю *МешканціКопія*.

Завдання 2

Видалити всі дані з таблиці *Мешканці*.

Виконання

- Відкрийте таблицю *Мешканці*, двічі клацнувши на її значку.
- Виділіть усі записи в таблиці, наприклад за допомогою комбінації клавіш **Ctrl+A**.
- На клавіатурі натисніть клавішу **Delete**.
- Відповідайте ствердно на запит, чи видаляти записи.

– Закрийте вікно таблиці *Мешканці*.

У результаті виконаних дій у таблиці *Мешканці* не залишиться жодного запису.

Завдання 3

Скопіювати дані з таблиці *МешканціКопія* в таблицю *Мешканці*.

Виконання

– Виділіть таблицю *МешканціКопія*, клацнувши на її значку.

– На панелі інструментів вікна Access клацніть на кнопці **Копировать**, а потім – на кнопці **Вставить**.

– У вікні, що з’явилося, введіть ім’я таблиці *Мешканці* та виберіть перемикач **Добавление данных в таблицу** (рис. 4.13).

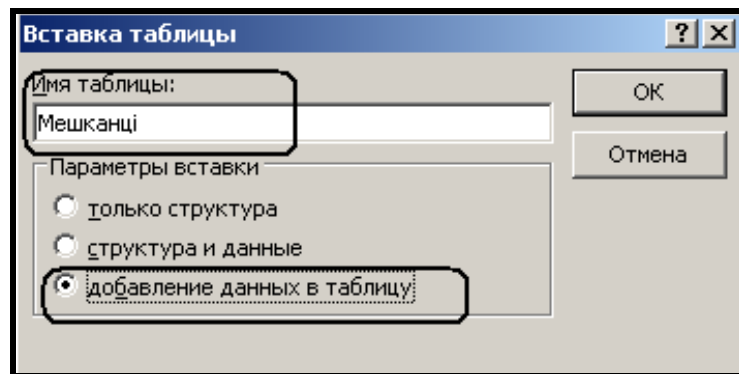


Рисунок 4.13 – Вікно вставки таблиці

– Відкрийте таблицю *Мешканці*, двічі клацнувши на її значку і переконаєтеся, що в ній відновлені первинні дані.

– Закрийте таблицю *Мешканці*.

Захист від помилок введення

Завдання

Встановити засоби захисту на таблиці БД «*Кадастр*», які дозволяють зменшити кількість помилок при введенні даних.

Основні етапи виконання

Розглянемо такі засоби захисту від помилок введення:

– Маска введення.

– Умова на значення.


Маска введення

Завдання 1

Задати маску введення для поля **Індекс** таблиці **Будівлі**.

Виходить з припущення, що будуть вводиться тільки харківські п'ятизначні індекси. Після перших двох цифр записується пробіл як роздільник, наприклад: 61 168.

Виконання

- Відкрийте таблицю **Будівлі** в режимі конструктора.
- Клацніть у рядку поля **Індекс**, а потім у рядку його властивості **Маска ввода** (вкладка **Общие** бланка запиту) і, нарешті, на кнопці  (рис. 4.14).

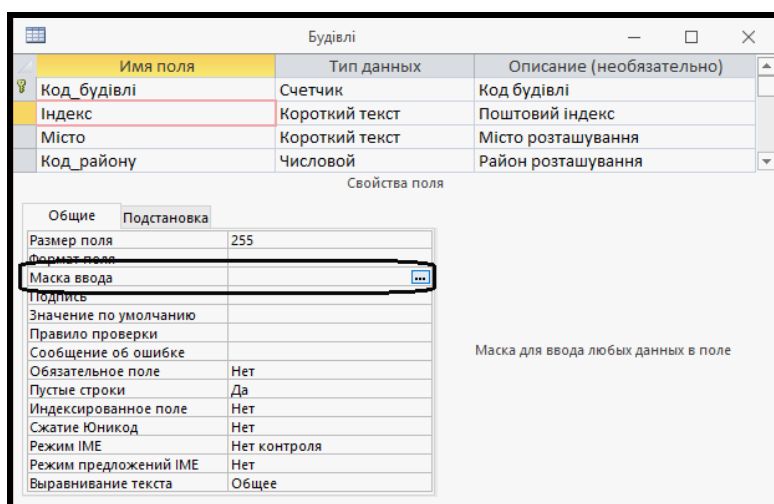


Рисунок 4.14 – Створення маски введення

- У вікні конструктора викличте майстра масок для поля **Індекс**. Клацніть на кнопці **Список**. У вікні **Создание масок ввода**, що з'явилося (рис. 4.15), заповніть поля згідно з даними таблиці 4.6.

Таблиця 4.6 – Структура полів

Поле	Значення
Опис	Індекс
Маска введення	61 000
Заповнювач	пробіл
Зразок даних	61 168
Тип маски	Текст/свободный

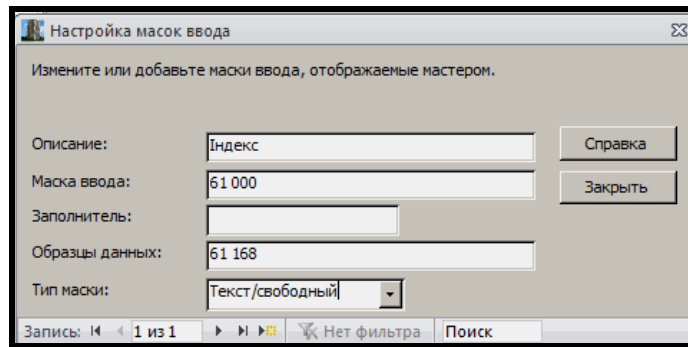


Рисунок 4.15 – Налаштування маски введення

- Клацніть на кнопці **Закреть**.
- Виділіть новий елемент **Индекс** у списку **Маска ввода** та клацніть на кнопці **Далее**.
- У другому вікні майстра можна змінити маску введення. Клацніть на кнопці **Далее**.
- У третьому вікні майстра виберіть перемикач **Вместе со знаками маски** і клацніть на кнопці **Готово**. У властивості **Маска ввода** для поля **Индекс** з'явиться значення "61 "000;0;_
- Перейдіть у режим таблиці та введіть дані по новому будинку.
ПРИМІТКА. При введенні індексу потрібно вводити тільки цифри. Закрийте вікно таблиці **Будівлі**.

Завдання 2

Задати маску введення для поля **Дата_народження** таблиці **Мешканці**. Виходити з припущення, що будуть вводитися дати в короткому форматі.

Виконання

- Відкрийте таблицю **Мешканці** в режимі конструктора.
- Клацніть у рядку поля **Дата_народження**, а потім у рядку його властивості **Маска ввода** і на кінець на кнопці
- У списку **Маска ввода** виберіть елемент **Краткий формат даты**, а потім клацніть на кнопці **Готово**.
- У властивості **Маска ввода** для поля **Дата_народження** з'явиться значення 00.00.0000;0;_.
- Перейдіть у режим таблиці та введіть дані по новому мешканцю.
ПРИМІТКА. При введенні дати потрібно вводити тільки цифри. Закрийте вікно таблиці **Мешканці**.

Завдання для самостійного виконання

У своїй базі даних виконаєте завдання, аналогічні запропонованим:

1. Додайте у таблицю *Мешканці* поле *Мобільний_телефон* і встановіть для нього відповідну маску.
2. Для таблиці *Будівлі* для полів *Код_району* та *Код_вулиці* встановіть значення за замовчуванням.
3. *Переведіть БД «*Кадастр*» на рівень житлово-експлуатаційної ділянки, яка відстежує дані тільки по одному будинку (скопійуйте БД «*Кадастр*» у БД «*Житлова експлуатаційна ділянка*», видаліть непотрібні таблиці та змініть деякі поля).

4.3 Практичне заняття. Операції з даними. Умова на значення.

Використання полів зі списком

Мета – придбати вміння та навички створення структури таблиць бази даних (БД), створення зв'язків між ними та заповнення таблиць даними.

Призначення – виконавши роботу, можна навчитися:

- захищати табличні дані від помилок введення;
- використовувати для зручності введення даних поля зі списком;
- переглядати таблиці разом із зв'язаними даними.

Отримані вміння й навички допоможуть розв'язувати інформаційні завдання у вашій професійній діяльності, створять основу для оволодіння промисловими базами даних, зокрема геоінформаційними.

ХІД РОБОТИ

Умова на значення


Завдання 1

Встановити обмеження на значення поля *Дата_народження* таблиці *Мешканці*.

Передбачається, що рік народження будь-кого з мешканців будинків не може бути менше 1910 року й більше поточного. При спробі ввести рік менше 1910 чи більше поточного необхідно вивести повідомлення «**Рік народження повинен знаходитися між 1910 та поточним**».

Виконання

- Відкрийте таблицю *Мешканці* в режимі конструктора.
- Для поля *Дата_народження* за допомогою будівника виразів задайте значення властивості **Правило перевірки** у вигляді виразу, відповідно до якого мешканці, що проживають у будинках, мають рік народження не раніше 1910 р. і не більше поточного. Для того щоб скористатися будівником виразів для задавання умови на значення, необхідно зробити таке:

- Натисніть кнопку виклику будівника виразів  у правій частині рядка цієї властивості (рис. 4.16).

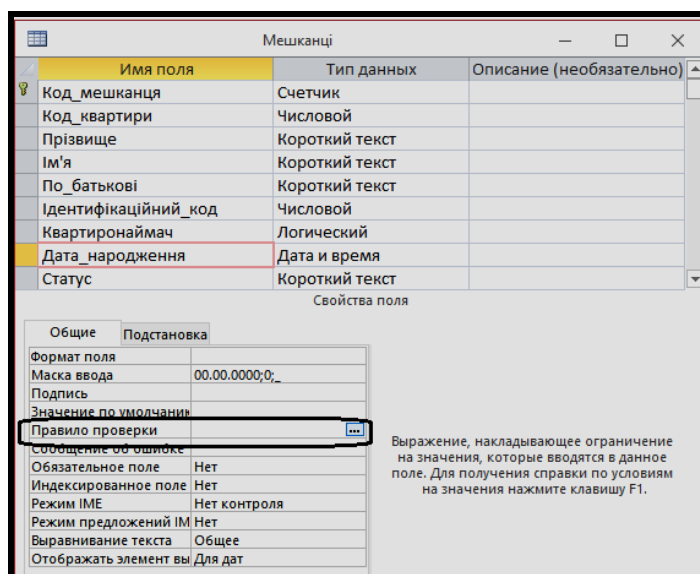


Рисунок 4.16 – Створення умови на значення

– У вікні, що з'явилося, виберіть у лівому списку вид елемента виразу **Операторы**, у середньому списку – **Сравнения**, а у правому списку – **Between**. З'явиться наступний вираз – **Between «Выражение», And «Выражение»**. Далі виділіть праву частину виразу, а саме **«Выражение»** і за допомогою лівого списку та виду об'єкта **Функции – Встроенные функции** виберіть функцію з категорії **Дата/время – Year** і виконайте по ній подвійне клацання мишкою. У результаті сформуєте такий вираз (рис. 4.17).

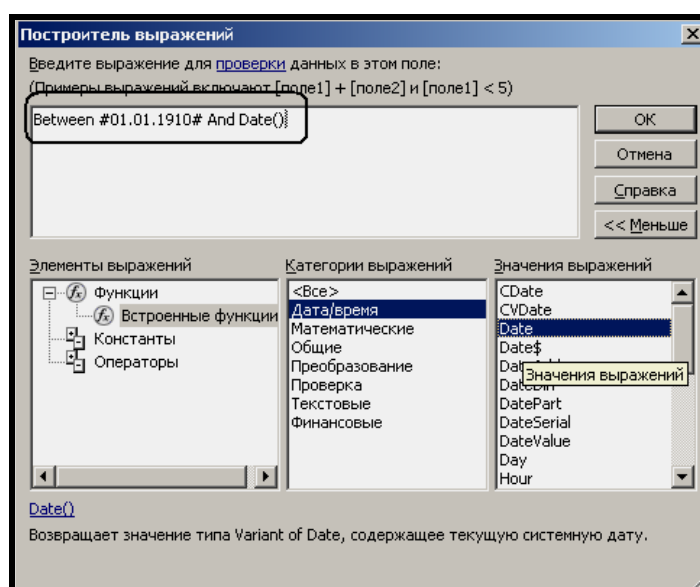


Рисунок 4.17 – Будівник виразів

– Натисніть кнопку **ОК**. Клацніть у рядку властивості **Сообщение об ошибке** і введіть повідомлення: **Рік народження повинен знаходитися між 1910 та поточним**. Перевірте правильність функціонування встановленого обмеження.

– Перейдіть у режим таблиці. У запиті, що з'явився, про перевірку цілісності даних клацніть на кнопці **ДА**.

У режимі таблиці спробуйте додати дані про нових мешканців.

При спробі перейти з поля *Дата_народження* до поля *Статус* з'являється вікно з попередженням: **Рік народження повинен знаходитися між 1910 та поточним** у разі невідповідності року народження встановленому обмеженню. Введіть придатний рік, натисніть клавішу **Enter**. Новий запис буде успішно додано до таблиці. Закрийте вікно таблиці *Мешканці*.


Завдання 2


Встановити обмеження на значення поля *Житлова_площа* таблиці *Квартири*.

Передбачається, що *Житлова_площа* будь-якої квартири у будівлі не може бути більша за загальну площу. При спробі порушити це обмеження необхідно вивести повідомлення «**Житлова площа менша за загальну**».

Виконання

Установка обмеження, в якому використовуються значення інших полів запису, проводиться на рівні таблиці.

– Знаходячись у вікні конструктора таблиці *Квартири*, відкрийте вікно властивостей таблиці. Для цього клацніть на кнопці **Страница свойств**  панелі **Показать или скрыть** вкладки **Конструктор** стрічки Access.

– У вікні властивостей, що відкрилося, клацніть у рядку **Условие на значение** та викличте будівника виразів .

– У вікні, що відкрилося, сформуйте вираз:

[Житлова_площа] < [Загальна_площа]

Для цього використовуйте значення середнього стовпця (рис. 4.18).

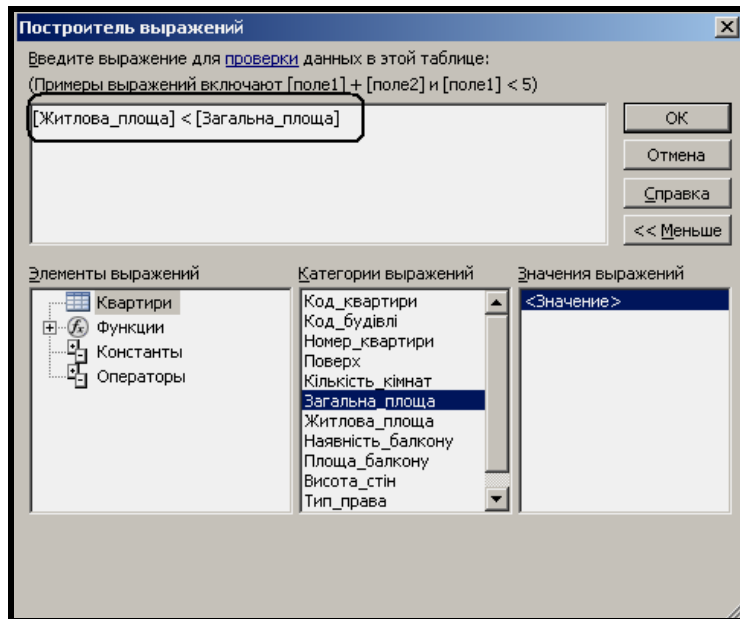



Рисунок 4.18 – Створення виразу

- Клацніть на кнопці **ОК**, щоб закрити вікно будівника виразів.
- Клацніть у рядку властивості **Сообщение об ошибке** вікна властивостей таблиці та введіть текст: **Житлова площа менша за загальну**. Закрийте вікно властивостей таблиці.
- Клацніть на кнопці **Сохранить** , щоб зафіксувати зміну властивостей таблиці. Перевірте правильність функціонування встановленого обмеження.
- Перейдіть у режим таблиці. У запиті, що з'явився, про перевірку цілісності даних клацніть на кнопці **Да**.

У режимі таблиці спробуйте додати дані про нову квартиру. При спробі перейти з поля *Загальна_площа* до поля *Житлова_площа* з'явиться вікно з попередженням «**Житлова площа менша за загальну**», у разі невідповідності введеної площі обмеженню. Введіть додатну площу, натисніть клавішу **Enter**. Новий запис буде успішно добавлено до таблиці. Закрийте вікно таблиці *Квартири*.

Перегляд таблиць і підтаблиць

При роботі з таблицями зручно одночасно переглядати дані з інших взаємопов'язаних таблиць. Для цього використовують підтаблиці.

Завдання

Набудувати вікно таблиці *Квартири* для одночасного перегляду

даних з інших таблиць.

Основні етапи виконання

Вивчення засобів настроювання підтаблиць будемо проводити у три етапи:

1. Перегляд даних про будівлі, у яких розташовані квартири.
2. Одночасне розгортання та згорнення всіх під таблиць.
3. Додавання й видалення підтаблиць.


Перегляд даних про будівлі, у яких розташовані квартири

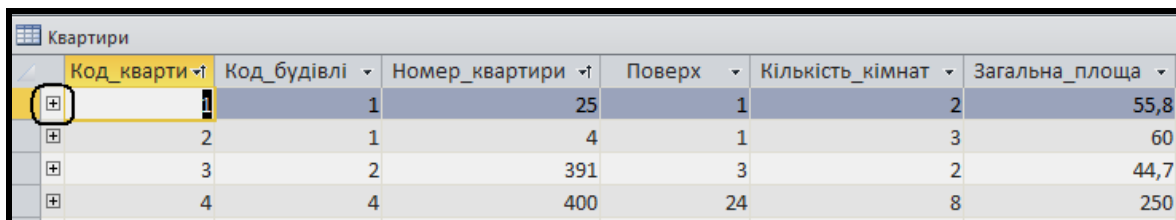
Завдання

Переглянути дані про будівлі.

Виконання


Подвійним клацанням на значку таблиці *Квартири* відкрийте її.

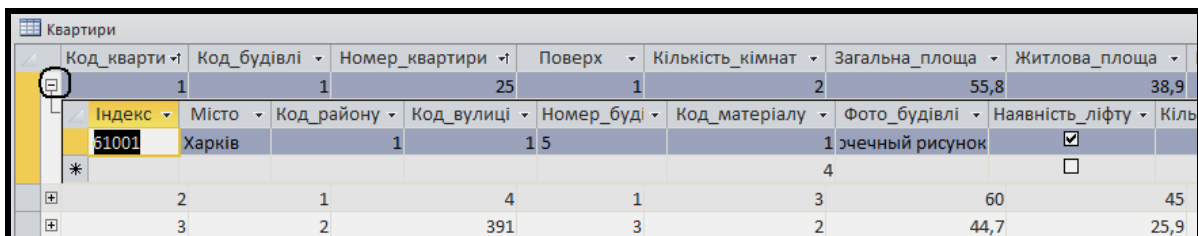
Клацаючи на кнопки , яка розташована зліва від даних, перегляньте дані про будівлі, у яких розташовані квартири (рис. 4.19).



Код_кварту	Код_будівлі	Номер_квартири	Поверх	Кількість_кімнат	Загальна_площа
1	1	25	1	2	55,8
2	1	4	1	3	60
3	2	391	3	2	44,7
4	4	400	24	8	250

Рисунок 4.19 – Таблиця з підтаблицею

Після ознайомлення з даними по кожній квартирі закривайте підтаблицю клацанням на кнопці  (рис. 4.20).



Код_кварту	Код_будівлі	Номер_квартири	Поверх	Кількість_кімнат	Загальна_площа	Житлова_площа		
1	1	25	1	2	55,8	38,9		
Індекс	Місто	Код_району	Код_вулиці	Номер_буді	Код_матеріалу	Фото_будівлі	Наявність_ліфту	Кільк
51001	Харків	1	1	5	1	зречений рисунок	<input checked="" type="checkbox"/>	
*					4		<input type="checkbox"/>	
2	1	4	1	3	60	45		
3	2	391	3	2	44,7	25,9		

Рисунок 4.20 – Розгорнута підтаблиця

Одночасне розгортання та згортання всіх підтаблиць

Завдання

Набудувати вікно таблиці *Квартири* так, щоб відразу були відкриті підтаблиці *Будівлі* по всіх квартирах.

Виконання

Знаходячись у вікні таблиці *Квартири*, виберіть на панелі **Записи** вкладки **Главное** стрічки Access команду **Дополнительно** і в підменю **Подтаблица** виберіть **Развернуть все**. Після цього можна згорнути всі підтаблиці за допомогою команди **Свернуть все** (рис. 4.21).

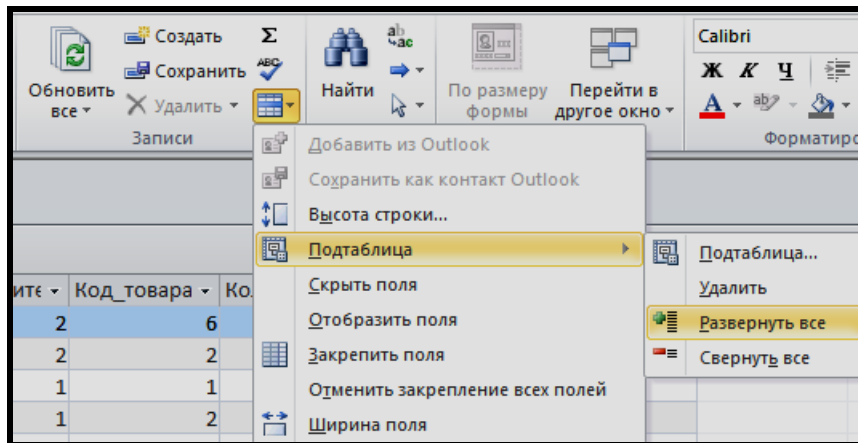


Рисунок 4.21 – Керування підтаблицями

Перегляньте дані про будівлі й закрийте вікно таблиці зі збереженням її макета.

Додавання та видалення підтаблиць

Завдання 1

При перегляді таблиці *Квартири* додати відомості про відповідних мешканців.

Виконання

Подвійним клацанням на значку таблиці *Квартири* відкрийте її. Клацніть по будь-якому запису.

На панелі **Записи** вкладки **Главное** стрічки Access виберіть команду **Дополнительно** і в підменю **Подтаблица** виберіть **Подтаблицы та**

встановіть ім'я підлеглої таблиці й поля зв'язку. Виберіть зі списку ім'я підлеглої таблиці **Мешканці**. У нижній частині вікна в списках **Подчиненные поля** та **Основные поля**, що розкриваються, встановіть значення **Код_квартири**. Клацніть на кнопки **ОК** (рис. 4.22).

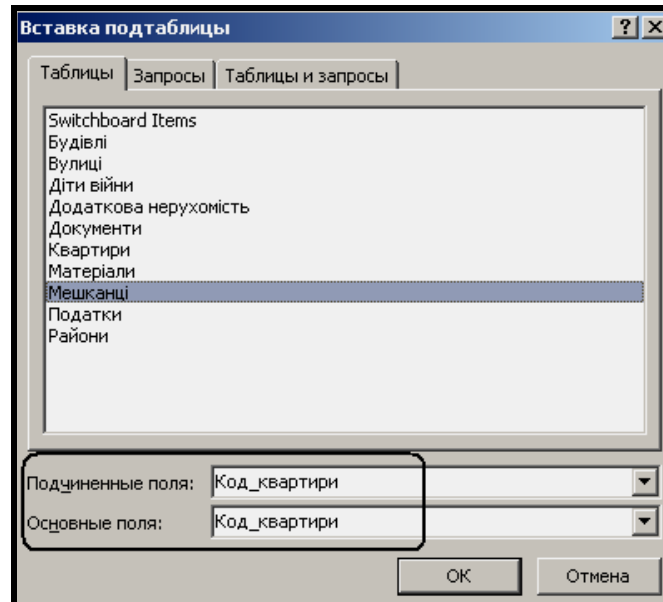


Рисунок 4.22 – Створення підтаблиці

У таблиці **Квартири** зліва від записів з'являться кнопки для розгортання підтаблиць. Клацніть на будь-якій із них. З'явиться підтаблиця **Мешканці** з відомостями про мешканців квартир (рис. 4.23).

Код_квартири	Код_будівлі	Номер_квартири	Поверх	Кількість_кімнат	Загальна_площа	Житлова_площа	Наявність_бал
1	1	25	1	2	55,8	38,9	<input type="checkbox"/>
Код_мешка	Прізвище	Ім'я	По_батьков	Ідентифікаційний_код	Квартиронаймач	Дата_народження	Статус
1	Іванов	Іван	Іванович	4567891245	<input checked="" type="checkbox"/>	03.05.1840	Батько
2	Іванова	Марія	Володимирів	2365487982	<input type="checkbox"/>	08.05.1945	Жінка
3	Іванов	Юрій	Іванович	1236548956	<input type="checkbox"/>	28.05.1975	Син
*	(№)				<input type="checkbox"/>		
2	1	4	1	3	60	45	<input type="checkbox"/>


Рисунок 4.23 – Підтаблиця з даними про мешканців

Перегляньте ще декілька записів і закрийте вікно таблиці **Квартири** зі збереженням її макета.

Завдання 2

При перегляді таблиці **Квартири** видалити дані про мешканців квартир, тобто видалити підтаблицю **Мешканці**.

Виконання

- Подвійним клацанням на значку таблиці **Квартири** відкрийте її.
- Клацніть на кнопці будь-якого запису. Клацніть на будь-якому записі підтаблиці **Мешканці**.
- Виберіть на панелі **Записи** вкладки **Главное** стрічки Access команду **Дополнительно** і в підменю **Подтаблица** виберіть команду **Удалить**. Зліва від записів таблиці **Квартири** зникли кнопки .
- Перегляньте дані про квартири та закрийте вікно таблиці **Квартири** зі збереженням її макета.

Використання полів зі списком

Завдання

У таблицях БД «**Кадастр**» використовувати поля зі списком для спрощення перегляду та введення даних.

У таблиці **Будівлі** наведено інформацію про матеріал, із якого зведено споруду. У полі **Код_матеріалу** зберігаються числові коди матеріалів, що не зовсім зручно. Набагато зручніше буде переглядати назви, а не коди матеріалів.

Для цього створимо таблицю **Матеріали**. Її значення будуть використовуватися в полі зі списком підстановок у таблиці **Будівлі**.

Основні етапи виконання

Завдання будемо виконувати у два етапи:

1. Створення таблиці **Матеріали**.
2. Підстановка значень з таблиці **Матеріали** у таблицю **Будівлі**.

Створення таблиці Матеріали

Завдання

Створимо таблицю **Матеріали**. Щоб показати, з якого матеріалу збудовано кожен дім, у таблиці **Будівлі** поле **Код_матеріалу** використаємо для зв'язку з таблицею **Матеріали** і встановимо для нього поле зі списком підстановки.

Виконання

Створіть таблицю **Матеріали** (табл. 4.7), у якій містяться назви

матеріалів, із яких збудовано будинки. Для цього в режимі конструктора створіть структуру таблиці *Матеріали* відповідно до таблиці 4.7:

Таблиця 4.7 – Структура таблиці *Матеріали*

Ім'я поля в БД	Тип даних (властивість)
Код_матеріалу	Счетчик
Назва_матеріалу	Текстовый (20 символів)

Ключ: Код_матеріалу

Встановіть ключ для поля *Код_матеріалу*. Закрийте вікно таблиці. При цьому введіть її ім'я *Матеріали*.

Відкрийте таблицю *Матеріали* в режимі таблиці та заповніть її такими даними:

- Цегла;
- Панелі;
- Шлакобетон;
- Залізобетон;
- Деревина.

Закрийте вікно таблиці *Матеріали*.

Підстановка значень з таблиці *Матеріали* у таблицю *Будівлі*.

Завдання

Тепер виконайте підстановку значень матеріалів із таблиці *Матеріали* у поле *Код_матеріалу* таблиці *Будівлі*.

Виконання

Відкрийте таблицю *Будівлі* в режимі конструктора. Клацніть на імені поля *Код_матеріалу*. У стовпці властивостей **Тип даних** виберіть елемент **Мастер подстановок**.

– У першому вікні майстра погодитися з вибором перемикача **Объект «Столбец подстановки» ...**

– У другому вікні майстра виберіть зі списку таблицю *Матеріали*.

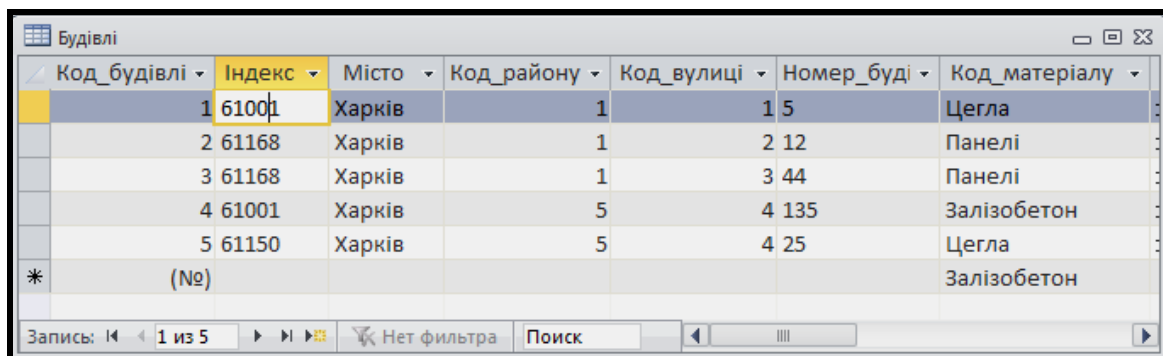
– У третьому вікні майстра перешліть у список **Выбранные поля** обидва поля *Код_матеріалу* і *Назва_матеріалу*.

– У четвертому та п'ятому вікнах майстра клацніть на кнопці **Далее**.

– В останньому вікні майстра клацніть на кнопці **Готово**.

У запиті, що з'явився, про необхідність зберегти таблицю клацніть на кнопці **Да**.

Перейдіть у режим таблиці. Зверніть увагу на те, що замість числових кодів матеріалів тепер у таблиці **Будівлі** фігурують назви матеріалів із таблиці **Матеріали**. Спробуйте змінити для будь-якого запису таблиці назву матеріалу. Зверніть увагу на випадаючий список, який з'являється, як тільки курсор потрапляє у клітинку поля **Код_матеріалу** (рис. 4.24).



Код_будівлі	Индекс	Місто	Код_району	Код_вулиці	Номер_буді	Код_матеріалу
	1	61001	Харків	1	1 5	Цегла
	2	61168	Харків	1	2 12	Панелі
	3	61168	Харків	1	3 44	Панелі
	4	61001	Харків	5	4 135	Залізобетон
	5	61150	Харків	5	4 25	Цегла
*	(№)					Залізобетон

Рисунок 4.24 – Поле зі списком **Код_матеріалу**

Тепер необхідно перевірити, чи правильно майстер підстановок встановив зв'язок між таблицями **Будівлі** і **Матеріали**.

Відкрийте вікно **Схема данных**. У ньому не відображається таблиця **Матеріали**.

Клацніть на панелі інструментів на кнопці **Отобразить таблицу** й у вікні **Добавление таблицы**, що з'явилося, виділіть таблицю **Матеріали**, клацніть на кнопці **Добавить** і закрийте це вікно.

У вікні **Схема данных** з'явилася таблиця **Матеріали**, але на кінцях лінії зв'язку з таблицею **Будівлі** не відображаються символи типу зв'язків.

Двічі клацніть на лінії зв'язку таблиць **Будівлі** та **Матеріали**.

У вікні **Изменение связей**, що з'явилося, встановіть прапорці:

- Обеспечение целостности данных;
- Каскадное обновление связанных полей;
- Каскадное удаление связанных записей.

і клацніть на кнопці **ОК**.

На кінцях лінії зв'язку таблиць *Будівлі* та *Матеріали* з'явилися символи типу зв'язків (рис. 4.25).

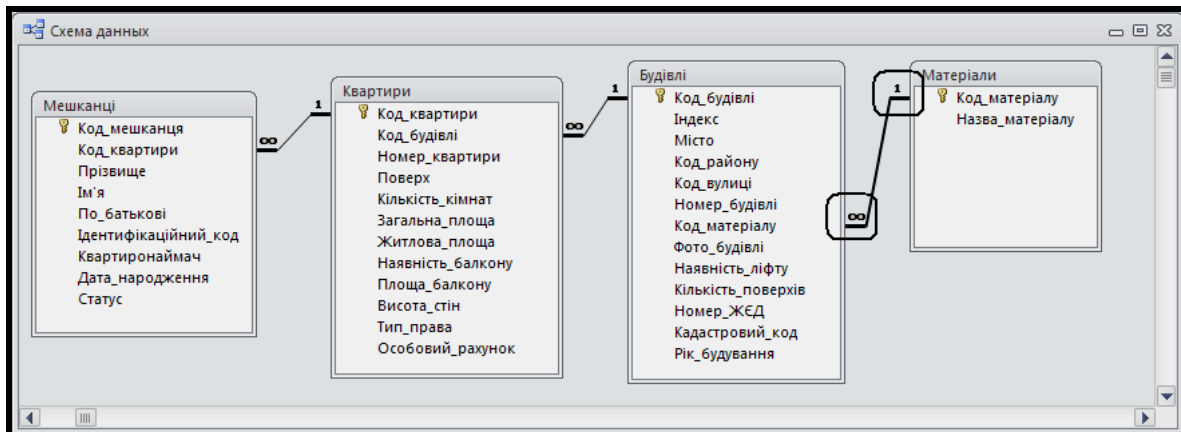


Рисунок 4.25 – Зв'язки між таблицями

Закрийте вікно *Схема данных* зі збереженням.

Список міст у таблиці Будівлі

Завдання

Додати у таблицю *Будівлі* список міст у поле *Місто*.

Виконання

– Відкрийте таблицю *Будівлі* в режимі конструктора. Для поля *Місто* у стовпці властивості **Тип данных** виберіть елемент **Мастер подстановок**.

– У першому вікні майстра виберіть перемикач **Будет введен фиксированный набор значений** (рис. 4.26).

– У другому вікні майстра погодьтеся, що число стовпців **1**, і введіть в поля списку **Столбец1** значення: *Харків, Суми, Київ, Чернігів*, тобто такі, які можуть міститися в полі *Місто* (рис. 4.27).

– В останньому вікні майстра клацніть на кнопці **Готово**.

– У нижній частині вікна конструктора таблиці для властивості **Размер поля** встановіть значення **20** (максимальна довжина найменування міста).

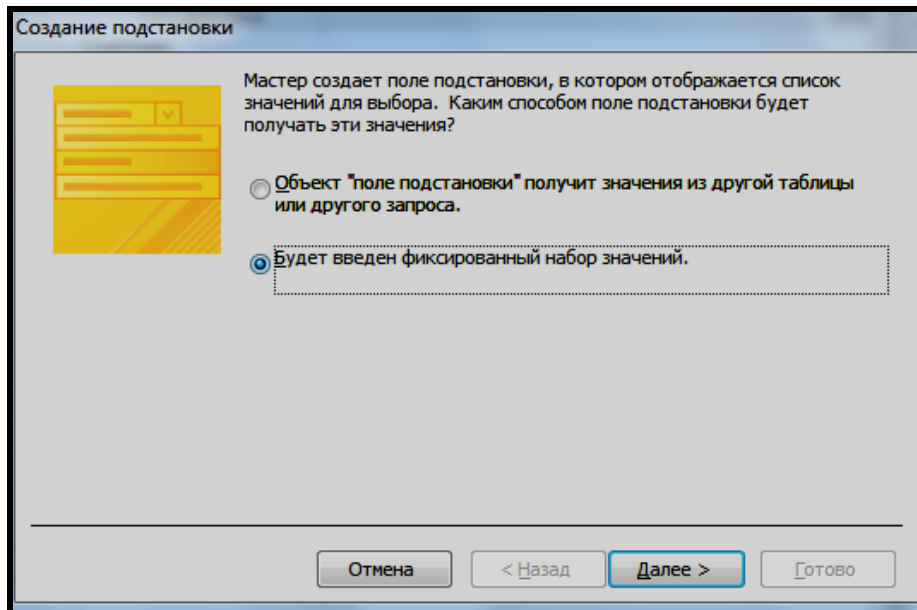


Рисунок 4.26 – Створення списку підстановки

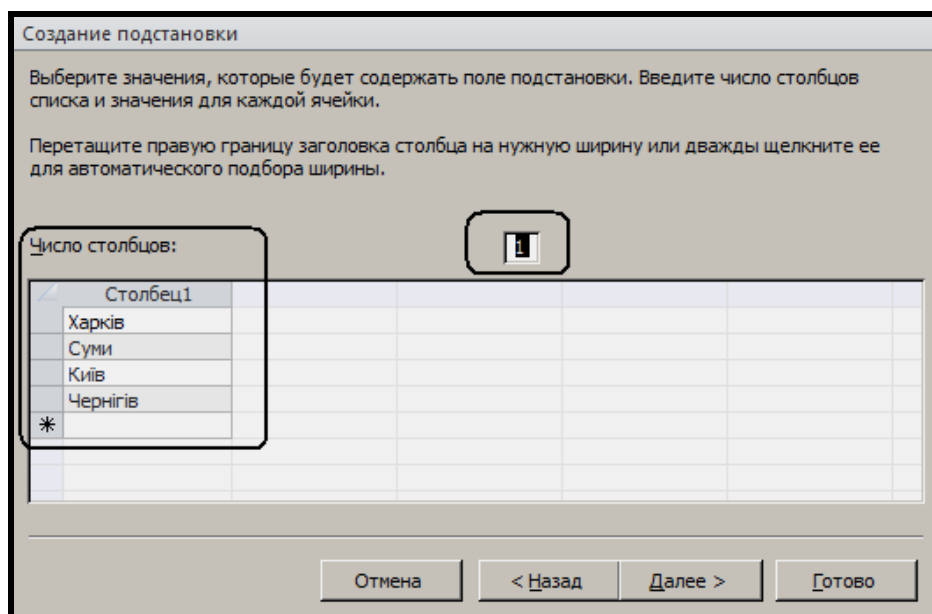


Рисунок 4.27 – Список підстановки

– Клацніть на кнопці **Сохранить**. Перейдіть у режим таблиці. Спробуйте змінити для будь-якого запису таблиці назву міста. Зверніть увагу на випадаючий список, який з'являється, як тільки курсор потрапляє у клітинку поля *Місто* (рис. 4.28).

Код_будівлі	Индекс	Місто	Код_району	Код_вулиці	Номер_буді	Код_матеріалу	Ф
1	61001	Харків	1	1	5	Цегла	зч
2	61168	Харків	1	2	12	Панелі	зч
3	61168	Суми	1	3	44	Панелі	зч
4	61001	Київ	5	4	135	Залізобетон	зч
5	61150	Чернігів	5	4	25	Цегла	зч
*	(№)					Залізобетон	

Рисунок 4.28 – Вибір даних із випадального списку

Закрийте вікно таблиці *Будівлі*.

Завдання для самостійного виконання

У своїй базі даних виконаєте завдання, аналогічні до запропонованих:

1. У таблиці *Будівлі* встановіть список підстановок із таблиць-довідників *Райони* та *Вулиці*.

2. У таблиці *Мешканці* створіть список у полі *Статус*, значення в якому вибираються зі списку: *Батько, Жінка, Донька, Син* та ін.

3. При перегляді таблиці *Квартири* додайте за допомогою підтаблиць відомості про мешканців.

4. *Доповніть свою БД даними про правоустановчі документи на квартиру (створіть таблицю *Документи* з полями: *Вид_договору, Номер_документа, Дата_реєстрації, Номер_у_книзі_записів* та зв'яжіть її з таблицею *Квартири*).

4.4 Практичне заняття. Створення запитів. Запити на вибірку

Мета – набути вмінь і навичок створення запитів на вибірку та зміну даних.

Призначення – виконавши роботу, можна навчитися:

- створювати запити за даними кількох таблиць;
- будувати нові запити на базі створених раніше;

Отримані вміння й навички допоможуть вирішувати інформаційні завдання у вашій професійній діяльності, створять основу для оволодіння промисловими базами даних, зокрема геоінформаційними.

ХІД РОБОТИ

Детальний запит на вибірку

Завдання

Подати дані про житловий фонд міста, тобто про будинки та квартири на території районів, з обчисленням податків у разі перевищення норм по житловій площі.

Основні етапи виконання

Виконання завдання складається з п'яти етапів:

1. Створення запиту *Квартири_Дані*.
2. Додавання обчислюваних полів.
3. Копіювання запиту.
4. Налаштування запиту *Кількість_кімнат*.
5. Перетворення запиту *Кількість_кімнат* у параметричний.

ПРИМІТКА. Побудований на першому етапі запит *Квартири_Дані* використовуватимемо також для виконання інших завдань. Тому якщо потрібно було б виконувати тільки завдання, що розглядається в цьому розділі, то можна на першому етапі відразу створити запит з ім'ям *Квартири_Податок* і третій етап пропустити.

Створення запиту *Квартири_Дані*

Завдання

За допомогою майстра створити запит на вибірку даних із таблиць

Квартири та Будівлі і пов'язаних із ними довідників (рис. 4.29).

Назва_району	тип_вулиці	Назва_вулиці	Номер_буді	Номер_квартири	Кількість_кімнат	Поверх	Загальна_площа	Житлова_площа
Дзержинський	проспект	Леніна	25	15	1	2	30	28
Дзержинський	проспект	Леніна	135	400	8	24	250	180
Київський	вулиця	Блохера	44	118	3	4	67	48
Київський	вулиця	Героїв праці	12	392	4	3	90,2	67
Київський	вулиця	Героїв праці	12	80	4	1	90,2	67
Київський	вулиця	Героїв праці	12	391	2	3	44,7	25,9
Київський	вулиця	Пушкінська	5	4	3	1	60	45
Київський	вулиця	Пушкінська	5	25	2	1	55,8	38,9

Рисунок 4.29 – Запит *Квартири_Дані*

Виконання

Відкрийте базу даних «Кадастр», яка залишилася після виконання лабораторної роботи № 1.

На вкладці **Создание** у групі **Запросы** натисніть кнопку **Мастер запросов**.

У вікні **Новый запрос** виберіть зі списку елемент **Простой запрос**, який викликає майстра **Создание простых запросов** і натисніть кнопку **ОК**.

У першому вікні майстра зі списку **Таблицы и запросы**, що розкривається, виберіть по черзі таблиці **Будівлі**, **Квартири**, **Райони**, **Вулиці**. Зі списку **Доступные поля** перешліть у список **Выбранные поля** потрібні поля з цих таблиць за допомогою кнопок зі стрілками (рис. 4.30). Потім натисніть кнопку **Далее**.

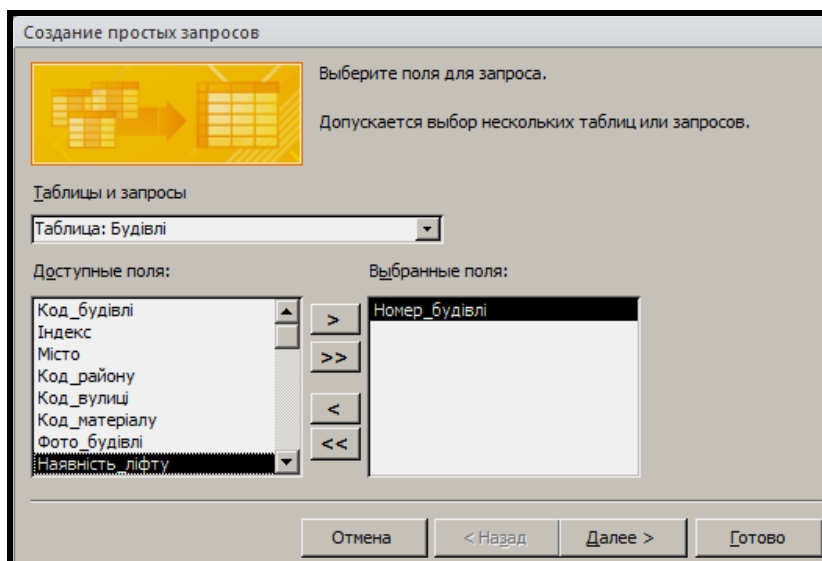


Рисунок 4.30 – Вибір полів для запиту

Послідовність вибору таблиць і пересилання полів із них наведено в

таблиці 4.8.

Таблиця 4.8 – Поля для запиту *Квартири_Дані*

Таблиця	Поле
Райони	Назва_району
Вулиці	Тип_вулиці
Вулиці	Назва_вулиці
Будівлі	Номер_будівлі
Квартири	Номер_квартири
Квартири	Кількість_кімнат
Квартири	Поверх
Квартири	Загальна_площа
Квартири	Житлова_площа
Квартири	Наявність_балкона
Квартири	Площа_балкона
Квартири	Висота_стін
Квартири	Тип_права

У другому вікні майстра підтвердіть вид запиту **Подробный**, натиснувши кнопку **Далее**.

В останньому вікні майстра введіть ім'я запиту *Квартири_Дані* і натисніть кнопку **Готово**.

Перегляньте результат виконання запиту та збережіть його, натиснувши на панелі швидкого доступу кнопку **Сохранить**.

Додавання обчислюваних полів

Завдання

У запиті *Квартири_Дані* додати поля, щоб виводилася нежитлова площа квартири (рис. 4.31). Ці величини обчислюються за такими формулами:

$$\text{Нежитлова_Площа} = \text{Загальна_площа} - \text{Житлова_площа}$$

The screenshot shows a table with the following data:

Назва_вулиці	Номер_буді	Номер_квартири	Кількість_кімнат	Поверх	Загальна_площа	Житлова_площа	Нежитлова_Площа
Леніна	25	15	1	2	30,00	28,00	2
Леніна	135	400	8	24	250,00	180,00	70
Блюхера	44	118	3	4	67,00	48,00	19


Рисунок 4.31 – Обчислюване поле *Нежитлова_Площа* у запиті *Квартири_Дані*

Виконання

Для реалізації формули, за якою обчислюється нежитлова площа, виконайте такі дії:

– Відкрийте вікно запиту *Квартири_Дані* в режимі конструктора, натиснувши кнопку **Режим**  на вкладці **Главная** в групі **Режимы**.

– Клацніть у рядку **Поле** першого вільного стовпця бланка запиту.

– Викличте будівника виразів, натиснувши кнопку **Построитель**  на контекстній вкладці **Конструктор** у групі **Настройка запроса**.

– У вікні будівника виразів у середньому стовпці двічі клацніть на полі *Загальна_площа*, щоб воно з'явилося у верхній частині вікна. Потім введіть знак «мінус». Після цього в середньому стовпці двічі клацніть на полі *Житлова_площа*. Закінчіть введення виразу натисканням кнопки **ОК**. Вікно будівника виразів подано на рисунку 4.32.

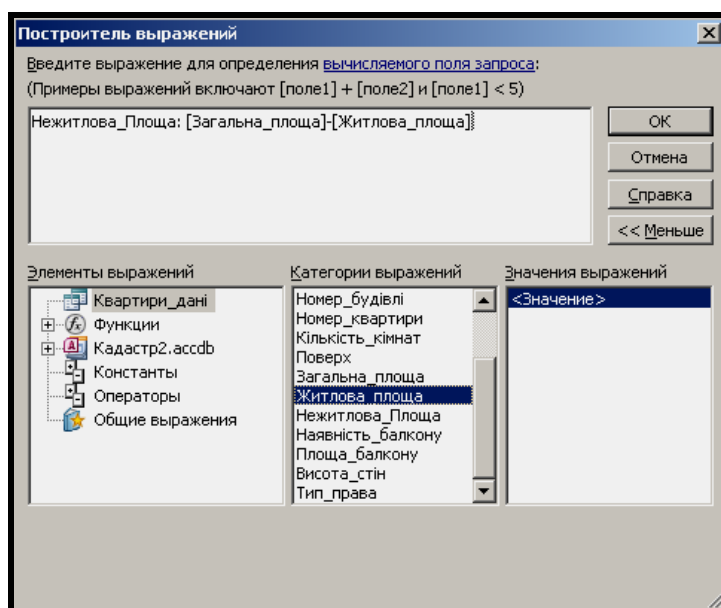





Рисунок 4.32 – Побудова виразу для нежитлової площі

– Перегляньте результат виконання запиту, натиснувши кнопку **Режим**  на вкладці **Главная** в групі **Результаты**, і поверніться в режим конструктора (кнопка ).

– Перед введеним виразом з'явиться слово *Выражение1* і двокрапка. Видаліть це слово, залишивши символ «:», а замість попереднього слова введіть слова *Нежитлова_Площа*.

– Відкрийте вікно властивостей поля, натиснувши кнопку

Страница свойств  у группе **Показать или скрыть** на контекстной вкладке **Конструктор**. У вікні, що з'явилося, установіть значення **С** разделителями разрядов для властивості **Формат поля** (рис. 4.33).

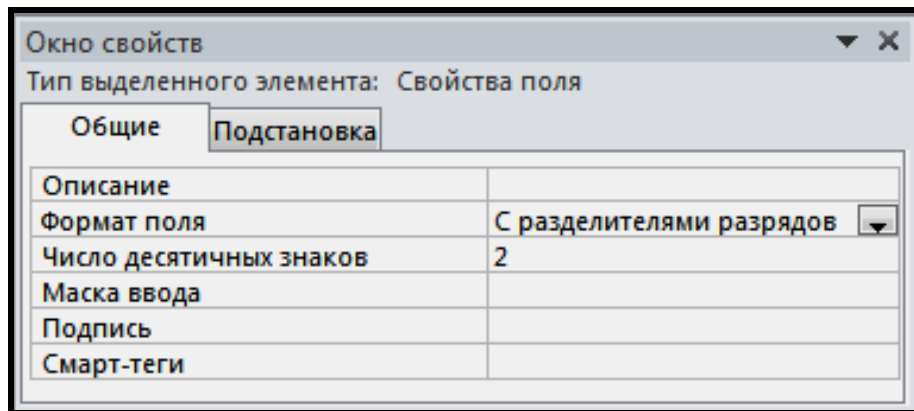


Рисунок 4.33 – Значення **С** разделителями разрядов властивості

Формат поля

Збережіть зміни в запиті та перегляньте результат його виконання в режимі таблиці.

Подібно до щойно вирішеного завдання створіть поле для обчислення загальної площі квартири. Виходять із припущення, що необхідно підсумувати площі всіх кімнат квартири, а також площі лоджій і балконів. Для балконів використовується знижувальний коефіцієнт – 0,3. Перегляньте результат виконання запиту й закрийте його зі збереженням змін.

Копіювання запиту


Завдання


Скопіювати запит **Квартири_Дані**, оскільки новий запит **Кількість_кімнат** за структурою повністю співпадає з початковим **Квартири_Дані** (рис. 4.34). У ньому потрібно змінити деякі властивості.

Назва_району	тип_вулиці	Назва_вулиці	Номер_буді	Номер_квартири	Кількість_кімнат	Поверх	Загальна_площа
Дзержинський	проспект	Леніна	25	15	1	2	30,00
Дзержинський	проспект	Леніна	135	400	8	24	250,00
Київський	вулиця	Блюхера	44	118	3	4	67,00
Київський	вулиця	Героїв праці	12	392	4	3	90,20
Київський	вулиця	Героїв праці	12	80	4	1	90,20
Київський	вулиця	Героїв праці	12	391	2	3	44,70
Київський	вулиця	Пушкінська	5	4	3	1	60,00

Рисунок 4.34 – Запит **Квартири_Дані**

Виконання

– Виділіть ім'я запиту *Квартири_Дані*, клацнувши на ньому в області переходів, і натисніть кнопку **Копировать** , що розташована в групі **Буфер обмена** на вкладці **Главная**.

– Натисніть кнопку **Вставить** , що розташована в групі **Буфер обмена** на вкладці **Главная**.

– Введіть ім'я запиту *Квартири_Дані* у вікні **Вставка**, що з'явилося, і натисніть кнопку **ОК**.

– Відкрийте запит *Квартири_Дані*, двічі клацнувши на його імені в області переходів. Перегляньте його дані, впевнившись, що відображаються дані про квартири будинків, потім закрийте запит.

Налаштування запиту *Кількість_кімнат*

Завдання

Налаштувати запит *Кількість_кімнат* так, щоб у ньому відображалися дані про розташування квартири та кількість кімнат у ній, до того ж дані повинні виводитися в алфавітному порядку найменувань районів та назв вулиць без відображення кодів (рис. 4.35).



Назва_району	тип_вулиці	Назва_вулиці	Номер_буді	Номер_квартири	Поверх	Кількість_кімнат	Наявність_балкону	Тип_права
Київський	вулиця	Пушкінська	5	25	1	2	<input type="checkbox"/>	Приватизована
Київський	вулиця	Пушкінська	5	4	1	3	<input type="checkbox"/>	Приватизована
Київський	вулиця	Героїв праці	12	391	3	2	<input checked="" type="checkbox"/>	Приватизована
Київський	вулиця	Героїв праці	12	80	1	4	<input checked="" type="checkbox"/>	Державна
Київський	вулиця	Героїв праці	12	392	3	4	<input checked="" type="checkbox"/>	Приватизована
Київський	вулиця	Блюхера	44	118	4	3	<input checked="" type="checkbox"/>	Приватизована
Дзержинський	проспект	Леніна	135	400	24	8	<input checked="" type="checkbox"/>	Приватизована
Дзержинський	проспект	Леніна	25	15	2	1	<input checked="" type="checkbox"/>	Приватизована

Рисунок 4.35 – Запит *Кількість_кімнат* у режимі таблиці


Виконання

– Відкрийте запит *Кількість_кімнат* у режимі конструктора.

– Щоб не виводилися поля *Загальна_площа*, *Житлова_площа*, *Нежитлова_Площа*, *Площа_балкону* і *Висота_стін*, клацанням миші послідовно приберіть прапорці цих полів у рядку **Вивод на екран**.

– Щоб назви районів виводилися в алфавітному порядку, поверніться в режим конструктора запиту та клацніть у рядку **Сортировка** поля *Назва_району*. Зі списку, що розкривається, виберіть елемент **По возрастанию**. Після цього знову перегляньте результат виконання запиту.

– Те саме (сортування в алфавітному порядку) зробіть і для поля **Назва_вулиці**.

– У рядку **Условие отбора** поля **Кількість_кімнат** введіть цифру 2 для відбору двокімнатних квартир. Перейдіть у режим відображення таблиці, натиснувши кнопку **Режим**  на вкладці **Главная** в групі **Результаты**. Переконайтеся, що відображаються тільки дані про двокімнатні квартири.

У результаті виконаних дій, отримаємо запит, що подано на рисунку 4.36.

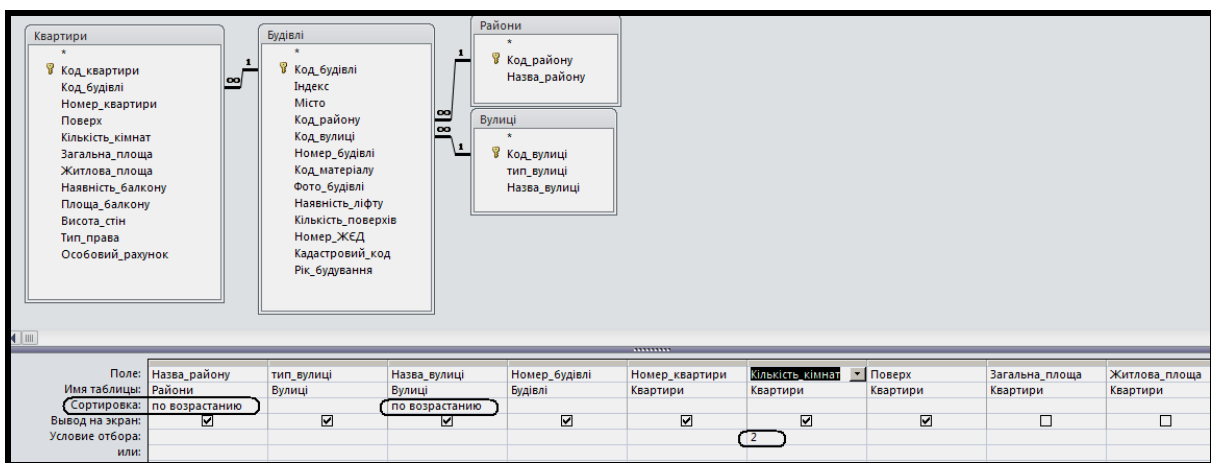


Рисунок 4.36 – Запит **Квартиры_Дані** в режимі конструктора

Перегляньте результат виконання запиту, перейшовши в режим таблиці, і натисніть кнопку **Сохранить**.

Перетворення запиту в параметричний

Завдання

Створити параметричний запит для відбору квартир за вибраною кількістю кімнат (рис. 4.37). Це позбавить від необхідності постійно перемикатися з режиму таблиці в режим конструктора та назад для введення нової кількості квартир.

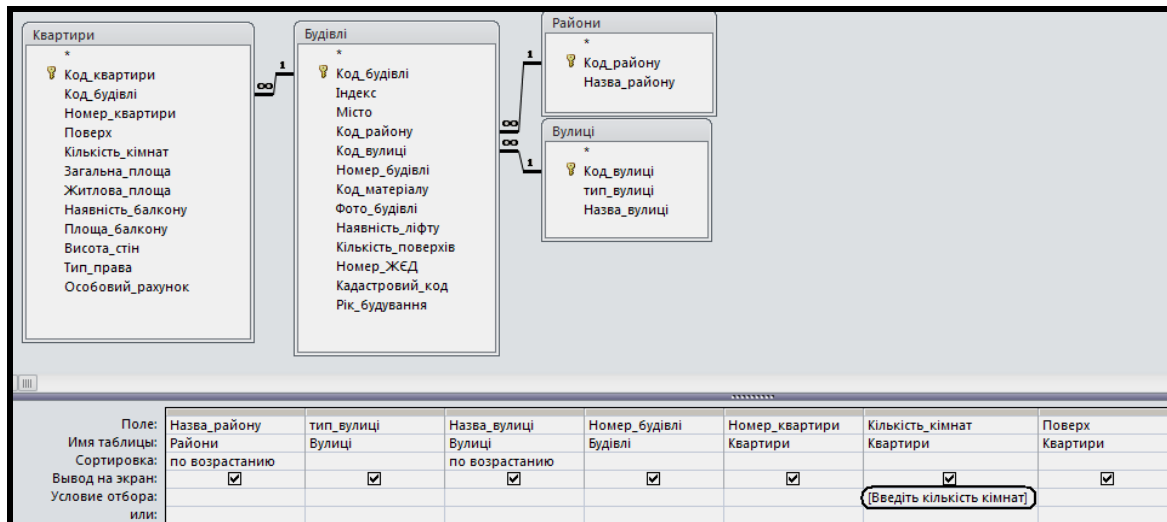


Рисунок 4.37 – Параметр у запиті

Виконання

У режимі конструктора запиту *Квартири_Дані* клацніть у рядку **Условие отбора** поля *Кількість_кімнат* і замість конкретного значення введіть у квадратних дужках текст підказки, наприклад [*Введіть кількість кімнат*].

ПРИМІТКА. Якщо закрити вікно запиту, а потім відкрити його в режимі конструктора, то поля, в яких знято прапорець **Вывод на экран**, але встановлено сортування або умови відбору, відображаються після полів, у яких є цей прапорець. У разі, коли не встановлено сортування або умови відбору, такі поля зі знятим прапорцем **Вывод на экран** у режимі конструктора можуть не відображатися зовсім.

Збережіть запит і закрийте його.

Відкрийте запит *Квартири_Дані*, двічі клацнувши на його значку в області переходів. У діалоговому вікні, що з'явилося, введіть необхідну кількість кімнат (наприклад, 2). Перегляньте результат виконання запиту та закрийте його вікно. Під час подальших відкриттів запиту можна вводити інші кількості кімнат у квартирах.

Дані про мешканців квартир

Завдання

Міському муніципальному відділу по субсидіях необхідні докладні дані про мешканців квартир для розрахунку субсидій. Визначити пільговий контингент мешканців.

Оскільки запит, із якого будуть взяті поля для запиту *Мешканці_квартир*, буде використовуватися як базовий в інших об'єктах, то побудуємо цей запит окремо та назвемо його *Дані_про_нерухомість*.

Отже, завдання виконується в три етапи:


1. Створення та налаштування запиту *Дані_про_нерухомість*.
2. Створення та налаштування запиту *Мешканці_квартир*.
3. Створення та налаштування запиту *Квартиронаймач*.

Запит Дані_про_нерухомість

Завдання

На основі запитів *Квартири_дані* та *Дані_про_нерухомість* створити новий запит *Мешканці_квартир*.

Виконання

– На стрічці Access перейдіть на вкладку **Создание**, на панелі **Запросы** клацніть на кнопці **Конструктор Запросов** .

– У вікні **Добавление таблицы**, що з'явилося, натиснувши клавішу **Ctrl**, виберіть зі списку такі таблиці: *Будівлі*, *Квартири*, *Вулиці*, *Мешканці* (рис. 4.38). Натисніть на кнопку **Добавить**.

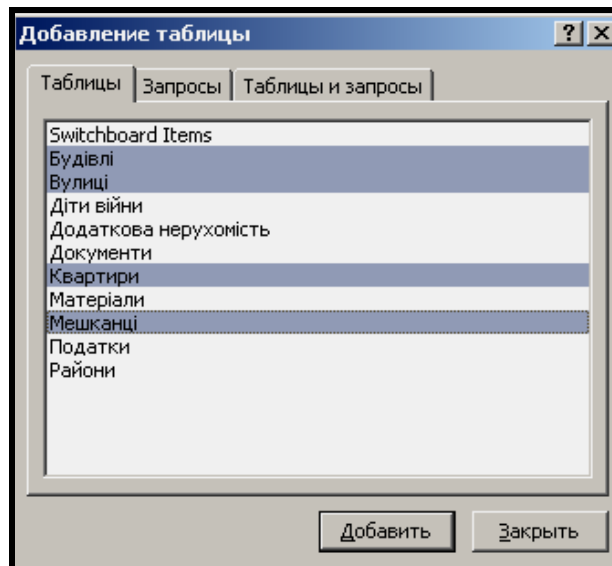


Рисунок 4.38 – Параметр у запиті

– Закрийте вікно **Добавление таблицы**, клацнувши на кнопці **Закреть**.

Додайте до запиту поля з таблиць перетаскуванням ЛКМ (лівою кнопкою миші) з верхньої частини вікна конструктора запитів, яка призначена для відображення таблиць, що беруть участь у запиті, до нижньої частини вікна бланка запиту, відповідно до наведеної нижче таблиці (табл. 4.9).

Таблиця 4.9 – Поля для запиту *Дані_про_нерухомість*


Таблиця	Поле
Мешканці	Прізвище
Мешканці	Ім'я
Мешканці	По_батькові
Мешканці	Дата_народження
Мешканці	Квартиронаймач
Вулиці	Назва_вулиці
Будівлі	Номер_будівлі
Квартири	Номер_квартири

Запит «Мешканці_квартир»

Завдання

На основі запитів *Квартири_дані* та *Дані_про_нерухомість* створити новий запит *Мешканці_квартир*.

Виконання

– На стрічці Access перейдіть на вкладку **Создание**, на панелі **Запросы** клацніть на кнопці **Конструктор Запросов** .

– У вікні **Добавление таблицы**, що з'явилося, натиснувши клавішу **Ctrl**, виберіть зі списку такі запити: *Квартири_дані* та *Дані_про_нерухомість*.

– Закрийте вікно **Добавление таблицы**, клацнувши на кнопці **Закрыть**.

– У вікні конструктора запитів, що з'явилося, в області **Схема данных** (верхня частина вікна) встановіть зв'язок між запитами за однойменними полями: *Назва_вулиці*, *Номер_будівлі* та *Номер_квартири*. Для цього виділіть поле в одному запиті й перетягніть його ЛКМ на однойменне поле в іншому (рис. 4.39).

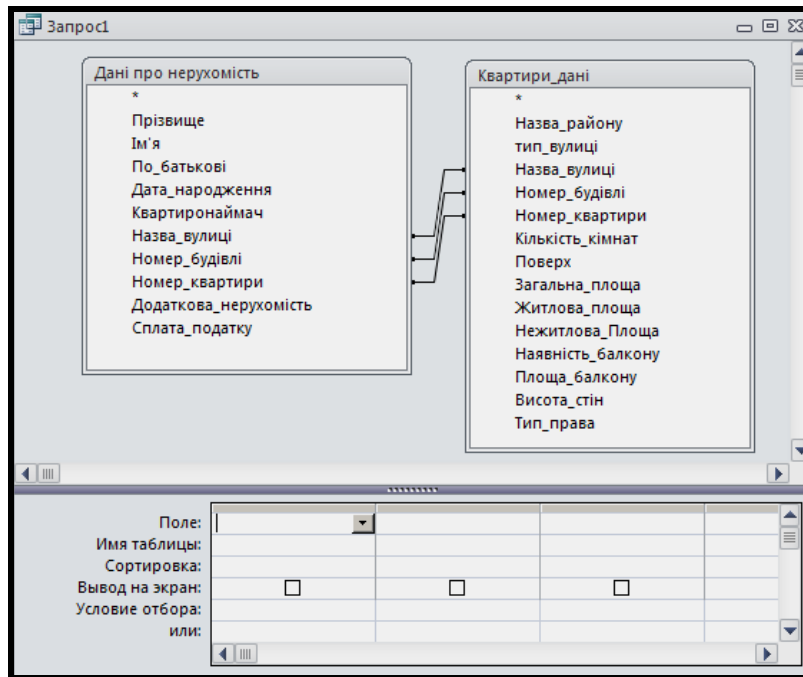


Рисунок 4.39 – Зв’язок між запитами

– Заповніть бланк запиту (нижня частина вікна) полями із запитів, які подані в таблиці 4.10.

Таблиця 4.10 – Поля для запиту «Мешканці_квартир»

Запит	Поле
Дані_про_нерухомість	Прізвище
Дані_про_нерухомість	Ім'я
Дані_про_нерухомість	По_батькові
Дані_про_нерухомість	Дата_народження
Дані_про_нерухомість	Квартиронаймач
Квартири_дані	Назва_вулиці
Квартири_дані	Номер_будівлі
Квартири_дані	Номер_квартири
Квартири_дані	Загальна_площа
Квартири_дані	Тип_права

Для цього послідовно перетягніть у рядок **Поле** на вільні стовпці імена полів із запитів, які розташовані у верхній частині вікна (рис. 4.40).

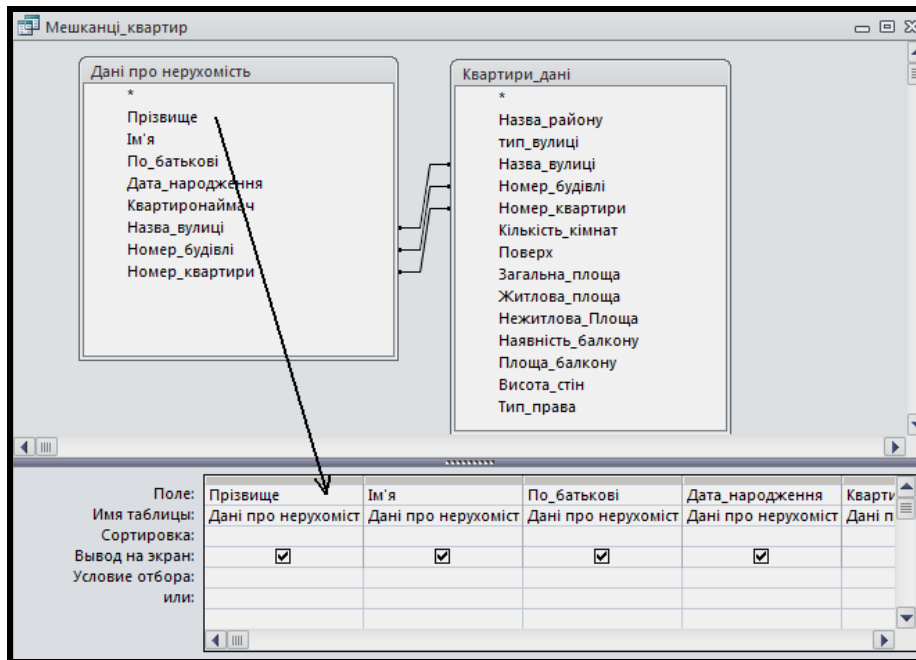



Рисунок 4.40 – Додавання полів із запиту

Збережіть запит, клацнувши на кнопці **Сохранить**, і введіть ім'я **Мешканці_квартир**.

Завдання

Щоб у запиті відображалися дані тільки про тих мешканців, які є дітьми війни (їм необхідно надати пільги), необхідно відібрати тільки тих, які мають дату народження з 1 червня 1941 р. по 9 травня 1945 р.

Виконання

Перейдіть у вікно конструктора запитів і клацніть у рядку **Условие отбора** поля **Дата_народження**. Викличте будівника виразів, натиснувши кнопку **Построитель**  на контекстній вкладці **Конструктор** у групі **Настройка запроса**. У лівому стовпці будівника виразів клацніть на значку **Операторы**, в середньому – на елементі **Сравнения**, а у правому – двічі на операторі **Between**. Потім замініть слова **Выражение** відповідними граничними датами й натисніть кнопку **ОК** (рис. 4.41).

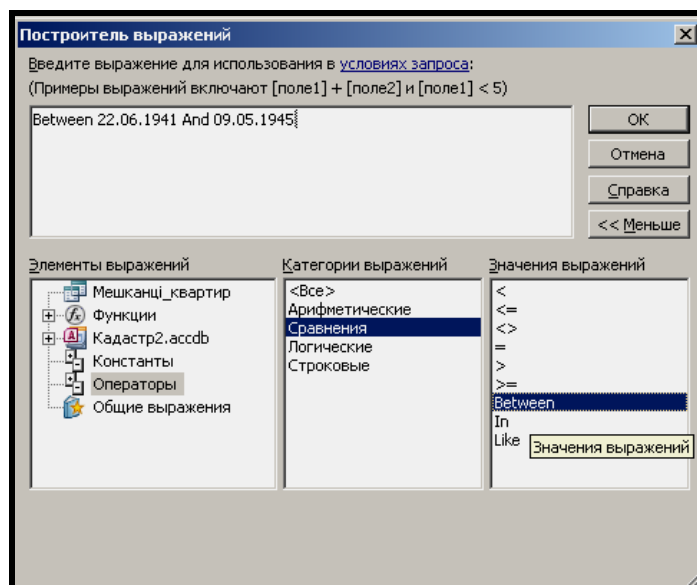


Рисунок 4.41 – Задавання діапазону дат в умові

У результаті виконаних дій отримаємо запит, що подано на рисунку 4.42.

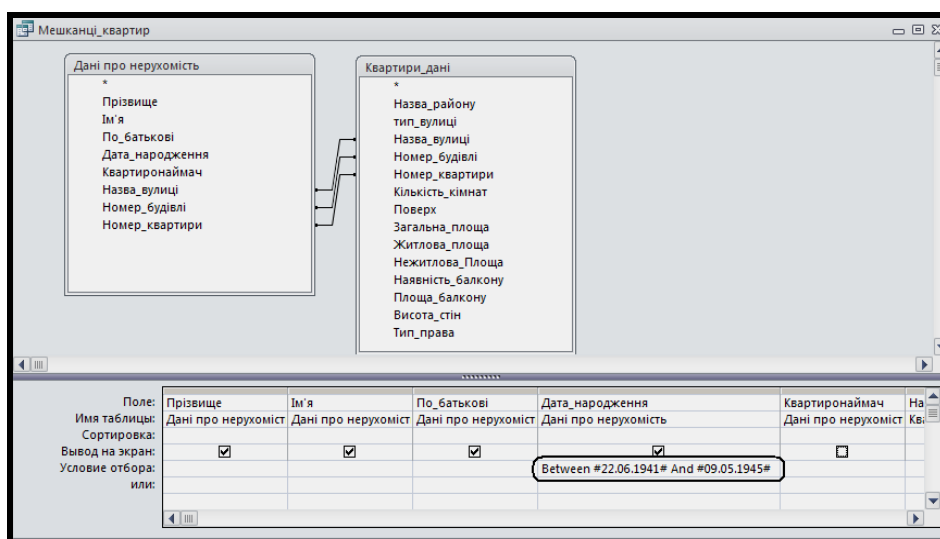


Рисунок 4.42 – Запит *Мешканці_квартир* в режимі конструктора

Перегляньте результат виконання запиту, перейшовши в режим таблиці, і натисніть кнопку **Сохранить**.

Перетворення запиту в параметричний

Завдання

Створити параметричний запит для відбору мешканців квартир за вибраним діапазоном дат народження (рис. 4.43). Це позбавить від

необхідності постійно перемикатися з режиму таблиці в режим конструктора й назад для введення нового діапазону дат народження.

Виконання

У режимі конструктора запиту *Мешканці_квартир* клацніть на рядку **Условие отбора** поля *Дата_народження* і замість конкретного значення введіть у квадратних дужках тексти підказок, наприклад [З] і [По], тобто сформууйте вираз **Between [З] And [По]**. Для зручності заміни конкретних значень текстами підказок скористайтеся вікном будівника виразів.

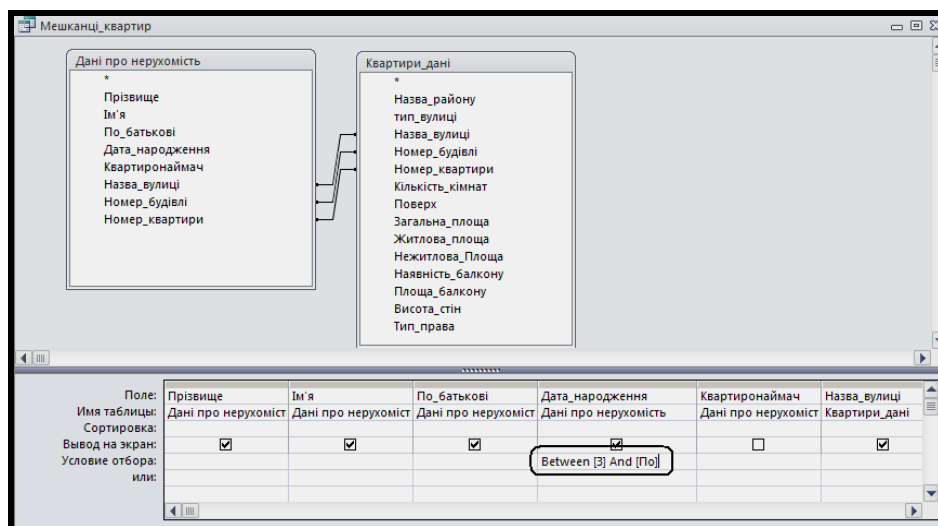


Рисунок 4.43 – Параметри в запиті

Збережіть запит і закрийте його.

Відкрийте запит *Мешканці_квартир*, двічі клацнувши на його значку в області переходів. У діалогових вікнах, що з'являються, введіть межі діапазону дат народження мешканців (наприклад, 1.09.1945 і 1.09.2013). Перегляньте результат виконання запиту та закрийте його вікно.

Під час подальших відкриттів запиту можна вводити інші дати й коди виробника. Закрийте вікно запиту зі збереженням.

Додавання умов відбору

Запит Квартиронаймач

Завдання

З усіх записів запиту *Квартиронаймач* вибрати тільки мешканців, які мешкають у певному будинку. Якщо номер будинку не введено, то відобразити всіх мешканців. Потім серед усіх мешканців відібрати тільки тих, які є відповідальними квартиронаймачами.

Виконання

Для виконання цього завдання створимо запит *Квартиронаймач* на підставі таблиць: *Мешканці*, *Квартири*, *Будівлі*.

Додайте до запиту поля з таблиць відповідно до наведеної нижче таблиці (табл. 4.11).

Таблиця 4.11 – Поля для запиту *Квартиронаймач*

Таблиця	Поле
Будівлі	Номер_будівлі
Квартири	Номер_квартири
Мешканці	Прізвище
Мешканці	Ім'я
Мешканці	По_батькові
Мешканці	Квартиронаймач

В умові відбору для поля *Номер_будівлі* запиту *Мешканці_квартир_будинки* використовуємо вбудовану функцію:

If («expr»; «truepart»; «falsepart»)

вона виконується так само, як і однорядковий оператор:

If ... Then ... Else ...

у мові **Visual Basic**.

У першому аргументі функції **If** указують умову, в другому – значення цієї функції, якщо умова виконується, а в третьому – значення функції, якщо умова не виконується.

Величина, що вводиться, має те саме ім'я, що й параметр в умові відбору. У цьому випадку це [*Введіть номер будинку*]. Значення цієї величини співпадає із введеним значенням.

Будемо перевіряти чи не порожнє поле введення [**Введіть номер будинку**]<>"""). Якщо воно не порожнє, то встановимо введене значення, тобто [**Введіть номер будинку**].


Інакше встановимо значення **Номер_будівлі**, тобто умова відбору співпадає з ім'ям поля. У цьому випадку при виконанні запиту Access буде користуватися рівнянням вигляду:

$$a = a$$

Тому всі записи задовольняють його.

Отже, потрібно сформулювати таку умову відбору:

**Иf([Введіть номер будинку]<>""; [Введіть номер будинку];
[Номер_будівлі]).**

- Перейдіть у вікно конструктора запиту **Квартиронаймач**.
- Клацніть на рядку **Условие отбора** поля **Номер_будівлі**.
- Викличте будівника виразів, натиснувши кнопку **Построить** .
- У лівому стовпці будівника виразів подвійним клацанням відкрийте елемент виразу **Функции**, потім одним клацанням – **Встроенные функции**, в середньому стовпці клацніть на категорії **Управление**, а в правому – двічі клацніть на функції **Иf**.
- Перший аргумент **expr** замініть на [**Введіть номер будинку**]<>"" (вводять за допомогою клавіатури).
- Виділіть ліву частину попереднього виразу [**Введіть номер будинку**] і скопіюйте його в буфер обміну (клавіші **Ctrl+C**), а потім вставте замість другого аргументу (клавіші **Ctrl+V**).
- Виділіть третій аргумент і у верхній частині лівого стовпця будівника виразів клацніть на назві запиту **Квартиронаймач**, а в середньому – двічі клацніть на полі **Номер_будівлі** (рис. 4.44).

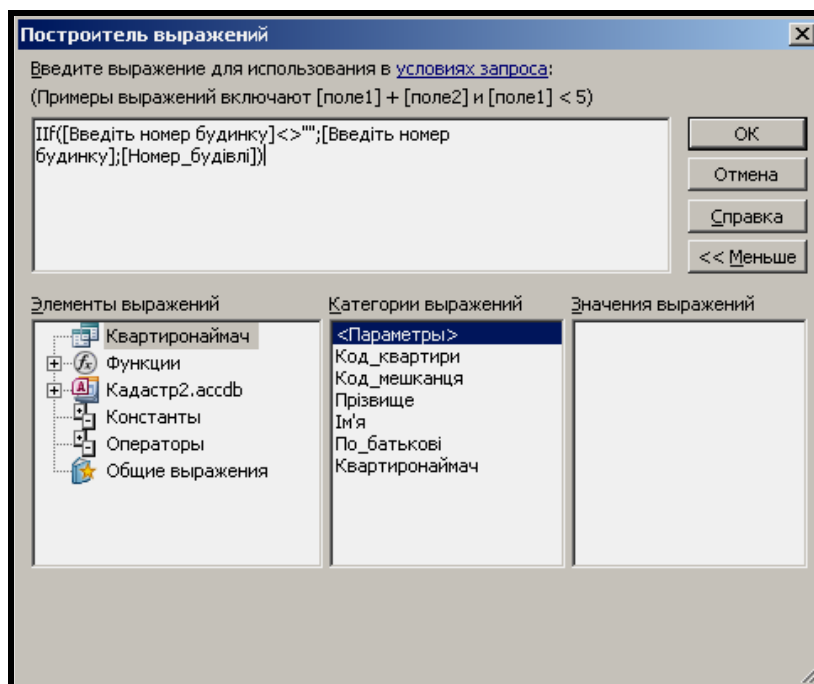


Рисунок 4.44 – Сформована умова відбору

– У вікні будівника виразів сформована необхідна умова, тому клацніть на кнопці **ОК**.

– Збережіть запит, клацнувши на кнопці **Сохранить** і закрийте вікно запиту.

– Перевірте роботу запити *Квартиронаймач*. Введіть номери будинків у вікні введення, яке з'являється при відкритті запиту (номери будинків повинні співпадати з тими, що фігурують у таблиці *Будівлі*). Спробуйте нічого не вводити у вікно параметра при відкритті запиту. Поясніть отримані результати.

Перейдіть до виконання другої частини завдання, а саме до відбору серед усіх мешканців тільки тих, які є відповідальними квартиронаймачами. Для цього створіть умову у полі *Квартиронаймач*.

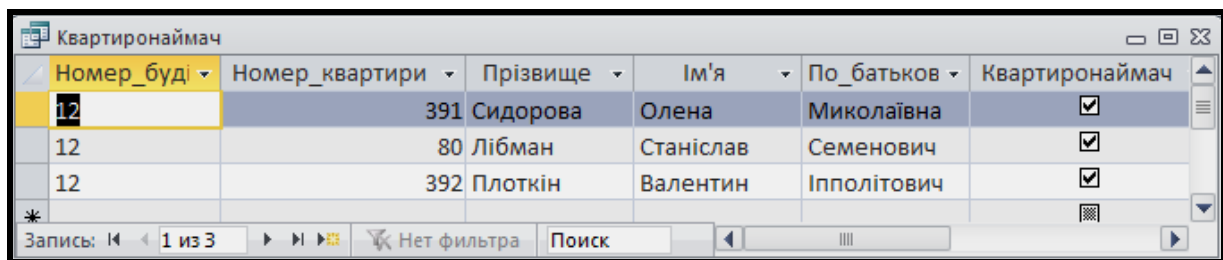
– Перейдіть у вікно конструктора запити *Квартиронаймач*.

– Клацніть на рядку **Условие отбора** поля *Квартиронаймач*.

Згадаємо про те, що тип даних у поля *Квартиронаймач* – **Логический**. Тому для того щоб відібрати лише відповідальних квартиронаймачів, достатньо у рядку **Условие отбора** написати вираз **Истина**.

– Перевірте роботу запити *Квартиронаймач*. Зверніть увагу на те, що кількість мешканців квартир, що відображає запит, значно зменшилася у зв'язку з тим, що тепер наведено дані тільки по відповідальному

квартиронаймачу, а не по всіх мешканцях квартири (рис. 4.45).



Номер_буді	Номер_квартири	Прізвище	Ім'я	По_батьков	Квартиронаймач
12	391	Сидорова	Олена	Миколаївна	<input checked="" type="checkbox"/>
12	80	Лібман	Станіслав	Семенович	<input checked="" type="checkbox"/>
12	392	Плоткін	Валентин	Іпполітович	<input checked="" type="checkbox"/>

Рисунок 4.45 – Запит *Квартиронаймач* із даними по будинку № 12

Завдання для самостійного виконання

1. У який день тижня були дні народження мешканців? (Для визначення назв днів тижня можна використати функції **Weekday** і **WeekdayName**).

2.*Побудуйте параметричний запит на вибірку про житловий фонд окремої вулиці (відповідно до введеного коду вулиці). Якщо код не введено, то виведіть інформацію про житловий фонд усіх вулиць міста. (В умові відбору можна скористатися вбудованою функцією **Иф**, яка перевіряє значення параметра [**Введіть код вулиці**] – чи він не порожній).

3.*Визначте вартість квартир, виходячи з припущення, що вартість 1 м² житла становить 1500 умовних одиниць, у разі якщо матеріал, з якого побудовано будинок, – цегла, у всіх інших випадках, вартість 1 м² житла – 1 000 умовних одиниць.

4.*Визначте, на які квартири було зареєстровано документи за останню неділю. (Для таблиці *Документи* у полі *Дата реєстрації* використайте функцію **Date**).

4.5 Практичне заняття. Створення нових форм у СКБД ACCESS. Робота з комбінованими формами

Мета – набути вмінь та навичок створення форм і їх використання під час ведення й аналізу даних, що зберігаються в базі.

Призначення – виконавши роботу, можна навчитися:

- створювати форми за допомогою майстра;
- змінювати форми в конструкторі;
- створювати комбіновані форми;

Отримані вміння й навички допоможуть створювати зручне середовище для роботи з базами даних у вашій професійній діяльності, закладуть основи розроблення користувальницького інтерфейсу під час розроблення застосувань із базами даних.

ХІД РОБОТИ

Форма для перегляду одного запису

Завдання

Створити форму *Мешканці*, за допомогою якої зручно переглядати, додавати й видаляти записи, змінювати значення полів таблиці *Мешканці* (рис. 4.46).

Field Name	Value
Код мешканця	
Прізвище	Сидорова
Ім'я	Олена
По батькові	Миколаївна
Дата народження	13.05.1956
Код квартири	3
Номер квартири	391
Код будівлі	2
Номер будівлі	12
Назва вулиці	Героїв праці
Назва району	Київський
Кількість кімнат	2
Загальна площа	44,7
Площа балкону	2,4
Загальна площа квартири з балконом	45,42

Рисунок 4.46 – Форма *Мешканці*

Основні етапи виконання

Для організації послідовної роботи з черговим записом найзручнішою вважається форма «в один стовпець», коли поля запису розташовують один під одним, а ліворуч від них – пояснювальні написи (найчастіше – це назви полів).

Побудова форми *Мешканці* та виконання операцій із даними за її допомогою складається з п'яти етапів:

1. Створення форми в один стовпець.
2. Використання форми.
3. Додавання обчислюваного поля.
4. Вставка поля зі списком.
5. Удосконалення форми.

Створення форми в один стовпець

Завдання

За допомогою майстра побудувати форму *Мешканці* для перегляду, додавання, видалення та зміни даних в однойменній таблиці (рис. 4.47).

Код_мешканця	<input type="text" value="22"/>
Прізвище	<input type="text" value="Поморцева"/>
Ім'я	<input type="text" value="Олена"/>
По_батькові	<input type="text" value="Євгенівна"/>
Дата_народження	<input type="text" value="10.08.1971"/>
Код_квартири	<input type="text" value="1"/>
Номер_квартири	<input type="text" value="25"/>
Код_будівлі	<input type="text" value="1"/>
Номер_будівлі	<input type="text" value="5"/>
Код_вулиці	<input type="text" value="1"/>
Назва_вулиці	<input type="text" value="Пушкінська"/>
Код_району	<input type="text" value="1"/>
Назва_району	<input type="text" value="Київський"/>
Кількість_кімнат	<input type="text" value="2"/>
Загальна_площа	<input type="text" value="55,8"/>
Площа_балкону	<input type="text" value="0"/>

Записи: 4 из 20 Нет фильтра Поиск

Рисунок 4.47 – Початковий вигляд форми *Мешканці*

Виконання

- Відкрийте базу даних «*Кадастр*», яка утворилася після виконання лабораторної роботи «Створення запитів».
- Виділіть таблицю *Мешканці* в області переходів.
- На вкладці **Создание** у групі **Формы** натисніть кнопку **Мастер форм**.
- У першому вікні майстра **Создание форм** виберіть поля з таблиць БД, які будуть відображатися на формі. Для цього у список **Выбранные поля** перешліть поля з таблиць даних, які наведено нижче (табл. 4.12).

Таблиця 4.12 – Поля для форми *Мешканці*

Таблиця	Поле
Мешканці	Код_мешканця
Мешканці	Прізвище
Мешканці	Ім'я
Мешканці	По_батькові
Мешканці	Дата_народження
Мешканці	Код_квартири
Квартири	Номер_квартири
Квартири	Код_будівлі
Будівлі	Номер_будівлі
Будівлі	Код_вулиці
Вулиці	Назва_вулиці
Будівлі	Код_району
Райони	Назва_району
Квартири	Кількість_кімнат
Квартири	Загальна_площа
Квартири	Площа_балкона

- У другому вікні майстра вкажіть базову таблицю для форми. За замовчуванням виділено таблицю *Мешканці*, тому натисніть кнопку **Далее**.
- У третьому вікні майстра виберіть зовнішній вигляд форми. Оскільки вигляд в один стовпець задано за замовчуванням, то достатньо натиснути кнопку **Далее**.
- В останньому вікні майстра погодьтеся з ім'ям форми *Мешканці*, натиснувши кнопку **Готово**.

За допомогою кнопок переходу записами перегляньте дані форми **Мешканці**.

Збережіть форму, натиснувши на панелі швидкого доступу кнопку **Сохранить**.

Використання форми

Завдання

У даних з проживання мешканців у квартирі за адресою: вул. Пушкінська, будинок № 5, квартира 25, відбулися зміни у зв'язку з одруженням сина. Треба додати до мешканців цієї квартири нову людину.


Завдання спрямоване на освоєння операцій додавання та зміни даних за допомогою форми.

Виконання

– Відкрийте вікно таблиці **Квартири**, двічі клацнувши на її значку на панелі **Все объекты Access**. Знайдіть у ній значення коду квартири для квартири № 25 (наприклад, 1), запам'ятайте його і закрийте вікно таблиці.

– Відкрийте вікно таблиці **Вулиці**, двічі клацнувши на її значку на панелі **Все объекты Access**. Знайдіть у ній значення коду вулиці Пушкінська (наприклад, 1), запам'ятайте його і закрийте вікно таблиці.

– Відкрийте вікно таблиці **Будівлі**, двічі клацнувши на її значку на панелі **Все объекты Access**. Знайдіть у ній запис зі значенням коду вулиці Пушкінська з номером будівлі 5 (наприклад, 1), запам'ятайте його і закрийте вікно таблиці.

– У формі **Мешканці** перейдіть до нового запису, натиснувши кнопку **Новая (пустая) запись**  у групі кнопок переходів записами.

– У поля нового запису введіть такі значення (табл. 4.13).

Таблиця 4.13 – Значення полів нового запису

Поле	Значення
Прізвище	Заворотнюк
Ім'я	Олена
По батькові	Євгенівна
Дата народження	10.08.1980
Код квартири	1

Зверніть увагу на те, що як тільки було введено код квартири, дані про номер будівлі, вулицю, район і дані про квартиру відразу з'явилися на

формі.

Щоб новий запис зберігся в БД, клацніть в області виділення, що розташована на лівій межі форми. При цьому значок олівця в цій області замінюється значком трикутника.

Припустимо, що була зроблена помилка при введенні даних про нового мешканця квартири (неправильно ввели дату народження). Необхідно її виправити.

– За допомогою кнопок переходу записами на формі *Мешканці* знайдіть запис, який було щойно додано. У полі *Дата_народження* замість 10.08.1980 введіть 15.10.1980.

– Відкрийте таблицю *Мешканці* і переконайтеся, що відбулися зміни.

Додавання обчислюваного поля


Завдання

На форму *Мешканці* помістити поле, в якому обчислюється загальна площа квартири з урахуванням площі балкона. Для балконів використовується знижувальний коефіцієнт – 0,3 (рис. 4.48).


Код_мешканця	
Прізвище	Сидорова
Ім'я	Олена
По_батькові	Миколаївна
Дата_народження	13.05.1956
Код_квартири	3
Номер_квартири	391
Код_будівлі	2
Номер_будівлі	12
Код_вулиці	2
Назва_вулиці	Героїв праці
Код_району	1
Назва_району	Київський
Кількість_кімнат	2
Загальна_площа	44,7
Площа_балкону	2,4
Загальна площа квартири з балконом	45,42


Рисунок 4.48 – Форма *Мешканці* з полем *Загальна площа квартири з балконом*

Виконання

– Відобразіть форму *Мешканці* в режимі конструктора, вибравши з випадального списку кнопку **Конструктор**  на панелі **Режим** на вкладці **Главная**.

– Збільшіть розміри області даних форми, щоб у ній могло поміститися нове поле. Для збільшення розмірів області даних установіть покажчик миші на межі між областю даних і приміткою форми, після чого перетягніть межу вниз приблизно на 1,5 см.

– На панелі **Элементы управления** вкладки **Конструктор** натисніть кнопку **Поле** , а потім – на формі під полем *Площа_балкона*. На формі з'являться виділене поле і напис **Поле...** Тут розміщуватиметься нове поле *Загальна площа квартири з балконом*.

– Відкрийте вікно властивостей нового поля, натиснувши кнопку **Страница свойств** , що розташована на панелі **Сервис** вкладки **Конструктор**.

– На вкладці **Данные** вікна **Окно свойств** що з'явилося, клацніть на рядку властивостей **Данные** і натисніть кнопку виклику будівника виразів.

– Побудуйте вираз для обчислення площі. Для цього у вікні будівника виразів у середньому стовпці двічі клацніть на полі *Загальна_площа*, щоб воно з'явилося у верхній частині вікна. Потім введіть знак додавання. Після цього у середньому стовпці двічі клацніть на полі *Площа_балкона* і побудуйте такий вираз:

$$[Загальна_площа]+([Площа_балкона]*0,3).$$

Закінчіть введення виразу натисканням кнопки **ОК** у вікні будівника виразів (рис. 4.49).

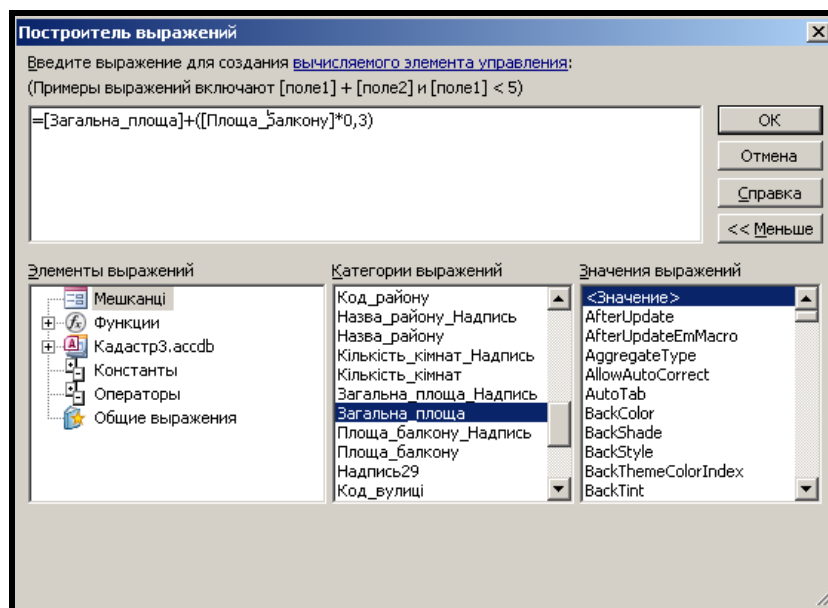


Рисунок 4.49 – Побудова виразу для вартості

– Для того щоб обчислюване поле було недоступним для випадкової зміни під час перегляду записів, на вкладці **Данные** вікна **Окно свойств** установіть значення таких властивостей (табл. 4.14).

Таблица 4.14 – Значення властивостей у вкладці **Данные**

Властивість	Значення
Доступ	Нет
Блокировка	Да

– Щоб результат обчислень виводився з двома десятковими знаками, на вкладці **Макет** вікна **Окно свойств** установіть значення таких властивостей (табл. 4.15).

Таблица 4.15 – Значення властивостей у вкладці **Макет**

Властивість	Значення
Формат поля	С разделителями разрядов
Число десятичных знаков	2

– Виходячи з естетичних міркувань, для поля *Загальна площа квартири з балконом* зробіть колір фону відмінним від інших полів. Для цього на вкладці **Макет** вікна **Окно свойств** установіть для властивості **Цвет фона** значення **Чередующаяся строка** (сірий колір), вибравши його з випадającego списку.

– Клацніть на написі **Поле...** один раз, а потім ще раз і замість попереднього тексту введіть назву поля *Загальна площа квартири з балконом*.


– Вирівняйте напис *Загальна площа квартири з балконом* по лівій межі інших написів. Для цього:

– Клацніть на написі *Загальна площа квартири з балконом*, а потім, утримуючи натиснутою клавішу **Shift** – на написі *Площа_балкона*, щоб виділити їх.

– Клацніть правою клавішею миші на одному з виділених написів і виберіть із контекстного меню команду **Выровняют – Слева**.

– Повторіть попередній пункт для текстових полів, що залишилися.

– Збережіть зміни у формі.


– Перейдіть у режим форми, натиснувши кнопку **Режим**  на вкладці **Главная** в панелі **Режимы**.

– Перегляньте записи, спостерігаючи за площею квартир.

Вставка поля зі списком

Завдання

Додати на форму *Мешканці* поля зі списком *Назва вулиці* і *Назва району*. Вони використовуються для введення в таблицю *Будівлі* кодів вілиць і районів шляхом вибору відповідних назв у списках (рис. 4.50).

– Відобразіть форму *Мешканці* в режимі конструктора, вибравши кнопку **Конструктор**  на панелі **Режим** на вкладці **Главная**.

– Збільшіть розміри області даних форми в горизонтальному напрямку приблизно на 5 см.

– Натисніть кнопку **Дополнительные параметры** на панелі **Элементы управления** на вкладці **Конструктор** (рис. 4.51).

– На розширеній панелі елементів клацніть на кнопці **Поле со списком** на панелі **Элементы управления** на вкладці **Конструктор** (рис. 4.52), а потім – на формі праворуч від поля *Назва_вулиці*. Простежте, щоб при цьому знаходилася в натиснутому стані кнопка **Использовать мастера** на панелі **Элементы управления**.

Мешканці

Код_мешканця:

Прізвище:

Ім'я:

По_батькові:

Дата_народження:

Код_квартири:

Номер_квартири:

Код_будівлі:

Номер_будівлі:

Код_вулиці:

Назва вулиці:

Назва району:

Кількість кімнат:

Загальна площа:

Площа балкону:

Загальна площа квартири з балконом:

Записи: 1 из 20 | Нет фильтра | Поиск

Рисунок 4.50 – Форма *Мешканці* з полями зі списком

Виконання

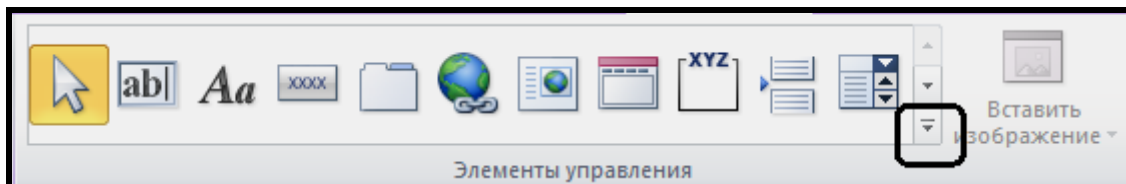


Рисунок 4.51 – Кнопка **Дополнительные параметры** на панелі **Элементы управления**

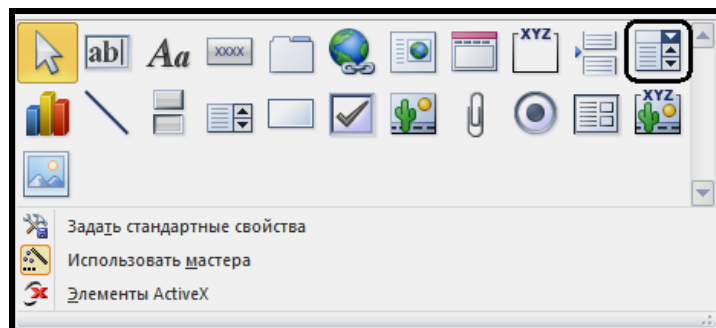


Рисунок 4.52 – Кнопка **Поле со списком** на розширеній панелі елементів

– У першому вікні майстра **Создание полей со списком**, що

з'явилося, встановлено за замовчуванням перемикач **Объект «поле со списком»** получит значения из другой таблицы или другого запроса. Він повністю підходить у цьому разі, тому натисніть кнопку **Далее**.

– У другому вікні майстра виберіть базову таблицю (в інших випадках це може бути й запит), дані з якої відобразатимуться в списку (в даному випадку це таблиця **Вулиці**), і натисніть кнопку **Далее**.

– У третьому вікні майстра виберіть потрібні для утворення списку поля з базової таблиці або запиту в тому порядку, в якому вони відобразатимуться в рядках списку (таких полів може бути декілька). Першим бажано вказати поле, що містить код. Його значення записується в таблицю, але не відображається в списку. У даному разі виберіть поля **Код_вулиці** і **Назва_вулиці** та натисніть кнопку **Далее** (рис. 4.53).

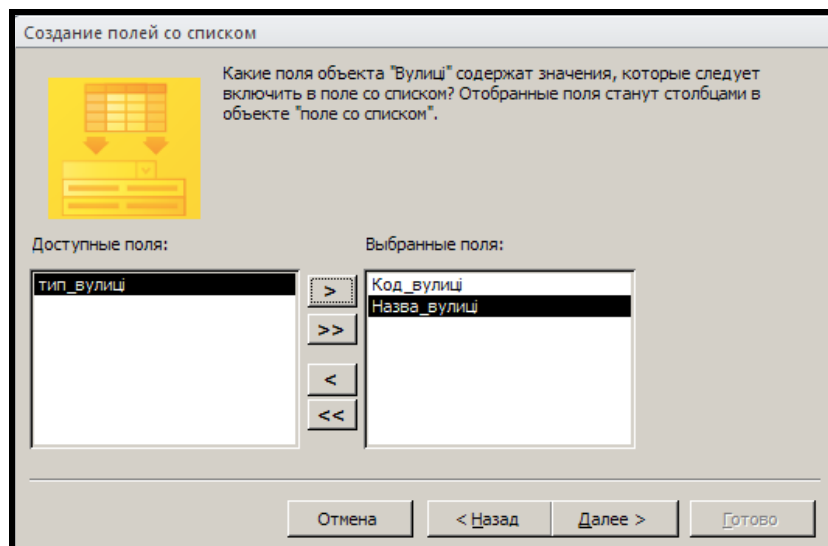


Рисунок 4.53 – Відбір полів для утворення списку

– У четвертому вікні майстра можна задати сортування за відібраними полями, а в п'ятому – відрегулювати ширину кожного стовпця списку шляхом перетягування меж заголовків. Ці дії виконувати не обов'язково. У кожному вікні натисніть кнопку **Далее**.

– У шостому вікні майстра вкажіть, як надалі використовувати вибране значення **Сохранить в поле**. При цьому виберіть поле базової таблиці, в якому буде збережено значення коду, відповідно до вибраного у списку найменування, в даному разі в полі **Код_вулиці** (рис. 4.54).

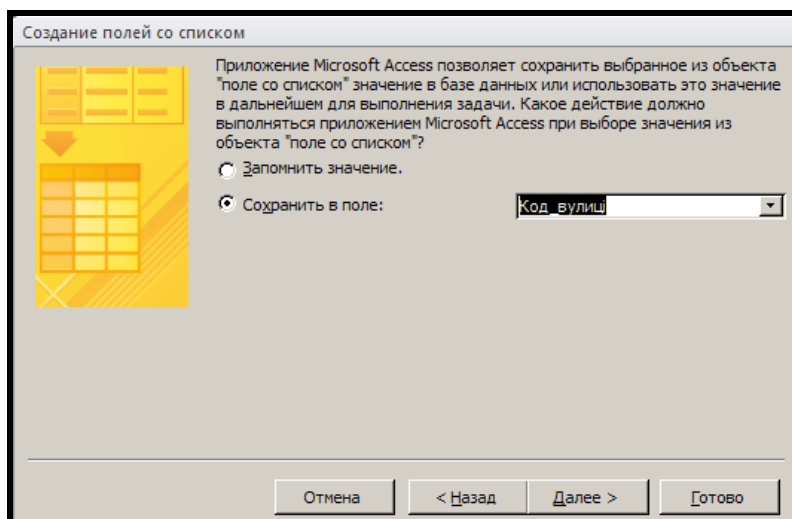


Рисунок 4.54 – Вибір перемикача **Сохранить в поле** і поле **Код_вулиці**

– В останньому вікні майстра задайте підпис, який виводитиметься на формі поряд із полем зі списком (у цьому разі **Назва вулиці**) і натисніть кнопку **Готово**.

Так само створіть поле зі списком **Назва району** для введення коду району з таблиці **Райони** в таблицю **Будівлі**.

Після створення полів зі списком **Назва вулиці** та **Назва району** видаліть відповідні текстові поля, оскільки вони дублюють значення полів зі списком. Поля, в які раніше вводилися значення відповідних кодів, залишіть на формі для візуального контролю. Зробіть поля кодів недоступними для зміни, встановивши значення таких властивостей (табл. 4.16):

Таблиця 4.16 – Значення властивостей полів кодів

Властивість	Значення
Доступ	Нет
Блокировка	Да
Цвет фона	Чередующаяся строка

На рисунку 4.55 подано зовнішній вигляд форми в режимі конструктора.

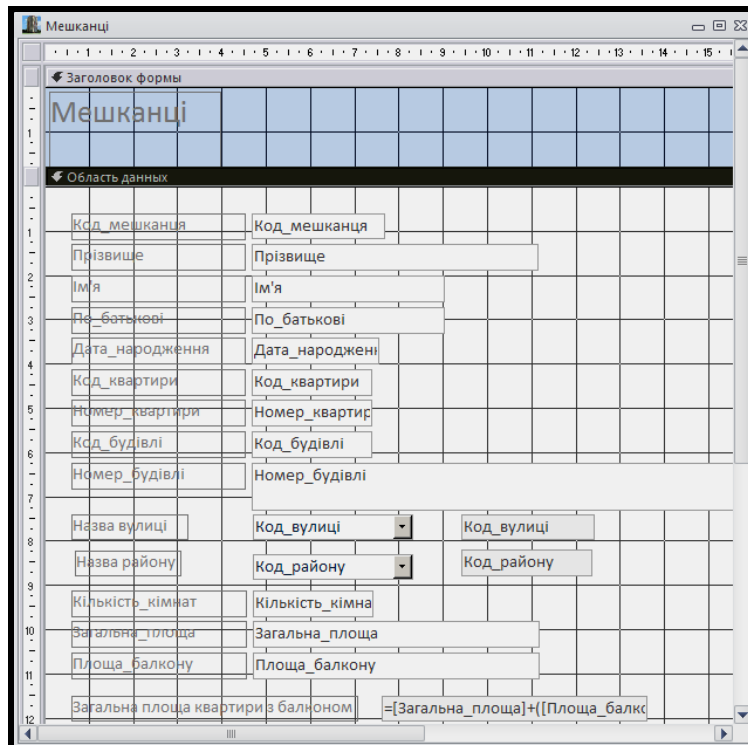



Рисунок 4.55 – Форма *Мешканці* в режимі конструктора після додавання полів зі списком

Збережіть зміни у формі *Мешканці* та перейдіть у режим відображення форми. Перейдіть до нового запису, натиснувши кнопку **Новая (пустая) запись**  у групі кнопок переходів записами.

У поля нового запису введіть такі значення (табл. 4.17):

Таблиця 4.17 – Значення полів нового запису

Поле	Значення
Прізвище	Фурман
Ім'я	Іван
По_батькові	Васильович
Дата_народження	05.03.1980
Код_квартири	11
Номер_квартири	30
Код_будівлі	8
Номер_будівлі	12а
Назва_вулиці	Героїв Праці
Назва_району	Київський
Кількість_кімнат	1
Загальна_площа	35
Площа_балкону	2

Щоб новий запис зберігся в БД, клацніть в області виділення, що розташована на лівій межі форми. При цьому значок олівця замінюється значком трикутника. Відкрийте таблиці *Квартири*, *Будівлі*, *Мешканці* та переконайтеся, що введені за допомогою форми дані потрапили до цих таблиць. У разі необхідності введіть відсутні дані до таблиць.

Удосконалення форми

Завдання 1

Зробити недоступним поле *Код_мешканця*. Воно має тип даних *Счетчик*, тому його не можна змінити на формі *Мешканці*.

Виконання


При виконанні скористайтеся описами попереднього підрозділу.

Завдання 2

Забезпечити природний порядок обходу елементів керування на формі *Мешканці* за допомогою клавіші **Tab** або **Enter**.

Виконання

Знаходячись у режимі конструктора форми, виділіть поле *Код_мешканця*, а потім відкрийте вікно властивостей.

– У вкладці **Другие** клацніть на рядку властивості **Индекс перехода по Tab**, а в ньому – на кнопці виклику будівника виразів .

– У вікні **Последовательность перехода**, що з'явилося, встановіть потрібний порядок. Для цього, скориставшись смугою виділення (праворуч у списку **Последовательность**), виділяйте назви полів і перетягуйте їх мишею за списком угору або вниз (рис. 4.56).

Для того щоб дізнатися назву поля (наприклад, *ПолеСоСписком17* для полів зі списком та формул), необхідно у режимі конструктора виділити необхідне поле та викликати для нього **Окно свойств**. У верхньому полі з випадаючим списком наведено назву поля (рис. 4.57).

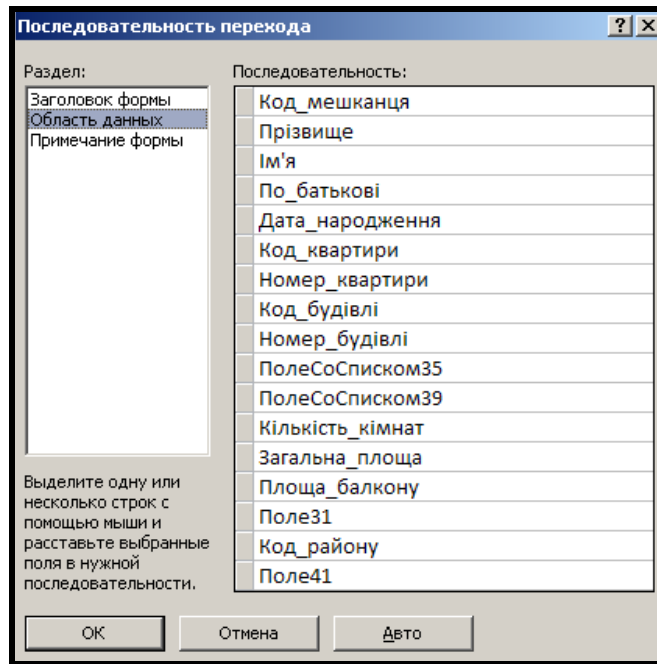


Рисунок 4.56 – Встановлення потрібної послідовності переходів шляхом перетягування імен полів

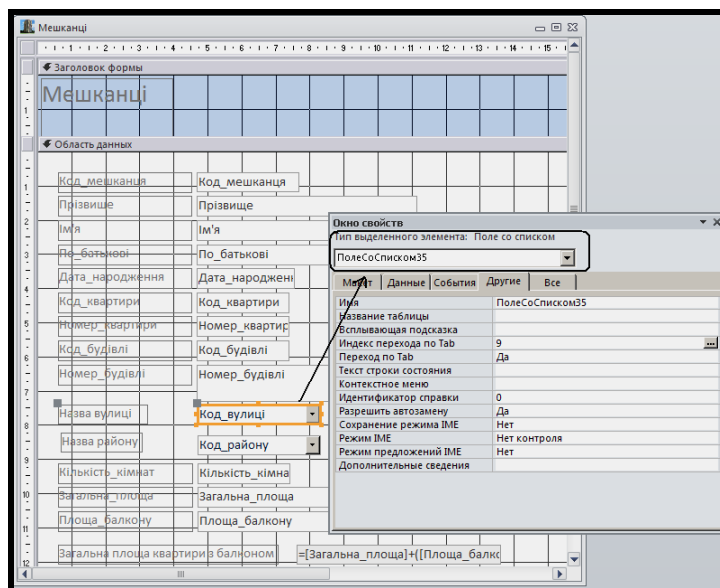


Рисунок 4.57 – Встановлення потрібної послідовності переходів шляхом перетягування імен полів

Завдання 3

Відокремити поле *Загальна площа квартири з балконом* від попередніх полів червоною горизонтальною лінією та виділити значення поля червоним кольором (рис. 4.58).

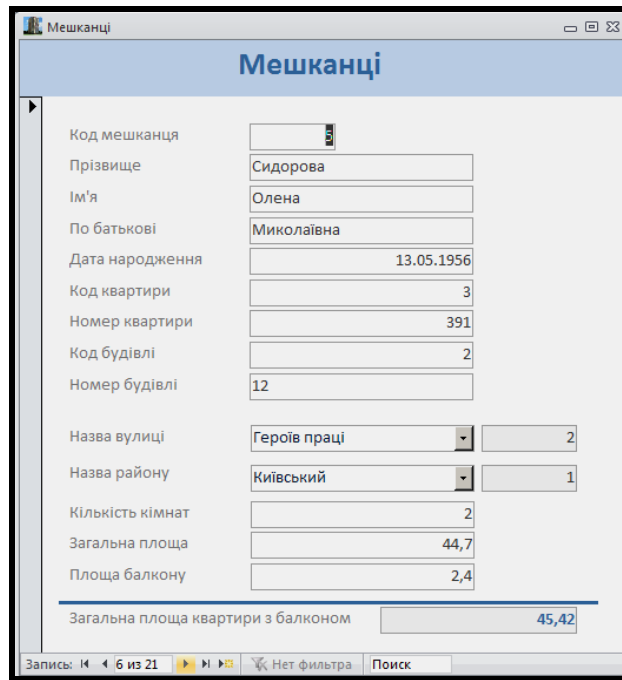


Рисунок 4.58 – Остаточний зовнішній вигляд форми *Мешканці*

Виконання

– Знаходячись у режимі конструктора форми, перетягніть на 1 см донизу поле *Загальна площа квартири з балконом* разом із написом, щоб звільнити місце для лінії.

– На розширеній панелі елементів натисніть кнопку **Лінія** на панелі **Елементи управління** на вкладці **Конструктор**, а потім проведіть горизонтальну лінію між полями *Площа балкона* і *Загальна площа квартири з балконом*.

Відкрийте вікно властивостей лінії. У вкладці **Макет** установіть значення таких властивостей (табл. 4.18):

Таблиця 4.18 – Значення властивостей лінії

Поле	Значення
Ширина границы	2 пункта
Цвет границы	Система – рабочий стол

– Виділіть поле *Загальна площа квартири з балконом* й у вкладці **Макет** вікна властивостей установіть значення таких властивостей (табл. 4.19):

Таблиця 4.19 – Значення властивостей поля *Загальна площа квартири з балконом*

Поле	Значення
Цвет текста	Система – рабочий стол
Насыщенность	Полужирный

– Виділіть усі назви полів та поля (для цього наведіть курсор миші на вертикальну лінійку зліва на формі та протягніть зверху донизу, натискаючи ліву кнопку миші). Відкрийте **Окно свойств** та на вкладці **Макет** у властивості **Высота** встановіть значення 0,582 см.

– Виділіть всі назви полів (для цього за допомогою клавіші **Ctrl** виділяйте послідовно поля), клацніть правою кнопкою миші на будь-якому виділеному полі та у меню, що з'явилося, виберіть команди **Выровнять / Слева**.

– Виділіть усі поля, відкрийте **Окно свойств** та на вкладці **Макет** у властивості **Ширина** встановіть значення 5 см. Для полів із кодами встановіть ширину 2 см.

– Виділяючи та переміщаючи поля за допомогою натискання лівої клавіші миші (потрібно, щоб показчик миші був у вигляді чотирьохнаправленої стрілки) підрівняйте поля на формі.

На рисунку 4.59 подано остаточний вигляд форми в режимі конструктора.

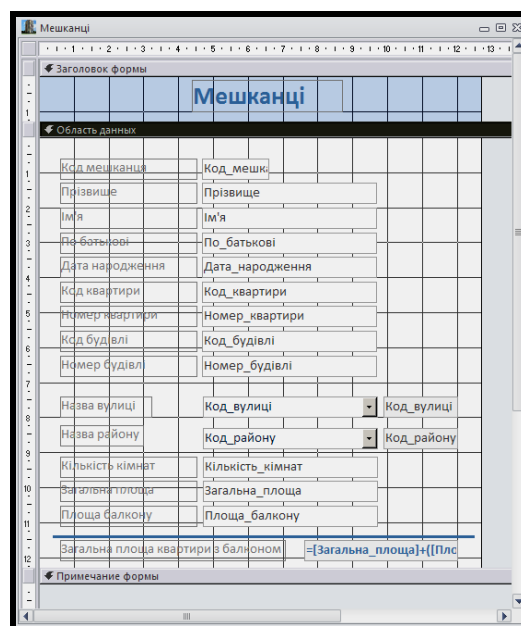


Рисунок 4.59 – Форма *Мешканці* в режимі конструктора

Збережіть зміни у формі *Мешканці* та перейдіть у режим відображення форми. Перегляньте записи й закрийте форму.

Комбінована форма

Завдання

Створити форму *Житловий фонд по районах* для відображення інформації про житловий фонд обраного району (рис. 4.60).

Тип вулиці	Вулиця	Номер будівлі	Матеріал	Ліфт	Кількість поверхів	Рік будівання	Загальна площа квартир
вулиця	Пушкінська	5	Цегла	<input checked="" type="checkbox"/>	3	1930	115,80
вулиця	Героїв праці	12	Панелі	<input checked="" type="checkbox"/>	12	1976	225,10
вулиця	Героїв праці	12а	Залізобетон	<input checked="" type="checkbox"/>	9	1977	35,00
вулиця	Блюхера	44	Панелі	<input checked="" type="checkbox"/>	9	1978	67,00
Загальна площа квартир району							442,90

Рисунок 4.60 – Комбінована форма *Житловий фонд по районах*

Основні етапи виконання

Для організації роботи із записами таблиць, що пов'язані відношенням «один-до-багатьох», зазвичай використовують комбіновані форми. Вони складаються з головної та підпорядкованої форм.

На головній формі відображають дані, що містяться в батьківській таблиці, а на підпорядкованій – у дочірній.

Для головної форми використовують вигляд «в один стовпець». Тут записи відображаються по одному.

Підпорядковану форму будують на основі стрічкової або табличної форми. На ній одночасно відображається кілька записів, а саме ті, які пов'язані із записом, що міститься на головній формі. Стрічковий вигляд використовують, коли потрібно виконати загальні обчислення (сума, середнє тощо), значення яких поміщають в області заголовка або приміток. Тут також можуть розташовуватися командні кнопки. У табличній формі цих розділів немає.

Побудова форми *Житловий фонд по районах* складається з таких етапів:

1. Створення головної форми.
2. Створення підпорядкованої форми.
3. Об'єднання підпорядкованої форми з головною.

На головній і підпорядкованій формах обов'язково має бути поле, за яким підпорядкована форма зв'язується з головною. За значенням цього поля на підпорядкованій формі відображаються записи, які відповідають запису, що переглядається на головній формі.

Найбільш надійно об'єднання підпорядкованої форми з головною проводиться за допомогою відповідного майстра.

Створення головної форми

Завдання

Створити головну форму *Житловий фонд по районах* на базі таблиці *Райони*, в якій відображаються поля *Код_району* і *Назва_району* (рис. 4.61).

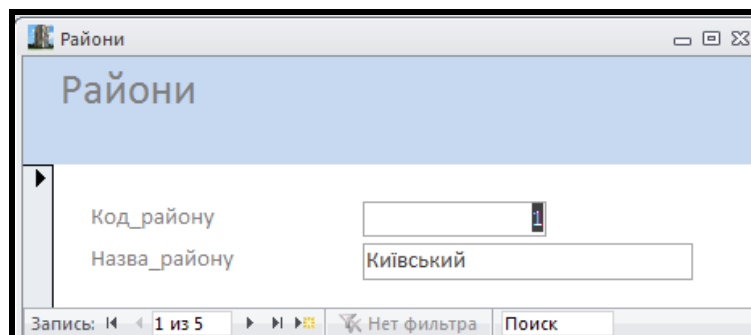



Рисунок 4.61 – Початковий вигляд головної форми
Житловий фонд по районах

- Відобразіть форму в режимі конструктора, вибравши кнопку **Конструктор**  на панелі **Режим** на вкладці **Главная**.
- Зверніть увагу на те, що в області даних наявні тільки поля *Код_району* і *Назва_району*.
- У написі, що розташований у заголовку форми, змініть текст *Райони* на *Житловий фонд по районах*.
- Збережіть форму, давши їй ім'я *Райони*. Зовнішній вигляд форми в режимі конструктора подано на рисунку 4.62.
- Закрийте форму *Райони*.

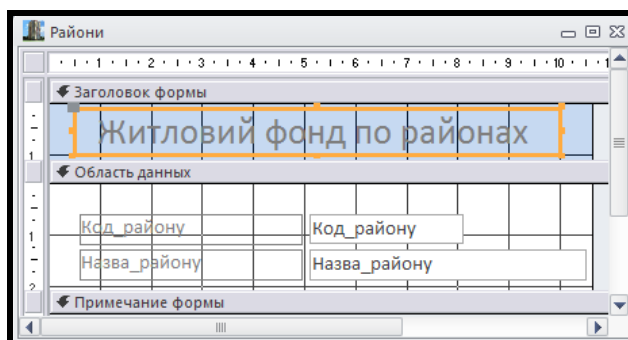


Рисунок 4.62 – Форма *Райони* у режимі конструктора

Створення підпорядкованої форми

Завдання 1

Для форми *Житловий фонд по районах* створити підпорядковану форму *Житловий фонд по районахII* (рис. 4.63).

 3 | 1930 | 115,80 |1
 вулиця | 2 | Героїв праці | 12 | 2 | Панелі | | 12 | 1976 | 225,10 |1
 вулиця | 2 | Героїв праці | 12a | 4 | Залізобето | | 9 | 1977 | 35,00 |1
 вулиця | 3 | Блюхера | 44 | 2 | Панелі | | 9 | 1978 | 67,00 |5
 проспект | 4 | Леніна | 135 | 4 | Залізобето | | 24 | 2010 | 250,00 |5
 проспект | 4 | Леніна | 25 | 1 | Цегла | | 3 | 1950 | 30,00 |

 The status bar at the bottom shows 'Запис: 1 из 6' and 'Нет фильтра'."/>

Рисунок 4.63 – Вигляд підпорядкованої форми в режимі макета

Виконання



Підпорядковану форму краще будувати на базі спеціально створеного підсумкового запиту *Житловий фонд по районахII*, оскільки в нього додано всі потрібні поля (табл. 4.20).

- За полем *Загальна_площа* оберіть підсумкову операцію **Sum**.
- Виділіть запит *Житловий фонд по районахII* в області **Все объекты Access**.
- На вкладці **Создание** на панелі **Формы** виберіть елемент **Несколько элементов** (рис. 4.64). Відповідний майстер побудує стрічкову форму.

Таблиця 4.20 – Поля для запиту *Житловий фонд по районахП*

Таблиця	Поле
Райони	Код району
Вулиці	Код вулиці
Вулиці	Тип вулиці
Вулиці	Назва вулиці
Будівлі	Номер будівлі
Матеріал	Код матеріалу
Матеріал	Назва матеріалу
Будівлі	Наявність ліфту
Будівлі	Кількість поверхів
Будівлі	Рік будовання
Квартири	Загальна площа

Виконання

- Виділіть таблицю **Райони** в області **Все объекты Access**.
- На вкладці **Создание** на панелі **Формы** натисніть кнопку **Мастер форм** . На першому кроці майстра виберіть усі поля з таблиці **Райони** за допомогою кнопки . На другому кроці виберіть зовнішній вигляд форми в «один стовпець». На третьому кроці назвіть форму **Райони**.

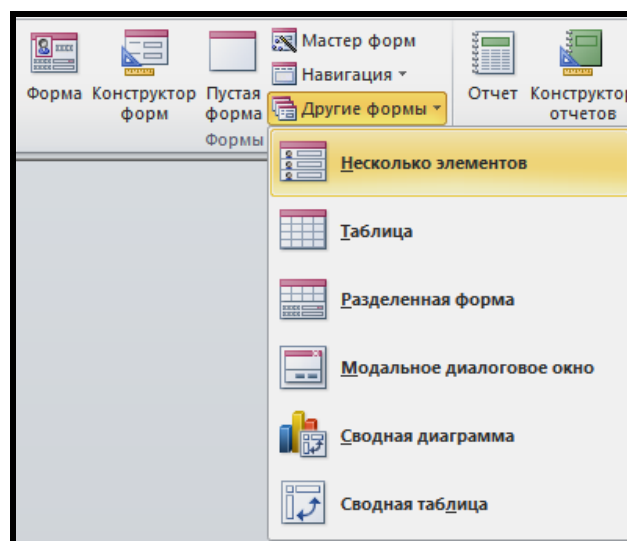


Рисунок 4.64 – Побудова стрічкової форми майстром **Несколько элементов**

- Збережіть форму, давши їй ім'я *Житловий фонд по районахП*.

Завдання 2

Замінити поля *Назва_вулиці* та *Назва_матеріалу* відповідним полем зі списком. Це полегшить надалі додавання нових даних (рис. 4.65).

Рисунок 4.65 – Підпорядкована форма з полем зі списком **Вулиця** та **Матеріал**

Виконання

Видаліть два стовпці **Код_вулиці** і **Назва_вулиці**. Для цього:

- У режимі конструктора виділіть у заголовку форми написи **Код_вулиці** і **Назва_вулиці**.
- Клацніть на одному з виділених об'єктів правою клавiшею миші і виберіть команду **Удалить** з контекстного меню.

В області даних створіть поле зі списком **Вулиця** на базі таблиці **Вулиці**. Для цього:


- На вкладці стрічки Access **Конструктор** на панелі **Элементы управления** натисніть кнопку **Поле со списком**, а потім клацніть на вільному місці в області даних. З'явиться вікно майстра **Создание полей со списком**.

- Відповідаючи на запитання майстра, вкажіть, що базовою є таблиця **Вулиці**, а з неї виберіть поля **Код_вулиці** і **Назва_вулиці**, результат зберігатиметься в полі **Код_вулиці**. У кінці введіть текст напису **Вулиця**.

- Перетягніть напис **Вулиця** з області даних у заголовок форми, розмістивши його над новим полем зі списком.

- Закрийте форму **Житловий фонд по районахП** зі збереженням зроблених змін.

- Відкрийте знову форму **Житловий фонд по районахП**, двічі клацнувши на її значку в області переходів. Зверніть увагу на те, що в новому полі зі списком відображаються назви вулиць, а не їхні коди.

- Перейдіть у режим макета, натиснувши кнопку **Режим**  на панелі **Режимы** на вкладці **Главная**.

- Клацніть на будь-якій клітинці стовпця **Вулиця** і перетягніть його

праву межу праворуч, щоб назви вулиць відображалися повністю.

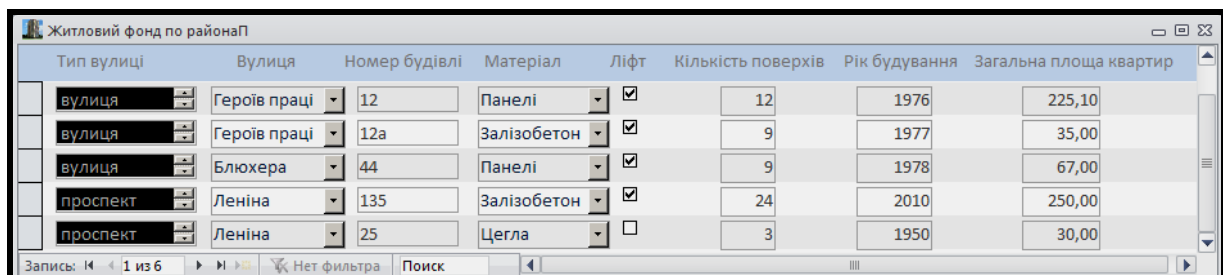
– Збережіть зроблені на формі зміни.

Поле *Назва_матеріалу* замініть відповідним полем зі списком за аналогією із попереднім полем.

Завдання 3

На підпорядкованій формі видалити непотрібні написи та поля *Код_вулиці*, *Назва_вулиці*, *Код_матеріалу*, *Назва_матеріалу*, а також емблему й напис із назвою форми у заголовку форми (рис. 4.66).

ПРИМІТКА. Не можна видаляти поле *Код_району*, оскільки воно слугує для зв'язку з головною формою, але потрібно зробити це поле разом із написом невидимим.



Тип вулиці	Вулиця	Номер будівлі	Матеріал	Ліфт	Кількість поверхів	Рік будування	Загальна площа квартир
вулиця	Героїв праці	12	Панелі	<input checked="" type="checkbox"/>	12	1976	225,10
вулиця	Героїв праці	12а	Залізобетон	<input checked="" type="checkbox"/>	9	1977	35,00
вулиця	Блюхера	44	Панелі	<input checked="" type="checkbox"/>	9	1978	67,00
проспект	Леніна	135	Залізобетон	<input checked="" type="checkbox"/>	24	2010	250,00
проспект	Леніна	25	Цегла	<input type="checkbox"/>	3	1950	30,00

Рисунок 4.66 – Підпорядкована форма після видалення елементів

Виконання

Перебуваючи в режимі макета форми *Житловий фонд по районахП*, видаліть стовпці *Код_вулиці*, *Назва_вулиці*, *Код_матеріалу*, *Назва_матеріалу* як це описано у попередньому завданні.

Зробіть невидимими поле *Код_району* і напис до нього. Для цього:

– Перейдіть у режим конструктора, зменшіть розмір поля *Код_району* разом із написом приблизно до 0,5 см і пересуньте їх на правий кінець форми.

– Відкрийте вікно властивостей цих елементів, перед цим виділивши їх, і у вкладці **Макет** встановіть для властивості **Вивод на екран** значення **Нет**.

Скоротіть висоту заголовка форми, залишивши в ньому тільки назви полів. Для цього:

– Перейдіть у режим конструктора. Клацніть на емблемі в заголовку форми, а потім при натиснутій клавіші **Shift** – на написі з назвою

форми праворуч, щоб виділити їх. Після цього видаліть елементи, натиснувши клавішу **Delete**.

- Виділіть усі написи в заголовку форми, клацнувши вертикальну лінійку навпроти цих написів, і перетягніть до верхнього краю всі написи.

- Установіть покажчик миші на межі між заголовком форми й областю даних, після чого перетягніть межу вгору до написів.

- Зменшіть висоту області даних приблизно до 0,75 см подібно до того, як це описано в попередньому пункті.

- Змініть назви полів (поля чіпати не потрібно) – видаліть нижні підкреслення між словами, назвіть їх зрозуміліше.

На рисунку 4.67 подано підпорядковану форму в режимі конструктора.

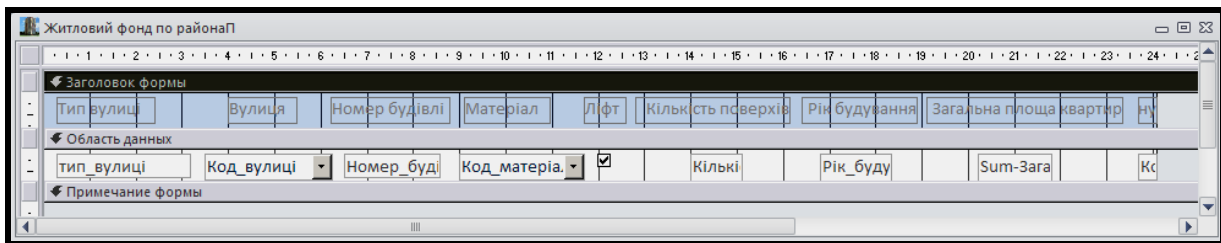


Рисунок 4.67 – Підпорядкована форма в режимі конструктора

- Збережіть зроблені зміни на формі.
- Відкрийте форму в режимі форми, перегляньте записи та закрийте її.

Завдання 4

В області примітки підпорядкованої форми додати поле для обчислення суми загальних площ квартир у районі (рис. 4.68).

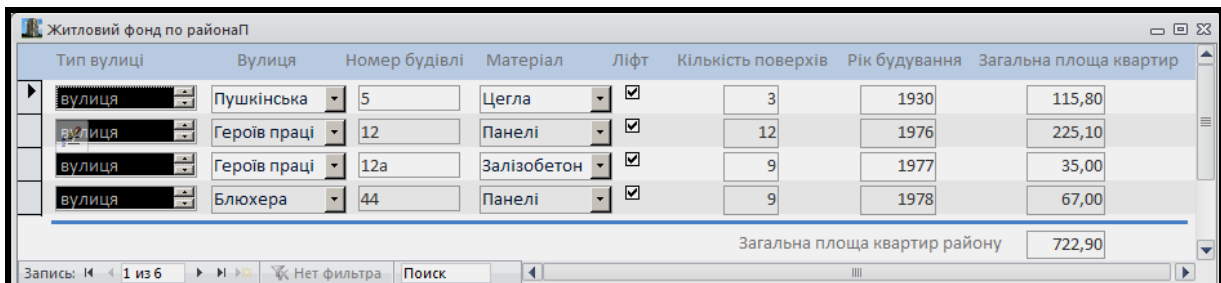



Рисунок 4.68 – Підпорядкована форма з полем

Загальна площа квартир району

Виконання

Відкрийте форму **Житловий фонд по районахП** у режимі конструктора, встановіть покажчик миші на нижній межі області примітки форми та розтягніть її вниз приблизно на 1–1,5 см.

На панелі елементів натисніть кнопку **Поле**  на панелі **Элементы управления** на вкладці **Конструктор**, а потім – на примітці форми під полем **Площа квартир будинку**.

Змініть текст напису нового поля, ввівши слово **Загальна площа квартир району**.

Вставте функцію для обчислення суми. Для цього:

- Клацніть на новому полі та відкрийте вікно його властивостей.
- У вкладці **Данные** клацніть на рядку **Данные**, а потім – на кнопці виклику будівника виразів.

- У вікні будівника виразів виберіть у лівому списку вид об'єкта **Функции – Встроенные функции**, в середньому списку – категорію функцій **Статистические**, а в правому списку – двічі клацніть на імені функції **Sum**.

- Замініть текст «**expression**» як аргумент функції ім'ям поля **[Sum-Загальна_площа]**. Виберіть його із середнього списку **Категории выражений** (рис. 4.69).

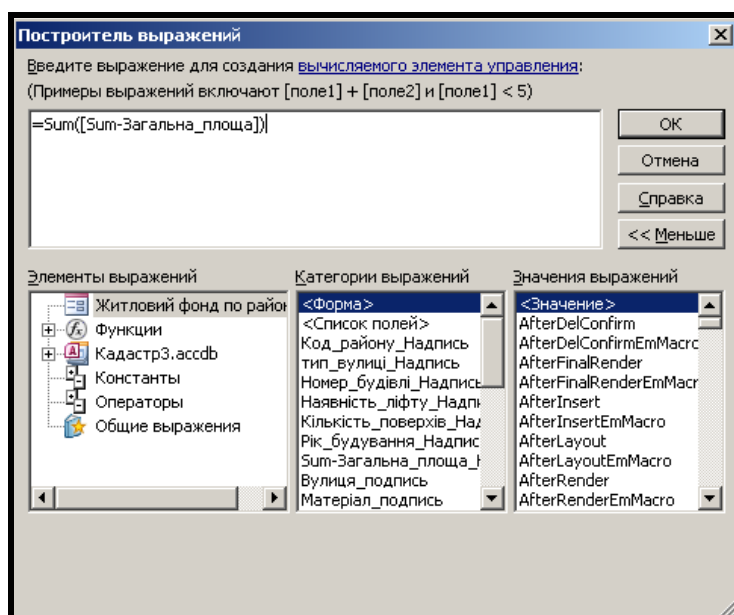


Рисунок 4.69 – Побудова функції **Sum**

- Клацніть кнопку **ОК**.

У вкладці **Макет** вікна властивостей установіть такі значення (табл. 4.21):

Таблиця 4.21 – Значення властивостей у вкладці **Макет**

Властивість	Значення
Формат поля	С разделителями разрядов
Число десятичных знаков	2

На рисунку 4.70 подано підпорядковану форму з полем **Загальна площа квартир району** у режимі конструктора.

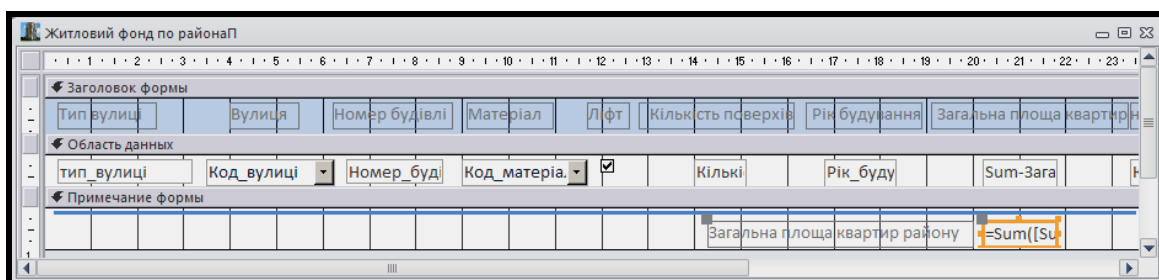


Рисунок 4.70 – Поле **Загальна площа квартир району** на підпорядкованій формі

Збережіть зроблені зміни на формі. Відкрийте форму в режимі форми, перегляньте записи, звернувши увагу на поле **Загальна площа квартир району**, і закрийте її.

Об'єднання підпорядкованої форми з головною

Завдання 1

Помістити підпорядковану форму на головну, зв'язавши їх за допомогою поля **Код_району** (рис. 4.71).

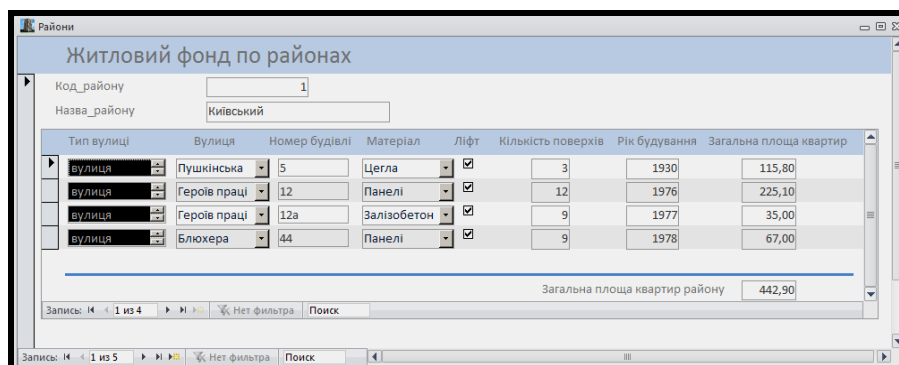




Рисунок 4.71 – Попередній вигляд форми **Житловий фонд по районах**

Виконання

Відкрийте головну форму *Райони* у режимі конструктора та збільшіть розмір області даних, щоб можна було помістити підпорядковану форму.

Натисніть кнопку **Дополнительные параметры** на панелі **Элементы управления** на вкладці **Конструктор** і на розширеній панелі елементів клацніть на кнопці **Подчиненная форма/отчет** , а потім – на формі ліворуч під написом *Назва_району*. Простежте, щоб при цьому знаходилася в натиснутому стані кнопка **Использовать мастера** .

– У першому вікні майстра вкажіть ім'я підпорядкованої форми, включивши перемикач **Имеющиеся формы** і вибравши у списку ім'я *Житловий фонд по районахП*. Натисніть кнопку **Далее**.

– У другому вікні майстра вкажіть поля зв'язків між формами. Для цього включіть перемикач **Самостоятельное определение** і встановіть в обох полях зі списком **Поля формы или отчета** і **Поля подчиненной формы или отчета** поле *Код_району* (рис. 4.72). Натисніть кнопку **Далее**.

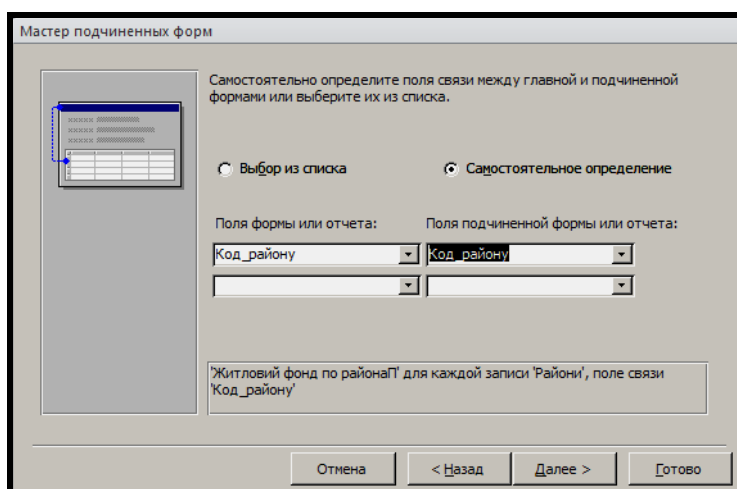


Рисунок 4.72 – Вибір поля зв'язку форм

– В останньому вікні майстра погодьтеся з ім'ям підпорядкованої форми *Житловий фонд по районамП* і натисніть кнопку **Готово**.

– Перетягніть нижній маркер підпорядкованої форми майже до кінця області даних головної форми. Це дасть змогу одночасно відображати більше записів на підпорядкованій формі (рис. 4.73).

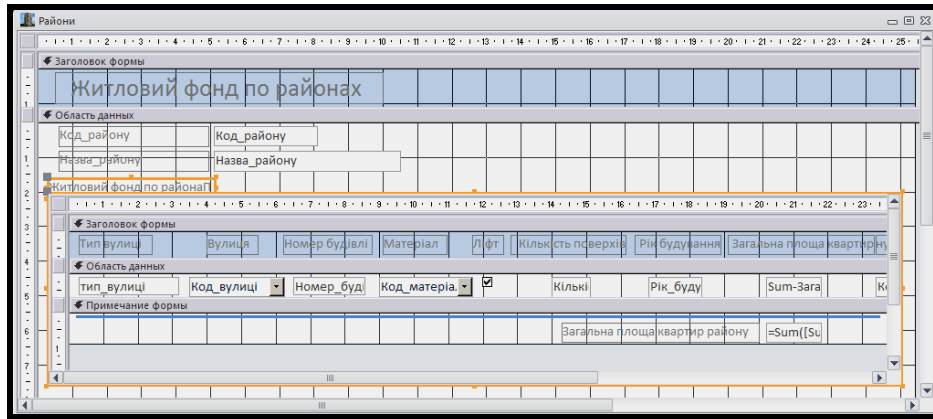


Рисунок 4.73 – Підпорядкована форма на головній формі у режимі конструктора

- Видалить напис над підпорядкованою формою.
- Збережіть комбіновану форму.

Відкрийте форму *Житловий фонд по районах* у режимі форми і перегляньте дані за житловим фондом по різних районах. Для переходу до іншого району скористайтеся кнопками переходу на головній формі.

Завдання 2

Додати поле зі списком для переходу записами на головній формі (рис. 4.74).

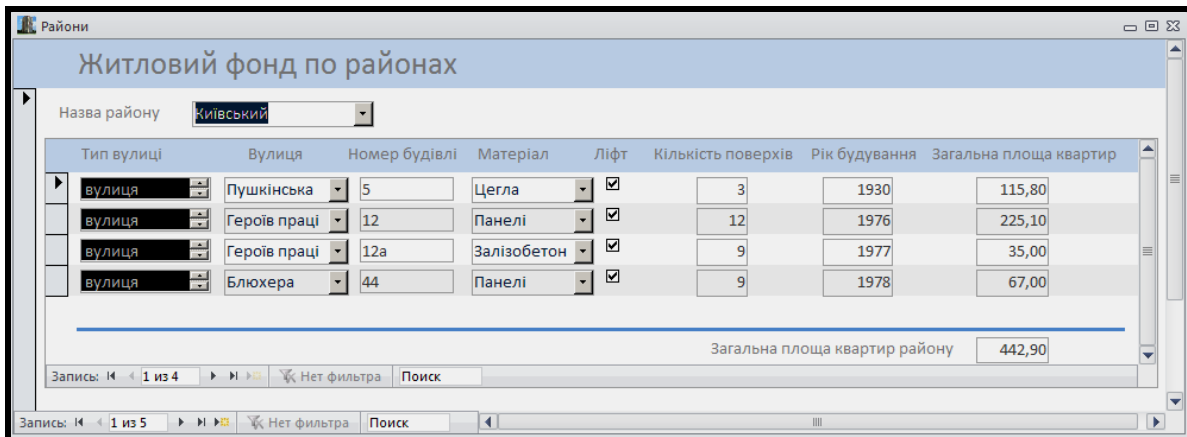




Рисунок 4.74 – Комбінована форма *Житловий фонд по районах* з полем зі списком

Щоб вибрати дані про продаж товарів якогось конкретного виробника, зручніше користуватися не кнопками переходу записами, а полем зі списком *Назва району*. Воно замінить два поля *Код_ району* та *Назва району*.

Виконання

– Відобразіть комбіновану форму *Житловий фонд по районах* у режимі конструктора, вибравши кнопку **Конструктор**  на панелі **Режим** на вкладці **Главная**.

– Видаліть із головної форми поля *Назва_району* і *Код_району*.

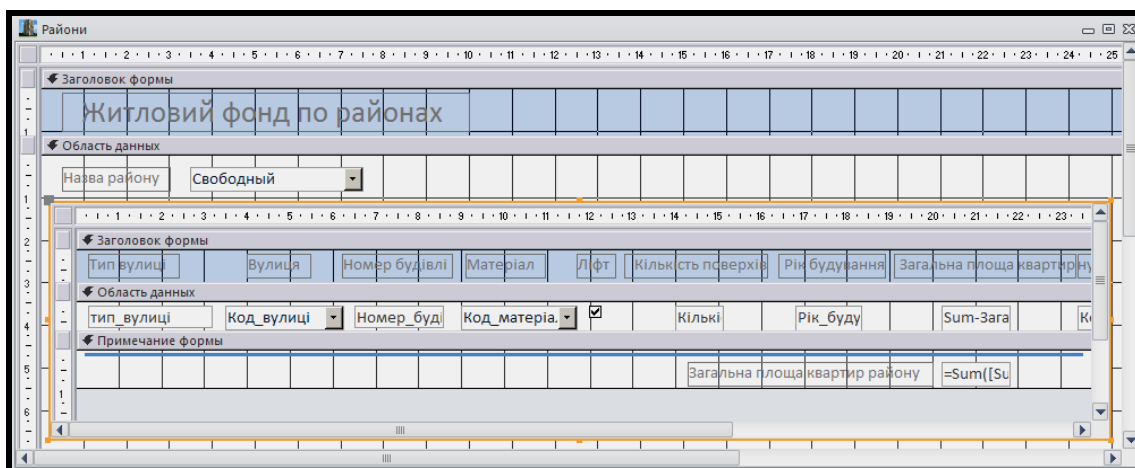
– Натисніть кнопку **Дополнительные параметры**  у групі **Элементы управления** на панелі **Конструктор** і на розширеній панелі елементів клацніть на кнопці **Поле со списком**, а потім – на верхній частині головної форми. Простежте, щоб при цьому знаходилася в натиснутому стані кнопка **Использовать мастера**.

– У першому вікні майстра, що з'явилося, виберіть перемикач **Поиск записи в форме на основе значения, которое содержит поле со списком** і натисніть кнопку **Далее**.

– У другому вікні майстра виберіть два поля *Назва_району* та *Код_району* і натисніть кнопку **Далее**.

– В останньому вікні погодьтеся з ім'ям *Назва району*.

На рисунку 4.75 подано комбіновану форму з полем зі списком *Назва району* у режимі конструктора.



Рисунку 4.75 – Комбінована форма *Житловий фонд по районах* з полем зі списком *Назва району* у режимі конструктора

Збережіть зроблені зміни на формі. Відкрийте форму в режимі форми, перегляньте дані про стан житлового фонду в різних районах міста, користуючись полем зі списком *Назва району*. Закрийте форму *Житловий фонд по районах*.

Побудова діаграми

Завдання

Побудувати на головній формі об'ємну кругову діаграму. На ній подати відсотковий вклад кожного будинку в загальну площу квартир району. Дані, що відображаються на діаграмі, змінюються залежно від району, який вибирається в полі зі списком (рис. 4.76).

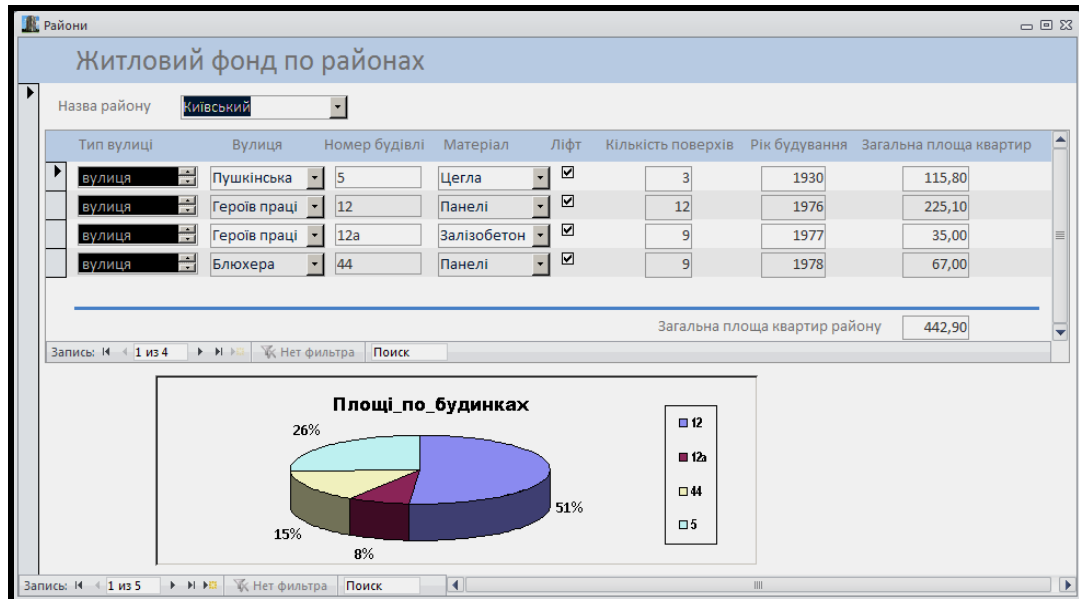


Рисунок 4.76 – Комбінована форма *Житловий фонд по районах* із круговою діаграмою

Основні етапи виконання

Діаграма, як і підпорядкована форма, зв'язується з головною формою за допомогою поля *Код_району*.

Найпростіше додати діаграму на форму за допомогою майстра діаграм. При цьому необхідно заздалегідь потурбуватися про джерело даних (таблицю або запит). У ньому повинні бути поля, значення яких відображаються на діаграмі, а також поля зв'язку з головною формою.

Як джерело даних для діаграми створимо підсумковий запит *Площі по будинках*, у який додамо всі необхідні поля – *Номер_будівлі* і *Загальна_площа_з_балконом* (групова операція – **Sum**) – для відображення на формі, а також *Код_району* – для зв'язку з головною формою.

Створення діаграми виконується у два етапи:

1. Створення базової діаграми майстром.
2. Доопрацювання діаграми.


Створення базової діаграми майстром

Завдання

Додати кругову діаграму на форму *Житловий фонд по районах*.

Виконання

– Відкрийте форму *Житловий фонд по районах* у режимі конструктора і збільшіть розмір області даних у вертикальному напрямі для розміщення діаграми.

– Натисніть кнопку **Дополнительные параметры**  у групі **Элементы управления** на вкладці **Конструктор** стрічки Access і на розширеній панелі елементів клацніть на кнопці **Диаграмма** (рис. 4.77), а потім – на формі знизу від підпорядкованої форми. Простежте, щоб при цьому знаходилася в натиснутому стані кнопка **Использовать мастера**

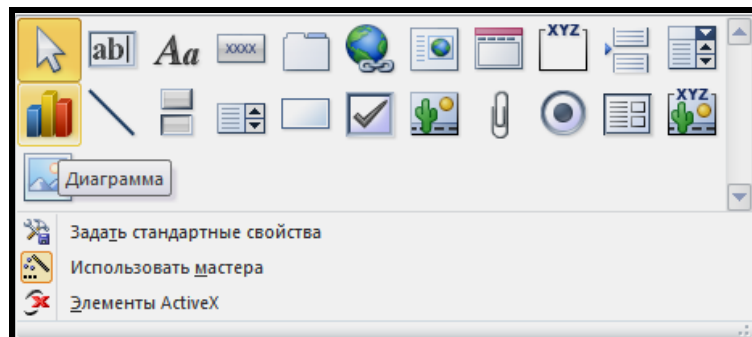


Рисунок 4.77 – Кнопка **Диаграмма** на розширеній панелі елементів

– У першому вікні майстра **Создание диаграмм** виберіть запит *Площі по будинках*, на базі якого буде створюватися діаграма. Натисніть кнопку **Далее**.

– У другому вікні майстра виберіть усі поля із запиту.

– У третьому вікні майстра виберіть тип діаграми, наприклад **Объемная круговая** (рис. 4.78).

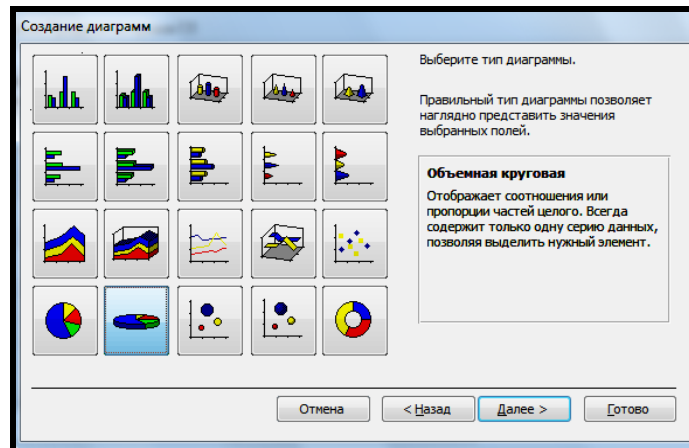


Рисунок 4.78 – Вибір типу діаграми *Объемная круговая*

– У четвертому вікні майстра можна змінювати дані, які відображаються на діаграмі. У цьому разі достатньо натиснути кнопку **Далее**.

– У п'ятому вікні майстра можна вибрати поля зв'язку головної форми з діаграмою. У цьому разі майстер автоматично вибрав поле **Код району**, тому достатньо натиснути кнопку **Далее**.

– В останньому вікні майстра введіть назву діаграми, наприклад **Площі по будинках**, і натисніть кнопку **Готово**.

На рисунку 4.79 подано комбіновану форму з діаграмою в режимі конструктора.

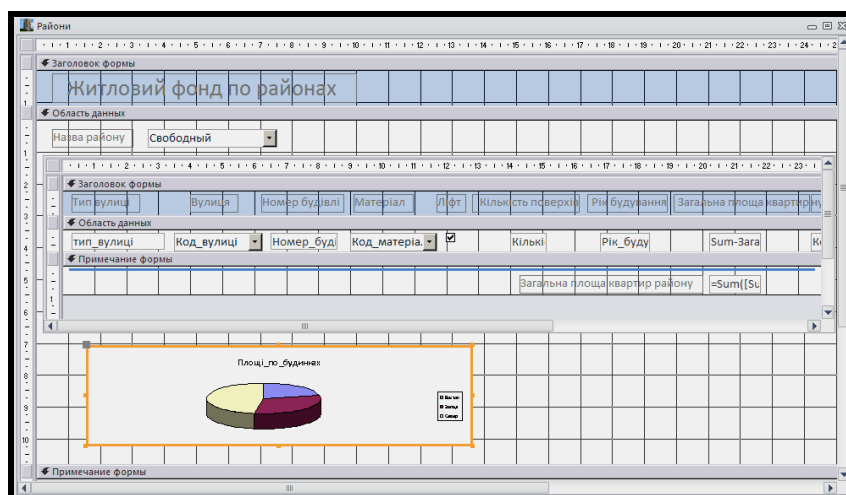


Рисунок 4.79 – Комбінована форма з діаграмою в режимі конструктора

Збережіть зроблені зміни на формі. Відкрийте форму в режимі форми, перегляньте дані про житловий фонд різних районів, звернувши увагу на зміну діаграми.

Доопрацювання діаграми

Завдання

Додати на діаграму значення у відсотках вкладу кожної будівлі в загальну площу житлового фонду району та збільшити розмір шрифту легенди.

Виконання

- Відкрийте форму *Житловий фонд по районах* у режимі конструктора.
- Двічі клацніть ЛКМ на діаграмі. Поряд із формою з'явиться таблиця з даними діаграми, які можна змінювати (рис. 4.80). Меню та панель інструментів Access заміняться відповідними об'єктами **Microsoft Graph Chart**.
- У меню **Диаграмма** виберіть команду **Параметры диаграммы**.
- На вкладці **Подписи данных** встановіть прапорець **Доли**, а потім натисніть кнопку **ОК** (рис. 4.81).
- Виділіть легенду діаграми й установіть розмір шрифту 10 пт, використавши панель форматування.
- Щоб вийти з режиму редагування діаграми, клацніть на будь-якому вільному місці форми.

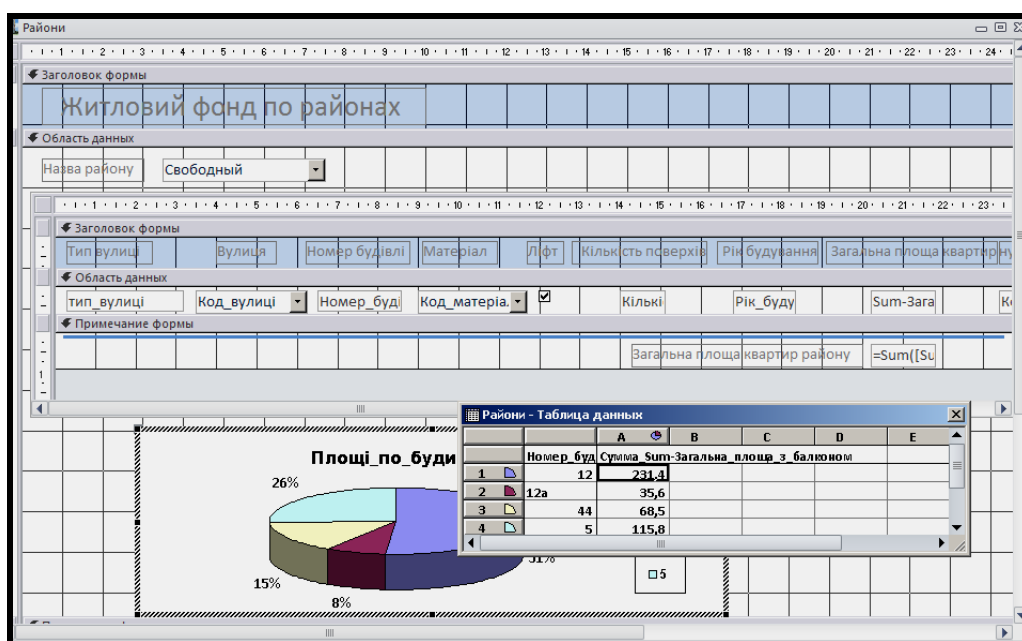


Рисунок 4.80 – Вікно Access з меню **Диаграмма**

Збережіть змінену форму та перейдіть у режим форми.

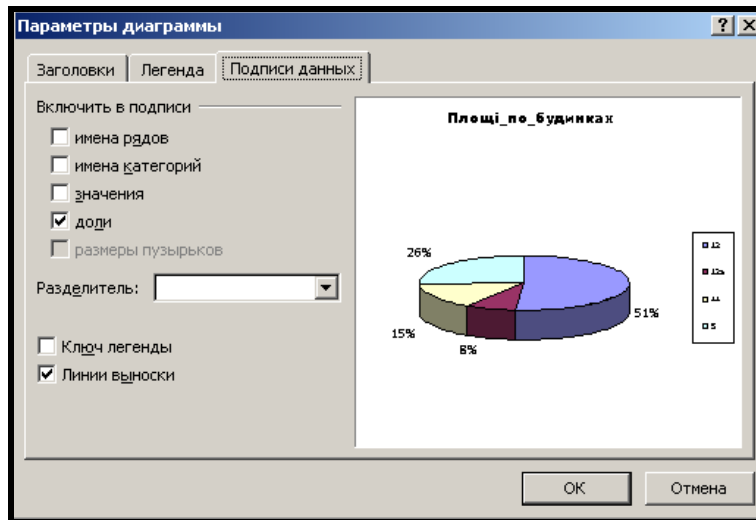


Рисунок 4.81 – Встановлення прапорця Доли

- Вибираючи різні райони міста у полі зі списком *Назва району*, порівняйте структуру розподілу площ по будинках.
- Закрийте форму *Житловий фонд по районах*.

Побудова графіка з лінією тренда

Дослідження змін обсягів за часом проводять за допомогою графіків. При цьому на горизонтальній осі відкладають тимчасові інтервали, а на вертикальній – значення досліджуваного обсягу.

На практиці для цих значень характерні випадкові коливання. Тому часто графік має складний вигляд. Щоб виявити загальну тенденцію досліджуваного процесу, графік замінюють простішим – лінією тренда. Вона приблизно відображає загальну тенденцію змін ряду обсягів. За допомогою лінії тренда можна робити прогноз за умови, що встановлена тенденція збережеться надалі.

Завдання

Побудувати на головній формі *Житловий фонд по районах* графік динаміки будівництва житлового фонду. На нього додати лінію тренда для визначення тенденцій. Дані, які будуть відображатися на графіці, змінюються залежно від району, який вибирається в полі зі списком (рис. 4.82).

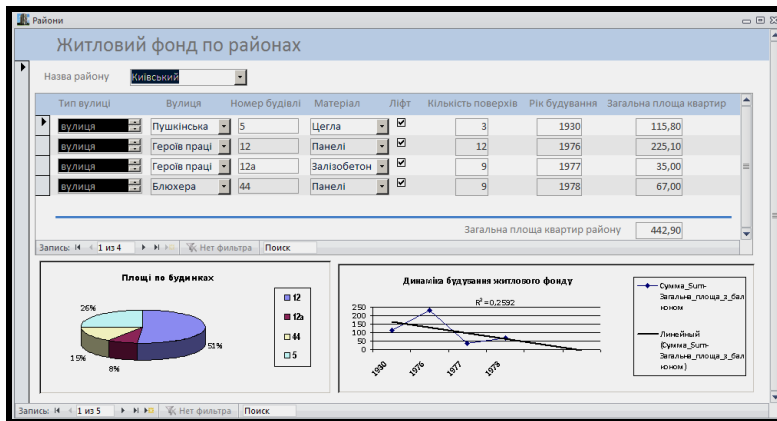


Рисунок 4.82 – Форма з графіком динаміки будівництва житлового фонду

Основні етапи виконання

Графік, як і діаграма та підлегла форма, зв'язується з головною формою за допомогою поля **Код_району**. Подібно до діаграми він будується на формі за допомогою майстра діаграм. Потім на нього додається лінія тренда.

Створення графіка виконується у два етапи:

1. Побудова графіка.
2. Додавання лінії тренда.

Побудова графіка

Завдання

Побудувати на головній формі **Житловий фонд по районах** графік динаміки будівництва житлового фонду. Дані, які будуть відобразитися на ньому, відповідають тому району, який вибирається в полі зі списком (рис. 4.83).

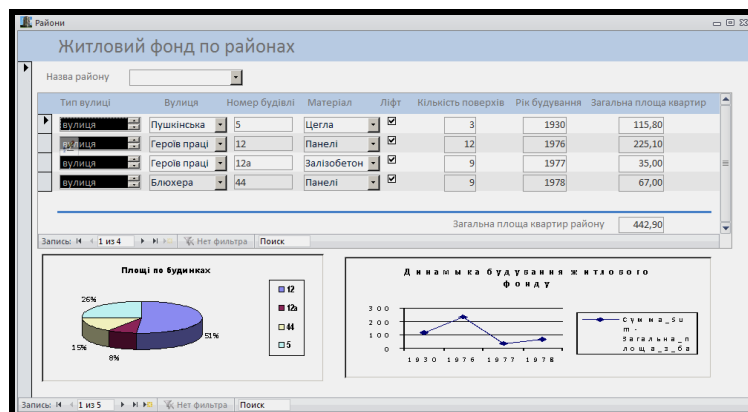



Рисунок 4.83 – Форма зі створеним графіком

Виконання

– Відкрийте головну форму *Житловий фонд по районах* у режимі конструктора та збільшіть розмір вікна та області даних для розміщення графіка (праворуч від діаграми).

– Натисніть кнопку **Дополнительные параметры**  у групі **Элементы управления** на вкладці **Конструктор** стрічки Access і на розширеній панелі елементів клацніть на кнопці **Диаграмма** і намалюйте мишею прямокутник на головній формі в тому місці, де буде розташовано графік. Далі включається в роботу майстер діаграм.

– У першому вікні майстра створення діаграм виберіть підсумковий запит *Площі по будинках*, у якому є всі необхідні поля – *Номер_будівлі*, *Загальна_площа_з_балконом*, *Рік_будування* – для відображення на формі, а також *Код_району* для зв'язку з головною формою, на базі якої створюється діаграма.

– У другому вікні майстра виберіть поля *Загальна_площа_з_балконом*, *Рік_будування*, *Код_району*.

– У третьому вікні майстра виберіть тип діаграми **График** (рис. 4.84).

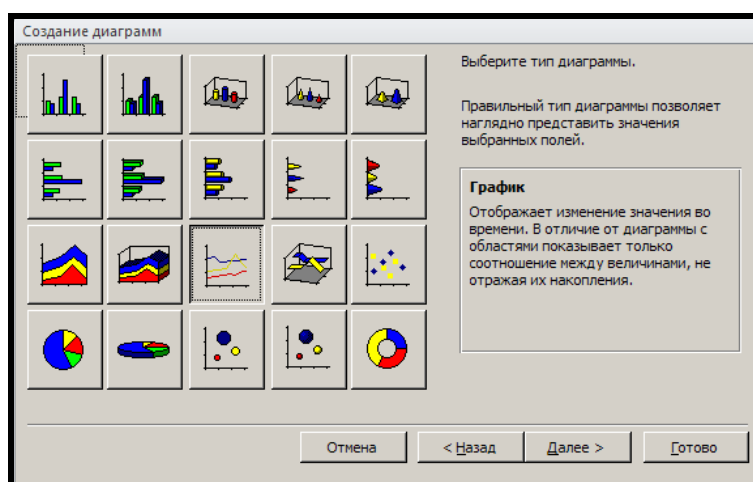


Рисунок 4.84 – Вибір типу діаграми

– У четвертому вікні майстра двічі клацніть на кнопці *Рік_будування*, щоб змінити тип групування, оскільки в даному випадку групування за роками дуже дрібне. Для того щоб вказати, по яких саме полях формувати осі графіка перетягніть ЛКМ назву поля на вісь графіка (рис. 4.85).

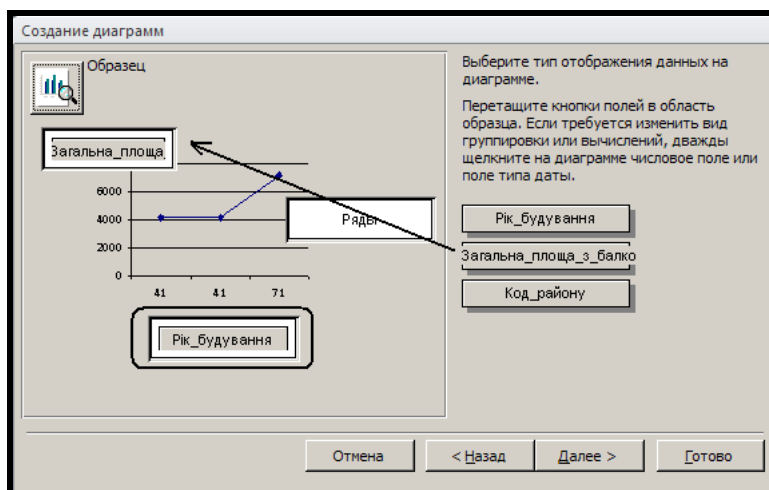


Рисунок 4.85 – Формування графіка

- У п'ятому вікні майстра можна вибрати поля зв'язку головної форми з діаграмою. У цьому випадку майстер автоматично вибрав поле **Код_району**, тому достатньо клацнути на кнопці **Далее**.
- В останньому вікні майстра введіть назву діаграми **Динаміка будування житлового фонду**.
- Збережіть змінену форму та перейдіть у режим відображення форми.
- Вибираючи різні райони в полі зі списком **Назва району**, порівняйте графіки динаміки будування житлового фонду по районах міста.

Додавання лінії тренда

Завдання

Додати на графік **Динаміка будування житлового фонду** лінію тренда з прогнозом на два роки вперед. Тут же відобразити значення достовірності наближення (апроксимації) ряду даних.

Виконання

- Поверніться в режим конструктора та двічі клацніть на області графіка. Поряд із формою з'явиться таблиця з даними графіка, які можна змінювати, а меню та панель інструментів вікна Access заміняться відповідними об'єктами **Microsoft Graph Chart**.
- Клацніть на графіку, щоб на ньому з'явилися маркери.
- У меню **Диаграмма** виберіть команду **Добавить линию тренда**.

– На вкладці **Тип вікна**, що з'явилося, можна вибрати різні типи ліній тренда. При невеликій кількості даних краще вибрати тип **Линейная** (рис. 4.86), який встановлений за замовчуванням. Тому в цьому випадку відразу перейдіть на вкладку **Параметры**.

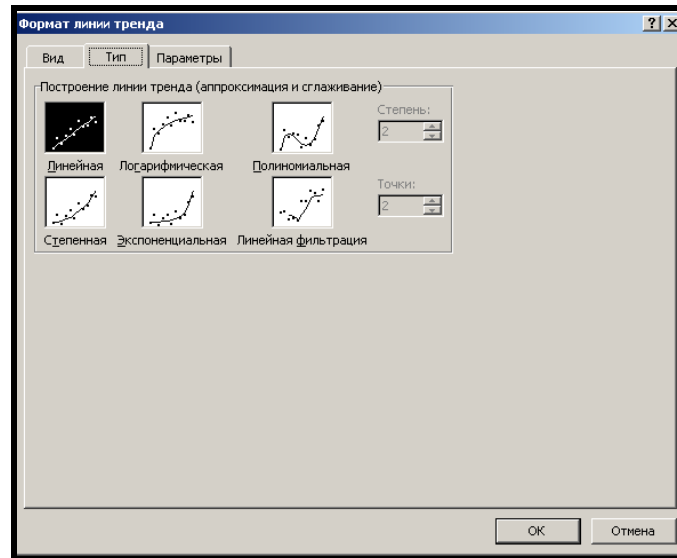


Рисунок 4.86 – Вибір лінійного типу лінії тренда

На вкладці **Параметры** встановіть такі значення:

- Прогноз вперед на 2 періода.
- **Поместить на диаграмму величину достоверности аппроксимации (R^2)** (рис. 4.87).

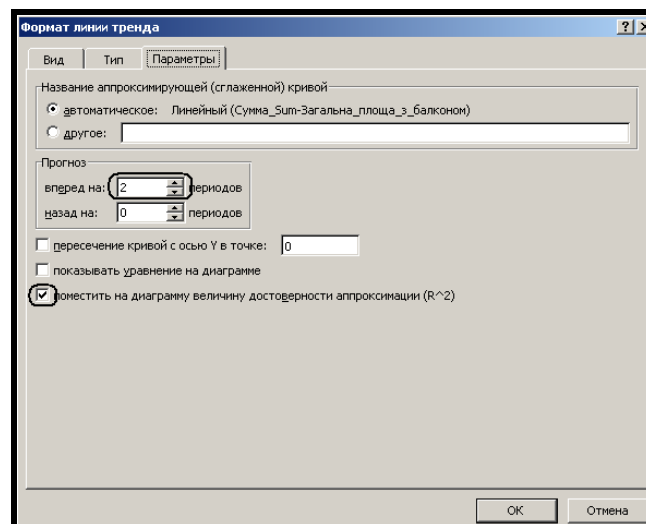


Рисунок 4.87 – Параметры лінії тренда

- Клацніть на кнопці **ОК**.
- Щоб вийти з режиму редагування діаграми, клацніть на будь-

якому вільному місці форми.

- Збережіть змінену форму та перейдіть у режим відображення форми.
- У полі зі списком **Назва району** виберіть по черзі різні назви районів міста. Лінія тренда показує динаміку забудови, але достовірність її не дуже висока, оскільки коефіцієнт $R^2 = 0,2592$ (менше 0,95) (рис. 4.88).



Рисунок 4.88 – Динаміки будівництва житлового фонду

- Закрийте вікно форми **Житловий фонд по районах**, а потім Access.

Завдання для самостійного виконання

1. *Створіть комбіновану форму **Будівлі**. На головній помістіть поле зі списком **Назва вулиці**, в якому відображаються всі вулиці міста (по одному разу), а на підпорядковану – інформацію про всі будинки цієї вулиці (створити підсумковий запит на базі таблиці **Будівлі**, в якому містяться по одному разу всі дані про будівлю, а потім на його основі побудувати підпорядковану форму).
2. *На формі **Будівлі**, за допомогою стовпчикової діаграми з лінією тренда відобразіть динаміку будівництва будинків на вулицях по роках.
3. Побудуйте форму в один стовпець для ведення таблиці **Будівлі**. За її допомогою додайте рисунки до будинків.
4. Створіть форми для заповнення всіх таблиць.
5. Створіть форму для пошуку потрібного документа.

4.6 Практичне заняття. Знайомство з ArcMap

Мета: виробити вміння та навички роботи з зображеннями карти, атрибутивною інформацією, створенню написів до об'єктів карти та підготовки необхідних фрагментів карти до друку.

Призначення: виконавши роботу, навчитися створювати написи до об'єктів карти, встановлювати необхідні масштаби карти, дізнаватися інформацію, що закріплена за об'єктами, підготовлювати компонування карт до друку.

ХІД РОБОТИ

- Увімкніть **ArcMap**.
- На початку роботи з **ArcMap** виберіть **Blank map**. **ArcMap** дозволяє бачити карту у вигляді даних або у вигляді компонування (макета для друку). Документ (файл) карти має розширення **.mxd**.
- Відкрийте карту. Відкрийте файл **Ukraine.mxd** з папки ЛР_1. Коли файл **Ukraine.mxd** відкриється, ви побачите карту України з межами областей та кордоном. Зверніть увагу, як організовані шари карти згідно тематичної інформації. Також зверніть увагу на те, що не всі об'єкти карти в даний момент можна переглянути – шар **Населенні_пункти_Anno** в даний момент не включений (рис. 4.89).

Взаємне розташування географічних об'єктів

Карти представляють географічні об'єкти в їх взаємному розташуванні. Навіть бігло глянувши на карту, ви здатні визначити місце розташування об'єктів один до одного. Подібні взаємні відносини розуміються програмним забезпеченням за допомогою топології (набору правил про взаємне розташування об'єктів).

<p>Задача: необхідно визначити області, що розташовані навколо Харківської.</p>
--

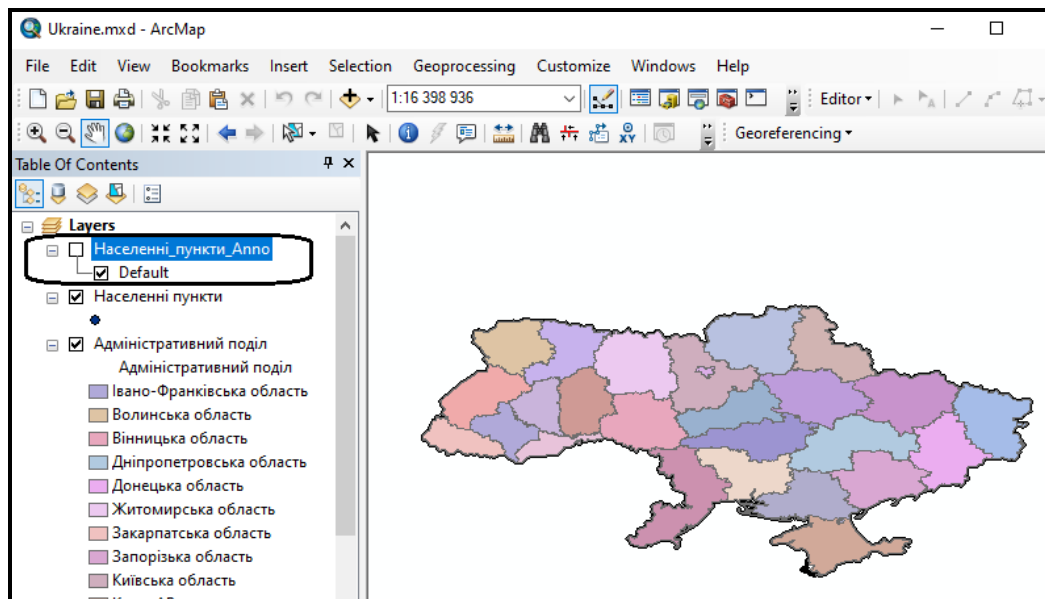


Рисунок 4.89 – Вид фрагмента карти з відключеним шаром Річки

– Для цього необхідно збільшити карту у вікні відображення так, щоб була видна територія Харківської області. Зручно скористатися для цього заздалегідь створеної закладкою, яка зберігає певний географічний вид.

– Для переходу до виду по закладці виберіть в рядку меню команду **Bookmarks / Харківська обл.** Зображення карти збільшиться і покаже ту частину карти, де представлена Харківська область і її околиці. Зверніть увагу, що написи (назви) областей відображаються при певному масштабі карти.

Задача: необхідно визначити які адміністративні одиниці оточують Харківську область, а також яка з областей – Полтавська або Сумська має більшу велику площу території.

– Для цього необхідно виділити Харківську область на карті. Вибір об'єктів можна виконати за допомогою вибору по атрибутам (меню **Selection / Select By Attributes**).

– У вікні вибору по атрибутам створіть запит. **Layer** – **Адміністративний поділ**, метод – **Create a new selection**. Для створення самої умови запиту у вікні вибору скористайтесь переліком полів – виберіть поле **NAME**, знак присвоїти «=**»** і за допомогою кнопки **Get Unique Values** виберіть **Харківська область** (рис. 4.90).

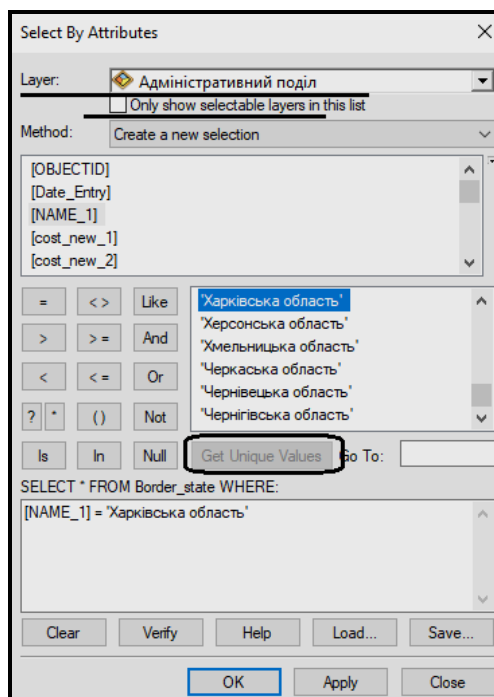


Рисунок 4.90 – Вигляд окна вибору з умовою

– Натиснувши на кнопку **OK** застосуйте і закрийте вікно вибору. В результаті на карті контури Харківської області стануть виділеними блакитним кольором.

– Вивчіть значення поля **Aria** у атрибутивній таблиці для Харківської області. Саме в цьому полі знаходиться значення площі адміністративної області.

Зв'язок описової (атрибутивної) інформації з зображенням на карті

Кожен об'єкт на карті має деякі властивості, тобто пов'язаний з певним рядком в атрибутивній таблиці, що містить інформацію про цей об'єкт карти.

– Для того щоб відкрити атрибутивну таблицю шару **Адміністративний поділ** необхідно в таблиці вмісту викликати контекстне меню на шарі і в випадаючому меню вибрати команду **Open Attribute Table** (рис. 4.91).

В результаті з'явиться атрибутивна таблиця, пов'язана з шаром. Кожен запис (рядок) в таблиці представляє окремий об'єкт на шарі **Адміністративний поділ**. У нижній частині таблиці відображається інформація про кількість записів в таблиці (4.92).

– Синім кольором виділений запис з обраною Харківською областю. Зверніть увагу на горизонтальну і вертикальну смуги прокрутки.

Перегляньте всі поля і всі записи атрибутивної таблиці з її допомогою.

– За допомогою горизонтальної полоси прокрутки знайдіть поле **Shape Area** та з’ясуйте які площі Полтавської та Сумської областей.

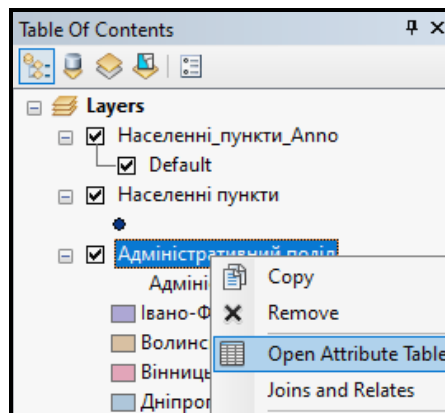


Рисунок 4.91 – Процес виклику атрибутивної таблиці

OBJECTID *	Shape *	Актуальна інформація на:	Адміністративний поділ	Вартість 1.
18	Polygon	<Null>	Полтавська область	
19	Polygon	<Null>	Рівненська область	
20	Polygon	<Null>	Сумська область	
21	Polygon	<Null>	Тернопільська область	
8	Polygon	<Null>	Харківська область	
9	Polygon	<Null>	Херсонська область	<Null>
10	Polygon	<Null>	Хмельницька область	
1	Polygon	<Null>	Черкаська область	
3	Polygon	<Null>	Чернівецька область	
2	Polygon	<Null>	Чернігівська область	

Рисунок 4.92 – Вигляд атрибутивної таблиці

Класифікація та присвоєння символів просторовим даним

Задача: Більш зручно з точки зору аналізу розміру області відобразити населені пункти чорними символами різного розміру в залежності від площі адміністративної області.

– За допомогою контекстного меню на імені шару **Адміністративний поділ** відкрийте вікно **Properties**.

– У вікні **Layer Properties** перейдіть на вкладку **Symbology** і заповніть всі поля, як показано на малюнку (рис. 4.93).

– Колір градувальних символів змініть на чорний. Для цього натисніть на кнопку **Template** і внесіть відповідні зміни.

– Після всіх змін клацніть кнопку **ОК**. На карті будуть зображені міста, класифіковані за площею області.

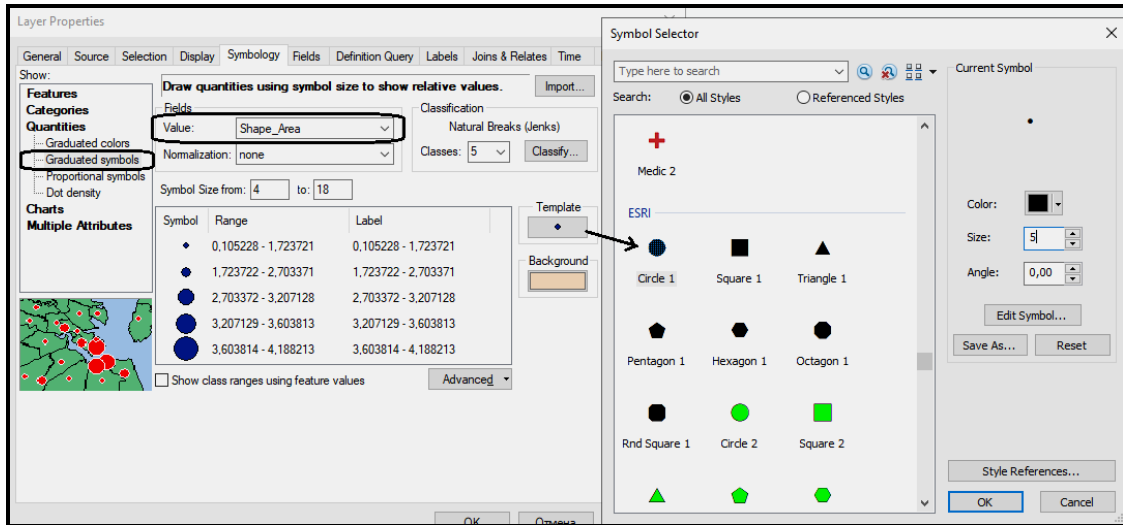


Рисунок 4.93 – Процес зміни градування символів

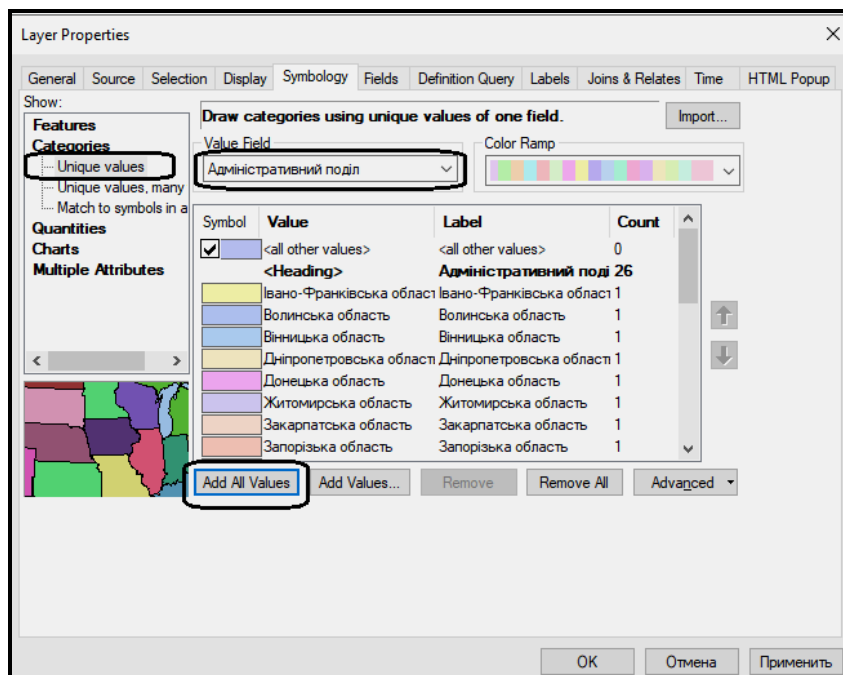


Рисунок 4.94 – Процес повернення до колишнього зображення шару

– Для того щоб зберегти класифікацію об'єктів в окремому шарі необхідно викликати контекстне меню на шарі **Адміністративний поділ** і вибрати команду **Save as layer File**, присвоївши шару ім'я **Площі областей.lyr**. Створений новий шар **Площі областей**, збережений як окремий файл з розширенням **lyr** дозволить використовувати створену


класифікацію об'єктів шару в будь-яких інших картах.

– Поверніться до колишнього зображення кольору областей на шарі **Адміністративний поділ**. Для цього у вікні **Layer Properties** виберіть категорію **Unique values** і виконайте налаштування як на рисунку 4.94.

Створення написів до об'єктів

– Передусім потрібно увімкнути (зробити видимим) шар **Населенні_пункти_Anno** і підписати назви областей.

Для цього необхідно відкрити вікно властивостей шару **Населенні_пункти_Anno**, перейти в ньому на вкладку **General** та активізувати опцію **Visible** для властивості **Layer Name**. Всі інші параметри залиште за замовчуванням.

Для того щоб на карті швидко обирати певні об'єкти, необхідно відкрити атрибутивну таблицю шару **Населенні_пункти_Anno**. В полі **Bold** вказана жирність шрифту, яким підписано населені пункти. Потрібно вибрати з усіх міст тільки столицю. На панелі інструментів атрибутивної таблиці натисніть кнопку **Select by attributes** .

– У вікні вибору створіть запит для вибору тільки міст, назви яких віділено жирним шрифтом (рис. 4.95).

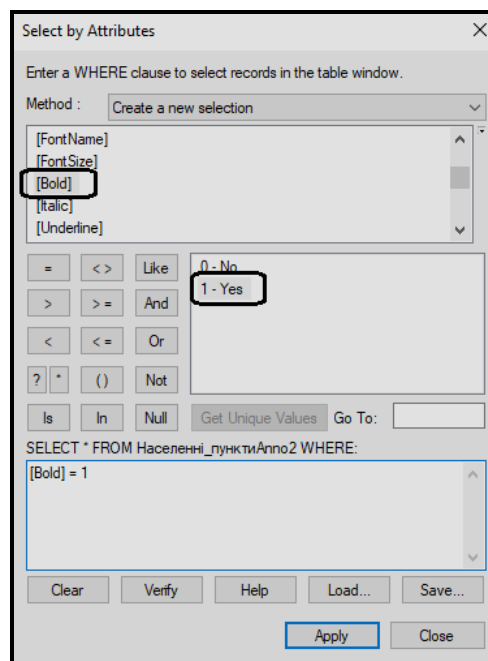


Рисунок 4.95 – Вигляд вікна вибору за атрибутам за створеним запитом

– Натисніть кнопку **Apply**. У таблиці атрибутів синім кольором

підсвітяться записи, назви яких вписані жирним шрифтом. Зверніть увагу, що внизу таблиці атрибутів вказано кількість таких міст.

Створення компоновки макету карти

– Перейдіть в режим макета карти, натиснувши на іконку у вигляді білого аркуша в нижній лівій частині зображення (рис. 4.96).

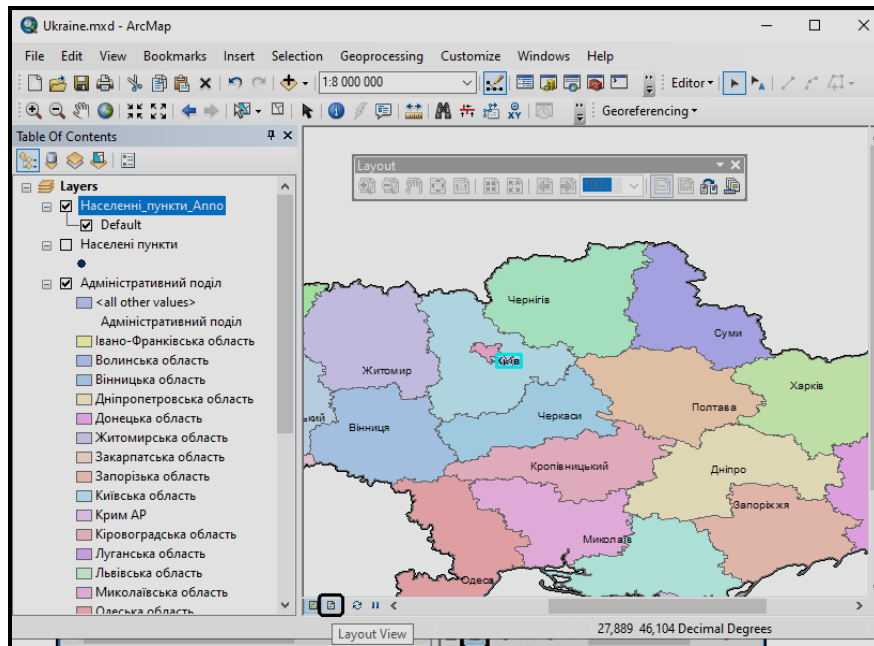


Рисунок 4.96 – Процес переходу в режим макету карти

На аркуші карти ви побачите ту частину карти, яка в момент створення макета була зображена на екрані. Для створення карти певної області або попередньо виберіть її за допомогою закладки, або в режимі перегляду даних перейдіть до необхідної ділянки території.

– Для того щоб додати заголовок на макет карти необхідно в меню **Insert** вибрати команду **Title**. Назва карти «Карта-схема Харківської області та прилеглих територій».

– Для того щоб додати легенду необхідно в меню **Insert** вибрати команду **Legend**. Виконайте кроки, які запропонує майстер створення легенди карти. Легенда повинна містити наступні шари: **Дороги, Залізні дороги, Населені пункти**.

– Перемістіть створену легенду з центру макета карти в нижній правий кут.

– Додайте позначення масштабу на карту. Для цього в меню **Вставка (Insert)** виберіть команду **Текст масштаба (Scale Text)**.

У разі нестандартного масштабу карти, а для масштабу краще використовувати 1: 10 000, 1: 1 000 000, 1: 50 000 000 і т.д., необхідно його задати.

Установка масштабу в макеті карти

- Викличте контекстне меню у вікні карти і виберіть **Properties**.
- У вікні **Data Frame Properties** перейдіть на вкладку **Data Frame** виберіть **Fixed Scale** і встановіть його рівним 1: 2 000 000 (рис. 4.87).

У макеті фрейм карти відразу придбає зазначений масштаб. Розмір області карти може дещо змінитися, відповідно до обраного масштабу.

Налаштування елементів макету карти

Необхідно задати параметри аркушу паперу, на якому буде роздруковано карту та зробити компоновку карти якнайкраще «читабельною».

- У меню **File** викличте команду **Page and Print Setup** та задайте необхідний розмір та орієнтацію аркушу паперу (рис. 4.98).

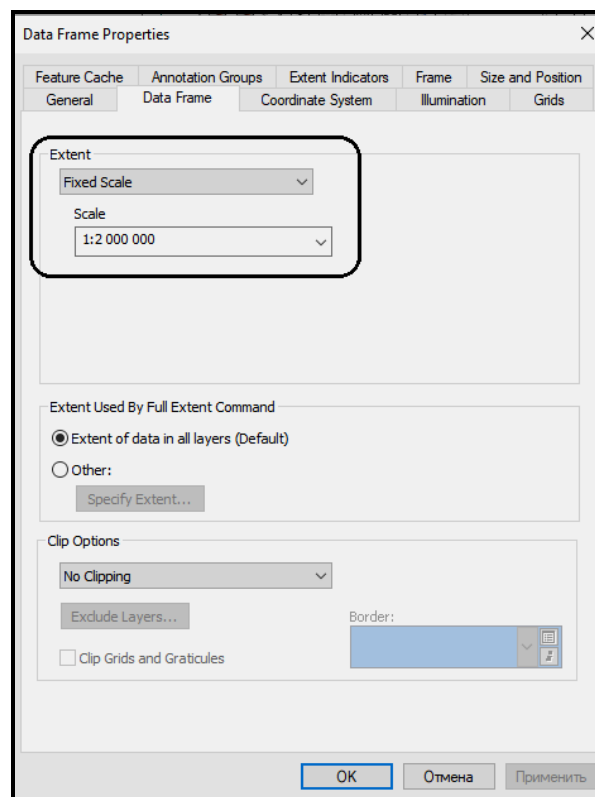


Рисунок 4.97 – Процес установки екстену карти

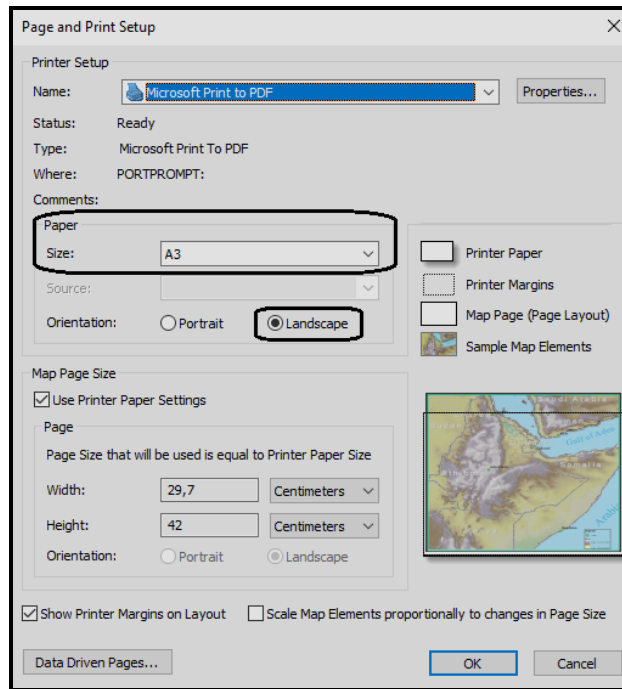


Рисунок 4.98 – Налаштування властивостей паперу

- Виконайте клацання лівою кнопкою миші (ЛКМ) по фрейму карти. За допомогою маркерів, які з'явилися в кутах фрейму, зробіть розмір фрейму карти відповідним розміру аркушу паперу.
- За допомогою клацання-протяжки ЛКМ розмістіть **Легенду** та **Текст масштабу** таким чином, щоб вони не перекривали площу Харківської області.
- Викличте **Properties** для **Легенди** та **Тексту масштабу** та налаштуйте розмір та накреслення тексту, щоб вони були більш виразними.

Збереження документа карти

- Зберегти зроблені зміни потрібно в новому документі карти.
- У меню **File** виберіть команду **Save As**.
- Збережіть карту під ім'ям **map_1.mxd**.

Завдання для самостійного виконання

1. Визначте п'ять найбільших областей України та виділіть їх на карті.
2. Визначте найменші за територією області України та виділіть їх на карті (створіть вираз, який вибере всі площі менші за $2 \cdot 10^4$ км²).
3. Створіть компоновку карти для Київської області для друку на

аркуші формату А4. Масштаб і легенду підберіть на свій розсуд.

Питання для самостійної роботи

13. Охарактеризуйте сфери використання ГІС з мережевою моделлю.

14.Що дозволяє ГІС в сфері впровадження сучасних технологій в міське управління?

15.Що ви розумієте під геоінформаційних аналізом? Наведіть приклади.

16.Назвіть складові ГІС, призначеної для просторового аналізу.

4.7 Практичне заняття. Знайомство з ARCGIS. Формати просторових даних

Мета: виробити вміння та навички створення баз геоданих та формування їхніх складників у програмному продукті **ArcCatalog** та роботі з базами геоданих у **ArcMap**.

Призначення: виконавши роботу, навчитися зв'язувати атрибутивні та графічні дані, працювати з шейп-файлами, растрами, ґрідами та покриттями.


ХІД РОБОТИ

Знайомство з ArcCatalog

Додаток **ArcCatalog** використовується для організації, пошуку, роботи і управління різними типами географічної інформації для ArcGIS for Desktop. У **ArcCatalog** можна організовувати і управляти наступними типами інформації:

- Бази геоданих.
- Файли растрів.
- Документи карт, тривимірні (3D) сцени і файли шарів.
- Набори інструментів геообработки, моделі, скрипти Python.
- ГІС-сервіси, опубліковані за допомогою ArcGIS for Server.
- Метадані в певних стандартах для цих ГІС-елементів.

Відкрийте програму **ArcCatalog**. У лівій частині вікна знаходиться **Дерево Каталог**, яке показує поточні з'єднання. У кожному конкретному випадку з'єднання можуть відрізнитися.

Для швидкого доступу до необхідної інформації необхідно додати нове з'єднання, клацнувши при цьому на кнопці **Connect to Folder** . Виберіть місце розташування папки «ЛР_2» і натисніть кнопку **ОК**. Тепер можна використовувати цей швидкий доступ для звернення до даних, що містяться в базі даних (рис. 4.99).

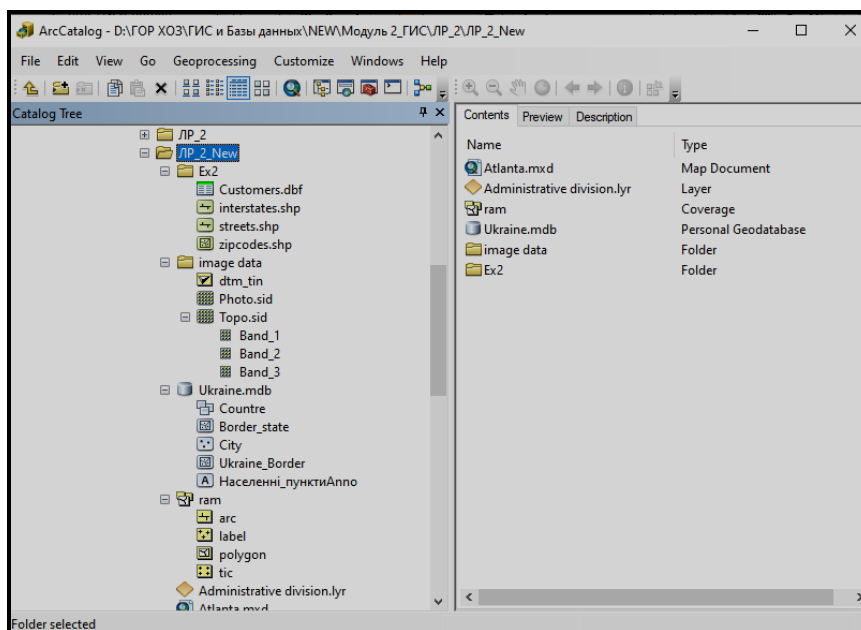


Рисунок 4.99 – Вид вікна ArcCatalog з підключеними даними

Клацніть по підключенню ЛР_2. Зверніть увагу, що різні типи просторових даних в папці представлені різними іконками (покриття, база геоданих, шейп-файли, шари, TIN і растрові дані). Іконки повідомляють про тип векторних об’єктів кожного просторового формату (точки, лінії, полігони).

Шейп-файли

Шейп-файли можуть містити тільки один клас об’єктів – точки, лінії або полігони. Клацніть по файлу **streets.shp** Перейдіть в **ArcCatalog** на вкладку **Preview** у верхній правій частині вікна. Ви побачите полігони класу об’єктів **streets** (дороги). Режим перегляду працює в двох варіантах. У нижній правій частині вікна **ArcCatalog** можна перемкнути режим **Geography** на режим **Таблиця** (рис. 4.100).

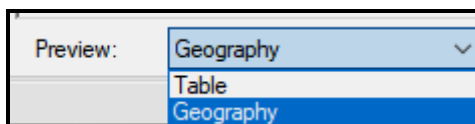
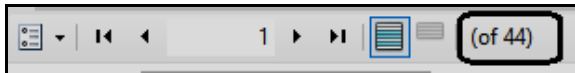





Рисунок 4.100 – Вид різних режимів перегляду інформації в ArcCatalog

– Перегляньте в різних режимах файли **streets.shp** (дороги) і **zipcodes.shp**. У режимі **Table** можна отримати відомості про кількість об’єктів класу (записах атрибутивної таблиці)



У графічному режимі перегляду, використовуючи кнопки керування зображенням на панелі **Geography** , можна збільшувати, зменшувати, рухати, переглядати повністю зображення, отримувати довідкову інформацію про об'єкт.

– Клацніть по файлу **streets.shp**, перейдіть в режим перегляду **Geography**. Збільшіть східну частину зображення, розгляньте мережу доріг більш докладно. Використовуючи інструмент **Identify**  отримаєте інформацію по вулицях міста Атланта.

– За допомогою кнопки **Create Thumbnail**  створіть картинку зображення для просмотра файлу на вкладці **Contents**. Для цього досить клацнути по зазначеній кнопці і переключитися на вкладку **Contents**. Обраний екстент стане відображатися у вигляді великого зразка зображення.

Покриття

На відміну від шейп-файлів покриття можуть містити різні типи класів об'єктів.

– У дереві **ArcCatalog** клацніть по покриттю **ram** (математична основа номенклатурного аркуша карти). Перейдіть на вкладку **Preview**. У покритті **ram** відображаються полігони.

Перейдіть на вкладку **Contents**. Покриття **ram** містить 4 класу об'єктів, кожен з яких можна переглянути: **arc** – лінії з яких складаються полігони, **label** – мітки полігонів, **polygon** – полігони, **tic** –реєстраційні точки покриття. Перегляньте кожен з цих класів в режимах **Geography** и **Table**.

– Перейдіть в режим перегляду **Geography** класу **tic**. Кожне покриття містить реєстраційні точки. Вони являють собою точки з відомими координатами, які використовуються для географічної прив'язки покриття. Перегляньте таблицю точок – в ній записані координати реєстраційних точок (поля ХТІС і УТІС).


Бази геоданих

Бази геоданих (БГД) є ще одним форматом, доступним для перегляду в **ArcCatalog**.

– Клацніть в дереві **ArcCatalog** на БГД **Ukraine.mdb**. Це персональна база геоданих, що має розширення **MDB** (формат, підтримуваний Microsoft Access). Перегляньте вміст бази геоданих. Вона містить класи просторових об'єктів.

Кожен клас представляє собою групу об'єктів з одним типом геометрії (точки, лінії, полігони). Іконка, яка використовується для кожного класу об'єктів, повідомляє про тип геометрії об'єкта.

Перехід у ArcMap

– Натисніть кнопку **Запустити ArcMap**  на **Standard** панелі інструментів **ArcCatalog**. Якщо з'явиться вікно, яке пропонує перейти до недавно відкритих файлів, виберіть **A new empty map**, потім натисніть на кнопку **ОК**.

– Змініть розмір вікон додатків **ArcMap** і **ArcCatalog** таким чином, щоб бачити їх одночасно на екрані.

– Перетягніть з **ArcCatalog** в область відображення **ArcMap** іконку шейп-файлу **streets.shp**. Зверніть увагу, що, колір поліліній обраний довільно, його можна змінити.

– Таким же чином перемістіть шейп-файл **zipcodes.shp** в **ArcMap**.

Зверніть увагу, що полігони додалися на карту і в таблицю змісту. Вони автоматично додалися в нижню частину списку таблиці змісту таким чином, що полілінії (вулиці) відображаються поверх полігонів, замість того щоб цими полігонами перекриватися.

Зверніть увагу, що **streets** і **zipcodes** суміщені в координатному просторі. Це відбувається тому, що вони знаходяться в одній системі координат.

Отримання інформації про властивості класів та наборів даних

– У БГД **Ukraine.mdb** клацніть правою кнопкою на клас об'єктів **Border_state**, виберіть у спадаючому меню команду **Properties** перейдіть на вкладку **Fields**. Інформація про просторову прив'язку зберігається в полі **Shape** (рис. 4.101).

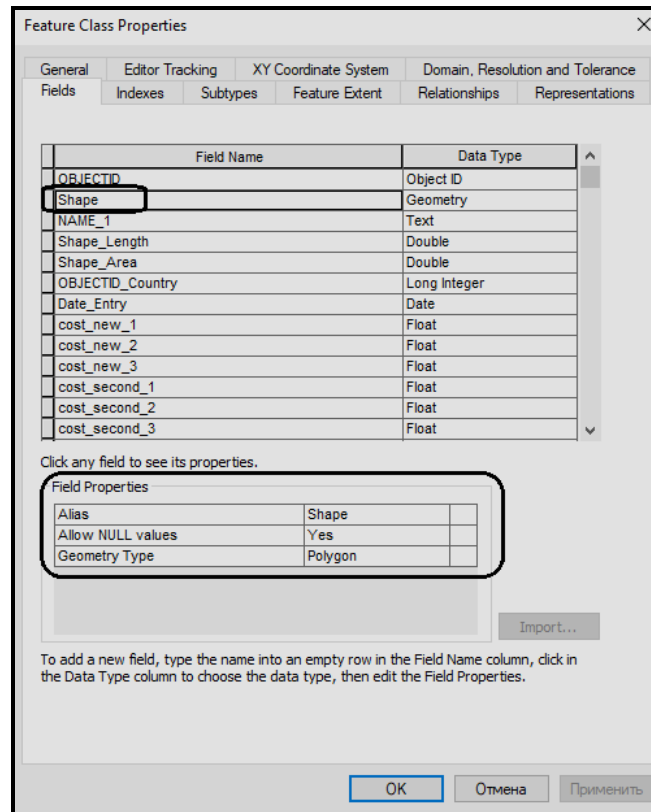



Рисунок 4.101 – Вид вікна **Properties** для класу об’єктів **Border_state**

- Закрийте вікно **Properties**.
- Для класу просторових об’єктів **City** відкрийте вікно **Properties**. Подивіться інформацію про просторову прив’язку на вкладці **XY Coordinate System**.
- Вивчіть вміст всіх вкладок вікна **Properties** для класу просторових об’єктів **City**.

Растри

Растри – це відскановані карти фотографій земної поверхні, дані супутникової зйомки. Растрові формати зберігають дані у вигляді клітинок однакового розміру або пікселів, організованих у рядки та стовпці. Розширення даних пов’язано з розміром осередку: менший розмір клітинки може показати більше деталей в межах досліджуваної області, але також збільшує розмір файлу.

- У дереві каталогу **ArcCatalog** міститься папка **image data** з растром **Photo.sid**. Перегляньте його, перейшовши на вкладку **Preview**.
- За допомогою інструменту **Zoom In**  збільште фрагмент ділянки зображення. Ви побачите пікселі, з яких складається растр.

– Вивчіть властивості растра **Photo.sid**, викликавши контекстне меню на назві файлу і вибравши команду **Properties**.

Tin

TIN (TIN) – триангуляційна нерегулярна мережа. Є моделлю поверхні. База геоданих зберігає TIN у вигляді інтегрованого набору вузлів із значеннями висоти і трикутників зі сторонами, що з'єднують вузли. Служить ефективною і точною моделлю для опису безперервних поверхонь. Для моделювання поверхонь можна використовувати також і растри, але TIN буде кращою, так як змінює щільність даних в залежності від висоти рельєфу. TIN може бути показана у вигляді дротяної моделі або моделі з зафарбованими гранями.

– У дереві каталогу **ArcCatalog** міститься папка **image data**, в якій розміщено TIN **dtm_tin**, перегляньте її, перейшовши на вкладку **Preview**.

Шар

Шар становить певний тип однотипних об'єктів або спеціалізовані дані (наприклад, космічні знімки або TIN), які вже були певним чином оформлені. Шар не містить власне географічних даних. Замість цього він посилається на дані, які зберігаються в покриттях, шейп-файлах, растрах.

– Відкрийте в **Properties** для шару **Administrative division.lyr**, перейдіть на вкладку **Source**. Як джерело даних повинен використовуватися полігональний клас об'єктів **Border_state**. Необхідно вказати правильний шлях до цього файлу за допомогою кнопки **Set Data Source**.

– Зверніть увагу, що полігональні об'єкти (адміністративні області) відображені різним розміром кольором. Це можна побачити на вкладці **Preview**.

– У вікні **Properties** перейдіть на вкладку **Symbology**. Зробимо щоб колір залежав від значення поля **Shape_Area** (територія області).

– Для зміни класифікації або кольору відображення необхідно виконати дії як на рисунку 4.102. Натисніть **OK**. У результаті колір областей зміниться. Темніший колір у областей з більшою територією.

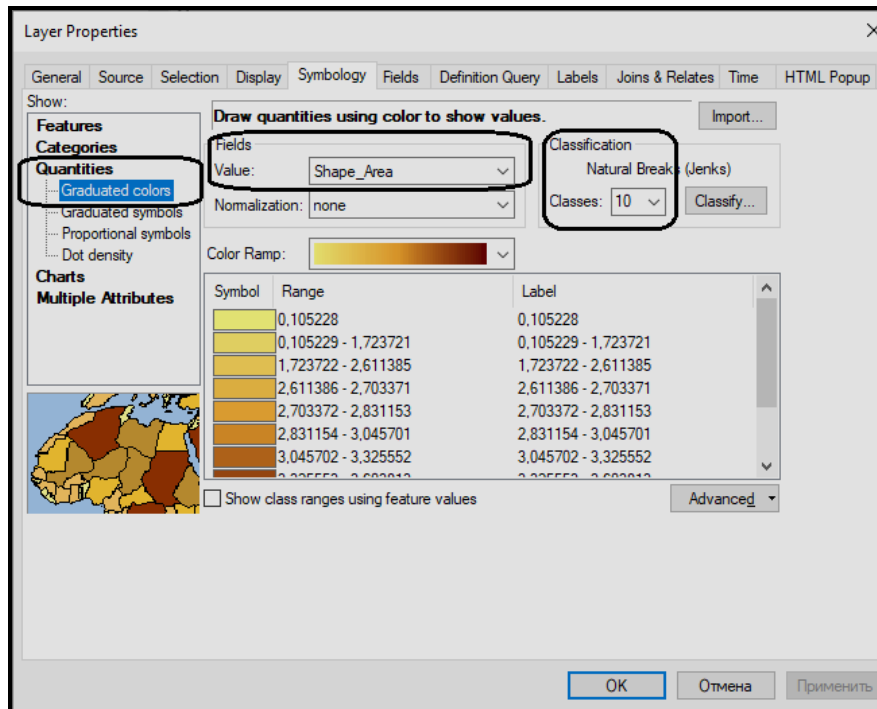


Рисунок 4.102 – Вид вікна **Properties** для зміни класифікації

Таблиця

Табличні дані можуть бути представлені у вигляді файлів dBASE або атрибутивних таблиць бази геоданих.

– У дереві Каталогу **ArcCatalog** Перейдіть до папки **Ex2** клацніть по таблиці **Customers.dbf**. Перейдіть на вкладку **Preview**. У таблиці **Customers.dbf** містяться деякі статистичні дані про деякі компанії у місті Атланта (адреса, назва, вид діяльності).

– Для більш зручної праці з таблицею відсортуйте її по назві компанії – поле **NAME**. Для цього клацніть правою кнопкою миші (ПКМ) по назві поля **NAME** і виберіть команду **Sort Ascending** (рис. 4.103).

OID	CUST_ID	NAME	ADDRESS	ZIP	TYPE
0	840367	Accessories Inc		30309	Retail Clothing
1	291063	Antiques Inc		30308	Retail
2	520746	Damar Sales		30308	Car Dealer
3	197302	Eastern Graphics		30308	Printing
4	914582	Gelroy Associates		30308	Real Estate
5	107892	Homes For Sale		30308	Real Estate
6	308574	Industrial Supply Co		30308	Wholesale
7	489041	Leonard Associates		30309	Stockbroker
8	734173	Lomer-Lockler Beauty College		30308	Trade School
9	227309	Oar Clothing Co		30308	Retail Clothing
10	197842	Ron's Food Mart	844 MIRTEL ST NE	30308	Retail Food
11	216438	S and S Industrial Products	711 KENNESAW AVE NE	30308	Wholesale
12	185628	Shawly and Company	911 MONROE DR NE	30308	Restaurant
13	913007	Sound Concepts	400 PONCE DE LEON AVE NE	30308	Retail Entertainment
14	166732	Southern Properties	699 JUNIPER ST NE	30308	Real Estate
15	318592	T & B Bookkeeping	42 NORTH NE	30308	Bookkeeping

Рисунок 4.103 – Сортування даних поля

Копіювання об'єктів в ArcCatalog

Всі типи даних, які ви переглядаєте в **ArcCatalog**, мають досить складну організацію, тобто можуть знаходитися не в одного файлі, а в декількох. Крім того, такі дані як покриття, організовані за допомогою папок, пов'язаних між собою. Тому операції по копіюванню, переміщенню, видалення даних доцільно проводити не в програмах для навігації між файлами, типу **Explorer**, а в **ArcCatalog**.

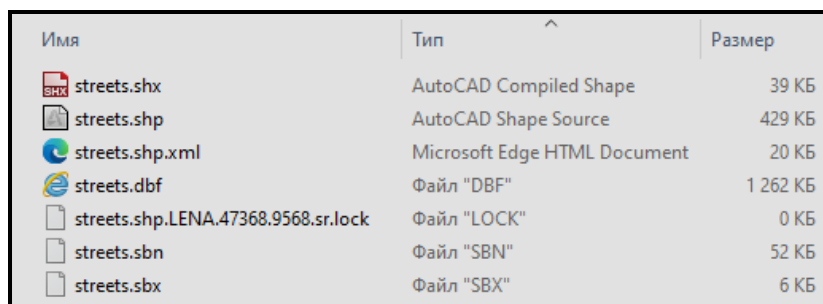
– Перегляньте вміст папки **ЛР_2** в програмі **Explorer**. Ті дані, які ви бачили в **ArcCatalog**, виглядають зовсім інакше. Зверніть увагу на шейп-файли. Кожен клас об'єктів описується декількома файлами. Покриття, TIN, ґрид містяться в папках, які не відрізняються від звичайних папок. Растр має також ряд додаткових файлів, які відповідають за просторову прив'язку растра. Крім того, в папці **ЛР_2** міститься база даних – **Ukraine.mdb**, яка служить для коректної роботи з базою геоданих з нерухомості України. Просте видалення або переміщення цього файлу може привести до безповоротної втрати даних.

– Закрийте програму **Explorer**.

– Створіть у своїй робочій папці в **ArcCatalog** папку з назвою **Mydata**. Для цього викличте контекстне меню на папці **ЛР_2** і виберіть команду **New/Folder**.

– Скопіюйте в папку **Mydata** шейп-файл **streets.shp**. Для цього клацніть ПКМ по файлу **streets.shp**, виберіть команду **Copy**. Потім клацніть ПКМ по папці **Mydata**, виберіть команду **Paste**.

Якщо б те саме ви робили в **Explorer**, то довелося б копіювати декілька файлів, що зберігають в собі інформацію про привязки, атрибутивну і іншу інформацію (рис. 4.104). Будь-який не скопійований файл руйнує формат даних і веде до їх втрати.



Имя	Тип	Размер
streets.shx	AutoCAD Compiled Shape	39 КБ
streets.shp	AutoCAD Shape Source	429 КБ
streets.shp.xml	Microsoft Edge HTML Document	20 КБ
streets.dbf	Файл "DBF"	1 262 КБ
streets.shp.LENA.47368.9568.sr.lock	Файл "LOCK"	0 КБ
streets.sbn	Файл "SBN"	52 КБ
streets.sbx	Файл "SBX"	6 КБ

Рисунок 4.104 – Вміст папки **Mydata** з шейпом **streets.shp** в програмі **Провідник**

– Перемістіть файл **interstates.shp** в папку **Mydata**. Для цього в дереві **ArcCatalog** лівою кнопкою миші візьміть файл і перетягніть на папку **Mydata**.

– Скопіюйте в папку **Mydata** файли **Customers.dbf, streets.shp**.

– Клацніть ЛКМ по знаку «плюс» поруч з іконкою папки **Mydata** в дереві **ArcCatalog**. Переконайтеся в тому, що всі об'єкти знаходяться в вашій папці.

– Зверніть увагу на те, що скопійований раніше шейп-файл **streets.shp** автоматично змінився на файл з ім'ям **streetsCopy.shp**. Таким чином, не відбулося перезапису одного файлу іншим при копіюванні. Тобто **ArcCatalog** зберігає дані в обох файлах з одним і тим же ім'ям.

– Видаліть файл **streetsCopy.shp** з папки **Mydata**.

Створення нового шейп-файлу

– Для створення в папці **Mydata** нового шейп-файлу точкових об'єктів необхідно клацнути ПКМ в дереві каталогу по папці **Mydata** і вибрати команди **New / Shapefile**.

– У вікні, що відкрилося, назвіть файл **mypoint**. Тип об'єкта – **Point**. Якщо ви не знаєте, як просторово описати новий шейп-файл, то просторову прив'язку залиште за умовчанням **Unknown Coordinate System**. Надалі це позначиться на одиницях виміру карти, вони будуть також не відомі.

– Натисніть **ОК**. Тільки що був створений новий клас точкових об'єктів – шейп-файл **mypoint** з невідомою системою координат. Поки в ньому немає жодного об'єкта. Переконайтеся в цьому, переглянувши файл в режимі **Geography** і **Table**.

– Наповнити файл змістом можна тільки в програмі **ArcMap**. **ArcCatalog** призначений для створення структури просторових даних – оболонки з певними властивостями, готової для наповнення.

– Створіть подібним чином клас лінійних об'єктів **myline** і клас полігональних об'єктів **mypoly** (рис. 4.105).

– Для будь-якого створеного шейп-файлу, що не містить просторової прив'язки, можна додати інформацію про систему координат вже після створення файлу. Для цього клацніть правою кнопкою миші по будь-якому створеному шейп-файлу, наприклад **mypoint.shp**, увійдіть в **Properties** / вкладка **XY система координат XY Coordinate System**.

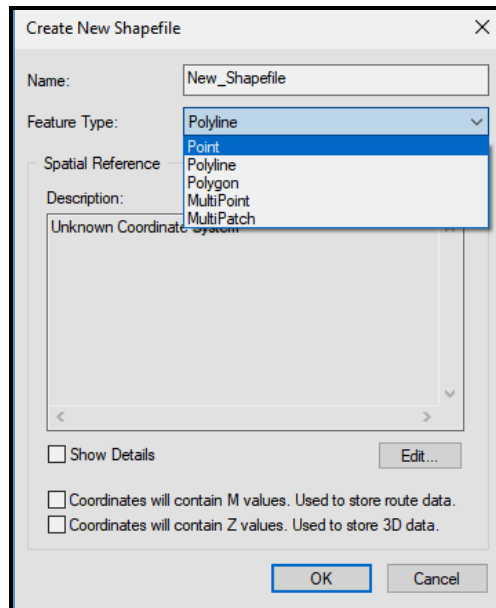


Рисунок 4.105 – Процес створення шейпфайлу

– Виберіть **Geographic Coordinate Systems / Asia / Pulkovo 1942.prj** (рис. 4.106). Таким чином ви визначили для своїх даних що вони будуть створюватися в географічній системі координат, на еліпсоїді Красовського, вимірювання – в десяткових градусах. Натисніть кілька разів **ОК** для того щоб закрити всі вікна і застосувати зміни.

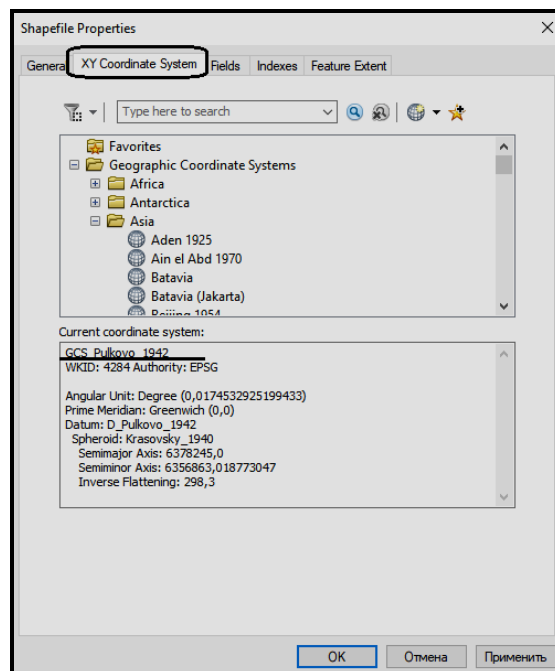


Рисунок 4.106 – Процес вибору системи координат для шейп-файлу

– Створіть таку ж просторову прив'язку для класу лінійних

об'єктів **myline** і класу полігональних об'єктів **mypoly**.

Створення бази геоданих

– Створіть в раніше створеній папці **Mydata** нову базу геоданих (БГД). Для цього викличте контекстне меню в дереві каталогу на папці **Mydata** і виберіть команди **New / Personal Geodatabase**. Файл назвіть **myGDB.mdb** (рис. 4.107).

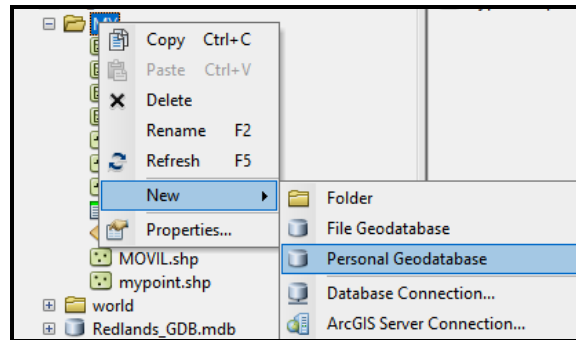


Рисунок 4.107 – Процес створення Персональної бази геоданих

База геоданих може містити набори класів об'єктів, класи просторових об'єктів, таблиці, класи відносин, файли баз даних.

– Створіть набір даних **Topography**. Для цього клацніть ПКМ по створеній БГД **myGDB**, в контекстному меню виберіть команди **New / Feature Dataset** введіть ім'я **Topography**. Цей набір буде зберігати класи об'єктів, об'єднані однією тематикою – топографічні елементи (ізолінії рельєфу, гідромережа, позначки висот, населені пункти).

– У наступному вікні майстра, що з'явився, для визначення просторової прив'язки всіх класів цього набору даних виберіть **Geographic Coordinate Systems / World / WGS 1984**. Таким чином була обрана географічна система координат, заснована на еліпсоїді **WGS 1984**. У наступних вікнах майстра всі настройки залишайте за замовчуванням.

Набір просторових об'єктів може включати в себе класи просторових об'єктів, класи відносин, тут же може створюватися топологія, полігони і геометричні мережі.

– Для створення нового класу гідрографічних лінійних об'єктів викличте контекстне меню на наборі просторових об'єктів **Topography** і виберіть команди **New / Feature Class** введіть ім'я **dnl**. В поле **Alias** введіть «Гідрографічні лінійні об'єкти». Тип об'єктів змініть на **Line**. Натисніть кнопку **Далі**.

При створенні класу можна додати користувацькі поля в атрибутивну таблицю, наприклад, призначений для користувача ідентифікатор (**dnl_id**), код легенди (**dnl_code**), власне ім'я об'єкта (**dnl_text**) та інші. Структуру атрибутивних таблиць можна дізнатися з відповідних нормативних або методичних документів або створити власну.

- Для цього в порожньому рядку **Field Name** введіть **dnl_id**, тип даних виберіть **Long Integer**.
- Для поля **dnl_code** тип даних – **Float**.
- Для поля **dnl_text** тип даних – **Text** (рис. 4.108).

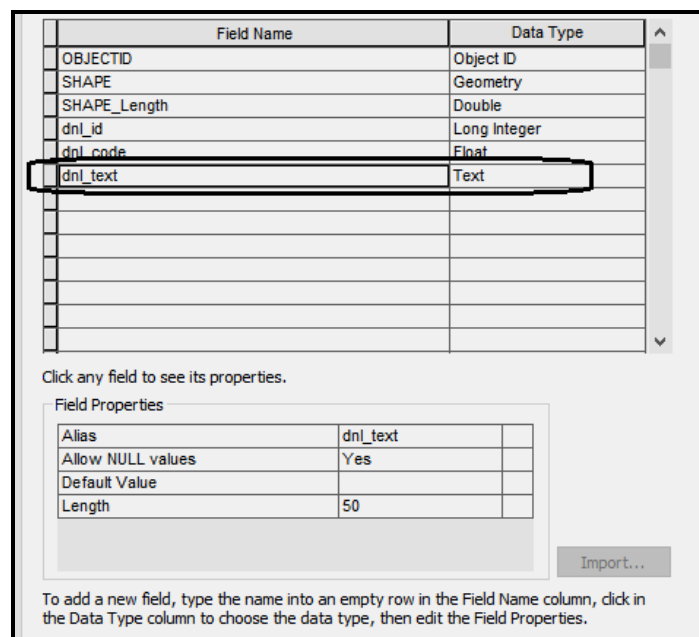


Рисунок 4.108 – Фрагмент вікна **New Feature Class** з властивостями полів

- Натисніть кнопку **Готово**.
- За аналогією створіть в наборі **Topography** клас полігональних об'єктів **dna** (площадні гідрографічні об'єкти). Додайте для цього класу поля **dna_id**, **dna_code**, **dna_text**. Типи даних кожного поля такі ж, як в класі **dnl**.
- Створіть в цьому ж наборі точковий клас об'єктів з ім'ям **ppp** (населені пункти точкові). Свої атрибутивні поля можна не створювати.

Повинна вийти структура підлеглих і взаємопов'язаних елементів в наборі класів **Topography** як на рисунку 4.109.

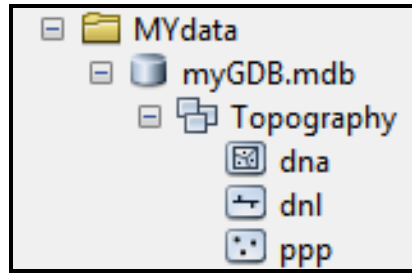


Рисунок 4.109 – Вид структури набору класів **Topography** в Дереві каталогу

В результаті ви навчилися створювати класи просторових об'єктів, об'єднувати їх в набори в **Personal Geodatabase**. У цій же БГД можна визначати і встановлювати взаємозалежну поведінку даних різного типу (це описується топологічними правилами), створювати геометричні мережі (топологічна модель, що складається з ліній і точок).

– Перегляньте у програмі **Провідник** або **Мій комп'ютер** файл **myGDB.mdb** в папці **LP_2**. Відкрийте за допомогою будь-якої СКБД цей файл, вивчіть вміст таблиць **dnl**, **dna** і **ppp**. Зверніть увагу на кількість полів і їх властивості. Вивчіть структуру системних таблиць – тих таблиць, які **ArcCatalog** створив самостійно при створенні персональної БГД.

Вивчення Метаданих

Метадані – дані про дані: створюються розробником або постачальником просторової інформації. Це всілякі каталоги, довідники, реєстри, бази метаданих, що містять відомості про зміст даних, статус, походження, місцезнаходження, формат і форму подання, умови доступу.

– Перейдіть до папки **LP_2 \ Ex2**, виберіть шейп-файл **zipcodes.shp**. Перейдіть на вкладку **Description** у вікні **ArcCatalog**. Зверніть увагу що вся представлена інформація розбита на групи для зручності.

Завдання для самостійного виконання

1. Скільки об'єктів міститься в шейп-файлах:
 - streets.shp;
 - interstates.shp;
 - zipcodes.shp.
2. Який з полігонів покриття RAM має найбільшу площу?
3. Скільки в БГД **Ukraine.mdb** міститься класів просторових

об'єктів?

4. Відкрийте БГД **Ukraine.mdb** в Microsoft Access. Додайте всі наявні таблиці в схему бази даних. Який вид зв'язків між таблицями? Чому саме такий?

5. Яка система координат застосована для класів просторових об'єктів БГД **Ukraine.mdb**?

6. Яка система координат використовується для шейп-файлу **zipcodes.shp**?

7. Яка система координат використовується для покриття **ram**?

8. Скільки рядків і стовпців містить растр **Photo.sid**? Який розмір клітинки цього растра?

9. Яка площа України?

10. Визначте мінімальний і максимальний за площею адміністративний район України.

11. Перерахуйте, якого роду інформацію можна отримати про шейп-файли **zipcodes.shp**?

Питання для самостійної роботи

1. Як відбувається класифікація ГІС?

2. Назвіть складові ГІС з просторового аналізу.

3. Перерахуйте види ГІС по області діяльності. Наведіть приклади.

4. Як ГІС розрізняються по функціональності?

5. Наведіть приклади ГІС по моделі даних, що в них використовується.

6. Як відбувається організація даних в ГІС?

4.8 Практичне заняття. Конвертація даних в ArcGIS

Мета: виробити вміння та навички конвертації файлів у різні формати, які сприймаються базою геоданих.

Призначення: виконавши роботу, навчитися конвертувати файли у формати, які сприймаються базою геоданих та виконувати зворотню конвертацію.

ХІД РОБОТИ

Початкові дані

- Формат Shape:
 - **DNL_ARC** – гідрографія (лінійні об'єкти);
 - **DNAR_POLY** – гідрографія (площадні об'єкти).
- Формат Бази геоданих – **MAP.mdb**;
- Формат AutoCad – **railroad.dxf** дороги (лінійні об'єкти);
- Текстовий файл – **uch.txt** межі ділянок (лінійні об'єкти).

Конвертація вихідних форматів ArcGIS

Конвертація основних форматів ArcGIS виконується в **ArcCatalog**. Необхідно виконати конвертацію лінійних об'єктів гідрографії (**DNL_ARC**) з формату **Shape** в базу геоданих.

– Для цього необхідно попередньо створити базу геоданих **Convert** з набором класів просторових даних **Con_1**. У контекстному меню для об'єкта **DNL_ARC** виберіть команди **Export / To Geodatabase (single)**.

– В якості вхідного об'єкта виберіть файл **DNL_ARC.shp**. У якості місця розташування – набір класів просторових даних **Con_1** в базі геоданих **Convert**. Ім'я вихідного класу об'єктів – **DNL** (рис. 4.110).

– Перегляньте після конвертації вміст набору клас просторових об'єктів (КПО) **Con_1**. Там з'явиться клас **DNL**. У разі, якщо КПО не відображує останні зміни, викличте в контекстному меню команду **Refresh**. Перевірте на вкладці **Description** екстенсивності вихідного і експортованого об'єктів (рис. 4.111). Відрізняються вони чи ні?

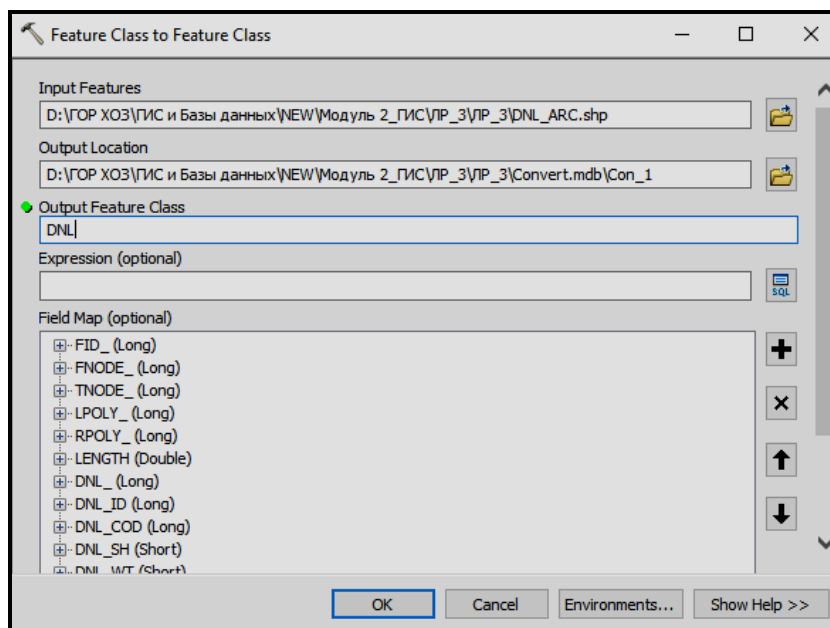


Рисунок 4.110 – Вид вікна експорту

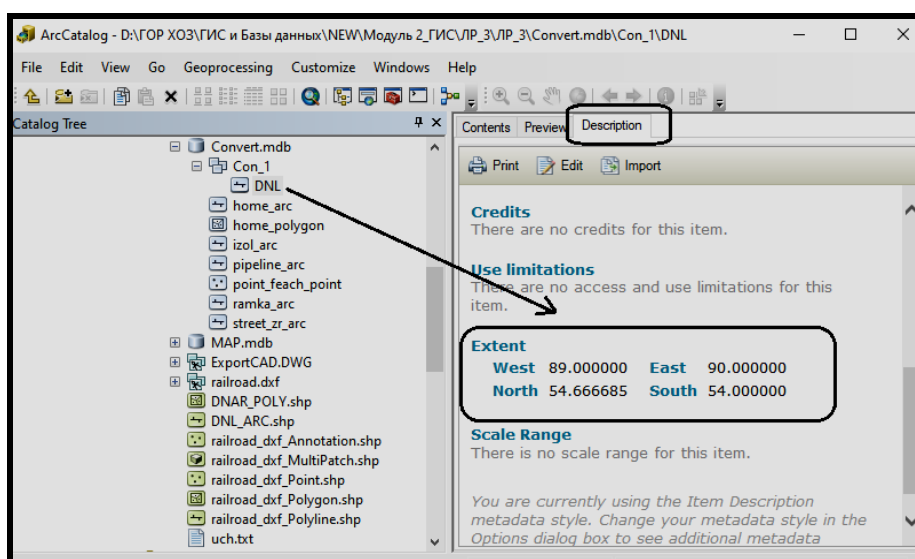


Рисунок 4.111 – Перегляд властивостей лінійного класу просторових об'єктів

Зворотній конвертація – з бази геоданих в формат Shape

У наборі класів може бути кілька класів об'єктів. Можлива конвертація одиничного класу або групи. Розглянемо як виконується конвертація групи об'єктів.

Набір КПО **Gaz** в базі геоданих **MAP** містить 7 просторових об'єктів. Для їх конвертації необхідно викликати контекстне меню і

вибрати команди **Export / To Geodatabase (multiple)**.

- У вікні конвертації з'явиться список класів об'єктів набору КПО **Gaz**, вкажіть вхідне місцерозташування – БГД **Convert** (рис. 4.112).

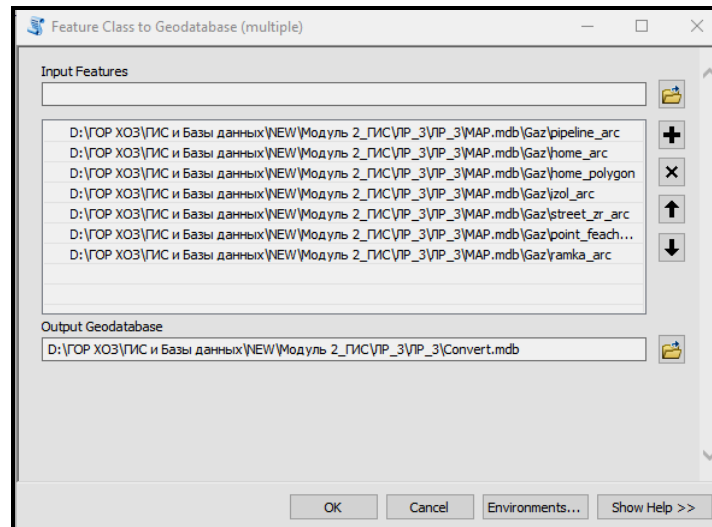


Рисунок 4.112 – Вид вікна екпорту в шейп-файл

- У вашій БГД **Convert** повинні з'явитися всі класи об'єктів з бази геоданих **MAP**.

Конвертація в зовнішні формати з форматів ArcGIS.

Конвертація початкових форматів ArcGIS в формат DWG

- Для конвертації набору КПО **Gaz** бази геоданих **MAP** в формат DWG необхідно в контекстному меню вибрати команди **Export / To CAD**.
- Змініть назву вихідного файлу на **ExportCAD.DWG** (рис. 4.113).

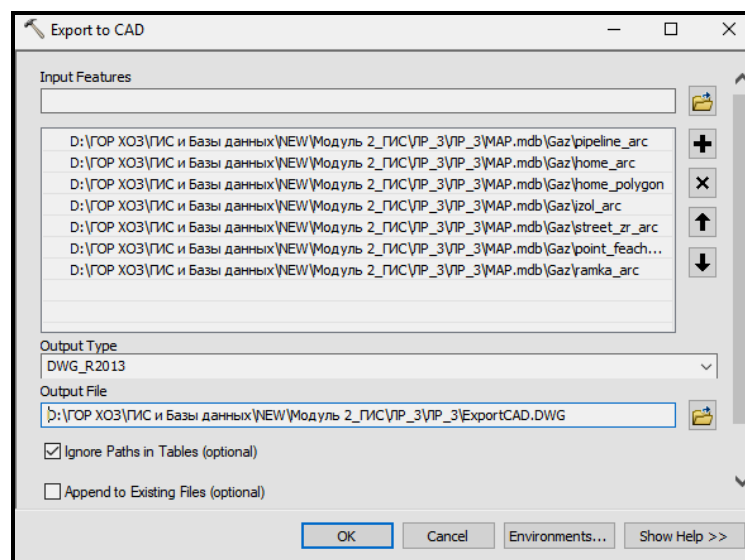


Рисунок 4.113 – Вид вікна екпорту в формат САПР

Обратна конвертація – з формату САПР у шейп-файл

– Самостійно конвертуйте в **shape**-формат файл **railroad.dxf** (набір просторових об'єктів САПР). В якості вихідної папки виберіть **ЛР_3**.

Завдання для самостійного виконання

1. Порівняйте атрибутивні таблиці лінійного класу просторових об'єктів **DNL** і файлу **DNL_ARC.shp**. Які поля з'явилися в КПО **DNL**? Які дані в них зберігаються?

2. Чи зберігаються атрибутивні дані об'єктів при конвертації з бази геоданих в формат Shape? Підтвердіть це на прикладі.

3. Порівняйте вихідні КПО **Gaz** бази геоданих і отримані об'єкти в форматі DWG. Які відбулися зміни в структурі даних?

4. Чи збереглися атрибутивні дані при конвертації з формату Shape в формат САПР? Підтвердіть це на прикладі.

5. Скільки файлів вийшло при конвертації файлу **railroad.dxf** в **shape**-формат?

6. Скільки об'єктів і яких описано парами координат в файлі **Uch.txt**?

Питання для самостійної роботи

1. Поняття інформації. Види інформації.

2. Охарактеризуйте основні інформаційні процеси: зберігання, передача та обробка інформації.

3. Охарактеризуйте дискретне подання інформації: двійкові числа; двійкове кодування тексту в пам'яті комп'ютера.

4. Охарактеризуйте дискретне подання інформації: кодування кольорового зображення в комп'ютері (растровий підхід).

5. Охарактеризуйте процес передачі інформації, джерело і приймач інформації, канал передачі інформації.

4.9 Практичне заняття. Географічна прив'язка зображень з використанням векторних шарів карти і за координатами. Створення мозаїки растрів

Мета: виробити вміння та навички з просторової прив'язки растрових зображень, оцінення її точності, створення нового трансформованого растра.

Призначення: виконавши роботу, навчитися завантажувати растрове зображення у проєкт, реєструвати растр за координатами, створювати нове зображення на підставі декількох.

ХІД РОБОТИ

Географічна прив'язка зображень з використанням векторних шарів карти

Базові функції ArcGIS включають потужні інструменти географічної прив'язки растрових зображень. У цій роботі використовуються знімки і векторні дані, що ілюструють зміни в період спорудження швидкісної автомагістралі в місті Солт-Лейк-Сіті, штат Юта.

У цій практичній роботі використовуються:

- два аерофотознімки в форматі JPG (файли **Slc_9910.jpg** і **Slc_0107.jpg**);
- файл прив'язки (**world file**) для одного зі знімків (**Slc_0107.jgw**);
- шейп-файл центральних ліній вулиць (**teleatl1**);
- опорні точки для прив'язки зображення (**controlpt**).

Векторні дані представлені в проєкції штату **Utah State Plane NAD83 Central Zone**, у якості одиниць виміру задані фути. Векторні дані по вулицях взяті для округу Солт-Лейк-Сіті. Векторні набори даних об'єднані в окремий шейп-файл і містять просту легенду, що міститься у файлі шару **ArcMap**. Для географічної прив'язки (реєстрації) знімка використовується шейп-файл з сімома опорними точками, які знаходяться на перехресті основних вулиць.

Зображення це два аерофотознімки, на яких показаний район виконання робіт по реконструкції автомагістралей I-15 близько від шосе I-80. Для зменшення розміру файлів якість знімків було зменшено з 2 до 8 футів, і вони були збережені у форматі JPG. Один зі знімків не має просторової прив'язки, інший приведений до проєкції **Utah State Plane**


NAD83 Central Zone.

Необхідно виконати реєстрацію знімка без просторової прив'язки. Після реєстрації обидва знімки можна буде порівняти, виявити відмінності, що з'явилися в ході будівництва і залежності від сезону року.

Загрузка даних

Щоб почати виконання вправи, треба активізувати панель інструментів геообробки і додати векторні дані і аерофотознімок.

– Запустіть **ArcMap** і додайте панель інструментів **Georeferencing**, вибравши в меню **View** команду **Customize / Toolbars**. Розмістіть цю панель горизонтально.

– **Add Data**  з папки **LP_4\SHPPfiles\UTSPN83C** і виберіть файли **teleatl.lyr** і **controlpt.lyr**. Червоні знаки оклику біля кожного з шарів показують, що файли цих шарів треба пов'язати з відповідним шейп-файлом (актуалізувати). Для цього викличте контекстне меню на кожному з цих шарів, виберіть команду **Properties**. У вікні, що відкрилося, перейдіть на вкладку **Sources** і за допомогою кнопки **Set Data Source** перейдіть до папки **LP_4\SHPPfiles\UTSPN83C** і виберіть пов'язаний шейп-файл.

– Дані **teleatl** і **controlpt** представлені в системі координат **Utah State Plane NAD83 Central Zone**. Щоб ці дані були відображені коректно, необхідно змінити одиниці вимірювання карти. Для цього викличте контекстне меню на групі **Layers** в таблиці вмісту, виберіть команду **Properties**. У вікні, що відкрилося, перейдіть на вкладку **General** і в випадаючому списку виберіть **feet** в якості одиниць виміру для карти.

– Тепер необхідно додати знімок за жовтень 1999 року (**Slc_9910.jpg**). Він знаходиться в папці **LP_4\Images\Unknown**. Виберіть опцію **Build Pyramids** на запит **Create Pyramids for Slc_9910.jpg?** не звертайте увагу. Це повідомлення сповіщає про те, що у зображення відсутня інформація про географічну прив'язку.

– За допомогою контекстного меню на файлі **Slc_9910.jpg** в таблиці вмісту виберіть команду **Zoon to Layer**. Зверніть увагу на те, які одиниці вимірювання показані в нижньому правому куті вікна карти. Зверніть увагу на невеликі значення координат знімка, вони знаходяться в межах від -1,000 до +1,500, що свідчить про те, що цей шар не спроецировано в проекцію **Utah State Plane**.

Геодезисти часто розміщують точки наземного контролю в важливих пунктах, які легко розпізнати, що допомагає географічній прив'язці і створенню стереомоделей. Зверніть увагу на сім невеликих білих хрестиків в кутах і в центральній частині знімка **Slc_9910.jpg**. Це псевдо-точки, що представляють фактичні контрольні точки, які були створені на цьому зображенні, щоб полегшити його прив'язку до геодезичних опорних точок, представлених в файлі **controlpt.lyr**.

Збережіть документ **ArcMap** під ім'ям **SLC_01**.

Географічна прив'язка знімка

Географічна прив'язка знімка до координат проекції **Utah State Plain** виконується за допомогою контрольних точок на знімку і шару вулиць трансформацією по контрольним точкам і приведенням до відповідної системи координат.

– На панелі інструментів **Georeferencing** виберіть в якості шару **Slc_9910.jpg**, потім у випадаючому меню виберіть команду **Fit to Display** (рис. 4.114). Тепер знімок Slc_9910 буде введений в загальний простір моделі.

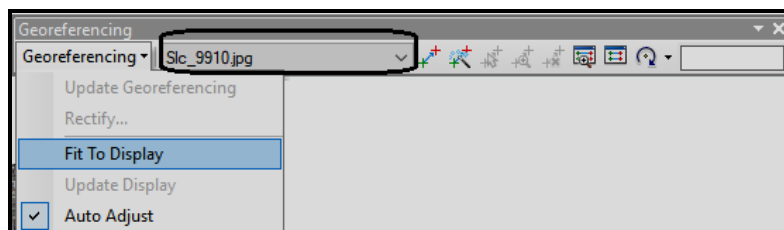



Рисунок 4.114 – Вид панелі **Прив'язка**

– Для суміщення опорних точок необхідно наблизити північно-західний кут знімка так, щоб було видно зелений значок **X** біля цифри **1** (перша опорна точка) і білий значок **X** псевдо-точки, що представляє контрольну точку. На панелі інструментів **Georeferencing** клацніть на кнопці **Add Control Points** . Курсор перетвориться на перехресті. Розташуйте курсор над псевдо-точкою, позначеною білим значком **X**, і зробіть клацання. Курсор перетвориться в кольорове перехресті. Перемістіть курсор на опорну точку, позначену зеленим значком **X**, і клацніть на ній. Зображення зрушиться і з'єднає два значка **X** прямою лінією (рис. 4.115). Цей зсув зображення являє собою трансформацію по одній точці, засновану на комбінації однієї контрольної точки на растрі і


відповідної опорної точки в цільових даних (в нашому випадку це **controlpt.lyr**), і називається зв'язком (**link**).



Рисунок 4.115 – Процес суміщення опорних точок

– Клацніть правою кнопкою миші на растрі **Slc_9910.jpg** в таблиці змісту і виберіть команду **Zoom to Layer**, щоб побачити весь знімок.

– Наблизьте зображення до південно-східного кута знімка (нижній правий) і, використовуючи команду **Add Control Points**, проведіть трансформацію псевдо-точки, позначеної білим значком **X**, і опорної точки номер 3 так, як описано у попередньому кроці. Знову перейдіть до масштабу всього знімка і подивіться, що ці точки також з'єднані лінією. Виконайте аналогічні дії з опорними точками 2 і 4 відповідно в північно-східному і південно-західному кутах знімка.

– Після завершення цих дій клацніть на кнопці  **View Link Table** на панелі інструментів **Georeferencing**.


У виконаних діях ArcGIS використовує поліноміальну трансформацію по опорних точках 1, 2, 3 і 4 і застосовує її таким чином, що вихідні місця розташування апроксимують задані вихідні місця розташування з використанням методу найменших квадратів. Поліноміальна трансформація найкращого наближення заснована на двох формулах: перша служить для розрахунку вихідної x -координати для вихідного (x, y) місця розташування, друга – для розрахунку y -координати для вихідного (x, y) місця розташування. В результаті розрахунку по методу найменших квадратів виводиться загальна формула, яка може бути застосована для всіх точок.

Коли ця загальна формула застосовується до опорної точки, то результатом розрахунку буде значення помилки. Ця помилка змінюється в

залежності від того, де задана початкова точка щодо фактично заданого положення (або від положення кінцевої точки, що задається). Чим більше використовується контрольних точок з однаковою якістю, тим точніший поліном може бути підібраний для трансформування вихідних даних по відношенню до вихідних координат.


Оскільки поки було задано тільки чотири опорні точки, ArcGIS може реалізувати трансформацію на основі полінома першого порядку, тобто так зване афінне перетворення. Кількість зв'язків, необхідних для трансформації растра, залежить від використовуваного методу. Однак застосування більшої кількості зв'язків не обов'язково призведе до більш точної прив'язки. В ідеалі, зв'язки, що вказують на просторові об'єкти, повинні бути розподілені по знімку, причому в кожному з його кутів має бути не менше одного зв'язку. Ступінь точності трансформації за всіма опорними точками вимірюється шляхом порівняння фактичного положення координат карти з їх розташуванням на трансформованому растрі. Ці вимірні значення для кожної зв'язку називаються помилками реєстрації.

Оцінка точності прив'язки зображення

- Клацніть на кнопці **Link Table**  на панелі інструментів **Georeferencing**, щоб подивитися значення помилки для знімка Slc_9910.jpg. На перший погляд, результати реєстрації виглядають дуже добре. Загальна помилка розраховується як сума середніх квадратичних відхилень (RMS) за всіма зв'язками. Ця помилка є оцінкою точності трансформації, але вона не обов'язково вказує на загальну якість прив'язки (реєстрації) знімка.

Проведення трансформації на основі полінома другого порядку

- Проведіть прив'язку опорних точок 5, 6, 7 і знову подивіться таблицю зв'язків. Кожна точка, що додається збільшує помилку RMS, але додавання опорних точок в центрі знімка покращує загальну точність прив'язки.

- Клацніть на кнопці **Link Table** . У діалоговому вікні, що відкрилося **Link Table** клацніть на випадуючому списку **Transformation** і виберіть **2nd Order Polynomial**. У ArcMap можна застосовувати більш складні математичні алгоритми для підгонки даних і виправлення помилки RMS. Точки, за якими отримана неприйнятно велика помилка, можна

вибирати, видаляти і замінювати. Для проведення трансформації на основі полінома другого порядку необхідно не менше шести точок прив'язки.

– Збережіть документ **ArcMap**. У процесі збереження документа ArcMap інформація про географічну прив'язку зберігається в окремому файлі з розширенням **.aux** і з тим же ім'ям, що і у растра.

– Для створення нового зображення знімка 1999 року в проекції **Utah State Plane** виберіть опцію **Rectify** зі списку панелі інструментів **Georeferencing**. Виберіть перерахунок з використанням методу **Nearest Neighbour** і збережіть знімок в папку **Images\UTSPN83C** в форматі TIFF під ім'ям **SLC_9910**. У форматі TIFF цей знімок буде більшого розміру, ніж вихідний файл у форматі JPG (рис. 4.116).

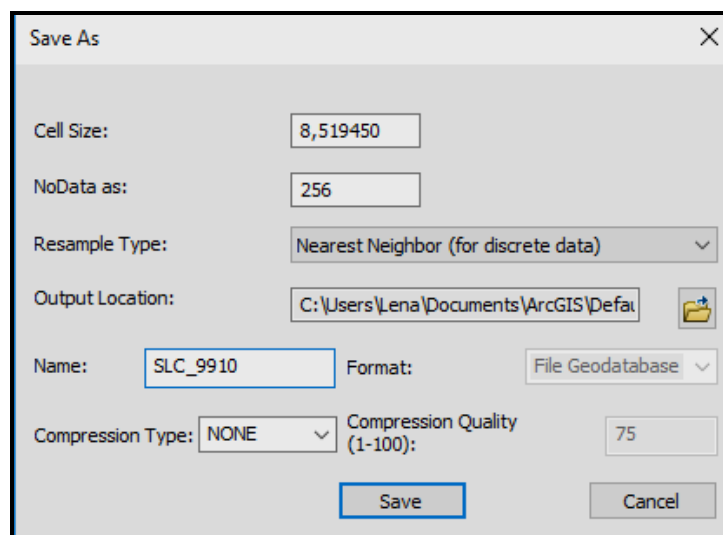


Рисунок 4.116 – Процес збереження зображення у форматі TIFF

– Додайте створений знімок у форматі TIFF до карти і відключіть його відображення. У форматі TIFF забезпечується коректне розташування знімка, але кольори іноді змінюються. Зате в ньому зберігається просторова прив'язка, і якщо потрібно, ви можете надати його разом з файлом прив'язки.

Порівняння растрів, датованих різними роками

– Після завершення прив'язки знімка, датованого жовтнем 1999 року, можна порівняти його зі знімком за липень 2001 року, на якому показана обстановка після завершення реконструкції автомагістралі I-15. Додайте до карти файл **Slc_0107.jpg**, що знаходиться в папці **LP_4\Images\UTSPN83C**. Помістіть цей знімок вище знімка **Slc_9910.jpg** в

таблиці змісту. Знімок **Slc_0107.jpg** вже прив'язаний в системі координат **Utah State Plane**, тому він завантажується прямо в просторову модель.

– Увімкніть і вимкніть відображення знімка **Slc_0107.jpg** за липень 2001 року при його збільшенні і переміщенні до західної частини долини Солт-Лейк-Сіті. Увімкніть відображення шару вулиць **teleatl1** і розмістіть його поверх знімків. Або виберіть самий верхній шар в таблиці змісту і використовуйте інструмент **Transparency** на панелі інструментів **Effects**, щоб подивитися, як змінюється загальна картина при тому чи іншому значенні прозорості знімків.

Детальніше розгляньте коридор навколо автомагістралі і відзначте зміни, що відбулися, наприклад, під'їзд до центру Солт-Лейк-Сіті і в районі з'єднання магістралей I-25/I-80. З'явилося нове приміщення близько опорної точки номер 7. Відзначте зміну кольорів між знімками. Це пов'язано з тим, що в жовтні 1999 року дерева вже скинули листу, а в липні 2001 року листя було рясним.

Географічна прив'язка зображень за координатами. Створення мозаїки растрів

У цьому завданні використовуються:

– карти, представлені растрами в форматі TIF – файли 1.tif, 2.tif (папка ЛР_5).

– координати прив'язки.

У програмі ArcMap необхідно виконати прив'язку растра **1.tif**, використовуючи координати кутів карти, наведені в таблиці 4.22.

Таблиця 4.22 – Координати углів карти для файла 1.tif

	X	Y
ПЗ	388409.473	6655876.040
ПС	444200.489	6654610.813
З	387288.593	6618750.127
С	443640.093	6617476.503

Для цього необхідно виконати наступні кроки.

– Додайте растр **1.tif** в таблицю змісту нового файлу, не звертаючи уваги на повідомлення, що у зображення відсутня інформація про географічну прив'язку і клацніть кнопку **ОК**.

- Виведіть панель **Georeferencing**.
- Збільшіть растрове зображення так, щоб добре було видно кути карти (рис. 4.117).

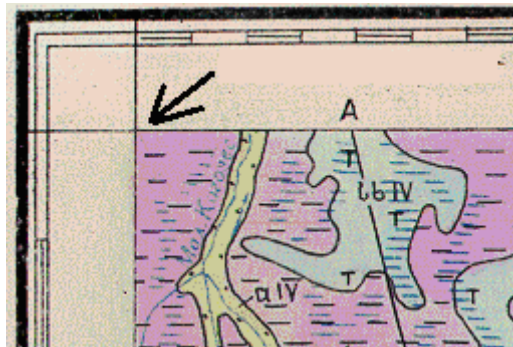



Рисунок 4.117 – Вид збільшеного фрагмента карти з точкою прив'язки

Після цього необхідно поєднати точки прив'язки карти з їх координатами. Для цього додають по черзі чотири реєстраційні точки і вводять координати цих точок. Почніть з верхнього лівого кута зображення (ПЗ точка в таблиці координат).

- Для цього скористайтеся на панелі інструментів **Georeferencing** кнопкою **Add Control Points** . Виконайте клік ЛКМ по опорній точці у верхньому лівому куті карти. Колір перехрестя зміниться, і не рухаючи ЛКМ, викличте контекстне меню. У контекстному меню виберіть команду **Input X and Y**. Введіть координати і натисніть кнопку **OK** у діалоговому вікні **Enter Coordinates** (рис. 4.118).

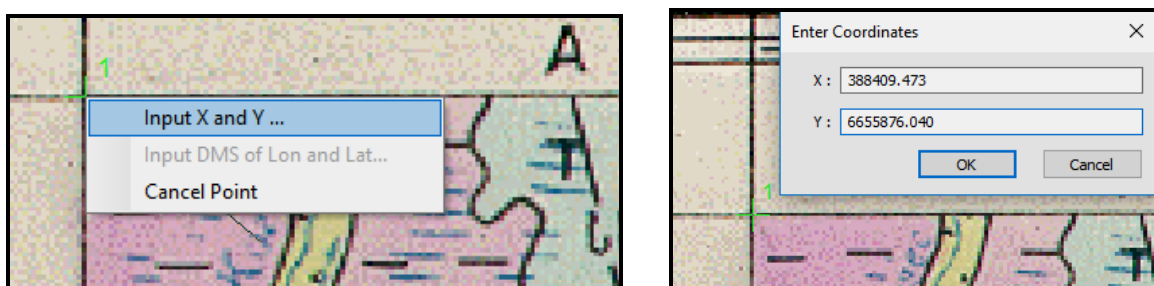

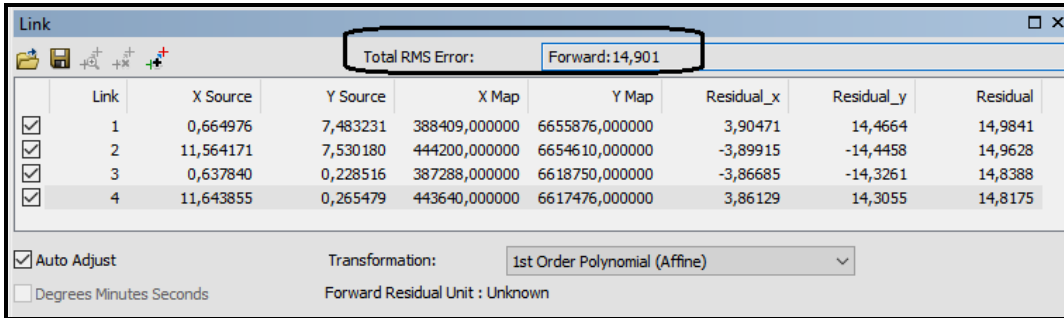


Рисунок 4.118 – Процес суміщення точок прив'язки карти з їх координатами

- Після введення координат першої точки растр може пропасти з екрану. У цьому випадку натисніть на кнопку **Full extent**  або використовуйте команду **Zoom To Layer** в контекстному меню шару.
- Так само введіть координати, що залишилися для трьох кутів

прикріпленого растра.

– Відкрийте таблицю зв'язків і перевірте помилки реєстрації растра (рис. 4.119). При появі значень, що різко відрізняються – помилок реєстрації растра, видаліть точку і знову повторіть процедуру введення реєстраційної точки. Растр повинен бути прив'язаний з точністю не більше 30 метрів.



Link	X Source	Y Source	X Map	Y Map	Residual_x	Residual_y	Residual
1	0,664976	7,483231	388409,000000	6655876,000000	3,90471	14,4664	14,9841
2	11,564171	7,530180	444200,000000	6654610,000000	-3,89915	-14,4458	14,9628
3	0,637840	0,228516	387288,000000	6618750,000000	-3,86685	-14,3261	14,8388
4	11,643855	0,265479	443640,000000	6617476,000000	3,86129	14,3055	14,8175

Рисунок 4.119 – Вид таблиці зв'язків

– Для створення нового трансформованого зображення знімка **1.tif** виберіть опцію **Rectify** зі списку панелі інструментів **Georeferencing**. Дайте новому файлу ім'я **Rectify1.tif** і збережіть в папку **LP_5**.

– Після трансформації видаліть растр **1.tif** з таблиці змісту **ArcMap** і додайте щойно створений растр **Rectify1.tif**. Після трансформації карта буде мати вигляд як на малюнку 4.120

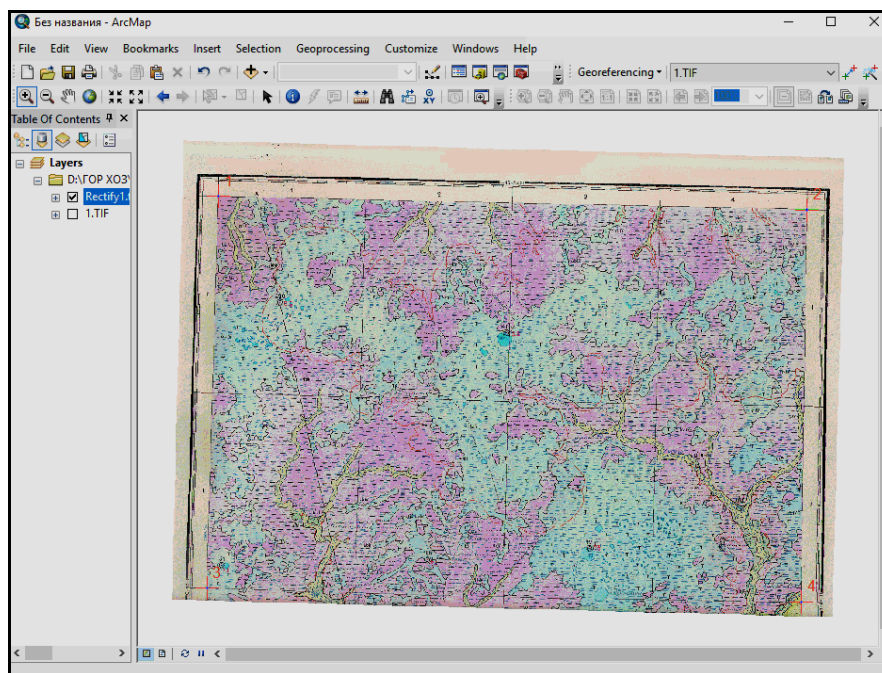


Рисунок 4.120 – Вид трансформованої карти

Для виконання прив'язки растра **2.tif** за вказаними в таблиці 4.23 реєстраційним точкам додайте в проект цей растр і повторіть описані вище дії.


В ході виконання прив'язки растра **2.tif** на панелі **Georeferencing** обов'язково вкажіть растр, для якого буде виконуватися прив'язка – **2.tif**.

Таблиця 4.23 – Координати углів карти для файлу 2.tif

	X	Y
З	387288.593	6618750.127
С	443640.093	6617476.503
ПС	443081.629	6580344.036
ПЗ	386171.571	6581625.885

В результаті виконання прив'язки растра **2.tif** повинні вийти два прив'язаних растра. Вони поєднані в одному координатному просторі і злегка повернені. Прив'язані растри можна об'єднати в один. Для цього виконується операція **Мозаїка**.

Об'єднання декількох растрів в один за допомогою операції Мозаїка

– У програмі **ArcMap**, відкрийте вікно **ArcToolbox**, натиснувши на кнопку **ArcToolbox**  на **Standard** панелі інструментів.

– У групі команд **Data Management Tools** знайдіть набір **Raster/Raster Dataset/Mosaic**.

– Увімкніть інструмент **Mosaic** і у вікні, що з'явилося, заповніть наступні поля (рис. 4.121):

Вхідні растри – **Rectify1.tif** і **2.tif**;

Цільовий растр – **Rectify1.tif**;

Метод мозаїки – **MINIMUM**;

Режим Mosaic Colormap – **FIRST**.

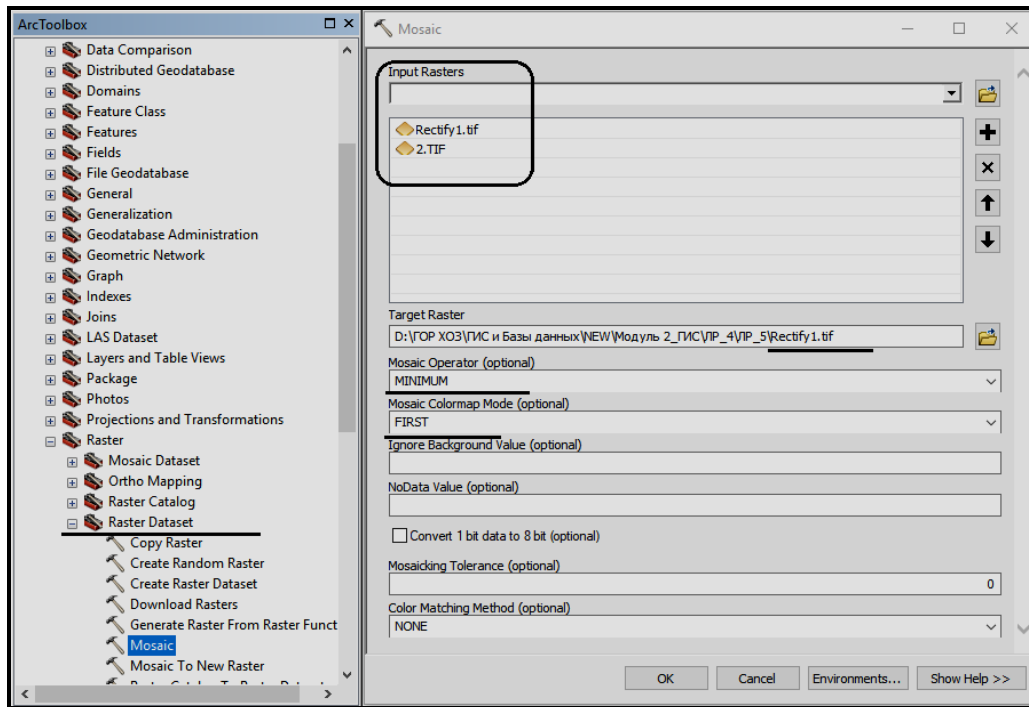


Рисунок 4.121 – Процес створення єдиного растра з декількох

Після виконання операції новий об'єднаний растр додався в проект. Тепер растр **Rectify1.tif** складається з двох частин, що злилися (рис. 4.122).

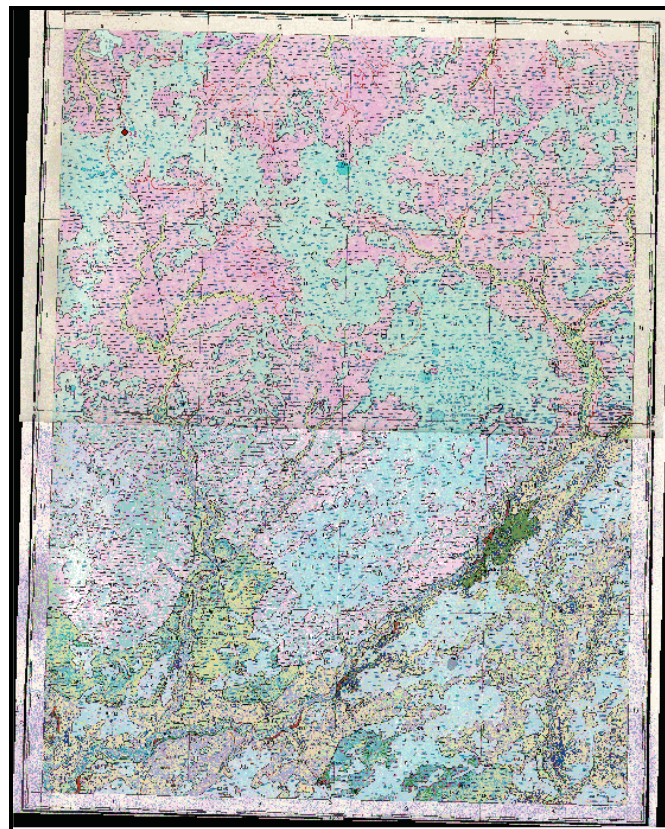


Рисунок 4.122 – Вид об'єднаного растра

Завдання для самостійного виконання

1. Якого типу перетворення можливо провести над растром 1.tif?
2. Яким чином внести зміни в координати вже заданим точкам прив'язки растра?
3. Видаліть всі растрові зображення з проекту. Додайте в проект отриманий файл **Rectify1.tif** і заповніть таблицю з координатами його реєстраційних точок.

	X	Y
СЗ		
СВ		
З		
В		

4. Порівняйте різні способи географічної прив'язки зображень. В яких випадках, на ваш погляд, краще використовувати прив'язку по координатах? А в яких – по векторним шарам карти?

Питання для самостійної роботи

1. Які основні операції мережевого аналізу існують в ГІС?
2. Які операції аналізу поверхонь існують в ГІС?
3. Перерахуйте вимірювальні операції, які здійснюються в ГІС. Наведіть приклади.
4. Де в виробництві використовуються ГІС? Наведіть приклади.
5. Наведіть приклади використання геокодованих просторових даних в екологічному моніторингу.
6. Охарактеризуйте складові ГІС проекту з моніторингу наслідків пожеж в лісовому господарстві.

4.10 Практичне заняття. Створення цифрової моделі карти

Мета: виробити вміння та навички зі створення векторних об'єктів карти та перевірки їх на відповідність заздалегідь створеним топологічним правилам.

Призначення: виконавши роботу, навчитися створювати шари з лінійними, точковими та полігональними об'єктами, а також перевіряти їхню коректність за допомогою топологічних правил.

ХІД РОБОТИ

У цій лабораторній роботі використовується карта у форматі TIF, прив'язана в координатах проекції Гаусса-Крюгера (файл **Rectify1.tif**, створений в практичній роботі 4).

Створення бази геоданих

– У **ArcCatalog** створіть базу геоданих **MAPS**, з новим набором класів просторових об'єктів **map_q**. Систему координат не задавайте (рис. 4.123).

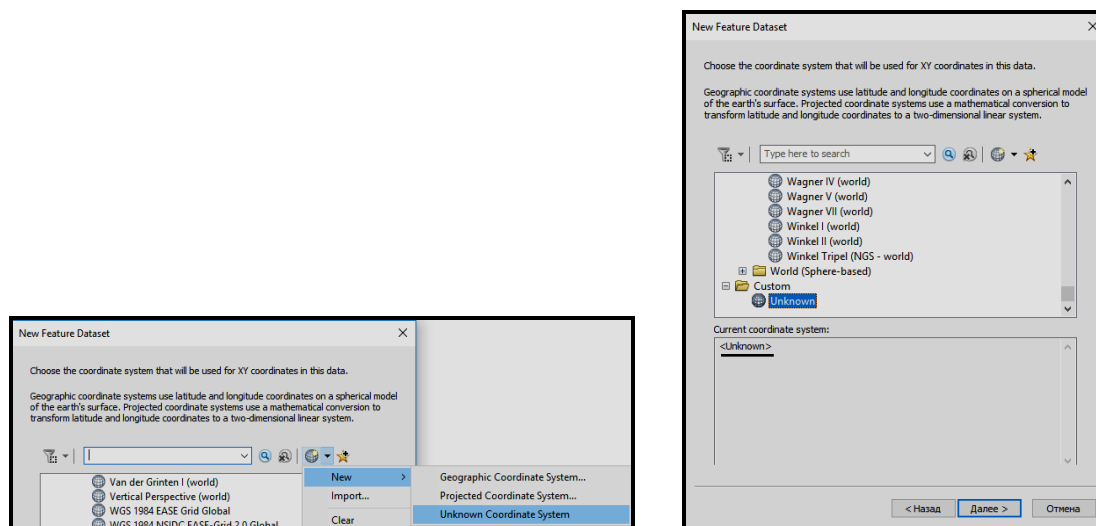


Рисунок 4.123 – Процес вибору системи координат

– У наборі класів просторових об'єктів **map_q** створіть класи просторових об'єктів (рис. 4.124):

1. **RAM_L** – рамка, лінійний;
2. **BAZA_P** – кордони четвертинних відкладень, полігональний;
3. **SKV** – свердловини, точковий.

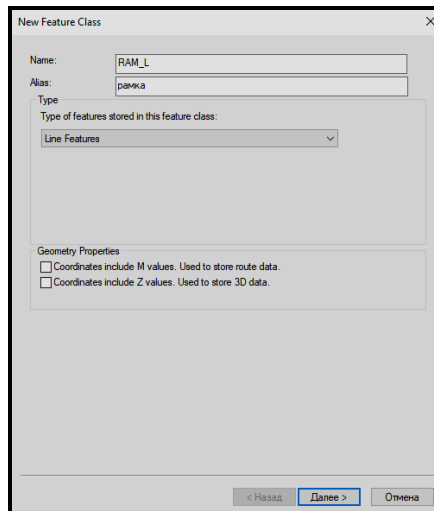


Рисунок 4.124 – Процес создания класса пространственных объектов

Створення топологічних правил

Топологія – це набір правил, які разом з інструментами і технологіями редагування дозволяють більш точно моделювати геометричні відносини в базі геоданих. Топологія описує просторові відносини між сполученими або прилеглими векторними об’єктами (точками, лініями і полігонами) в ГІС. Топологічні дані корисні при пошуку і виправлення помилок оцифровки (наприклад, дві лінії в шарі доріг не з’єднані в точці перетину).

Необхідно створити правила топології для лінійного набору просторових об’єктів **RAM_L**.

– Для цього в **ArcCatalog** для класу просторових об’єктів **map_q** створіть новий тип даних – **Topology**.

– Додайте в набір топологічних правил **Topology_line** лінійний клас об’єктів **RAM_L**.

– Додайте в набір топологічних правил **Topology_line** наступні правила поведінки лінійних об’єктів (рис. 4.125, 4.126):

1. **Mast Not Have Dangles**. Говорить про те, що лінійний об’єкт одного шару повинен стикатися з лінійним об’єктом з цього ж шару в обох кінцевих точках.

2. **Mast Not Intersect**. Говорить про те, що лінія не повинна перетинати або мати загальні сегменти з іншими лініями цього ж шару).

3. **Mast Not Overlap**. Говорить про те, що лінії не повинні мати перетину або перекриттів між собою. Лінія може торкатися іншої лінії в

кінцевій точці.

4. **Mast Not Intersect or Touch Interior.** Говорить про те, що лінійний об'єкт не повинен перетинати або торкатися лінійного об'єкта того ж шару, крім як в кінцевих точках.

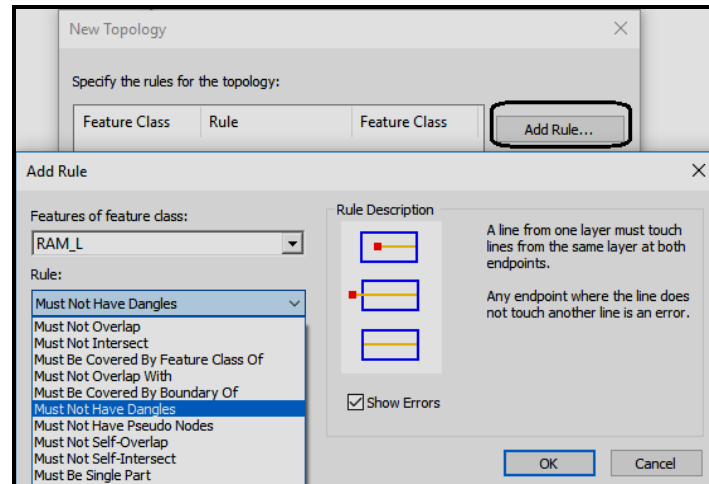


Рисунок 4.125 – Процес вибору топологічних правил

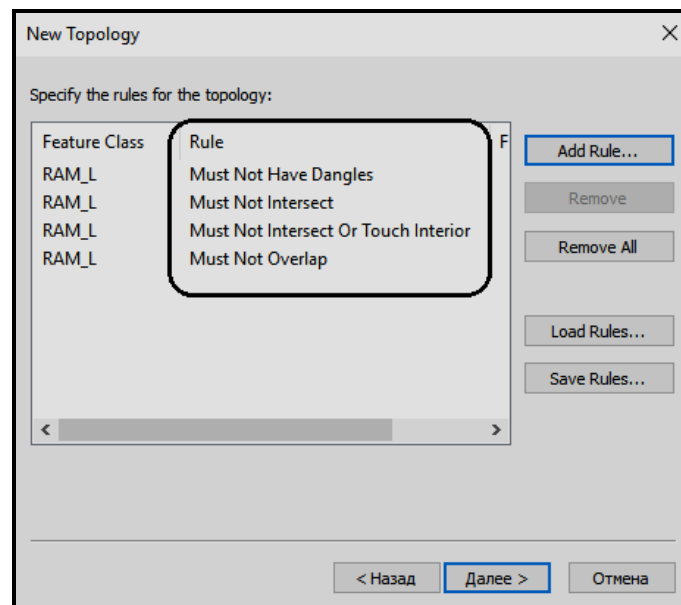


Рисунок 4.126 – Вид вікна мастера створення топології з переліком обраних правил

Після завдання правил топології перелік об'єктів набору класу просторових об'єктів **map_q** поповниться об'єктом **Topology_line** (рис. 4.127).

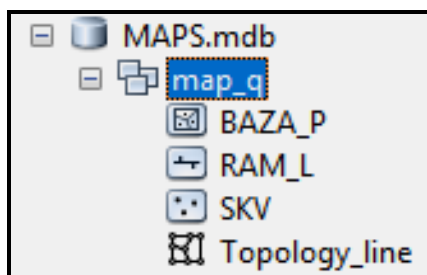


Рисунок 4.127 – Вид об’єктів в наборі класів map_q

- Закрийте вікно програми **ArcCatalog**.

Створення векторних об’єктів карти

- Відкрийте **ArcMap** з порожньою картою. Додайте прив’язаний в практичній роботі 4 растр **Rectify1.tif** і нові набори просторових об’єктів: **RAM_L**, **BAZA_P**, **SKV**, **Topology_line**.

Для кодування ліній рамки необхідно додати призначене для користувача поле **L_Code** до існуючих стандартних полів **OBJECTID ***, **Shape ***, **Shape_Length** лінійного просторового об’єкта **RAM_L**.

- Для цього відкрийте атрибутивом таблицю лінійного просторового об’єкта **RAM_L** – рамка. У вікні атрибутивної таблиці в випадаючому списку **Options** виберіть команду **Add Field** (рис. 4.128).

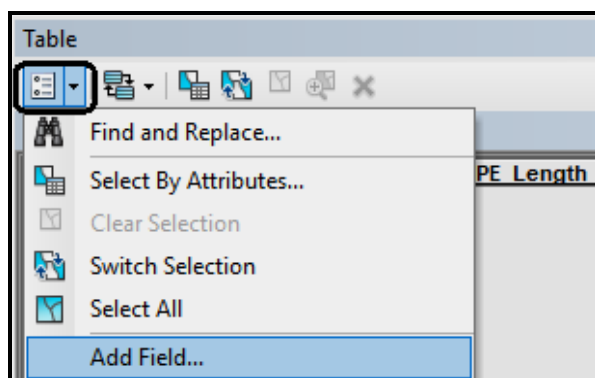


Рисунок 4.128 – Процес додавання поля в атрибутивну таблицю

- Заповніть вікно **Add Field**. Ім’я поля: **L_Code**, тип даних: **Short Integer**. Тип поля визначається типом даних, які в ньому зберігаються (в даному випадку це будуть цілі числа, наприклад, код рамки – 4100) (рис. 4.129).

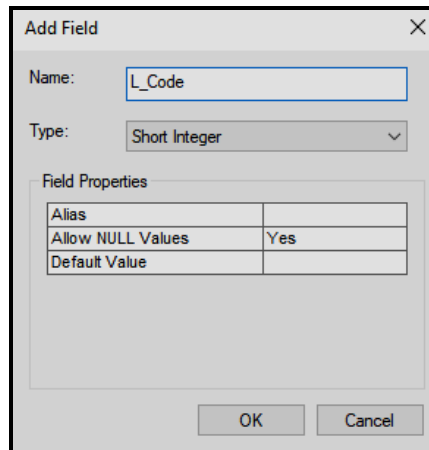


Рисунок 4.129 – Процес заповнення даних поля атрибутивної таблиці

Начало редагування

Сеанс редагування карти починається з підключення панелі **Editor**. Для створення за допомогою **ArcMap** нового векторного об'єкта необхідно побудувати **Sketch**, що буде редагуватися. Ескіз – це форма, яку ви малюєте шляхом оцифровки її вершин. Над ескізом можна виконувати різні операції редагування. За допомогою ескізу можна створювати нові об'єкти, модифікувати їх, подовжувати або обрізати об'єкти, змінювати форму об'єктів. Ескіз складається з вершин – точок, в яких ескіз змінює напрямок і сегментів – ліній, що з'єднують вершини. Початкова і кінцева точка ескізу називається вузлом (кінцевою точкою).

Тип створюваного об'єкта обирається на панелі інструментів **Construction Tools** (рис. 4.130).

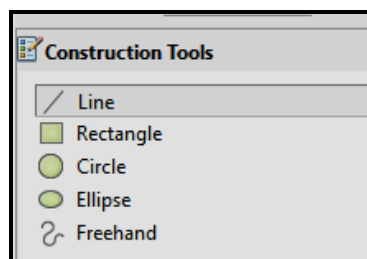


Рисунок 4.130 – Вид панелі рисовання

Список **Create Features** складається з імен всіх шарів в наборах даних, які були створені в **ArcCatalog**. Для створення **Point** потрібно клацнути на карті один раз.

Перш ніж створювати лінійні об'єкти карти необхідно подбати про дотримання раніше створених правил поведінки об'єктів – топології. За

правилами топології необхідно, щоб були відсутні «висячі» лінійні сегменти (крім особливо обумовлених випадків). Необхідно налаштувати автоматичну прив'язку до характерних точок (кінцевих точок ліній, проміжним вершинам), щоб вузол лінії точно поєднувався з іншим вузлом або сегментом ескізу.

Допуск замикавання дозволяє забезпечити точність розташування щодо інших об'єктів. Для цього потрібно встановити допуск замикавання (відстань, в межах якої курсор або об'єкт поєднується з іншою точкою об'єкта). Якщо місце, до якого відбувається прив'язка знаходиться в межах заданої відстані, курсор автоматично поєднується (притягується) до цієї точки.

Установка допуску замикавання

– У списку панелі **Editor** виберіть команду **Options** у вікні, перейдіть на вкладку **General**. У вікні **Editing Options** встановіть допуск **20** пікселів (рис. 4.131).

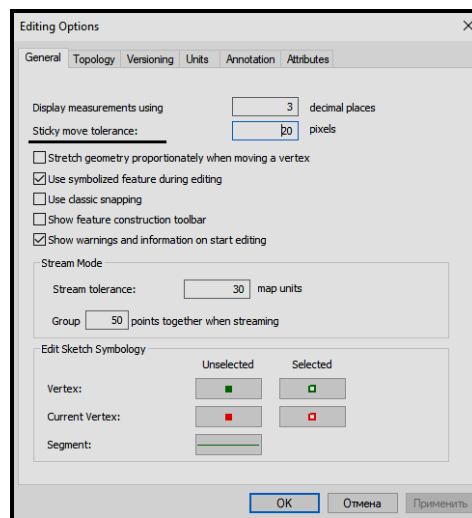


Рисунок 4.131 – Процес налаштування допуску замикавання

Установка пріоритету і характерних точок при замиканні об'єктів

Для дотримання точності в процесі векторизації шару **RAM_L** – рамки карти, необхідно налаштувати автоматичну прив'язку до характерних точок ескізу.

– Для цього необхідно вивести панель **Snapping**, вибравши її в випадаючому списку панелі **Editor** і зазначивши необхідні точки для прив'язок (рис. 4.132).

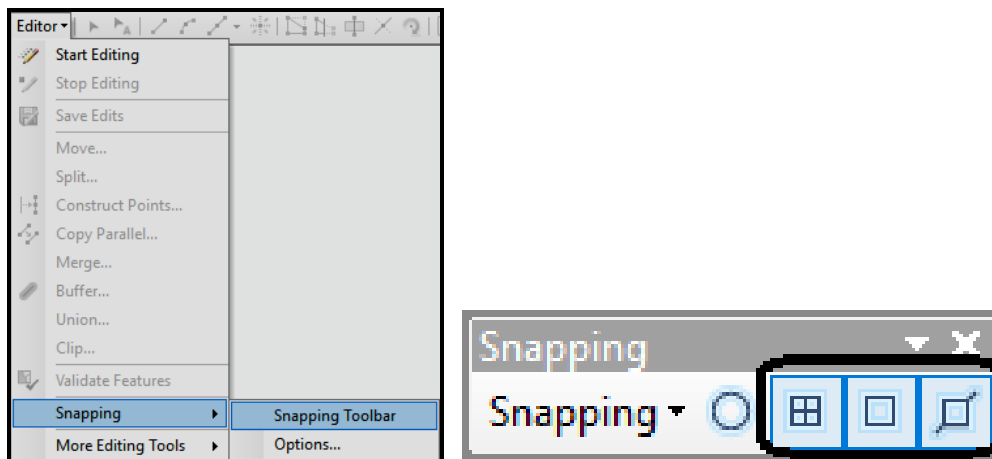



Рисунок 4.132 – Процес налаштування прив'язки до точок ескізу

Зверніть увагу на те, що в разі замикання сегментів ескізу не тільки всередині одного шару, а між кількома шарами, пріоритет замикання визначає порядок шарів в списку **Table of Contents**. Спочатку виконується замикання по шару на початку списку (зверху), а потім по кожному наступному шару зі списку. Пріоритет замикання можна замінити, помінявши шари місцями.

Створення лінійних об'єктів карти

- Перейдіть в режим редагування карти, вибравши на панелі **Editor** у меню, що випадає команду **Start Editing**.
- Спочатку необхідно створити рамку карти. Для цього на панелі **Editor** натисніть на кнопку **Create Features** . У вікні **Create Features** виберіть цільовий шар – шар **Рамка** і клацніть по кнопці **Line**.
- Клацніть лівою кнопкою миші в верхньому лівому куті карти і тягніть скетч вправо, до північно-східного кута карти, клацніть мишкою, поставивши вершину, потім вниз до південно-східного кута, вліво до південно-західного кута і вгору, до першої вершини. Клацніть два рази і завершіть скетч. Тим самим була створена рамка карти в вигляді прямокутника.


Завершити ескіз можна трьома способами:

- двічі клацнути лівою кнопкою мишки;
- використовуючи контекстне меню ескізу **Finish Sketch**;
- натиснути клавішу **F2** на клавіатурі.

Перевірка топології



- Виведіть панель **Topology**. За допомогою кнопки **Select Topology**

 виберіть шар карти **Рамка** для редагування топології.

– За допомогою кнопок  на панелі **Topology** перевірте дотримання топологічних правил в створеному векторному об'єкті – рамці карти. Ці команди дозволяють перевіряти дотримання топологічних правил або для всіх об'єктів шару, або тільки для тих, які видно на екрані монітора – це прискорює час перевірки.

У разі виявлення помилок, вони будуть підсвічені червоним кольором. Після побудови топології шару перевірте, замкнувся ескіз, чи ні, чи немає «висячих» вузлів)? Присутність «висячих» вузлів означає, що відстань між кінцевими вузлами лінії більше, ніж встановлений кластерний допуск. Якщо помилок багато, можна збільшити кластерний допуск і знову перевірити топологію. Якщо вони поодинокі, то простіше виправить їх вручну.

– Для виправлення помилок векторизації необхідно відключити шар растра – це збільшить чіткість перегляду векторів.

– На панелі **Topology** клацніть по кнопці **Edit Vertices**. У разі неправильного замикання лінійних сегментів рамки клацніть по інструменту редагування вершин  на панелі **Edit Vertices** і підтягніть вузли один до одного. Завершіть редагування вершин клацанням по кнопці **Finish Sketch**  (рис. 4.133).

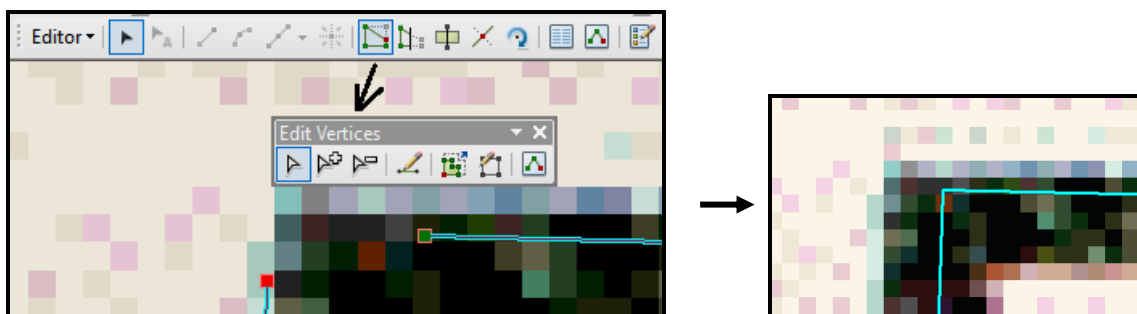



Рисунок 4.133 – Процес редагування вершин рамки

– Увімкніть перевірку топології ще раз. Переконайтеся в тому, що помилок більше немає.

– У цьому ж шарі оцифруйте лінії сітки.

– Ще раз запустіть перевірку топології. При наявності помилок, виправте їх. Для того щоб зрозуміти, які саме топологічні правила порушені, зручно відкрити вікно **Error Inspector** за допомогою відповідної

кнопки  на панелі на панелі **Topology**. У цьому вікні за допомогою кнопки **Search Now** можна проаналізувати яке саме правило і якими векторними об'єктами порушено. При клацанні ЛКМ по правилу, відповідна вершина виділяється чорним кольором (рис. 4.134).

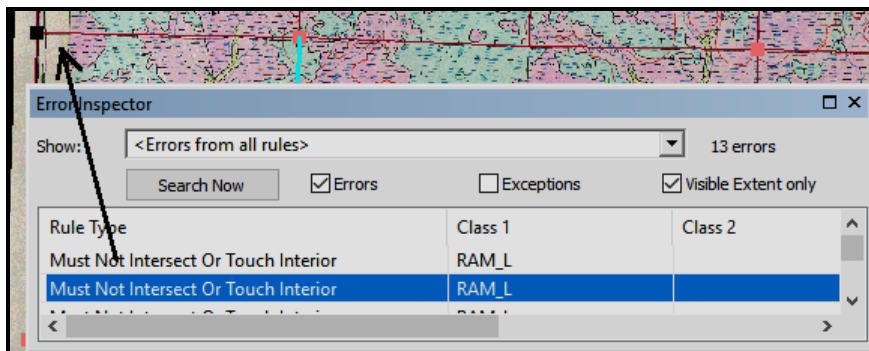


Рисунок 4.134 – Процес перевірки топологічних правил за допомогою вікна Поиск ошибок

– Відредагуйте лінії на шарі **Рамка**, таким чином, щоб топологія дотримувалася. Скористайтеся для цього відповідними командами на панелі **Editor** (рис. 5.13).



Рисунок 5.13 – Вид команд редагування векторних об'єктів

– Ще раз запустіть перевірку топології. Переконайтеся в тому, що помилок більше немає.

Зверніть увагу на те, що в режимі редагування карти немає можливості виробляти додавання або редагування топологічних правил в **ArcCatalog**. Для внесення будь-яких змін в шари або їх властивості необхідно завершити редагування і закрити **ArcMap**.

Створення точкових об'єктів карти

– У таблиці вмісту зробіть активним шар **Скважини**.
 – Створіть точковий об'єкт на місці свердловини (рис. 4.135).
 Оцифрує всі свердловини на карті.

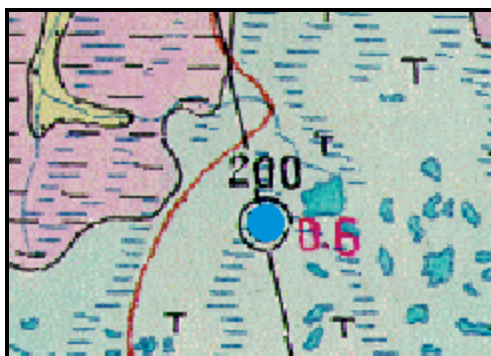
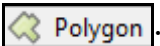


Рисунок 4.135 – Вид точкового об'єкта на карті

Створення полігональних об'єктів карти

– У таблиці вмісту зробить активним шар **Границі четвертинних відкладень**.

– Оцифрує кордони четвертинних відкладень за допомогою **Полігонов (Polygon)** . Поставте курсор в початкову точку полігону, клацніть ЛКМ і ведіть лінію по межі, ставлячи вершини в усіх точках перегину кордону, клацаючи мишкою. Завершіть полігон, використовуючи контекстне меню ескізу **Finish Sketch**.

– Вид дуги з вершинами після оцифровки буде приблизно як на малюнку 4.136.

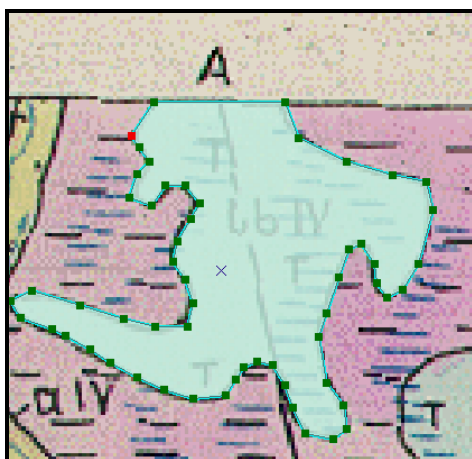


Рисунок 4.136 – Вид полігону на карті

– оцифруйте всі межі четвертинних відкладень в межах будь-якого квадрата карти.

Завдання для самостійного виконання

1. Скільки записів в атрибутивній таблиці шару **Рамка**? Чому саме така кількість?
2. Перерахуйте поля в атрибутивній таблиці шару **Рамка**. Поясніть призначення кожного з полів.
3. Створіть топологічні правила для набору просторових об'єктів **Границы четвертичных отложений**, які будуть дозволяти контролювати те, що полігональні області четвертинних відкладень осадових порід не перекриваються одна з одною.
4. Змініть властивість полігонального шару **Границі четвертинних відкладень** таким чином, щоб крізь заливку полігону була видна підкладка карти.
5. Скільки записів в атрибутивній таблиці шару **Скважини**? Чому саме така кількість?

Питання для самостійної роботи

1. Структура системи ДЗЗ. Наземний і орбітальний сегменти. Способи передачі даних ДЗЗ.
2. Якими способами зі знімків витягується необхідна для досліджень інформація?
3. Які існують методи дешифрування?
4. Перерахуйте переваги автоматичного дешифрування знімків перед візуальними.
5. Які види аерокосмічних знімків існують?
6. Основні характеристики даних ДЗЗ.
7. Вимоги, що пред'являються до даних ДЗЗ при вирішенні різних прикладних задач.

4.11 Індивідуальне завдання

Загальні вимоги до виконання

Розроблювана БД повинна включати такі об'єкти: таблиці, запити, форми, звіти та макроси.

Схема бази даних, необхідна кількість таблиць, їхня структура та властивості полів визначаються студентом самостійно, виходячи із постановки задачі згідно з варіантом завдання. Кількість записів у довідкових таблицях повинна бути не менше 7–10, а в таблицях із змінними даними – 30 записів.

Відбір даних для аналізу реалізується в запитах. Необхідна кількість запитів визначається студентом самостійно згідно з постановкою задачі, але в розроблюваній БД повинні бути подані всі види запитів: запити на вибірку даних з різними умовами відбору даних; запити з обчислюваними полями; підсумкові запити; перехресні запити; запити на оновлення, видалення та додавання даних, а також запити на створення таблиць.

Інтерфейс користувача реалізується за допомогою форм, кількість і зовнішній вигляд яких визначаються студентом самостійно, але в розроблюваній БД повинні бути створені такі форми: головна кнопкова форма; форми для редагування довідників; форма для редагування таблиць зі змінною інформацією; форми з аналітичними даними, реалізовані у вигляді підпорядкованих або пов'язаних форм, які вміщують різні види діаграм. У БД обов'язково потрібно використовувати зведені таблиці та діаграми для аналізу процесів. Робота форм повинна бути автоматизована за допомогою макросів, які включають макрокоманди для роботи із записами, керування застосування та фільтри.

Кількість і зміст звітів визначається студентом самостійно відповідно до поставленої задачі, але у створених звітах повинно бути використано всі можливі розділи для звіту; реалізовано декілька (не менше двох) рівнів групування даних; організована нумерація записів за допомогою поля із сумою з накопиченням; використані обчислювані поля.

Індивідуальні навчально-дослідні завдання (ІНДЗ) мають на меті:

– систематизацію, закріплення, розширення теоретичних і практичних знань, умінь з питань створення проєктів застосувань за допомогою СКБД Access;

- використання набутих знань і вмінь під час реалізації конкретних економічних проєктів;
- набуття компетентності в галузі застосування технологій баз даних до розв’язання інформаційних задач у фаховій діяльності;
- розвиток навичок самостійної організації роботи при виконанні ІНДЗ.

Актуальність цього виду роботи зумовлена розповсюдженістю проблем збирання, зберігання та ефективної обробки інформації при організації та повсякденному веденні справ.

Отже, індивідуальні завдання є важливою частиною навчального модуля дисципліни та виконуються студентами самостійно під керівництвом викладачів, які забезпечують викладання дисципліни.

Теми ІНДЗ охоплюють широке коло різноманітних прикладних завдань, які відповідають реальним ситуаціям і пов’язані єдиними підходами та методами для їхнього виконання в середовищі MS Access: розробка логічної структури розподілу даних і зв’язків між ними, визначення відповідних типів та форматів для зберігання даних, створення дружнього інтерфейсу користувача, виконання необхідного аналізу даних, подання результатів аналізу у вигляді звітів тощо. Перелік тем ІНДЗ наведено в таблиці 4.25.

Основною **метою** індивідуального завдання є формування у студентів основ професійного підходу до розв’язання інформаційних задач у майбутній фаховій діяльності: вміння сформулювати постановку задачі; знаходити, обмежувати та обґрунтовувати саме ті атрибути предметної області, які необхідні для її розв’язання; розуміння та вміння формулювати задачі аналізу, а також здійснювати програмну реалізацію з використанням системи керування базами даних MS Access та інших застосувань MS Office.

Усі студенти виконують роботу за таким загальним змістом:

- постановка задачі. Розробка логічної схеми БД та перелік аналітичних задач;
- створення таблиць БД, заповнення таблиць економічно обґрунтованими даними;
- проєктування запитів як основного інструмента розв’язання інформаційних задач;
- проєктування форм як основних елементів інтерфейсу

користувача створюваного додатка (бази даних);

– технологія створення, редагування та використання звітів у базі даних MS Access;

– автоматизація та створення застосувань у БД.

Таблиця 4.25 – Шкала оцінювання ІЗ

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проєкту, ІНДЗ	для заліку
90 – 100	A	відмінно	зараховано
82 – 89	B	добре	
74 – 81	C		
64 – 73	D	задовільно	
60 – 63	E		
35 – 59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0 – 34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Аналітична частина завдання полягає в постановці задачі та визначенні завдяки цьому необхідної для зберігання в базі інформації, побудові логічної схеми бази даних у вигляді пов'язаних між собою таблиць, визначенні властивостей атрибутів (полів) таблиць та властивостей зв'язків між ними.

Дослідницька частина роботи полягає в тому, що студенти повинні самостійно з'ясувати необхідну інформацію стосовно заданої предметної області, виконати розподіл інформації на довідкову та змінну, визначити такі обмеження щодо конкретних даних, які будуть зберігатися в базі, щоб ці дані відображали можливу реальну ситуацію в предметній області, а також повинні самостійно визначити коло аналітичних задач, пов'язаних із діяльністю, яка фіксується в базі.

Проєктна частина ІНДЗ передбачає безпосередню роботу в MS Access і створення необхідних об'єктів бази із визначеними властивостями.

У результаті виконання індивідуального завдання студент повинен

розробити інтерфейс користувача за допомогою відповідних форм та провести економічний аналіз даних за допомогою відповідних об'єктів бази.

Кожен студент оформляє звіти засобами MS Word, які зберігаються на будь-яких зовнішніх носіях інформації, роздруковуються та захищаються. Результати роботи та захисту оцінюються згідно зі шкалою, що подана в таблиці 4.1.

Варіанти завдань

Для свого варіанта згідно зі студентським журналом (табл. 4.26) розробити базу даних і створити файл бази даних у MS Access.

Таблиця 4.26 – Варіанти завдань

№ варіанту	Назва	Загальна постановка задачі
1	2	3
1	Автомайстерня	Облік виконаних ремонтних робіт в автомайстерні для поточного та підсумкового аналізу роботи майстерні. Ведеться облік клієнтів та автомобілів
2	Мережа АЗС	Облік операцій із продажу різних видів палива в мережі АЗС. Ведеться облік поставок і постачальників
3	Аптека	Облік операцій із реалізації медичних препаратів при роздрібній торгівлі в окремій аптеці міста. Ведеться облік поставок препаратів і постачальників
4	Бібліотека	Облік операцій обслуговування читачів. Ведеться облік читачів, літератури, а також роботи персоналу бібліотеки
5	ДАІ. Технічний догляд автомобілів	Облік технічного стану автомобілів на рівні районної ДАІ
6	Клієнти готелю	Оперативний облік обслуговування клієнтів готелю. Ведеться облік клієнтів, персоналу, номерів
7	Деканат. Успішність студентів на одному курсі	Оперативний облік успішності студентів за дисциплінами одного курсу
8	Кадрова агенція	Облік резюме здобувачів та заявок роботодавців, а також облік виконаного підбору персоналу (укладених контрактів)

Продовження таблиці 4.26

1	2	3
9	Магазин «Книжковий світ»	Облік роздрібного продажу книжок у книжковому магазині. Необхідно мати інформацію про постачальників і постачання літератури різних жанрів
10	Меблі на замовлення	Облік видів меблів, замовників, майстрів, матеріалів, комплектування
11	Обмін валют	Облік роздрібного продажу та купівлі валют у мережі пунктів обміну. У базі фіксується кожна проведена операція з обміну валюти, а також поточний курс купівлі та продажу
12	Продаж авіаквитків	Облік рейсів, пасажирів, продажу квитків авіакомпанією
13	Продаж залізничних квитків	Облік залізничних маршрутів, зон, продажу квитків
14	Розклад занять у ВНЗ	Облік аудиторій, дисциплін, викладачів, груп, розкладу
15	Склад. Облік матеріалів	Облік постачальників, споживачів, матеріалів, облік прийому/відпуску матеріалів
16	Ремонтна майстерня	Облік видів техніки, клієнтів, облік ремонтних операцій
17	Фірма з продажу комп'ютерів	Облік комплектвальних ПК, формування конфігурації
18	Поліклініка	Оперативний облік звертань пацієнтів до різних лікарів, діагнози
19	Концерти	Організація концертів, артисти, номери, місця проведення
20	Спортивні досягнення	Облік змагань, суперників, результатів, складу команд
21	Виробництво продукції	Облік видів продукції, матеріалів і комплектуючих, собівартості продукції (планової та фактичної). Випуск продукції
22	Будівництво котеджу	Облік усіх заходів, трудових та матеріальних ресурсів, необхідних для будівництва невеликої одно- або двоповерхової будівлі
23	Облік маршрутів міського транспорту	Облік маршрутів міського транспорту (автобусів, електротранспорту) з урахуванням кількості пасажирів та зупинок, а також графік руху
24	Паспортизація геодезичного устаткування	Облік термінів перевірок придатності та надійності роботи обладнання
25	Облік комунальних платежів мешканцями квартир	Облік щомісячної плати за квартиру, опалення, воду, каналізацію, електроенергію, газ

Виконання індивідуального завдання

Індивідуальне завдання рекомендується виконувати за кроками у такій послідовності.

Крок 1. Постановка задачі. Розробка логічної схеми БД та переліку аналітичних задач

Відповідно до індивідуального завдання описати конкретну предметну область, яка вимагає розроблення відповідної бази даних. Сформулювати перелік основних задач, які будуть розв'язуватися на основі даних, що зберігаються в БД. З'ясувати, яка саме та в якому вигляді інформація буде зберігатися в базі, на підставі цього визначити тип даних для кожного атрибута й діапазон імовірних значень. Розробити детальну логічну схему з дотриманням правил нормалізації.

Визначити ключові поля, з'ясувати типи відношень між пов'язаними полями. Розробити схему бази даних.

Отримані результати оформити як документ «Звіт з розрахунково-графічної роботи з дисципліни «Бази даних» засобами Word. У ньому висвітлити такі питання:

- предметна область, у якій буде використовуватися БД;
- основні задачі, які будуть розв'язуватися на основі даних, що зберігаються в БД;
- схема БД із зазначенням ключових полів, зв'язків між таблицями, їхніми видами та призначенням таблиць;
- структура кожної таблиці (імена полів, типи даних, властивості);
- зовнішні джерела даних.

Проект бази даних туристичної агенції

Предметна область

Предметною областю для розроблюваної бази даних є надання інформаційних та туристичних послуг населенню туристичною агенцією. Передбачається, що користувачами бази можуть бути менеджери агенції, які безпосередньо реєструють придбані населенням тури (подорожі), виконують пошук та підбір турів, а також мають змогу проводити оперативний аналіз виконаної роботи: підраховувати кількість проданих турів, зокрема кожним менеджером за певний проміжок часу, попит на

певну категорію подорожей (країну).

Основні задачі

За допомогою бази даних розв'язуються такі задачі:

- Облік видів сервісу за наданими турами.
- Облік проданих турів (укладених договорів) клієнтам.
- Облік роботи кожного менеджера.
- Аналіз господарської діяльності туристичної агенції.

Отже, основною метою розробки такої бази даних є реалізація можливості поповнення та зберігання інформації про туристичні подорожі. Виходячи з аналізу предметної області й поставлених задач, робиться висновок щодо інформації, яка буде зберігатися в базі. У цьому випадку це є інформація про клієнтів агенції, наявні тури (подорожі), країни, умови проживання в готелях та транспорт, співробітників агенції, а також інформація про власне укладені договори (основна робоча таблиця бази).

Схема бази даних

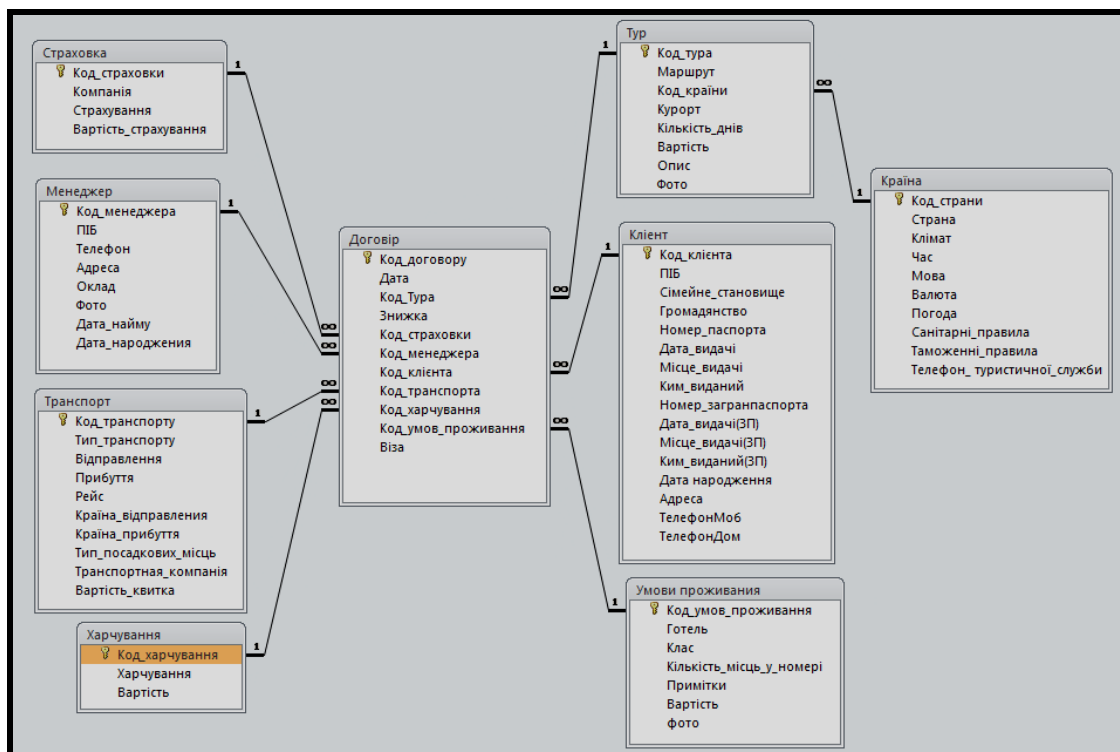


Рисунок 4.137 – Завершений вигляд схеми даних для БД «Туристична агенція»

Таблиці мають таке призначення:

Страховка – прайс-лист послуг зі страхування життя (довідкові дані).

Менеджер – дані про співробітників (довідкові дані).

Транспорт – дані про транспортні послуги (довідкові дані).

Харчування – прайс-лист послуг харчування туристів (довідкові дані).

Тур – опис путівки (довідкові дані).

Клієнт – дані про клієнтів (довідкові дані).

Умови проживання – опис номерів та послуг проживання (довідкові дані).

Країна – дані про країну у яку їде турист (довідкові дані).

Договір – облік укладених договорів з клієнтами (змінні дані).

Структура таблиць

Нижче подано структуру довідкових таблиць і таблиць зі змінними даними.

Поля довідкової таблиці з типами даних надано у таблиці 4.27.

Таблиця 4.27 – Таблиця **Страховка**

Ім'я поля у БД	Тип даних (властивість)
Код страховки	Счетчик (длинное целое)
Компанія	Текстовый (25 символів)
Страхування	Текстовый (15 символів)
Вартість страхування	Денежный (формат поля – процентный)

У такий спосіб навести структуру всіх таблиць-довідників.

Таблиці зі змінними даними (табл. 4.28).

Таблиця 4.28 – Таблиця **Договір**

Ім'я поля у БД	Тип даних (властивість)
Код_договору	Счетчик (длинное целое)
Дата	Дата/время
Код_Тура	Числовой (длинное целое)
Знижка	Денежный (формат поля – процентный)
Код страховки	Числовой (длинное целое)
Код менеджера	Числовой (длинное целое)
Код клієнта	Числовой (длинное целое)
Код транспорту	Числовой (длинное целое)
Код харчування	Числовой (длинное целое)
Код умов проживання	Числовой (длинное целое)
Віза	Логический

У такий спосіб навести структуру всіх таблиць зі змінними даними.

Крок 2. Створення таблиць БД і їх заповнення

На основі виконаного на першому кроці аналізу поставленої задачі й розробленої логічної схеми даних розпочати до створення бази, починаючи зі створення таблиць. При визначенні властивостей полів необхідно звертати увагу й задавати не тільки тип та формат даних, але й додаткові властивості, такі як значення за замовчуванням, умова на значення, маска введення (якщо це є доцільним) та ін.

Внести дані до бази в такій послідовності: спочатку заповнюються батьківські таблиці, а потім – дочірні. Створити та визначити властивості зв'язків між первинними та зовнішніми ключами таблиць.

За результатами оформити звіт, у якому:

- описати додаткові властивості окремих полів і таблиць загалом (значення за замовчуванням, умови на значення, маски введення тощо) (рис. 1, 2);
- надрукувати дані таблиць.

Нижче подано маски введення даних, які вимагають дотримання певного формату, наприклад номери телефонів, різноманітні поля зі списками, поля дат та ін. Відповідні приклади показані на рисунках 4.138, 4.139.

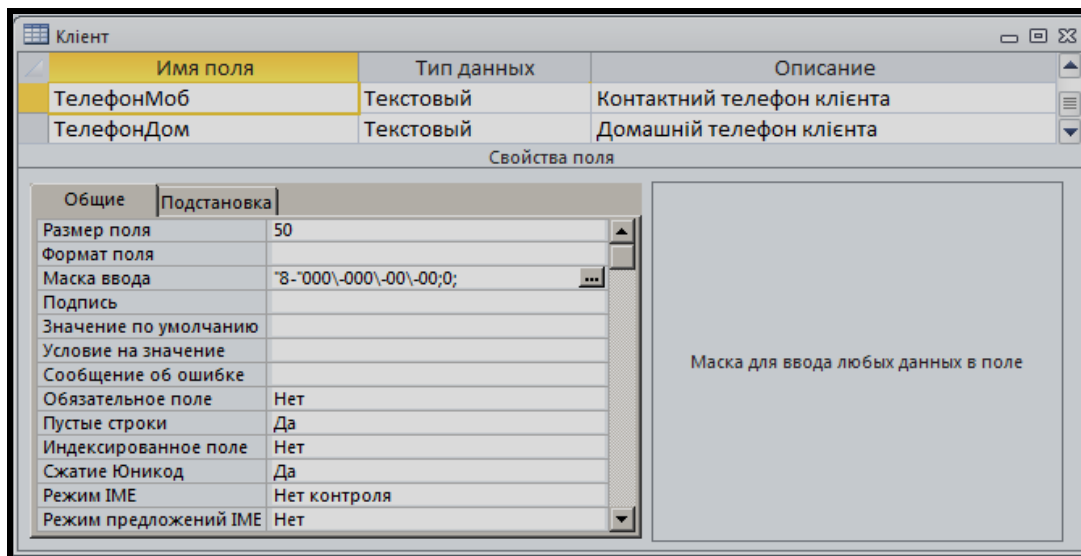


Рисунок 4.138 – Використання маски введення для забезпечення єдиного формату введення номера телефону

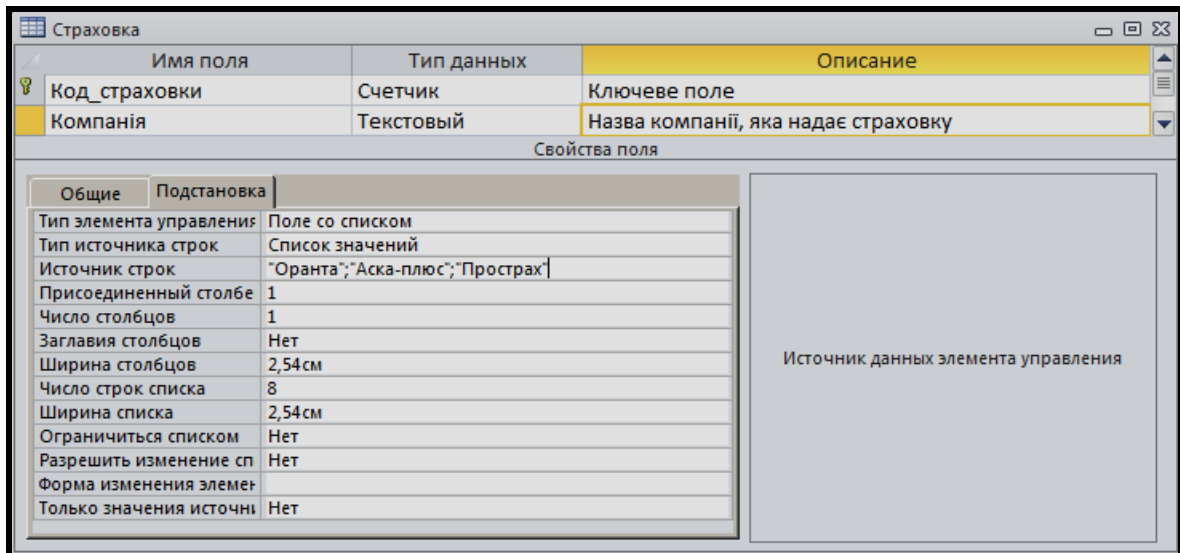


Рисунок 4.139 – Створення списку підстановки для обмеження введення можливих значень поля

ПРИМІТКА. Нагадаємо, що кількість записів у довідкових таблицях повинна бути не менше ніж 7–10, а в таблицях зі змінними даними – 30 записів. Дані за певний період підібрати самостійно. Вони повинні відображати господарську діяльність підприємства, щоб на їхній основі можна було у подальшому проаналізувати діяльність.

Крок 3. Проектування запитів для розв’язання інформаційних задач

Розробити та створити запити для розв’язання задач, що сформульовані в постановці задачі: вибірки за певними умовами, визначення підсумкових даних, оновлення існуючих та видалення застарілих даних, додавання записів в існуючу або створювану таблицю. Для досягнення поставленої мети необхідно ознайомитися з різними типами запитів.

На цьому кроці виконання ІНДЗ студенти повинні детально опанувати різні засоби створення запитів: за допомогою майстрів, у режимі конструктора; ознайомитися із вбудованими функціями MS Access та технологією використання будівника виразів.

Для кожної інформаційної задачі результати подати в такому вигляді:

- формулювання задачі;
- вид запиту (детальний, підсумковий тощо);

- базові джерела даних (таблиці, запити);
- перелік полів із зазначенням джерела (ім'я таблиці чи запиту), групування, сортування тощо (в разі потреби);
- обчислювані поля;
- зв'язки між джерелами даних із зазначенням полів, якщо їх встановлено під час побудови запиту;
- умови відбору (в разі потреби).

Нижче наведений перелік можливих запитів для БД «Туристична агенція». Запити на вибірку даних:

1. Знайти список прізвищ та даних по клієнтах, які придбали тури в агенції.
2. Знайти список потенційно можливих подорожей, тобто список маршрутів із зазначенням країни, курорту, кількості днів та вартості туру.
3. Знайти список укладених договорів, для яких необхідно оформлення візи (параметричний запит).
4. Обчислити вартість однієї подорожі (туру).

Нижче як приклад наведено перелік запитів та їх розв'язок за даними БД «Туристична агенція».

1. Визначити вартість того, яким менеджером за певний проміжок часу було укладено договори, у яку країну, на який курорт.

Розв'язок. Детальний запит на вибірку з параметром по даті укладання договору (рис. 4.140).

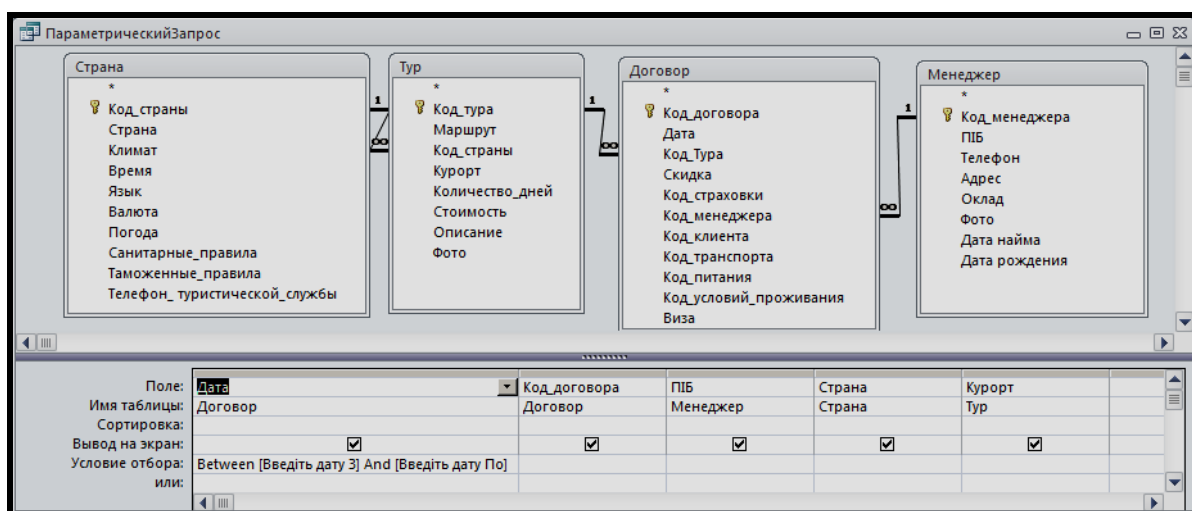
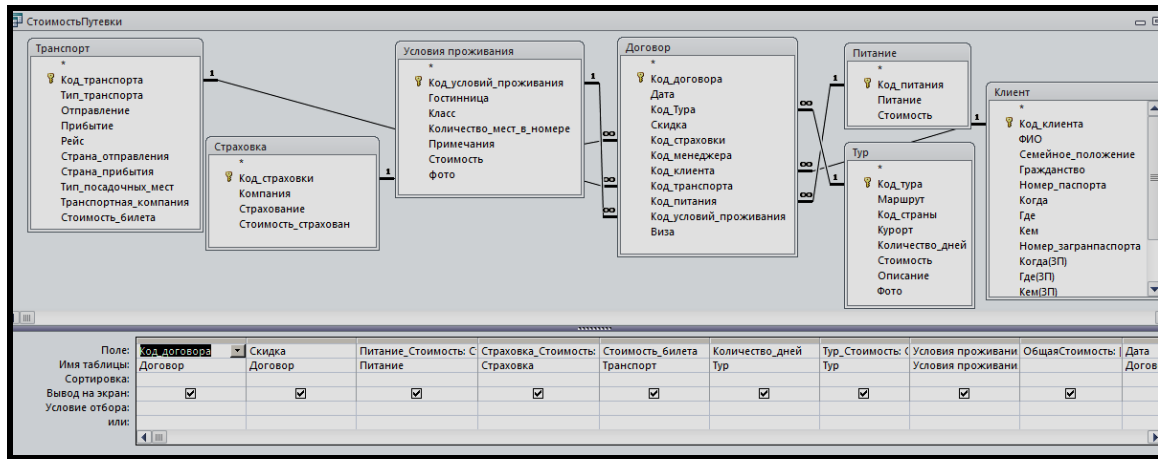


Рисунок 4.140 – Вигляд запиту з параметром по даті укладання договору

2. Визначити вартість укладеного договору (путівка) з переліком усіх складників вартості та зазначенням остаточної вартості путівки.

Розв'язок. Детальний запит на вибірку з обчислюваним полем «Загальна вартість» (рис. 4.141).



Поле:	ОбщаяСтоимость:	Дата	Код_клиента	ФИО	Код_менеджера
Имя таблицы:		Договор	Договор	Клиент	Договор
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					
или:					

Рисунок 4.141 – Видгляд детальних запитів на вибірку

Вигляд обчислювального поля наведено на рисунку 4.142.

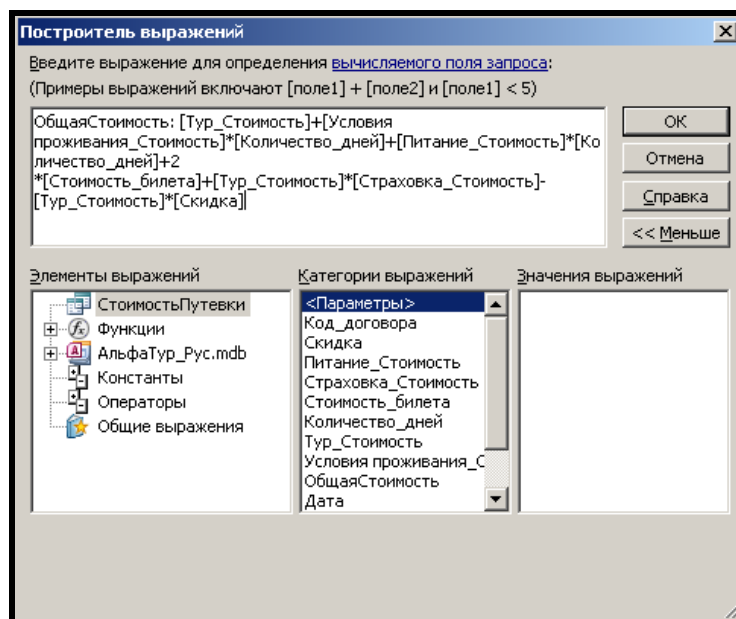


Рисунок 4.142 – Видгляд обчислювального поля у запиті

У такий спосіб навести всі створені вами запити у базі даних.

Крок 4. Проектування форм як основних елементів інтерфейсу користувача створюваного застосування (базі даних)

Відповідно до схеми даних розробити необхідні форми для ефективного ведення бази (введення, зміна та вилучення даних із таблиць). Створити форми, за допомогою яких розв'язуватимуться вказані задачі. На формах використати відповідні елементи керування, діаграми, а також зведені таблиці та зведеної діаграми для аналізу даних. Як джерела даних при цьому доцільно вживати запити, що побудовані на кроці 3.

На цьому етапі виконання студенти повинні ознайомитися з різними режимами роботи з формами – режимом конструктора, режимом форми або таблиці. З'ясувати види та специфічне призначення різних способів подання даних у режимі форми: в один стовпчик, стрічкова форма, зведена таблиця або зведена діаграма. Навчитися створювати складні форми: підпорядковані та зв'язані форми.

У процесі розробки форм студенти повинні детально ознайомитися з властивостями основних елементів керування форм: полів, списків, полів зі списками, кнопок.

Навчитися методам візуалізації даних за допомогою побудови діаграм, уміти налаштовувати властивості діаграм.

За результатами виконання кроку подати форми для виконання таких видів робіт:

- ведення бази даних (БД) (додавання, змінювання та вилучення даних);
- аналіз даних (зокрема й з використанням зведених таблиць і діаграм).

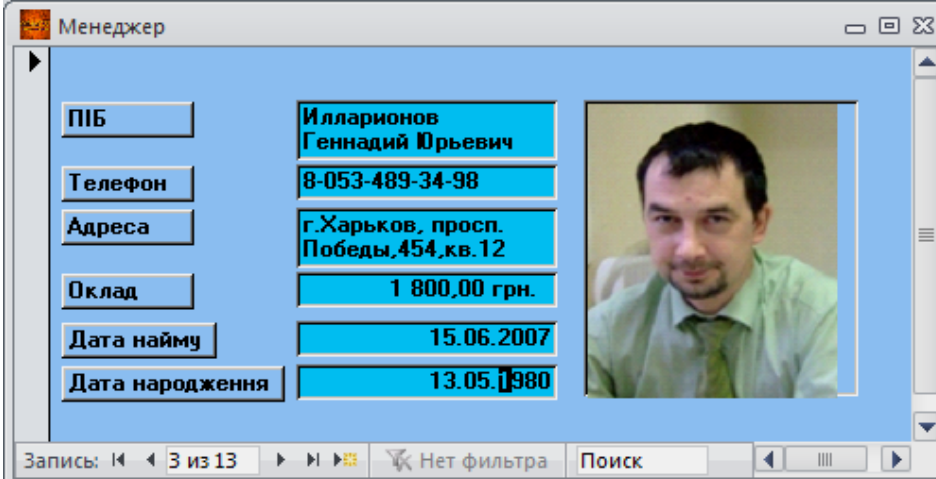
Для кожної форми подати такі дані:

- назва форми із зазначенням виду;
- призначення;
- рисунок форми в режимі конструктора (частина копії екрана);
- рисунок форми в режимі форми (частина копії екрана).


Зміст і дизайн кожної форми студенти повинні розробити та виконати самостійно.

Завдання. За допомогою форми показати дані про всіх співробітників туристичної агенції з можливістю внесення змін або додавання даних про нових співробітників.

Розв'язання. Форма створена на базі таблиці *Менеджер* за допомогою **Мастера форм**. Зовнішній вигляд форми – **в один стовбец**. Доопрацьована форма у режимі конструктора – змінені назви полів та їхнє розташування. Також змінено дизайн форми (рис. 4.143).



The screenshot shows a web form window titled "Менеджер". The form contains the following fields and values:

ПІБ	Илларионов Геннадий Юрьевич	
Телефон	8-053-489-34-98	
Адреса	г.Харьков, просп. Победы, 454, кв. 12	
Оклад	1 800,00 грн.	
Дата найму	15.06.2007	
Дата народження	13.05.1980	

At the bottom of the window, there is a status bar with the text "Запись: 3 из 13", "Нет фильтра", and "Поиск".

Рисунок 4.143 – Завершений вигляд форми «Менеджер»

У такий спосіб навести структуру всіх форм для заповнення таблиць-довідників, для перегляду, зміни та аналізу даних вашої бази даних (обов'язково створити форму з графіком і лінією тренду для прогнозування даних).

Загальні вимоги до розроблюваної БД

1. Розроблювана БД повинна включати такі об'єкти: таблиці, запити, форми.

2. Схема бази даних, необхідна кількість таблиць, їхня структура та властивості полів визначаються студентом самостійно, відповідно до постановки задачі згідно з обраною темою. **Кількість записів в довідкових таблицях повинна бути не менше 7–10, а в таблицях зі змінними даними – 30 записів.**

3. Відбір даних для аналізу реалізується в запитах. Необхідна кількість запитів визначається студентом самостійно згідно з постановкою задачі, але в розроблюваній БД повинні бути представлені всі види запитів: запити на вибірку даних із різними умовами відбору даних; запити з

обчислюваними полями; підсумкові запити; перехресні запити.

4. Інтерфейс користувача реалізується за допомогою форм, кількість і зовнішній вигляд яких визначається студентом самостійно. Але в розроблюваній БД повинні бути створені такі форми: форми для редагування довідників; форма для редагування таблиць зі змінною інформацією; форми з аналітичними даними, реалізовані у вигляді підпорядкованих або зв'язаних форм, які вміщують різні види діаграм для аналізу процесів.

У кінці модуля подається звіт із виконання всіх кроків.

Захист розроблених застосувань організується у вигляді виступу студентів перед групою з демонстрацією роботи застосування у середовищі MS Access, а також за допомогою презентації у PowerPoint. Студент повинен сформулювати, для розв'язування яких задач було розроблено застосування, і продемонструвати знайдені для цього рішення, а також вміти відповідати на питання інших студентів групи.

ГЛОСАРІЙ

А

- Анотація** Різновид просторового об'єкта, що становить текстовий опис ділянки місцевості чи просторового об'єкта
- Архітектура ГІС** Система побудови взаємозв'язку елементів структури ГІС, що включає компоненти логічної, фізичної та програмної структур.

Б

- База геоданих** База даних або набір файлів, які використовуються для зберігання, створення запитів і управління просторовими даними. У базі геоданих зберігається геометрія об'єктів, просторова прив'язка, атрибути та інші характеристики. До єдиної бази геоданих можуть бути зібрані різні типи географічних наборів даних. Наприклад: різні класи просторових об'єктів, атрибутивні таблиці, набори растрових і мережесих даних, топологія та ін. Бази геоданих можуть зберігатися в таких об'єктно-реляційних системах керування базами даних, як IBM DB2, IBM Informix, Oracle, Microsoft Access, Microsoft SQL Server і PostgreSQL, а також у наборах файлів, таких як файлова база геоданих. Програмний продукт ESRI ArcGIS дозволяє працювати з базами геоданих і створювати їх в ArcCatalog.
- База даних** Сукупність взаємозалежних даних, організованих відповідно до схеми бази даних у такий спосіб, щоб з ними міг працювати користувач.
- Блок векторних даних** Сукупність векторних об'єктів, яка може включати в себе кілька шарів різних типів (наприклад, точкові та лінійні) або бути представлена одним шаром. У ArcGIS набір векторних даних різних типів (точки, лінії, полігони) об'єднані єдиною системою координат.
- Буферна зона** Область навколо будь-якого об'єкта карти, яка вимірюється в одиницях відстані або часу. Також під буферною зоною розуміють полігон, який оточує точку, лінію або інший полігональний об'єкт на заданій фіксованій відстані. Створення буферних зон є одним із методів просторового аналізу, які застосовуються при оцінці близькості об'єктів.

В

Вид	Область графічного інтерфейсу програми, призначена для управління візуалізацією геоданих. Як правило, відмінна від вікна компоновання (макета), де здійснюється компоновання остаточного вигляду карти, що включає також зарамкове й додаткове оформлення та вміст.
Вираз	Обчислює значення об'єктів бази даних. Може містити математичні й логічні операції, викликати функції, вбудовані користувачем.
Відображення	Набори растрових даних можуть бути відображені, або візуалізовані, на карті декількома способами. Візуалізація – це процес генерації даних на екрані. Як саме відбувається візуалізація набору растрових даних, залежить від типу використовуваних даних. Деякі растри мають задану колірну схему, яку ArcMap автоматично використовує для відображення. Якщо колірної схеми немає, ArcMap сам вибере відповідну, потім, якщо буде потреба, можна її налаштувати.
Вікно бази даних	Містить списки всіх об'єктів бази даних і забезпечує до них доступ.
Вікно властивостей	Призначене для перегляду й зміни властивостей таблиць, запитів, форм, звітів до даних й елементів керування.
Вікно перегляду	У вікні перегляду в ArcCatalog відображається інформація про елементи, які виділяються в дереві каталогу. Є кілька опцій попереднього перегляду. Можна переглядати географію, атрибутивні таблиці й описи елементів.
Вільний елемент керування	Елемент керування у формі, звіті або на сторінці доступу до даних, не пов'язаний з даними.
Властивість	Характеристика об'єкта, що може приймати певне значення.
Векторизація	Конвертація осередків растра у векторні дані (набори крапок, ліній або полігонів).
Векторні дані	Вміст векторного об'єкта, координатна (геометрія) і атрибутивна інформація.

Г

Географічна координатна система	Двомірна координатна система, в якій положення точки на еліпсоїді (або сфері) визначається двома географічними координатами (широтою та довготою), вимірюваними в градусах. Для того щоб задати географічну систему
--	---

координат, необхідно вказати параметри еліпсоїда, нульовий меридіан і одиниці вимірювання кутів (градуси, радіани).

Геокодування

Процес, широко застосовуваний в геоінформатиці, що полягає в призначенні координат об'єктам на карті. У тому випадку, якщо координати присвоюються, виходячи з положення цих об'єктів на місцевості (наприклад, призначенні координат X і Y об'єкту за його поштовою адресою) – цей процес називається прямим геокодуванням. При зворотньому геокодуванні все відбувається навпаки: формальна адреса (номер будинку, назва вулиці, міста) призначається об'єкту, чие місце розташування вже визначено координатами.

Головна форма

Форма, що включає іншу форму.

Групування

Поділ даних на групи за певним критерієм.

Д

Дані

Довільна інформація, представлена в цифровій формі.

Джерело даних

Таблиця або запит, на основі даних яких заповнюються форма, звіт, елемент керування або інший об'єкт.

Джерело записів

Таблиця, запит або інструкція SQL, що забезпечує дані для форми або звіту.

Діаграма

Графічний образ, що відображає кількісну залежність даних.

Додаток у БД

Програма, що слугує для зручної роботи з об'єктами БД.

Документ карти

У ArcMap: файл, який містить карту, її макет, а також пов'язані з нею шари, таблиці, діаграми та звіти. Документ карти можна роздрукувати або вставити в інші документи. Розширення файлів документів карти *.mxd.

Е

Експорт даних

Копіювання об'єкта із БД.

З

Записи БД

Рядки таблиці.

Запити на видалення

Призначені для видалення з таблиці записів за заданою умовою.

Запити на відновлення

Призначені для зміни значення полів таблиці для відібраних записів.

Запити на додавання

Призначені для додавання кількох записів із запиту в таблицю.

Запити на створення таблиці	Призначені для створення нової таблиці, в яку можуть додаватись записи із таблиць та запитів.
Звіт	Об'єкт, призначений для виводу обраної із БД інформації у вигляді документа на принтер або монітор.
Зовнішній ключ	Поле, що слугує для зв'язку з іншою таблицею.
Закладка	В інтернет-службі ESRI: закладка, в якій для подальшого використання збережений поточний вид карти, зокрема охоплення, активні шари, стилі їх оформлення та інше.
Запит	Вибірка за різними параметрами (властивостями) обраних об'єктів або записів у базі даних або атрибутивній таблиці шару. Запит задається у формі твердження (рівності, нерівності та інше) або логічного виразу.

І

Ідентифікатор (просторового об'єкта)	Унікальна характеристика просторового об'єкта, що привласнюється йому користувачем або призначається інформаційною системою, яка використовується для фіксації зв'язку координатних і адресних даних просторових об'єктів.
Ідентифікувати	Отримати інформацію про атрибути об'єкта. У ArcGIS для цього використовується інструмент Identify.
Імена полів	Назви колонок таблиці.
Імпорт даних	Копіювання об'єкта в БД.

К

Калібрування знімка	Технологічний процес, що полягає у визначенні елементів внутрішнього орієнтування та геометричних перетворень знімка. Існує кілька способів калібрування: зоряне калібрування, польове калібрування, самокалібрування.
Канали	Растри можуть мати один або кілька каналів. Багатоканальні растри зазвичай називаються мультиспектральними зображеннями, а растри, що містять до декількох сотень каналів – гіперспектральними. Одноканальний набір растрових даних становить одне явище, наприклад висоту або довжину хвилі тільки одного діапазону в електромагнітному спектрі (чорно-білий знімок). Канали часто пов'язані зі спектральним розподілом.

Каскадне видалення	Засіб підтримки цілісності даних у пов'язаних таблицях, який при видаленні запису в головній (батьківській) таблиці забезпечує видалення всіх пов'язаних записів у залежній (дочірній) таблиці.
Каскадне відновлення	Засіб підтримки цілісності даних у пов'язаних таблицях, при якому в разі зміни значення ключового поля в головній (батьківській) таблиці забезпечується відновлення всіх пов'язаних записів у залежній (дочірній) таблиці.
Каталоги растрів	Каталогом растрів є колекція наборів растрових даних, подана у форматі таблиці, в якій кожен рядок вказує на окремий набір растрових даних, включений в каталог. Каталог растрів може містити тисячі окремих зображень. Каталог растрів зазвичай використовується для відображення сусідніх, наборів растрових даних, які повністю або частково накладаються, без побудови одного великого мозаїчного набору.
Кнопки переходу	Кнопки, які розміщені на нижній границі вікон у режимі таблиць і форм та використовують для переміщення по записах.
Кнопкова форма	Форма, яка забезпечує користувачу доступ до функцій додатка.
М	
Макрос	Це об'єкт, призначений для автоматизації дій, які виконуються за допомогою команд меню або кнопок панелей інструментів.
Метадані	Термін, який використовується геопросторовими організаціями для опису ГІС-елементів за стандартною схемою. Це дозволяє користувачам ArcGIS документувати свої елементи відповідно до стандартів метаданих, які підтримуються величезним співтовариством користувачів.
Місцезнаходження	Ідентифікатор географічної області, заданий координатами.
Модуль	Об'єкт, що становить програму, написану мовою VBA, і використовується під час створення складних додатків.

Н

Набір даних мозаїки	Набір даних мозаїки – це сукупність наборів растрових даних (зображень), що зберігається як каталог і відображається як єдине мозаїчне зображення або окремі зображення (растри). Набори даних мозаїки можуть бути величезними і за розміром файлу, і за кількістю наборів растрових даних. Набори растрових даних у наборі даних мозаїки можуть залишатися в початковому форматі на диску або, за необхідності, бути завантажені в базу геоданих. Метаданими можна керувати при записі растра так само, як і атрибутами в таблиці атрибутів. Зберігання метаданих як атрибутів дозволяє виконувати швидкі запити для виконання вибірок.
Набір растрових даних	Набором растрових даних називається будь-який підтримуваний растровий формат, організований в один або кілька каналів. Кожен канал складається з масиву пікселів (осередків); кожен піксель має якесь значення. Набір растрових даних має як мінімум один канал. Кілька наборів растрових даних можуть бути просторово з'єднані разом, у єдиний, великий, безперервний набір.

О

Обчислювальне поле	Поле в запиті або формі, як значення якого виводиться результат обчислення виразу, а не дані, збережені в таблиці.
Одиниця виміру відстаней	Одиницями виміру відстаней є, наприклад, кілометр, метр, фут, миля та інше.
Операція	Дія, виконувана над даними таблиць бази даних.

П

Панель елементів	Панель інструментів, що містить кнопки, за допомогою яких створюються елементи керування у формах та звітах.
Первинний ключ	Поле або сукупність полів, які однозначно визначають запис таблиці.
Перетворення даних	Перетворення (трансформація) даних – це процес реорганізації або реструктуризації даних з формату джерела у формат цільових даних.
Підлегла форма (звіт)	Підлегла форма (звіт), що вбудовується в головну форму (звіт).
Піксель	Пікселі часто використовуються як синоніми клітинки.

Клітинки й піксель є найменшими одиницями інформації растрових даних. Піксель – це аббревіатура від словосполучення picture element (елемент зображення) і часто використовується для опису зображення, тоді як клітинка часто використовується для опису растрових даних. Пікселі мають вимірювання та значення. Вони представляють таку інформацію, як температура, тип ґрунту, висота, і такі об'єкти реального світу, як парки, озера й будівлі.

Повідомлення помилку	про	Властивість поля, що визначає текст повідомлення, коли порушується умова на значення.
Поле зі списком		Комбінований елемент керування, що поєднує в собі текстове поле для введення й відображення даних і список для вибору значення.
Поле		Дані колонок таблиці.
Поле підстановок		Список можливих значень поля в таблиці, що використовується для спрощення введення даних.
Пряме зчитування		Пряме зчитування – це здатність швидко розпізнавати нерідний формат в ArcGIS, не використовуючи інструменти конвертації.

Р

Растр		Це модель даних, що описує збережене зображення. Растр визначає пікселі (осередки) в рядках і стовпцях, кількість каналів і бітову глибину, які складають зображення. При перегляді растра, ви переглядаєте зображення растрових даних. Також зустрічається інше визначення растра, засноване на осередках наборів даних. Воно зазвичай не використовується в документації ArcGIS.
Реляційна БД		Дані, подані у вигляді взаємозалежних таблиць.
Розподіл		Просторову роздільну здатність встановлює зв'язок між масштабом і розміром осередку. Просторова роздільна здатність (також відоме як розмір клітинки) – це вимір області на поверхні землі, представленої одним осередком. Просторова роздільна здатність або розмір клітинки впливає на рівень деталей, представлений зображенням у будь-якому масштабі.

С

Система керування базами даних (СКБД)	Сукупність програмних та мовних засобів, які забезпечують керування БД.
Складений ключ	Первинний ключ таблиці, що складається з декількох полів.
Сортування	Зміна порядку, в якому подані дані.
Стрічкова форма	Вид форми, в якій виводиться одночасно кілька записів із базової таблиці або запиту у формі рядка.

Т

Таблиця	Це об'єкт, що слугує для визначення й зберігання даних одного виду.
Транслятор даних	Транслятор даних конвертує дані з одного формату в інший. Наприклад, транслятор даних може конвертувати дані MapInfo в шейп-файл Esri.

У

Умова на значення	Властивість поля, що визначає обмеження, які повинні задовольняти правильні значення поля.
--------------------------	--

Ф

Форма	Об'єкт, що створює зручний інтерфейс із даними (однієї або декількома таблицями). Форма використовується для введення даних, відображення їх на екрані й керування роботою додатка.
Формат даних	Формат даних визначає кодування цифрової інформації та її збереження у файлі або іншому джерелі даних.
Формат і тип растру	Формат растра визначає пікселі, що зберігаються, кількість рядків, стовпців і кількість каналів, фактичні значення пікселів та інші характерні параметри. Тип растра, наприклад, допомагає ідентифікувати метадані, просторову прив'язку, дату придбання й тип сенсора, поряд із форматом растра.

Ц

Цілісність даних	Система правил, що використовується для підтримки зв'язків між записами у пов'язаних таблицях.
Центроїд	Це геометричний центр об'єкта. Для лінії – середина лінії, для полігону – центр полігону.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Основні

1. Поморцева О. Є. Бази даних: проектування для обліку нерухомого майна: лабораторний практикум : навч. посіб. : для студентів напряму підгот. 6.080101 «Геодезія, картографія та землеустрій» / О. Є. Поморцева ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2014. – 192 с.

2. Толстохатко В. А. Бази даних: проектування та використання для обліку нерухомого майна : навч. посіб. / В. А. Толстохатко, О. Є. Поморцева, І. М. Патракеєв ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова – Харків : ХНУМГ ім. О. М. Бекетова, 2014. – 173 с.

Додаткові

3. Kobzan S. Real Estate Market of Ukraine: Practical Aspects and Trends : monograph / S. Kobzan, O. Pomortseva ; O. M. Beketov National University of Urban Economy in Kharkiv. – Switzerland : Springer Nature, 2023. – 146 p. – (SpringerBriefs in Geography). – DOI: https://doi.org/10.1007/978-3-031-31248-9_1.

Електронні ресурси

4. Сайт додатків Office корпорації Microsoft [Електрон. ресурс] / Microsoft : сайт. – Електрон. дані. – Оновлюється постійно. – Режим доступу: <http://office.microsoft.com/ru-ru/access-help>, вільний (дата звернення: 17.04.2024). – Назва з титул. екрана.

5. Сайт корпорації «ArcGis Resource Center» по роботі з програмним продуктом [Електрон. ресурс] / Esri : сайт. – Електрон. дані. – Оновлюється постійно. – Режим доступу: <https://www.esri.com/en-us/arcgis/products/arcgis-desktop/resources>, вільний (дата звернення: 17.04.2024). – Назва з титул. екрана.

Навчальне видання

ПОМОРЦЕВА Олена Євгенівна

**ОСНОВИ ГЕОІНФОРМАЦІЙНИХ СИСТЕМ
І БАЗИ ДАНИХ**

ПІДРУЧНИК

Відповідальний за випуск *О. Є. Поморцева*

Редактор *В. І. Шалда*

Комп'ютерне верстання *О. Є. Поморцева*

Дизайн обкладинки: *К. П. Растріполко*

Підп. до друку 18.10.2021. Формат 60 × 84/16.
Тираж 60 пр. Зам. № 22-02528. Ум. друк. арк. 20,1.

Видавець:

Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Маршала Бажанова, 17, Харків, 61002.

Електронна адреса: office@kname.edu.ua

Свідоцтво суб'єкта видавничої справи:

ДК № 5328 от 11.04.2017.

Надруковано в типографії:

«Impress» (ФОП Болібок О.В.)

Україна, м. Харків, вул. Пушкінська, 56.

тел.: (057) 714-42-11.