

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

Є. Г. Пономаренко, О. С. Ломакіна

ЗАСОБИ ТА ІНСТРУМЕНТИ
ІНЖЕНЕРНО-ЕКОЛОГІЧНИХ РОЗРАХУНКІВ

КОНСПЕКТ ЛЕКЦІЙ

*(для студентів 1 та 2 курсу усіх форм навчання
освітньо-кваліфікаційного рівня «бакалавр»
спеціальності 183 – Технології захисту навколишнього середовища)*

Харків
ХНУМГ ім. О. М. Бекетова
2018

Пономаренко Є. Г. Засоби та інструменти інженерно-екологічних розрахунків. Конспект лекцій (для студентів 1 та 2 курсу усіх форм навчання освітньо-кваліфікаційного рівня «бакалавр» спеціальності 183 – Технології захисту навколишнього середовища) / Є. Г. Пономаренко, О. С. Ломакіна; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2018. – 23 с.

Автори:

канд. техн. наук. Є. Г. Пономаренко,
ст. викладач О. С. Ломакіна

Рецензент

Ф. В. Стольберг, доктор технічних наук, професор, завідувач кафедри інженерної екології міст

Рекомендовано кафедрою інженерної екології міст, протокол № 1 від 29. 08. 2017.

Конспект лекцій складено з метою допомогти студентам спеціальності 183 – Технології захисту навколишнього середовища під час підготовки до занять та контролю з дисципліни «Засоби та інструменти інженерно-екологічних розрахунків».

© Є. Г. Пономаренко, О. С. Ломакіна, 2018
© ХНУМГ ім. О. М. Бекетова, 2018

ЗМІСТ

Вступ.....	4
1 Методи та інструменти обробки масивів даних та виконання прогнозних розрахунків.....	5
1.1 Застосування програмних засобів для вирішення розрахункових завдань у моделюванні та прогнозуванні стану довкілля	5
1.1.1 Програмні засоби для аналітичної обробки інформації.....	5
1.1.2 Використання програмних заходів для вирішення складних розрахункових завдань.....	6
1.2 Методи обробки натурних і експериментальних даних.....	11
2 Інформаційно-пошукові системи і бази даних.....	14
2.1 Бази даних	14
2.2 Побудова інформаційно-пошукової системи	17
2.2.1 Принципи побудови, реалізації та використанні ІПС	17
2.2.2 Сутність алгоритму та пошуку інформації	19
2.2.3 Програмна реалізація ІПС	22
Список рекомендованих джерел	23

ВСТУП

Сучасний етап розвитку промисловості та науки потребує підготовки висококваліфікованих фахівців з питань захисту довкілля та оцінки екологічної ефективності технологій захисту навколишнього середовища.

В теперішній час перед фахівцями-екологами все частіше постають завдання з дослідження впливу технологій на довкілля та оцінки ступеню цього впливу, а також питання моделювання та прогнозування можливих змін у навколишньому середовищі. Вирішення цих завдань вимагає застосування складних інженерно-екологічних розрахунків або обробки значних масивів натурних та експериментальних даних.

Сьогодні також характеризується загальною комп'ютеризацією та розробкою значної кількості програмних продуктів, що можуть стати у нагоді фахівцям з захисту довкілля у вирішенні вищезазначених завдань.

Дисципліна «Засоби та інструменти інженерно-екологічних розрахунків» введена до курсу підготовки бакалаврів з технологій захисту навколишнього середовища з метою формування у студентів теоретичних та практичних знань та навичок застосування основних законів, принципів та методів розробки алгоритмів для вирішення задач різного типу (алгебраїчних, диференціальних, інтегральних рівнянь) та задач з інженерної практики в екологічних дослідженнях та їх комп'ютерна реалізація, придбання навичок застосування спеціалізованих програмних середовищ для математичних розрахунків.

1 МЕТОДИ ТА ІНСТРУМЕНТИ ОБРОБКИ МАСИВІВ ДАНИХ ТА ВИКОНАННЯ ПРОГНОЗНИХ РОЗРАХУНКІВ

1.1 Застосування програмних засобів для вирішення розрахункових завдань у моделюванні та прогнозуванні стану довкілля

1.1.1 Програмні засоби для аналітичної обробки інформації

Найвідомішими програмними засобами, що можуть бути рекомендовані для аналітичної обробки екологічної інформації та вирішення складних розрахункових завдань є Matlab, Mathematica та MathCad.

Matlab – одна з найстарших, ретельно пророблених і перевічених часом систем автоматизації математичних розрахунків, побудована на представленні і застосуванні матричних операцій. Незважаючи на те, що Matlab спочатку призначалась винятково для обчислень, у подальшому було придбане ядро символічних перетворень Maple, а також з'явилися бібліотеки, які забезпечують в Matlab унікальні для математичних пакетів функції. Наприклад, широко відома бібліотека «Simulink», яка реалізує принцип візуального програмування і дозволяє побудувати логічну схему складної системи управління з одних тільки стандартних блоків, не написавши при цьому ні строчки програмного коду. Після конструювання такої схеми можна детально проаналізувати її роботу.

Пакет Mathematica широко застосовується при виконанні розрахунків у сучасних наукових дослідженнях і одержав велику популярність у науковому і освітньому середовищі. Система Mathematica є однією з найбільших програмних систем і реалізує найбільш ефективні алгоритми обчислень. Система Mathematica сьогодні розглядається як світовий лідер серед комп'ютерних систем символічної математики для ПК, що забезпечують не тільки виконання складних чисельних розрахунків з виводом їх результатів у

графічному виді, а й проведення особливо трудомістких аналітичних перетворень і обчислень.

Програмний комплекс MathCad сьогодні є майже міжнародним стандартом для технічних обчислень. Поряд з наймогутнішими засобами розрахунків, має вбудований (на движку Maple) блок символічних обчислень.

Однак слід зазначити, що поширений програмний продукт «Excel» також має досить потужні обчислювальні ресурси, що дозволяють робити обчислення інженерно-екологічного характеру. Це можливо завдяки, по-перше, наявності великої кількості вбудованих функцій; по-друге, завдяки наявності спеціальних сервісних засобів і, по-третє, внаслідок можливості створювати й використовувати свої власні функції. Слід зазначити, що використання останньої можливості вимагає від користувача знання мови програмування Basic.

1.1.2 Використання програмних заходів для вирішення складних розрахункових завдань

Програмний комплекс MathCad може бути використаний для виконання наступних видів складних розрахункових екологічних завдань, що можуть бути вирішені:

- шляхом знаходження коренів рівнянь довільного виду;
- шляхом розв'язання системи лінійних рівнянь;
- шляхом розв'язання системи нелінійних рівнянь;
- шляхом розв'язання системи диференціальних рівнянь.

Розглянемо детальніше кожен процедуру вирішення розрахункових екологічних завдань за допомогою MathCad.

Процедура розв'язання розрахункових завдань, що вирішуються шляхом знаходження коренів рівнянь довільного виду передбачає наступні етапи:

- 1) визначення кількості коренів рівняння;
- 2) визначення початкових наближень для коренів та діапазону розташування коренів;
- 3) задавання необхідної точності пошуку рішення;
- 4) знаходження рішень (коренів) рівняння за допомогою функції *root* або *Find*.

Для вирішення одного рівняння з однією невідомою використовується функція *root(F(x),x)*, що повертає значення x , при якому вираз або функція $F(x)$ обертається в нуль. Обидва аргументи цієї функції повинні бути скалярами. Відповідно, функція повертає скаляр. При використанні функції *root* обов'язково необхідно впевнитися, що змінній присвоєно початкове значення до початку використання функції *root*. У разі вирішення рівняння, що має декілька коренів, наприклад квадратного рівняння, рекомендовано побудувати графік функції для визначення кількості коренів та початкових наближень для кожного кореню.

Для знаходження коренів рівняння, що має вигляд $a_0 + a_1x^1 + a_2x^2 + \dots + a_nx^n$, кращим є використання функції *polyroots(a)*. Використання цієї функції не потребує початкових наближень і функція відразу повертає всі корені рівняння. Перед використанням функції необхідно сформувати вектор коефіцієнтів полінома, при чому коефіцієнти поліному повинні бути розташовані у порядку зростання ступеню x у векторі a довжини $n + 1$.

Вирішення рівняння також можливе з використанням панелі символічних обчислень (функція *solve*). Послідовність дій при використанні цієї функції наступна:

- введення рівняння з використанням знаку логічної рівності;
- обрання за допомогою комп'ютерної миші змінної, відносно якої вирішується рівняння;
- обрання на панелі *Символьна* пункту *Solve*.

При використанні символьних обчислень немає необхідності задавати початкові наближення.

Процедура розв'язання розрахункових завдань, що вирішуються шляхом розв'язання системи лінійних рівнянь передбачає наступні етапи:

- 1) представлення системи рівнянь у вигляді, при якому вільні члени рівняння перебувають у правій частині;
- 2) завдання матриці коефіцієнтів рівняння і вектору вільних членів;
- 3) знаходження вирішення системи за допомогою зворотної матриці або функції *lsolve*;
- 4) оцінка точності отриманого рішення.

Матриця коефіцієнтів повинна бути квадратною і її порядок (розмірність) повинен відповідати кількості невідомих, що дорівнюють кількості рівнянь системи.

Вектор правої частини – це вектор-стовпчик, розмірність якого дорівнює кількості невідомих.

Для одержання результату можна використовувати один із двох способів:

- 1) розв'язання матричного рівняння $AX=b$ у вигляді:

$$X=A^{-1}b,$$

де A – назва матриці коефіцієнтів;

b – назва вектору правої частини.

- 2) використання функції *lsolve* (A, b). Функція повертає рішення у вигляді вектора.

Процедура розв'язання розрахункових завдань, що вирішуються шляхом розв'язання системи нелінійних рівнянь передбачає наступні етапи:

– завдання початкових наближень для всіх невідомих рівнянь, що входять у систему;

– введення ключового слова *Given*. Воно вказує Mathcad, що далі буде наведено систему рівнянь. При цьому слід переконатися, що курсор не перебуває у текстовій області;

– введення рівняння нерівності в будь-якому порядку нижче ключового слова *Given*. При цьому слід переконатися, що між лівими і правими частинами рівнянь встановлено символ логічної рівності =, надрукований комбінацією клавіш [Ctrl+=] або обраний з відповідної палітри;

– введення виразу, що містить *Find* (z_1, z_2, z_3, \dots). Ця функція повертає розв'язання системи рівнянь. Число аргументів повинне дорівнювати числу невідомих.

Для аналізу правильності вирішення рівняння рекомендується використовувати контурний графік.

*Процедура розв'язання розрахункових завдань, що вирішуються шляхом розв'язання диференціальних рівнянь з використанням функції *odesolve* передбачає наступні етапи:*

– введення ключового слова *Given*;

– визначення нижче виду диференціального рівняння, дотримуючись наступних правил:

1) диференціальне рівняння повинне бути строго лінійним;

2) похідні в диференціальній формі можуть бути задані або за допомогою спеціальних операторів панелі *Обчислення*, або за допомогою клавіш *Ctrl + F7*;

3) шукані функції повинні бути визначені чітко з вказанням імені змінної, наприклад, $f(x)$;

4) в якості знаку рівняння використовується тільки логічне рівняння;

5) всередині блоку, що обчислюється, необхідно задати початкові або граничні умови.

– визначити рішення за допомогою спеціальної функції *odesolve*, яка має вигляд:

$$\text{odesolve}(t,b,[\text{step}]),$$

де t – змінна, від якої залежить функція;

b – права межа інтервалу пошуку рішення;

step – параметр, що визначає кількість кроків, що використовуються чисельним методом.

Складні розрахункові завдання, що передбачають розв'язання систем диференціальних рівнянь можуть бути вирішені двома методами:

– використання розрахункового блоку *Given-Odesolve*;

– використання вбудованої функції *rkfixed*.

Розрахунковий блок *Given-Odesolve* був розглянутий вище. При цьому спеціальна функція має вигляд

$$\text{odesolve}(\text{vector},t,b,[\text{step}]),$$

де *vector* – вектор функцій, відносно яких вирішується система диференціальних рівнянь. Цей вектор повинен містити без імені змінної.

Чисельне вирішення диференціальних рівнянь методом Рунге – Кута виконується за допомогою функції *rkfixed*. Функція *rkfixed* має наступний вигляд:

$$\text{rkfixed}(y, x1, x2, \text{npoints}, D),$$

де y – вектор-стовпчик початкових значень;

$x1$ – ліва межа діапазону змін незалежної змінної;

$x2$ – права межа діапазону змін незалежної змінної;

npoints – кількість розрахункових точок;

$D(x, Y)$ – вектор-функція, що визначає праву частину диференціальних рівнянь;

x – незалежна змінна;

Y – вектор, компоненти якого є співпоставленими з іменами шуканих функцій;

1.2 Методи обробки натурних і експериментальних даних

Основними статистичними характеристиками, що найчастіше визначаються при обробці натурних та експериментальних даних, є статистичне середнє значення, максимальне та мінімальне значення вибірки, значення вибірки, що зустрічається найчастіше, вибіркова дисперсія та середньоквадратичне відхилення, тощо.

В MathCAD є низка вбудованих функцій для розрахунку числових статистичних характеристик випадкових даних:

- $mean(x)$ – статистичне середнє значення;
- $max(x)$, $min(x)$ – максимальне та мінімальне значення вибірки;
- $mode(x)$ – значення вибірки, що зустрічається найчастіше;
- $Var(x)$ – вибіркова дисперсія;
- $Stdev(x)$ – середньоквадратичне відхилення;
- x – вектор (або матриця) з вибіркою випадкових даних.

Вихідні дані для аналізу за допомогою *MathCAD* зазвичай задаються або у вигляді матриць (або векторів). При цьому дані можливо вводити у матрицю (або вектор) безпосередньо (вручну), або імпортувати їх з файлу, якщо вони вже збережені в одному з доступних для імпорту форматі.

Вибірка експериментальних (натурних) найчастіше представляється у вигляді масиву (матриці), що складається з пар чисел $(x_i; y_i)$. Тому виникає завдання *апроксимації дискретної залежності $y(x_i)$ неперервною функцією $f(x)$* .

Функція $f(x)$ в залежності від специфіки завдання, може відповідати різним вимогам:

– функція $f(x)$ повинна проходити через точки $(x_i; y_i)$. У цьому випадку говорять про інтерполяцію даних функцією $f(x)$;

– функція $f(x)$ повинна деяким чином (наприклад, у вигляді певної аналітичної залежності) наближати $y(x_i)$, не обов'язково проходячи через точки (x_i, y_i) . Такою є постановка задачі побудови регресійної моделі, яку в багатьох випадках також можна назвати згладжуванням даних.

За своїм визначенням, *інтерполяція* означає побудову функції $Y(x)$, у якій значення у точках x повинні співпадати з вихідними даними, тобто $Y(x_i) = y_i$.

Для побудови інтерполяції за допомогою MathCAD можна використовувати вбудовані функції, що дозволяють з'єднати точки вибірки даних (x_i, y_i) кривою різного ступеню гладкості.

Найпростіший вид інтерполяції – лінійна, що відображає шукану залежність $Y(x)$ у вигляді ломаної лінії. Інтерполуюча функція $Y(x)$ складається з відрізків прямих, що поєднують точки (x_i, y_i) .

Для побудови лінійної інтерполяції слугує вбудована функція *linterp*, яка має наступний вигляд:

$$linterp(x, y, t),$$

де t — значення аргументу, при якому розраховується інтерполуюча функція.

При використанні функції *linterp* дані повинні бути попередньо відсортовані за зростанням x .

Завдання *математичної регресії* має на меті приближення вибірки даних (x_i, y_i) функцією $f(x)$, що певним чином мінімізує сукупність помилок $f(x_i) - y_i$. Регресія зводиться до підбору невідомих коефіцієнтів, що визначають аналітичну залежність $f(x)$. Як правило, регресія є дуже ефективною у випадку, коли заздалегідь відомий, або, принаймні, добре вгадується) закон розподілу даних (x_i, y_i) .

Найпростішим і найбільш часто використаним є такий вид регресії, як лінійна регресія. Наближення даних здійснюється лінійною функцією $y(x)=ax+b$.

Побудова рівняння лінійної регресії можливо двома способами:

– з використанням функції $line(x,y)$, що повертає вектор двох елементів (b,a) коефіцієнтів рівняння лінійної регресії;

– з використанням функцій $intercept(x,y)$ та $slope(x,y)$. Функція $intercept(x,y)$ повертає коефіцієнт b у рівнянні лінійної регресії. Функція $slope(x,y)$ відповідно повертає коефіцієнт a .

Для рівняння регресії виду $y(x) = a_1f_1(x) + a_2f_2(x) + \dots + a_nf_n(x)$, що відображає лінійну відносно невідомих коефіцієнтів a_i комбінацію заданих функцій $f_i(x)$, коефіцієнти можуть бути визначені за допомогою функції $linfit$, що має вигляд

$$linfit(A,B,F),$$

де A – назва вектору вхідних величин;

B – назва вектору вихідних величин;

F – вектор-стовпчик функцій $f_i(x)$.

Функція $linfit$ вертає вектор-стовпчик коефіцієнтів a_i .

2 ІНФОРМАЦІЙНО-ПОШУКОВІ СИСТЕМИ І БАЗИ ДАНИХ

2.1 Бази даних

База даних (БД) – це один або кілька файлів, закодованих спеціальним способом. Комп'ютерна програма, яка управляє базою даних, називається системою управління базою даних, або СУБД.

За свою суттю база даних – це деяка подоба електронної картотеки, електронного сховища даних, яке зберігається в комп'ютері у вигляді одного або декількох файлів. При цьому виникає необхідність у виконанні низки операцій із БД. У першу чергу це:

- додавання нової інформації в існуючі файли БД;
- зміна (модифікація) інформації в існуючих файлах БД;
- пошук інформації в БД;
- видалення інформації з існуючих файлів БД.

Концепція баз даних мінялася протягом тривалого часу. Найпоширенішою стала реляційна модель.

Реляційна база даних – це база даних, у якій усі дані, доступні користувачеві, організовано у вигляді взаємозалежних таблиць, а всі операції над даними зводяться до операцій над цими таблицями.

У кожній таблиці повинне бути унікальне ім'я.

Таблиці розділені на рядки і стовпці, на перетинанні яких містяться значення даних. Рядки в таблицях звичайно називають записами, а стовпці – полями.

Щоб відрізнити один рядок від іншого використовується поняття *ключа*. Ключем називається стовпець або група стовпців таблиці, значення яких унікальним образом ідентифікують кожний рядок таблиці. У таблиці може бути тільки один ключ. Усі записи в ключовому полі, як існуючі так і можливі в

майбутньому повинні бути різними. Не важливо якими – але різними. Саме по ключовім полю різняться записи в таблиці.

Індекс - цей засіб, що забезпечує швидкий доступ до рядків таблиці на основі значень одного або декількох стовпців. За замовчуванням, коли користувач переглядає дані, які зберігаються в таблиці, вони відображаються в тому порядку, у якому лежать у базі, тобто усі відомості з одного запису з'являються в одному рядку, але записи ніяк не відсортовані. Це означає, що для пошуку необхідного запису необхідно буде переглянути увесь список.

Для зручності використання реляційні бази даних надають засіб для швидкого пошуку потрібної інформації, а саме – індекси. СУБД користується індексом так само, як людина користується предметним покажчиком книги. В індексі зберігаються значення даних і покажчики на рядки, де ці дані зустрічаються. Значення даних в індексі розташовуються в порядку зростання або убутання, щоб СУБД могла швидко знайти необхідне значення.

По суті, індекси таблиць – це відсортовані списки. Кожен елемент індексу вказує на деякий запис у таблиці. Тому при пошуку запису система управління базами даних спочатку переглядає індекси, що займає зовсім небагато часу. Індекс можна побудувати по полю майже будь-якого типу. Інденси значно пришвидшують пошук даних.

Структура реляційної БД задається *схемою даних*. Схема даних містить у собі визначення структури кожної таблиці і визначення зв'язків між таблицями.

Для роботи із БД використовуються спеціальні мови, у цілому називані мовами баз даних. У реляційних СУБД підтримується єдина мова, що містить усі необхідні засоби - від створення БД до забезпечення користувацького інтерфейсу при роботі з даними - мова SQL (Structured Query Language).

Нормалізація баз даних – це формалізована процедура, у процесі виконання якої атрибути даних (поля) групуються в таблиці, а таблиці, у свою чергу, у бази даних.

Цілі нормалізації наступні:

- виключити дублювання інформації в таблицях.
- забезпечити можливість змін у структурі таблиць.
- зменшити вплив структурних змін бази даних на роботу додатків, що забезпечують користувачам доступ до даних.

Процес нормалізації складається з декількох етапів.

На першому етапі необхідно позбутися повторювання полів і розділити складові поля на окремі елементи даних. Осередки таблиці не можуть містити більше одного значення.

На другому етапі забезпечується повна залежність стовпців, які не є ключовими, від первинного ключа, а якщо цей ключ є складовим, то від кожного його елемента. Під повною залежністю розуміється можливість однозначного визначення значення кожного неключового поля за допомогою значення первинного ключа.

На третьому етапі забезпечують незалежність будь-якого неключового стовпця від усіх інших стовпців (крім первинного ключа таблиці). Повинна бути можливість змінювати значення кожного поля (яке не входить в первинний ключ) без впливу на дані інших полів.

Описати всі типи можливих об'єктів баз даних складно, оскільки СУБД може реалізувати свої типи об'єктів.

До основних типів об'єктів відносять:

– таблиці – це основні об'єкти будь-якої бази даних. У таблицях зберігаються як усі дані, наявні в базі, так і структура бази (поля, їхні типи і властивості).

– запити – ці об'єкти слугують для вилучення даних з таблиць і надання їх користувачеві в зручному вигляді. За допомогою запитів виконують такі операції, як відбір даних, їх сортування і фільтрацію. За допомогою запитів можна виконувати перетворення даних по заданому алгоритму, створювати нові таблиці, виконувати автоматичне наповнення таблиць даними,

імпортованими з інших джерел, виконувати найпростіші обчислення в таблицях і інше. У принципі, ці ж операції можна робити і з таблицями. Але з міркувань зручності (у першу чергу швидкодії) і безпеки повинні використовуватися запити. По-перше, знижується ризик того, що необережними діями користувачі ушкодять дані в таблицях. По-друге, надавши різним користувачам різні запити, можна ефективно розмежувати їхній доступ до даних у строгій відповідності з колом персональних обов'язків. Тому в промислових СУБД користувач, як правило, не одержує доступу безпосередньо до таблиць, а працює із запитами. Особливість запитів полягає в тому, що вони беруть дані з базових таблиць і створюють на їхній основі тимчасову результуючу таблицю (її ще називають моментальним знімком).

– форми – це засоби для введення даних. Призначення форм – надати користувачеві засоби для заповнення тільки тих полів, до яких у нього відкритий доступ. Одночасно із цим у формі можна розмістити спеціальні елементи управління (лічильники, списки, перемикачі, прапорці та інші) для автоматизації вводу. Переваги форм розкриваються особливо наочно, коли відбувається введення даних із заповнених бланків. У цьому випадку форму роблять графічними засобами так, щоб вона повторювала оформлення бланка, — це помітно спрощує роботу і запобігає появі помилок.

2.2 Побудова інформаційно-пошукової системи (ІПС)

2.2.1 Принципи побудови, реалізації та використання ІПС

Інформаційно-пошукова система (ІПС) – це прикладне комп'ютерне середовище для обробки, зберігання, сортування, фільтрації і пошуку більших масивів структурованої інформації.

Кожна ІПС призначена для вирішення певного класу задач, для яких характерний свій набір об'єктів і їх ознак.

ІПС бувають двох типів:

– документографічні ІПС. У документографічних ІПС усі збережені документи індексуються спеціальним способом, тобто кожному документу привласнюється індивідуальний код, що становить пошуковий образ. Пошук відбувається не по самих документах, а по їхніх пошукових образах. Саме так шукають книги в більших бібліотеках. Спочатку відшуковують картку в каталозі, а потім по номеру, зазначеному на ній, відшукується й сама книга.

– фактографічні ІПС. У фактографічних ІПС зберігаються не документи, а факти, що ставляться до якої-небудь предметної області. Пошук здійснюється за зразком факту.

Кожна ІПС складається з бази даних і системи керування базами даних. Оскільки бази даних є основою для побудови інформаційно-пошукових систем, до таких баз даних пред'являють спеціальні умови, в першу чергу з точки зору надійності та безпеки.

До баз даних пред'являються особливі вимоги з точки зору безпеки, тому в них реалізований особливий підхід до збереження даних.

При роботі зі звичайними додатками для збереження даних ми видаємо відповідну команду, задаємо ім'я файлу і довіряємося операційній системі. Якщо ми закриємо файл, не зберігши його, то вся робота зі створення або редагування файлу пропаде безповоротно.

Бази даних — це особливі структури, цілісність яких не повинна залежати як від конкретних дій користувача, так і від збоїв операційної системи або відключень живлення. Проблема безпеки баз даних вирішується тим, що в СУБД для збереження інформації використовується подвійний підхід. У частині операцій, як звичайно, бере участь операційна система комп'ютера, але деякі операції збереження відбуваються в обхід операційної системи.

Операції зміни структури бази даних, створення нових таблиць або інших об'єктів відбуваються при збереженні файлу бази даних. Про ці операції СУБД попереджає користувача. Ці операції ніколи не проводять із базою даних, що перебуває в комерційній експлуатації, — тільки з її копією. У цьому випадку

будь-які збої в роботі обчислювальних систем не страшні. Операції ж по зміні змісту даних бази, що не зачіпають структуру, максимально автоматизовані і виконуються негайно та автоматично (без попередження).

Зазвичай, вирішивши відмовитися від змін у документі, його просто закривають без збереження і знову відкривають попередню копію. У СУБД цей прийом не працює. Усі зміни, внесені в таблиці бази, зберігаються на диску без нашого відома, тому спроба закрити базу «без збереження» нічого не дасть, тому що все вже збережене. Таким чином, редагуючи таблиці баз даних, створюючи нові записи і видаляючи старі, ми як би працюємо з жорстким диском напряму, мінаючи операційну систему.

Звичайно з базами даних працюють дві категорії виконавців. Перша категорія – це проектувальники. Їхнє завдання полягає в розробці структури таблиць бази даних і узгодженні її із замовником. Крім таблиць проектувальники розробляють і інші об'єкти бази даних, призначені, з одного боку, для автоматизації роботи з базою, а з іншого боку – для обмеження функціональних можливостей роботи з базою (якщо це необхідно з міркувань безпеки). Проектувальники не наповнюють базу конкретними даними.

Друга категорія виконавців, що працюють із базами даних, – це користувачі. Вони одержують вихідну базу даних від проектувальників і займаються її наповненням і обслуговуванням. У загальному випадку користувачі не мають засобів доступу до управління структурою бази – тільки до даних, та й то не до всіх, а до тих, робота з якими передбачена на конкретнім робочім місці.

2.2.2 Сутність та алгоритми пошуку інформації

Основні принципи інформаційного пошуку були сформульовані ще в першій половині минулого століття. По-перше, повинен бути створений масив покажчиків на інформаційні ресурси. Покажчик (index) містить у собі якусь властивість документа і посилання на документи, що володіють цією

властивістю. Показчики можуть бути різних видів. Широко розповсюджений, наприклад, авторський показчик. Такий показчик дозволяє одержати посилання на роботи автора, що цікавить нас. Також показчики можуть бути складені і по інших атрибутах документа. Процес створення показчиків на документи називається індексуванням, а терміни, що використовуються для індексування, називаються термінами індексування. Сукупність використовуваних термінів індексування називається словником.

Після створення індексу до нього звертаються за допомогою запитів. У зв'язку з тим, що процес пошуку полягає в зіставленні запиту користувача з наявними даними, отриманий запит також повинен бути переведений на мову індексування. В індексі виконується пошук відповідних до запиту документів, користувачеві видається список посилань на відповідні ресурси. Для підвищення швидкості індексування і пошуку, словник і індекс повинні бути впорядковані по системі, найбільш відповідній завданням пошуку в даній предметній області.

Функціонування сучасних ІПС засноване на двох припущеннях:

- документи, що необхідні користувачеві, об'єднані наявністю деякої ознаки або комбінації ознак;
- користувач здатний указати цю ознаку.

Обидва ці припущення на практиці не виконуються, і можна говорити тільки про ймовірність їх виконання. Користувач звичайно не має вичерпних знань про інформаційний зміст ресурсу, у якому проводить пошук.

Для підвищення якості пошуку можна використовувати різні методи. Найбільш уживаний з них - використання логічних операторів И (AND), АБО (OR), НЕМАЄ (NOT).

Використання логічних операторів – досить простий спосіб підвищити релевантність (відповідності видаваної інформації запиту), але він має і свої недоліки. Застосування оператора AND може сильно звузити видачу, а оператора OR - сильно розширити. Ступінь точності і повноти пошуку

залежить від того, наскільки загальні терміни використовувалися при формулюванні запиту. Може бути невірним використання як найбільш загальних термінів (зростає рівень інформаційного шуму), так і занадто специфічних термінів (знижується повнота пошуку). Використання занадто специфічних термінів може призвести до того, що в словнику ІПС цей термін може не виявитися.

У загальному вигляді процедура пошуку є процедурою ітеративної, тобто поетапної видачі результатів пошуку, коли відбувається корекція запиту, пошук по цьому запиту, тощо. Корекція запиту відбувається виходячи з кількості отриманої інформації і її релевантності і може виконуватися як користувачем, так і самою інформаційно-пошуковою системою.

Залежно від співвідношення повноти і точності знайдених документів, користувач може звужити або розширити область пошуку, перейшовши до більш загальних або, навпаки, більш специфічних термінів, а також використавши споріднені поняття.

У випадку пошуку по декільком термінах така корекція області пошуку може відбуватися по одному з декількох термінів, що дозволяє змінювати цю область досить плавно. Може виявитися корисним знання користувача про наявність безперечно релевантних документів. Не знайшовши їх у списку знайдених документів, область пошуку треба розширити

Таким чином, процес пошуку можна представити у вигляді наступних основних компонентів:

- формулювання запиту природньою мовою;
- формалізація запиту на відповідній ІПС;
- проведення пошуку в ІПС;
- огляд отриманих результатів (посилань);
- попередня обробка отриманих результатів: перегляд змісту посилань, добування і збереження релевантних даних;
- модифікація запиту у разі необхідності і проведення повторного (уточнюючого) пошуку з наступною обробкою отриманих результатів.

2.2.3 Програмна реалізація ППС

У реляційних СУБД підтримується єдина мова, що містить усі необхідні засоби – від створення БД до забезпечення користувацького інтерфейсу при роботі з даними - мова SQL (Structured Query Language). Однак, практично в кожній СУБД крім підтримки мови SQL (у її базовій версії) є також своя унікальна мова, орієнтована на особливості цієї СУБД і не придатна для інших систем. Тобто, усі СУБД незважаючи на зовнішню подібність, насправді суттєво різні

До найпоширеніших СУБД відносять:

– Microsoft Access – реляційна СУБД корпорації Microsoft. Має широкий спектр функцій, включаючи зв'язані запити, зв'язок із зовнішніми таблицями і базами даних. Завдяки вбудованій мові VBA, у самому Access можна писати додатки, що працюють із базами даних. СУБД Access проста у вивченні і експлуатації і тому доступна для користувачів з невисокою кваліфікацією, Як правило, Access використовується для створення особистих баз даних (довідники, записні книжки і т.д.), що не мають комерційного поширення.

– Paradox – одна з найстаріших локальних баз даних. Спочатку розвивалася фірмою Borland, згодом фірмою Corel. Має власну мову розробки додатків. Доцільна для невеликих БД персонального користування. У теперішній час розробка нових версій продукту припинена.

– Foxpro - реляційна СУБД корпорації Microsoft. Кожна таблиця зберігається в окремому файлі з розширенням dbf. Усі інші об'єкти – форми (form), запити (query), звіти (report), програми (program), меню (menu), уявлення (view) теж зберігаються в окремих файлах з відповідними типами. Використовує власну мову програмування Foxpro. У теперішній час розробка нових версій продукту припинена.

– Oracle database – об'єктно-реляційна система управління базами даних компанії Oracle. Включає управляючі файли, які містять дані про саму базу даних та без яких не може бути відкритий доступ до інформації бази даних, і також файли даних, що містять інформацію бази даних і оперативні журнали і дозволяють відновити базу даних після збою екземпляра.

– DB2 — сімейство систем керування реляційними базами даних, що випускаються корпорацією IBM.

– Microsoft SQL Server – СУБД, розроблена корпорацією Microsoft. Основна використовувана мова запитів — Transact-sql, створений разом з фірмою Sybase.

– MySQL – СУБД, що використовується в якості сервера, до якого звертаються локальні або віддалені клієнти. Добре взаємодіє з мовою програмування PHP, що визначає її популярність як бази даних серверних додатків.

СУБД Microsoft Access, Paradox та Foxpro належать до класу файл-серверних, у котрих база даних розташовується централізовано на файл-сервері, а сама СУБД розташовується на кожному клієнтському комп'ютері (робочої станції). Доступ СУБД до бази даних здійснюється через локальну мережу. На даний момент файл-серверна технологія вважається застарілою.

СУБД Oracle database, DB2, Microsoft SQL Server, та MySQL відносяться до клієнт-серверних СУБД, які розташовуються на сервері разом із БД і здійснюють доступ до БД безпосередньо, у монопольному режимі. Усі клієнтські запити на обробку даних обробляються клієнт-серверної СУБД централізовано.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Гурский Д. А. Вычисления в MathCAD 12 / Д. А. Гурский, Е. С. Турбина. – СПб. : Питер, 2006. – 544 с.

2. Макаров Е. Г. Инженерные расчеты в MatchCAD : учебный курс / Е. Г. Макаров. – СПб. : Питер, 2005. – 448 с.

3. Бююль А. SPSS искусство обработки информации. Анализ статистических данных и восстановление скрытых закономерностей : пер. с нем. / А. Бююль, П. Цефель. – СПб. : ДиаСофтЮП, 2005. – 608 с.

4. Салманов О. Н. Математическая экономика с применением MathCAD и Excel / О. Н. Салманов. – СПб. : БХВ-Петербург, 2003. – 464 с.

Навчальне видання

**ПОНОМАРЕНКО Євгеній Георгійович,
ЛОМАКІНА Ольга Сергіївна**

**ЗАСОБИ ТА ІНСТРУМЕНТИ
ІНЖЕНЕРНО-ЕКОЛОГІЧНИХ РОЗРАХУНКІВ**

КОНСПЕКТ ЛЕКЦІЙ

*(для студентів 1 та 2 курсу усіх форм навчання
освітньо-кваліфікаційного рівня «бакалавр»
спеціальності 183 – Технології захисту навколишнього середовища)*

Відповідальний за випуск *Т. В. Дмитренко*

За авторською редакцією

Комп'ютерне верстання *О. С. Ломакіна*

План 2017, поз 77 Л

Підп. до друку 05.03.2018. Формат 60 × 84/16.

Друк на ризографі. Ум. друк арк. 1,0

Тираж 50 пр. Зам. №

Видавець і виготовлювач:

Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Маршала Бажанова, 17, Харків, 61002.
Електронна адреса rectorat@kname.edu.ua.

Свідоцтво суб'єкта видавничої справи:

ДК № 5328 від 11.04.2017.