БАЗА ДАНИХ — ОСНОВНА СКЛАДОВА ІНФОРМАЦІЙНОЇ СИСТЕМИ

Руденко В.Д.

Анотація. Бази даних є невід'ємною складовою сучасних інформаційних систем. Тому вивчення БД необхідно починати з розгляду сутності інформаційних систем. Необхідно свідомо розуміти їхні основні функції, загальну структуру, класифікацію, а також роль та місце баз даних в інформаційних системах, а вже потім переходити безпосередньо до вивчення баз даних.

Ключові слова. Бази даних, інформаційна система, основні функції, індексування, мови баз даних.

* * *

Поняття інформаційної системи

Інформаційна система (ІС) — це система, у якій можуть накопичуватися величезні обсяги даних, і яка забезпечує автоматичний пошук, обробку і видачу необхідної інформації. Основою сучасних ІС є комп'ютери. Інформаційні системи складаються з технічного, програмного, інформаційного забезпечення, а також нормативно-правового забезпечення, яке регламентує порядок її функціонування та права й обов'язки посадовців, які з нею працюють або обслуговують.

Технічне забезпечення може містити не тільки комп'ютери, але й локальні мережі, канали зв'язку, по яких інформація приймається й передається на великі відстані, а також пристрої збирання і відображення інформації, наприклад, проектори, спеціальні табло тощо.

Програмне забезпечення — це сукупність програм, яка забезпечує створення і підтримку функціонування інформаційної системи. Воно може містити також **застосування** — програми, які розробляються, як правило, для певної категорії посадовців (не фахівців у галузі комп'ютерної техніки) з метою автоматизованого розв'язування необхідних задач. Приклад застосування — програма, яка розраховує прибутки супермаркету за добу за видами реалізованих товарів (овочі, м'ясо, крупи тощо). Для розробки застосувань часто використовуються CASE-засоби, тобто засоби, які автоматизують цей процес.

Інформаційне забезпечення — це сукупність накопичених даних, які можуть бути структурованими і неструктурованими. **Неструктуровані** дані — це звичайні документи: накази, статті, постанови, реферати тощо. **Структуровані** мають певну форму, тип даних, розмір тощо. Найчастіше вони оформлюються у вигляді двовимірних таблиць і називаються **реляційними базами даних**. Приклад таких даних подано у табл. 1.

Інформаційні системи нині застосовуються в різноманітних сферах людської діяльності, наприклад, бухгалтерський облік та аудит, бібліотечна справа, банківські та фінансові установи, системи продажу та резервування залізничних квитків, податкова служба, системи управління послугами та товарами. Структура та функції ІС значною мірою визначаються тією предметною галуззю, у якій вона застосовується. Але головна функція інформаційної системи незалежно від предметної галузі залишається незмінною і полягає в підвищенні **ефективності інтелектуальної праці** людини. Вона звільняється від виконання рутинних операцій. Забезпечується висока швидкість доступу до даних та отримання їх із джерел, розташованих на великих відстанях.

Інформаційні системи класифікуються не тільки за предметною ознакою, але й за багатьма іншими, основними з яких, на наш погляд, є призначення і спосіб доступу до даних. За призначенням розрізняють системи, зображені на рис. 1.

Рис. 1

Офісні системи призначені в основному для забезпечення життєдіяльності фірм, компаній, навчальних закладів, державних установ та ін. У них накопичуються дані про матеріальні цінності, людські ресурси, рух товарів тощо. За їх допомогою здійснюється автоматизація діловодства, документообіг.

Інформаційно-довідкові системи здійснюють пошук і видачу даних за певними критеріями, наприклад, видачу розкладу руху потягів із Києва до Львова.

Системи обробки транзакцій інтенсивно використовуються в банківських установах, системах резервування та продажу авіаквитків тощо. У них сукупність операцій розглядається як одна єдина операція. Якщо хоча б одна операція з цієї сукупності не виконана або виконана некоректно, здійснюється відновлення бази даних на початок транзакції.

Системи підтримки прийняття рішень видають у зручній формі, наприклад у вигляді гістограми, необхідні дані, аналіз яких дозволяє прийняти обґрунтоване рішення з керування певним об'єктом.

За способом доступу до даних інформаційні системи можна поділити на автономні, групові, корпоративні та інформаційні системи в Інтернеті (рис. 2).

Рис. 2

Автономні ІС — це, як правило, індивідуальні системи, реалізовані на персональних комп'ютерах, наприклад, на домашньому комп'ютері.

Таблиця 1

Номер студентського квитка	Прізвище студента	Факультет	Предмет	Оцінка
25120	Березко О.Н.	АСУ	Математика	4
25121	Сокіл М.П.	Інформатика	Бази даних	5

Групові інформаційні системи функціонують в основному в локальних мережах. У них можуть використовуватися як клієнтські бази даних (розташовані на комп'ютері клієнта), так і серверні бази даних (розташовані на серверному комп'ютері).

Корпоративні ІС використовуються у великих організаціях, підрозділи яких можуть бути розташовані на великих відстанях. Вони можуть об'єднувати декілька локальних мереж, причому сервером бази даних може бути будь-який сервер локальної мережі.

Інформаційні системи в Інтернеті суттєво відрізняються від попередніх. Інформація розміщується на вебсторінках, які розробляються за допомогою спеціальних мов, наприклад, мови HTML. Веб-сторінка може містити графічну, текстову, аудіо- та відеоінформацію, а також посилання на інші сторінки. Доступ до Інтернет-ресурсів здійснюється за допомогою веб-браузера, який фактично є програмою-клієнтом, а програма, яка надає ресурси, є веб-сервером.

Сутність реляційної бази даних

Є різні способи або моделі організації даних. Модель даних визначає, у який спосіб здійснюється об'єднання даних у структури різної складності. Надалі будемо дотримуватися такого визначення: база даних — це організована сукупність даних певної предметної галузі, які відображають стан об'єкта та зв'язки між ними. Предметною галуззю може бути, наприклад, школа, поліклініка, фірма, будівельний супермаркет тощо. На різних етапах розвитку баз даних застосовувалися різні моделі даних, наприклад, мережева, ієрархічна. Нині основною моделлю є реляційна. Перспективною вважається також об'єктно-орієнтована модель даних. Надалі будуть розглядатися тільки реляційні бази даних.

Реляційна модель даних була запропонована у 1970 році Е.Ф. Коддом. Її основою є математичне поняття відношення (англійською — relation), яке зображується у вигляді двовимірної таблиці. На рис. 3 зображена таблиця відношення УЧНІ, яка може належати базі даних ШКОЛА.

Рядок таблиці (або запис) називають кортежем відношення, а ім'я стовпця (поля) — атрибутом. Усю сукупність імен атрибутів відношення називають його схемою. Зазвичай, схема зображується так: вказується великими літерами ім'я таблиці, а в дужках — перелік імен стовпців. Наприклад, відношення, зображене на рис. 3, має таку схему: УЧНІ (Прізвище, Дата народження, Клас, Домашній телефон). Кількість атрибутів відношення (стовпців) називається його степенем, а кількість кортежів — кардинальністю або потужністю. Степінь відношення УЧНІ та його кардинальність дорівнюють 4. Сукупність всіх значень стовпців називають його доменом.

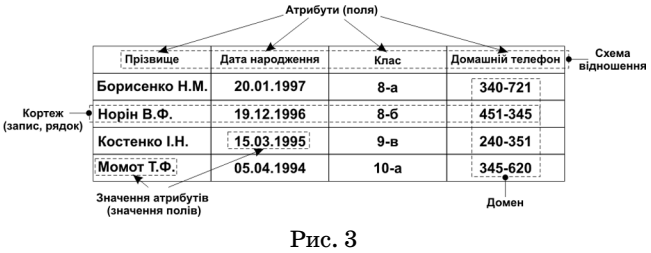

Рис. 3

До реляційної бази даних ставляться такі вимоги:

- у таблиці не можуть міститися однакові рядки;
- усі стовпці таблиці повинні мати різні назви;
- у кожному стовпці можуть міститися тільки однакові типи даних;
- у кожній клітинці таблиці повинні міститися тільки прості дані (не можна в одній клітинці вказувати, наприклад, дату народження і місце народження).

Таблиця повинна містити стовпець або декілька стовпців, які однозначно ідентифікують будь-який рядок. Такий стовпець (або сукупність стовпців) називають **ключем** відношення. Якщо ключ містить один стовпець, його називають простим, інакше — складеним. Таблиця може містити декілька ключів, але використовується лише один з них, який називають **первинним.**

Отже, основою реляційної бази даних є відношення (таблиця). Бази даних, які містять тільки одну таблицю, називають **простими**. Більшість баз даних містять десятки і сотні взаємопов'язаних таблиць. Зв'язки між таблицями здійснюються за допомогою ключових полів. Процес зв'язування таблиць і нормалізацію баз даних ми вже розглядали (див. журнал «Комп'ютер у школі та сім'ї» №8 за 2008 рік, стор. 46–49).

Індексування

Таблиці баз даних зберігаються на зовнішніх пристроях у тому вигляді, у якому вони створювалися і наповнювалися даними. Кожний наступний рядок вводиться в таблицю безпосередньо за попередньо введеним. У цих таблицях часто доводиться знаходити рядки за значенням ключового поля або іншого, а також за значеннями сукупності кількох полів. Наприклад, в таблиці ТОВАРИ необхідно знайти рядок, у якому значення поля ЦІНА дорівнює 300, або знайти рядки, у яких значиться, що товари надійшли після 10.01.2009 і мають ціну, більшу 500.

ТОВАРИ

Назва	Дата надходження	Ціна	Кількість
Костюм чоловічий	20.01.2009	600	5
Пальто жіноче	22.01.2009	730	3
Сорочка чоловіча	15.01.2009	70	10
Туфлі жіночі	10.01.2009	300	12

Пошук таких рядків можна виконати різними методами, наприклад, методом прямого перебору всіх рядків. Однак цей метод потребує багато часу на його реалізацію. Значно швидшим є бінарний метод пошуку, але він вимагає впорядкування рядків за значенням певного поля. Процес упорядкування даних у таблиці є тривалим. Він вимагає багато машинного часу, а збереження додаткових (упорядкованих) таблиць пов'язане зі значними витратами пам'яті.

Для розв'язання цієї проблеми використовуються **індексні таблиці** або просто **індекси**, процес створення яких називається **індексуванням**. Індекс найчастіше містить один стовпець, у якому записується вказівник на певний рядок таблиці з даними. Порядок розміщення вказівників в індексній таблиці визначається значенням певного поля таблиці даних. Наприклад, індексна таблиця для таблиці ТОВАРИ за значенням поля ЦІНА може мати такий вміст:

2
1
4
3

У даному випадку цифра 2 вказує на те, що першим рядком таблиці ТОВАРИ є її другий рядок. Цифра 1

— на те, що другим рядком є її перший рядок і т. д.

Таким чином, у процесі індексування не здійснюється упорядкування реально існуючих таблиць і не створюються нові таблиці даних, а створюються індекси, у яких вказівники визначають, у якому порядку повинні розміщуватися рядки основної таблиці за значенням певного її поля.

Індексні таблиці можна створювати як у порядку збільшення значень полів, так і у порядку їх зменшення. Якщо в таблиці змінюються, видаляються або додаються нові рядки, то сортувати цю таблицю не потрібно. Необхідно тільки оновити її індекси.

Створення індексних таблиць не потребує великих обсягів пам'яті для їх збереження, тому що вони, зазвичай, містять тільки один стовпець. Час їх створення значно менший часу впорядкування рядків у реальних таблицях даних.

Слід однак зауважити, що індексування не завжди дає позитивний ефект. Воно не доцільне для малих таблиць. Індексування доцільно виконувати для тих таблиць, які містять велику кількість рядків. Індексувати потрібно по тих полях, які будуть використовуватися для доступу до рядків таблиці даних. Не слід індексувати поля, які регулярно оновлюються, тому що це призведе до частого оновлення індексних таблиць і витрат машинного часу. Первинний ключ завжди необхідно індексувати. У СКБД MS Access індексування первинного ключа виконується автоматично.

Зауважимо також, що в деяких СКБД створюються індексні файли, які зберігаються окремо від основних таблиць. Для кожної таблиці можуть створюватися кілька індексних файлів. Наприклад, система FoxBase+ дозволяє для однієї таблиці створювати до 7-ми індексних файлів.

Функції та класифікація систем керування базами даних

З точки зору користувача будь-яка система керування базами даних виконує такі основні функції:

- створення і модифікація структури БД (визначення кількості таблиць, розподіл полів по таблицях, встановлення зв'язків між таблицями, визначення типів полів тощо);
- введення даних;
- маніпулювання даними (редагування, видалення даних та ін.);
- відбір необхідних даних та їх обробка;
- розробка й отримання різних вихідних документів.

Для виконання цих та інших функцій СКБД містить мову опису даних (МОД) і мову маніпулювання даними (ММД). МОД — це високорівнева, не процедурна мова декларативного типу, яка призначена для опису логічної структури даних. ММД — це мова, яка забезпечує виконання основних операцій над даними (введення, модифікацію, вибірку й обробку даних за допомогою запитів). У сучасних СКБД функції обох мов виконує єдина мова — мова SQL — структурована мова запитів.

На фізичному рівні СКБД виконує багато інших функцій, у тому числі управління даними в зовнішній пам'яті, забезпечення цілістності та безпеки даних, ведення журналу змін у БД та інші.

Нині є сотні різноманітних СКБД. Вибрати необхідну з них не просто. Для цього потрібно орієнтуватися в їх структурі та можливостях. Сучасні СКБД класифікуються за багатьма ознаками, основні з яких наведені на рис. 4.

Повнофункціональні СКБД повністю забезпечують виконання над даними різноманітних операцій. Вони виконують усі перелічені вище функції. Багато з них мають засоби програмування для професійних розробників, засоби автоматизації розробки баз даних (їх називають CASE — засобами), а також мову SQL-запитів. Цей клас СКБД є найбільш масовим. До нього відносяться, наприклад, такі системи: dBase IV, DataEase, MS Access, MS FoxPro та інші.

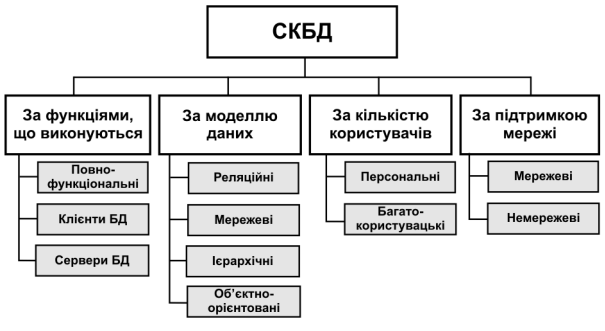

Рис. 4

Сервери баз даних — це потужні програми, які встановлюються на комп'ютерах-серверах. Зазвичай, дані від них запрошуються клієнтськими програмами. Якщо ці дані запрошуються операторами мови SQL, то такі сервери називаються SQL-серверами. Серверами баз даних є, наприклад, такі програми: InterBase, MS SQL, NetWare SQL та інші.

Клієнти — це програми, які встановлюються на комп'ютерах-клієнтах і які здійснюють зв'язок із серверами БД. Клієнтом можуть бути різні за призначенням стандартні програми, наприклад, повнофункціональні СКБД, програми електронної пошти, текстові процесори, електронні процесори та інші. Клієнтом також може бути і спеціально розроблене застосування. Доцільно, щоб клієнтські і серверні програми були розроблені однією фірмою, тому що у такому випадку розподіл функцій між ними здійснюється найраціональніше.

Для розробки користувацьких застосувань використовуються різноманітні програмні засоби, наприклад, Delphi, Visual Basic, Pover Builder та інші.

За моделлю використовуваних даних розрізняють реляційні, мережеві, ієрархічні, об'єктно-орієнтовані СКБД. Деякі СКБД підтримують роботу одночасно з кількома моделями даних. Найрозповсюдженішими нині є реляційні системи керування базами даних.

За кількістю користувачів бази даних розрізняють персональні і багатокористувацькі СКБД. Персональні системи використовуються для створення і підтримки персональних баз даних. Такі СКБД і розроблені за їх допомогою застосування використовуються як клієнтська частина багатокористувацької СКБД. Багатокористувацькі поєднують в собі сервер бази даних і клієнтську частини. У багатьох випадках вони забезпечують роботу в неоднорідному середовищі (системи з різними типами комп'ютерів і операційних систем). Прикладами таких систем є Oracle і Informix.

За підтримкою комп'ютерної мережі розрізняють мережеві і немережеві системи керування базами даних. Мережеві безпосередньо спрямовані на роботу в комп'ютерній мережі. Немережеві теж можуть використовуватися в комп'ютерних мережах, однак об-

мін даними між комп'ютерами здійснюється засобами операційних систем.

Мови баз даних

Зручний графічний інтерфейс сучасних систем керування базами даних забезпечує виконання різноманітних операцій над базами даних, починаючи від їх створення до отримання необхідних даних. Для непрофесійних користувачів не потрібно додаткових засобів для роботи з базами даних, їх потреби майже в повному обсязі забезпечують СКБД.

Разом з тим роботу з базами даних у багатьох випадках ефективніше виконувати за допомогою спеціальних мов баз даних, а для професійної роботи без них важко обійтися. Для реляційних баз даних розроблено кілька мов. Найпопулярнішими з них є такі:

- структурована мова запитів SQL (Structured Query Language);
- мова запиту за зразком QBE (Query By Example);
- мова запитів QUEL (Query Language).

Останнім часом фактично стандартом мов реляційних баз даних стала мова SQL, історія розвитку якої почалася у 1970 році. Нині є декілька стандартів цієї мови. Одним з останніх є стандарт SQL-2003. Команди мови SQL умовно поділяють на такі групи:

- DDL — мова визначення даних. Команди цієї групи призначені для створення і зміни структури об'єктів бази даних, наприклад, для створення і редагування таблиць;
- DML — мова маніпулювання даними. Використовується для зміни даних в об'єктах бази даних;
- DCL — мова управління даними. Команди використовуються для керування доступом до даних;
- DQL — мова запитів. Команди цієї групи забезпечують створення запитів до бази даних. **Запит** — це сукупність команд, яка забезпечує отримання з бази даних необхідної інформації;
- мова адміністрування базою даних і управління транзакціями.

Мова SQL — це єдина мова, яка поєднує функції усіх перелічених груп. Незважаючи на те, що її називають мовою запитів, вона містить усі необхідні засоби з керування базами даних. SQL є непроцедурною мовою реляційних баз даних. Вона описує, що потрібно одержати, а не як це зробити. Мова не містить операторів керування, введення-виведення та інших. Тому вона використовується, зазвичай, разом із вбудованою мовою програмування системи керування базами даних. У СКБД Access такою мовою є Visual Basic for Applications. Однак останнім часом мова SQL стала доповнюватися деякими процедурними можливостями. Наприклад, стандарт SQL-2003 має такі засоби процедурної мови, як блоки BEGIN, умовний оператор IF, функції і процедури. Але для створення повноцінних програм цих засобів не вистачає. Для цього необхідно використовувати інші класичні мови програмування.

Мова SQL існує у двох формах. В інтерактивній SQL користувач безпосередньо вводить запит і отримує результат. Команди вбудованої SQL включаються у тексти програм на інших мовах. Звернення до бази даних, а також обробка результатів виконується цими програмами або застосуваннями.

Отже, мова SQL — це потужний інструментарій роботи з базами даних. Різні стандарти цієї мови мають різні можливості. Крім того, один й той же стандарт мови в різних СКБД реалізуються не однаково. До основних загальних можливостей мови SQL можна віднести такі:

- створення і видалення таблиць бази даних та її об'єктів (доменів, індексів, процедур, що зберігаються тощо);
- маніпулювання даними в таблицях бази даних (вставлення, зміна, видалення даних);
- виконання пошуку даних у базах даних і їх упорядкування;
- підтримка обмежень цілісності та несуперечливості даних;
- здійснення резервного копіювання і відновлення бази даних.

Операції над даними виконуються за допомогою операторів SQL. Наведемо найголовніші з них.

1. Оператор створення таблиць:

```
CREATE TABLE <ім'я таблиці>
(<ім'я стовпця><тип даних>,
<ім'я стовпця><тип даних>...);
```

Обов'язковими операндами наведеного оператора є ім'я таблиці і хоча б одного імені поля з вказівкою типу даних, які будуть зберігатися в ньому. Мова SQL підтримує широкий набір даних, у тому числі рядкові, логічні, числові, дати і часу.

2. Оператор зміни структури таблиці:

```
ALTER TABLE <ім'я таблиці>
({ADD, MODIFY, DROP}<ім'я стовпця>[<тип даних>},
{ADD, MODIFY, DROP} <ім'я стовпця>[<тип даних>]...);
```

Службові слова оператора визначають: ADD — додати стовпець, MODIFY — змінити стовпець і DROP — видалити стовпець.

3. Оператор видалення таблиці:

```
DROP TABLE<ім'я таблиці>;
```

4. Оператор вибирання даних із таблиці:

```
SELECT <список імен стовпців>
FROM <ім'я таблиці>;
```

Цей оператор забезпечує вибірку всіх рядків зі значеннями вказаних в операторі SELECT стовпців. Рядки також можна відбирати за певною умовою за допомогою такого оператора:

```
SELECT <список імен стовпців>
FROM <ім'я таблиці>
WHERE <умова>
```

Цей оператор дозволяє також здійснювати вибірку рядків з кількох таблиць.

5. Оператор зміни рядків

```
UPDATE <ім'я таблиці>
SET <ім'я стовпця>=<вираз>
SET <ім'я стовпця>=<вираз>...
WHERE <умова>;
```

У рядках, які задовольняють умову, будуть змінюватися значення вказаних стовпців.

6. Оператор додавання нового рядка:

```
INSERT INTO <ім'я таблиці>
(<список стовпців>)
VALUES (<список значень>);
```

7. Оператор видалення рядків:

```
DELETE FROM <ім'я таблиці>
WHERE <умова>;
```

З таблиці видаляються рядки, які задовольняють зазначену умову.

Розглянемо приклад. Нехай таблиця COMPUTER має такий зміст:

COMPUTER

KOD	NAZVA	DATA	CINA
0250	Duron 1800	20.03.2009	2700
0315	Celeron 2600	22.03.2009	2450
0273	Athlon 2800	17.04.2009	2500
0309	Pentium 4 3000	19.04.2009	3200

Створити структуру цієї таблиці в комп'ютері можна за допомогою операторів:

```
CREATE TABLE COMPUTER
(KOD INTEGER,
NAZVA CHAR,
DATA DATE,
CINA INTERGER);
```

Вибрати з таблиці всі рядки зі стовпцями NAZVA і CINA можна за допомогою запиту:

```
SELECT NAZVA, CINA
FROM COMPUTER;
```

Вибрати з таблиці рядки з усіма стовпцями, ціна яких менше 2700, можна так:

```
SELECT *
FROM COMPUTER
WHERE CINA<2700;
```

На заключення відзначимо, що мова SQL особливо добре працює в системі клієнт-сервер. На комп'ютері-клієнті створюються запити мовою SQL, які передаються на сервер. Сервер бази даних інтерпретує, виконує запит і результат відправляє клієнту.

Поняття про розподілені та паралельні бази даних

У перших комп'ютерних мережах використовувалися централізовані бази даних, тобто бази даних, які зберігалися як єдине сховище даних на одному потужному комп'ютері. Усі користувачі цієї мережі мали доступ до єдиного сховища даних. У принципі на комп'ютерах-клієнтах могли зберігатися і власні (індивідуальні, автономні) бази даних, але вони не були пов'язані з централізованою базою даних.

Розподілені бази даних зберігаються частинами на комп'ютерах мережі (вузлах мережі), які логічно взаємопов'язані між собою. Є різні способи розподілу даних. Один з них передбачає наявність частини БД на кожному комп'ютері мережі (рис. 5).

Отже, на комп'ютерах зберігаються окремі частини БД, але ці частини мають між собою зв'язки і в сукупності вони є логічно єдиною базою даних. Для керування такими БД використовуються розподілені системи керування базами даних (РСКБД), які поділяються на два типи: однорідні та неоднорідні.

Однорідні РСКБД передбачають використання на кожному вузлі мережі однотипних моделей баз даних (наприклад, реляційних). Як правило, застосовуються однотипні комп'ютери з однаковою операційною системою та однаковим програмним забезпеченням. У неоднорідних РСКБД можуть використовуватися вузли різної апаратно-програмної платформи і різні моделі даних.

Зазначимо, що користувач кожного комп'ютера сприймає розподілену БД як єдине сховище даних. Для нього немає значення, чи зберігаються дані, якими він цікавиться, тільки на третьому комп'ютері, чи, наприклад, на першому, третьому і п'ятому. Він сприймає ці дані як єдине ціле.

Є два основних варіанти розподілу даних між вузлами мережі: фрагментація і реплікація. Сутність **фрагментації** полягає в тому, що БД поділяється на фрагменти, кожний з яких зберігається на певному вузлі мережі. Одиницею фрагментації може бути, наприклад, таблиця або сукупність взаємопов'язаних таблиць. У той же час допускається, що таблиця може бути представлена певною сукупністю фрагментів.

Рис. 5

Метод **реплікації** передбачає наявність копій одних і тих же даних на кількох вузлах мережі (не менше двох). Якщо вся база даних зберігається на всіх вузлах мережі, таку реплікацію називають повною.

Паралельні бази даних використовуються в багатопроцесорних системах. У процесі паралельної обробки даних одна задача поділяється на кілька менших, які одночасно виконуються на кількох вузлах комп'ютерної мережі, що прискорює час її виконання.

Розподіл БД у системах паралельної обробки даних залежить від архітектури багатопроцесорних систем, яка визначає спосіб взаємодії між процесором, оперативною пам'яттю і дисковим простором. Однак, незалежно від архітектури багатопроцесорних систем дані в них зберігаються на кількох магнітних дисках. Розподіл даних за багатьма дисками здійснюється для того, щоб кілька процесорів отримали одночасний доступ до даних, розміщених на магнітних дисках. Водночас на різних дисках можуть розміщуватися як окремі таблиці БД, так і окремі їх кортежі.

Зазначимо також, що в цих системах розпаралелюватися можуть не тільки дані між запитами, але й самі запити між різними процесорами.

Перспективними вважаються об'єктно-орієнтовані бази даних. У нинішній час ведуться активні дослідження в цій галузі. Однак на сьогодні відсутня загальноприйнята модель даних, а також єдина формальна теорія цих баз даних. Можна однак сподіватися, що ці проблеми будуть розв'язані й об'єктно-орієнтовані бази даних у майбутньому отримають широке розповсюдження.

* * *

Аннотация. Базы данных являются неотъемлемой составляющей современных информационных систем. Поэтому изучение БД необходимо начинать с рассмотрения сущности информационных систем. Необходимо сознательно понимать их основные функции, общую структуру, классификацию, а также роль и место баз данных в информационных системах, а уже затем переходить непосредственно к изучению баз данных.

Ключевые слова. Базы данных, информационная система, основные функции, индексации, языки баз данных.

Література

1. Базовий курс інформатики / В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу; За заг. ред. В.Ю. Бикова: [Навч. посіб.]. — К.: Вид. група BHV. — Кн. 2: Інформаційні технології. —2006. —368 с.: іл.
2. Информационные системы / Петров В.Н. — СПб.: Питер, 2002. — 688 с.: ил.
3. *Дж. Мартин.* Организация баз данных в вычислительных системах. Пер. с англ. — М.: Мир, 1980. — 662 с.: ил.