

## 4.6 Системи керування базами даних

Система керування (управління) базами даних СКБД (СУБД), англійською Database Management System (DBMS) – це набір взаємопов'язаних даних (база даних, БД) і програм для доступу до цих даних. Надає можливості створення, збереження, оновлення та пошуку інформації в БД з контролем доступу до даних.

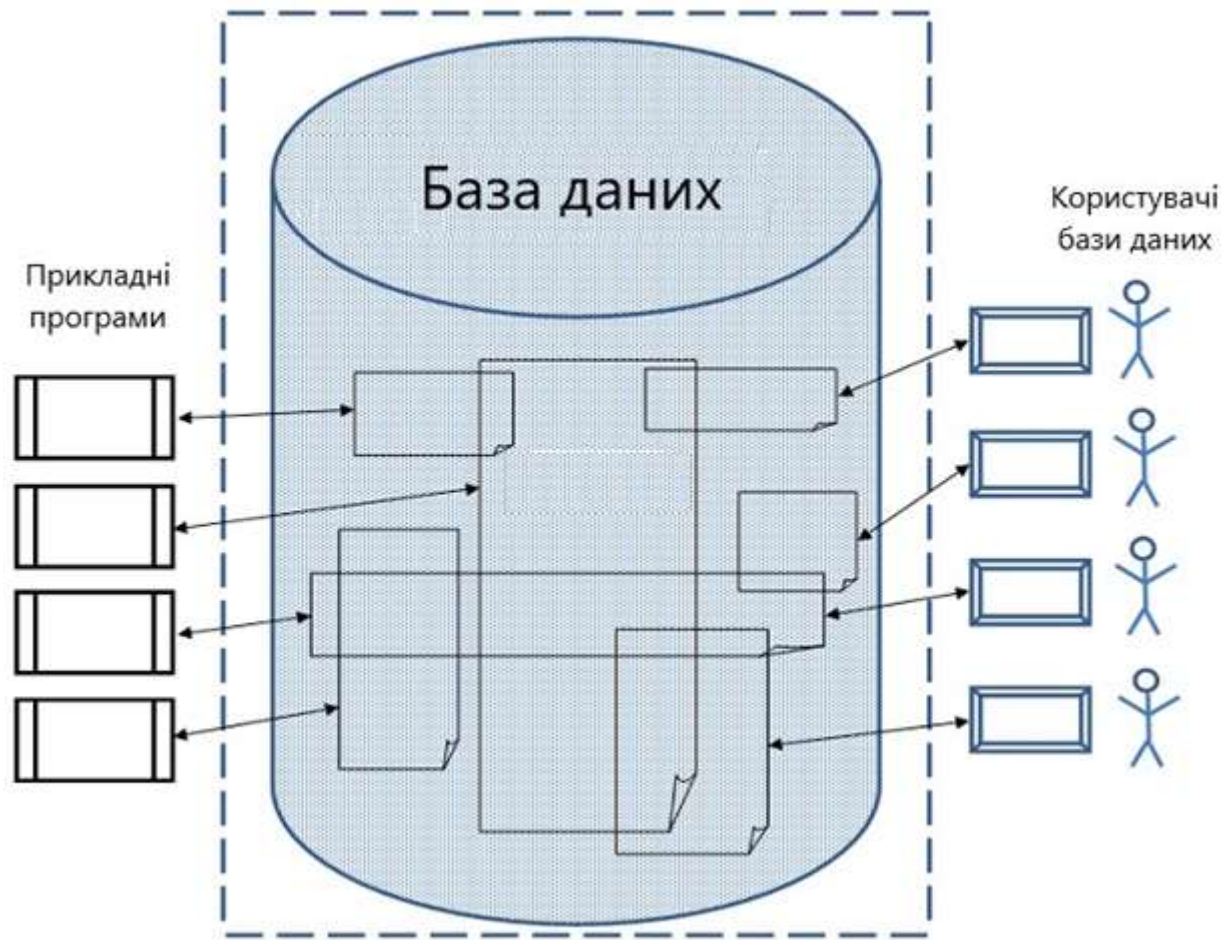
**Дані** в базі даних – це реальні значення даних, що зберігаються.

**Інформація** в базі даних – це пояснення змісту або суті реально збережених даних у БД.

Можливості СКБД:

- створення БД за допомогою мови визначення даних;
- додавання, оновлення, видалення та читання інформації з БД за допомогою мови запитів;
- контрольований доступ до БД із запобіганням несанкціонованого доступу;
- відновлення БД до стану, що був порушений в результаті збою апаратного або програмного забезпечення.

## Схема багатокористувацької бази даних



Трирівнева система організації СКБД:

- зовнішній рівень — представлення БД з точки зору користувача.
- концептуальний — описує які дані зберігаються і зв'язки між ними.
- внутрішній — фізичне представлення БД в комп'ютері.

## Основні характеристики СКБД:

- цілісність бази даних;
- інтегрованість даних;
- розділюваність даних;
- незалежність прикладних програм від даних;
- можливість спільного використання даних;
- підвищений рівень безпеки зберігання даних.

**Цілісність** бази даних [database integrity] — стан бази даних, коли всі значення даних правильні в тому сенсі, що відображають стан реального світу (в межах заданих обмежень по точності та часовій узгодженості) і підпорядковуються правилам взаємної не суперечливості.

**Інтегрованість даних** – це можливість представити базу даних різними файлами даних, які:

- можуть бути між собою об'єднані, наприклад, база даних реалізована в декількох файлах даних, інформація в яких доповнює одна одну;
- повністю або частково виключають надлишковість інформації. У цьому випадку структура файлів організована таким чином, що немає потреби задавати додаткову уточнюючу інформацію в одному файлі, яка вже є представлена в іншому файлі.

**Розділюваність даних** – це можливість доступу до одного й того ж елементу даних, що розміщується в базі даних, різними користувачами. Іншими словами, кожен з користувачів може використовувати (мати доступ) один і той же елемент даних на свій розсуд. Користувачі можуть мати паралельний доступ до елементу даних.

До програмного забезпечення систем баз даних відносяться:

- менеджер баз даних (Database Manager, DM) або сервер баз даних (Database Server, DS), DM є рівнем, що зв'язує між собою фізичну базу даних (дані, які реально зберігаються) та користувача системи;
- утиліти для організації роботи з базою даних;
- засоби розробки додатків, що взаємодіють з відомими базами даних;
- засоби проектування баз даних;
- генератори звітів на основі даних, що містяться у базі даних;
- менеджер транзакцій (transaction manager) або диспетчер виконання транзакцій (TP monitor).

**Транзакція** (transaction) — група послідовних операцій з базою даних, яка є логічною одиницею роботи з даними. Транзакція може бути виконана або цілком і успішно, дотримуючись цілісності даних і незалежно від інших транзакцій, що йдуть паралельно, або не виконана зовсім, і тоді вона не може справити ніякого ефекту.

# РЕЛЯЦІЙНА МОДЕЛЬ ДАНИХ

Реляційна модель даних для систем управління базами даних була розроблена співробітником дослідницької лабораторії компанії IBM Е.Ф. Коддом.

Основними поняттями реляційних баз даних є тип даних, домен, атрибут, кортеж, первинний ключ і ставлення.

**Тип даних** в реляційній моделі адекватний поняттю типу даних в мовах програмування. Основними типами даних в сучасних реляційних БД є символні, числові, бітові рядки, дата і час.

**Домен** – допустима потенційна множина значень даного типу. Наприклад, домен «Телефони» визначено на базовому типі рядків символів, але в число його значень можуть входити тільки ті рядки, які можуть зображати телефонний номер.

**Атрибут** – величина, що характеризує деяку властивість об'єкта, явища, процесу. Всі допустимі значення атрибута утворюють множину, звану доменом цього атрибута.

**Відносини** (relation) містять інформацію про один клас об'єктів предметної області. Кожному класу  $P$  об'єктів предметної області ставиться у відповідність деяка множина атрибутів  $A_1, A_2, \dots, A_n$ . Кожен об'єкт класу  $P$  описується рядком величин  $(a_1, a_2, \dots, a_n)$ , де який називається кортежем.

Таким чином, відносна  $P$  – множина кортежів, що описує весь клас об'єктів  $P$ .

Наочною формою подання відносин є таблиця, в якій значення атрибутів представлені стовпцями, а кортежі - рядками. База даних являє собою сукупність таких таблиць. Відзначимо, що при реалізації бази даних в СУБД стовпці таблиці називають **полями**, а рядки – **записами**.

У таблиці показаний приклад відносин класу Покупець, який містить кілька кортежів (рядків символів, що описують покупців).

| Код покупця | Прізвище | Телефон   | Електронна адреса | Поштова адреса   |
|-------------|----------|-----------|-------------------|------------------|
| 4551        | Петрова  | 125-15-97 | lpetr@newmail.ru  | вул. Зелена, 2-4 |

Реляційна модель даних деякої предметної області являє собою набір таких відносин або таблиць. Відносини будуються за певними правилами, що забезпечує мінімальну надмірність зберігання інформації, цілісність бази даних, легкість модифікації бази даних.

Основними правилами формування відносин є наступні:

- в таблиці не може бути повторюваних рядків;
- порядок рядків і стовпців в таблиці довільний;
- на перетині стовпця і рядка може перебувати тільки одне значення, наявність багатозначних атрибутів не допускається.

У кожній відносині повинен бути атрибут або набір атрибутів, значення яких однозначно визначають кортеж відносини. У зв'язку з цим вводиться поняття первинного ключа.

**Первинний ключ** - це атрибут або мінімальний набір атрибутів, що однозначно визначають кожен рядок. У наведеному прикладі первинним ключем є «Код покупця».



Первинний ключ є важливим поняттям в реляційних базах даних. Пошук будь-якого конкретного елемента даних проводиться за іменем таблиці, первинним ключем рядку й іменем стовпця (атрибута).

Первинні ключі використовуються для таких цілей:

- ідентифікації рядків в таблиці;
- прискорення роботи з рядками таблиці;
- зв'язування таблиць.

**Зв'язки між об'єктами** представляються в реляційної моделі зв'язками між таблицями за значеннями ключових атрибутів.

Зв'язок «один-до-багатьох» означає, що для запису з деяким значенням первинного ключа однієї таблиці може існувати 0, 1 або більше записів з тим же значенням відповідного поля в іншій таблиці.

Таблиця "Покупець"

| Код покупця | Фамилия | Телефон   |
|-------------|---------|-----------|
| 4551        | Петрова | 125-15-97 |
| 4552        | Краснов | 447-85-96 |
| ...         |         |           |
| 5875        | Иванова | 345-67-89 |

Таблиця "Замовлення"

| Код заказа | Код покупця | Дата доставки |
|------------|-------------|---------------|
| 0991       | 4552        | 08/06/2012    |
| 0992       | 5875        | 09/06/2012    |
| ...        |             |               |
| 1858       | 4552        | 21/07/2012    |

Найважливішою частиною інструментарію будь-якої СУБД є **мова опису даних і побудови запитів**.

Більшість використовуваних в наш час реляційних СУБД підтримують стандартизовану структурну мову запитів SQL (Structured Query Language).

Мова SQL призначена для виконання операцій над таблицями і над даними таблиць. До операцій над таблицями відносяться операції створення, видалення і зміни структури таблиць. До операцій над даними відносяться вибірка, зміна, додавання і видалення записів таблиці.

SQL є, таким чином, одночасно і мовою опису даних, і мовою побудови запитів.

Мова SQL є непроцедурною мовою, не містить операторів передачі управління, організації та виклику підпрограм. У зв'язку з цим SQL не використовується автономно, а являє собою засіб маніпулювання даними в рамках СУБД і її вбудованої мови програмування (наприклад, мови Visual Basic for Applications СУБД MS Access).

Основними об'єктами, з якими оперує SQL, є схеми, таблиці та подання, записи і поля.

**Поданням** називається множина записів, отримана в результаті вибірки даних з таблиць. Власне уявлення - це структура запиту, який зберігається в словнику СУБД, а таблиця уявлення створюється із записів таблиць в момент виконання запиту.

Основними об'єктами, з якими оперує SQL, є схеми, таблиці та подання, записи і поля.

**Поданням** називається множина записів, отримана в результаті вибірки даних з таблиць. Власне уявлення - це структура запиту, який зберігається в словнику СУБД, а таблиця уявлення створюється із записів таблиць в момент виконання запиту.

**Імена** таблиць і уявлень повинні бути унікальними в межах даної схеми, а імена стовпців - унікальними в межах даної таблиці. Імена об'єктів можуть мати до 128 символів і складатися з букв латинського алфавіту, цифр і символів підкреслення. Регістр символів (прописні або малі літери) значення не має.

**Оператори** мови SQL поділяються на оператори створення об'єктів бази даних і оператори маніпулювання даними.

Відзначимо, що мова SQL є основою багатьох СУБД, оскільки не залежить від специфіки комп'ютерних технологій. Вона дозволяє приймати запити від інших компонентів СУБД і призначених для користувача додатків.

## Достоїнства реляційної моделі:

- простота і доступність для розуміння кінцевим користувачем, оскільки єдиною інформаційною конструкцією є таблиця;
- при проектуванні реляційних баз даних застосовуються суворі правила, що базуються на математичному апараті;
- реляційна модель забезпечує повну незалежність даних, при зміні структури реляційної бази даних зміни, які потрібно зробити в прикладних програмах, як правило, мінімальні;
- для побудови запитів і написання прикладних програм немає необхідності знання конкретної організації бази даних у зовнішній пам'яті. Звичайно, при виконанні запитів на фізичному рівні виконується навігація по записах таблиць, однак ці дії проводяться процедурами самої СУБД.

Недоліком є те, що не завжди предметну область можна представити у вигляді сукупності таблиць. Так, в системах автоматизації проектування і автоматизованої розробки ППЗ потрібні набагато більш складні структури даних. Для цього використовуються постреляційні об'єктно-орієнтовані моделі.

У КІСУ в переважній більшості випадків для доступу до баз даних використовуються комп'ютерні мережі.

Архітектура клієнт-сервер є найбільш перспективною при організації роботи в мережах. У клієнт-серверних СУБД програми функціонально розділені на дві частини, звані сервером і клієнтом.

Серверна частина СУБД є основною програмою, яка виконує функції управління і захисту даних в базі даних, яка перебуває на комп'ютері-сервері. Якщо виклик функцій сервера виконується мовою SQL, то він називається SQL-сервером.

Клієнтська частина СУБД виконується на комп'ютері користувача і забезпечує інтерфейс користувача з базою даних, а саме: перетворює запити користувача в команди запитів до серверної частини, а при отриманні результатів може конвертувати і відображення інформації. Клієнтською програмою може бути будь-яка програма, що має інтерфейс з серверною програмою або СУБД.

Між клієнтської і серверної частинами системи можливі різні варіанти розподілу функцій.

**Модель віддаленого представлення даних** інакше називають моделлю сервера БД (DBS - DataBase Server). У цій моделі функції комп'ютера-клієнта полягають тільки в поданні отриманої з сервера інформації, а все управління і прикладні функції зосереджені на комп'ютері-сервері. Додатки реалізуються у вигляді так званих збережених процедур. Всі основні дії по колективному доступу до бази даних та їх обробки виконуються на потужному сервері, а клієнтам пересилаються тільки необхідні дані. Перевагою такого підходу є централізоване адміністрування БД і додатків на всіх етапах розробки, супроводу і модифікації системи, а також низьке завантаження комунікаційних каналів.

## **Віддалений доступ до даних (RDA - Remote Data Access )**

характеризується тим, що обробка і представлення даних виконуються на комп'ютерах користувачів (клієнтів). Звернення до сервера виробляється за допомогою SQL-запитів або викликом функцій спеціальної бібліотеки API (Application Program Interface). Основною перевагою такої схеми є той факт, що більшість існуючих СУБД підтримують SQL-інтерфейси і існує велика кількість систем розробки додатків клієнтської частини.

Недоліками є великий потік даних, переданих по мережі, і відсутність можливості централізованого адміністрування.

**У моделі розподіленої БД** дані зберігаються як на комп'ютері-сервері, так і на комп'ютері-клієнті. При цьому в локальній і віддаленій базі даних можуть зберігатися різні частини єдиної БД або ж локальна та дистанційна БД є синхронізуються одна з одною копіями. Системи з розподіленою БД мають велику гнучкість і живучість, однак складні в розробці і вимагають великих витрат комунікаційних і обчислювальних ресурсів.



