

Лекція 3. Поняття систем баз даних

Системи баз даних

Для створення, збереження і використання даних були розроблені спеціальні програмні засоби — *системи управління базами даних* (СУБД), де БД, як ми вже говорили, це структурована сукупність даних, збережена в обчислювальній системі.

Сукупність БД і СУБД являють собою *банк даних* (БНД).

СУБД — набір програмних засобів, що дозволяють [17]:

а) забезпечити користувачів мовними засобами визначення і маніпуляції даними (вибірка, відновлення, видалення). Такими засобами є мова визначення даних (МВД) і мова маніпулювання даними (ММД). „Мова даних“, що позначає або одну, або обидві ці мови, може бути включена в універсальну мову програмування (C, Pascal, C++, Delphi, C#, Java, ...) і називається *тоді підмовою* даних, або бути автономною і називаною *мовою запитів*;

б) забезпечити підтримку моделей даних користувача.

При логічному проектуванні апріорно визначені рамки, які використовуються для абстрагування предметної області, називають моделлю даних. Тобто, модель даних — це засіб для визначення логічного представлення фізичних даних, що відносяться до деякого додатка. Модель даних дозволяє представити різні зв'язки або властивості об'єктів реального світу, визначити способи одержання нових понять на основі наявних тощо;

в) забезпечити програму, що реалізує функції МВД і ММД, у яких можливе визначення, створення і маніпуляція даними.

г) забезпечити захист і цілісність даних. Мається на увазі захист від несанкціонованого доступу або несанкціонованих дій над даними з боку користувачів у багатокористувальницькому або мережевому режимах.

Цілісність має сенс при роботі в тих же режимах. Тобто, якщо один з користувачів вніс зміну в загальні дані, то ці зміни повинні бути доступні іншим користувачам чи отримані ними. У протилежному випадку виникають колізії, наприклад, знайомі ситуації продажу залізничних квитків на одне і теж місце.

Архітектура системи БД

Графічне представлення архітектури СБД [1] має такий вигляд (рис. 1).

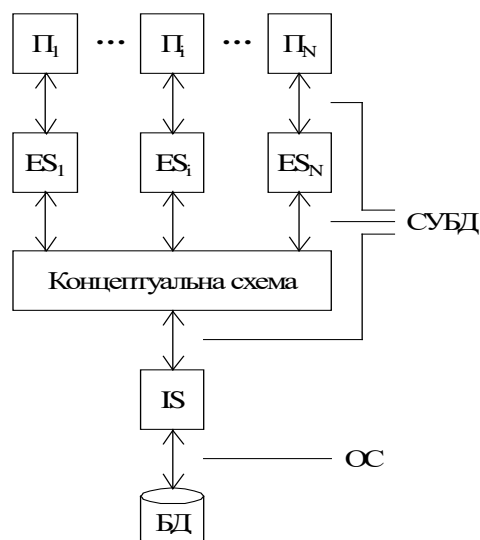


Рисунок 10 – Архітектура системи баз даних

Опис предметної області в термінах деякої моделі даних називають **концептуальною схемою**. Іншими словами концептуальна схема — представлення *усього* вмісту бази даних, яке описане за допомогою *концептуальної мови* визначення даних, плюс засоби безпеки і правила забезпечення цілісності. Ми підкреслюємо саме *концептуальної* МВД, тому що її визначення має відноситися тільки до змісту інформації. Тобто, у концептуальній схемі *не повинно бути* згадування про представлення збереженого файлу, послідовності збережених записів, індексування й інших подробиць збереження.

Такою системою може бути, наприклад, представлення предметної області у вигляді сукупності зв'язаних таблиць, що описують об'єкти і зв'язки між ними.

Відображення концептуальної схеми на фізичний рівень називають **внутрішньою моделлю** (схемою), що обумовлює різні типи збережених записів, індекси, фізичну послідовність збережених записів тощо.

На цьому рівні обумовлюється, чи зберігається БД у вигляді сукупності файлів, як у локальних *dBase*-подібних системах, або у вигляді таблиць бази даних на сервері баз даних у клієнт-серверних системах (типу *Oracle*, *Informix* тощо).

Однак ця схема не торкається реального фізичного боку збереження даних — дисків, доріжок, секторів тощо. Це турбота ОС (наприклад, єдиний диск або RAID-масив).

І, нарешті, конкретного користувача може цікавити не вся предметна область, описана концептуальною схемою, а лише деяка її частина. Представлення частини концептуальної схеми в термінах предметної області зветься **зовнішньою схемою** (моделлю) даних, прив'язаною до конкретного додатка. Таким чином, зовнішня схема — це вміст БД, яким його бачить певний користувач. Інакше кажучи, для користувача — зовнішнє представлення і є БД. Саме на цьому рівні створюється (реалізується) *користувальницький інтерфейс*, що з одного боку (з боку користувача) підтримує термінологію предметної області, а з іншого боку — бере участь у відображенні інформації на концептуальний рівень.

Тут необхідно ще раз згадати про мову даних. Точніше, про підмову. Частина зовнішньої схеми, що відповідає користувальницькому інтерфейсу, реалізується за допомогою *універсальної мови*, а та, котра відповідає за взаємодію з БД, точніше з концептуальною схемою, — за допомогою *підмови даних*. Звичайно, можна скористатися і мовою запитів, однак це значно звузить можливості організації користувальницького інтерфейсу. Очевидно, що при такому підході до зовнішньої схеми переважніше, щоб підмова даних і універсальна мова були *формально нерозрізнювані*. Такі нерозрізнювані або важкорозрізнювані мови називаються *сильнозв'язаними*. Прикладами можуть служити *dBase*-подібні мови і, відповідно, СУБД *Clipper*, *Paradox*, *dBase* тощо. Однак вони або дістали розвитку для ОС *MS-DOS*, або орієнтовані на роботу з локальними БД (максимум — на обчислювальну мережу в режимі файлу-сервера), або і те, й інше.

Інші універсальні мови (*C*, *Pascal*, *Basic* тощо) і вбудовані в них підмови даних є *слабкозв'язаними*, що не є перешкодою для створення якісних додатків.

Зокрема, існує мова даних, що може використовуватися і як мова запитів, і як вбудована практично в усі перераховані універсальні мови. Це мова *SQL* — Structured Query Language, що дозволяє працювати із системами клієнт/сервер, розподіленими і локальними БД. До неї ми повернемося пізніше.

На закінчення даного розділу відзначимо ще один аспект архітектури БД, а саме: **файлову архітектуру** і **архітектуру клієнт/сервер**. Причому і та й інша можуть бути реалізовані як на локальній машині, так і на мережі ЕОМ.

У випадку **файлової** архітектури (в обчислювальній мережі — файл-серверної) і

визначення, і маніпулювання даними, і взаємодія з користувачем (породження запитів і виведення результатів) здійснюється користувальницьким додатком, створеним за допомогою сильнозв'язаних мов. У багатокористувальницьких системах такого типу, тобто в архітектурі файл-сервер, користувач, точніше програміст повинен самостійно вирішувати питання забезпечення захисту і цілісності даних, що згадувалися раніше.

При реалізації архітектури **клієнт/сервер** систему баз даних поділяють на 2 частини: один або кілька *серверів* (машин) *БД* і множину *клієнтів* (множина може бути одиничною).

Сервер — це власне СУБД, що задовольняє усі відзначені раніше критерії. Відповідно, він підтримує всі основні функції СУБД на внутрішньому, концептуальному і, частково, зовнішньому рівнях, а саме, визначення даних, маніпулювання ними, забезпечення захисту і цілісності даних, обробку запитів тощо.

Клієнти — це різні користувальницькі і системні додатки, що реалізують інтерфейси користувачів, генерацію запитів і відображення результатів. Такі додатки створюються найчастіше за допомогою універсальних мов, які обов'язково включають підмови даних, що розуміються сервером.