

Лекція №4 ПОіП «СКБД ЯК ППП, ЩО ПОШИРЮЮТЬ МОЖЛИВОСТІ ОС»

(Модуль 1 -)

План лекції.

Основні поняття баз даних і системи керування БД (СКБД)	1
Організація даних	2
Властивості Відношень (таблиць БД)	4
Потенційні ключі	4
Цілісність даних	6
Функції системи керування БД	6

Серед ППП, що поширюють можливості ОС, особливе місце посідають системи керування базами даних.

Основні поняття баз даних і системи керування БД (СКБД)

Концепція БД ґрунтується на відокремленні процесів формування, введення та вибирання даних від процесу безпосереднього розв'язування проблемних задач.

Відмінність від підходу, пов'язаного з використанням процедурних мов програмування, існує у двох аспектах: 1) організація БД; 2) організація керування даними.

БД — це, по суті, пойменована сукупність даних, що відбиває стан об'єктів реального світу та відношень між ними в розглядуваній предметній сфері. Отже, **БД є моделлю предметної сфери.**

За структурою БД — це сукупність даних, організованих за певними правилами, що передбачають спільні принципи опису, збереження та маніпулювання даними *незалежно від прикладних програм.*

Атрибут (реквізит) — найменша одиниця даних.

Ключові атрибути (поля) — елементи, за якими можна впорядкувати певну сукупність даних.

Запис — одиниця обміну даними між програмою та зовнішньою пам'яттю.

Файл — пойменована сукупність усіх записів даного типу; найменша пойменована одиниця організації даних на зовнішньому носії.

Набір файлів — сукупність поріднених (тобто таких, що за певної умови та за визначеними ознаками можуть бути поєднані в одній групі) файлів.

Набір файлів утворює БД, якщо додержано певних вимог. Ідеться про те, що файли, поєднані в БД, мають бути:

- 1) інтегрованими — характеризуватися відсутністю або невеликою кількістю даних;
- 2) взаємозв'язаними — забезпечувати повну та несуперечливу інформацію про предметну сферу;
- 3) інформаційно-орієнтованими — містити такі відомості про об'єкти предметної сфери, в яких зацікавлений користувач;
- 4) незалежними — існувати незалежно від функцій, що їх вони підтримують.

Керування БД спрямоване на забезпечення розподілу даних, збереження їх незалежності, збільшення доступності, зменшення надмірності.

СКБД — це пакет прикладних програм, що містить сукупність мовних і програмних засобів, призначених для створення, ведення та спільного використання БД багатьма користувачами.

Основні вимоги до СКБД:

- 1) ненадмірність даних;
- 2) незалежність даних;
- 3) цілісність даних;
- 4) авторизація доступу.

Розглянемо кожну з цих вимог докладніше.

1. СКБД містять засоби, що дають змогу зменшити дублювання даних в інформаційній базі. Основним засобом досягнення ненадмірності є їх структуризація.

Зміст структуризації даних у базі: інформація, що є спільною для множини примірників даних, може бути розміщена в одному примірнику (запису) окремої групи даних.

2. У традиційних системах обробки даних, що базуються на процедурних мовах програмування, опис даних виконується безпосередньо у прикладних програмах, а тому будь-яка зміна опису даних призводить до необхідності модифікації прикладних програм. Під час використання СКБД описи даних зберігаються окремо, незалежно від прикладних програм. Завдяки цьому уможлиблюється модифікація описів даних, після виконання якої раніше написані прикладні програми виконуються успішно, без будь-якої зміни.

3. Існують поняття фізичної та логічної цілісності даних.

Фізична цілісність — це захист інформації від ненавмисного руйнування та збоїв обладнання. Забезпечується, як правило, спеціальними програмними засобами.

Логічна цілісність — це захист даних від неузгоджених або помилкових дій користувача.

4. За умов авторизованого (санкціонованого) доступу до БД передбачається введення засобів захисту, що забезпечує обмеження доступу до інформації або заборону доступу до даних небажаних користувачів.

Організація даних

Моделі даних, на яких базується СУБД:

- ієрархічна (деревоподібна);
- сіткова;
- реляційна.

Підвищення продуктивності прикладних програм, що працюють у середовищі СКБД, можна досягти за рахунок впорядкування даних та встановлення відношень між об'єктами БД.

Упорядкування даних можна здійснити:

- 1) сортуванням даних за ключем або ключами;
- 2) індексацією даних.

Термін «реляційний» (від англійського *relation* — відношення) указує перш за все на те, що така модель зберігання даних побудована на взаємовідношенні складових її частин, які зручно представляти у вигляді двовимірної таблиці. Таким чином, реляційна модель даних представляє інформацію у вигляді сукупності взаємозв'язаних таблиць.

Реляційний підхід до побудови бази даних дає змогу усунути усі недоліки, притаманні ієрархічним і сітковим моделям даних, і забезпечує такі переваги:

1. Простоту. Використання двовимірних таблиць — найпростіший засіб роботи з базами даних для непідготовленого чи не дуже досвідченого користувача.

2. Гнучкість. Полягає в тому, що відношення можна «розрізати» чи «склеювати» за допомогою операцій проєкції і об'єднання, що дозволяє користувачам діставати різноманітні файли в потрібній формі.

3. Точність. Напрямки зв'язку, які звичайно присутні у структурах баз, можуть бути пропущені. Відношення за своєю природою мають конкретніше значення і піддаються математично точним методам маніпулювання з використанням таких засобів, як алгебра відношень та обчислення відношень.

4. Реляційне зображення дає чітку картину взаємозв'язків реквізитів із різних відношень чи обчислень.

5. Реляційна база даних забезпечує незалежність даних, оскільки припускає можливість збільшення бази даних без зміни прикладних програм. Незалежність даних простіше забезпечити в нормалізованій базі, ніж у випадку ієрархічних — мережоподібних — структур. Реляційна база даних виглядає зрозумілішою, тоді як логічна схема бази даних під час зображення зв'язків залишається зрозумілою доти, доки кількість стрілок невелика. Зростання бази даних призводить до збільшення кількості зв'язків, що неможливо чітко відобразити на схемі з напрямленими зв'язками. А проте, використовуючи нормалізовані структури, можна задовольнити всі вимоги зростання і модифікації бази.

6. Таємність. Контроль таємності спрощується, оскільки для кожного відношення може бути задано правомірність доступу.

7. Простоту впровадження. Фізичне розміщення плоских файлів набагато простіше, ніж розміщення дерево- та мережоподібних структур.

Таким чином, таблиця має містити дані, що безпосередньо між собою пов'язані та визначаються предметною сферою.

Реляційна модель припускає три концептуальні елементи:

- Структура;
- Цілісність;
- Обробка даних.

Таблиця - це деяка регулярна структура, що складається з кінцевого набору однотипних записів.

Таблиця містить набір даних про об'єкт, наприклад автомобіль або перевезення. У таблицях дані розподіляються по стовпцях (які називають полями, **атрибутами**) і рядках (які називають записами, **кортеж**).

Наприклад, **кожне поле** в таблиці «Тип транспортного засобу» містить дані одного типу для кожного транспортного засобу (ТС), наприклад марка ТС. **Кожен запис** в таблиці містить повні відомості про одне ТС, такі як назва ТС, код ТС, тип ТС, вантажопідйомність і т.д.

Кожна таблиця представляє одну суть (наприклад, автомобілі), **кожен рядок в таблиці** містить дані про один екземпляр цієї суті (наприклад, про конкретний автомобіль). **Кожен стовець таблиці відповідає** одному атрибуту суті (наприклад, марка автомобіля, вантажопідйомність).

Властивості Відношень (таблиць БД)

У цих властивостях в основному і полягають відмінності між таблицями БД і таблицями.

1. У відношенні немає однакових кортежів. Таблиці на відміну від відносин можуть містити однакові рядки.
2. Рядки(записи) не впорядковані (зверху вниз). Дійсно, не дивлячись на те, що ми зобразили відношення "Співробітники" у вигляді таблиці, не можна сказати, що співробітник Іванов "передус" співробітникові Петрову. Рядки в таблицях впорядковані.
3. Атрибути не впорядковані (зліва направо). Оскільки кожен атрибут має унікальне ім'я в межах відношення, то порядок атрибутів не має значення. Стовпці в таблиці впорядковані.
4. Всі значення атрибутів атомарні(неподільні). Це витікає з того, що лежачі в їх основі атрибути мають атомарні значення. Це четверта відмінність відносин від таблиць - в елементи таблиць можна помістити що завгодно - масиви, структури, і навіть інші таблиці.

З цих властивостей відношення витікають важливі следствия.

Унікальність кортежів визначає, що у відношенні *завжди* є атрибут або набір атрибутів, що дозволяють *ідентифікувати* кортеж, іншими словами, у відношенні *завжди* є первинний ключ.

Кожна таблиця повинна містити ключ.

Ключем (Ключовим полем) називається таке поле(або декілька полій) таблиці, по якому можна визначити значення інших полів таблиці.

Потенційні ключі

Таблиця БД не може містити однакових рядків(записів). Це означає, що кожен запис унікальний. Насправді, **властивістю унікальності в межах таблиці можуть володіти окремі атрибути записів або групи атрибутів**. Такі унікальні атрибути зручно використовувати для ідентифікації кортежів.

Потенційний ключ, що складається з одного атрибуту, називається **простим**.

Потенційний ключ, що складається з декількох атрибутів, називається **складеним**.

Відношення може мати декілька потенційних ключів. Традиційно, один з потенційних ключів оголошується *первинним*, а інші - *альтернативними*.

Первинний ключ - це атрибут (або група атрибутів), які єдиним чином ідентифікують кожен рядок в таблиці. (т.е. значення в стовпці ніколи не повторюються).

Відмінності між первинним і альтернативними ключами можуть бути важливі в конкретній реалізації реляційної СУБД, але з погляду реляційної моделі даних, немає підстав виділяти таким чином один з потенційних ключів.

Зауваження. **Поняття потенційного ключа є семантичним поняттям і відображає деякий сенс (трактування) понять з конкретної наочної області.** Для того, щоб проілюструвати цей факт розглянемо наступне відношення "Співробітники":

Табельний номер	Прізвище	Зарплата
1	Іванов	1000
2	Петров	2000
3	Сидоров	3000

Таблиця 4 Відношення "Співробітники"

При першому погляді на таблицю, що зображає це відношення, може здатися, що в таблиці є три потенційні ключі - **в кожній колонці таблиці містяться унікальні дані**. Проте серед співробітників можуть бути однофамільці і співробітники з однаковою зарплатою. **Табельний же номер по суті свій унікальний для кожного співробітника.** Які ж міркування привели нас до розуміння того, що в даному відношенні тільки один потенційний ключ - "Табельний номер"? **Саме розуміння сенсу даних, що містяться у відношенні.**

Спробуємо представити це відношення в іншому вигляді, змінивши найменування атрибутів:

	А	В	С
1	Іванов		1000
2	Петров		2000
3	Сидоров		3000

Пред'явимо кому-небудь цю таблицю і не повідомимо сенс найменувань атрибутів. Очевидно, що неможливо судити, не розуміючи сенсу даних, може або не може в цьому відношенні з'явитися, наприклад, кортеж (1, Петров, 3000). Якби, до речі, такий кортеж з'явився (що, на перший погляд, цілком можливо, оскільки не порушується унікальність кортежів), то ми точно змогли б сказати, що *не є* альтернативним ключем - жоден з атрибутів окремо. Але ми не зможемо сказати, що ж *є* первинним ключем.

Поняття первинного ключа є виключно важливим у зв'язку з поняттям цілісності баз даних.

Взаємозв'язок таблиць є найважливішим елементом реляційної моделі даних.

Вона підтримується зовнішніми ключами.

Зовнішній ключ - це атрибут таблиці, який є первинним ключем іншої таблиці і служить для зв'язку з іншими таблицями.

Розрізняють чотири типи зв'язків між таблицями реляційної бази даних:

- *один до одного* — кожному запису однієї таблиці відповідає тільки один запис іншої таблиці;
- *один до багатьом* — одному запису головної таблиці можуть відповідати декілька записів підлеглої таблиці;
- *багато до одного* — декільком записам головної таблиці може відповідати один і той же запис підлеглої таблиці;
- *багато до багатьом* — один запис головної таблиці пов'язаний з декількома записами підлеглої таблиці, а один запис підлеглої таблиці пов'язаний з декількома записами головної таблиці.

Цілісність даних

Цілісність – властивість бази даних, що означає що вона містить повну, несуперечливу і адекватно наочну область, що відображає.

Реляційна модель даних накладає обмеження на структуру таблиць, які покликані забезпечити цілісність і точність трактування тих, що містяться в них даних. Деякі з них здаються очевидними, але вони корисні на етапі моделювання даних.

1-е обмеження називають **правилом цілісності суті**:

1. *Первинний ключ повинен бути єдиним.* **Первинний ключ** - це атрибут, який однозначно визначає кожен рядок таблиці (наприклад, **Код типові ТС** може бути первинним ключем для таблиці ТС). оскільки якщо у двох екземплярів суті однаковий первинний ключ, то їх неможливо відрізнити.

2-е обмеження називають **правилом посилальної цілісності**:

2. *Кожен зовнішній ключ повинен бути або порожнім, або відповідати одному із значень первинного ключа в іншій таблиці.* **Зовнішній ключ** - це атрибут таблиці, який є первинним ключем іншої таблиці. Зовнішні ключі використовуються для опису зв'язків між суттю.

Структура ідеальної БД повинна бути чіткою, ясною і прозорою. БД повинна бути вільною від баластної інформації: дублювання даних, синонімів, омонімів, груп даних, які повторюються, обчислювальних елементів.

БД повинна містити мінімальну кількість елементів даних, які несуть максимум інформації про стан ПС.

Функції системи керування БД

Загальну структурну схему взаємодії СКБД наведено на рис. 2.6.

Рис. 2.6. Загальна структурна схема взаємодії СКБД

Основні функції СКБД:

1. Обробка даних за алгоритмами прикладних програм. Засоби, що для цього використовуються, — вбудована в СКБД мова програмування, засоби автоматизації програмування.

2. Обробка даних за запитами користувачів — SQL (структурна мова запитів).

3. Реорганізація БД, тобто зміна на фізичному носії. Засоби реорганізації БД — інтерфейс користувача, вбудована мова програмування.

4. Забезпечення захисту даних. У VFP існують засоби підтримки авторизованого доступу до даних: ідентифікація користувача, ієрархічна схема прав користувача.

5. Підтримка цілісності даних:

- правила контролю індексів;
- правила контролю рівня полів та їх захисту, які дають змогу перевірити певні логічні умови та повернути повідомлення, якщо вони порушені;
- правила контролю рівня таблиць, правила, що одночасно застосовуються для двох або більше полів;
- збережені процедури — правило щодо використання двох або більше таблиць;
- правила цілісності посилань — регламентують коректність між таблицями.

Категорії користувачів СКБД за функціями, що їх вони виконують:

- ◆ кінцеві користувачі, які використовують СКБД для реалізації виробничих потреб;
- ◆ групи внутрішніх користувачів, що забезпечують нормальне функціонування БД;
- ◆ адміністратор БД і прикладних програм;
- ◆ системний програміст.

Адміністратор БД — особа, відповідальна за функціонування БД, забезпечення захисту БД від несанкціонованого доступу, некоректних користувачів, а також від збоїв обладнання. Крім того, до його функцій належать аналіз статистики використання БД, прийняття рішень про проведення реорганізації БД і здійснення реорганізації БД засобами СКБД або спеціально створеного програмного забезпечення.

Адміністратор прикладних програм формує (виділяє) для конкретних прикладних програм необхідну частку загальної моделі БД.

Системні програмісти виконують генерацію СКБД, стежать за нормальним функціонуванням СКБД у середовищі ОС, розробляють за завданням адміністратора БД програмні компоненти, що поширюють можливості БД.

У забезпеченні функціонування та використанні СКБД беруть участь різні категорії користувачів, тому функціональна, повна СКБД повинна мати засоби, що дають змогу задовольнити потреби користувачів різних категорій на всіх етапах життєвого циклу систем БД: проектування, створення, експлуатація БД.