**1. Ключі та нормалізація даних: основні нормальні форми (INF, 2NF, 3NF, BCNF)**

***Нормалізація* -** техніка розробки бази даних, яка зменшує надмірність даних і усуває небажані характеристики, такі як аномалії вставки, оновлення та видалення. Правила нормалізації розділяють великі таблиці на менші та зв’язують їх за допомогою зв’язків. Метою нормалізації в SQL є усунення надлишкових (повторюваних) даних і забезпечення логічного зберігання даних.

Ключі та нормалізація даних - це методи організації баз даних для забезпечення їх ефективності, цілісності та оптимальності. Основні нормальні форми включають:

1. *Перша нормальна форма (1NF):* Вимагає, щоб кожне поле в таблиці містило атомарні значення, тобто дані повинні бути атомарними, а не складними.

2. *Друга нормальна форма (2NF):* Таблиця повинна бути в 1NF, і кожен неключовий атрибут повинен залежати від цілого первинного ключа, а не від його складових частин.

3. *Третя нормальна форма (3NF):* Таблиця повинна бути в 2NF, і кожен неключовий атрибут повинен залежати тільки від первинного ключа, а не від інших неключових атрибутів.

4. *Форма Бойса-Кодда (BCNF):* Вимагає, щоб кожен неключовий атрибут повністю функціонально залежав тільки від первинного ключа.

**5. *Четверта нормальна форма (4NF)*:** Вирішує багатозначні залежності. Це гарантує відсутність кількох незалежних багатозначних фактів про сутність у записі.

**6. *П'ята нормальна форма (5NF)*:** Також відомий як «звичайна форма проекції-об’єднання» (PJNF), він стосується реконструкції інформації з менших, по-різному впорядкованих фрагментів даних.

**7. *Шоста нормальна форма (6NF):*** Теоретичний і мало впроваджений. Він має справу з тимчасовими даними (обробка змін у часі) шляхом подальшої декомпозиції таблиць для усунення всіх нечасових надмірностей.

Ці нормальні форми допомагають зменшити дублювання даних, забезпечити консистентність та ефективність запитів у базах даних.

Теорія нормалізації даних в MySQL сервер продовжує розвиватися. Наприклад, є дискусії навіть на 6th Нормальна форма. **Однак у більшості практичних застосувань нормалізація досягає найкращих результатів у 3rd Нормальна форма**. Еволюція нормалізації в теоріях SQL проілюстрована нижче:

**2. Основні концепції систем баз даних:**

**модель даних;**

**мова запитів;**

**транзакція;**

**ACID- властивості транзакції, індексування;**

**резервне копіювання та відновлення;**

**розподілепість і реплікація даних; безпека даних**

Основні концепції систем баз даних охоплюють такі аспекти:

*1. Модель даних*: Це абстрактне представлення даних та їх взаємозв'язків у системі баз даних. Приклади моделей даних включають ієрархічну, мережеву, реляційну та об'єктно-орієнтовану моделі.

*2. Мова запитів*: Це набір конструкцій, які дозволяють користувачам створювати запити до бази даних для отримання, вставки, оновлення або видалення даних.

*3. Транзакція*: Це логічна одиниця обробки даних, яка включає один або кілька запитів до бази даних. Транзакція має властивості ACID:

**Атомарність (Atomicity)**: Транзакція виконується повністю або не виконується взагалі. Якщо частина транзакції не вдалася, всі зміни скасовуються.

**Цілісність (Consistency)**: Транзакція переводить базу даних з одного цілісного стану в інший, дотримуючись всіх правил та обмежень.

**Ізоляція (Isolation)**: Транзакції виконуються незалежно одна від одної. Проміжні стани транзакції не видимі іншим транзакціям до її завершення.

**Стійкість (Durability)**: Після завершення транзакції її результати зберігаються навіть у разі збою системи.

4. Індексування: Це процес створення і управління індексами для прискорення пошуку та доступу до даних у базі даних. **Індекси в базах даних** – це спеціальні структури, які використовуються для прискорення пошуку та доступу до даних у таблиці. Вони діють подібно до індексу в книзі, дозволяючи швидко знаходити потрібні записи без необхідності переглядати всю таблицю.

1. **Первинний індекс (Primary Index)**:
   * Автоматично створюється на стовпці, оголошеному як первинний ключ.
   * Гарантує унікальність значень у стовпці.
2. **Унікальний індекс (Unique Index)**:
   * Забезпечує унікальність значень у стовпці, але дозволяє одне NULL значення.
   * Використовується для стовпців, де потрібно уникнути дублювання значень.
3. **Неунікальний індекс (Non-Unique Index)**:
   * Використовується для прискорення пошуку по стовпцях, які можуть містити повторювані значення.
4. **Складений індекс (Composite Index)**:
   * Створюється на кількох стовпцях однієї таблиці.
   * Корисний для запитів, що фільтрують або сортують за кількома стовпцями.
5. **Індекс повнотекстового пошуку (Full-Text Index)**:
   * Використовується для швидкого пошуку тексту у великих текстових полях.
6. **Кластеризований індекс (Clustered Index)**:
   * Змінює фізичний порядок рядків у таблиці для відповідності порядку індексу.
   * В таблиці може бути тільки один кластеризований індекс.
7. **Некластеризований індекс (Non-Clustered Index)**:
   * Створює окрему структуру, що вказує на фізичне розташування даних у таблиці.
   * Може бути декілька некластеризованих індексів у таблиці.

### Переваги індексів:

* **Прискорення запитів**: значно зменшують час виконання запитів на пошук, сортування і фільтрацію.
* **Підвищення продуктивності**: знижують навантаження на систему, оптимізуючи доступ до даних.

### Недоліки індексів:

* **Витрати на зберігання**: займають додатковий простір на диску.
* **Витрати на обслуговування**: потребують оновлення при додаванні, видаленні або зміні даних, що може уповільнювати операції запису.

5. Резервне копіювання та відновлення: Це процес створення резервних копій даних для забезпечення їх збереження та можливості відновлення у випадку виникнення непередбачуваних ситуацій.

6. Розподілепість і реплікація даних: Це можливості системи баз даних розподіляти та реплікувати дані на різних серверах з метою підвищення доступності та ефективності.

7. Безпека даних: Це набір заходів для захисту даних від несанкціонованого доступу, модифікації або видалення, включаючи аутентифікацію, авторизацію, шифрування та журналювання.

**3. Моделювання даних:**

**створення моделі даних для інформаційної системи;**

**концептуальна, логічна, фізична моделі даних;**

**ER -модсль;**

**нотації ER-моделей;**

*Моделювання даних* - це процес створення абстрактної представлення структури даних та їх взаємозв'язків у інформаційній системі. Основні аспекти моделювання даних включають:

*1. Концептуальна модель*: Це високорівневе представлення даних та їх взаємозв'язків без зв'язку з конкретною базою даних. Зазвичай використовується ER-модель для візуалізації цієї концепції.

*2. Логічна модель*: Це подальше уточнення концептуальної моделі з урахуванням специфікацій конкретної системи баз даних. Логічна модель визначає структуру даних, типи даних та взаємозв'язки між ними, використовуючи, наприклад, сутностно-реляційну модель.

*3. Фізична модель*: Це конкретне втілення логічної моделі на рівні фізичної бази даних. Вона визначає спосіб зберігання даних, включаючи таблиці, індекси, кластери та інші структури.

*ER-модель (сутність-зв'язок)* - це графічний спосіб представлення концептуальної моделі даних, що використовує сутності (об'єкти) та їх взаємозв'язки. Нотації ER-моделей включають сутності, атрибути, зв'язки та агрегації, які допомагають візуалізувати структуру даних та їх зв'язки.

**4. Реляційні бази даних:**

**особливості організації та зберігання даних у рсляційних базах даних;**

**основні характеристики реляцінпих баз даних;**

**DBMS (Database Management System)**

*Реляційні бази даних (РБД)* — це системи керування базами даних, що ґрунтуються на моделі даних, яка використовує реляційну алгебру для організації та зберігання даних. Основні особливості організації даних у РБД включають використання таблиць для представлення даних, де кожний рядок представляє запис, а кожний стовпчик — атрибут, а також використання ключів для забезпечення унікальності та відносин між записами.

Основні характеристики РБД включають **ACID-властивості** (атомарність, консистентність, ізольованість, довершеність) для забезпечення цілісності та надійності даних, а також мову запитів SQL для взаємодії з базою даних.

*Система управління базами даних (СУБД)* — це програмне забезпечення, що дозволяє створювати, керувати та маніпулювати реляційними базами даних. Вона забезпечує доступ до даних, забезпечує безпеку, підтримує виконання запитів та інші операції з даними. Наприклад, MySQL, PostgreSQL, Oracle, Microsoft SQL Server - це різні DBMS.

**5. Побудова запиту: мови**

**SQL (structured query language),**

**DDL (Data Definition Language),**

**DML (Data Manipulation Language),**

**DCL (Data Control Language),**

**TCL (Transaction Control Language)**

*SQL (Structured Query Language)* - це стандартна мова запитів, яка використовується для взаємодії з реляційними базами даних. Вона дозволяє створювати, змінювати та видаляти дані, визначати структуру бази даних та контролювати доступ до неї. SQL складається з різних підмов: DDL, DML, DCL та TCL.

*DDL (Data Definition Language)* - це підмова SQL, що використовується для визначення структури бази даних. Вона включає команди для створення, модифікації та видалення об'єктів бази даних, таких як таблиці, індекси, представлення тощо. (**CREATE, ALTER, DROP, TRUNCATE**)

*DML (Data Manipulation Language)* - це підмова SQL, яка використовується для маніпулювання даними в базі даних. Вона містить команди для вставки, оновлення, видалення та вибірки даних з таблиць.

(**SELECT, INSERT, UPDATE, DELETE**)

*DCL (Data Control Language)* - це підмова SQL, яка використовується для керування правами доступу до даних. Вона включає команди для надання або відмови у доступі до бази даних та її об'єктів.

(**GRANT, REVOKE**)

*TCL (Transaction Control Language)* - це підмова SQL, що використовується для керування транзакціями, які змінюють дані в базі даних. Вона містить команди для керування транзакціями, такі як COMMIT для фіксації змін, ROLLBACK для скасування транзакції та SAVEPOINT для встановлення точки відновлення.

**(COMMIT, ROLLBACK, SAVEPOINT, RELEASE SAVEPOINT, SET TRANSACTION)**

**6. Обробка запитів: основні операції реляційної алгебри:**

**відбір (selection),**

**проекція (projection),**

**об'єднання (union),**

**перетин (intersection),**

**різниця (difference),**

**декартовийдобуток (cartesian product),**

**об'єднання за атрибутом (Join),**

**ділення (Division)**

Основні операції реляційної алгебри використовуються для обробки запитів в реляційних базах даних:

*1. Відбір (Selection)*: Ця операція дозволяє вибрати рядки з таблиці, які задовольняють певний умовний вираз.

*2. Проекція (Projection)*: Проекція дозволяє вибрати лише певні стовпці з таблиці, виключаючи інші.

*3. Об'єднання (Union)*: Об'єднання дозволяє об'єднати результати двох або більше запитів у єдиний набір рядків.

*4. Перетин (Intersection)*: Ця операція повертає тільки ті рядки, які присутні в обох наборах результатів запитів.

*5. Різниця (Difference)*: Різниця повертає рядки, які присутні в першому наборі результатів, але відсутні в другому.

*6. Декартовий добуток* (Cartesian Product): Декартовий добуток повертає комбінації всіх рядків з двох або більше таблиць.

*7. Об'єднання за атрибутом (Join)*: Ця операція використовується для об'єднання даних з двох або більше таблиць на основі спільного атрибуту.

*8. Ділення (Division)*: Ділення є операцією, яка застосовується до двох відношень і повертає тільки ті значення, які мають специфічні відношення між ними.