

Введення до баз даних та ORM (Object-Relational Mapping)

Основи роботи з реляційними базами даних.

Реляційна база даних (РБД) - це колекція даних, організованих у вигляді таблиць, де дані зберігаються у рядках і стовпцях. Така структура дозволяє ефективно зберігати та керувати великими обсягами інформації, що є основою для багатьох сучасних додатків та систем.

1. Таблиці (Tables): Основна структурна одиниця РБД. Кожна таблиця має набір стовпців і рядків, де зберігаються дані.

The diagram shows a table with four columns: ID, Name, Phone, and Birthday. The first three rows contain data: (1, Andrew, 222-31-31, 1976), (2, Bohdan, 264-05-06, 1990), and (3, Ivan, 237-01-02, 1989). The last two rows are empty. A bracket on the left labeled 'Відношення (таблиця)' spans the entire table. A bracket above the 'Phone' column labeled 'Атрибут (стовпець)' spans that column. A bracket on the right labeled 'Кортеж (стрічка)' spans one row.

ID	Name	Phone	Birthday
1	Andrew	222-31-31	1976
2	Bohdan	264-05-06	1990
3	Ivan	237-01-02	1989

2. Стовпці (Columns): Кожен стовпець в таблиці представляє певний тип даних, такий як числа, рядки, дати тощо. Вони визначають структуру даних у таблиці.

3. Рядки (Rows): Кожен рядок таблиці містить конкретний набір даних, які відповідають стовпцям. Рядки представляють окремі записи чи кортежі даних.

4. Ключі (Keys): Ключі використовуються для ідентифікації записів в таблиці. Основними ключами є первинний ключ (Primary Key) і зовнішній ключ (Foreign Key). Первинний ключ гарантує унікальність записів, а зовнішній ключ встановлює зв'язки між таблицями.

5. SQL (Structured Query Language): SQL - це мова запитів, яка використовується для взаємодії з реляційними базами даних. Ви використовуєте SQL для створення, читання, оновлення та видалення даних.

6. Запити (Queries): Запити SQL використовуються для вибору конкретних даних з таблиць. За допомогою запитів ви можете фільтрувати, сортувати і агрегувати дані.

7. Нормалізація (Normalization): Нормалізація - це процес організації даних в РБД з метою запобігання дублюванню і забезпечення цілісності даних. Це важливий аспект проектування бази даних.

8. Транзакції (Transactions): Транзакції - це групи операцій з даними, які виконуються як один атомарний процес. Транзакції забезпечують консистентність та цілісність даних.

9. Індeksi (Indexes): Індeksi використовуються для прискорення пошуку даних в таблицях. Вони створюються на основі одного або декількох стовпців.

Поняття ORM та переваги його використання.

Об'єктно-реляційне відображення (ORM) - це технологія, яка дозволяє робити роботу з реляційними базами даних (РБД) більш простою та об'єктно-орієнтованою. Замість того, щоб писати запити SQL безпосередньо, ви взаємодієте з даними в РБД за допомогою об'єктів та методів, схожих на ті, які використовуються в об'єктно-орієнтованому програмуванні.

Основні поняття та переваги ORM включають наступне:

1. Об'єктна модель даних: ORM дозволяє вам визначати модель даних у вигляді об'єктів та класів. Кожна таблиця в РБД відображається на клас у вашій програмі, а кожен рядок - на об'єкт.
2. Спрощена робота з даними: ORM приховує складність SQL-запитів, що дозволяє вам взаємодіяти з даними за допомогою об'єктів та методів. Ви можете вставляти, оновлювати, видаляти та вибирати дані з РБД, використовуючи стандартні мови програмування.
3. Переносимість: ORM робить вашу програму менш залежною від конкретної РБД, оскільки більшість ORM-бібліотек підтримують різні системи баз даних. Ви можете легко переміщати вашу програму з однієї РБД на іншу без зміни коду.
4. Автоматична генерація SQL: ORM генерує SQL-запити за кулісами, що вимагає менше ручного коду. Це спрощує розробку та зменшує ризик помилок.
5. Розширені можливості: ORM надає багато додаткових можливостей, таких як кешування, ліниве завантаження, міграція даних, підтримка зв'язків між таблицями і багато інших.
6. Зручність у роботі з об'єктами: ORM дозволяє вам працювати з об'єктами даних, що робить код більш зрозумілим і підтримує більш зручну розробку.

Інтеграція ORM SQLAlchemy з FastAPI.

- Встановимо та налаштуємо PostgreSQL
- Налаштуємо наше середовище та підключимось до БД