Web Application Development

Database

KIEU Quoc Viet HUYNH Vinh Nam

Information and Communication Technology Laboratory (ICTLab),
University of Science and Technology of Hanoi

Hanoi, Oct 2024

Table of Contents

Why Care About Database Systems?

- Database systems store and manage web contents.
- Popular DB systems include: MySQL, Microsoft SQL, PostgreSQL, Oracle, DB2.
- MySQL is widely used by top websites like Facebook, Twitter, and Wikipedia.

Types of Databases

- **Relational Databases (RDBMS)**:
  - Store structured data in tables with relationships.
  - Examples: MySQL, PostgreSQL, SQLite.
- **NoSQL Databases**:
  - Store unstructured data and allow for flexible schema.
  - Examples: MongoDB, CouchDB.

Applications of Databases

- Relational DB: Best for structured data, complex queries, and transactional applications like banking systems.
- NoSQL DB: Used for large-scale, schema-less data handling, suitable for social media, real-time analytics.

Focus on Relational Databases: MySQL and SQLite

Relational Databases

- **MySQL**: Open-source, most popular relational DB for web applications.
- **SQLite**: Lightweight, often used in mobile apps or for local development.
- Structured query language (SQL) is used for managing relational databases.

Setting Up MySQL with XAMPP

- **XAMPP** includes MySQL, PHP, and Apache in one package, simplifying local development.
- It's cross-platform, making it easy to develop and test web apps locally.
- MySQL setup: Can be done easily within XAMPP.

Flask Framework for Web Development

Flask and ORM

- Flask is a lightweight Python framework that simplifies web application development.
- Perfect for integrating databases such as MySQL and SQLite using Object Relational Mapping (ORM).

ORM with SQLAlchemy in Flask

- SQLAlchemy allows you to map Python objects to database tables, simplifying database operations.
- **Benefits of ORM**:
  - Reduces the need to write raw SQL queries.
  - Makes it easier to switch between different types of databases (e.g., SQLite, MySQL).

Installing SQLAlchemy and MySQL Driver

- Install MySQL driver: `pip install mysqlclient`
- Install SQLAlchemy: `pip install SQLAlchemy`
- Setup connection in Flask:

Sample Project: Register Page with Flask and MySQL

- Objective: Build a simple user registration page using Flask and MySQL.
- User data will be saved to a MySQL database.
- Project structure:
  - `app.py`: Main Flask application.
  - `templates/`: HTML files for front-end.
  - `config.py`: Database configuration.

MySQL Query: Connecting Flask and MySQL

- First, connect to MySQL from Flask using SQLAlchemy:
- Create a user model for registration:

Handling Registration Form

- Flask-WTF can be used to create forms in Flask.
- Registration form structure:

Insert User Data into MySQL Database

- In the register route, collect form data and save it to the MySQL database:

Conclusion

- Databases are essential for storing and managing web content.
- MySQL and Flask make a powerful combination for building dynamic web applications.
- SQLAlchemy simplifies database interaction by providing an ORM.

Thank you for listening!