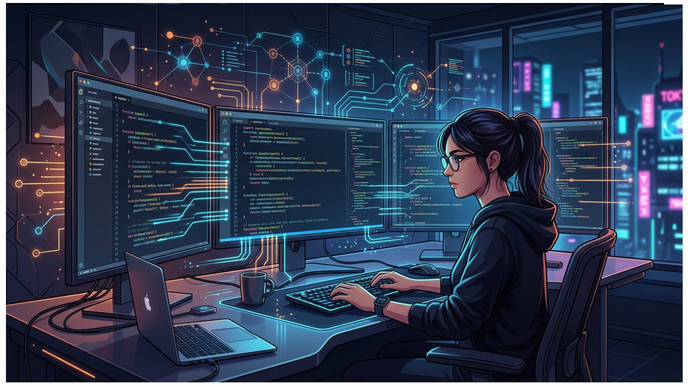
1/5/26 09:54

CSCI 275
Software Engineering
Lecture 3: Design SQL



Jetic Gū

Overview

- Focus: Course Introduction
- Architecture: Computer Systems
- Core Ideas:
    1. What is SQL
    2. Basic Tutorial on SQL

What is SQL?

- Structured Query Language
- Standard Language for **Relational Database Systems**
  - Modern software projects often rely on clearly designed database structures
  - Then, you can use whatever backend you want to access the data
- Advantage
  - Speed, Security, Easy to backup and deploy, Simultaneous Access, etc.

Concept

P1
What?

Common SQL Systems

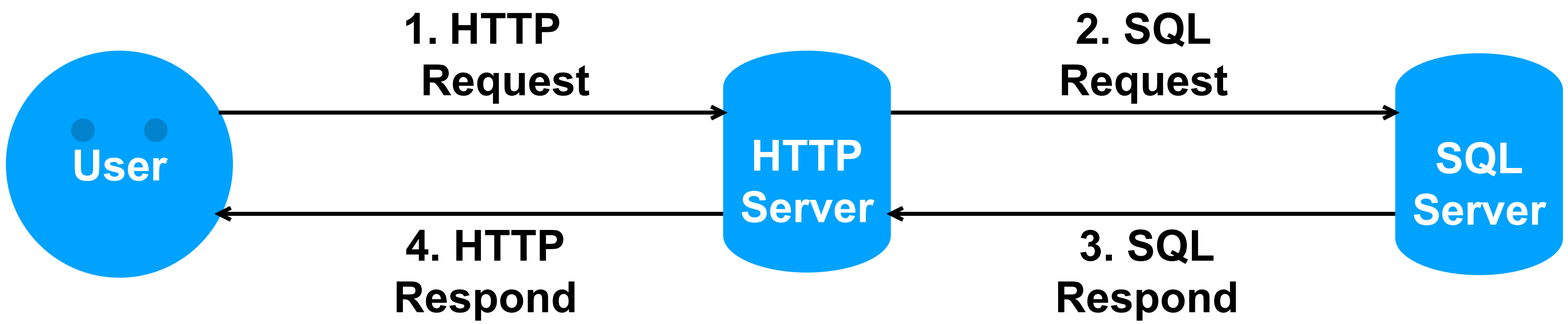
- OpenSource
  - MySQL
  - PostgreSQL
  - SQLite
- Cloud/Managed
  - Azure, Google Cloud, Amazon Aurora

- Are they the same?
  - No. Some SQL Databases have unique features
- Are they compatible with one another?
  - Different databases have different APIs to send SQL requests
  - SQL requests are mostly the same, with some offering unique features

Technical

P1
What?

How SQL Works



- An HTTP server communicates with an SQL server through query written in SQL
- An SQL server responds to queries. HTTP server combines the query responses with HTTP content, send to user
- HTTP server knows how the pages is styled, what **kind** of information should be presented. SQL servers store the **actual information** the should be presented
- Think: Why not let your HTTP server handle everything?

Concept

Some Key Items

- Users: an SQL server may have a lot of users. For example, an HTTP server can have one
  - I have one for my blog, another one for my online judge system, and another one for my experimental platform
  - Every separate HTTP service I provide has a separate one
- Databases: an SQL server may host different databases. Users have different permissions to these databases
  - Again, one for my blog to store all of the HTML code for my courses, one for my online judge to store all of the questions and submissions, one for my online expt to store experimental results

P1
What?

Some Key Items

- Tables
  - Inside an SQL database you can have multiple tables. These are where the data is actually stored
  - A table is like an excel sheet
  - Columns are usually typed, so numbers and strings can be queried differently. You can also have default values so when you create a new entry, you don't have to immediately populate all cells
- Records
  - Inside each table, a Record is like a row of data

Concept

P1
What?

SQL

- DDL: Data Definition Language -> Allows you to create databases, tables
  - `Create, Alter, Drop`
- DML: Data Manipulation Language -> Allows you to update records
  - `Insert, Update, Delete`
- DCL: Data Control Language -> Manage users
  - `Grant, Revoke`
- DQL: Data Query Language -> Get information from the database
  - `Select`

Concept

P2
Small Tutorial

What you need

- LAMP Stack
  - Linux, Apache2, MySQL, PHP
- phpmyadmin (only works for MySQL or MariaDB)
  - note: We are cheating (a bit) over here, you should learn the syntax and all in your own time, but this will get you foot in the door much quicker
- All demo done using WSL

Demo

P2
Small Tutorial

Step 1: Using APT

- On Linux, we use APT to install system packages and apps
- Use command line
  - `sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql`

Demo

P2
Small Tutorial

Step 2: Setting Up Apache2+PHP

- Apache2 usually keeps all your HTML stuff under `/var/www/html`
  - For example: `index.html`
  - You access this file using `http://localhost` or `http://localhost/index.html`
- Write a new `index.php` file

```
<?php
phpinfo();
?>
```

Demo

P2
Small Tutorial

Step 3: Setting Up MySQL

- I usually test my setup using phpmyadmin, which is an app built using php
- Why?
  - Important: command line is irreplaceable
  - But: I get lazy. If I can get it done in less time using GUI, I will do it
  - Plus: it shows you what SQL query it's going to use before you do stuff

Demo