

Week 1: Introduction & Foundational Skills (Focus on Project Relevance)

Part 1: Understanding SQL (30 minutes)

1. Research

- i. In a single Word document, summarize your findings in a short paragraph (3-5 sentences).**

Web Applications:

SQL (Structured Query Language) acts as the unseen hero in managing data for dynamic websites like online stores. It facilitates the storage and retrieval of crucial information, including product details (descriptions, prices, stock levels), user accounts (login credentials, profiles), and order history (items purchased, customer information). Essentially, SQL enables the website to efficiently interact with the database behind the scenes, ensuring a smooth user experience.

- ii. Write a short explanation (3-5 sentences) in your document about the role of SQL in web applications.**

- SQL enables databases to store and organize large amounts of data efficiently, such as product information, user accounts, and order details.
- SQL allows web applications to retrieve specific data based on user requests. For instance, when a user searches for a product on an online store, SQL queries the database to find matching products and return the results to the user's browser.
- SQL helps maintain data integrity and security by providing mechanisms to control access to data.
- SQL facilitates updates to product information, user accounts, and order details. For example, when a user adds an item to their shopping cart or modifies their profile information, SQL commands are executed to update the database accordingly.

- iii. List 3 benefits of using SQL for web applications.**

- *Data Integrity:* SQL ensures data consistency and accuracy through constraints such as unique keys, foreign keys, and data validation rules, maintaining data integrity across the application.
- *Scalability:* SQL databases are designed to handle high volumes of data and concurrent user requests effectively, making them scalable solutions for growing web applications without compromising performance.
- *Efficient Querying:* SQL's powerful query language allows developers to retrieve specific data quickly and efficiently using various commands like SELECT, JOIN, and WHERE, optimizing performance even with large datasets.

- *Security*: SQL databases offer robust security features such as access control, encryption of sensitive data, and transaction management (ACID properties), ensuring that only authorized users can access and manipulate data, thereby enhancing overall application security.

iv. Think about efficiency, data organization, and data retrieval capabilities.

Briefly explain each benefit in your document (1-2 sentences per benefit).

- *Efficiency*: SQL databases are optimized for efficient data storage and retrieval. They use indexing, which speeds up data retrieval by allowing the database to locate specific records quickly.
- *Data Organization*: SQL databases structure data into tables with defined schemas, facilitating systematic and consistent storage of information, which simplifies data maintenance and enhances application performance.
- *Data Retrieval Capabilities*: SQL's query language enables precise data retrieval through features like filtering (WHERE clause) and joining tables (JOIN clause), empowering developers to fetch specific information efficiently for web applications.

v. List any 3 Database Management Systems.

- *MySQL*: A popular open-source relational DBMS known for its ease of use and wide adoption.
- *PostgreSQL*: Another powerful open-source relational DBMS offering advanced features and strong community support.
- *Microsoft SQL Server*: A widely used commercial relational DBMS known for its scalability and integration with other Microsoft products.

Part 2: Database Fundamentals (45 minutes)

Question 2.1: Tables

I. Think about how data is organized in rows and columns.

In your document, define a database table and explain its similarity to a spreadsheet (2-3 sentences).

A database table is a structured collection of data organized into rows and columns. Each row represents a unique record or entity, while each column represents a different attribute or data field. This structure is similar to a spreadsheet, where rows correspond to individual data entries (like cells in a spreadsheet) and columns represent specific types of information (such as categories or fields). Both tables and spreadsheets provide a systematic way to organize and manage data, with rows and columns facilitating easy access, manipulation, and analysis of information.

Question 2.2: Columns

Consider different types of data like text, numbers, and dates.

Define "columns" and provide an example with an explanation (2-3 sentences) in your document.

- Columns in a database refer to the vertical entities within a table that define the attributes or fields of the data being stored. For instance, you might have a customer table with columns for customer names, ages, and email addresses. Each row in the table would then represent a single customer record, containing their corresponding name, age, and email address.

Data Types: Why are data types important in a database? Briefly explain 3 common data types (e.g., Text, Number, Date).

- Data types define the format and kind of data that each column in a database table can hold.

Using appropriate data types is crucial for several reasons:

- Maintaining data integrity: Ensures data is stored and processed correctly. For example, a column storing age should use a numeric data type to allow for proper mathematical operations.
- Saving storage space: Data types use optimized storage based on the data kind. For instance, a column storing postal codes might use a text data type that only allocates enough space for the specific characters needed.
- Saving storage space: Data types use optimized storage based on the data kind. For instance, a column storing postal codes might use a text data type that only allocates enough space for the specific characters needed.

Common data types:

- *Text*: Stores textual data like names, addresses, or descriptions.
- *Number*: Stores numeric data like integers, decimals, or currency values.
- *Date*: Stores date and time information.

Question 2.3: Data Types

Think about how data types ensure data integrity and efficient storage.

Explain the importance of data types and provide brief explanations of 3 common types (2-3 sentences each) in your document.

- *Data Integrity*: Data types act like gatekeepers, only allowing valid values into their respective columns. Imagine a column for age - a text data type wouldn't prevent someone from entering "hello" instead of a number. Data types ensure data remains consistent and usable for its intended purpose.

- *Efficient Storage:* Data types allocate storage space based on the data's characteristics. A numeric data type for a zip code uses less space compared to a generic text type that could hold lengthy descriptions. This minimizes wasted storage and optimizes database performance.

Common Data Types:

- Text: Stores variable-length character strings, ideal for names, addresses, or descriptions. It offers flexibility but doesn't enforce specific formatting or length limitations.
- Number: Represents numeric values like whole numbers (integers), decimals, or currency. This type ensures calculations and comparisons involving these values are accurate and reliable.
- Date: Dedicated to storing date and time information. It ensures consistent formatting and allows for specialized functions like date calculations and comparisons within the database.

Part 3: Expense Tracker Database Design (45 minutes)

3.1. Planning:

We'll be building an Expense Tracker application. What kind of data do you think we'll need to track? List at least 5 data points relevant to our project.

- * Consider information like expense amount, date, and category.
- * List your identified data points in your document.

3.2. Tables:

Considering the data points you listed, design a basic database schema with one main table (likely named "Expenses").

- * Define the columns needed for this table.
- * Assign appropriate data types to each column based on the kind of data it will hold. (e.g., amount: number, date: date, category: text)

In your document, create a table structure that includes:

- * Table name (e.g., Expenses)
- * Column names (e.g., expense_id, amount, date, category)
- * Data type for each column **(e.g., INT, DECIMAL, DATE, TEXT)**

Table Name: Expenses

Column Names	Data Types
Expense_id	INT(11)
amout	DECIMAL
date	DATE
category	VARCHAR(255)
description	TEXT
Payment_method	VARCHAR(255)

expense_id (INT): This column uniquely identifies each expense record.

amount (DECIMAL): This column stores the monetary amount of each expense, accommodating numbers with decimal points for precision.

date (DATE): This column records the date on which the expense occurred, facilitating date-based queries and reporting.

category (VARCHAR): This column categorizes expenses (e.g., food, travel, utilities), storing variable-length text entries.

description (TEXT): This column provides additional details about the expense, allowing for longer text entries.

payment_method (VARCHAR): This column specifies the payment method used for the expense, storing variable-length text entries.

Bonus:

