

Name: Leah Sigana

1.1 Imagine a dynamic website like an online store. How do you think SQL plays a role in managing data behind the scenes? Consider how product information, user accounts, and order details might be stored and accessed.

SQL is essential for maintaining data in a dynamic web application, such as an online store, behind the scenes. Product data is kept in tables with columns denoting different details, such as the product's name, price, and description. Tables are also used to hold user accounts, including usernames, passwords, and addresses.

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

This data is inserted, updated, and retrieved via SQL queries. Based on the search parameters, SQL queries effectively obtain pertinent data from the product table when a user looks for a product. Similar to this, SQL is used to add new entries into the order table when a user puts an order, capturing information such as the goods bought, their quantities, and the total amount.

1.3 & 1.4. List 3 benefits of using SQL for web applications. Think about efficiency, data organization, and data retrieval capabilities. Briefly explain each benefit in your document

- **Efficiency:** SQL's robust query features enable efficient data retrieval. This guarantees quick loads and a seamless user experience—particularly when working with big datasets.
- **Data Organization:** Tables and data types are how SQL enforces data structure. This encourages consistency and integrity in the data, which makes managing and maintaining the data easier as the program expands.
- **Data Retrieval Capabilities:** SQL provides an expressive and versatile language for data queries. To extract precise information, you can design complicated queries, filter, sort, and merge data from various tables.

1.5. List any 3 Database Management Systems.

PostgreSQL

Microsoft SQL Server

MySQL

2.1: 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 structured set of data arranged in rows and columns is called a database table. Comparable to a spreadsheet, each row denotes a record (a single product or person, for example), and each column

denotes a particular attribute of that record (product name, price, username, email, etc.). It is possible to store, retrieve, and manipulate data efficiently thanks to its hierarchical format.

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

Columns define the specific attributes or fields within a table. Each column has a designated data type, which specifies the kind of data it can hold. Data types are crucial for several reasons:

- i. **Data Integrity:** Data types guarantee that only legitimate information is put into a given column. To enforce numerical values with decimals, a price column could, for instance, have a decimal data type. By doing this, errors and inconsistencies in the database are avoided.
- ii. **Efficient Storage:** Data types enable the database to allocate memory in an efficient manner by taking into account the type of data (text takes up more space than numbers, for example).
- iii. **Search and Retrieval:** Data types facilitate effective data filtering and searching. For example, you can use a numerical comparison on the price column to look for products that fall into a certain price range.

2.3: 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.

- i. **Text (VARCHAR):** This type of data stores alphanumeric characters (letters, numerals, and symbols) for textual information such as names or product descriptions.
- ii. **Number (INT, DECIMAL):** Indicates values in numerical form. Product quantity and other whole numbers are represented by INT, but product price and other decimal numbers are represented by DECIMAL.
- iii. **Date (DATE):** Keeps track of calendar dates for time-sensitive data, such as purchase or expiration dates.

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.

- i. **Expense ID (INT):** Unique identifier for each expense entry.
- ii. **Amount (DECIMAL):** Amount of the expense (e.g., 25.50).
- iii. **Date (DATE):** Date the expense was incurred.
- iv. **Category (TEXT):** Category of the expense (e.g., groceries, transportation, entertainment).
- v. **Description (TEXT):** Optional field for a brief description of the expense.

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 Name	Data Type
expense_id	INT (Primary Key)
amount	DECIMAL
date	DATE
category	TEXT
description	TEXT (Optional)

Bonus:

Sketch a simple Entity Relational Diagram (ERD) of your table structure, including column names and data types.

Use drawing software or a simple table format to visually represent your schema.

```
Expenses
+-----+-----+-----+-----+
| expense_id | X | amount | date | category |
+-----+-----+-----+-----+
| INT | DECIMAL | DATE | TEXT | TEXT |
+-----+-----+-----+-----+
```

****Key:****
X - Primary Key

