Part 1: Understanding SQL

Question 1.1: Web Applications

In a dynamic website like an online store, SQL plays a crucial role in managing data. It allows the storage and retrieval of product information, user accounts, and order details efficiently. SQL ensures that data is organized and easily accessible, facilitating a seamless user experience on the front end.

Question 1.2: Role of SQL in Web Applications

SQL is used in web applications to manage and manipulate data stored in relational databases. It allows developers to create, read, update, and delete data, ensuring that the application can interact with the database efficiently.

Question 1.3: Benefits of Using SQL for Web Applications

1. Efficiency
2. Data Organization
3. Data Retrieval Capabilities

Question 1.4: Explanation of Benefits

1. **Efficiency:** SQL is optimized for managing large volumes of data, making it efficient for handling the complex queries required in web applications.
2. **Data Organization:** SQL uses a structured format to store data in tables, which helps in maintaining a well-organized database.
3. **Data Retrieval Capabilities:** SQL's powerful query capabilities allow for quick and precise data retrieval, enhancing the performance of web applications.

Question 1.5: Database Management Systems

1. MySQL
2. PostgreSQL
3. Microsoft SQL Server

Part 2: Database Fundamentals

Question 2.1: Tables

A database table is a structured collection of data organized in rows and columns. It is similar to a spreadsheet because both use a tabular format to display and manage data.

Question 2.2: Columns

Columns represent the different attributes of data in a table, such as text, numbers, and dates. For example, in a table of employees, columns might include EmployeeID, Name, and HireDate. Each column holds a specific type of data, ensuring consistency.

Question 2.3: Data Types

Data types are important because they define the kind of data that can be stored in each column, ensuring data integrity and efficient storage.

1. **Text:** Used to store non-numeric data, such as names and descriptions.
2. **Number:** Used to store numeric data, such as prices and quantities.
3. **Date:** Used to store dates and times, such as order dates and birthdates.

Part 3: Expense Tracker Database Design

Question 3.1: Planning

For our Expense Tracker application, we need to track the following data points:

1. Expense Amount
2. Date of Expense
3. Category of Expense
4. Description of Expense
5. Payment Method
6. Vendor

Question 3.2: Tables

Based on the data points listed, we will design a basic database schema with one main table named "Expenses." Below is the structure of the "Expenses" table:

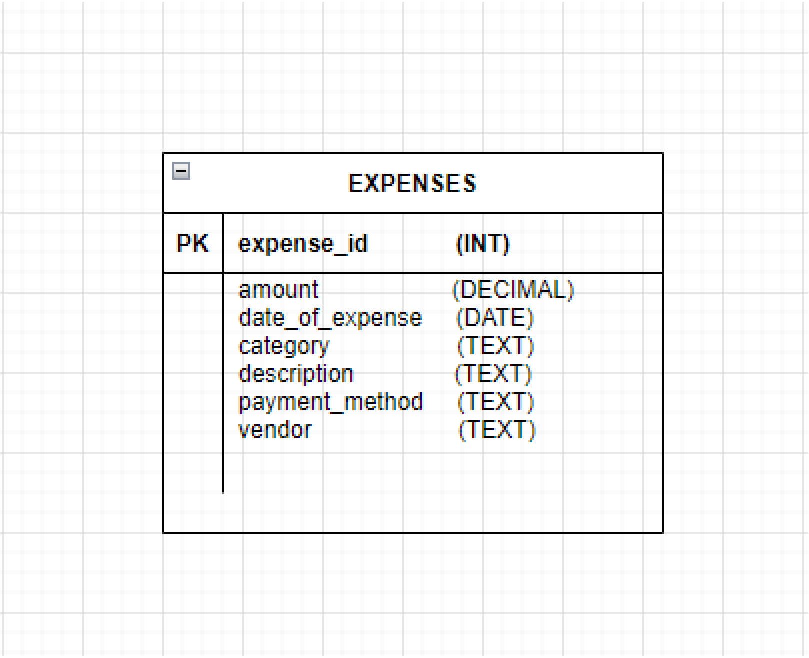
Table Name: Expenses

Column Name	Data Type	Description
expense_id	INT	Primary key, unique identifier
amount	DECIMAL	The amount of the expense
date_of_expense	DATE	The date the expense was made
category	TEXT	The category of the expense
description	TEXT	A description of the expense
payment_method	TEXT	The method of payment used
vendor	TEXT	The vendor or place of purchase

Data Type Explanations:

- **expense_id (INT):** A unique identifier for each expense.
- **amount (DECIMAL):** The monetary value of the expense.
- **date_of_expense (DATE):** The date when the expense was incurred.
- **category (TEXT):** The category to which the expense belongs (e.g., food, transport).
- **description (TEXT):** A brief description of the expense.
- **payment_method (TEXT):** The method used to pay for the expense (e.g., cash, credit card).
- **vendor (TEXT):** The vendor or place where the expense was made.

Part 4: Entity Relationship Diagram



EXPENSES

PK	**expense_id**	**(INT)**
	amount	(DECIMAL)
	date_of_expense	(DATE)
	category	(TEXT)
	description	(TEXT)
	payment_method	(TEXT)
	vendor	(TEXT)