### Answers

Part 1: Understanding SQL

1.1. In web applications like an online store, SQL plays a crucial role in managing the data behind the scenes. Product information, user accounts, and order details are typically stored in a relational database, and SQL is used to interact with this data. SQL allows the web application to efficiently store, retrieve, and manipulate the data as needed to support the dynamic functionality of the website.

1.2. SQL enables web applications to manage data effectively by providing a standardized language for interacting with relational databases. SQL allows developers to create, update, and query the database, ensuring that the data is organized, secure, and readily accessible to the application.

1.3. Benefits of using SQL for web applications:

1. Efficiency: SQL allows for efficient data retrieval and manipulation, enabling the web application to respond quickly to user requests.

2. Data Organization: SQL helps maintain a well-structured and organized database, making it easier to manage and scale the application as it grows.

3. Data Retrieval Capabilities: SQL's powerful querying capabilities enable the web application to quickly and accurately retrieve the specific data it needs, improving the overall user experience.

1.4. Efficiency: SQL's ability to quickly and efficiently retrieve and manipulate data is crucial for web applications, as they need to respond to user requests in real-time. This helps ensure a smooth and responsive user experience.

Data Organization: SQL's relational database model allows web applications to maintain a well-structured and organized data storage system. This facilitates data management, data integrity, and scalability as the application grows.

Data Retrieval Capabilities: SQL's advanced querying features, such as filtering, sorting, and joining data from multiple tables, enable web applications to quickly and accurately retrieve the specific data they need. This improves the overall functionality and performance of the application.

1.5. Three Database Management Systems (DBMS):

1. MySQL

2. PostgreSQL

3. Microsoft SQL Server

Part 2: Database Fundamentals

2.1: Tables

A database table is similar to a spreadsheet, as it organizes data into rows and columns. Just like a spreadsheet, a database table consists of records (rows) and fields (columns), allowing for structured storage and retrieval of information.

2.2: Columns

Columns in a database table represent the different types of data stored for each record. For example, a "Products" table might have columns for "Product Name," "Price," and "Category." The data types assigned to these columns determine the kind of information that can be stored, such as text, numbers, or dates.

Data Types:

Data types are important in a database because they ensure data integrity and efficient storage. Common data types include:

Text: Used to store alphanumeric characters, such as names, descriptions, and addresses.

Number: Used to store numeric values, such as prices, quantities, and IDs.

Date: Used to store calendar dates, which are often important for tracking order history, event scheduling, and other time-based information.

2.3: Data Types

Data types are crucial in a database because they ensure data integrity and efficient storage. Common data types include:

Text: Text data types, such as VARCHAR or CHAR, are used to store alphanumeric characters, including letters, numbers, and special symbols. They allow for the storage of information like names, descriptions, and addresses.

Number: Numeric data types, such as INT or DECIMAL, are used to store numeric values, including integers, decimals, and other mathematical quantities. These data types are essential for storing and manipulating information like prices, quantities, and IDs.

Date: Date data types, such as DATE or DATETIME, are used to store calendar dates and, in some cases, times. These data types are important for tracking time-based information, such as order history, event scheduling, and other applications where the timing of events is crucial.

Part 3: Expense Tracker Database Design (45 minutes)

3.1. Planning:

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

1. Expense Amount: The monetary value of the expense.

2. Expense Date: The date when the expense was incurred.

3. Expense Category: The category or type of the expense (e.g., food, transportation, utilities).

4. Expense Description: A brief description of the expense.

5. Payee: The person or entity to whom the expense was paid.

3.2. Tables:

Based on the identified data points, the main table for the Expense Tracker application could be designed as follows:

Table Name: Expenses

|  |  |
| --- | --- |
| Column Name | Data Type |
| expense\_id | INT |
| amount | DECIMAL(10,2) |
| date | DATE |
| category | TEXT |
| description | TEXT |
| payee | TEXT |

Explanation:

- `expense\_id`: An unique identifier for each expense record (primary key).

- `amount`: A decimal value to store the monetary amount of the expense, with a maximum of 10 digits and 2 decimal places.

- `date`: A date value to store the date when the expense was incurred.

- `category`: A text field to store the category or type of the expense.

- `description`: A text field to store a brief description of the expense.

- `payee`: A text field to store the name of the person or entity to whom the expense was paid.

This table structure should provide a solid foundation for the Expense Tracker application, allowing users to store and manage their expenses effectively.