**QUESTION 1: Understanding SQL**

**Question 1.1: Summary of Findings**

SQL plays a crucial role in managing data behind the scenes of dynamic websites like online stores. It facilitates the storage, retrieval, and manipulation of various types of data including product information, user accounts, and order details. By utilizing Structured Query Language (SQL), web developers can effectively handle complex data operations such as selecting, inserting, updating, and deleting records, ensuring that the website's functionality remains robust and efficient.

**Question 1.2: Explanation of SQL's Role in Web Applications**

SQL acts as the backbone for most modern web applications by enabling the creation and maintenance of relational databases. It allows developers to write commands that query and transform data stored in these databases, making it possible to dynamically populate web pages with personalized content and perform actions based on user interactions.

**Question 1.3: Benefits of Using SQL for Web Applications**

1. **Display of User Data**: SQL helps in displaying user data accurately by retrieving and formatting information from the database according to the application's requirements.
2. **Data Mining**: SQL aids in extracting valuable insights from large datasets, helping developers analyze user behavior and improve the site's features accordingly.
3. **Data Manipulation**: SQL simplifies the process of manipulating data within the database, allowing quick updates and modifications without needing to manually adjust HTML pages.

**Question 1.4: Efficiency, Data Organization, and Data Retrieval Capabilities**

1. **Efficiency**: SQL enhances efficiency by providing optimized methods for querying large datasets quickly, ensuring that web applications load faster.
2. **Data Organization**: SQL organizes data into structured tables, rows, and columns, which makes it easier to manage complex databases and scale them according to application demands.
3. **Data Retrieval Capabilities**: The ability to write precise queries allows developers to retrieve specific pieces of information from the database rapidly, supporting real-time updates within dynamic websites.

**Question 1.5: Database Management Systems**

1. **MySQL**: Known for its ease of use and robust community support.
2. **PostgreSQL**: Preferred for its reliability and advanced features like ACID compliance.
3. **Microsoft SQL Server**: Offers comprehensive management capabilities along with tight integration with other Microsoft products.

**QUESTION 2: Database Fundamentals**

**Question 2.1: Definition of a Database Table**

A database table is a structured collection of data organized in rows and columns, similar to a spreadsheet. Each row represents a unique record, while each column represents a specific attribute or field of that record.

**Question 2.2: Definition of Columns**

Columns in a database table define the specific attributes or fields that each record will contain. For example, in a table named "Users," a column called "username" would store text data representing each user's name.

**Data Types Importance**: Data types are crucial in a database as they dictate how data is stored, processed, and validated. Common data types include:

1. **Text**: Used for storing alphanumeric characters; examples include names or descriptions.
2. **Number**: Used for storing numerical values; examples include integers or decimal numbers representing quantities or prices.
3. **Date**: Used for storing date values; examples include timestamps for when records were created or modified.

**Question 2.3: Importance of Data Types**

Data types ensure data integrity by enforcing constraints on what kind of data can be stored in each column, preventing errors during data entry or processing. Here are three common types explained:

1. **Text**: This type allows for the storage of strings (e.g., names or addresses) and is essential for any non-numeric information.
2. **Number**: This type is used for calculations and quantitative analysis; it ensures that only numeric values are entered where necessary.
3. **Date**: This type is critical for tracking time-sensitive information; it allows for date comparisons and calculations (e.g., finding durations).

**Part 3: Expense Tracker Database Design**

**Question 3.1: Relevant Data Points.**

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

1. Expense ID (unique identifier)
2. Amount (the cost of the expense)
3. Date (when the expense occurred)
4. Category (the type of expense)
5. Description (details about the expense)

**Question 3.2: Basic Database Schema**

Here is a basic database schema for our "Expenses" table:

**TABLE NAME: EXPENSES**

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| expense\_id | INT |
| amount | DECIMAL |
| date | DATE |
| category | TEXT |
| description | TEXT |