**Question 1.1: Research**

In a dynamic website like an online store, SQL plays a crucial role in managing data behind the scenes by storing and retrieving information such as product details (name, price, description), user accounts (username, password, email), and order details (items purchased, quantities, delivery addresses).

**Question 1.2: SQL in Web Applications**

SQL (Structured Query Language) is used in web applications to interact with databases. It allows developers to efficiently create, read, update, and delete data (CRUD operations) from databases.

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

1. **Efficiency**: SQL queries are optimized for fast data retrieval and manipulation, enhancing web application performance.
2. **Data Organization**: SQL databases organize data into structured tables, enforcing data integrity through constraints like primary keys and foreign keys.
3. **Data Retrieval Capabilities**: SQL provides powerful querying capabilities (e.g., SELECT statements with filters and joins) that enable developers to retrieve complex datasets easily.

**Question 1.4: Benefits Explained**

* **Efficiency**: SQL databases are optimized for querying and indexing, which speeds up data retrieval, especially when dealing with large datasets.
* **Data Organization**: SQL enforces relational integrity, ensuring data relationships are maintained through constraints like primary and foreign keys.
* **Data Retrieval Capabilities**: SQL's querying language allows developers to perform complex queries across multiple tables, aggregating data as needed.

**Question 1.5: Database Management Systems**

Three popular Database Management Systems (DBMS) are:

1. MySQL
2. PostgreSQL
3. SQLite

**Part 2: Database Fundamentals**

**Question 2.1: Tables**

A database table is similar to a spreadsheet in that it organizes data into rows and columns. Each row represents a record, and each column represents a specific data attribute or field.

**Question 2.2: Columns**

Columns in a database table define the different types of data that can be stored. For example, a "name" column might store text data like "John Doe", while a "price" column might store numerical data like "25.99".

**Question 2.3: Data Types**

Data types in databases define the kind of data that can be stored in each column. They are important for ensuring data integrity and efficient storage.

* **Text**: Stores alphanumeric characters (e.g., VARCHAR).
* **Number**: Stores numeric values (e.g., INT for integers, DECIMAL for floating-point numbers).
* **Date**: Stores date and time values (e.g., DATE, DATETIME).

**Part 3: Expense Tracker Database Design**

**Question 3.1: Data Points for Expense Tracker**

1. **expense\_id**: Unique identifier for each expense record (INT).
2. **amount**: Amount spent (DECIMAL).
3. **date**: Date of the expense (DATE).
4. **category**: Category of the expense (TEXT).
5. **description**: Description of the expense (TEXT).

**Question 3.2: Database Schema**

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

**Bonus: Entity Relational Diagram (ERD)**

Visual representation (ERD) helps to illustrate relationships between tables, but for a single table like "Expenses," its straightforward:

SQL

Expenses Table

-----------------------------------

| expense\_id (INT) | amount (DECIMAL) | date (DATE) | category (TEXT) | description (TEXT) |

-----------------------------------

This basic structure outlines how expenses will be stored in the database, with each column defined by its data type to ensure efficient storage and retrieval.