**Introduction & Foundational Skills**

**Part 1: Understanding SQL**

1. **Research**

**SQL** - Structured Query Language: - is the standardized language for accessing and manipulating databases. It allows users to execute queries to retrieve, insert, update, and delete data within databases. SQL also supports creating new databases, defining tables, and setting permissions. Although SQL adheres to ANSI/ISO standards since 1986 and 1987 respectively, variations exist across implementations, ensuring compatibility with fundamental commands like SELECT, UPDATE, DELETE, INSERT, and WHERE across different platforms.

* 1. **Web Applications:  (Behind the scenes –{Back-End of the Website)**

**SQL** plays a crucial role in managing data behind the scenes of dynamic websites like online stores. It organizes product information such as name, description, price, and inventory levels in SQL tables. User accounts, which include usernames, hashed passwords for security, and personal details, are also efficiently managed using SQL databases. Furthermore, SQL stores order details such as customer information, purchased items, quantities, prices, and shipping details, enabling streamlined order processing, tracking, and customer service. This structured approach to data management ensures efficient querying, updating, and retrieval, maintaining smooth and secure website operations for both users and administrators.

* 1. **Role of SQL in Web Applications**

SQL (Structured Query Language) plays a pivotal role in web applications by providing a standardized language for managing relational databases. It enables developers to create, retrieve, update, and delete data efficiently, ensuring seamless interaction between the application and its underlying database. SQL's capabilities in handling complex queries and transactions (e.g., SELECT, UPDATE, DELETE, INSERT) facilitate dynamic content generation, user authentication, and data-driven decision-making processes essential for modern web applications. Its robustness and versatility make SQL indispensable for maintaining data integrity, optimizing performance, and enhancing overall user experience across various web platforms.

**1.3. 3 benefits of using SQL for web applications**

* Efficiency
* Data Integrity
* Scalability

**1.4. Explanation of Benefits**

* **Efficiency:** SQL optimizes query execution and resource utilization, ensuring fast data retrieval and processing, which is crucial for maintaining responsive web applications.
* **Data Organization:** SQL databases structure data into tables with defined relationships, promoting organized storage that facilitates easier data management, maintenance, and scalability as applications grow.
* **Data Retrieval Capabilities:** SQL's robust query language allows developers to construct complex queries to retrieve specific data subsets efficiently, supporting dynamic content generation and personalized user experiences on web platforms.

**1.5. Database Management Systems {DBMS**

1. MySQL
2. MongoDB
3. Microsoft SQL Server

**Part 2: Database Fundamentals**

**2.1. Tables**

Table is a structured collection of data organized into rows and columns, similar to a spreadsheet where rows represent individual records and columns represents a specific attribute or fields. This organization allows for efficient storage and retrieval of data, providing a systematic way to manage and manipulate information

**2.2. Columns**

Column is the structure of the data stored within it. Each column represents a specific attribute or field of the records stored in the table. For example, in a **"Students"** table, a column named **"StudentID”** stores unique numeric identifiers for each student. That ensures each **StudentID** is distinct and can be efficiently used for referencing and querying student records.

**Data Types:**

Data types are essential in databases because they ensure that data is stored in a structured format suitable for efficient storage and retrieval.

* **Text** data types like VARCHAR accommodate variable-length strings such as names or addresses.
* **Number** data types like INTEGER or DECIMAL handle numerical values for accurate calculations.
* **Date** data types like DATE or TIMESTAMP facilitate precise storage and manipulation of date and time information, crucial for chronological operations and temporal analysis within databases.

**2.3. Data Types**

**Importance of Data Types**

Data types ensure data integrity by defining the kind of data that can be stored, preventing errors and inconsistencies. They also optimize memory usage and processing speed, leading to more efficient storage and performance.

**Common Data Types**

* **Integer:** An integer is a whole number without a decimal point, used for counting and indexing. It ensures efficient storage and fast arithmetic operations.
* **String:** A string is a sequence of characters used for storing text, including letters, numbers, and symbols. It provides flexibility in handling various textual information.
* **Boolean:** A Boolean represents a binary value, either true or false. It is essential for logical operations and controlling program flow based on simple true/false conditions.

.

**Part 3: Expense Tracker Database Design**

**3.1. Planning**

Data points for Expense Tracker:

* **Amount:** The monetary value of the expense, inputted by the user as a number.
* **Category:** The type of expense selected from predefined options (e.g., food, transport, housing, entertainment, others).
* **Date of Expense:** The specific date when the expense occurred (not explicitly in the form, but should be included).
* **Description:** A brief description or note about the expense for additional context (not explicitly in the form, but typically necessary).
* **Payment Method:** The method used to pay for the expense, such as cash, credit card, or bank transfer (not explicitly in the form, but useful for tracking).

**3.2. Tables**

**Expenses**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| Id | INTEGER | A unique identifier for each expense (Primary Key) |
| Amount | DECIMAL | The monetary value of the expense |
| Date | DATE | The specific date when the expense occurred |
| Category | TEXT | The type of expense (e.g., food, transport) |
| Description | TEXT | A brief description or note about the expense |
| payment\_method | TEXT | The method used to pay for the expense |

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

