Remofilwe Semaushu

Introduction & Foundational Skills Assignment

Question 1: Research

* 1. There would be at least three tables of information, User Information table, Product Information table and an Order Information table.

1. User Information - Database Schema: User information is stored in tables containing fields such as userid, username, email, password, and other relevant details.
2. Product Information - Database Schema: Product details are stored in tables with fields such as productid, productname, productcost, and potentially other attributes like productdescription, productquantity, etc.
3. Order Information - Database Schema: Orders are managed through tables that include fields such as orderid, userid (for linking to users), productid (for linking to products), orderquantity, and other relevant details.
   1. The role of SQL in web applications is data management, data integrity and being able to retrieve data. The users will be able to perform CRUD operations (create, retrieve, update and delete data in the tables). To maintain data integrity, SQL has some constraints for data storage. There are primary keys and foreign keys; a primary key uniquely identifies each row in that table while a foreign key refers to the primary key in another table which establishes a link or relationship between the two tables. Data retrieval queries enable efficient retrieval of specific information such as user details, product information and order history.
   2. a) Efficiency: SQL allows for optimized data manipulation operations, ensuring fast response times for web applications. This efficiency is crucial for handling concurrent user requests and large datasets efficiently.

b) Data Organization: SQL provides a structured approach to data storage with tables, enforcing relationships between entities using primary and foreign keys. This organization ensures data integrity and simplifies maintenance and scaling of web applications.

c) Data Retrieval Capabilities: SQL offers powerful querying capabilities, including complex joins, aggregations, and filtering. This enables precise and flexible retrieval of data, supporting dynamic content generation and reporting in web application

1.4. a) Efficiency: SQL’s optimized queries and transaction management ensure rapid data retrieval and manipulation, enhancing overall application performance and user experience.

1. Data Organization: By structuring data into tables with defined relationships, SQL promotes systematic data management. This approach facilitates efficient data access, modification, and scalability as the application grows.
2. Data Retrieval Capabilities: SQL's rich query language allows developers to retrieve specific data subsets easily. This capability supports dynamic content generation, personalized user experiences, and robust reporting functionalities in web applications.
   1. a) MySQL: A widely-used open-source relational database management system known for its speed, reliability, and scalability. It is ideal for web applications requiring structured data management and ACID compliance.
3. PostgreSQL: An advanced open-source relational DBMS known for its extensibility, SQL compliance, and support for complex queries. It is favoured for applications needing strong data integrity, concurrency control, and compatibility with various platforms.
4. MongoDB: A leading NoSQL database management system that stores data in flexible, JSON-like documents. MongoDB excels in handling unstructured and semi-structured data, making it suitable for agile development, scalability, and real-time analytics in web applications.

Question 2: Database Fundamentals

2.1. A database table is similar to a spreadsheet in that both organize data into rows and columns. However, database tables are designed for storing large volumes of structured data efficiently, with features like indexing, data integrity constraints, and support for transactions, making them suitable for handling complex applications and simultaneous user interactions.

2.2. A vertical arrangement of data elements of a particular type within a table. Each column has a name and a data type that defines the kind of data it can store. In a users table, columns could include user\_id (an integer), username (text), email (text), and birthdate (date).

a) Text(VARCHAR/CHAR) – Stores variable-length strings of characters, such as names, addresses or descriptions. For example, VARCHAR(255) can store up to 255 characters of text.

b) Number(INTEGAR/DECIMAL) – Used for storing numeric data. INTEGER stores whole numbers while DECIMAL(precision, scale) stores fixed-point numbers with specific precision and scale. For example DECIMAL(5,2) can store numbers like 123.45 with 5 digits in total and 2 digits after the decimal point.

c) Date and Time(DATE/TIMESTAMP) – Used for storing date and time values. DATE stores dates (e.g., 2024-06-24), while TIMESTAMP stores both date and time (e.g., 2024-06-24 12:43:27). These data types are essential for tracking events, scheduling tasks, and managing temporal data in applications.

2.3. a) Enforcing Constraints: They restrict the type of data that can be stored in a column, preventing incompatible data from being inserted.

b) Optimizing Storage: Data types allocate appropriate storage space, minimizing storage requirements and improving performance.

c) Supporting Operations: Different data types support specific operations (e.g., arithmetic calculations for numbers, date manipulation for date types), enhancing the database's functionality and usability.

Brief Explanations of 3 Common Data Types:

1. VARCHAR (Variable-length character): Efficiently stores variable-length strings of characters, such as names or descriptions. It allows flexibility in data input and storage.
2. INTEGER (Integer): Stores whole numbers without decimal places, optimizing storage and enabling efficient arithmetic operations like addition and subtraction.
3. DATE (Date): Stores calendar dates (year, month, day) without time components. It's crucial for applications that need to track events, appointments, and scheduling tasks based on specific dates.

Question 3: Expense Tracker Database Design

3.1. a) Expense ID: A unique identifier for each expense record.

b) Product\_ID: A unique identifier to reference the products table which product the expense is associated with.

c) Amount: The numerical value of the expense.

d) Date: The date when the expense occurred.

e) Category: The category or type of expense (e.g., groceries, utilities, entertainment).

f) Payment Method: The method used to make the payment, limited to predefined options (e.g., cash, credit card, bank transfer).

3.2. Table Name: Expenses

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| expense\_id | INT | Primary key, unique identifier for each expense record. |
| product\_id | INT | Reference to the products table, indicating which product the expense is associated with. |
| amount | DECIMAL(10,2) | Numeric value representing the expense amount. Allows up to 10 digits with 2 decimal places for precision and to show monetary value. |
| date | DATE | Date of purchase. |
| category | VARCHAR(50) | Textual category or type of expense. |
| Payment\_method | ENUM('cash', 'credit card', 'bank transfer', ...) | Enumerated type restricting the payment method options. |

BONUS Question

|  |
| --- |
| Users |
| user\_id  username  email  password\_hash  registration\_date |

|  |
| --- |
| Orders |
| order\_id  user\_id  product\_id  quantity  order\_date |

|  |
| --- |
| Products |
| product\_id  product\_name  product\_description  price |

|  |
| --- |
| Expenses |
| expense\_id  product\_id  amount  date  category  payment\_method |