Part 1: Understanding SQL (30 minutes)

Question 1. Research

Use online resources like websites or PowerPoint slides.

1.1. In a single Word document, summarize your findings in a short paragraph (3-5 sentences). Web Applications:

Imagine a dynamic website like an online store. How do you think SQL plays a role in managing data behind the scenes? Consider how product information, user accounts, and order details might be stored and accessed.

SQL provides the backbone for managing data in an online store, ensuring that product catalogs are up-to-date, user accounts are secure and accessible, and orders are processed and tracked effectively.

1.2. Write a short explanation (3-5 sentences) in your document about the role of SQL in web applications.

SQL (Structured Query Language) plays a fundamental role in web applications by enabling efficient management and manipulation of data stored in databases. It allows web developers to create, retrieve, update, and delete data, essential for functionalities like user authentication, content management, transaction processing, and more. SQL ensures data integrity, scalability, and security, forming the backbone of dynamic web applications where real-time data interaction and reliability are critical.

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

Data Integrity: SQL provides mechanisms such as transactions and constraints (like foreign key relationships and unique constraints) that ensure data consistency and integrity. This reliability is crucial for maintaining accurate and trustworthy data in web applications.
Scalability: SQL databases are designed to handle large amounts of data and high concurrent access efficiently. They support indexing and query optimization techniques that allow applications to scale as user traffic and data volume increase.
Ease of Use and Standardization: SQL is a standardized language with a well-defined syntax and semantics. This standardization makes it easier for

developers to write and maintain database queries across different platforms and databases (like MySQL, PostgreSQL, SQL Server), promoting interoperability and reducing development time.

1.4. Think about efficiency, data organization, and data retrieval capabilities. Briefly explain each benefit in your document (1-2 sentences per benefit).

SQL offers efficient query execution and data manipulation, thanks to indexing and optimization techniques, ensuring rapid response times in web applications. It organizes data into structured tables with enforced integrity constraints, promoting consistency and reliability. SQL's powerful querying capabilities enable developers to retrieve and manipulate data flexibly, supporting dynamic content generation and enhancing user interaction on websites.

1.5. List any 3 Database Management Systems.

MySQL
PostgreSQL
Microsoft SQL Server.

Part 2: Database Fundamentals (45 minutes)

Question 2.1: Tables

Think about how data is organized in rows and columns. In your document, define a database table and explain its similarity to a spreadsheet (2-3 sentences).

A database table is a fundamental structure in a relational database management system (RDBMS) where data is organized into rows and columns. Each row represents a single record or entity, while each column represents a specific attribute or field of that record. This structure is similar to a spreadsheet where rows correspond to individual entries (e.g., products, users) and columns represent different characteristics (e.g., name, price, quantity). Both tables and spreadsheets facilitate structured data storage and retrieval, making them essential tools for organizing and managing information effectively.

Question 2.2: Columns

Consider different types of data like text, numbers, and dates. Define "columns" and provide an example with an explanation (2-3 sentences) in your document. Data Types: Why are data types important in a database? Briefly explain 3 common data types (e.g., Text, Number, Date).

Text (or VARCHAR): Used for storing alphanumeric characters such as names, addresses, or descriptions. Text data types can vary in length and are suitable for flexible content.
Number (or INTEGER, DECIMAL): Used for storing numeric values, including integers (whole numbers) and decimals (floating-point numbers). Numeric data types support mathematical operations and ensure precision in calculations.
Date (or DATE): Specifically designed for storing date and time values. Date data types facilitate chronological sorting, date arithmetic, and ensure that date-related operations (like comparisons and calculations) are accurate and efficient.

Question 2.3: Data Types

Think about how data types ensure data integrity and efficient storage. Explain the importance of data types and provide brief explanations of 3 common types (2-3 sentences each) in your document.

Data types in databases are crucial for maintaining data integrity and optimizing storage efficiency. They define the format and constraints for data stored in each column, ensuring accurate representation and efficient storage utilization. Common types like Text (VARCHAR), Numeric (INTEGER, DECIMAL), and Date and Time (DATE, TIMESTAMP) provide specific structures and validations that support accurate data storage, manipulation, and retrieval, essential for reliable database operations in various applications.

Part 3: Expense Tracker Database Design (45 minutes)

3.1. Planning: We'll be building an Expense Tracker application. What kind of data do you think we'll need to track? List at least 5 data points relevant to our project.

- Consider information like expense amount, date, and category.
- List your identified data points in your document.

1. Expense Category: Categories such as groceries, utilities, rent, entertainment, etc., to classify different types of expenses.

2. Expense Amount: The monetary value of each expense entry, specifying how much was spent.

3. Date of Expense: The date when the expense occurred, crucial for tracking spending patterns over time.

4. Payment Method: How the expense was paid, whether by cash, credit card, debit card, or other methods.

5. Notes/Description: Optional notes or descriptions explaining the purpose or details of the expense, providing context for each entry.

3.2. Tables: Considering the data points you listed, design a basic database schema with one main table (likely named "Expenses").

- Define the columns needed for this table.
- Assign appropriate data types to each column based on the kind of data it will hold. (e.g., amount: number, date: date, category: text)
  In your document, create a table structure that includes:
- Table name (e.g., Expenses)
- Column names (e.g., expense_id, amount, date, category)
- Data type for each column (e.g., INT, DECIMAL, DATE, TEXT)

expense_id: This column serves as the primary key, uniquely identifying each expense record. It is typically an integer (`INT`) and may auto-increment to ensure each entry has a unique identifier.
category: Stored as `VARCHAR`, this column holds textual data for categorizing expenses into different types such as groceries, utilities, etc.
amount: Defined as `DECIMAL`, it stores the monetary value of each expense, allowing for accurate calculations and storage of decimal values.
date: Using the `DATE` data type, this column captures the date when the expense was incurred, enabling date-based queries and analysis.
payment_method: Stored as `VARCHAR`, this column records how the expense was paid, offering flexibility to accommodate various payment methods.
notes: Utilizing the `TEXT` data type, this optional column allows users to add additional details or descriptions related to each expense entry.

Bonus:

Sketch a simple Entity Relational Diagram (ERD) of your table structure, including column names and data types.

Use drawing software or a simple table format to visually represent your schema.

** Remember: There might be multiple ways to design your database schema. The goal is to understand the concepts and create a logical structure to store our expense tracking data.

```
+-----------------+
|    Expenses     |
+-----------------+
| expense_id      | INT
| category        | VARCHAR
| amount          | DECIMAL
| date            | DATE
| payment_method  | VARCHAR
| notes           | TEXT
+-----------------+
```