 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.

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

The Role of SQL in Web Applications

SQL, or Structured Query Language, plays a vital role in web applications. It acts as a bridge between the web application and the database, allowing the application to store, retrieve, and manipulate data. This data can be anything from user information and product details to shopping cart contents and past orders. By efficiently managing this data, SQL helps web applications function smoothly and deliver a dynamic user experience.

Example: Managing Data in an Online Store

Let's consider an online store as an example. SQL plays a critical role in managing various types of data behind the scenes:

Product Information: SQL can be used to store product details like descriptions, prices, and images in a database. The web application can then query the database using SQL to display these details on product pages.

User Accounts: Customer information like names, addresses, and shopping preferences can be securely stored in an SQL database. The web application can use SQL to create new accounts, verify login credentials, and retrieve user data for personalized experiences.

Order Details: Order information like items purchased, quantities, and shipping addresses are stored in the database using SQL. The web application can use SQL queries to process orders, track order status, and generate invoices.

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

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

Benefits of Using SQL for Web Applications

Efficient Data Manipulation: SQL offers a structured and efficient way to manage data.  For web applications, this translates to fast retrieval, insertion, and modification of data within the database.  This efficiency is crucial for ensuring smooth operation and responsiveness, especially for applications handling large amounts of data or frequent user interactions.

Organized Data Structure:  SQL enforces a structured approach to data storage using tables and relationships. This organization makes data easier to understand, manage, and query.  For web applications, this translates to simpler development and maintenance, as developers can easily locate and manipulate specific data points.

Powerful Data Retrieval Capabilities:  SQL provides a robust set of commands for retrieving data from the database.  Web applications can leverage these commands to filter, sort, and search through vast amounts of data based on specific criteria.  This allows for functionalities like personalized product recommendations, targeted search results, and dynamic content generation.

**\*\*1.5.\*\*** List any 3 Database Management Systems.

MySQL: An open-source relational database management system (RDBMS) known for its ease of use, affordability, and widespread adoption.

PostgreSQL: Another open-source RDBMS offering a powerful feature set, strong data integrity, and scalability for larger applications.

Microsoft SQL Server: A commercial RDBMS popular in enterprise environments, known for its reliability, performance, and integration with other Microsoft products.

**##**

**## 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).

Database Table

A database table is a fundamental unit for organizing data in a relational database. It resembles a spreadsheet in its structure, with rows and columns. Each row, typically identified by a unique identifier, represents a single record or data point.  Columns, on the other hand, represent specific attributes or categories of data associated with each record. This structured organization allows for efficient storage, retrieval, and manipulation of data within the database.  Just like spreadsheets, database tables can be linked together to establish relationships between different data sets.

**\*\*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).

* Database Table

A database table is a fundamental unit for organizing data in a relational database. It resembles a spreadsheet in its structure, with rows and columns. Each row, typically identified by a unique identifier, represents a single record or data point. Columns, on the other hand, define specific characteristics or attributes associated with each record. These columns can hold different data types, such as text, numbers, and dates.  This structured organization allows for efficient storage, retrieval, and manipulation of data within the database.  Just like spreadsheets, database tables can be linked together to establish relationships between different data sets.

Columns

Each column in a database table represents a specific attribute or characteristic of the data being stored.  Think of a column like a category in a spreadsheet; it holds a particular type of information for each record in the table.  For instance, in a table storing information about products in an online store, you might have columns named "product\_id" (unique identifier for each product), "product\_name", "price", and "description".

Data Types: Why They Matter

Data types are essential in a database because they specify the format and range of values that a particular column can hold. This ensures data integrity, accuracy, and efficient storage utilization. Here are three common data types:

* Text: This data type stores textual information, such as product descriptions, names, or addresses.
* Number: This data type stores numeric values, including integers (whole numbers) and decimals. It can be further categorized based on precision and range to optimize storage for specific data like prices or product quantities.
* Date: This data type is used to store calendar dates and times. It allows for accurate representation and manipulation of date-related information, such as order timestamps or product launch dates.

**\*\*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: Why They Matter

Data types are essential in a database because they specify the format and range of values that a particular column can hold. This plays a crucial role in ensuring:

Data Integrity: By defining data types, the database enforces consistency and accuracy of the information being stored. For instance, a column designated for a product price wouldn't accept text characters, preventing invalid data entry.

Efficient Storage: Data types allow the database to optimize storage space by allocating only the necessary amount of memory based on the data type's characteristics. For example, a column storing product IDs (positive integers) would use less storage space compared to a column containing product descriptions (text).

Here are three common data types:

Text: This data type stores textual information, such as product descriptions, names, or addresses. Specifying a text data type ensures the column can hold varying lengths of characters, while also preventing the entry of incompatible data like numbers or symbols.

Number: This data type stores numeric values, including integers (whole numbers) and decimals. It can be further categorized based on precision and range to optimize storage for specific data like prices or product quantities. For instance, a price column might use a numeric data type with two decimal places to accommodate prices with cents.

Date: This data type is used to store calendar dates and times. It allows for accurate representation and manipulation of date-related information, such as order timestamps or product launch dates. Using a date data type ensures consistent formatting and simplifies tasks like filtering or sorting data based on dates.

**## 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 Amount: This is the core data point, representing the monetary value of the expense incurred. It should be a numeric data type to allow for calculations and aggregations.

2.Expense Date: Capturing the date of the expense is crucial for understanding spending patterns over time. A date data type will ensure accurate storage and manipulation of date information.

3.Expense Category: Categorizing expenses helps identify areas of spending. This could be a text data type allowing users to input categories like "groceries," "entertainment," or "transportation," or it could be a selection from a predefined list of categories within the application.

4.Payment Method: Knowing the payment method used (cash, debit card, credit card) can provide valuable insights into spending habits. This could be another text data type with options for selection or user input.

5.Optional: Description: A brief description of the expense can be helpful for adding context, especially for recurring or non-obvious expenses. This can be a text data type allowing users to enter additional details.

**\*\*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)

Absolutely! Here's the basic database schema with one main table named "Expenses" for our Expense Tracker application:

Table Name  Column Name Data Type Description

Expenses  expense\_id  INT Unique identifier (primary key) for each expense record.

Expenses  amount  DECIMAL Monetary value of the expense.

Expenses  date  DATE  Date on which the expense was incurred.

Expenses  category  TEXT  Category of the expense (e.g., groceries, entertainment, transportation).

Expenses (Optional) description TEXT  Optional field for a brief description of the expense.

Primary Key: The "expense\_id" column is designated as the primary key, ensuring each expense record has a unique identifier within the table. This allows for efficient data retrieval and manipulation.

Data Types:

INT: Used for "expense\_id" as it represents a unique whole number identifier.

DECIMAL: Used for "amount" to accommodate both whole numbers and decimal values for precise storage of expense amounts.

DATE: Used for "date" to ensure accurate storage and manipulation of date information.

TEXT: Used for "category" and (if included) "description" to allow flexibility in user input for expense categories and descriptions.