1.1

toring Data:

- **Product Information:** SQL can be used to create tables within a database to store product details. These tables might have columns for product ID (primary key), name, description, price, stock quantity, images, and any other relevant information.
- **User Accounts:** User accounts would likely be stored in another table with columns for user ID (primary key), username, password (hashed for security), email address, shipping address, and any other user-specific details.
- **Order Details:** Orders would likely be represented by another table with columns for order ID (primary key), user ID (foreign key referencing the Users table), product ID (foreign key referencing the Products table), quantity ordered, total price, order status (e.g., pending, shipped, delivered), and other relevant details like timestamps.

Accessing Data:

- When a user browses products, the website can use SQL queries to retrieve product information from the database based on certain criteria (e.g., category, price range).
- During user login, SQL queries can check the username and password against the Users table to verify login credentials.
- When a user places an order, SQL statements can insert new entries into the Orders table, capturing the order details. Additionally, SQL can update product stock quantities after an order is placed.

1.2

SQL acts as the backbone for many web applications. It facilitates communication between the web application and the database, where all the crucial information is stored. Using SQL, developers can create, update, and retrieve data efficiently. This data could be product information in an online store or user account details in a social media platform. Essentially, SQL empowers web applications to manage and utilize the data that drives their functionality.

1.3

1. Scalability: As a web application grows, the database needs to handle more data. SQL facilitates scaling the database to accommodate this growth by adding more storage or processing power. This ensures smooth performance even with increasing traffic or data volume.

2. Data Integrity and Security: SQL enforces data integrity through features like primary and foreign keys. This prevents inconsistencies and ensures data accuracy. Additionally, SQL allows for implementing user permissions and access controls, safeguarding sensitive information within the web application.

3.Structured Data Management: SQL allows for storing and retrieving data in a well-organized way using tables and defined data types. This makes it efficient to manage large amounts of information in a web application, like product catalogs or user accounts.

1.4

- **Efficiency:** SQL lets you write concise commands to retrieve specific data or update large datasets. This translates to faster loading times for web pages and quicker response to user actions, like adding items to a shopping cart or searching for products.
- **Data Organization:** SQL uses tables with defined structures to store data. This makes it easy to find and manage specific information. Imagine an online store; SQL keeps product details, user accounts, and order information organized and readily accessible.
- **Data Retrieval Capabilities:** SQL provides powerful tools for filtering and sorting data based on various criteria. This allows web applications to efficiently retrieve the exact information users need. For instance, a website can use SQL to find products matching a user's search terms or filter a news feed based on a user's preferences.

1.5

. MySQL

. Microsoft SQL sever

. PostgreSQL

PART 2

2.1

database table is essentially a giant spreadsheet where data is organized in rows and columns. Each row, also called a record or tuple, represents a single entity, like a customer or product. The columns, also known as fields or attributes, define specific details about that entity. This structure is like a spreadsheet where rows hold individual entries and columns define categories for that data, making it easy to add, edit, and retrieve information.

2.2

In a database table, columns represent specific categories of data for each record (row). Imagine a "Customers" table; one column might be named "Name" to store text data like customer names. Another column, "Age," could hold numeric data representing customer ages. This way, each column is dedicated to a particular type of information, keeping the data organized and easy to understand.

Data types are essential in a database because they define the format and allowed values for each column. This ensures data integrity and efficient storage. Here are 3 common data types:

2.3

Here's why they are important:

- **Data Integrity:** By specifying data types, you ensure that only the correct kind of information is entered into each column. For example, an "Age" column defined as a number would prevent someone from accidentally entering text like "twenty-five." This reduces errors and maintains the accuracy of your data.
- **Efficient Storage:** Data types allow the database to allocate the appropriate amount of space for each piece of information. Numbers require less storage than text, and dates have a specific format. This optimized storage saves space and improves overall database performance.

PART 3

1. Expense Amount

The actual cost of the expense, which can be a numerical value (e.g., 10.99, 500.00).

2. Date

The date when the expense was incurred, which can be a date object or a string in a specific format (e.g., YYYY-MM-DD, MM/DD/YYYY).

3. Category

The type of expense, such as Food, Transportation, Housing, Entertainment, etc. This can be a string or an enumeration.

4. Description

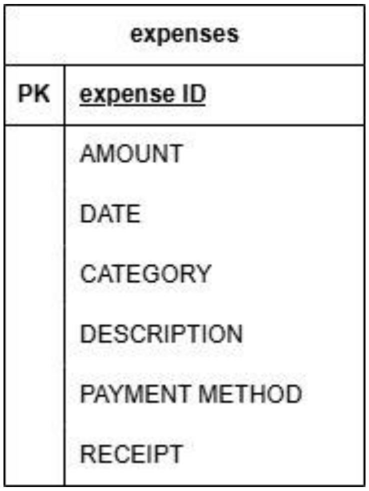
A brief description of the expense, which can be a string (e.g., "Lunch at McDonald's", "Rent for January").

5. Payment Method

The method used to pay for the expense, such as Cash, Credit Card, Debit Card, Online Transfer, etc. This can be a string or an enumeration.

3.2

Column Name	Data Type	Description
Expense ID	INT	Unique identifier for each expense
amount	DECIMAL(10, 2)	The actual cost of the expense
date	DATE	The date when the expense was incurred
category	TEXT	The type of expense (e.g., Food, Transportation)
description	TEXT	A brief description of the expense
payment method	TEXT	The method used to pay for the expense
created at	TIMESTAMP	Timestamp when the expense was recorded
updated at	TIMESTAMP	Timestamp when the expense was last updated



expenses	
PK	expense ID
	AMOUNT DATE CATEGORY DESCRIPTION PAYMENT METHOD RECEIPT