

### 1.1. Importance of SQL

SQL is important for web applications such as a shopping website. First, SQL can be used to **retrieve Specific Data** such as product information based on your search query.

Second, SQL can be used to **organise Data for Display** such as product listings by category or price for user convenience. Lastly, SQL can be used to **update Data** as Needed e.g when adding items to your shopping cart, SQL facilitates the process of updating your cart information within the database.

### 1.2. The Role of SQL in Web Applications

SQL is used for data manipulation to view, manage, and access data. It is also used to store, retrieve, and manipulate data in relational databases. It makes it easier for developers to test and manipulate data. It allows end-users to communicate with databases and perform tasks like creating, updating, and deleting databases.

### 1.3. Benefits of Using SQL for Web Applications

Benefits of SQL in web applications

- ➔ Efficiency and ability to update data.
- ➔ Data Organization
- ➔ Data Retrieval Capabilities

### 1.4. Explanation of Benefits

- **Efficiency:** SQL enables quick and efficient data management, allowing web applications to handle large volumes of data without compromising performance.
- **Data Organization:** SQL provides a structured way to organise data in tables, making it easy to maintain and update information systematically.
- **Data Retrieval Capabilities:** SQL's powerful querying capabilities allow for fast and precise data retrieval, ensuring users get the information they need promptly.

### 1.5 Database Management Systems

1. MySQL
2. PostgreSQL
3. Microsoft SQL Server

## 2.0. Database Tables

A database table is a collection of data organised in rows and columns whereby each row represents a single record, and each column represents a specific field within the record. For example, in a table of customers, each row might represent an individual customer, while columns might include the customer's phone number. Similarly, data in spreadsheets is also organised in rows and columns.

### 2.1. Columns

Columns in a database table define the type of data that can be stored in each cell of the column whereby each column has a name and a data type, indicating what kind of information it holds. For example, a column named "Age" might store numerical data, while a column named "Name" might store text data. Similarly, spreadsheets have columns containing data.

**Example:** In a "Students" table, columns might include "StudentID" (number)

### 2.2. Data Types

Data types are crucial in a database because they ensure data integrity and efficient storage in programming. They define what kind of data can be stored in each column, helping to prevent errors and optimise performance. Additionally, data types help in optimising memory usage by allocating the appropriate amount of memory to store different types of data.

1. **Text:** Stores strings of characters, such as names or descriptions. Text data stores non-numeric information.
  - **Example:** A column named "Name" with a text data type might store values like "Rose"
2. **Number:** Stores numerical values such as age and prices, which can be used for calculations and statistical analysis.

**Example:** A column named "Age" with a number data type might store values like 30 or 40.

3. **Date:** Stores dates and times or recording events, deadlines, and schedules, allowing for easy manipulation and comparison of temporal data.

**Example:** A column named "Registration Date" with a date data type might store values like "2024-07-03"

- **Integer (int):** This data type is used to store whole numbers without any decimal points. It is efficient in terms of memory usage and is commonly used for counting and indexing in programming.

### 3.1. Planning: Relevant Data Points

For our Expense Tracker application, we need to track the following data points:

1. **Amount:** The amount of money spent.
2. **Date:** The date when the expense occurred.
3. **Description:** A brief description of the expense.
4. **Method of payment** used e.g., Bank Transfer
5. **Store information** where the expense was made.

expense\_id: INT - A unique identifier for each expense entry. This could be set as a primary key.

amount: DECIMAL - The amount of money spent

date: DATE - The specific date when the expense occurred.

description: TEXT - Additional details or description of the expense.

payment\_method: TEXT - Indicates how the payment was made.

Expenses		
expense_id	INT	A unique identifier for each expense entry
amount	DECIMAL	The amount of money spent
date	DATE	The specific date when the expense occurred
description	TEXT	Additional details or description of the expense
payment_method	TEXT	Indicates how the payment was made.

