Session – 1

Introduction to SQL

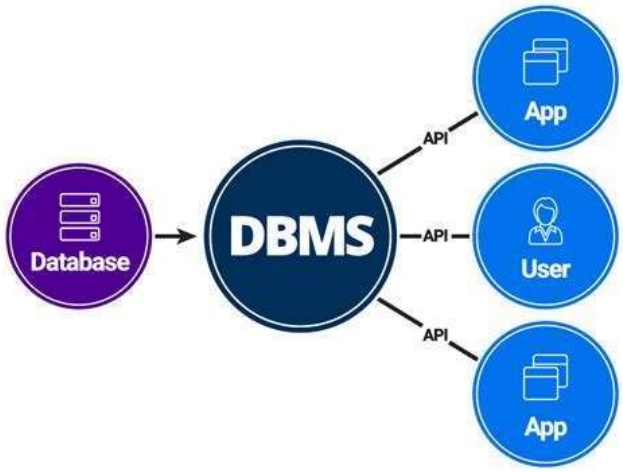
- **Objective:**
    - Familiarize yourself with the fundamentals of Database, DBMS and SQL command structure.
    - Learn how to import and create new tables in a database, understanding the importance of specifying appropriate data types and constraints for each column.
    - Learn how to insert data into newly created tables using the `INSERT INTO` statement, ensuring data integrity and accuracy.
    - Identify and understand common SQL clauses (SELECT, FROM, WERE, ORDER BY).

1. What is a database?

A database is like an organized collection of information stored on a computer. It's a structured way to store, manage, and retrieve data.

2. Database Management Systems (DBMS)

A **Database Management System (DBMS)** is software designed to store, retrieve, define, and manage data in a database. It serves as an interface between the database and its end-users or application programs, ensuring that data is consistently organized and remains easily accessible.



3. Introduction to SQL (Structured Query Language):

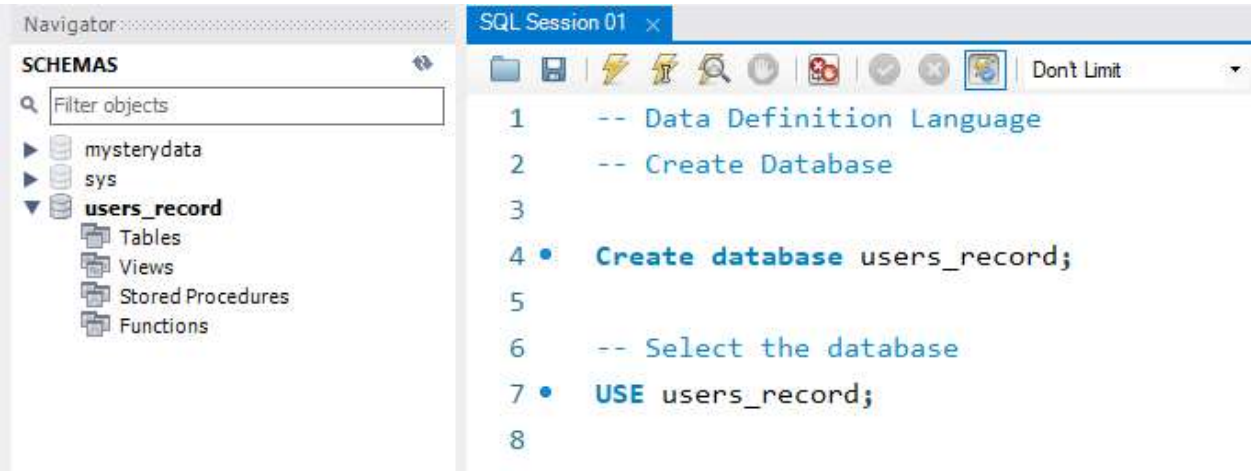
SQL is a language used to communicate with databases. It helps in performing various tasks like retrieving, updating, and managing data within a database.

Create a new Database

Creating a new database involves defining the database's name with the keyword `CREATE DATABASE` statement.

Syntax

```
CREATE DATABASE database_name;
```



Create a new Table

Creating a new table involves defining the table's structure with the `CREATE TABLE` statement. You specify the table name, column names, and data types for each column.

Syntax

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ...
);
```

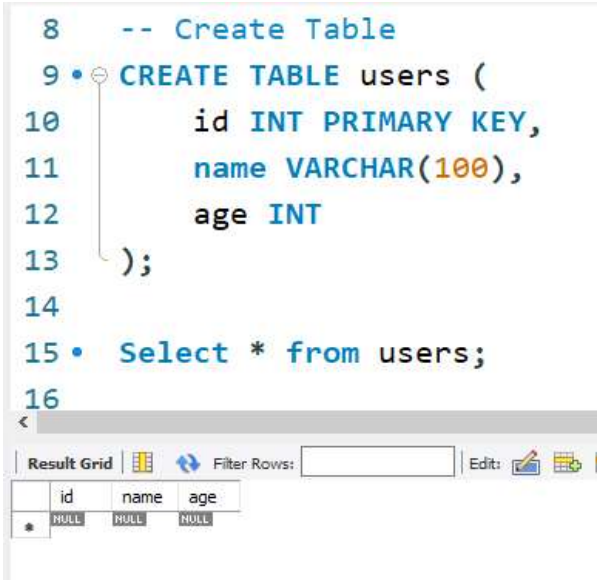
Common Data Types:

- **Numeric Types:** INT, FLOAT, DECIMAL
- **String Types:** VARCHAR, CHAR, TEXT
- **Date and Time Types:** DATE, TIME, DATETIME
- **Binary Types:** BLOB
- **Boolean Types:** BOOLEAN

Example

Let's create a table named `users` with columns for ID, name, and age.

```
-- Create Table
CREATE TABLE users (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    age INT
);
```



Inserting Values into New Tables

Once a table is created, you can insert data into it using the `INSERT INTO` statement.

Syntax

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

Example

Insert a new user record into the `users` table.

```
INSERT INTO users (id, name, age)
VALUES (1, 'Ahmad', 23);
```

Insert multiple new users records into the `users` table at once.

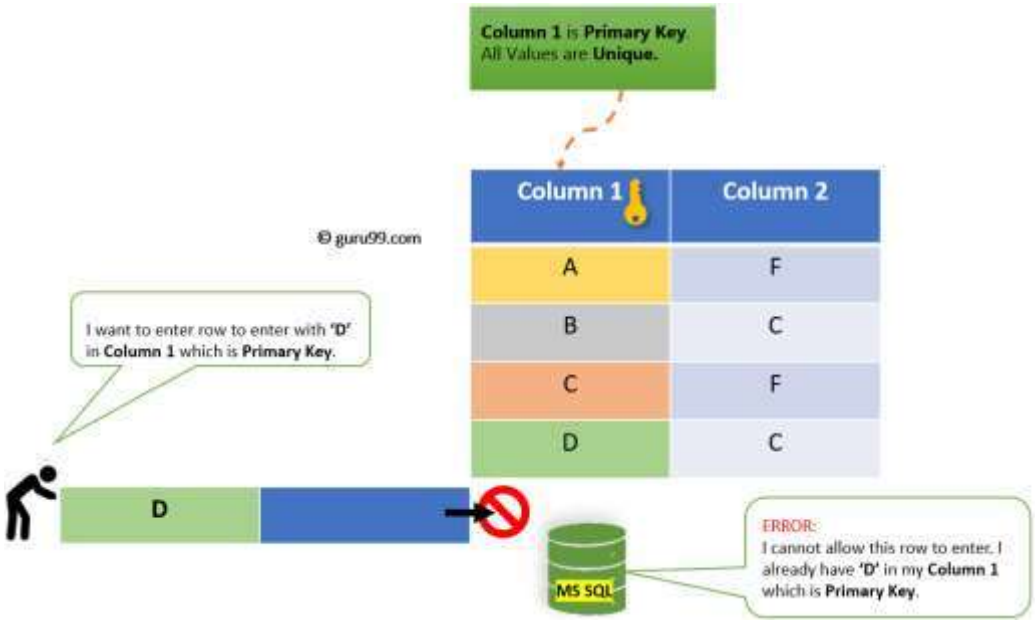
```
INSERT INTO users (id, name, age)
VALUES
(2, 'Ali', 22),
(3, 'Aliyan', 24),
(4, 'Ariyan', 28),
(5, 'Bilal', 23),
(6, 'Basit', 21);
```

```
17    -- Insert a new record
18 •  INSERT INTO users (id, name, age)
19    VALUES (1, 'Ahmad', 23);
20
21    -- Insert multiple rows at once
22 •  INSERT INTO users (id, name, age)
23    VALUES
24    (2, 'Ali', 22),
25    (3, 'Aliyan', 24),
26    (4, 'Ariyan', 28),
27    (5, 'Bilal', 23),
28    (6, 'Basit', 21);
29
30 •  select * from users;
```

Result Grid | Filter Rows: | Edit:

id	name	age
1	Ahmad	23
2	Ali	22
3	Aliyan	24
4	Ariyan	28
5	Bilal	23
6	Basit	21
NULL	NULL	NULL

Make sure to enter unique ID for users. Because ID is a primary key.



Data Retrieval

The SELECT statement is used to query the database and retrieve data.

Basic SELECT Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition
ORDER BY column1, column2, ... ASC|DESC;
```

Code Example:

```
SELECT name, age
FROM users
WHERE age > 18
ORDER BY age DESC;
```

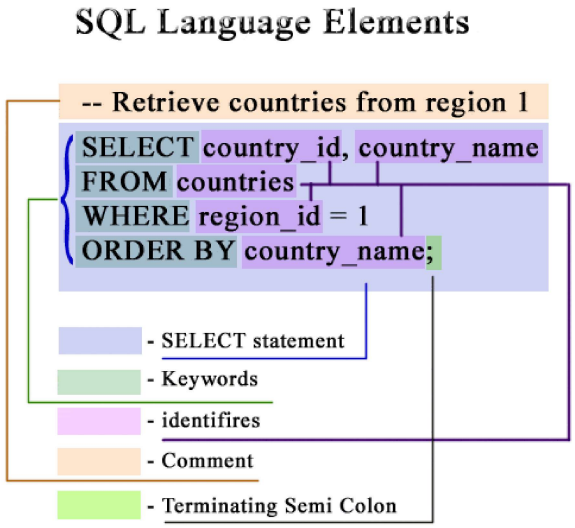
Common SQL Clauses:

- **SELECT:** Specifies the columns to be displayed.
- **FROM:** Specifies the table to retrieve data from.
- **WHERE:** Filters records based on a specified condition.
- **ORDER BY:** Sorts the result set in ascending or descending order.

Flow of SQL Commands

SQL commands are executed in a specific order, typically:

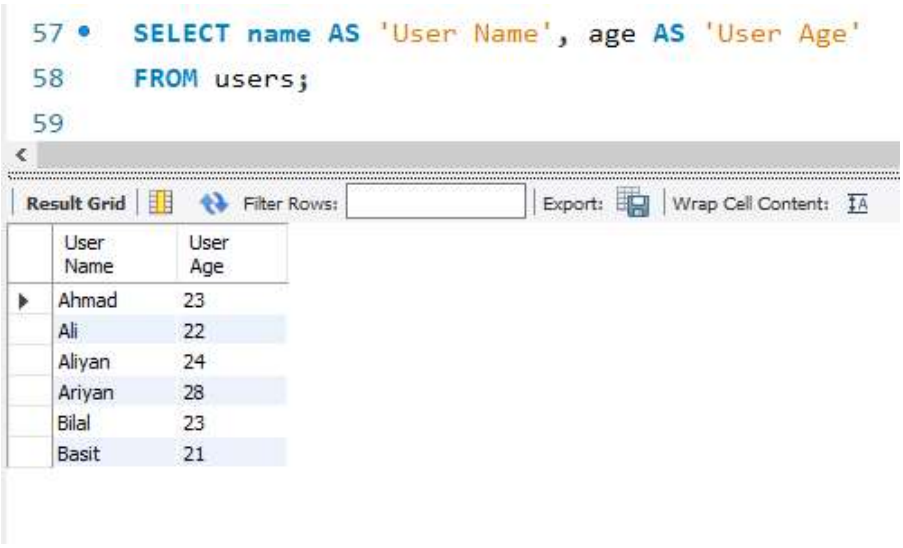
1. **FROM:** Identify the table(s).
2. **WHERE:** Filter the rows.
3. **SELECT:** Choose the columns.
4. **ORDER BY:** Sort the results.



Using Aliases to temporary Rename Columns:

Utilize column aliases for improved readability and clarity in results.

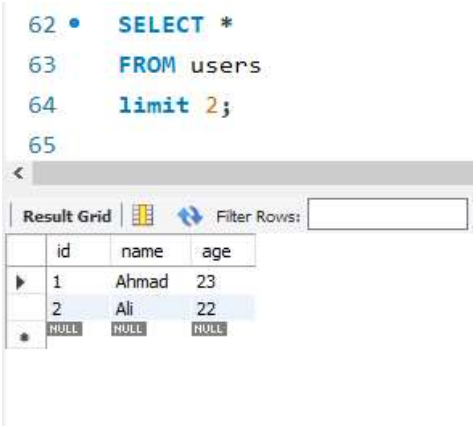
```
SELECT name AS 'User Name', age AS 'User Age'
FROM users;
```



Using Limit to Display Limited rows:

LIMIT clause: Sets a maximum number of results to be retrieved

```
SELECT *
FROM users
limit 2;
```



This manual provides a foundational understanding of SQL, covering essential topics from DBMS overview to executing and refining SQL queries. By achieving the outlined objectives, you'll build a solid foundation in SQL, enabling you to manage and manipulate data in relational databases effectively.