



### Learning Objectives:

- Understanding and Implementing SELECT statement
- Understanding wildcards, GROUP BY and ORDER BY

## 1. Introduction:

Today we will learn how we can display our tables using queries. The SELECT statement is used to select data from a database.

SELECT statement can be used in multiple ways, such as.

### Select all (\*):

To select and show all the columns in the table we use the SELECT statement.

#### Syntax:

```
Select * from TableName;                                -- Display all the contents of a table
```

```
use SE_21 -- mentioned to use a particular database  
SELECT * from Student
```

### Select column:

To select a column from the table, specify the correct column name.

#### Syntax:

```
Select columnName from TableName;                      -- Display a specific column of a table
```

```
SELECT age from Student
```

### Select multiple columns:

For selecting multiple columns from a table write names of all columns separated by comma (,)

#### Syntax:

```
Select column1, column2 from TableName;                 -- Display specific columns of a table
```

```
SELECT firstName, lastName from Student
```

## Select top:

If we want to show the top few values from a table, we use TOP command and a number, that is the rows you want to show, with SELECT command.

### Syntax:

```
Select TOP number columnName from TableName;           -- Display top rows of a table
```

```
SELECT TOP 5 age from Student -- Display top 5 rows of age column from Students table
```

## Select distinct:

To know the unique values in a column we use DISTINCT, with SELECT command.

### Syntax:

```
Select DISTINCT (columnName) from TableName;           -- Display all the unique values in a column
```

```
SELECT DISTINCT(AGE) from Student
```

## Select count:

If you want to know the number of entries in a row, use COUNT command.

### Syntax:

```
Select COUNT (columnName) from TableName;           -- Display number of entries in a column
```

```
SELECT COUNT(firstName) from Student
```

## Select max:

To obtain the maximum values from a column.

### Syntax:

```
Select MAX (columnName) from TableName;           -- Display maximum value of a column
```

```
SELECT MAX(age) from Student
```

## Select min:

To obtain the lowest values from a column.

### Syntax:

```
Select MIN (columnName) from TableName; -- Display minimum value of a column
```

```
SELECT MIN(age) from Student
```

## Select average:

To obtain the average of values from a column.

### Syntax:

```
Select AVERAGE (columnName) from TableName; -- Display average of a column
```

```
SELECT AVERAGE(age) from Student
```

*The functions COUNT, MAX, MIN, and AVERAGE are called **Aggregate Functions**.*

## Fully Qualified Names:

A fully qualified object name includes the names of all parent objects up to the level of the containing database. A common use case is a fully qualified column name, which consists of a **database name**, **table name**, and **column name**.

### Syntax:

```
Select * from DBN.dbo.TbN -- Display all values using fully qualified name
```

```
SELECT MAX(SE_21.dbo.STUDENT.age) from SE_21.dbo.Student
```

## Wildcard Characters:

A wildcard character is used to substitute one or more characters in a string.

Wildcard characters are used with the **LIKE** operator. The **LIKE** operator is used in a **WHERE** clause to search for a specified pattern in a column.

The variations of wildcards and their usages are mentioned in the table below

Symbol	Description	Example
%	Represents zero or more characters	<b>bl%</b> finds bl, black, blue, and blob
_	Represents a single character	<b>h_t</b> finds hot, hat, and hit
[]	Represents any single character within the brackets	<b>h[oa]t</b> finds hot and hat, but not hit
^	Represents any character not in the brackets	<b>h[^oa]t</b> finds hit, but not hot and hat
-	Represents any single character within the specified range	<b>c[a-b]t</b> finds cat and cbt

## Group By:

Group by is used to show all the unique values in a column.

### Syntax:

```
Select column1, column2 from tableName GROUP BY column1, column2,...
```

Now, you must be wondering if we already have DISTINCT statement, why are we using GROUP BY for the same purpose?

The answer is that the difference between GROUP BY and DISTINCT is that the later only tells you the unique values in a column, while the former not only tells you the unique values in the column but also tells the number of entries under each value.

### For example:

If we write this query,

```
SELECT Gender from Student GROUP BY Gender
```

It will return the same result as the DISCTINCT statement. But if we modify our query as

```
SELECT Gender, COUNT(Gender) from Student GROUP BY Gender
```

It will show how many numbers of entries are there in male and female category.

## Order By:

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

### Syntax:

```
Select column1, column2 from tableName ORDER BY column1, column2 ASC|DESC; -- Sort ascending/
descending
```

```
SELECT LastName, Age from Student ORDER BY Student ASC
```

This query will select the last names and ages of all the students in the ascending order of their ages.

## 2. Lab Tasks

Import Northwind database into SQL server and perform following operations

1. *Write a query to display all the entries of Customers whose names start with B and have l in it.*
2. *Write a query to retrieve top 5 records from Shippers table.*
3. *Write a query to count the number of employees, and their ID and filter out those single women who live in USA.*
4. *Write a query to get most expensive and least expensive Product list (name and unit price).*
5. *Write a query to get Product list (id, name, unit price) where **current** products cost less than \$20.*

## 3. HOME ASSIGNMENT:

Depending on different tables in the database of your management system, write SQL queries of all statements learnt in this lab.

