SQL Select Statement

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

Syntax:

```
Select column1, column2, . . . .
From table_name;
```

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

Select * From table_name;

Example 1:

Below is the selection of all data from "Student" table from a "University" database.



Example 2:

The following SQL statement selects the "First Name" and "Email" columns from the "Student" table:



SQL SELECT DISTINCT Statement

The SELECT DISTINCT statement is used to return only distinct (different) values.

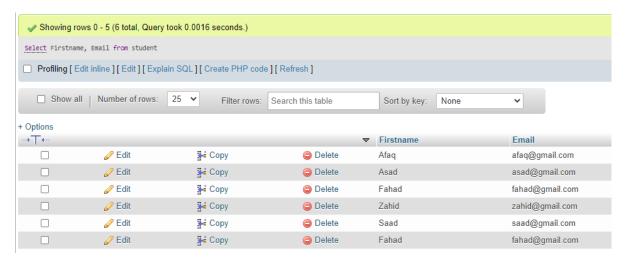
Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

Syntax:

SELECT DISTINCT column1, column2, ...
FROM table_name;

Example 1:

The following SQL statement selects all (including the duplicates) values from the "First Name" and "Email" column in the "Student" table:



Now, let us use the SELECT DISTINCT statement and see the result.

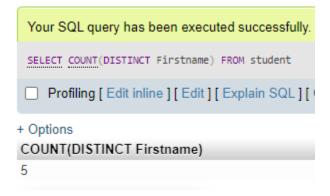
Example 2:

The following SQL statement selects only the DISTINCT values from the "First Name" and "Email" column in the "Student" table:



Example 3:

The following SQL statement lists the number of different (distinct) students First Names:



SQL Where

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Example:

The following SQL statement selects all the Students from the city "Lahore", in the "Student" table:



Text Fields vs. Numeric Fields

SQL requires single quotes around text values (most database systems will also allow double quotes).

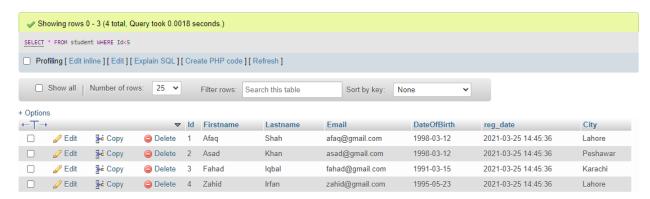
However, numeric fields should not be enclosed in quotes:



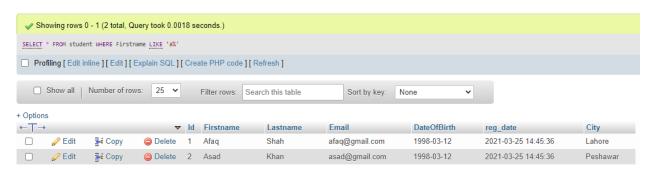
Operators in the WHERE Clause

S No.	Operator	Description
1	=	Equal
2	>	Greater than
3	<	Less than
4	>=	Greater than or equal
5	<=	Less than or equal
6	<>	Not equal. Note: In some versions of SQL this operator
		may be written as !=
7	BETWEEN	Between a certain range
8	LIKE	Search for a pattern
9	IN	To specify multiple possible values for a column

Less than Operator:



• LIKE Operator:



(Try more operators)

SQL AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.
- The NOT operator displays a record if the condition(s) is NOT TRUE.

Syntaxes:

AND: SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;

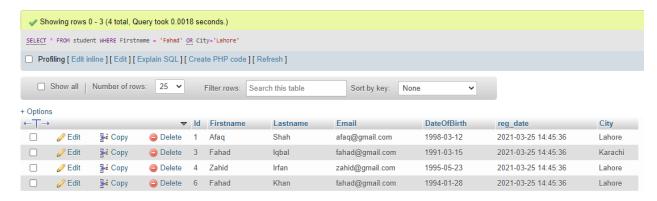
OR: SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;

NOT: SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;

Example 1(AND):



Example 2 (OR):



Example 3 (NOT):



SQL Order By Keyword

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

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

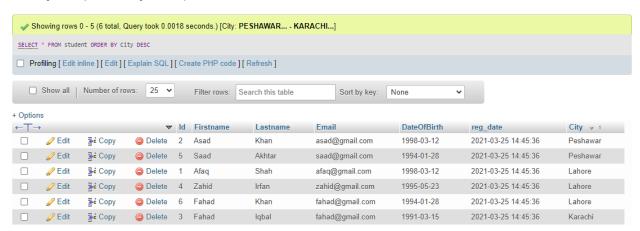
Syntax:

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

Example 1:



Example 2 (Order by DESC):



THE END