

SQL SELECT Statement

The SELECT statement is the most commonly used command in Structured Query Language. **It is used to access the records from one or more database tables and views.** It also retrieves the selected data that follow the conditions we want.

By using this command, we can also access the particular record from the particular column of the table. The table which stores the record returned by the SELECT statement is called a result-set table.

Syntax of SELECT Statement in SQL

1. **SELECT** Column_Name_1, Column_Name_2,, Column_Name_N **FROM** Table_Name;

In this SELECT syntax, **Column_Name_1, Column_Name_2,, Column_Name_N** are the name of those columns in the table whose data we want to read.

If you **want to access all rows from all fields of the table**, use the following SQL SELECT syntax with * asterisk sign:

1. **SELECT * FROM** table_name;

Examples of SELECT Statement in SQL

Here, we took the following two different SQL examples which will help you to execute the SELECT statement for retrieving the records:

Example 1:

Firstly, we have to create the new table and then insert some dummy records into it.

Use the following query to create the **Student_Records** table in SQL:

1. **CREATE TABLE** Student_Records
2. (
3. Student_Id **Int PRIMARY KEY**,
4. First_Name **VARCHAR** (20),
5. Address **VARCHAR** (20),

6. Age **Int** NOT NULL,
7. Percentage **Int** NOT NULL,
8. Grade **VARCHAR** (10)
9.);

The following query inserts the record of intelligent students into the **Student_Records** table:

1. **INSERT INTO** Student **VALUES** (201, Akash, Delhi, 18, 89, A2),
2. (202, Bhavesh, Kanpur, 19, 93, A1),
3. (203, Yash, Delhi, 20, 89, A2),
4. (204, Bhavna, Delhi, 19, 78, B1),
5. (05, Yatin, Lucknow, 20, 75, B1),
6. (206, Ishika, Ghaziabad, 19, 51, C1),
7. (207, Vivek, Goa, 20, 62, B2);

The following SQL query displays all the values of each column from the above **Student_records** table:

1. **SELECT** * **FROM** Student_Records;

The output of the above query is:

Student_ID	First_Name	Address	Age	Percentage	Grade
201	Akash	Delhi	18	89	A2
202	Bhavesh	Kanpur	19	93	A1
203	Yash	Delhi	20	89	A2
204	Bhavna	Delhi	19	78	B1
205	Yatin	Lucknow	20	75	B1
206	Ishika	Ghaziabad	19	91	C1
207	Vivek	Goa	20	80	B2

Example 2:

The following **query displays the values of particular column** from the above **Student_Record** table:

1. **SELECT** Student_Id, Age, Percentage, Grade **FROM** Employee;

Student_ID	Age	Percentage	Grade
201	18	89	A2
202	19	93	A1
203	20	89	A2
204	19	78	B1
205	20	75	B1
206	19	91	C1
207	20	80	B2

SELECT Statement with WHERE clause

The WHERE clause is used with SELECT statement **to return only those rows from the table, which satisfy the specified condition in the query.**

In SQL, the WHERE clause is not only used with SELECT, but it is also used with other SQL statements such as UPDATE, ALTER, and DELETE statements.

Syntax of SELECT Statement with WHERE clause

1. **SELECT * FROM** Name_of_Table **WHERE** [condition];

In the syntax, we specify the condition in the WHERE clause using SQL logical or comparison operators.

Example of SELECT Statement with WHERE clause

Firstly, we have to create the new table and then insert some dummy records into it.

Use the following query to create the **Employee_Details** table in SQL:

1. **CREATE TABLE** Employee_Details
2. (
3. Employee_ID **INT** AUTO_INCREMENT **PRIMARY KEY**,
4. Emp_Name **VARCHAR** (50),
5. Emp_City **VARCHAR** (20),
6. Emp_Salary **INT** NOT NULL,
7. Emp_Panelty **INT** NOT NULL
8.);

The following INSERT query inserts the record of employees into the Employee_Details table:

1. **INSERT INTO** Employee_Details (Employee_ID, Emp_Name, Emp_City, Emp_Salary, Emp_Panelty) **VALUES** (101, Anuj, Ghaziabad, 25000, 500),
2. (102, Tushar, Lucknow, 29000, 1000),
3. (103, Vivek, Kolkata, 35000, 500),
4. (104, Shivam, Goa, 22000, 500);

The following SELECT query shows the data of the **Employee_Details** table:

1. **SELECT * FROM** Employee_Details;

Employee_Id	Emp_Name	Emp_City	Emp_Salary	Emp_Panelty
101	Anuj	Ghaziabad	25000	500
102	Tushar	Lucknow	29000	1000
103	Vivek	Kolkata	35000	500
104	Shivam	Goa	22000	500

The following query shows the record of those employees from the above table whose Emp_Panelty is 500:

1. **SELECT * FROM** Employee_Details **WHERE** Emp_Panelty = 500;

This SELECT query displays the following table in result:

Employee_Id	Emp_Name	Emp_City	Emp_Salary	Emp_Panelty
101	Anuj	Ghaziabad	25000	500
103	Vivek	Kolkata	35000	500
104	Shivam	Goa	22000	500

SQL SELECT Statement with GROUP BY clause

The GROUP BY clause is used with the SELECT statement to show the common data of the column from the table:

Syntax of SELECT Statement with GROUP BY clause

1. **SELECT** column_Name_1, column_Name_2,, column_Name_N aggregate_function_name(column_Name2) **FROM** table_name **GROUP BY** column_Name1;

Example of SELECT Statement with GROUP BY clause

Use the following query to create the **Cars_Details** table:

1. **CREATE TABLE** Cars_Details
2. (
3. Car_Number **INT PRIMARY KEY**,
4. Car_Name **VARCHAR** (50),
5. Car_Price **INT** NOT NULL,
6. Car_Amount **INT** NOT NULL
7.);

The following INSERT query inserts the record of cars into the **Cars_Details** table:

1. **INSERT INTO** Cars_Details (Car_Number, Car_Name, Car_Amount, Car_Price)
2. **VALUES** (2578, Creta, 3, 1500000),
3. (9258, Audi, 2, 3000000),
4. (8233, Venue, 6, 900000),
5. (6214, Nexon, 7, 1000000);

The following SELECT query displays the values in the output:

1. **SELECT * FROM** Cars_Details;

Car_Number	Car_Name	Car_Amount	Car_Price
2578	Creta	3	1000000
9258	Audi	2	900000
8233	Venue	6	900000
6214	Nexon	7	1000000

The following SELECT with GROUP BY query lists the number of cars of the same price:

1. **SELECT COUNT** (Car_Name), Car_Price **FROM** Cars_Details **GROUP BY** Car_Price;

The output of above GROUP BY query is shown below:

Output:

Count (Car_Name)	Car_Price
2	1000000
2	900000

SQL SELECT Statement with HAVING clause

The HAVING clause in the SELECT statement creates a selection in those groups which are defined by the GROUP BY clause.

Syntax of SELECT Statement with HAVING clause

1. **SELECT** column_Name_1, column_Name_2,, column_Name_N aggregate_function_name(column_Name_2) **FROM** table_name **GROUP BY** column_Name1 **HAVING** ;

Example of SELECT Statement with HAVING clause

Let's create the **Employee_Having** table in SQL using the below CREATE command:

1. **CREATE TABLE** Employee_Having
2. (
3. Employee_Id **INT PRIMARY KEY**,
4. Employee_Name **VARCHAR** (50),
5. Employee_Salary **INT** NOT NULL,
6. Employee_City **VARCHAR** (50)
7.);

The following INSERT query inserts the record of employees into the Employee_Having table:

1. **INSERT INTO** Employee_Having (Employee_Id, Employee_Name, Employee_Salary, Employee_City)
2. **VALUES** (201, Jone, 20000, Goa),
3. (202, Basant, 40000, Delhi),
4. (203, Rashet, 80000, Jaipur),
5. (204, Aunj, 20000, Goa),
6. (205, Sumit, 50000, Delhi);

The following SELECT query shows the values of Employee_Having table in the output:

1. **SELECT * FROM** Employee_Having;

Employee_Id	Employee_Name	Employee_Salary	Employee_City
201	Jone	20000	Goa
202	Basant	40000	Delhi
203	Rashet	80000	Jaipur
204	Anuj	20000	Goa
205	Sumit	50000	Delhi

The following query shows the total salary of those employees having more than 5000 from the above Employee_Having table:

1. **SELECT SUM** (Employee_Salary), Employee_City **FROM** Employee_Having **GROUP BY** Employee_City **HAVING SUM**(Employee_Salary)>5000;

This HAVING query with SELECT statement shows the following table:

Output:

SUM (Employee_Salary)	Employee_City
90000	Delhi
80000	Jaipur

SELECT Statement with ORDER BY clause

The ORDER BY clause with the SQL SELECT statement shows the records or rows in a sorted manner.

The ORDER BY clause arranges the values in both ascending and descending order. Few database systems arrange the values of column in ascending order by default.

Syntax of SELECT Statement with ORDER BY clause

1. **SELECT** Column_Name_1, Column_Name_2,, column_Name_N **FROM** table_name **WHERE** [Condition] **ORDER BY**[column_Name_1, column_Name_2,, column_Name_N **asc** | **desc**];

Example of SELECT Statement with ORDER BY clause in SQL

1. **CREATE TABLE** Employee_Order
2. (
3. Id **INT** NOT NULL,
4. FirstName **VARCHAR** (50),
5. Salary **INT**,
6. City **VARCHAR** (50)
7.);
8. The following INSERT query inserts the record of employees into the Employee_Having table:
 1. **INSERT INTO** Employee_Order (Id, FirstName, Salary, City)
 2. **VALUES** (201, Jone, 20000, Goa),
 3. (202, Basant, 15000, Delhi),
 4. (203, Rashet, 80000,Jaipur),
 5. (204, Aunj, 90000, Goa),
 6. (205, Sumit, 50000, Delhi);

The following SELECT query shows the values of the table in the output:

1. **SELECT** * **FROM** Employee_Order;

Id	FirstName	Salary	City
201	Jone	20000	Goa
202	Basant	15000	Delhi
203	Rashet	80000	Jaipur
204	Anuj	90000	Goa
205	Sumit	50000	Delhi

The following query sorts the salary of employees in descending order from the above Employee_Order table:

1. **SELECT** * **FROM** Employee_Order **ORDER BY** Emp_Salary **DESC**;

This SQL query shows the following table in result:

Output:

Emp_Id	Emp_Name	Emp_Salary	Emp_City
204	Anuj	90000	Goa
203	Rashet	80000	Jaipur
205	Sumit	50000	Delhi
201	Jone	20000	Goa
202	Basant	15000	Delhi