

[Trending Now](#) [DSA](#) [Data Structures](#) [Algorithms](#) [Interview Preparation](#) [Data Science](#) [Topic-wise Practice](#) [J.](#)

Comparison Operators in SQL



abhisri459

[Read](#)[Discuss](#)[Courses](#)[Practice](#)

In SQL, there are six comparison operators available which help us run queries to perform various operations. We will use the [WHERE](#) command along with the [conditional operator](#) to achieve this in SQL. For this article, we will be using the Microsoft SQL Server as our database.

Syntax:

```
SELECT * FROM TABLE_NAME WHERE  
ATTRIBUTE CONDITION_OPERATOR GIVEN_VALUE;
```

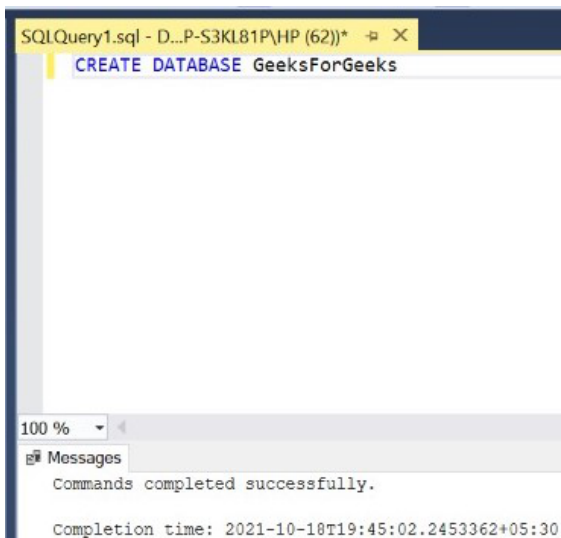
Step 1: Create a Database. For this use the below command to create a database named GeeksForGeeks.

Query:

AD

```
CREATE DATABASE GeeksForGeeks
```

Output:



The screenshot shows a SQL query window titled 'SQLQuery1.sql - D:\P-S3KL81P\HP (62))' with the command `CREATE DATABASE GeeksForGeeks` entered. Below the query window, the 'Messages' pane displays the output: 'Commands completed successfully.' and 'Completion time: 2021-10-18T19:45:02.2453362+05:30'.

Step 2: Use the GeeksForGeeks database. For this use the below command.

Query:

```
USE GeeksForGeeks
```

Output:



The screenshot shows a SQL query window titled 'SQLQuery1.sql - D:\P-S3KL81P\HP (62))' with the command `USE GeeksForGeeks` entered. Below the query window, the 'Messages' pane displays the output: 'Commands completed successfully.' and 'Completion time: 2021-10-18T19:45:33.4253364+05:30'.

Step 3: Create a table MATHS inside the database GeeksForGeeks. This table has 3 columns namely ROLL_NUMBER, S_NAME and MARKS containing roll number, student name, and marks obtained in math's subject by various students.

Query:

```
CREATE TABLE MATHS(  
  ROLL_NUMBER INT,  
  S_NAME VARCHAR(10),  
  MARKS INT);
```

Output:

SQLQuery1.sql - D:\P-S3KL81P\HP (60))*

```
CREATE TABLE MATHS(
  ROLL_NUMBER INT,
  S_NAME VARCHAR(10),
  MARKS INT);
```

100 %

Messages

Commands completed successfully.

Completion time: 2021-10-27T13:09:26.0163002+05:30

Step 4: Display the structure of the MATHS table.

Query:

```
EXEC SP_COLUMNS 'MATHS';
```

Output:

SQLQuery1.sql - D:\P-S3KL81P\HP (60))*

```
EXEC SP_COLUMNS 'MATHS';
```

100 %

Results Messages

	TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	PRECISION	LENGTH	SCALE	RADIX	NULLABLE	REMARKS	COLUMN_DEF
1	master	dbo	MATHS	ROLL_NUMBER	4	int	10	4	0	10	1	NULL	NULL
2	master	dbo	MATHS	S_NAME	12	varchar	10	10	NULL	NULL	1	NULL	NULL
3	master	dbo	MATHS	MARKS	4	int	10	4	0	10	1	NULL	NULL

Step 5: Insert 10 rows into the MATHS table.

Query:

```
INSERT INTO MATHS VALUES(1, 'ABHI', 70);
INSERT INTO MATHS VALUES(2, 'RAVI', 80);
INSERT INTO MATHS VALUES(3, 'ARJUN', 90);
INSERT INTO MATHS VALUES(4, 'SAM', 100);
INSERT INTO MATHS VALUES(5, 'MOHAN', 50);
INSERT INTO MATHS VALUES(6, 'ROHAN', 10);
INSERT INTO MATHS VALUES(7, 'ROCKY', 20);
INSERT INTO MATHS VALUES(8, 'AYUSH', 40);
```

```
INSERT INTO MATHS VALUES(9, 'NEHA', 30);
INSERT INTO MATHS VALUES(10, 'KRITI', 60);
```

Output:

SQLQuery1.sql - D:\P-S3KL81P\HP (60)*

```
INSERT INTO MATHS VALUES(1, 'ABHI', 70);
INSERT INTO MATHS VALUES(2, 'RAVI', 80);
INSERT INTO MATHS VALUES(3, 'ARJUN', 90);
INSERT INTO MATHS VALUES(4, 'SAM', 100);
INSERT INTO MATHS VALUES(5, 'MOHAN', 50);
INSERT INTO MATHS VALUES(6, 'ROHAN', 10);
INSERT INTO MATHS VALUES(7, 'ROCKY', 20);
INSERT INTO MATHS VALUES(8, 'AYUSH', 40);
INSERT INTO MATHS VALUES(9, 'NEHA', 30);
INSERT INTO MATHS VALUES(10, 'KRITI', 60);
```

100 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2021-10-27T13:33:16.0801864+05:30

Step 6: Display all the rows of the MATHS table.

Query:

```
SELECT * FROM MATHS;
```

Output:

SQLQuery1.sql - D:\P-S3KL81P\HP (60)*

```
SELECT * FROM MATHS;
```

100 %

Results Messages

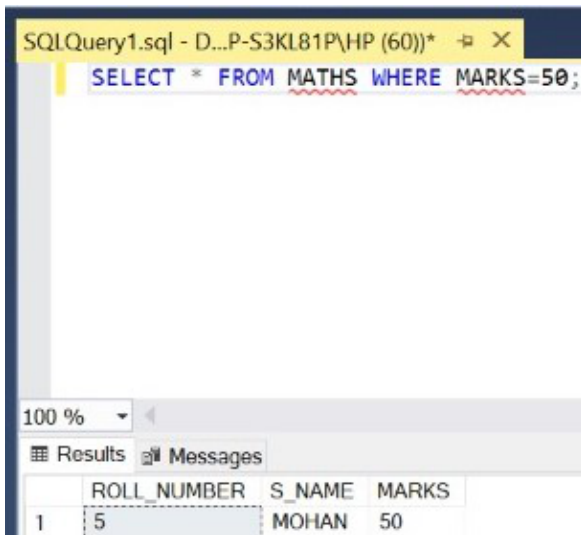
	ROLL_NUMBER	S_NAME	MARKS
1	1	ABHI	70
2	2	RAVI	80
3	3	ARJUN	90
4	4	SAM	100
5	5	MOHAN	50
6	6	ROHAN	10
7	7	ROCKY	20
8	8	AYUSH	40
9	9	NEHA	30
10	10	KRITI	60

Demonstration of various Comparison Operators in SQL:

- **Equal to (=) Operator:** It returns the rows/tuples which have the value of the attribute equal to the given value.

Query:

```
SELECT * FROM MATHS WHERE MARKS=50;
```

Output:

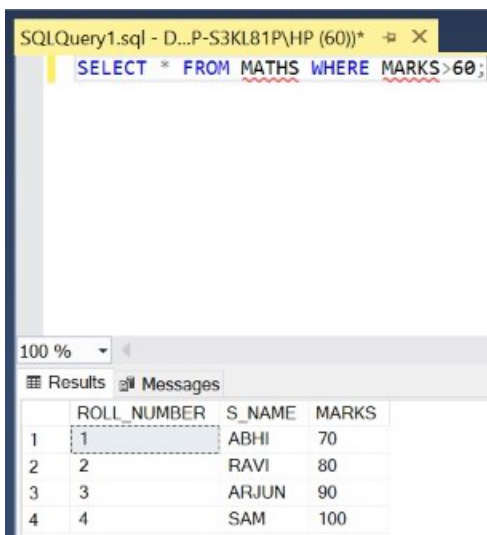
The screenshot shows a SQL query window with the text: `SELECT * FROM MATHS WHERE MARKS=50;`. Below the query, the results are displayed in a table with columns ROLL_NUMBER, S_NAME, and MARKS. The table contains one row with ROLL_NUMBER 5, S_NAME MOHAN, and MARKS 50.

	ROLL_NUMBER	S_NAME	MARKS
1	5	MOHAN	50

- **Greater than (>) Operator:** It returns the rows/tuples which have the value of the attribute greater than the given value.

Query:

```
SELECT * FROM MATHS WHERE MARKS>60;
```

Output:

The screenshot shows a SQL query window with the text: `SELECT * FROM MATHS WHERE MARKS>60;`. Below the query, the results are displayed in a table with columns ROLL_NUMBER, S_NAME, and MARKS. The table contains four rows with ROLL_NUMBER 1, 2, 3, and 4, S_NAME ABHI, RAVI, ARJUN, and SAM, and MARKS 70, 80, 90, and 100 respectively.

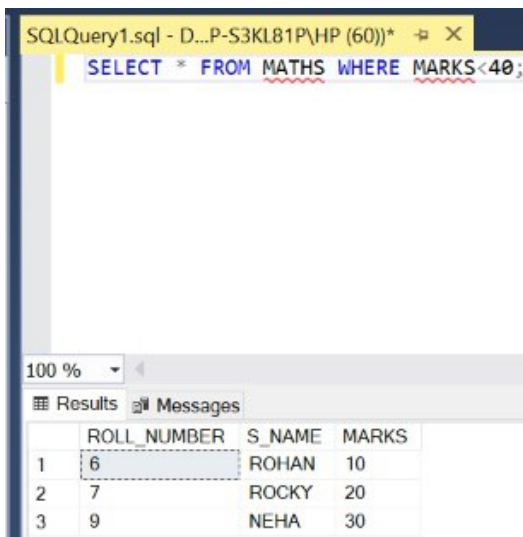
	ROLL_NUMBER	S_NAME	MARKS
1	1	ABHI	70
2	2	RAVI	80
3	3	ARJUN	90
4	4	SAM	100

- **Less than (<) Operator:** It returns the rows/tuples which have the value of the attribute lesser than the given value.

Query:

```
SELECT * FROM MATHS WHERE MARKS<40;
```

Output:



The screenshot shows a SQL query editor with the query: `SELECT * FROM MATHS WHERE MARKS<40;`. Below the editor, the 'Results' tab is active, displaying a table with 3 rows and 4 columns: ROLL_NUMBER, S_NAME, and MARKS. The results are as follows:

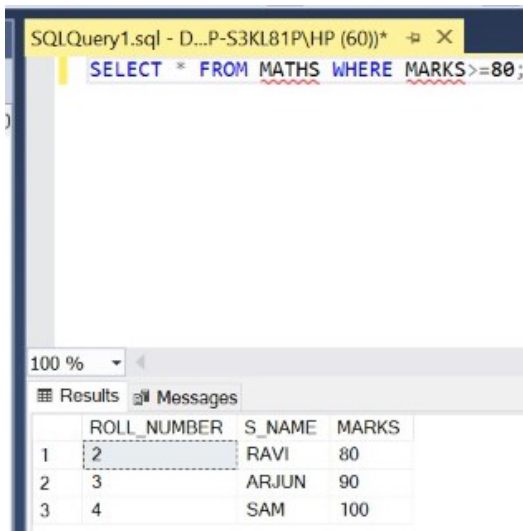
	ROLL_NUMBER	S_NAME	MARKS
1	6	ROHAN	10
2	7	ROCKY	20
3	9	NEHA	30

- **Greater than or equal to (>=) Operator:** It returns the rows/tuples which have the value of the attribute greater or equal to the given value.

Query:

```
SELECT * FROM MATHS WHERE MARKS>=80;
```

Output:



The screenshot shows a SQL query editor with the query: `SELECT * FROM MATHS WHERE MARKS>=80;`. Below the editor, the 'Results' tab is active, displaying a table with 3 rows and 4 columns: ROLL_NUMBER, S_NAME, and MARKS. The results are as follows:

	ROLL_NUMBER	S_NAME	MARKS
1	2	RAVI	80
2	3	ARJUN	90
3	4	SAM	100

- **Less than or equal to (<=) Operator:** It returns the rows/tuples which have the value of the attribute lesser or equal to the given value.

Query:

```
SELECT * FROM MATHS WHERE MARKS<=30;
```

Output:

SQLQuery1.sql - D:\P-S3KL81P\HP (60))*)

```
SELECT * FROM MATHS WHERE MARKS<=30;
```

100 %

Results Messages

	ROLL_NUMBER	S_NAME	MARKS
1	6	ROHAN	10
2	7	ROCKY	20
3	9	NEHA	30

- **Not equal to (<>) Operator:** It returns the rows/tuples which have the value of the attribute not equal to the given value.

Query:

```
SELECT * FROM MATHS WHERE MARKS<>70;
```

Output:

SQLQuery1.sql - D:\P-S3KL81P\HP (60))*)

```
SELECT * FROM MATHS WHERE MARKS<>70;
```

100 %

Results Messages

	ROLL_NUMBER	S_NAME	MARKS
1	2	RAVI	80
2	3	ARJUN	90
3	4	SAM	100
4	5	MOHAN	50
5	6	ROHAN	10
6	7	ROCKY	20
7	8	AYUSH	40
8	9	NEHA	30
9	10	KRITI	60

Last Updated : 14 Nov, 2021

2

Similar Reads

1. Difference between Structured Query Language (SQL) and Transact-SQL (T-SQL)
2. Configure SQL Jobs in SQL Server using T-SQL
3. SQL AND and OR operators
4. SQL | Wildcard operators