



SQL – ORDER BY

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

The ORDER BY statement in [SQL](#) is used to sort the fetched data in either ascending or descending according to one or more columns.

- By default ORDER BY sorts the data in ascending order.
- We can use the keyword DESC to sort the data in descending order and the keyword ASC to sort in ascending order.

Sort According To One Column

To sort in ascending or descending order we can use the keywords ASC or DESC respectively.

Syntax:

```
SELECT * FROM table_name ORDER BY column_name ASC | DESC
```

AD

//Where

table_name: name of the table.

column_name: name of the column according to which the data is needed to be arranged.

ASC: to sort the data in ascending order.

DESC: to sort the data in descending order.

|: use either ASC or DESC to sort in ascending or descending order//

Sort According To Multiple Columns

To sort in ascending or descending order we can use the keywords ASC or DESC respectively. To sort according to multiple columns, separate the names of columns by the (,) operator.

Syntax:

```
SELECT * FROM table_name ORDER BY column1 ASC/DESC , column2 ASC/DESC
```

Query:

```
CREATE TABLE students (  
    roll_no INT NOT NULL,  
    age INT NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    address VARCHAR(100) NOT NULL,  
    phone VARCHAR(20) NOT NULL,  
    PRIMARY KEY (roll_no)  
);  
INSERT INTO students (roll_no, age, name, address, phone)  
VALUES  
    (1, 18, 'Shubham Thakur', '123 Main St, Mumbai', '9876543210'),  
    (2, 19, 'Aman Chopra', '456 Park Ave, Delhi', '9876543211'),  
    (3, 20, 'Naveen Tulasi', '789 Broadway, Ahmedabad', '9876543212'),  
    (4, 21, 'Aditya arpan', '246 5th Ave, Kolkata', '9876543213'),  
    (5, 22, 'Nishant Jain', '369 3rd St, Bengaluru', '9876543214');
```

Output:

roll_no	age	name	address	phone
1	18	Shubham Thakur	123 Main St, Mumbai	9876543210
2	19	Aman Chopra	456 Park Ave, Delhi	9876543211
3	20	Naveen Tulasi	789 Broadway, Ahmedabad	9876543212
4	21	Aditya arpan	246 5th Ave, Kolkata	9876543213
5	22	Nishant Jain	369 3rd St, Bengaluru	9876543214

Now consider the above database table and find the results of different queries.

Sort According To a Single Column

In this example, we will fetch all data from the table Student and sort the result in descending order according to the column ROLL_NO.

Query:

```
SELECT * FROM students ORDER BY ROLL_NO DESC;
```

Output:

roll_no	age	name	address	phone
5	22	Nishant Jain	369 3rd St, Bengaluru	9876543214
4	21	Aditya arpan	246 5th Ave, Kolkata	9876543213
3	20	Naveen Tulasi	789 Broadway, Ahmedabad	9876543212
2	19	Aman Chopra	456 Park Ave, Delhi	9876543211
1	18	Shubham Thakur	123 Main St, Mumbai	9876543210

In the above example, if we want to sort in ascending order we have to use ASC in place of DESC.

Sort According To Multiple Columns

In this example, we will fetch all data from the table Student and then sort the result in ascending order first according to the column Age. and then in descending order according to the column ROLL_NO.

Query:

```
SELECT * FROM students ORDER BY Age ASC , ROLL_NO DESC;
```

Output:

roll_no	age	name	address	phone
1	18	Shubham Thakur	123 Main St, Mumbai	9876543210
2	19	Aman Chopra	456 Park Ave, Delhi	9876543211
3	20	Naveen Tulasi	789 Broadway, Ahmedabad	9876543212
4	21	Aditya arpan	246 5th Ave, Kolkata	9876543213
5	22	Nishant Jain	369 3rd St, Bengaluru	9876543214

In the above output, we can see that first the result is sorted in ascending order according to Age. There are multiple rows of having the same Age. Now, sorting further this result-set according to ROLL_NO will sort the rows with the same Age according to ROLL_NO in descending order.

Note:

ASC is the default value for the ORDER BY clause. So, if we don't specify anything after the column name in the ORDER BY clause, the output will be sorted in ascending order by default.

Take another example of the following query that will give similar output as the above:

Query:

```
SELECT * FROM students ORDER BY Age , ROLL_NO DESC;
```

Output:

roll_no	age	name	address	phone
1	18	Shubham Thakur	123 Main St, Mumbai	9876543210
2	19	Aman Chopra	456 Park Ave, Delhi	9876543211
3	20	Naveen Tulasi	789 Broadway, Ahmedabad	9876543212
4	21	Aditya arpan	246 5th Ave, Kolkata	9876543213
5	22	Nishant Jain	369 3rd St, Bengaluru	9876543214

Sorting By Column Number (instead of name)

An integer that identifies the number of the column in the SelectItems in the underlying query of the [SELECT statement](#). Column number must be greater than 0 and not greater than the number of columns in the result table. In other words, if we want to order by a column, that column must be specified in the SELECT list.

The rule checks for ORDER BY clauses that reference select list columns using the column number instead of the column name. The column numbers in the ORDER BY clause impair the readability of the SQL statement. Further, changing the order of columns in the SELECT list has no impact on the ORDER BY when the columns are referred to by names instead of numbers.

Syntax:

Order by Column_Number asc/desc

Here we take an example to sort a database table according to column 1 i.e Roll_Number. For this a query will be:

Query:

```
CREATE TABLE studentinfo
( Roll_no INT,
  NAME VARCHAR(25),
  Address VARCHAR(20),
  CONTACTNO BIGINT NOT NULL,
  Age INT );

INSERT INTO studentinfo
VALUES (7, 'ROHIT', 'GHAZIABAD', 9193458625, 18),
(4, 'DEEP', 'RAMNAGAR', 9193458546, 18),
(1, 'HARSH', 'DELHI', 9193342625, 18),
```

```
(8, 'NIRAJ', 'ALIPUR', 9193678625, 19),
(5, 'SAPTARHI', 'KOLKATA', 9193789625, 19),
(2, 'PRATIK', 'BIHAR', 9193457825, 19),
(6, 'DHANRAJ', 'BARABAJAR', 9193358625, 20),
(3, 'RIYANKA', 'SILIGURI', 9193218625, 20);
```

```
SELECT Name, Address
FROM studentinfo
ORDER BY 1
```

Output:

NAME	Address
DEEP	RAMNAGAR
DHANRAJ	BARABAJAR
HARSH	DELHI
NIRAJ	ALIPUR
PRATIK	BIHAR
RIYANKA	SILIGURI
ROHIT	GHAZIABAD
SAPTARHI	KOLKATA

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 vs NO SQL vs NEW SQL
4. SQL | Procedures in PL/SQL
5. SQL | Difference between functions and stored procedures in PL/SQL
6. SQL SERVER – Input and Output Parameter For Dynamic SQL
7. Difference between T-SQL and PL-SQL
8. Difference between SQL and T-SQL
9. SQL Server | Convert tables in T-SQL into XML