

ORDER BY

SQL ORDER BY clause sorts the result of the **SELECT** statement either in ascending or descending order.

In this article, we'll explore the **ORDER BY** clause, exploring its syntax, functionality, and usage with detailed examples.

ORDER BY in SQL

The **ORDER BY** statement in **SQL** is used to **sort the fetched data** in either ascending or descending according to one or more columns. It is very useful to present data in a structured manner.

SQL ORDER BY default mode is sorting data into ascending order. To sort data in descending order use the **DESC keyword** with **ORDER BY** clause.

Syntax

The syntax to use **ORDER BY** clause in **SQL** is:

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

Key Terms

- + **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.

SQL ORDER BY Clause Examples

Let's look at some examples of the **SQL ORDER BY** clause to understand it's working in **SQL**.

We will use the following table in examples.

roll_no	age	name	address	phone
1	18	Shubham Thakur	123 Main St, Mumbai	9876543210
2	18	Mohit Thakur	321 Main St, Mumbai	9876543201
3	19	Abhishek	567 New Way, Mumbai	9876543219
4	19	Aman Chopra	456 Park Ave, Delhi	9876543211
5	20	Naveen Tulasi	789 Broadway, Ahmedabad	9876543212
6	21	Aditya arpan	246 5th Ave, Kolkata	9876543213
7	22	Nishant Jain	369 3rd St, Bengaluru	9876543214

Student_Table

To create this table, write the following SQL queries:

```
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');
```

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

Sort According To a Single Column using ORDER BY Clause Example

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
7	22	Nishant Jain	369 3rd St, Bengaluru	9876543214
6	21	Aditya arpan	246 5th Ave, Kolkata	9876543213
5	20	Naveen Tulasi	789 Broadway, Ahmedabad	9876543212
4	19	Aman Chopra	456 Park Ave, Delhi	9876543211
3	19	Abhishek	567 New Way, Mumbai	9876543219
2	18	Mohit Thakur	321 Main St, Mumbai	9876543201
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 using ORDER BY Clause Example

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
```

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

Query

```
SELECT * FROM students ORDER BY age DESC , name ASC;
```

Output

roll_no	age	name	address	phone
7	22	Nishant Jain	369 3rd St, Bengaluru	9876543214
6	21	Aditya arpan	246 5th Ave, Kolkata	9876543213
5	20	Naveen Tulasi	789 Broadway, Ahmedabad	9876543212
3	19	Abhishek	567 New Way, Mumbai	9876543219
4	19	Aman Chopra	456 Park Ave, Delhi	9876543211
2	18	Mohit Thakur	321 Main St, Mumbai	9876543201
1	18	Shubham Thakur	123 Main St, Mumbai	9876543210

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

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

The Syntax to use ORDER BY Clause with Column Number

```
ORDER BY Column_Number ASC/DESC
```

Sorting By Column Number Example

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

Explanation

ORDER BY 1 means sorting values according to first column in the SELECT statement.

Important Points About ORDER BY Clause in SQL

- + The ORDER BY clause in SQL is used to sort the result set of a SELECT statement based on specified columns.
- + It is essential for organizing query results and presenting data in a structured manner.
- + It can sort data in either ascending (ASC) or descending (DESC) order.
- + Multiple columns can be specified for sorting, allowing for more complex sorting criteria.
- + We can use **ORDER BY** with **WHERE** clause, **GROUP BY** clause, and **HAVING** clause.