Introduction to the SQL Server ORDER BY clause

When you use the [SELECT](https://www.sqlservertutorial.net/sql-server-basics/sql-server-select/) statement to query data from a table, the order of rows in the result set is not guaranteed. It means that SQL Server can return a result set with an unspecified order of rows.

The only way for you to guarantee that the rows in the result set are sorted is to use theORDER BY clause. The following illustrates the ORDER BY clause syntax:

SELECT

select\_list

FROM

table\_name

ORDER BY

Code language: SQL (Structured Query Language) (sql)

[ASC | DESC ]

In this syntax:

column\_name | expression

First, you specify a column name or an expression on which to sort the result set of the query. If you specify multiple columns, the result set is sorted by the first column and then that sorted result set is sorted by the second column, and so on.

The columns that appear in the ORDER BY clause must correspond to either column in the select list or to columns defined in the table specified in the FROM clause.

ASC | DESC

Second, use ASC or DESC to specify the whether the values in the specified column should be sorted in ascending or descending order.

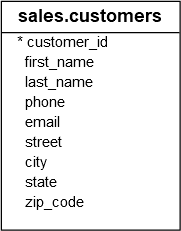
The ASC sorts the result from the lowest value to the highest value while the DESC sorts the result set from the highest value to the lowest one.

If you don’t explicitly specify ASC or DESC, SQL Server uses ASC as the default sort order. Also, SQL Server treats [NULL](https://www.sqlservertutorial.net/sql-server-basics/sql-server-null/) as the lowest values.

When processing the SELECT statement that has an ORDER BY clause, the ORDER BY clause is the very last clause to be processed.

SQL Server ORDER BY clause example

We will use the customers table in the [sample database](https://www.sqlservertutorial.net/sql-server-sample-database/) from the demonstration.



A) Sort a result set by one column in ascending order

The following statement sorts the customer list by the first name in ascending order:

SELECT

first\_name,

last\_name

FROM

sales.customers

ORDER BY

first\_name;

Code language: SQL (Structured Query Language) (sql)



In this example, because we did not specify ASC or DESC, the ORDER BY clause used ASC by default.

B) Sort a result set by one column in descending order

The following statement sorts the customer list by the first name in descending order.

SELECT

firstname,

lastname

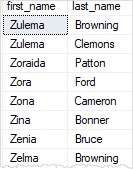
FROM

sales.customers

ORDER BY

first\_name DESC;

Code language: SQL (Structured Query Language) (sql)



In this example, because we specified the DESC explicitly, the ORDER BY clause sorted the result set by values in the first\_name column in the descending order.

C) Sort a result set by multiple columns

The following statement retrieves the first name, last name, and city of the customers. It sorts the customer list by the city first and then by the first name.

SELECT

city,

first\_name,

last\_name

FROM

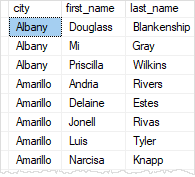
sales.customers

ORDER BY

city,

first\_name;

Code language: SQL (Structured Query Language) (sql)



D) Sort a result set by multiple columns and different orders

The following statement sorts the customers by the city in descending order and the sort the sorted result set by the first name in ascending order.

SELECT

city,

first\_name,

last\_name

FROM

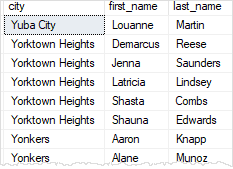
sales.customers

ORDER BY

city DESC,

first\_name ASC;

Code language: SQL (Structured Query Language) (sql)



E) Sort a result set by a column that is not in the select list

It is possible to sort the result set by a column that does not appear on the select list. For example, the following statement sorts the customer by the state even though the state column does not appear on the select list.

SELECT

city,

first\_name,

last\_name

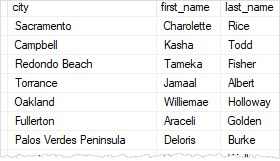
FROM

sales.customers

ORDER BY

state;

Code language: SQL (Structured Query Language) (sql)



Note that the state column is defined in the  customers table. If it was not, then you would have an invalid query.

F) Sort a result set by an expression

The [LEN()](https://www.sqlservertutorial.net/sql-server-string-functions/sql-server-len-function/) function returns the number of characters of a string. The following statement uses the [LEN()](https://www.sqlservertutorial.net/sql-server-string-functions/sql-server-len-function/) function in the ORDER BY clause to retrieve a customer list sorted by the length of the first name.

SELECT

first\_name,

last\_name

FROM

sales.customers

ORDER BY

LEN(first\_name) DESC;

Code language: SQL (Structured Query Language) (sql)



G) Sort by ordinal positions of columns

SQL Server allows you to sort the result set based on the ordinal positions of columns that appear in the select list.

The following statement sorts the customers by first name and last name. But instead of specifying the column names explicitly, it uses the ordinal positions of the columns:

SELECT

first\_name,

last\_name

FROM

sales.customers

ORDER BY

1,

2;

Code language: SQL (Structured Query Language) (sql)

In this example, 1 means the first\_name column and 2 means the last\_name column.

Using the ordinal positions of columns in the ORDER BY clause is considered as bad programming practice for a couple of reasons.

* First, the columns in a table don’t have ordinal positions and need to be referenced by name.
* Second, when you modify the select list, you may forget to make the corresponding changes in the ORDER BY clause.

Therefore, it is a good practice to always specify the column names explicitly in the ORDER BY clause.