

Lesson 7: Sort with ORDER BY and export query results

note

The order of statements

SQL statements have to be written in a specific order. SELECT usually comes first, followed by FROM.

You haven't seen every statement available yet, but here is the order that they should be written:

```
USE
SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY
```

note

Default sorting

If you don't supply an *ORDER BY* statement, the query results generally show the records in the order in which they were added to the database.

It's possible for records to be updated, deleted and replaced at any time so you shouldn't rely on this to show records in order of date.

In this lesson you'll learn how to sort the results of queries and how to export query results as files that can be sent to other users.

- 1 Open SSMS and connect to your SQL Server (if you haven't already done this).
- 2 Write a query to extract all records from the *Orders* table.
You saw how to do this in *Lesson 5: Write your first database query*. The query should be:

```
SELECT *
FROM Northwind.dbo.Orders
```

- 3 Execute the query.

All records in the *Orders* table are displayed. Notice that they are not shown in any particular order.

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
1	10248	VINET	5	1996-07-04 00:00:00.000	1996-08-01 00:00:00.000
2	10249	TOMSP	6	1996-07-05 00:00:00.000	1996-08-16 00:00:00.000
3	10250	HANAR	4	1996-07-08 00:00:00.000	1996-08-05 00:00:00.000
4	10251	VICTE	3	1996-07-08 00:00:00.000	1996-08-05 00:00:00.000
5	10252	SUPRD	4	1996-07-09 00:00:00.000	1996-08-06 00:00:00.000

- 4 Modify the query to sort the results by *CustomerID*.

The ORDER BY statement lets you sort your query results.

SQL statements must be written in the correct order. ORDER BY must be placed after the SELECT and FROM statements.

Add an ORDER BY statement to the query as shown:

```
SELECT *
FROM Northwind.dbo.Orders
ORDER BY CustomerID
```

- 5 Execute the query.

The results are now sorted by *CustomerID*, in ascending alphabetical order. This is the default sort order.

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
1	10643	ALFKI	6	1997-08-25 00:00:00.000	1997-09-22 00:00:00.000
2	10692	ALFKI	4	1997-10-03 00:00:00.000	1997-10-31 00:00:00.000
3	10702	ALFKI	4	1997-10-13 00:00:00.000	1997-11-24 00:00:00.000
4	10835	ALFKI	1	1998-01-15 00:00:00.000	1998-02-12 00:00:00.000
5	10952	ALFKI	1	1998-03-16 00:00:00.000	1998-04-27 00:00:00.000

- 6 Modify the query to sort by first by *CustomerID* and then by *OrderDate*.

The results are sorted by customer ID, but they are shown in no particular date order. Modify the query as follows:

note

Syntactic sugar

The SQL language often allows commands to be entered in multiple ways, usually offering a more verbose alternative that is intended to make your code easier to read. This is known as “syntactic sugar”.

In this lesson you use the *ASC* and *DESC* keywords to control sort order, but they can also be written as *ASCENDING* and *DESCENDING* to make their purpose clearer.

This course generally uses the shorter forms of commands.

note

CSV files

CSV stands for “Comma Separated Values”. A CSV file contains data separated by commas. For example:

```
11011,ALFKI,3,
1998-04-09,1998-05-07
```

CSV files can be opened by Microsoft Excel and can be opened and imported by many other programs.

note

Copying and pasting

As well as saving the query results as a file, you can also copy results directly from the *Results* pane and paste them into another program, usually an Excel spreadsheet. This is often the quickest way to extract the results of a query.

You can copy query results by selecting them in the *Results* pane and clicking *Edit* → *Copy* or using the <Ctrl>+<C> keyboard shortcut.

```
SELECT *
FROM Northwind.dbo.Orders
ORDER BY CustomerID, OrderDate
```

Just like the *SELECT* statement, you can sort by as many columns as you need by separating them with commas.

The order of the columns is important. This *ORDER BY* statement will first sort the results by *CustomerID* and then by *OrderDate*.

7 Execute the query.

The results are now sorted by customer ID and order date, both in ascending order.

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
1	10643	ALFKI	6	1997-08-25 00:00:00.000	1997-09-22 00:00:00.000
2	10692	ALFKI	4	1997-10-03 00:00:00.000	1997-10-31 00:00:00.000
3	10702	ALFKI	4	1997-10-13 00:00:00.000	1997-11-24 00:00:00.000
4	10835	ALFKI	1	1998-01-15 00:00:00.000	1998-02-12 00:00:00.000
5	10952	ALFKI	1	1998-03-16 00:00:00.000	1998-04-27 00:00:00.000

8 Modify the query to sort the dates in descending order (most recent first).

ORDER BY allows you to use the keywords *ASCENDING* and *DESCENDING* to control the sort order. You can shorten these to *ASC* and *DESC*.

Modify the query to:

```
SELECT *
FROM Northwind.dbo.Orders
ORDER BY CustomerID, OrderDate DESC
```

9 Execute the query.

The results are now sorted by *CustomerID* in ascending order and by *OrderDate* in descending order.

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
1	11011	ALFKI	3	1998-04-09 00:00:00.000	1998-05-07 00:00:00.000
2	10952	ALFKI	1	1998-03-16 00:00:00.000	1998-04-27 00:00:00.000
3	10835	ALFKI	1	1998-01-15 00:00:00.000	1998-02-12 00:00:00.000
4	10702	ALFKI	4	1997-10-13 00:00:00.000	1997-11-24 00:00:00.000
5	10692	ALFKI	4	1997-10-03 00:00:00.000	1997-10-31 00:00:00.000

10 Export the query results as a CSV file.

You can export the results of a query as a file, making it easy to distribute query results.

1. Right click in the query results and click *Save Results As* from the shortcut menu.
2. Save the results as: **SortedOrders.csv**

The query results are saved as a CSV file, which can be opened in Microsoft Excel and many other programs.