

Copyright © 2014 by Software Craftsmanship Guild.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the Software Craftsmanship Guild. For permission requests, write to the Software Craftsmanship Guild, addressed “Attention: Permissions Coordinator,” at the address below.

Software Craftsmanship Guild

526 S. Main St, Suite 609

Akron, OH 44311

Sorting Data

Software Craftsmanship Guild

Lesson Goals

- Learn how to sort data

SQL Server can Sort Data

- In SQL Server we can sort data in ascending or descending order across one or many columns
- It is important to note that sorting does not impact the physical order on the disk, only the order in the current result set.
- The default for sorting is ascending order, so we don't need to specify ascending if we choose not to.

Using Order By

- The Order By statement is the last statement in a SQL Query. Try the following in SWCCorp:

```
SELECT LastName, FirstName  
FROM Employee  
ORDER BY LastName;
```

```
SELECT LastName, FirstName  
FROM Employee  
ORDER BY LastName DESC
```

Multiple Table Queries

- These work the same way, you can sort on any column in any table from the query *regardless of whether it is in the select statement.*

```
SELECT LastName, FirstName, HireDate, City
FROM Employee
    INNER JOIN Location ON Employee.LocationID = Location.LocationID
ORDER BY HireDate;
```

```
SELECT LastName, FirstName
FROM Employee
    INNER JOIN Location ON Employee.LocationID = Location.LocationID
ORDER BY City;
```

Sorting Multiple Columns

- We can comma separate multiple columns in an order by statement. SQL Server will do the ordering from left to right.

```
-- tenure by city
SELECT LastName, FirstName, HireDate, City
FROM Employee
    INNER JOIN Location ON Employee.LocationID = Location.LocationID
ORDER BY City, HireDate;
```

Sorting Data With Nulls

- Nulls, being unknown, will appear first in ascending order and last in descending order
- Use a WHERE <column> IS NOT NULL to remove nulls if necessary

```
SELECT *  
FROM Employee  
ORDER BY [Status]
```

```
SELECT *  
FROM Employee  
ORDER BY [Status] DESC
```


Example: Removing Nulls For Ordering

```
-- Highest Paid Salary Employees
SELECT FirstName, LastName, HireDate, YearlySalary
FROM Employee e
      INNER JOIN PayRates pr ON e.EmpID = pr.EmpID
WHERE YearlySalary IS NOT NULL
ORDER BY YearlySalary DESC
```

Lab Time!

1. Show all the records from the Grant Table sorted alphabetically by Grant Name
2. Show all the employees in the Employee table from newest hire to oldest
3. Query the Current Products table for just the ProductName and Category fields from most expensive to least expensive retail price
4. Sort the grant table from highest to lowest amount, if there is a tie, then sort the ties alphabetically by grant name
5. Join the Employee and Location tables using an Outer Join that shows all employee records even if they have no location, show the FirstName, LastName, and City such that NULL cities show up first.

Things to Remember

- Use ORDER BY to sort query results
- Use DESC or ASC after column names to sort descending or ascending
 - The default is ASC if not specified
- You can sort more than one field, it will evaluate left to right and this is useful for tie breakers
- You can sort on fields not in your SELECT list.
- NULLs will always be first or last depending on the sort direction
- If no sort is specified, the results will be in the *natural order* which is typically the primary key

Fin

- Next up: Table Relationships In Depth