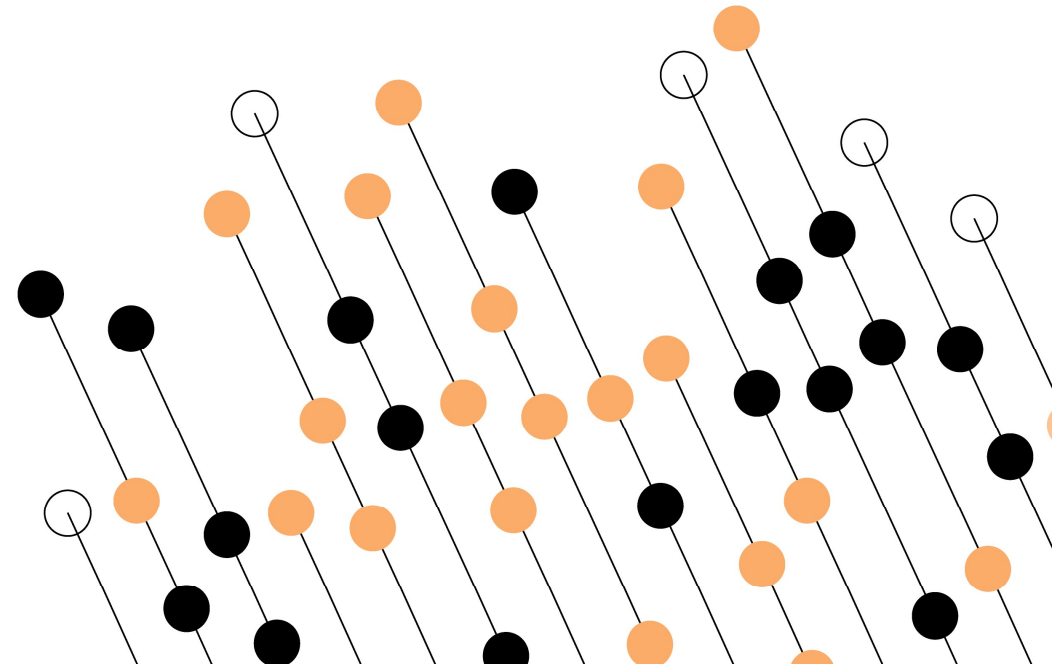


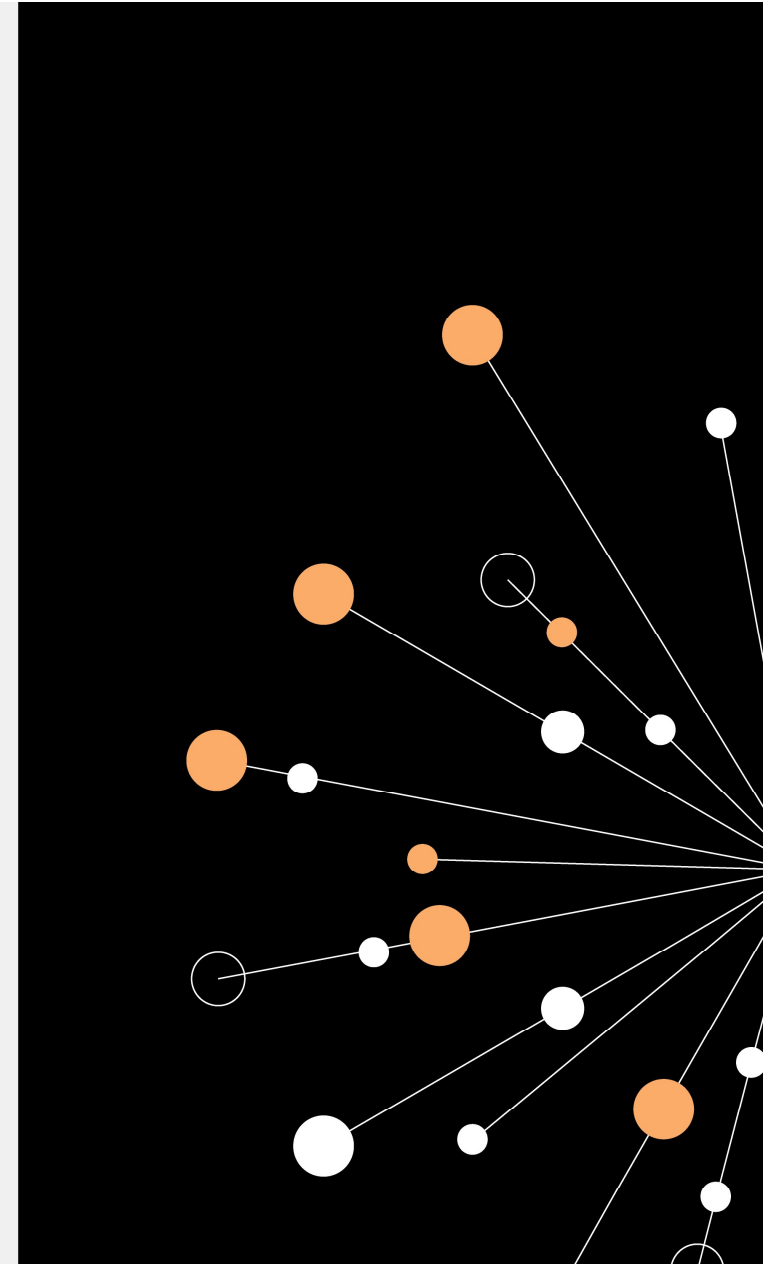
SQL: Sorting Rows



SQL

Contents

- Order By
- Where and Order By with Expressions



SELECT statement

SELECT <<field(s)>>

FROM <<table(s)>>

WHERE <<condition(s)>>

GROUP BY <<field(s)>>

HAVING <<condition(s)>>

ORDER BY <<field(s)>>



SELECT statement: Order by

- Row ordering is done last – after all filtering and calculating
- ORDER BY is the last clause in the SELECT statement
 - ASC – ascending - Default
 - DESC – descending

USE Northwind

```
SELECT ProductName, UnitPrice FROM Products  
ORDER By UnitPrice DESC
```

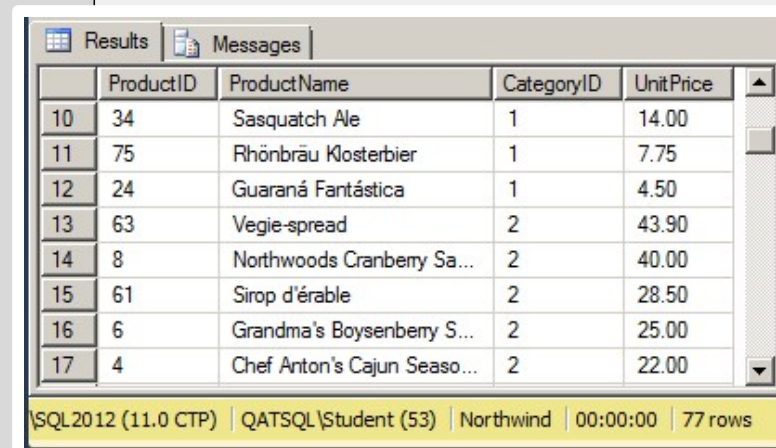
```
SELECT CategoryID, ProductName, UnitPrice  
FROM Products  
ORDER By CategoryID, UnitPrice DESC
```



Sort on multiple columns

- There can be up to 16 columns in ORDER BY clause.
- Sort columns do not have to be included in select list.

```
SELECT
    ProductID,
    ProductName,
    CategoryID,
    UnitPrice
FROM
    dbo.Products
ORDER BY
    CategoryID,
    UnitPrice DESC
```



The screenshot shows a SQL Server query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with 5 columns: ProductID, ProductName, CategoryID, and UnitPrice. The data is sorted by CategoryID and then by UnitPrice in descending order. The status bar at the bottom indicates the query was executed on SQL2012 (11.0 CTP) for the QATSQL\Student (53) database, showing 77 rows.

| | ProductID | ProductName | CategoryID | UnitPrice |
|----|-----------|-----------------------------|------------|-----------|
| 10 | 34 | Sasquatch Ale | 1 | 14.00 |
| 11 | 75 | Rhönbräu Klosterbier | 1 | 7.75 |
| 12 | 24 | Guaraná Fantástica | 1 | 4.50 |
| 13 | 63 | Vegie-spread | 2 | 43.90 |
| 14 | 8 | Northwoods Cranberry Sa... | 2 | 40.00 |
| 15 | 61 | Sirop d'érable | 2 | 28.50 |
| 16 | 6 | Grandma's Boysenberry S... | 2 | 25.00 |
| 17 | 4 | Chef Anton's Cajun Seaso... | 2 | 22.00 |

SQL2012 (11.0 CTP) | QATSQL\Student (53) | Northwind | 00:00:00 | 77 rows

SELECT statement: Order by **With Expressions**

```
SELECT ProductID, ProductName,  
(UnitsInStock+UnitsOnOrder)*UnitPrice AS Revenue  
FROM Products  
  
WHERE (UnitsInStock+UnitsOnOrder)*UnitPrice > 1000  
  
ORDER BY (UnitsInStock+UnitsOnOrder)*UnitPrice
```

```
SELECT ProductID, ProductName,  
(UnitsInStock+UnitsOnOrder)*UnitPrice AS Revenue  
FROM Products  
  
WHERE (UnitsInStock+UnitsOnOrder)*UnitPrice > 1000  
  
ORDER BY Revenue
```

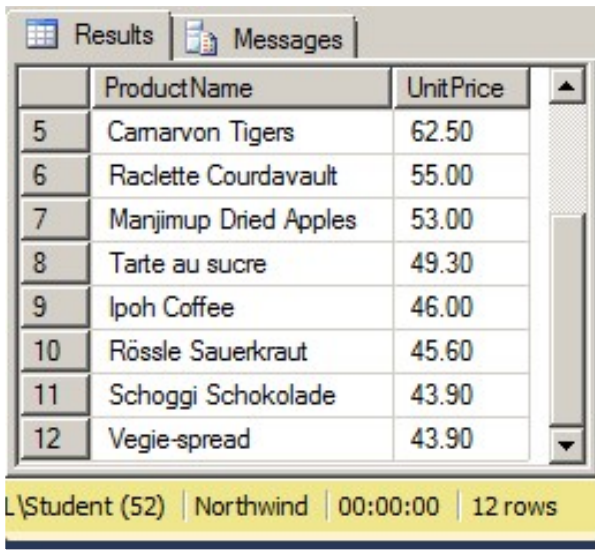
Alias can be used in ORDER BY but not in WHERE clause.



Ordering and filtering: select top

- **SELECT TOP n**
- **SELECT TOP n PERCENT**
- **SELECT TOP n ... WITH TIES**

```
SELECT TOP 11 WITH TIES
  ProductName,
  UnitPrice
FROM
  dbo.Products
ORDER BY
  UnitPrice DESC
```



| | ProductName | UnitPrice |
|----|-----------------------|-----------|
| 5 | Camavon Tigers | 62.50 |
| 6 | Raclette Courdavault | 55.00 |
| 7 | Manjimup Dried Apples | 53.00 |
| 8 | Tarte au sucre | 49.30 |
| 9 | Ipoh Coffee | 46.00 |
| 10 | Rössle Sauerkraut | 45.60 |
| 11 | Schoggi Schokolade | 43.90 |
| 12 | Vegie-spread | 43.90 |

Student (52) | Northwind | 00:00:00 | 12 rows

- **What of the TOP n options would you use in your work?**