

Copyright © 2015 by The Learning House.

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 Learning House. For permission requests, write to The Learning House, addressed “Attention: Permissions Coordinator,” at the address below.

The Learning House
427 S. 4th Street #300
Louisville KY 40202

Basic Queries

.NET Cohort

Coding Bootcamp

- Be aware of database context
- Learn to select data with... the **SELECT** command!

- It is a common best practice to terminate your queries in T-SQL with semi-colons, but in most cases it is optional
- I personally tend to not use them unless they are required
- Oh... and SQL is not case-sensitive

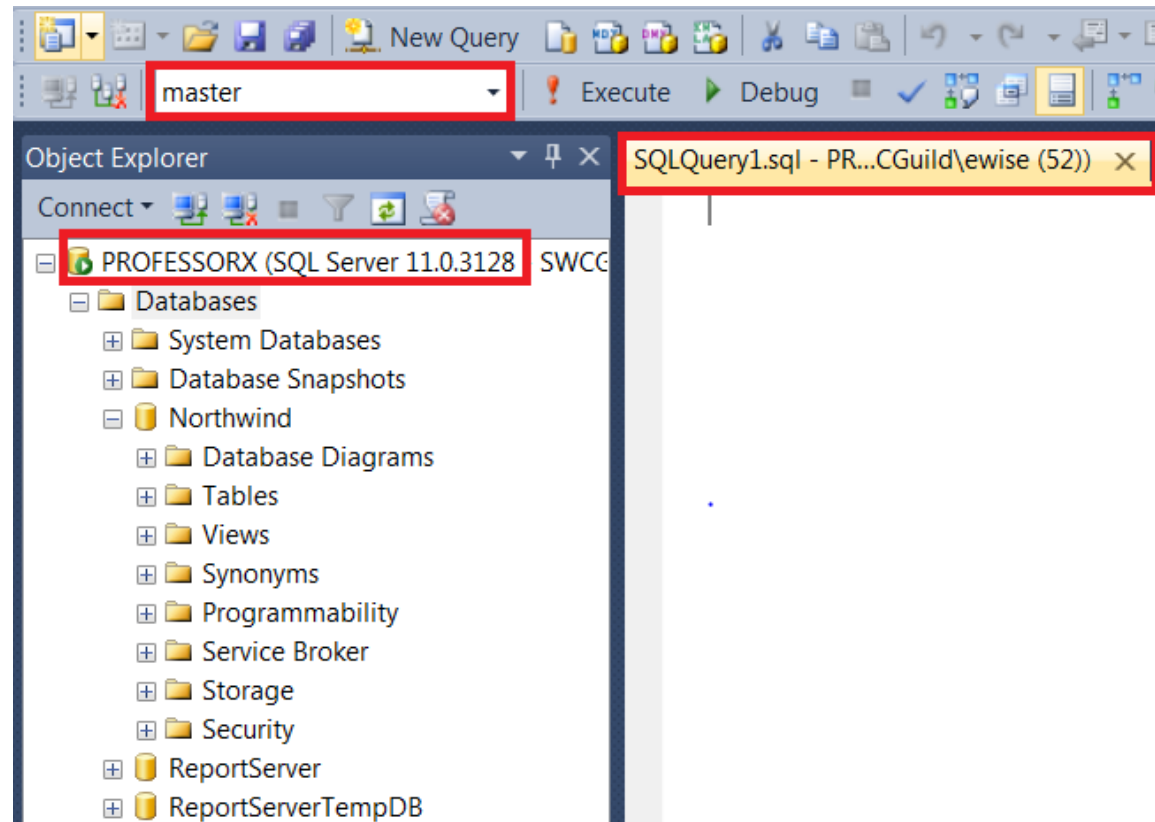
Open SQL Server Management Studio

This is the primary interface for working with SQL Server

Notice the Object Explorer on the left. This allows you to navigate your databases (you can have multiple databases on a server) and drill down into their objects.

IMPORTANT

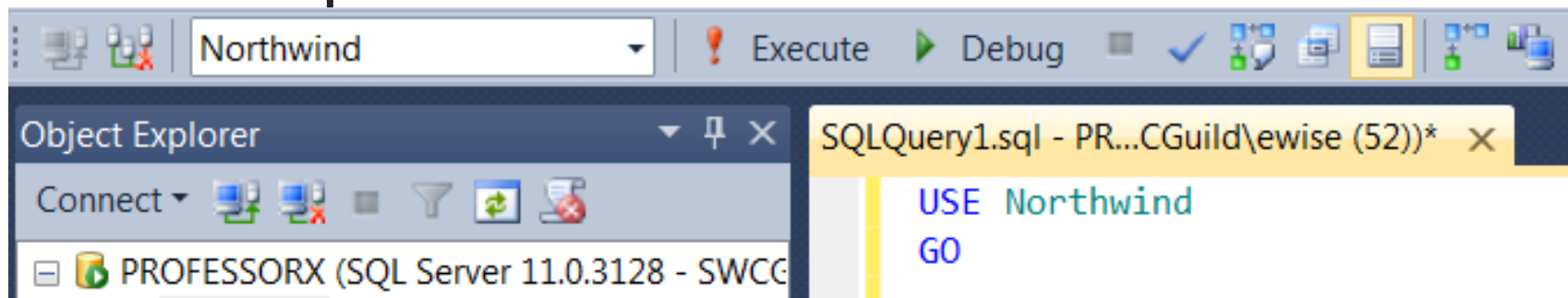
Always make sure you are in the right database context for the current query window or put `USE <database name>` at the top of your query.



1. The master database is a system database, usually the default
2. ProfessorX is my machine, I can get a list of all databases
3. My query tab is open on the right, this is where I write my SQL

I can use the New Query up above and the Execute buttons to create new tabs and run the SQL

- To be safe, we can put the USE statement to switch databases (usually at the top, but we can switch context in the middle of a script as well).
- Otherwise, you have to remember to pick it in the drop down.



Single Table Queries

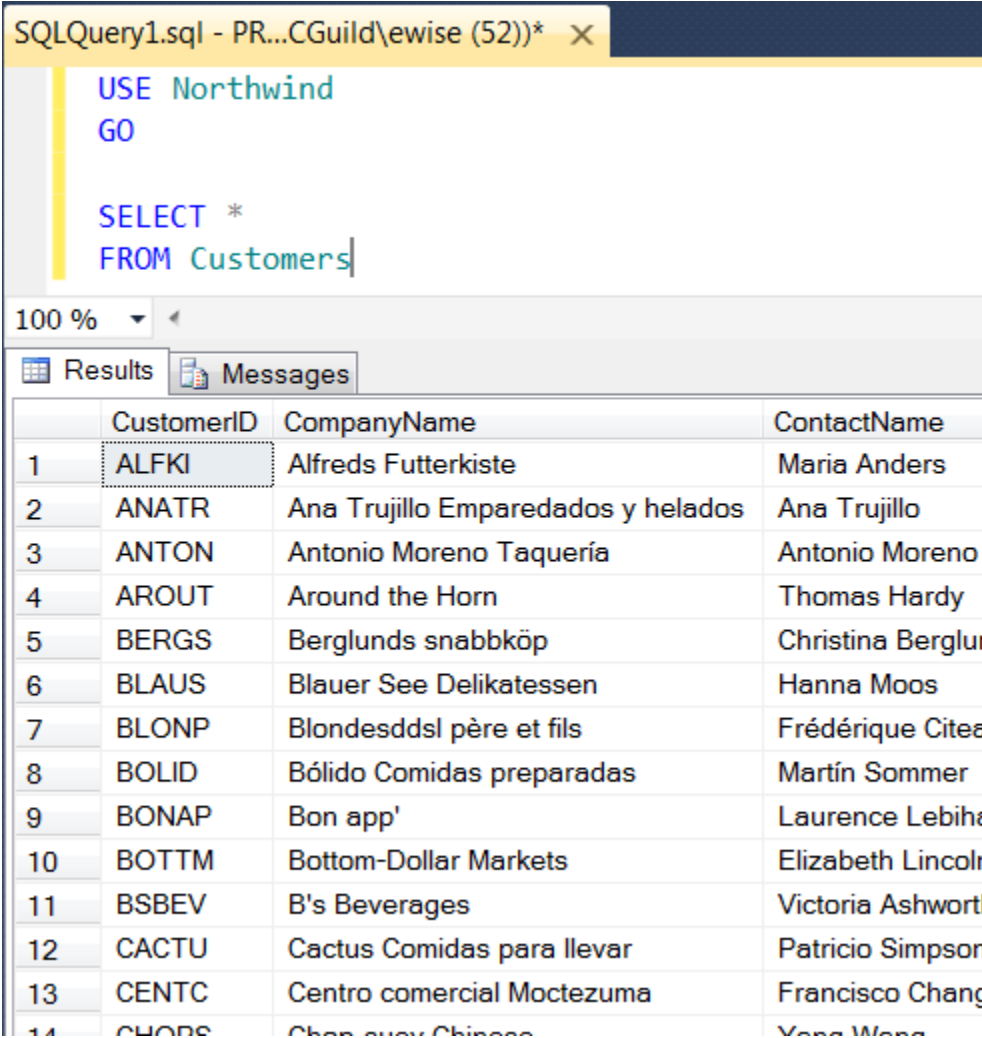
When we want work with a database, we write a *Query*.

The simplest query is the SELECT query. The syntax for this is:

```
SELECT <fields>  
FROM <tablename>
```

We can list the columns for the table or use an asterisk (*) to select all columns.

Check the bottom for some other helpful info (rows returned, db, time elapsed, etc.)



The screenshot shows a SQL query window titled 'SQLQuery1.sql - PR...CGuild\ewise (52))' with the following SQL code:

```
USE Northwind  
GO  
  
SELECT *  
FROM Customers
```

Below the query window, the 'Results' tab is active, displaying a table with 14 rows and 4 columns: CustomerID, CompanyName, and ContactName. The first row is highlighted.

	CustomerID	CompanyName	ContactName
1	ALFKI	Alfreds Futterkiste	Maria Anders
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo
3	ANTON	Antonio Moreno Taquería	Antonio Moreno
4	AROUT	Around the Horn	Thomas Hardy
5	BERGS	Berglunds snabbköp	Christina Berglund
6	BLAUS	Blauer See Delikatessen	Hanna Moos
7	BLONP	Blondesddsl père et fils	Frédérique Citeau
8	BOLID	Bólido Comidas preparadas	Martín Sommer
9	BONAP	Bon app'	Laurence Leblond
10	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln
11	BSBEV	B's Beverages	Victoria Ashworth
12	CACTU	Cactus Comidas para llevar	Patricio Simpson
13	CENTC	Centro comercial Moctezuma	Francisco Chang
14	CHOPS	Chop-suey Chinese	Yong Wong

At the bottom of the window, a status bar displays the following information: PROFESSORX (11.0 SP1) | SWCGuild\ewise (52) | Northwind | 00:00:00 | 91 rows

```
]-- Two dashes put in a comment
/*
    Multiple line comments work just like
    in c#
*/

/*
    The square brackets are optional unless
    some doofus puts spaces in the table or
    column name or (more commonly) names
    a table or column the same as a reserved
    word
*/
SELECT * FROM [Customers]
```


Filtering Results With Where

The WHERE clause is the most common optional command. It is used to add criteria to filter the total number of results selected.

In the query to the right, we filter only customers whose Country field is set to 'USA.'

Note that SQL would like us to use a single quote ' character for text.

The logical statement (Country = 'USA') is called a *predicate*. Predicates can contain quite a few useful operands, as seen on the right.

```
SELECT *  
FROM Customers  
WHERE Country = 'USA'
```

```
SELECT *  
FROM Customers  
WHERE Country != 'USA'
```

```
SELECT *  
FROM Customers  
WHERE Country = 'USA' AND Region = 'OR'
```

```
SELECT *  
FROM Customers  
WHERE Country IN ('USA', 'UK')
```

```
SELECT *  
FROM Customers  
WHERE Country NOT IN ('USA', 'UK')
```

From the Products table, write these queries:

1. Select all of the fields and rows.
2. Find all the product information for the product with the name Queso Cabrales.
3. Using one query, display the name and number of units in stock for the products Laughing Lumberjack Lager, Outback Lager, and Ravioli Angelo.

1. A query written in the SQL language is a request for information from data stored in a table within a database.
2. Microsoft SQL Server uses the Transact Structured Query Language (T-SQL) standard for writing database queries.
3. Database context refers to which database to run the current query against.
4. The FROM clause tells SQL which table or tables contain searchable data.
5. When writing a SQL query, if a letter is missed, a punctuation mark forgotten, or a spelling error is made, SQL Server returns an error message.

Numeric Ranges

Numeric ranges can be used to find data that falls under, over, or between values.

Notice that the Between statement is inclusive (in this example, this means that 10 and 13 are included in the results).

```
SELECT *  
FROM Products  
WHERE UnitPrice > 20.00
```

```
SELECT *  
FROM Products  
WHERE UnitPrice <= 20.00
```

```
SELECT *  
FROM Products  
WHERE UnitPrice BETWEEN 10.00 AND 13.00
```

- Sometimes (especially with keys) we will have the same named column in two different tables. In order for SQL Server to tell which column we actually want, we need to *qualify* it with the table name like so:

```
SELECT Products.CategoryID  
FROM Products
```

From the Orders table, write these queries:

1. Select all the order information for the customer QUEDE.
2. Select the orders whose freight is more than \$100.00.
3. Select the orders shipping to the USA whose freight is between \$10 and \$20.

Pattern Matching

We can use the LIKE operator with a wildcard expression to do pattern matching on text fields.

% means any number of characters

_ (underscore) means exactly one character

[] (square brackets) can have a list of viable letters

(In MySQL, replace LIKE with REGEXP.)

```
SELECT *  
FROM Customers  
WHERE CompanyName LIKE 'A%'
```

```
SELECT *  
FROM Customers  
WHERE CompanyName LIKE '%Z%'
```

```
SELECT *  
FROM Customers  
WHERE CompanyName LIKE '_A%'
```

```
SELECT *  
FROM Customers  
WHERE CompanyName LIKE '[AB]%'
```

```
SELECT *  
FROM Customers  
WHERE CompanyName LIKE '[A-K]%'
```

```
SELECT *  
FROM Customers  
WHERE CompanyName LIKE 'A[^N]%'
```

```
SELECT *  
FROM Customers  
WHERE CompanyName LIKE 'A[^N-Z]%'
```


- Next up, joining tables!