## Overview

In this group assignment you will complete a tutorial that will help to familiarize you with SQL Server Management Studio.

*Please note that Microsoft SQL Server and SQL Server Management Studio are not the same thing!*

- Microsoft SQL Server is an enterprise-level database management system (DBMS) that we will use extensively throughout our course. SQL Server runs as a background service on the computer, and is not directly visible to the user.
- SQL Server Management Studio is a software tool that provides users with a visual interface that allows them to configure, manage, and administer Microsoft SQL Server.

*Each member of your group should complete this tutorial independently, however only one completed tutorial will need to be turned in for grading.*

This tutorial assumes that you are using a computer on which SQL Server 2016 or 2017 has been installed. If your computer does not already have SQL Server, you will need to install it in order to complete this tutorial. You can obtain a full version of SQL Server 2016 or 2017 for free through the Microsoft Imagine program (you should receive an email from the College with instructions on how to access this program).

- If you download a copy of SQL Server from Microsoft Imagine, please be sure to select a version that includes SQL Server Management Studio, otherwise you will need to download and install SQL Server Management Studio separately.

If you would prefer not to use the Microsoft Imagine system, you may obtain a free copy of SQL Server 2017 Express (which will be sufficient for our class) from Microsoft by navigating to the following URL:

https://www.microsoft.com/en-us/sql-server/sql-server-editions-express

Note that if you choose to install SQL Server 2017 Express, you will also need to download and install SQL Server Management Studio. **You should only install SQL Server Management Studio** *after* **you have installed SQL Server!** You can obtain SQL Server Management Studio from Microsoft by navigating to the following URL:

https://msdn.microsoft.com/en-us/library/mt238290.aspx

## Tutorial

I. Start SQL Server Management Studio (SSMS). When the "Connect to Server" dialog box appears, select a server (e.g., "SQLEXPRESS", "MSSQLSERVER", etc.), and then click "Connect".

- **<u>Note</u>**: if no options appear in the dialog box, then the SQL Server database service may not be running on your machine. To check whether the SQL Server service is running, press (Windows Key + R) to activate the "Run" dialog box, type "services.msc", and then click the "OK" button. When the "Services" window appears, scroll down, look for a service named "SQL Server", and ensure that the status for the service is "Running". If the service is not running, then right-click on the service name and select "Start". If a service named "SQL Server" does not appear in your services window, then the SQL Server database engine is not installed on your machine (follow the instructions above to install SQL Server).

II. SQL Server Management Studio (SSMS) is the primary administration console for SQL Server. SSMS enables you to create database objects (such as databases, tables, views, constraints, etc.), view the data within your database, configure user accounts, transfer data between databases, and much more. Here's an example of the SQL Server Management Studio interface:
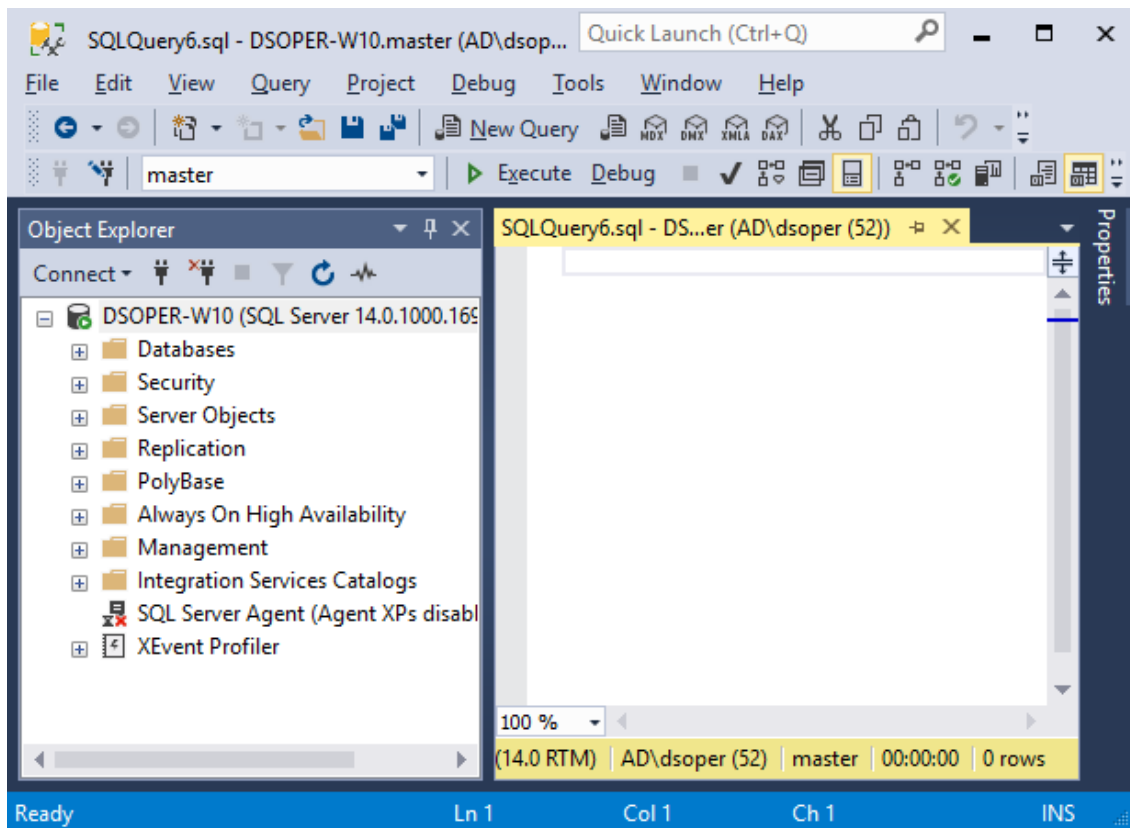
The left pane contains the Object Explorer. The Object Explorer provides navigation to databases, server objects (such as triggers and stored procedures), log files, and more.

The right pane allows you to write database queries and view the results of those queries. In this screenshot a blank query has been opened by clicking the "New Query" button.

You may use SQL Server Management Studio to create as many databases as you like. You may also use SSMS to connect to as many databases on remote servers as you like.

**TIP**: Most of the tasks performed with SQL Server Management Studio are initiated either from the top menu or by right-clicking on an icon or object.

Throughout this tutorial, we'll be exploring some of the various things that you can do via SQL Server Management Studio.

III.    One of the first things we should do within SQL Server Management Studio is to create a database. After all, most of the tasks we perform with SQL Server will involve one or more databases.

### System Databases
If you're working with a "fresh" installation of SQL Server, you might notice that some databases have already been created. These are *system databases*.

| Database | Type | Description |
|---|---|---|
| **master** | System database | Stores system-level information such as user accounts, configuration settings, and info on all other databases. |
| **model** | System database | This database is used as a template for all other databases that are created. |
| **msdb** | System database | Used by the SQL Server Agent for configuring alerts, scheduled jobs, etc. |
| **tempdb** | System database | Holds all temporary tables, temporary stored procedures, and any other temporary storage requirements generated by SQL Server. |

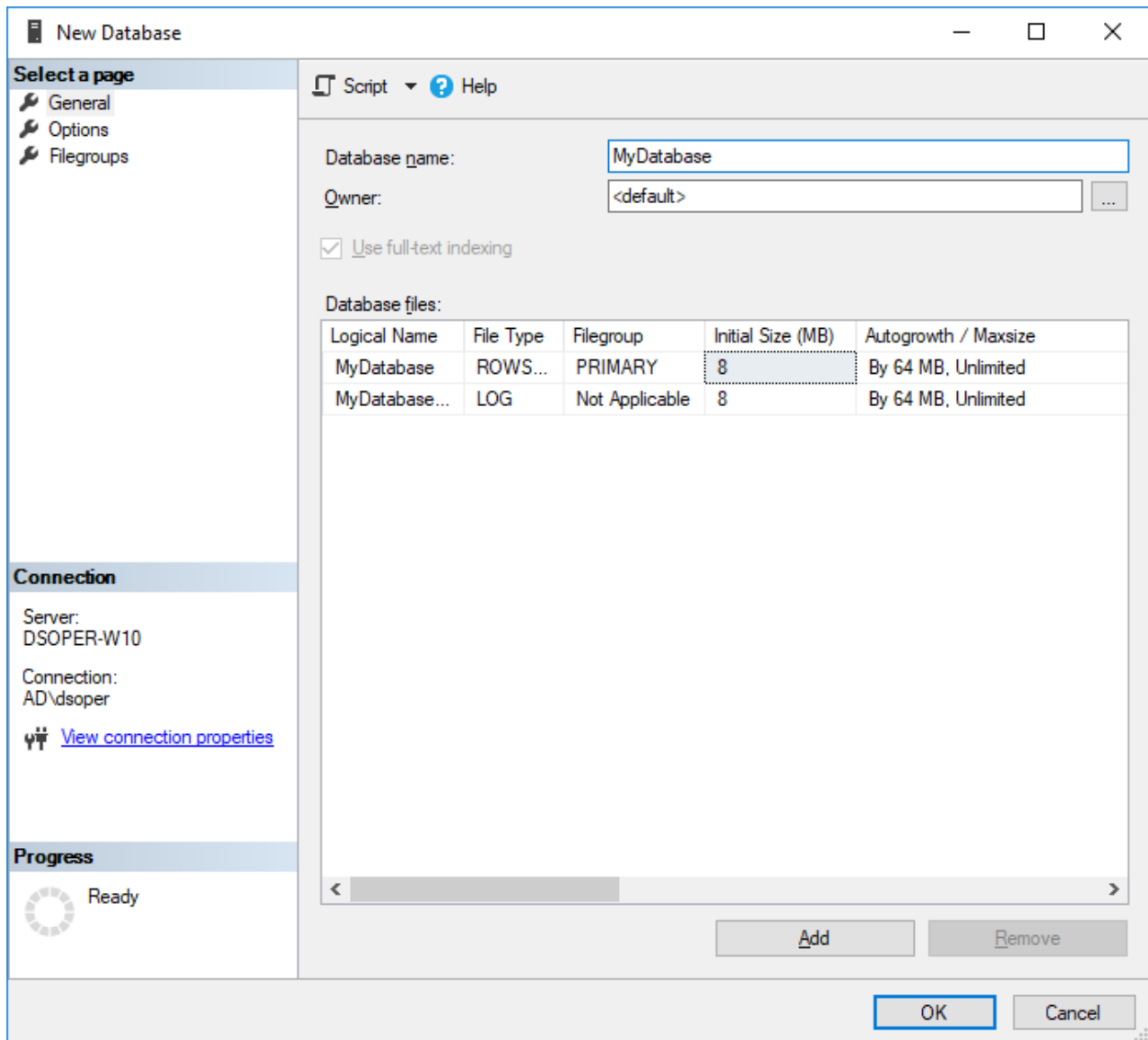We will now create another database for our own use.

### Creating a New Database

The following steps demonstrate how to create a database in SQL Server using SQL Server Management Studio.

- Right click on the "Databases" icon and select "New Database...":
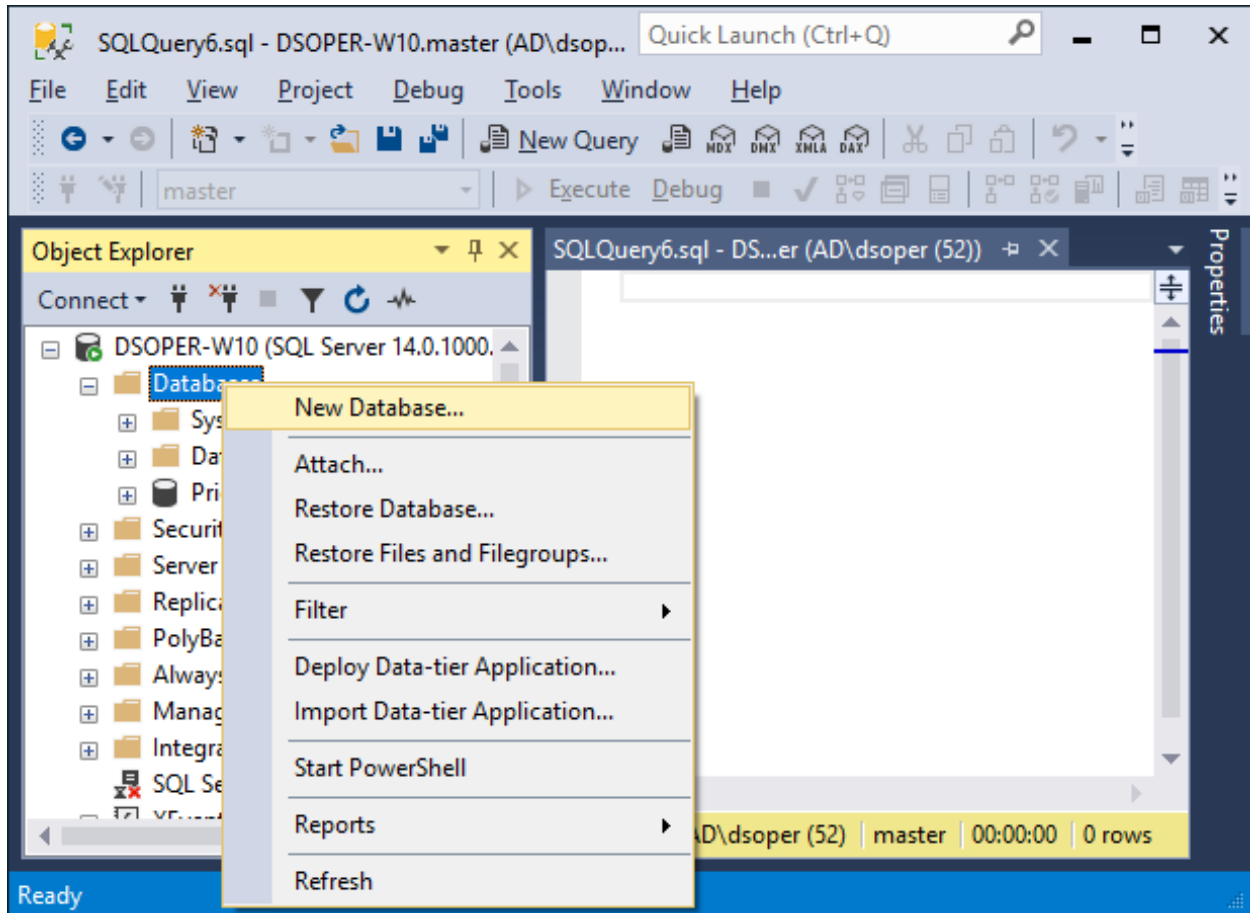
- Type a name for your new database and click "OK":



### Your New Database
You will notice that your new database now appears under the "Databases" item in the Object Explorer.

Your new database is based on the "Model" database. The Model database is a system database that is used as a template whenever a new database is created. If you use the Object Explorer to navigate to your database and expand the tree, you will notice that your database includes a number of folders, such as *Database Diagrams, Tables, Views*, etc.
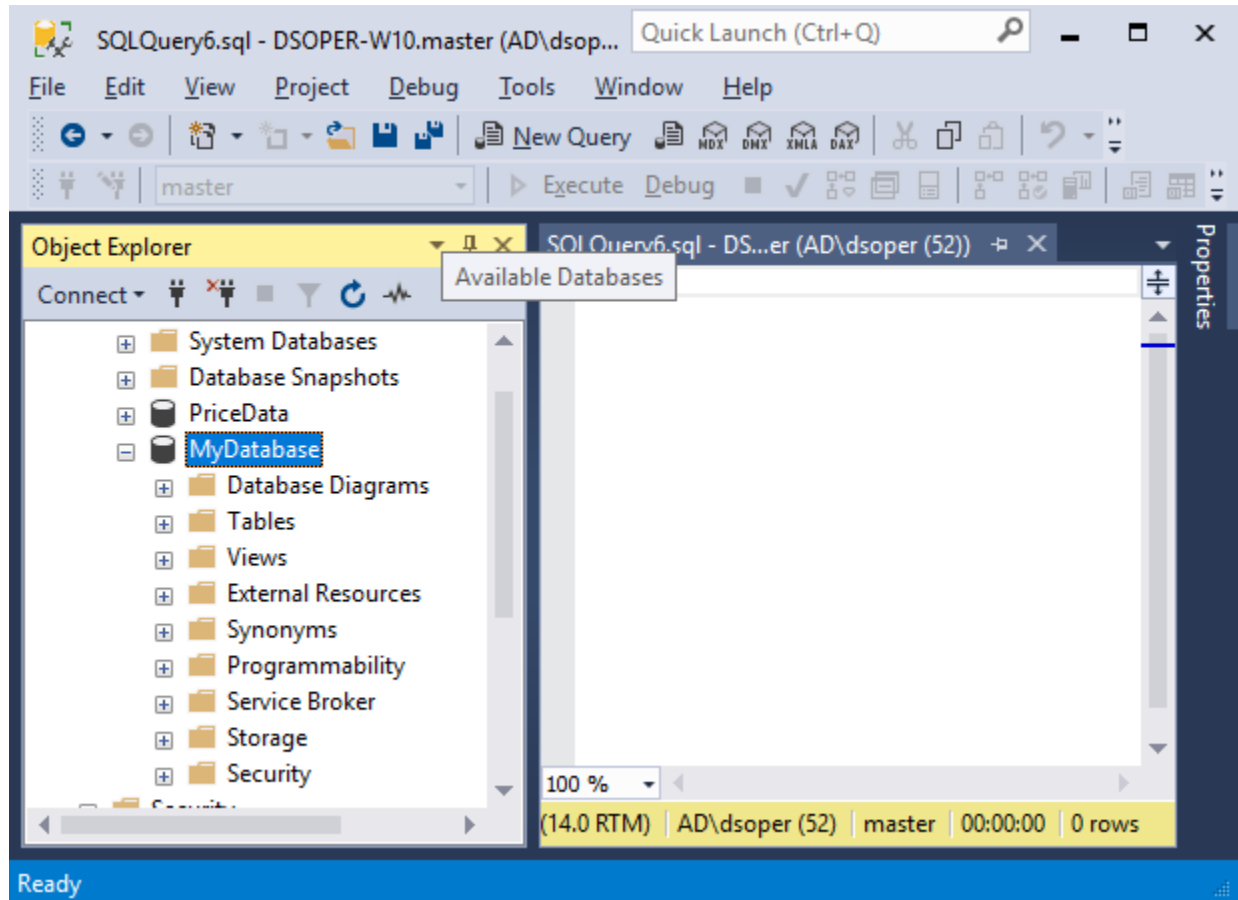
### Other Options

We have just created a database using the default options. When we created the database, a "Data File" and a "Transaction Log" were created. They were created in the default location for our server.

If we had wanted to do so, we could have specified a different location for these files. We also could have changed whether to allow the data file to grow automatically (as it stores more and more data), and if so, how that growth should be managed. Although we could have done these things in step 2 above, we can also modify these options via the Properties dialog box.

To view or change the database's properties, simply right click on the name of your database and select "Properties":
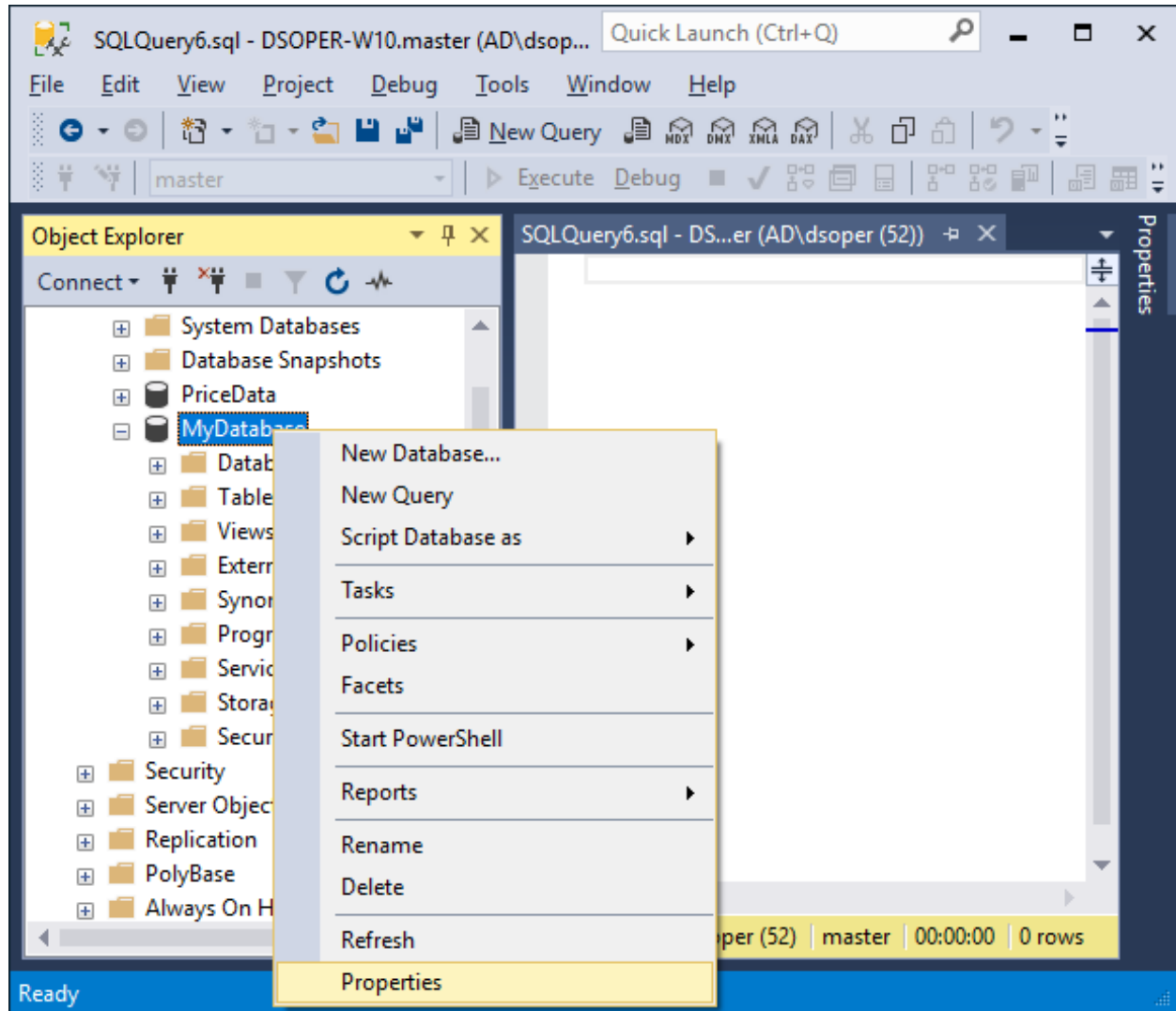
The Properties dialog (shown on the next page) contains a large number of options for changing the configuration of your database. For now, we can leave everything at its default setting.

Database Properties - MyDatabase — □ ×

**Select a page**
- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties
- Mirroring
- Transaction Log Shipping
- Query Store

Script ▼ ❓ Help

| Collation: | SQL_Latin1_General_CP1_CI_AS | ˅ |
|---|---|---|
| Recovery model: | Full | ˅ |
| Compatibility level: | SQL Server 2017 (140) | ˅ |
| Containment type: | None | ˅ |

Other options:

| | | |
|---|---|---|
| ˅ | **Database Scoped Configurations** | |
| | Legacy Cardinality Estimation | **OFF** |
| | Legacy Cardinality Estimation For Secondary | **PRIMARY** |
| | Max DOP | **0** |
| | Max DOP For Secondary | |
| | Parameter Sniffing | **ON** |
| | Parameter Sniffing For Secondary | **PRIMARY** |
| | Query Optimizer Fixes | **OFF** |
| | Query Optimizer Fixes For Secondary | **PRIMARY** |
| ˅ | **FILESTREAM** | |
| | FILESTREAM Directory Name | |
| | FILESTREAM Non-Transacted Access | **Off** |
| ˅ | **Misc** | |
| | AllowScripting | True |
| | HideFileSettings | False |
| ˅ | **Miscellaneous** | |
| | Allow Snapshot Isolation | **False** |
| | ANSI NULL Default | **False** |

**Allow Snapshot Isolation**

**Connection**

Server:
DSOPER-W10

Connection:
AD\dsoper

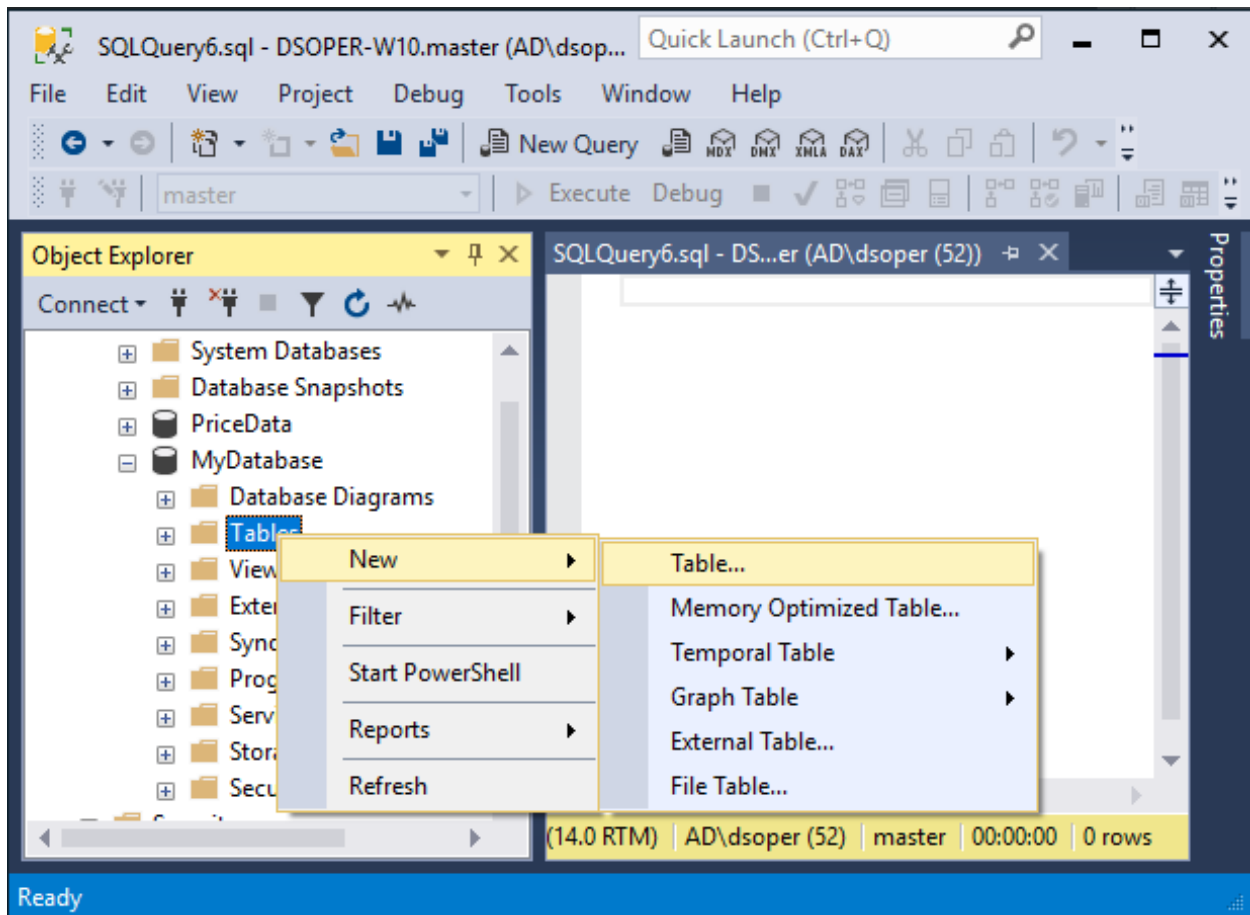View connection properties

**Progress**

Ready

OK     Cancel

IV. Next, we'll learn how to create a table in a SQL Server database using SQL Server Management Studio (SSMS).

- Ensuring that you have your database expanded in the Object Explorer, right click on the "Tables" icon and select "New" and then "Table...":

- Next, perform the following tasks to define the structure of your new table:
  - i. Use the values in the screenshot below to fill in the details for the "Column Name", "Data Type", and "Allow Nulls" columns:
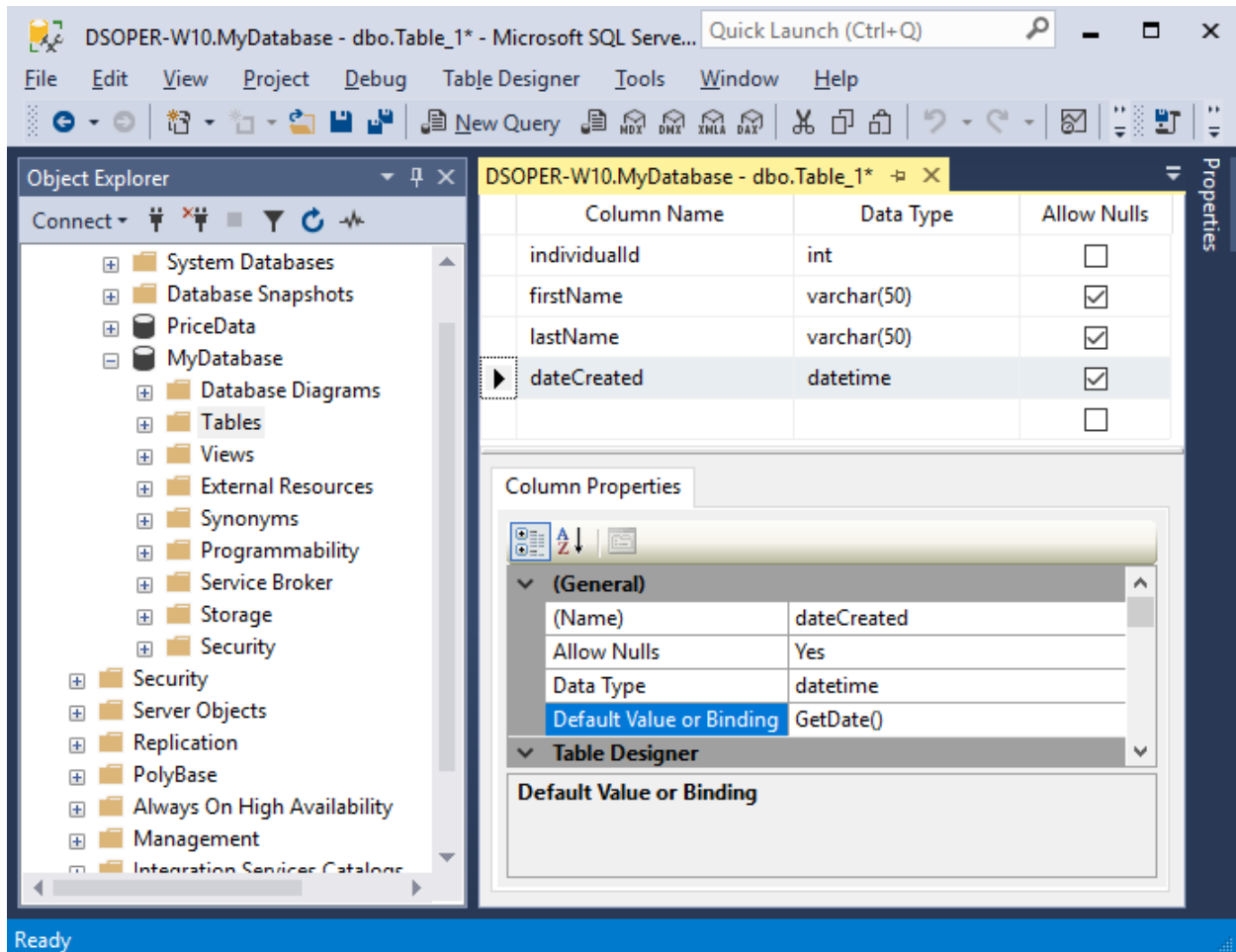


  - ii. Make the "individualId" column into an "identity column" by setting its "Is Identity" property to "Yes" (this option is under the "Identity Specification" section in the bottom pane). Note that to set values in the bottom pane, you need to select the column name in the top pane first. This column is going to be an auto-number column; i.e., it will contain a unique, automatically incrementing number for each record that is created.
  - iii. Set the "Default Value or Binding" of the dateCreated column to "GetDate()" (without the double quotes, as shown above). This will automatically insert the current date and time into that field whenever a new record is created.

- You have just finished creating the column names, specifying the type of data that can be entered into each column, and setting default values for each column. Specifying the correct data type for each column is very important because it helps to maintain data integrity. For example, it can prevent us from accidentally entering an email address into a field that is designed for storing the current date.

- Save your table by selecting File >> Save Table_1:
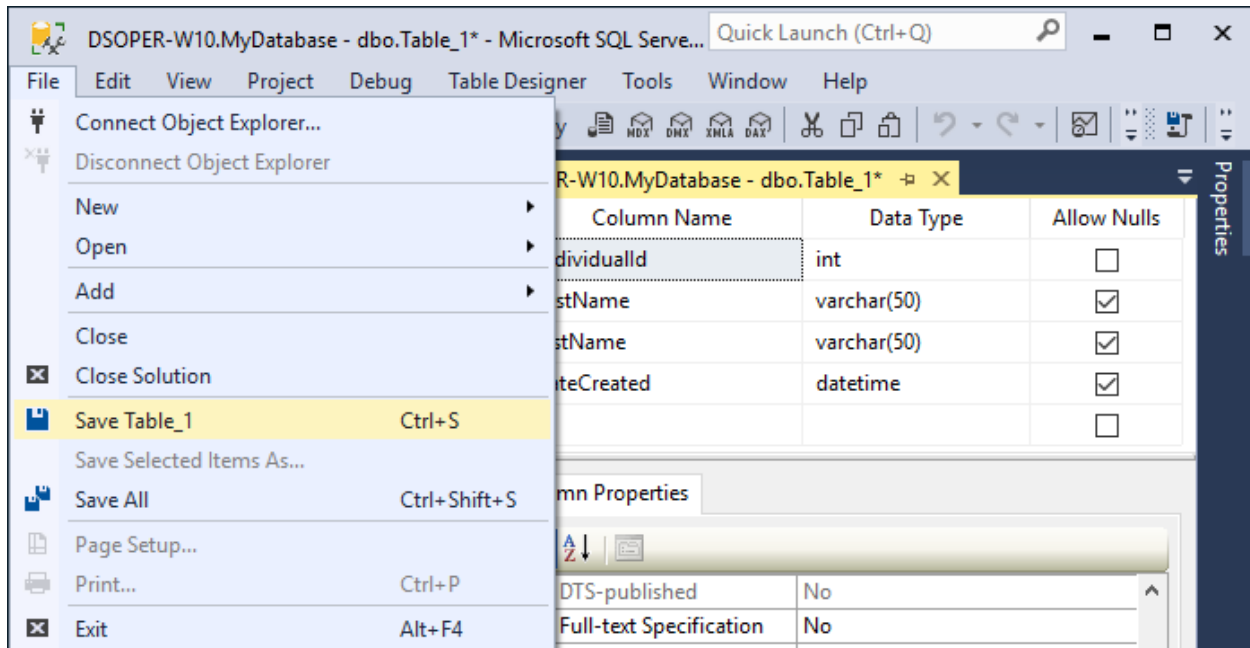


- Name your table "Individual" when prompted:
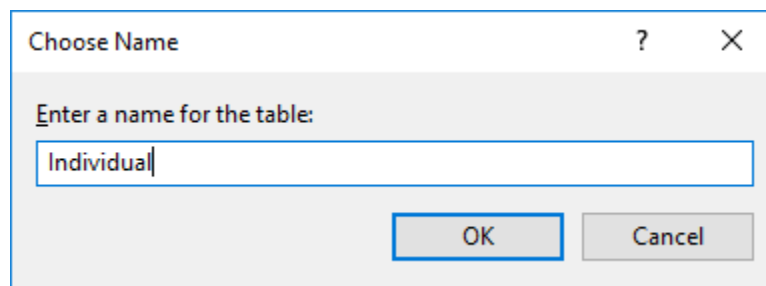


- Click "OK". Note that your new table appears in the "Tables" section under your database in the Object Explorer. If your table does not appear, right-click on the "Tables" folder, and then click "Refresh".

V.    Next, we'll use the "Edit Top 200 Rows" option to add data to our table.

**Editing Data**
- Right click on the table you just created, and select "Edit Top 200 Rows":
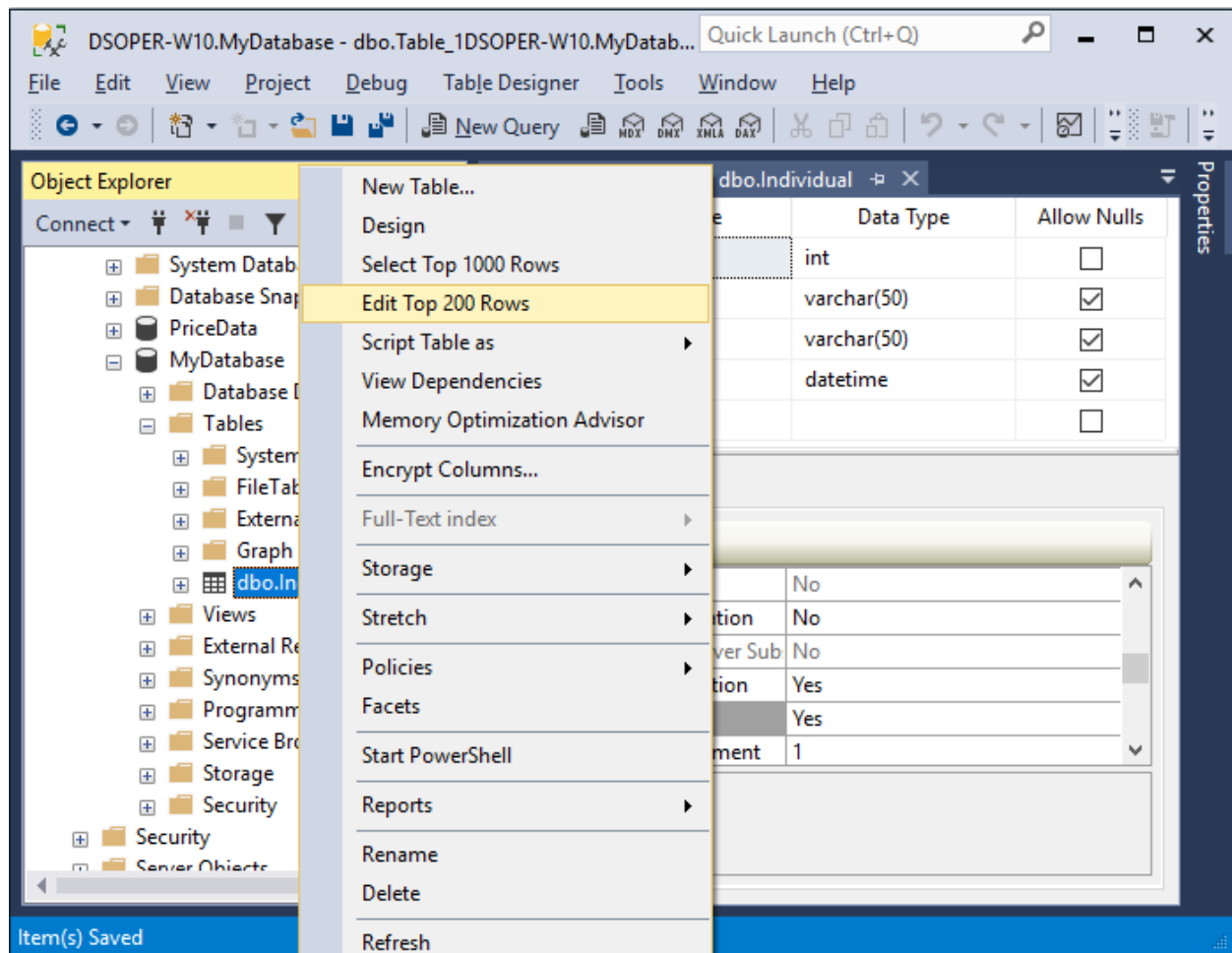


- You may now begin entering data directly into your table. Note that you don't need to enter any data into the individualId or dateCreated columns, since they will be populated automatically (remember that we set individualId to "Is Identity" and dateCreated to "GetDate()").

- Add four new individuals to the table by specifying values for the firstName and lastName attributes. An example is shown below, but feel free to choose your own names:
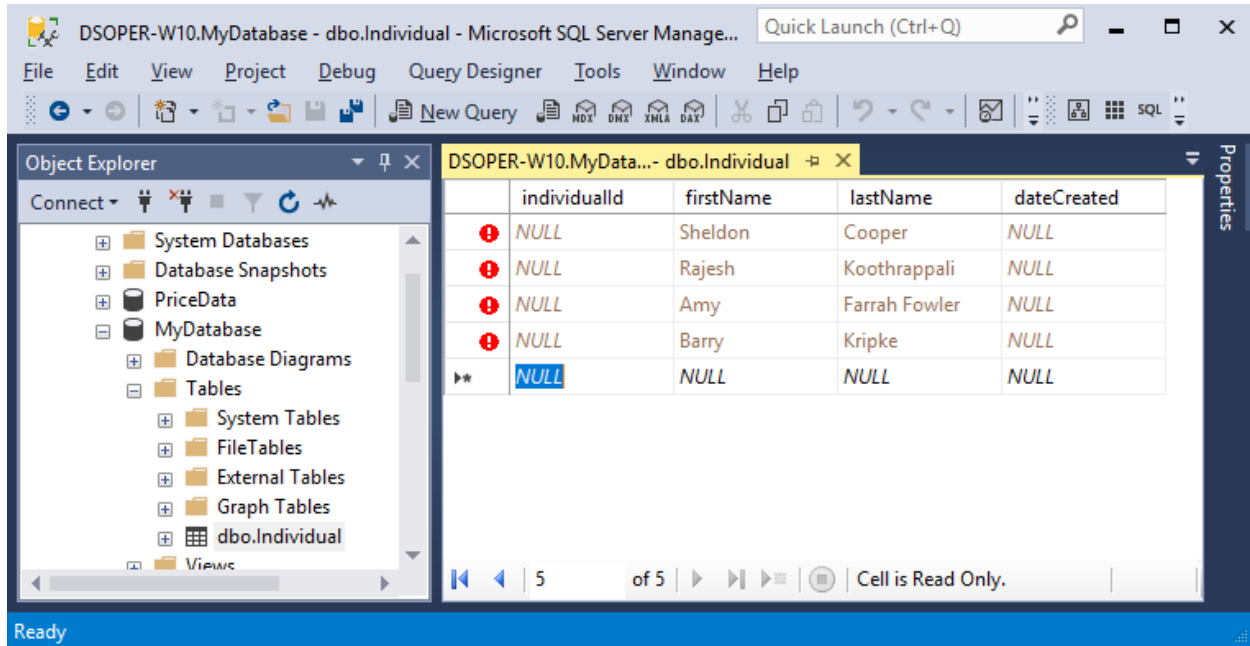


After you have entered your four individuals, close the data editor by clicking the black "X" that appears on the right side of the tab. **Warning**: This is not the same as the "X" button that closes the entire application!!

VI.  Finally, we'll create and run a simple SQL query to retrieve some data from the database.

**<u>Create a New Query</u>**

- Before we can write our SQL query, we need somewhere in which to type it. This part is easy. Just click the "New Query" button:
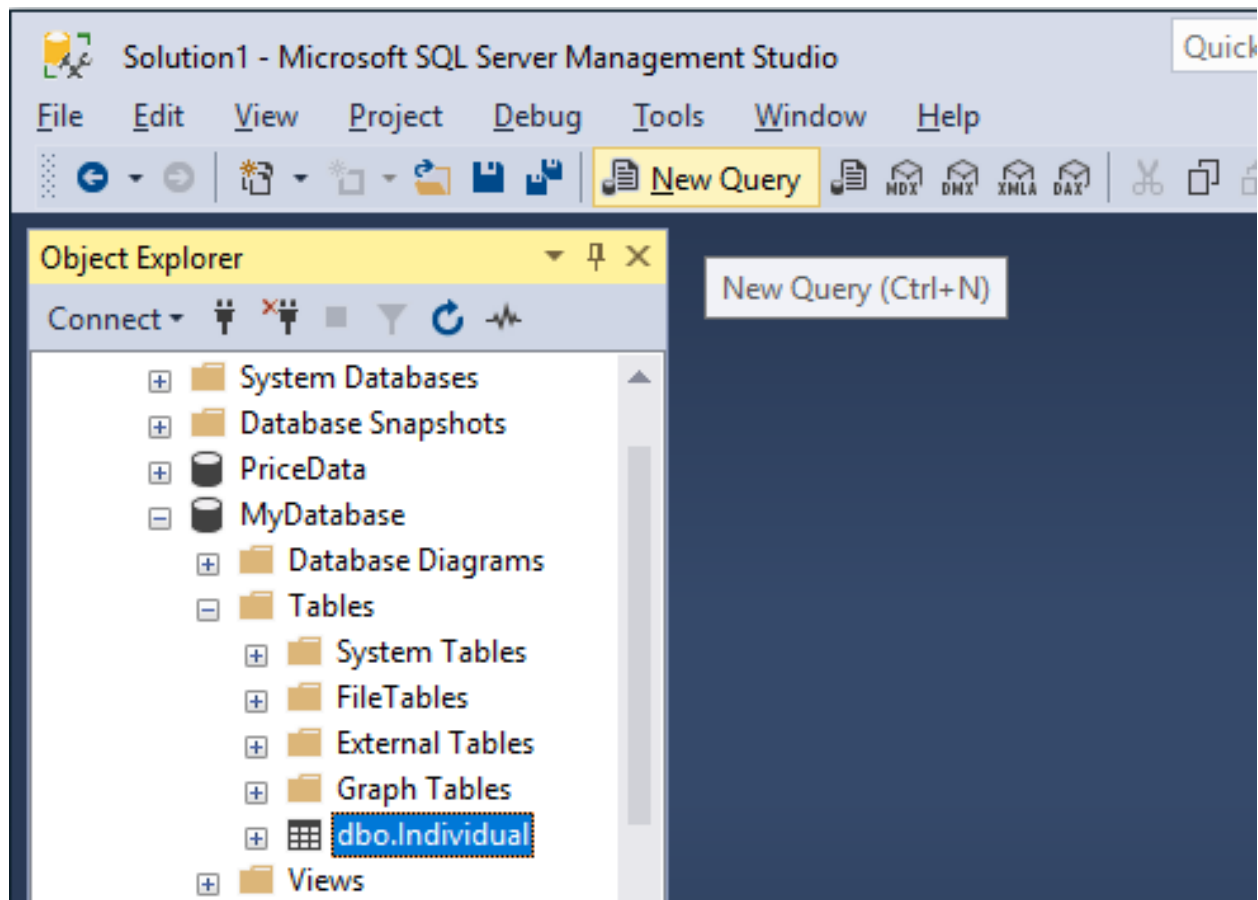


A blank, white workspace should now appear in the right pane.

- To write and run a SQL query, you'll first need to type your query into the workspace on the right pane, and then click "Execute". The following screenshot shows an example of using a SQL 'SELECT' statement to select all of the data from your table:
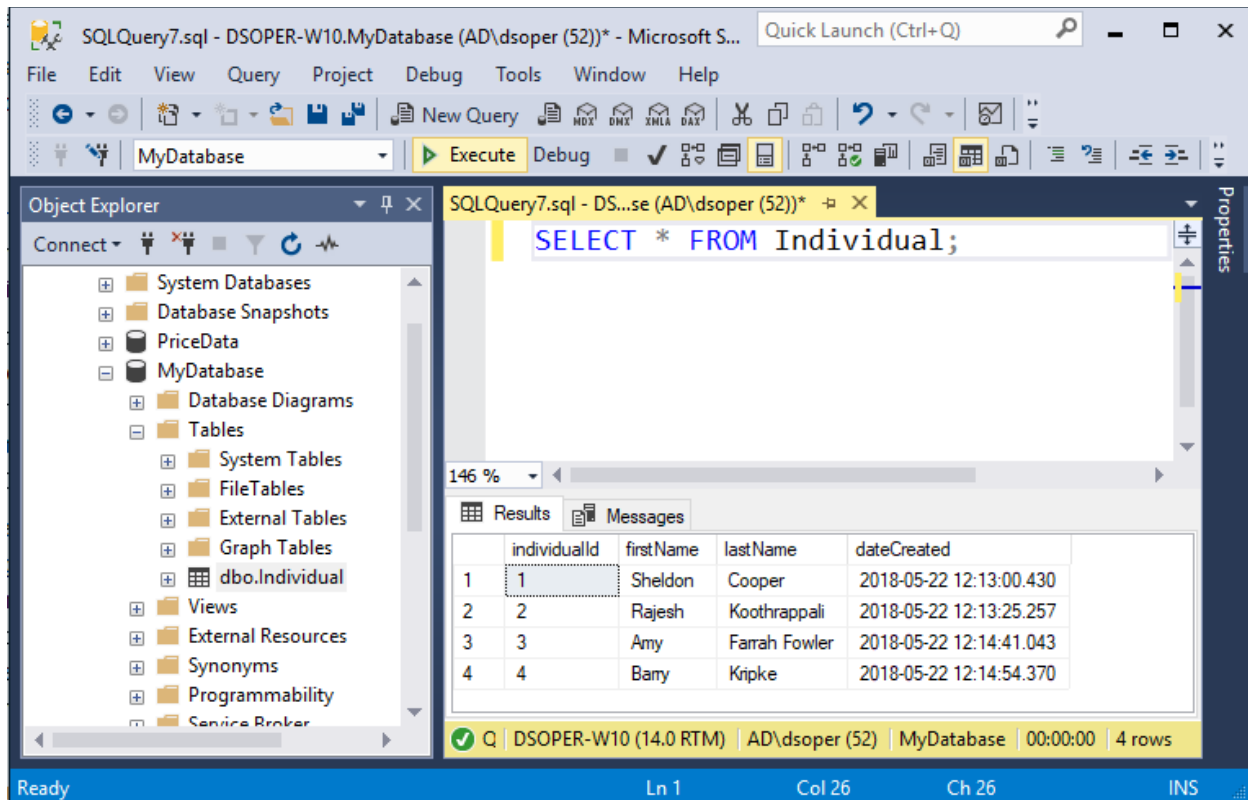


As you can see, the results of the query are displayed in the bottom pane. Note that values for the "individualId" and "dateCreated" attributes have been provided automatically by SQL Server (since we set individualId to "Is Identity" and dateCreated to "GetDate()").

- Run the SQL query shown above. If you've done everything correctly, you should get a table of results that contains all of the individuals in your "Individual" table.

- Take a screenshot of your results by pressing Alt+PrintScreen, and then paste the results into a new Microsoft Word document.

## Deliverables

To ensure that you receive credit for this assignment, please complete the following tasks:

1. Complete the *Group Assignment Participation Form* for this assignment (available on the course website). **Each group member who contributed to the assignment should complete this task.**

2. Submit the Word document that shows your group's query results using the *Assignment 01* link on the course website. **Each group should submit just one document.**