# Homework #2: Mastering the Basics of SQL Server 2016

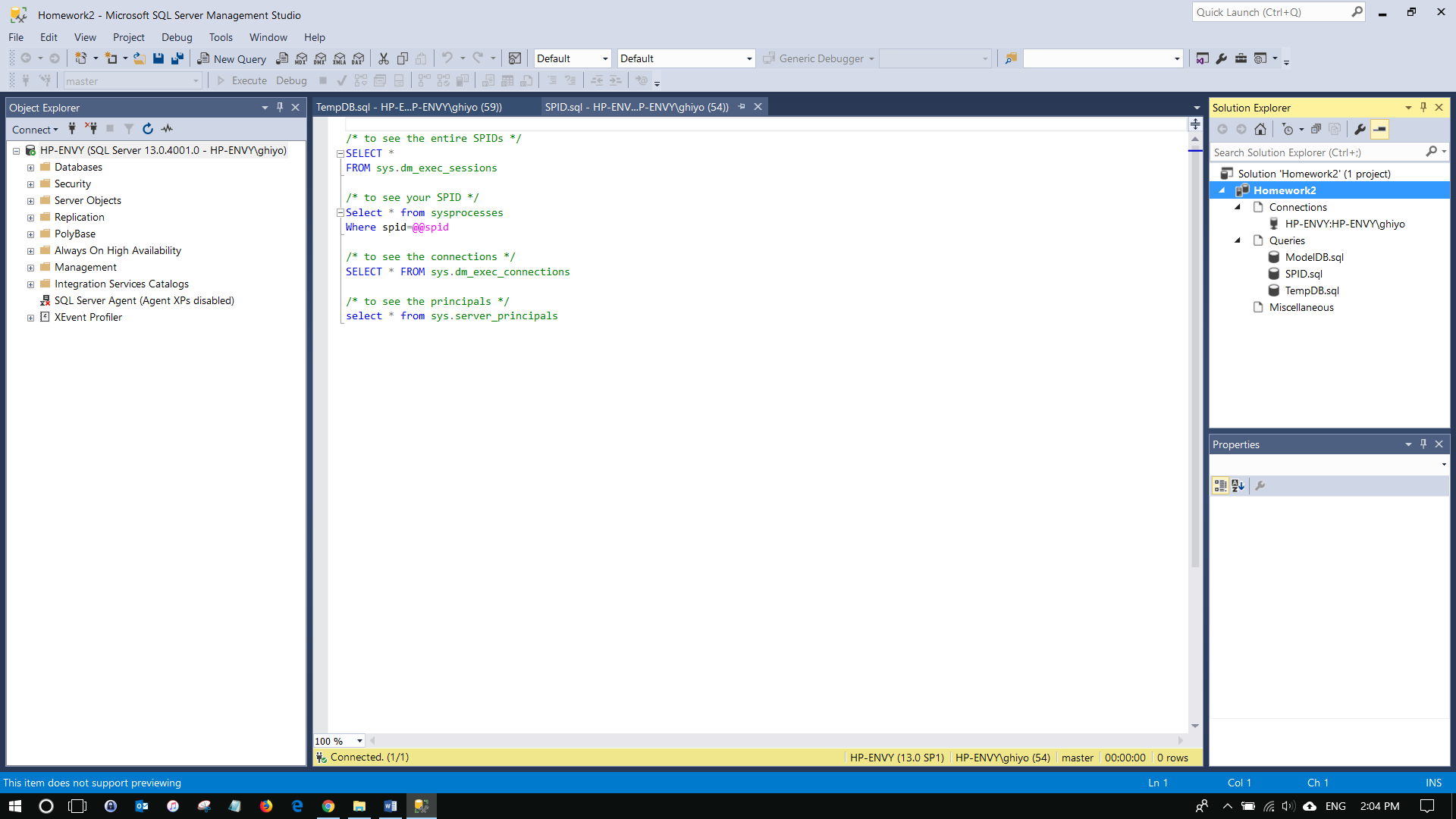
|  |
| --- |
| * This is an individual assignment, and is worth 20 points. * The due date is 11:00 AM (sec 02) / 5:30 PM (sec 76) on Tuesday, January 29th. * You have to use the “Homework #2-Tasks.docx” file to provide your answers. Change this file name according the naming convention below. * Use the following naming convention: homework, underscore, last name, first initial, and extension (e.g., Homework #2\_ImG.docx). If you do not follow the convention, I will deduct 1.0. * Do not copy any of the sample screenshots provided as illustrations. * Report any issues before the deadline. |

# I. SQL Server Management Studio

## Task #1: How to Use Solution Explorer[[1]](#footnote-1)

We use this to manage a list of work items in a project. To create a new solution and project:

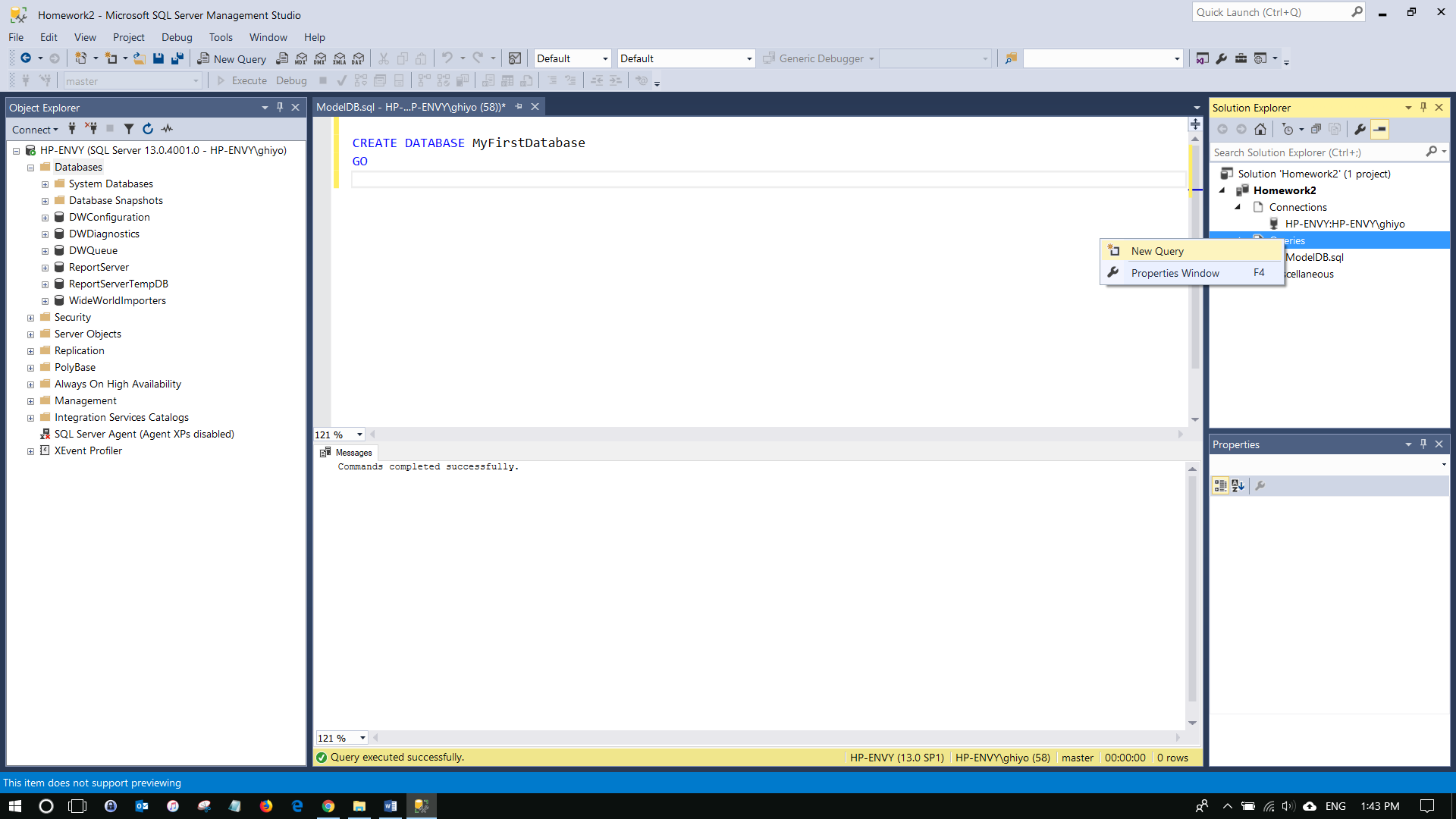
* First, connect to the Server.
* On the File menu, point to New, and then click Project.
* In the New Project dialog box, select SQL Server Scripts.
* Optionally modify the project name in the Name text box.
* Optionally modify the path for the solution in the Location text box.
* Optionally modify the solution name in the Solution Name text box.
* Click OK.
* Create Connections, Queries, and change properties
* (**Task 1: 5 points**) Complete Task #2 through Task #4 while placing your sql files in Queries. Show in a screenshot that your Homework 2 project in Solution Explorer contains **three** sql files in **Solution Explorer**. Refer to the example below.



## Task #2: How to create ModelDB

When a CREATE DATABASE statement is issued, the first part of the database is created by copying the contents in the model database. The rest of the new database is then filled with empty pages. If you modify the model database, all databases created afterward will inherit those changes. For example, you could set permissions or database options, or add objects such as tables, functions, or stored procedures.[[2]](#footnote-2)

* Create a new query file named **ModelDB** **within the project** you created in Task #1.

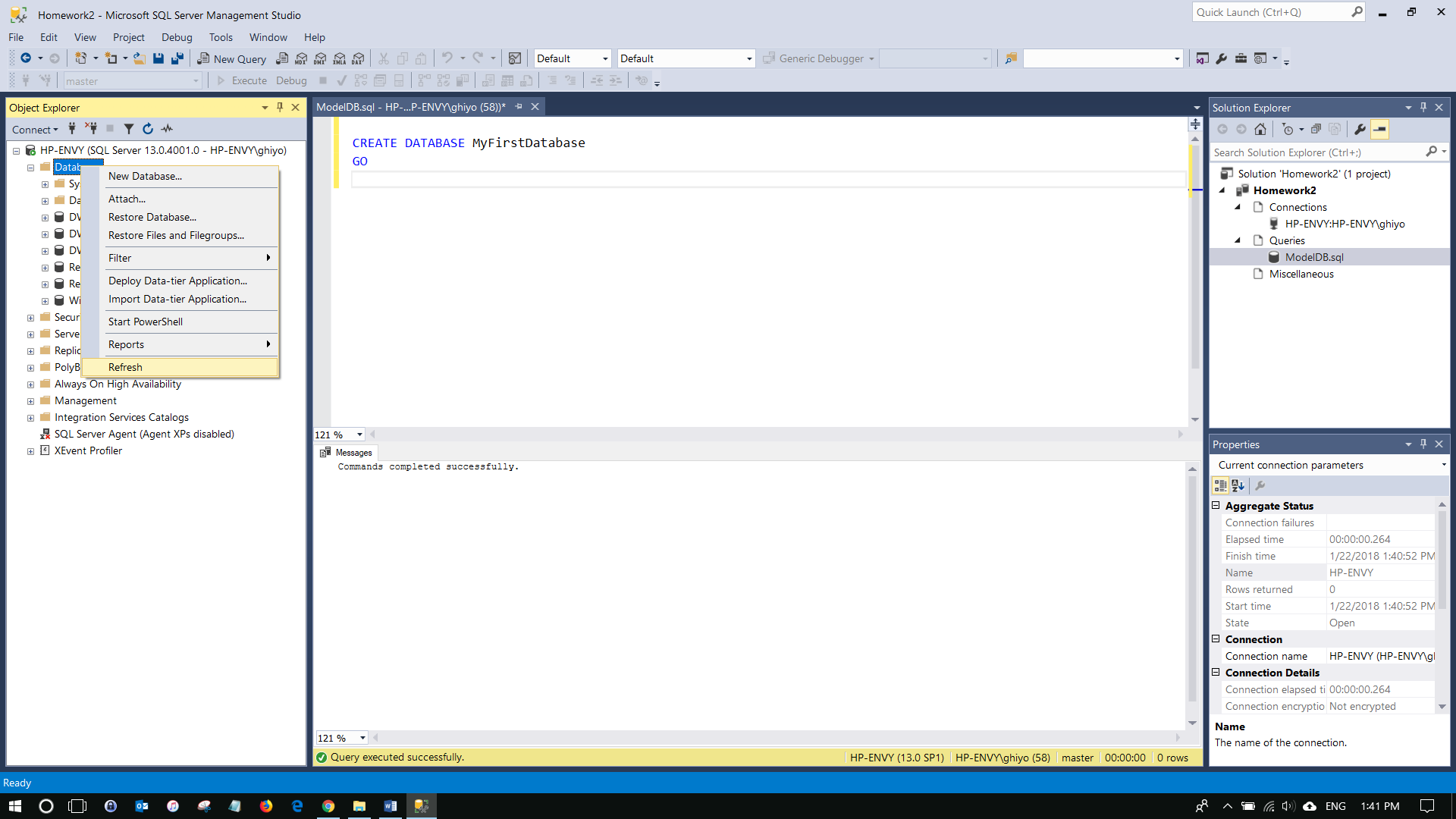


* **Create** the following database.

CREATE DATABASE MyFirstDatabase

GO

* Refresh **Databases** to confirm the creation.



* Change to **model database**.

USE model

GO

* Now the database to use is changed to **model**.
* Run the following query to create **DatabaseLog\_test** table in the **model** database.

CREATE TABLE dbo.DatabaseLog\_test (

EventId INT

, EventName SYSNAME

, EventType SYSNAME

, EventText NVARCHAR(3000)

)

GO

* Next create the second database.

CREATE DATABASE MySecondDatabase

GO

* **(Task 2: 5 points**) Show in a screenshot that **DatabaseLog\_test** table is created in **MySecondDatabase**, and **not** in **MyFirstDatabase**.
* After this task, **delete DatabaseLog\_test table** in the **model** database.

## Task #3: How to Create TempDB

* For a nice overview of temp tables, go to the following link. Review it carefully to answer the questions below (This is the same in SQL Server 2005 and later).

<http://www.codeproject.com/Articles/42553/Quick-Overview-Temporary-Tables-in-SQL-Server-2005>

* Create a new query file named **TempDB** *within the project* you created in Task #1.

/\* to view the temporary database file in the system \*/

USE tempdb

SELECT \* FROM sys.database\_files

/\* to create local temp tables. ‘#’means temp table.\*/

/\* available only to the current connection for the user \*/

CREATE TABLE #TempTableA (colA CHAR(8000))

CREATE TABLE #TempTableB (colB CHAR(8000))

CREATE TABLE #TempTableC (colC CHAR(8000))

/\* to create global temp tables \*/

/\* available to any user by any connection \*/

CREATE TABLE ##GlobTempTable (colD CHAR(8000))

/\* to view the existence of the temp tables that were just created \*/

SELECT \* FROM sys.objects

SELECT \* FROM sys.objects WHERE name like '#%'

/\* to insert values into the table \*/

DECLARE @i INT

SET @i = 1

WHILE @i < 5000

BEGIN

INSERT #TempTableA values ('Database Security is Fun!!' )

SET @i = @i +1

END

/\* to view the inserted entries\*/

SELECT \* FROM #TempTableA

/\* to drop temp tables \*/

DROP TABLE #TempTableA

DROP TABLE #TempTableB

DROP TABLE #TempTableC

DROP TABLE ##GlobTempTable

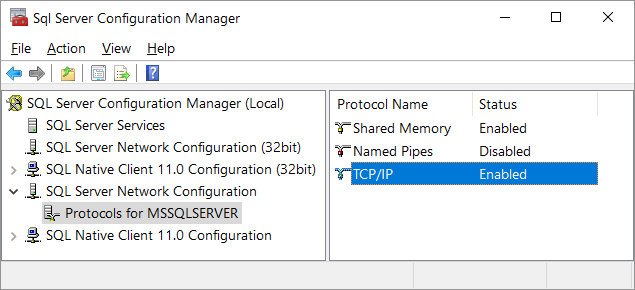
* (**Task 3: 4 points**) Create the three temp tables listed above and show their existence using the SELECT command above. Attach a screenshot.
* (**Question: 1 points**) When do we use temporary Tables? List three scenarios. When do we use table variable over temp table? For the answers, go to the link above.

# II. Configuration Manager

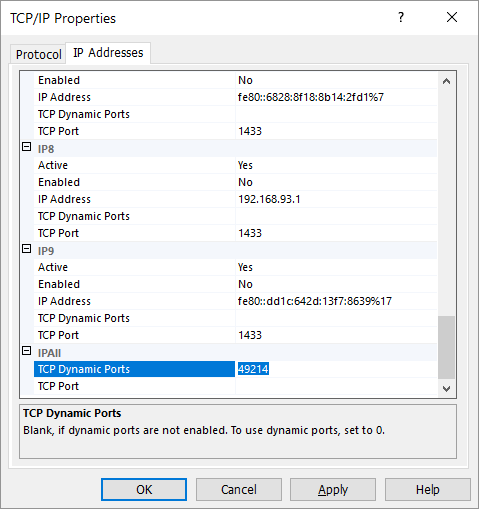
For Configuration Manager, go to **Configuration Tools** > **SQL Server Configuration Manager**

## Task #4: How to Connect Using Different Protocols/Ports

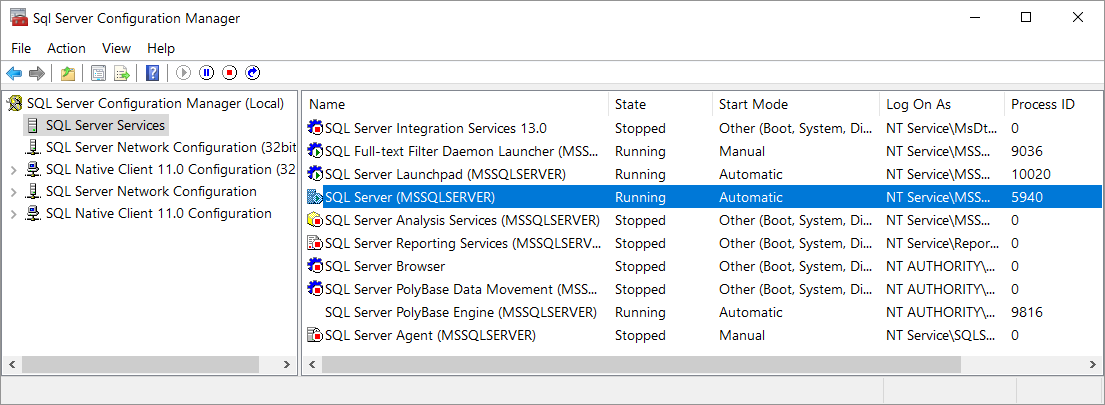
* Make sure that **TCP/IP** is enabled for the default instance. If you are connecting to the default instance that was created in SQL Server 2016, your TCP/IP connection will use the default port (1433).



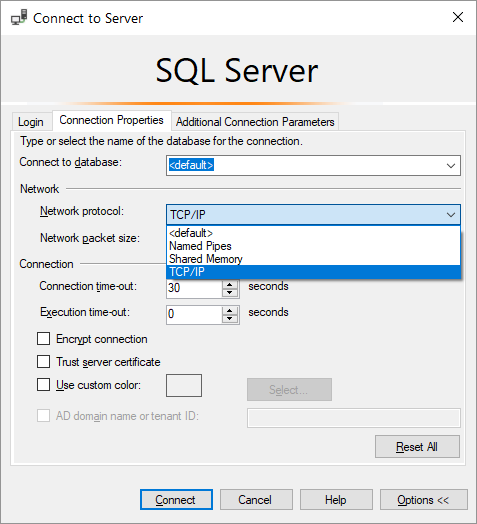
* To connect using a port other than the default port, we have to use **TCP Dynamic Ports**. We are going to connect using the port **49214**. Find the dynamic port number in **IPAll** for TCP/IP connection and change it to **49214**. And make the TCP port in IPAll **blank**.



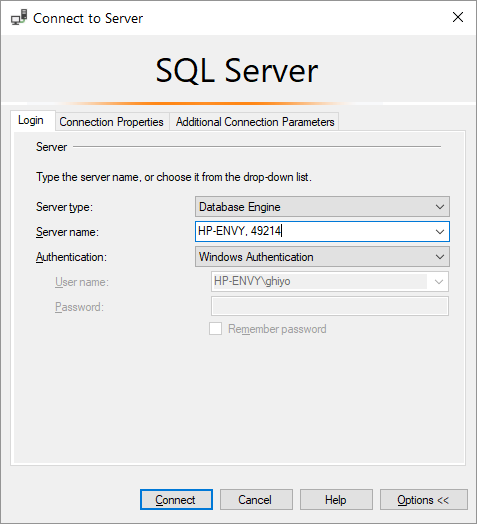
* For the changes to take effect, we must restart the SQL Server Service (MSSQLSERVER) using Configuration Manager. Right-click on it and select **Restart**.



* Re-launch MS Management Studio and **Connection Properties**. Change the **network protocol** to TCP/IP.



* Add the port number (**, 49214**) after the host to the box for the Server name.



* You now are connected to the Server using the port 49214. Please maintain the connection for the following task.
* Please understand what SPIDs are about after reading the following.

<http://sqlserverplanet.com/dba/spid-what-is-it/>

“A SPID in SQL Server is a Server Process ID. These process ID’s are essentially sessions in SQL Server. Everytime an application connects to SQL Server, a new connection (or SPID) is created. This connection has a defined scope and memory space and cannot interact with other SPIDs. The term SPID is synonymous with Connection, or Session.”

* Please understand what IDs and SIDs are about after reading the following.

<http://msdn.microsoft.com/en-us/library/ms403629(v=sql.105).aspx>

* Execute the following. Create a sql file named **SPID** to place the following queries.

/\* to see the entire SPIDs \*/

SELECT \*

FROM sys.dm\_exec\_sessions

/\* to see your SPID \*/

Select \* from sysprocesses

Where spid=@@spid

/\* to see the connections \*/

SELECT \* FROM sys.dm\_exec\_connections

/\* to see the principals \*/

select \* from sys.server\_principals

* (**Task 4: 5 points**) Fill in **all** the blanks. Keep in mind that you have made a connection via a dynamic port not the default number. Your answers to the second column must be consistent with the connection you made.

|  |  |  |
| --- | --- | --- |
|  | **Name/Number** | **Description (definition)** |
| **SPID** |  |  |
| **SID** |  |  |
| **principal\_id** |  |  |
| **net\_library** |  |  |
| **local\_tcp\_port** |  |  |

In addition, show the result of each of the four queries in a screenshot.

*You can obtain the necessary info by following the following procedure:*

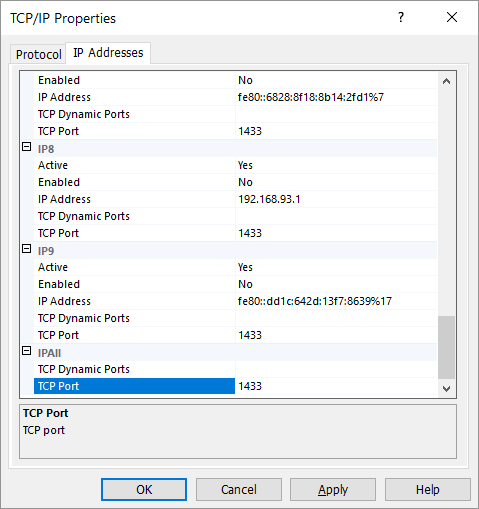
First, obtain your **SPID** by running the *second* query above. From the result, you can obtain **sid** and **net\_library**. net\_library must be TCP/IP. Otherwise you are wrong.

Second, run the *third* query and **locate your SPID**. And obtain **local\_tcp\_port**. It must be 49214!

Third, run the *first* query and find your **session\_id**, which is **SPID**. Find the **host name**. Host name is basically your login.

Last, run the *last* query and locate the **host name**. And find the **principal\_id**.

* After this step, you must restore to the default setting. Go to **IPAll** and make **TCP Dynamic Ports** Blank and **TCP Port** 1433. Finally, restart the service.



1. Source: www.microsoft.com [↑](#footnote-ref-1)
2. Source: SQL Server books online [↑](#footnote-ref-2)