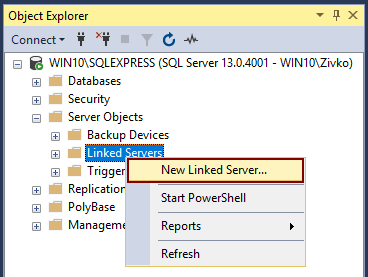
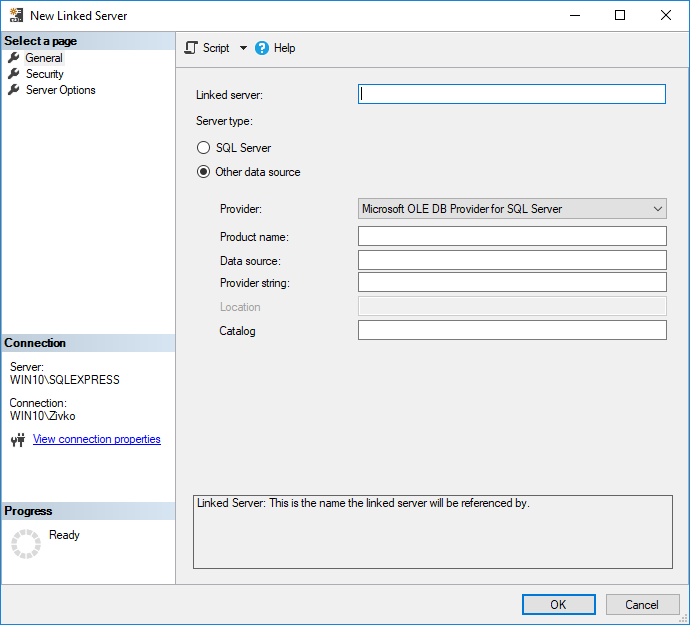
# Create a linked server to an Azure SQL database via SSMS

To create a linked server via [SSMS](https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms), go to **Object Explorer** under the **Server Objects** folder, right click on the **Linked Servers** folder and from the context menu choose the **New Linked Server** command:

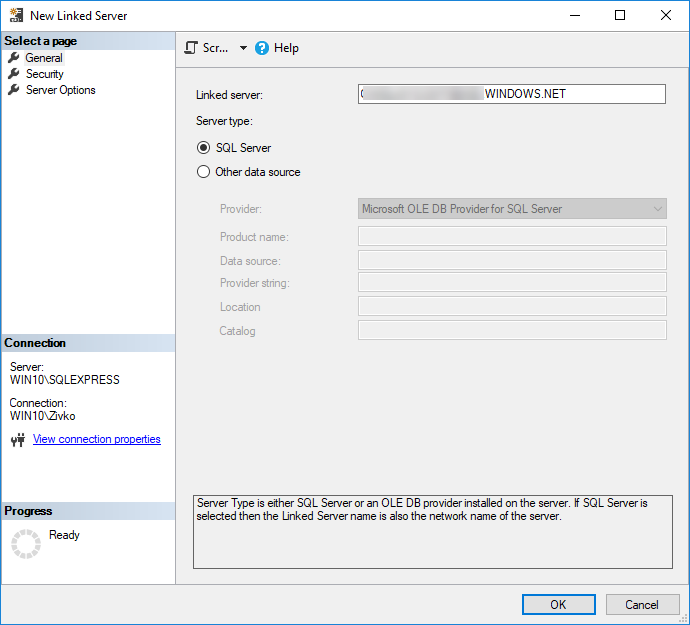


The **New Linked Server** window will be opened:

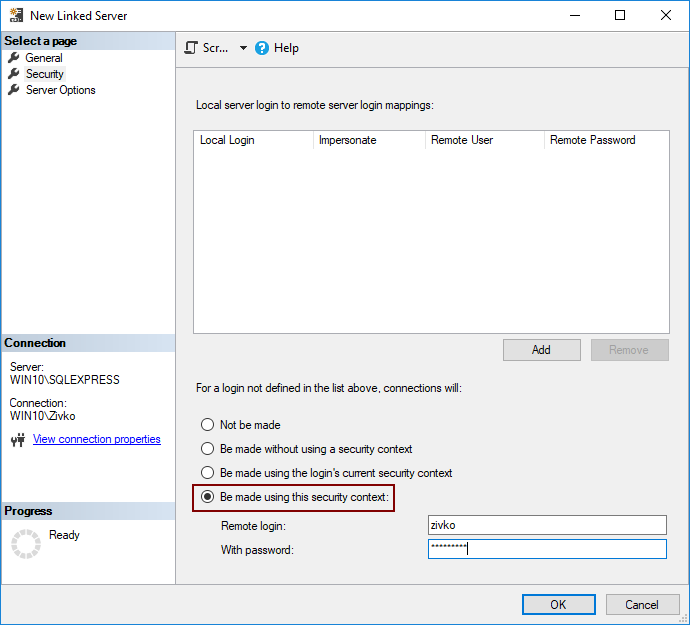


In this window, the first thing that needs to be done is to enter a name for a linked server in the **Linked server** box and to choose the server type by clicking the **SQL Server** or **Other data source** radio button under the **General** tab.

Let’s first choose the **SQL Server** radio button under the **Server type** section and see what will happen. Also, on the **General** tab in the **Linked server**box, enter the name of the Azure SQL server for which a linked server is created:

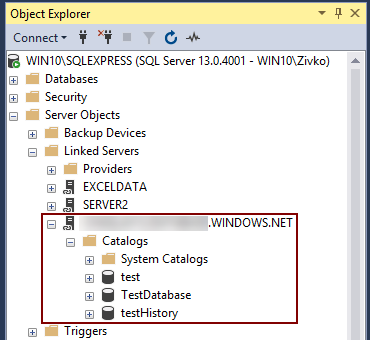


Under the**Security** tab, select the **Be made using this security context** radio button and enter user credentials that exist on Azure server:



More about the **Security** and **Server Options** tab can be found on the [How to create and configure a linked server in SQL Server Management Studio](https://www.sqlshack.com/how-to-create-and-configure-a-linked-server-in-sql-server-management-studio/) article.

After entering the user credentials, press the **OK** button to create a linked server to an Azure SQL database. Now, under the **Linked Servers** folder, Azure linked server that we created will appear and in the **Catalogs** folder, all available databases will be listed:



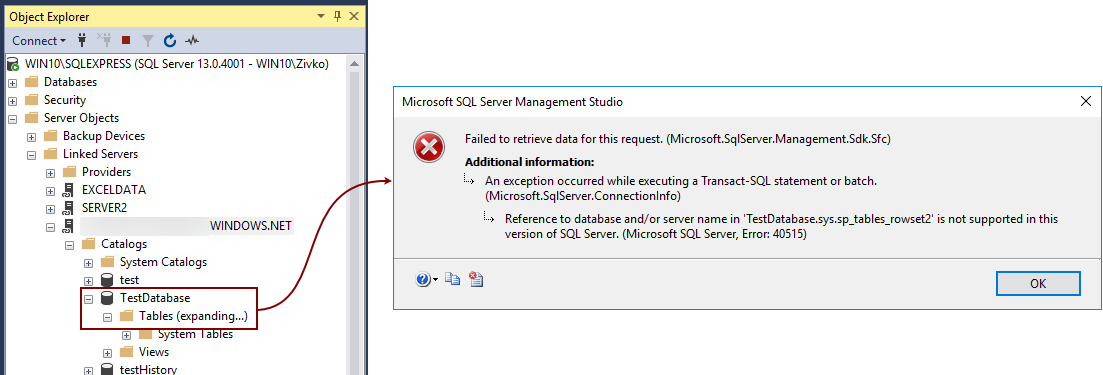
But, when expanding a particular database (e.g.TestDatabase) is needed in order to see tables of the database, the following error message will appear:

Failed to retrieve data for this request. (Microsoft.SqlServer.Management.Sdk.Sfc)

ADDITIONAL INFORMATION:

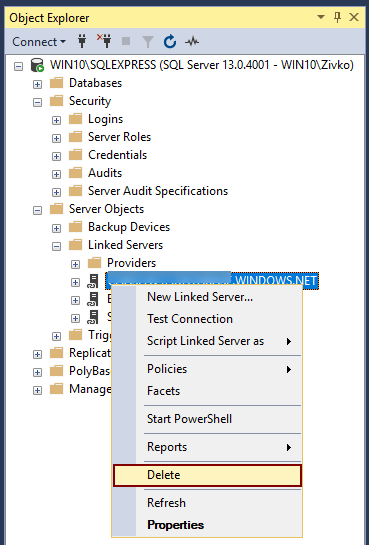
An exception occurred while executing a Transact-SQL statement or batch. (Microsoft.SqlServer.ConnectionInfo)

Reference to database and/or server name in ‘TestDatabase.sys.sp\_tables\_rowset2’ is not supported in this version of SQL Server. (Microsoft SQL Server, Error: 40515)

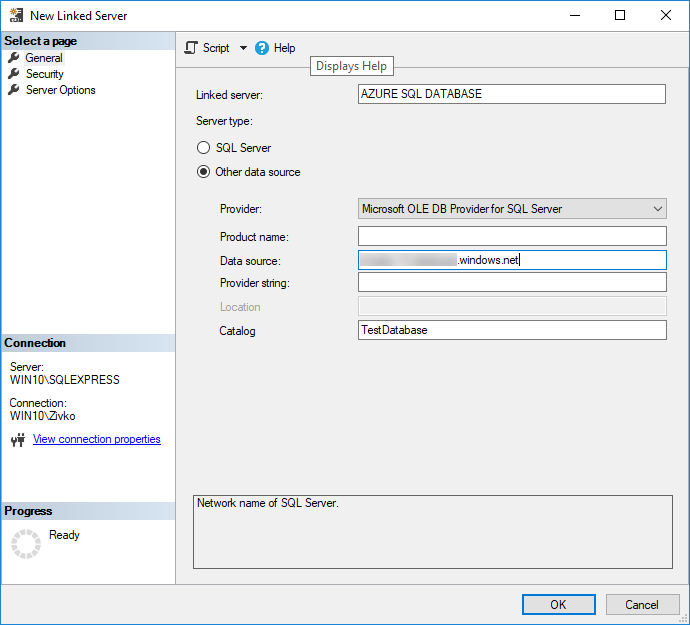
[](https://www.sqlshack.com/wp-content/uploads/2017/09/c-users-marko-appdata-local-microsoft-windows-ine-5.png)

This error occurs because Azure does not allow to alter the master database. To resolve this, you need to connect directly to the Azure database you are going to use.

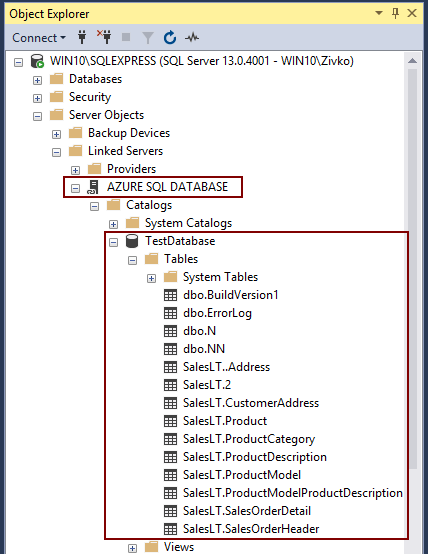
Delete the SQL Azure linked server that we created and create a new one:



This time, under the **Server type** section of the **General** tab, choose the **Other data source** radio button. The name for the linked server in the **Linked server** text box can be, this time, whatever you like (e.g. AZURE SQL DATABASE). Under the **Provider** drop down box, choose the**Microsoft OLE DB Provider SQL Server** item. In the **Data source** text box, enter the name of the SQL Azure (e.g. server.database.windows.net). The most **important** setting in order to correctly create a linked server to an Azure SQL database is to enter the name in the **Catalog** text box (e.g. TestDatabase) of an Azure SQL database for which you want to create a linked server to an Azure SQL database. Otherwise, if this field is left empty, we will encounter the same **40515** error when trying to get a list of the tables under the **Catalogs** folder.



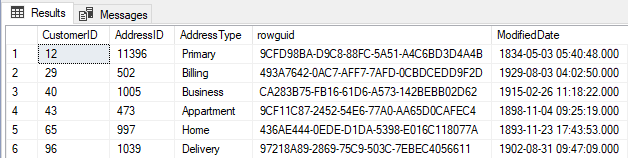
Under the**Security** tab, use the same setting that we used in the previous example and press the **OK** button. This will create a linked server to an Azure SQL database (TestDatabase database), and when the plus (+) sign next to the Tables folder is pressed, the Tables folder will expand and show all tables for the TestDatabase database:



To retrieve data from the SQL Azure table (e.g. CustomerAddress), type the following code:

|  |
| --- |
| SELECT \* FROM [AZURE SQL DATABASE].[TestDatabase].[SalesLT].[CustomerAddress] |

If everything goes well, the data from the CustomerAddress table will appear in the **Results** grid:

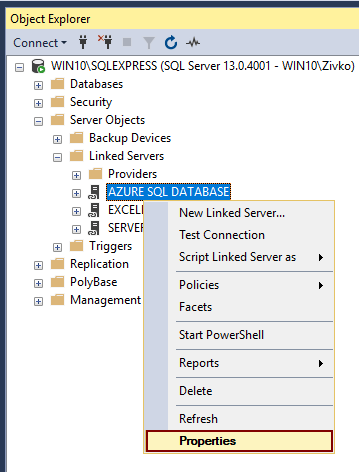


Now, if you try to execute a stored procedure from the AZURE SQL DATABASE linked server, the following message may appear:

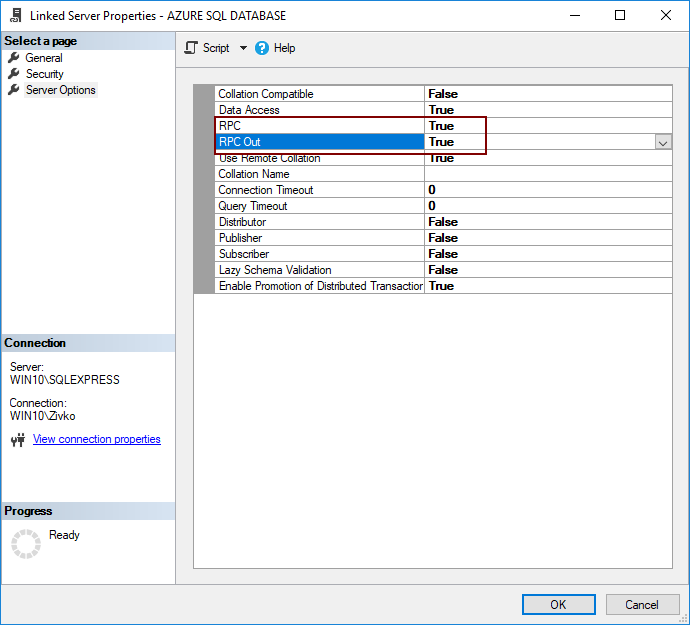
Msg 7411, Level 16, State 1, Line 48  
Server ‘AZURE SQL DATABASE’ is not configured for RPC.

This is because the RPC and RCP Out ([Remote Procedure Call](https://blogs.msdn.microsoft.com/jason_howell/2014/10/01/what-is-the-rpc-and-rpc-out-option-on-a-sql-server-linked-server/)) options that allow remote procedures to be called from the linked server or to be called to the linked server by default set to false.

To fix that, right click on the AZURE SQL DATABASE linked server, choose the **Properties** options:



Under the **Server Options** tab, set the **RPC** and **RPC Out** options to **True:**



Or in a query editor, paste and execute the following code:

|  |
| --- |
| EXEC master.dbo.sp\_serveroption @server=N'AZURE SQL DATABASE', @optname=N'rpc', @optvalue=N'true'  GO  EXEC master.dbo.sp\_serveroption @server=N'AZURE SQL DATABASE', @optname=N'rpc out', @optvalue=N'true'  GO |

OPENQUERY function

The [OPENQUERY](https://docs.microsoft.com/en-us/sql/t-sql/functions/openquery-transact-sql) function can be used to get data from the linked server by executing code like this:

|  |
| --- |
| SELECT \* FROM OPENQUERY([AZURE SQL DATABASE],'SELECT \* FROM SalesLT.CustomerAddress') |

The results will be the same as from the example above.

This is the ad hoc method for connection to a remote server using the linked server and querying data from a remote server. If the connection to remote server is frequently used, then using the linked server is better solution instead of using the OPENQUERY function.

This function can be used in the [FROM](https://docs.microsoft.com/en-us/sql/t-sql/queries/from-transact-sql) clause of the [INSERT](https://docs.microsoft.com/en-us/sql/t-sql/statements/insert-transact-sql), [SELECT](https://docs.microsoft.com/en-us/sql/t-sql/queries/select-transact-sql), [DELETE](https://docs.microsoft.com/en-us/sql/t-sql/statements/delete-transact-sql) or [UPDATE](https://docs.microsoft.com/en-us/sql/t-sql/queries/update-transact-sql) statement.

Create a linked server to an Azure SQL database using Transact-SQL

In order to create a linked server to an Azure SQL database, type the following code in a query editor:

|  |
| --- |
| EXEC master.dbo.sp\_addlinkedserver  @server = N'AZURE SQL DATABASE',  @srvproduct=N'',    @provider=N'SQLNCLI',     @datasrc=N'server.database.windows.net',      @catalog=N'TestDatabase'  /\* For security reasons, the linked server remote logins password is changed with ######## \*/  EXEC master.dbo.sp\_addlinkedsrvlogin  @rmtsrvname=N'AZURE SQL DATABASE',  @useself=N'False',  @locallogin=NULL,  @rmtuser=N'zivko',@rmtpassword='#########'  GO |

More about how to set linked server by using T-SQL can be found on the [How to create, configure and drop a SQL Server linked server using Transact-SQL](https://www.sqlshack.com/create-configure-drop-sql-server-linked-server-using-transact-sql/) page

Connecting to Azure SQL database and querying data using the distributed queries

To connect to Azure SQL database and access data without creating a linked server first, use the T-SQL [OPENROWSET](https://docs.microsoft.com/en-us/sql/t-sql/functions/openrowset-transact-sql) or [OPENDATASOURCE](https://docs.microsoft.com/en-us/sql/t-sql/functions/opendatasource-transact-sql) functions.

To open a connection and querying data from the Azure SQL database using the [OPENROWSET](https://docs.microsoft.com/en-us/sql/t-sql/functions/openrowset-transact-sql) function, type the following code in a query editor:

|  |
| --- |
| SELECT \* FROM OPENROWSET('SQLNCLI', 'Server=server.database.windows.net;Database=TestDatabase;UID=zivko;PWD==######;', 'SELECT \* FROM SalesLT.CustomerAddress') |

If, for some reasons, the above code does not work, use the code below to connect and query data from Azure SQL database:

|  |
| --- |
| SELECT \* FROM  OPENROWSET('MSDASQL', 'Driver={SQL SERVER}; Server=server.database.windows.net;Database=TestDatabase;UID=zivko; PWD=######;', 'SELECT \* FROM SalesLT.CustomerAddress') |

Another way of connecting and querying data from the Azure SQL database is by using the [OPENDATASOURCE](https://docs.microsoft.com/en-us/sql/t-sql/functions/opendatasource-transact-sql) function.

In a query editor, paste and execute one of the following codes:

|  |
| --- |
| SELECT \*  FROM OPENDATASOURCE('MSDASQL', 'Driver={SQL SERVER}; Server=server.database.windows.net;Database=TestDatabase;UID=zivko;PWD==######;').TestDatabase.SalesLT.CustomerAddress |

Or

|  |
| --- |
| SELECT \* FROM OPENDATASOURCE('SQLNCLI', 'Server=server.database.windows.net;Database=TestDatabase;UID=zivko;PWD==######;').TestDatabase.SalesLT.CustomerAddress |

Common error that may occur when using the T-SQL [OPENROWSET](https://docs.microsoft.com/en-us/sql/t-sql/functions/openrowset-transact-sql) and [OPENDATASOURCE](https://docs.microsoft.com/en-us/sql/t-sql/functions/opendatasource-transact-sql) functions:

Msg 15281, Level 16, State 1, Line 1

SQL Server blocked access to STATEMENT ‘OpenRowset/OpenDatasource’ of component ‘Ad Hoc Distributed Queries’ because this component is turned off as part of the security configuration for this server. A system administrator can enable the use of ‘Ad Hoc Distributed Queries’ by using sp\_configure. For more information about enabling ‘Ad Hoc Distributed Queries’, search for ‘Ad Hoc Distributed Queries’ in SQL Server Books On

To resolve this the [Ad Hoc Distributed Queries](https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/ad-hoc-distributed-queries-server-configuration-option) option should be enabled. To enable the Ad Hoc Distributed Queries option, use the [sp\_configure](https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-configure-transact-sql" \t "_blank) procedure and in a query editor, paste and execute the following code:

|  |
| --- |
| EXEC sp\_configure 'show advanced options', 1  RECONFIGURE  GO  EXEC sp\_configure 'ad hoc distributed queries', 1  RECONFIGURE  GO |