

My First Web Service with WCF in Azure

STEP 1 – WE NEW CONNECT WITH OUR DATA

- Now, we need add our data azure server

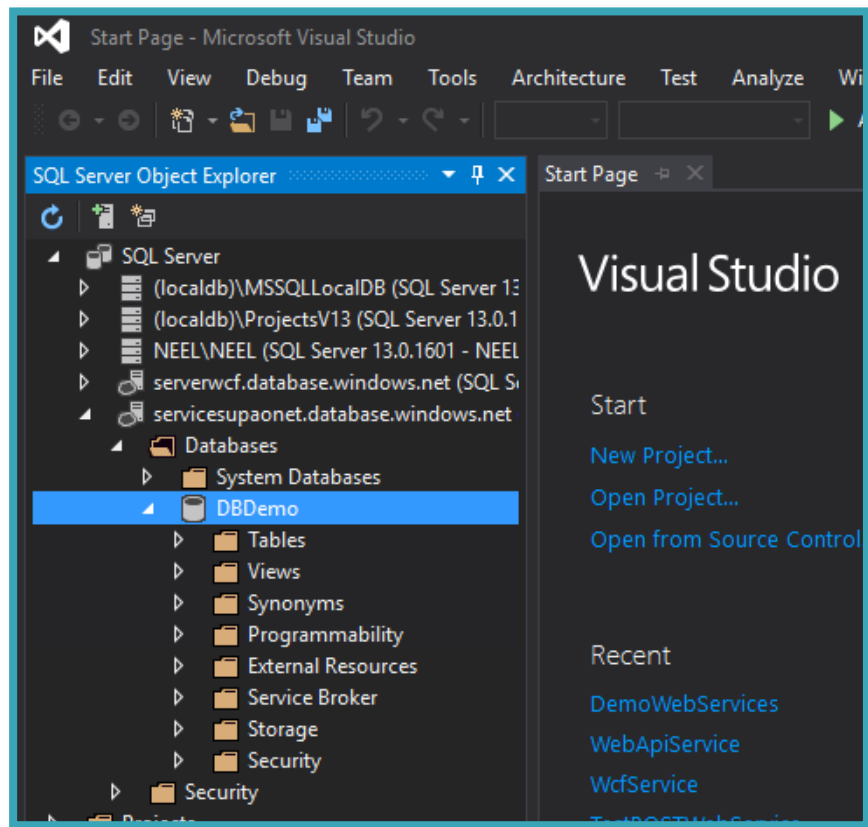
The screenshot shows the 'Connect' dialog box with the following fields and values:

- Server Name:** servicesupaonet.database.windows.net
- Authentication:** Sql Server Authentication
- User Name:** neel
- Password:** (masked with dots)
- ☐ Remember Password
- Database Name:** DBDemo

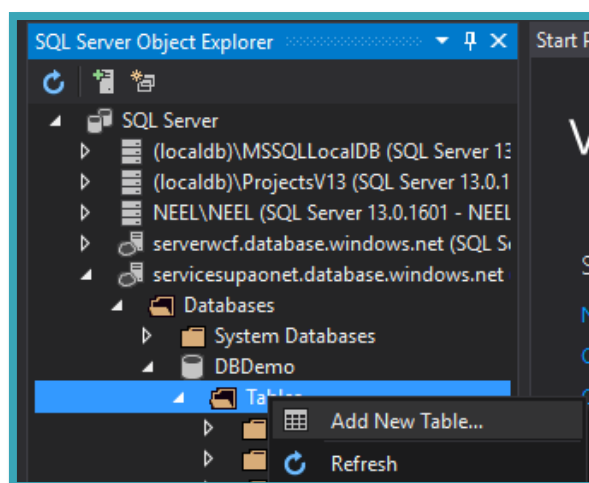
Buttons at the bottom: Connect, Cancel

STEP 2 – WE CHECK OUR CONNECTION TO OUR DATABASE IN AZURE FROM VISUAL STUDIO

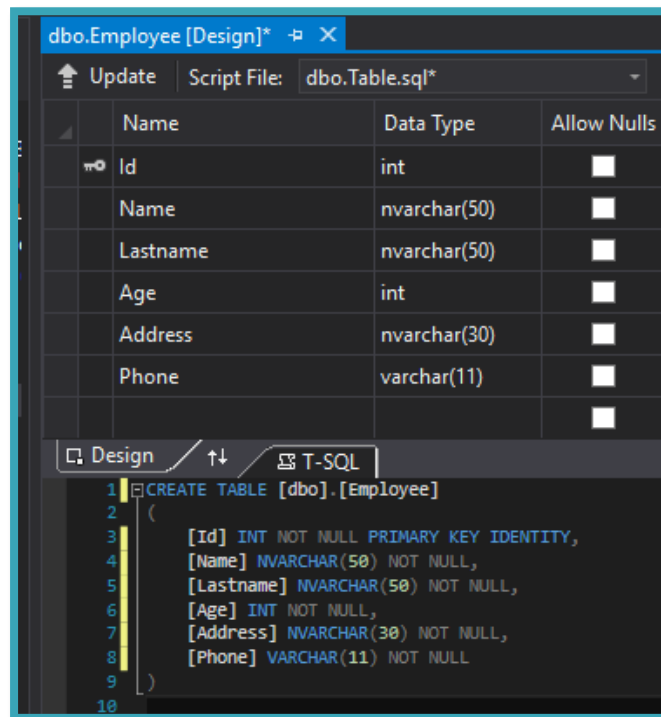
- that's it, now we can use our azure database from visual studio



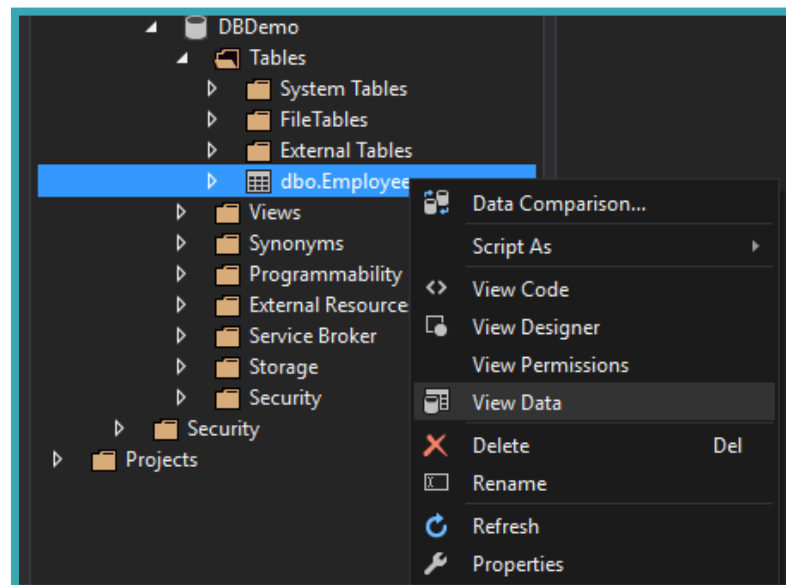
STEP 3 – WE NEED CREATE A TABLE WITHIN OF OUR DATABASE



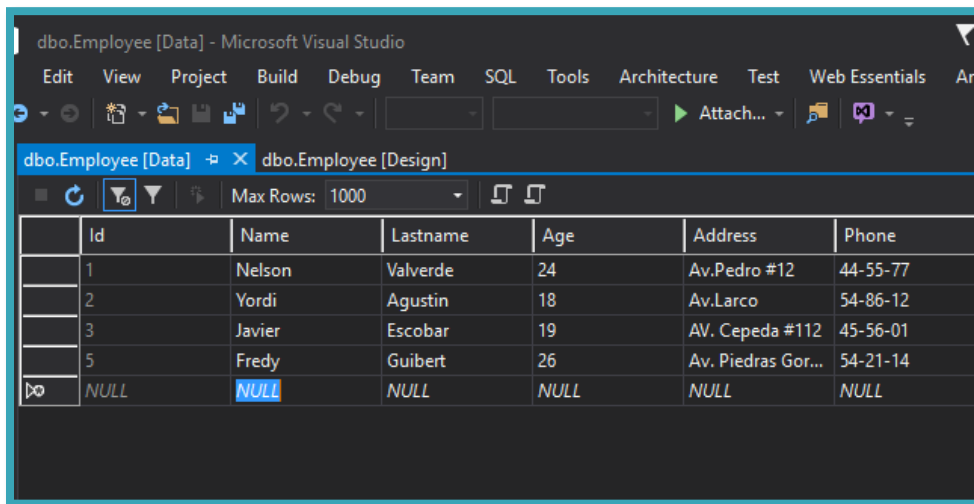
- We need create a table **Employee** and update database



- Then add data in our table



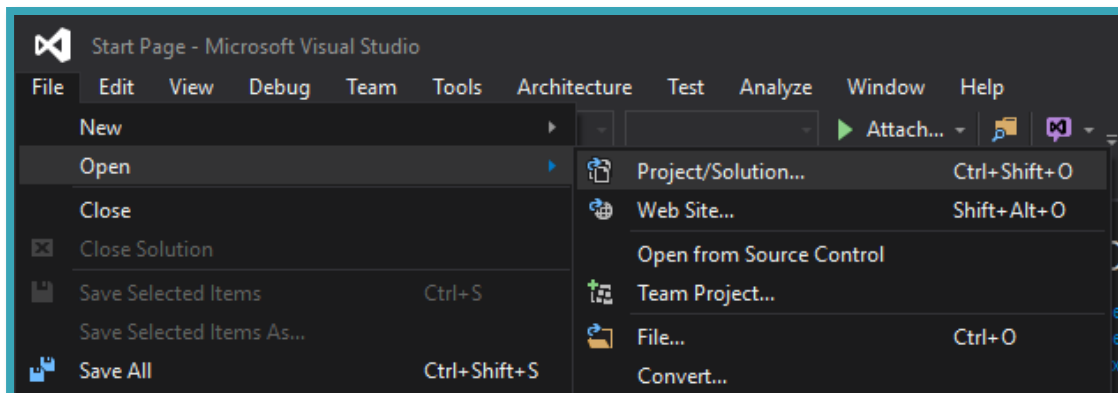
- This is my table **Employee** with my data



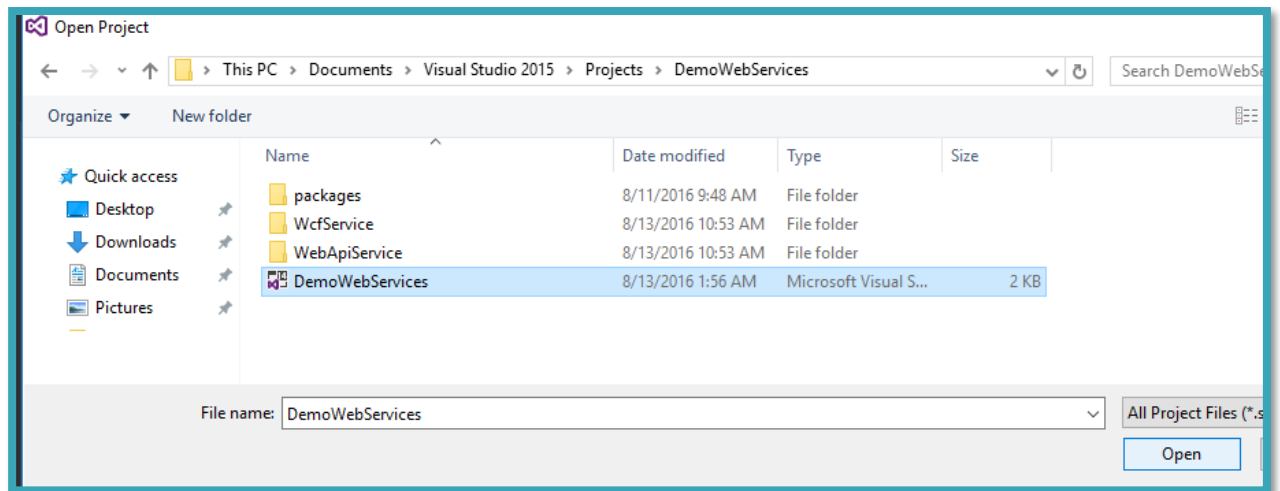
	Id	Name	Lastname	Age	Address	Phone
	1	Nelson	Valverde	24	Av. Pedro #12	44-55-77
	2	Yordi	Agustin	18	Av. Larco	54-86-12
	3	Javier	Escobar	19	Av. Cepeda #112	45-56-01
	5	Fredy	Guibert	26	Av. Piedras Gor...	54-21-14
	NULL	NULL	NULL	NULL	NULL	NULL

STEP 4 - WE PREPARED OUR PROJECT OR SOLUTION

- We Open our Solution

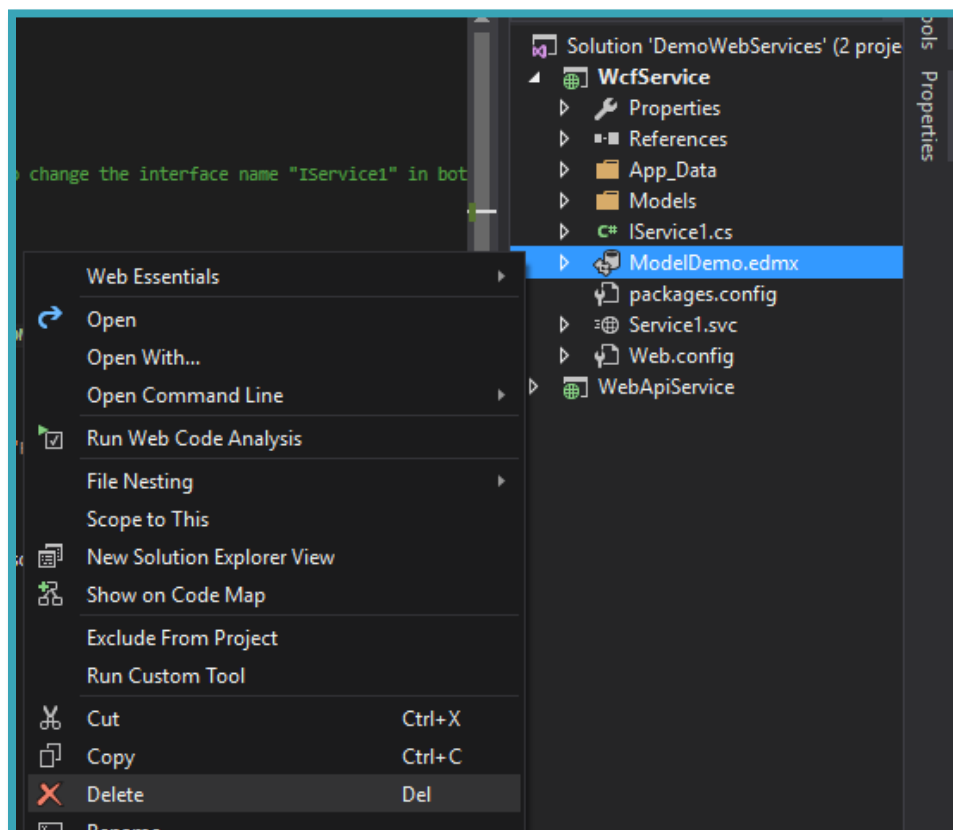


- We select our project and open

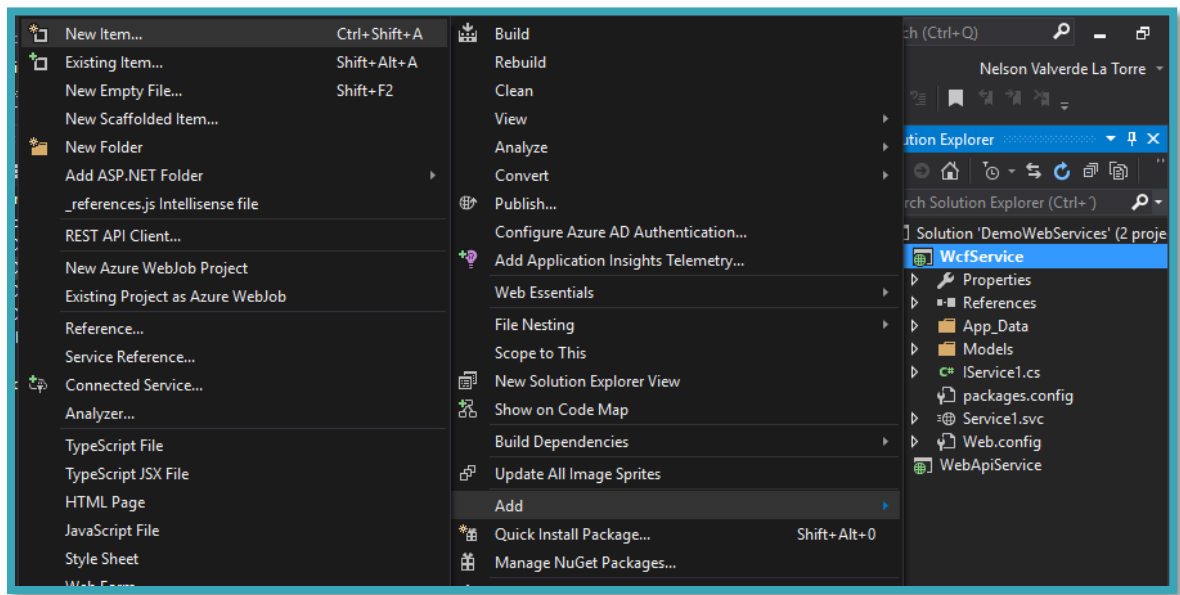


- Delete our ModelDemo.edmx

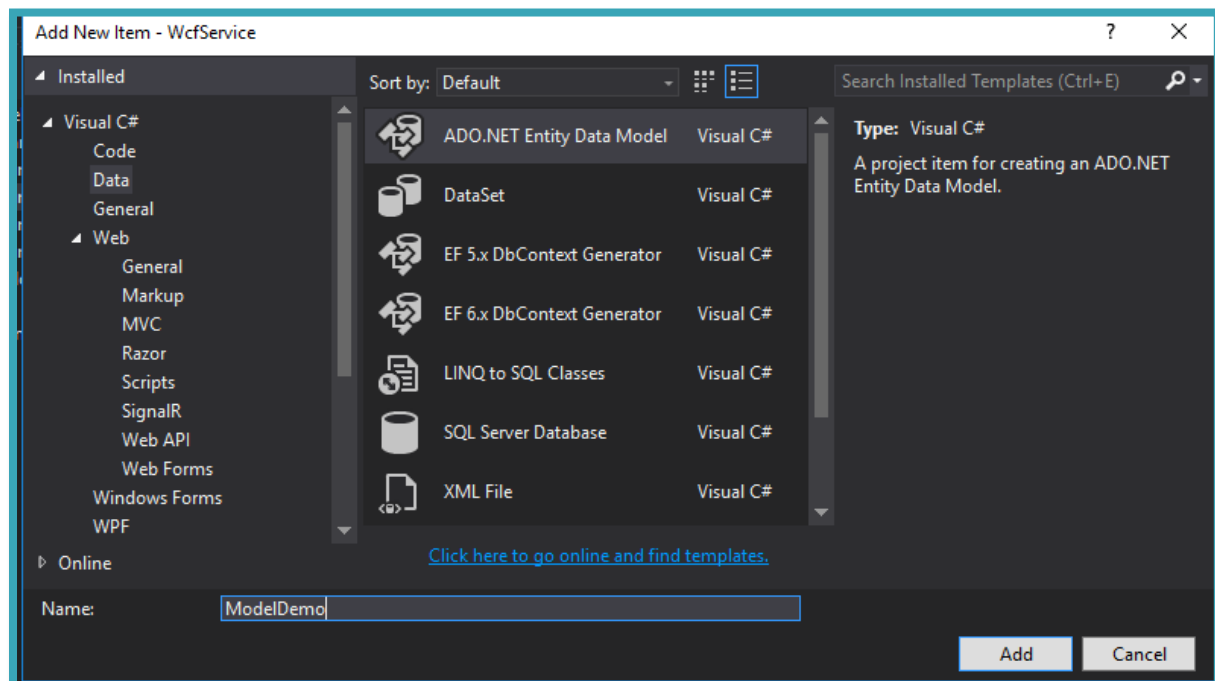
This is our Local Web service of our previous project



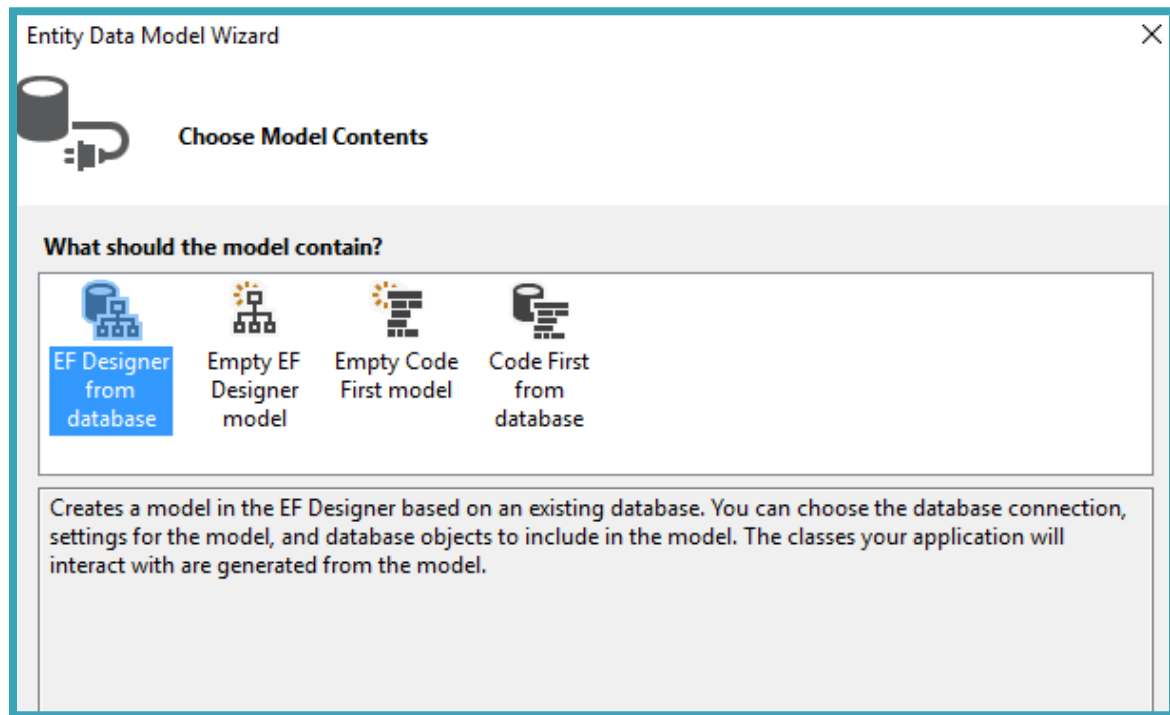
We Add a new item



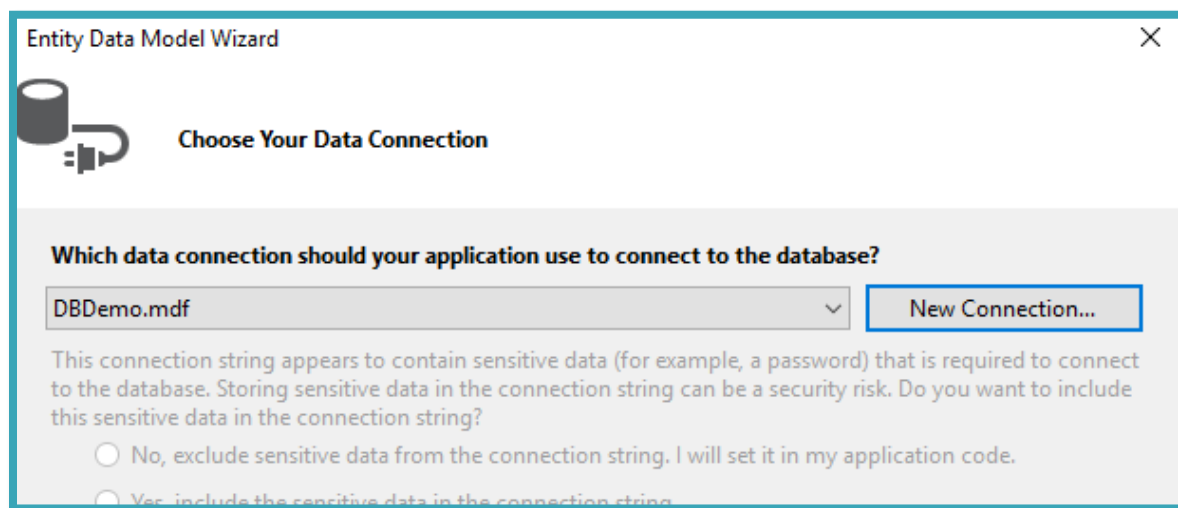
Now we create our new ModelDemo.edmx



We select EF Designer from database and Next



We select New connection



Now we add our information of azure database

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
servicesupaonet.database.windows.net Refresh

Log on to the server

Authentication: SQL Server Authentication

User name: neel

Password: ••••••••

☐ Save my password

Connect to a database

☒ Select or enter a database name:
DBDemo

☐ Attach a database file:
Browse...

Logical name:

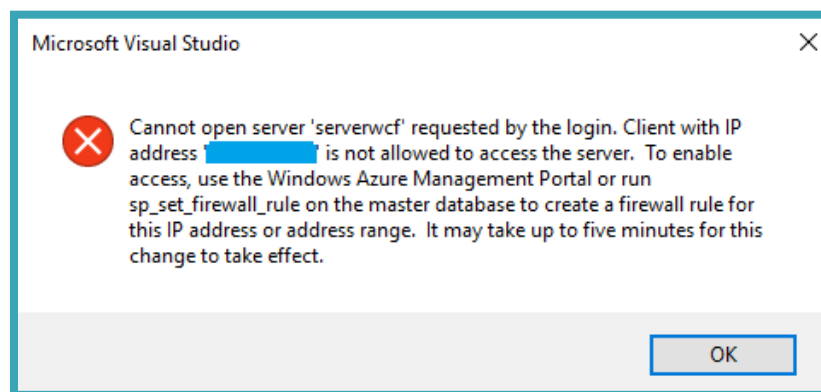
Advanced...

Test Connection OK Cancel

IMPORTANT

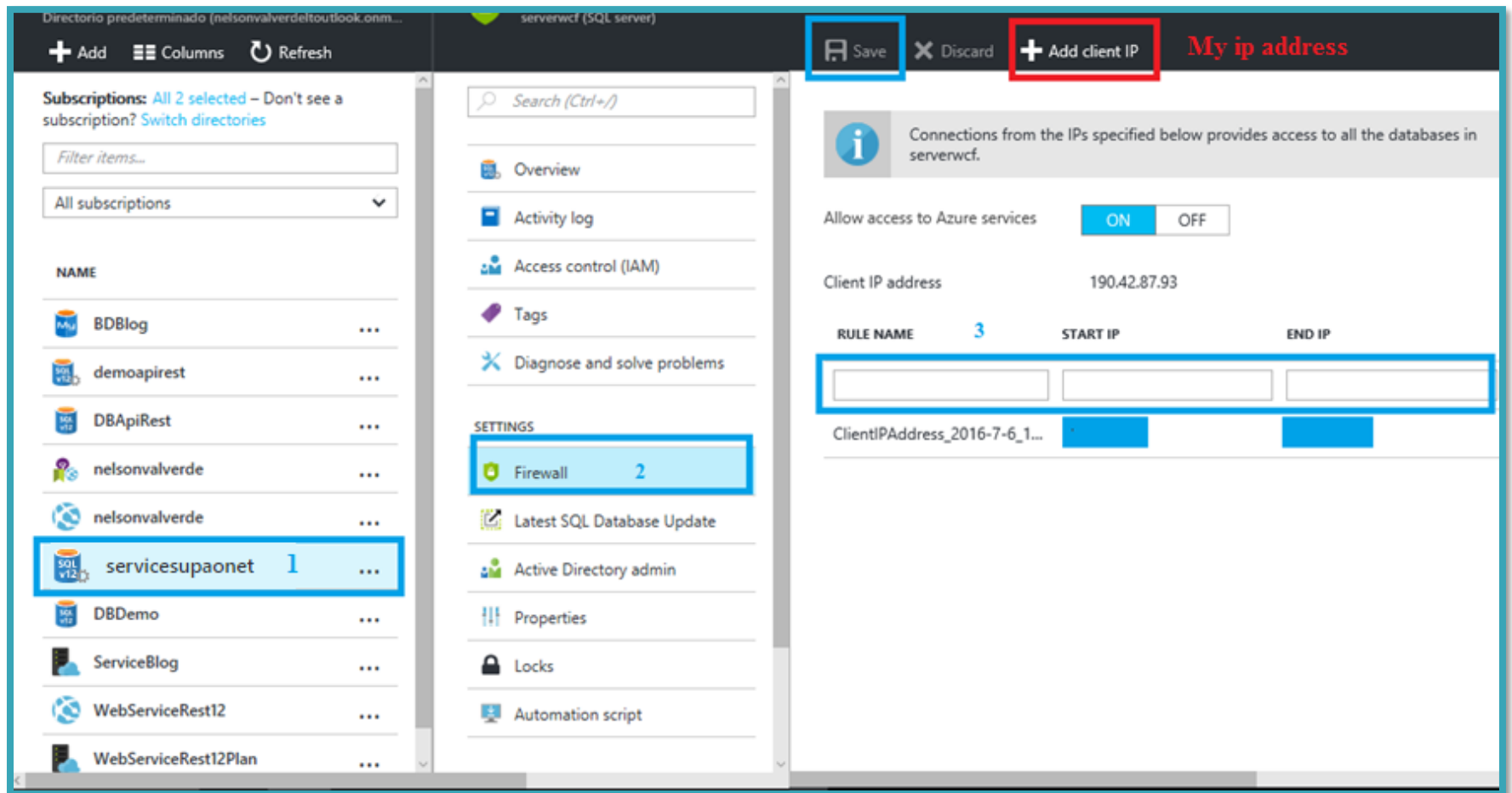
Solution N° 1:

If you have this error, you need add our IP in the section of firewall of our azure database



You have to do this:

Add your ip public and save or add a custom ip and Save, then Follow these steps.



Solution N° 2:

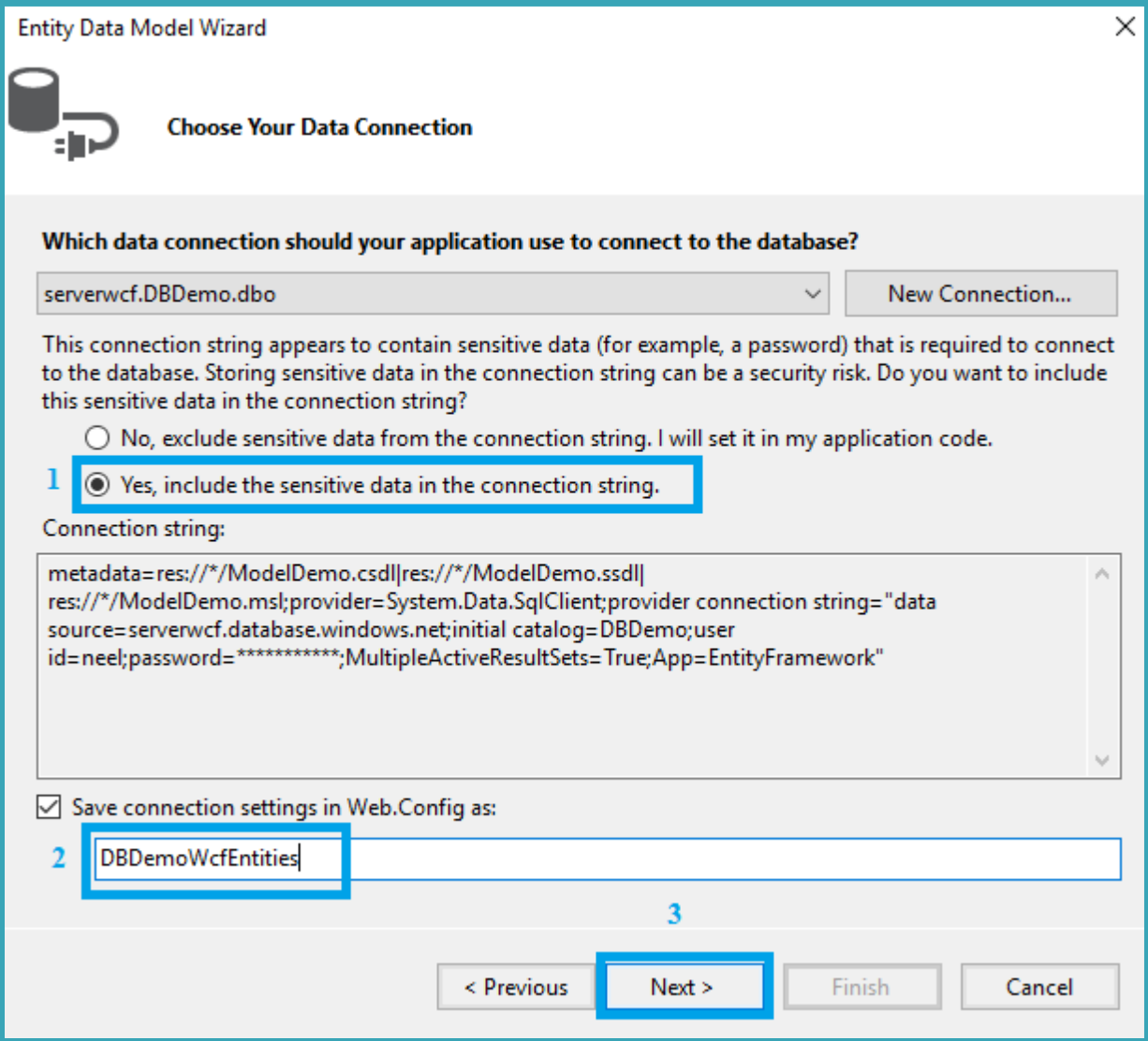
You don't know about your data connection of azure? Get in here, then copy and paste in our Connection Properties of the Page 8

The screenshot displays the Microsoft Azure portal interface. The left sidebar contains navigation links: Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases (highlighted with a blue box and a '1'), Cloud services (classic), Security Center, Subscriptions, and More Services. The middle pane shows the 'SQL databases' view for the 'DBDemo' resource. It lists two databases: 'DBApiRest' and 'DBDemo' (highlighted with a blue box and a '2'). The right pane shows the 'DBDemo - Properties' page. It includes a search bar, a list of tabs (Overview, Activity log, Tags, Diagnose and solve problems), and a 'SETTINGS' section with various options like Pricing tier, Geo-Replication, Auditing & Threat detection, Dynamic data masking, Transparent data encryption, Properties (highlighted with a blue box and a '3'), Locks, and Automation script. The 'Properties' tab is active, showing details such as 'SQL_Latin1_General_CP1_CI_AS', 'CREATION DATE' (8/13/2016 9:50:54 AM), 'CONNECTION STRINGS' (with a link to 'Show database connection strings'), 'SERVER NAME' (servicesupaonet.database.windows.net, highlighted with a blue box), 'LOCATION' (East US 2), 'SERVER ADMIN LOGIN' (neel, highlighted with a blue box), and 'ACTIVE DIRECTORY ADMIN' (Not configured).

We need do this:

1- this allows add a string connection in our web.config.

2 - this allows add a string connection in our web.config and allows connect our azure database.



The image shows a screenshot of the 'Entity Data Model Wizard' window, specifically the 'Choose Your Data Connection' step. The window has a title bar with the text 'Entity Data Model Wizard' and a close button. Below the title bar is a logo of a database cylinder and a plug icon, followed by the text 'Choose Your Data Connection'. The main content area is titled 'Which data connection should your application use to connect to the database?'. It features a dropdown menu with 'serverwcf.DBDemo.dbo' selected and a 'New Connection...' button. Below this, a text block explains that the connection string may contain sensitive data and asks if it should be included. Two radio buttons are present: 'No, exclude sensitive data from the connection string. I will set it in my application code.' and 'Yes, include the sensitive data in the connection string.' The second option is selected and highlighted with a blue box and the number '1'. Below the radio buttons, the 'Connection string:' is displayed in a text area, showing a complex string for connecting to a SQL Server database. At the bottom, there is a checkbox labeled 'Save connection settings in Web.Config as:' which is checked. Below the checkbox is a text box containing 'DBDemoWcfEntities', highlighted with a blue box and the number '2'. At the very bottom, there are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted with a blue box and the number '3'.

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

serverwcf.DBDemo.dbo New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

1 ☒ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/ModelDemo.csdl|res://*/ModelDemo.ssdl|
res://*/ModelDemo.msl;provider=System.Data.SqlClient;provider connection string="data
source=serverwcf.database.windows.net;initial catalog=DBDemo;user
id=neel;password=*****;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in Web.Config as:


2 DBDemoWcfEntities

3

< Previous Next > Finish Cancel


- We need add our tables of our database
- Usually, you want entity names singular and entity set names plural.
- This is necessary specify If you have a database relational


Entity Data Model Wizard ×


 Choose Your Database Objects and Settings


Which database objects do you want to include in your model?


1

☒  Tables

☒  dbo

☒  Employee

☐  Views

☐  Stored Procedures and Functions

2

☒ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

☐ Import selected stored procedures and functions into the entity model

3

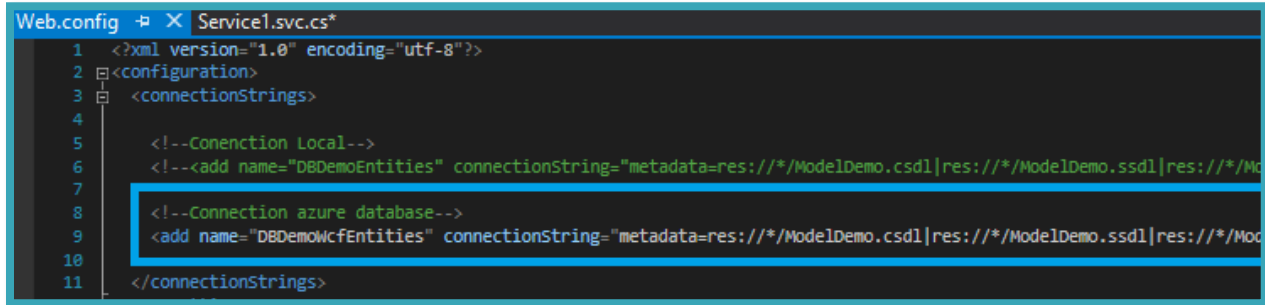
Model Namespace:

DBDemoModel

4

< Previous Next > **Finish** Cancel

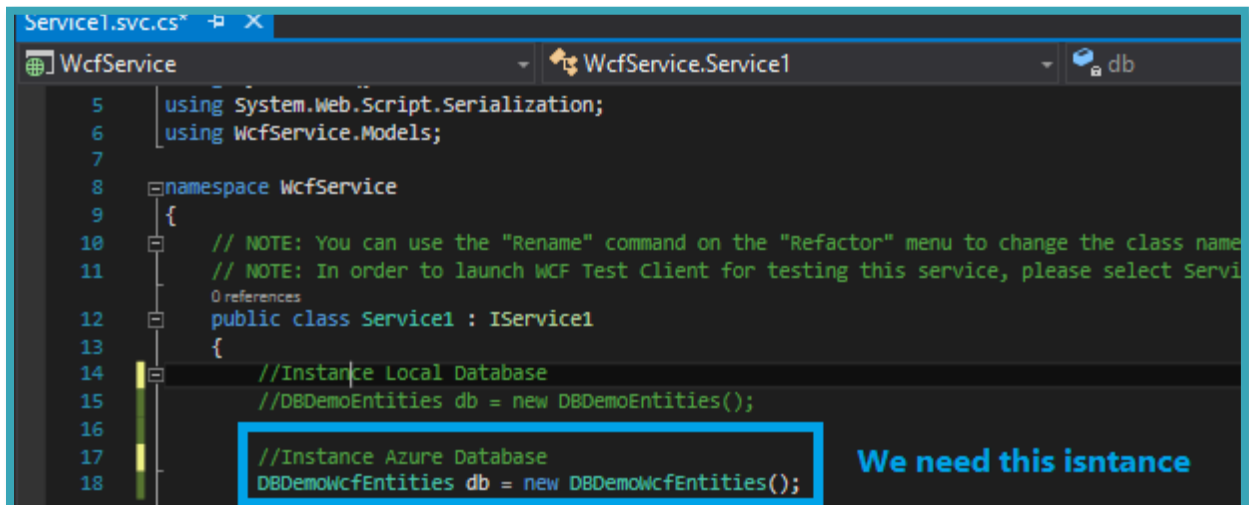
Verify if our connection string exists in our "**web.config**".



The screenshot shows the 'Web.config' file in Visual Studio. The file contains XML configuration for connection strings. A blue box highlights the following configuration:

```
<!--Connection azure database-->
<add name="DBDemoWcfEntities" connectionString="metadata=res://*/ModelDemo.csdl|res://*/ModelDemo.ssdl|res://*/ModelDemo.mdl" provider="System.Data.SqlClient" providerAssemblyName="System.Data.SqlClient"/>
```

Then need get in our Service1.svc and add this new instance



The screenshot shows the 'Service1.svc.cs' file in Visual Studio. The file contains C# code for the 'Service1' class. A blue box highlights the following code:

```
//Instance Azure Database
DBDemoWcfEntities db = new DBDemoWcfEntities();
```

To the right of the code, the text "We need this instance" is written in blue.

TEST

This is a simple test of our [ListEmployees](#)



The screenshot shows a web browser window with a single tab titled "Request Error". The address bar displays the URL "localhost:39649/Service1.svc/ListEmployees". The main content area of the browser shows a JSON response, which is a list of employee records. The JSON is formatted with syntax highlighting: strings are in green, numbers in blue, and the structure is indicated by brackets and commas. The response is a JSON object with a property "ListEmployeesResult" that contains an array of four employee objects. Each object has properties for "Address", "Age", "Lastname", "Name", and "Phone".

```
{
  - ListEmployeesResult: [
    - {
      Address: "Av.Pedro #12",
      Age: 24,
      Lastname: "Valverde",
      Name: "Nelson",
      Phone: "44-55-77"
    },
    - {
      Address: "Av.Larco",
      Age: 18,
      Lastname: "Agustin",
      Name: "Yordi",
      Phone: "54-86-12"
    },
    - {
      Address: "AV. Cepeda #112",
      Age: 19,
      Lastname: "Escobar",
      Name: "Javier",
      Phone: "45-56-01"
    },
    - {
      Address: "Av. Piedras Gordas",
      Age: 26,
      Lastname: "Guibert",
      Name: "Fredy",
      Phone: "54-21-14"
    }
  ]
}
```