**Connecting to power Bi form Synapse SQL serverless pool**

**To fetch the data stored in ADLS/Lake house tables to power bi we will be requiring two different interfaces which needed to be set up first.**

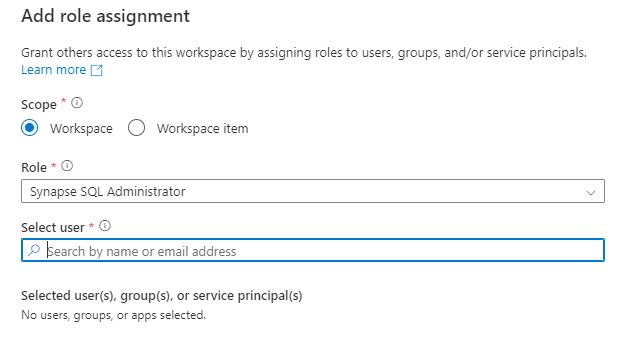
* **In which first part will be creating the login accounts and user accounts creation which we can use in power bi Synapse SQL Server connector to connect to the Serverless pool to fetch the external table or views helpful for reporting.**
* **The second interface which we need is between the serverless pool and the ADLS/Lake house tables where we will be creating interface to pull data from the storage account to serverless pool using views and tables.**

**Prerequisites:**

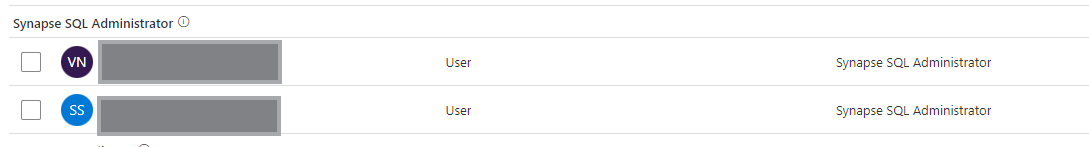
1. **Synapse Sql Serverless pool**
2. **Database in serverless pool**
3. **Admin access in synapse**
4. **SSMS (Optional)**

**Working Process:**

1. Create an Azure Synapse analytics resource.
2. Once the resource is created and able to be access the workspace
3. Go to ‘Manage’ tab and select ‘Access control.’
   1. Click on ‘Add’ and in Workspace scope select the role ‘Synapse SQL Administrator’ and provide the user mail id for access as mentioned below.

****

1. **Post assigning the required roles to the user validate the roles in same table.**

****

1. **Once the access part is sorted out now we have to create a data base in synapse serverless pool as per the naming convention required**

**Query: Create Database <Database name> go;**

1. **Once the data base is created, we have to create a login account for that specific database and run this query in ‘Master’ database which was created**

**Query: create login <Login User name> with password=’<Login password>’;**

* **To validate if login is created or not please use the below command**

**Query:**

**SELECT name,\* FROM sys.database\_principals WHERE type\_desc = 'DATABASE\_ROLE';**

1. **Once the Login account have been created, we can create Users for the specific Login account and run it on your specific data base which was created in step number (f).**

* **We can create n number of user accounts for a specific login account.**

**Query: create users <Username > for login <login account user name >;**

1. **As the user have been created in the previous step, we have to assign the role to the specific user based on the requirement. In our case we are going to give him the owner access.**

**Query: Alter Role db\_owner add member <Username> ;**

* **To find different roles available in the database we can use following command**

**Query:**

**SELECT name,\* FROM sys.database\_principals WHERE type\_desc = 'DATABASE\_ROLE';**

1. **By this step we will be having a login account and users assigned to those specific login accounts which we requires for the first part and we can use the login in details to connect to Power bi using synapse connector.**
2. **Now we have to establish an interface between Lakehouse table/ADLS paths to fetch the data from the storage account to serverless pool to create that we will require a scooped credentials which will help in authenticating the serverless pool user to interact with the data stored storage account.**
3. **Scooped credentials creation can be done by of two types**
   * 1. **Using SAS tokens**
     2. **Using Managed Identity**
4. **To create a Scooped credentials using SAS token we have to first generate the SAS tokens in the storage account with the specific set of access level based on the project requirements.**

**(\*\*\* From this point on wards we can use SQL editor available in Synapse workspace or we can use SSMS and connect with Login credentials which we have created previous steps \*\*\*)**

1. **Once the SAS token is generated, we can create a scooped credentials by creating the master key.**

**Query: CREATE Master Key;**

1. **Once master key is created we have to configure the Scooped credentials using the SAS tokens which we have generated.**

**Query: CREATE DATABASE SCOPED CREDENTIAL [< Name of the credentials >]**

**WITH IDENTITY='SHARED ACCESS SIGNATURE', SECRET =’< SAS Token > ';**

1. **As the SAS token have an expiry date and we don’t want to reupdate the SAS token we can create Scooped credentials by the second approach where we use Managed Identity to handle it**

**Query: CREATE DATABASE SCOPED CREDENTIAL < Name of the credentials > WITH IDENTITY = 'Managed Identity';**

1. **Once the Scooped credentials are created we have to give the control to the user which we have created in step number (g) so that when we use powerbi with that specific user they will be able to pull the data from the Storage account directly**

**Query: Grant CONTROL on DATABASE scoped credential :: [<Name of the credentials >] to [< User Name >]**

**Creating data source :**

**CREATE EXTERNAL DATA SOURCE AzureOpenData WITH ( LOCATION = 'https://azureopendatastorage.blob.core.windows.net/')**

**Creating view:**

**CREATE VIEW usPopulationView AS**

**SELECT**

**\***

**FROM**

**OPENROWSET(**

**BULK 'censusdatacontainer/release/us\_population\_county/year=20\*/\*.parquet',**

**DATA\_SOURCE = 'AzureOpenData',**

**FORMAT='PARQUET'**

**) AS uspv;**