**Use Cases for Serverless SQL Pool and Dedicated SQL Pool in Azure Synapse Analytics**

Azure Synapse Analytics is a powerful tool that combines big data and data warehousing. It has two important features: serverless SQL pool and dedicated SQL pool. Each feature has specific uses, helping organizations manage different data tasks effectively.

**Use Case of Serverless SQL Pool:-**

**Steps:-  
1.** Uploaded the data in Azure Data Lake storage gen2 > demo > employees.csv

A screenshot of a computer

Description automatically generated

2. In Develop we create an SQL script and query the employee\_sample.csv file using OPENROWSET()  
**OPENROWSET():-** It allows you to access files in Azure storage and returns the content as rows.  
**PARSER version 2.0:-** In Azure Synapse Analytics it is an improved feature that enhances query performance and compatibility. This version also provides better error messages and debugging capabilities, helping users to troubleshoot issues more efficiently.

A screenshot of a computer

Description automatically generated

3. To Find the total\_sal for each department

A screenshot of a computer

Description automatically generated

4. The Data can be visualized in Synapse Studio by switching from Table to the Chart view.  
Below is the result using a pie chart that shows the total salary by each department

A screenshot of a computer

Description automatically generated

**Uses Case of Dedicated SQL Pool:-**

**Steps:-**  
**1.** To create a Dedicated SQL Pool:-   
  Click the **Manage Hub — -> SQL pools — -> New**

A screenshot of a computer

Description automatically generated

While creating the Dedicated SQL Pool we specify the **name** and choose the **performance level**

**DWU (Data Warehouse Units):**

* DWU is the measure of compute resources available to your dedicated SQL pool.

**DW100c:**

* The **“DW100c”** designation specifies a configuration of 100 Data Warehouse Units.
* The **“c” in DW100c** stands for “compute optimized.” This means it is designed to balance computing power and cost-efficiently, efficiently handling data analysis tasks.

A screenshot of a computer

Description automatically generated

2. Uploaded the **NYCTripsmall.parquet** within **Data Hub — -> ADLS Gen 2 — → Demo folder**

A screenshot of a computer

Description automatically generated

The Below code first will check whether the table is present in the database (or) It will create a new table as we mentioned with the column name and its datatype.

Once the table is created copy the entire data into the table “dbo.NYCTaxiTripSmall” from the parquet file mentioned as the URL

**Code:-**

IF NOT EXISTS (SELECT \* FROM sys.objects O JOIN sys.schemas S ON O.schema\_id = S.schema\_id WHERE O.NAME = 'NYCTaxiTripSmall' AND O.TYPE = 'U' AND S.NAME = 'dbo')  
CREATE TABLE dbo.NYCTaxiTripSmall  
(  
[DateID] int,  
[MedallionID] int,  
[HackneyLicenseID] int,  
[PickupTimeID] int,  
[DropoffTimeID] int,  
[PickupGeographyID] int,  
[DropoffGeographyID] int,  
[PickupLatitude] float,  
[PickupLongitude] float,  
[PickupLatLong] NVARCHAR(4000),  
[DropoffLatitude] float,  
[DropoffLongitude] float,  
[DropoffLatLong] NVARCHAR(4000),  
[PassengerCount] int,  
[TripDurationSeconds] int,  
[TripDistanceMiles] float,  
[PaymentType] NVARCHAR(4000),  
[FareAmount] NUMERIC(19,4),  
[SurchargeAmount] NUMERIC(19,4),  
[TaxAmount] NUMERIC(19,4),  
[TipAmount] NUMERIC(19,4),  
[TollsAmount] NUMERIC(19,4),  
[TotalAmount] NUMERIC(19,4)  
)  
WITH  
(  
DISTRIBUTION = ROUND\_ROBIN,  
CLUSTERED COLUMNSTORE INDEX  
 - HEAP  
)  
GO  
COPY INTO dbo.NYCTaxiTripSmall  
(DateID 1, MedallionID 2, HackneyLicenseID 3, PickupTimeID 4, DropoffTimeID 5, PickupGeographyID 6, DropoffGeographyID 7, PickupLatitude 8, PickupLongitude 9, PickupLatLong 10,   
DropoffLatitude 11, DropoffLongitude 12, DropoffLatLong 13, PassengerCount 14, TripDurationSeconds 15, TripDistanceMiles 16, PaymentType 17, FareAmount 18, SurchargeAmount 19, TaxAmount 20,   
TipAmount 21, TollsAmount 22, TotalAmount 23)  
  
FROM 'https://sharandemoadlsgen2.dfs.core.windows.net/demo/NYCTripSmall.parquet'  
WITH  
(  
FILE\_TYPE = 'PARQUET',  
MAXERRORS = 0  
,IDENTITY\_INSERT = 'OFF'  
)  
SELECT COUNT(\*) FROM dbo.NYCTaxiTripSmall

3. Using the query **Select count(\*) from dbo.NYCTaxiTripSmall** we get a count of the records

A screenshot of a computer

Description automatically generated

4. Navigate to **Data Hub — -> Workspace — -> Databases — -> Tables — ->**we can see the **dbo.NYCTaxiTripSmall** stored within Tables

A screenshot of a computer

Description automatically generated

5. We can write our query to find the Total trip distance and Average Trip Distance

A screenshot of a computer

Description automatically generated

6. Results represented in Chart:-

A screenshot of a computer

Description automatically generated

**Conclusion:-**

Azure Synapse Analytics offers powerful tools for managing big data and data warehousing, specifically through its serverless SQL pool and dedicated SQL pool. The serverless SQL pool allows you to query data directly from Azure Data Lake storage using OPENROWSET(), making it ideal for quick and flexible data exploration. For example, you can easily find the total salary for each department and visualize it in a pie chart.

On the other hand, the dedicated SQL pool provides a more robust solution for large-scale data processing and storage. Creating a dedicated SQL pool with a specified performance level allows you to efficiently load and query large datasets, such as the NYC Taxi Trip data. This allows for detailed  
analysis, including calculating total and average trip distances and visualizing the results in Synapse Studio. Together, these tools help organizations manage and analyze their data more effectively.