😊 Loading data for Polybase

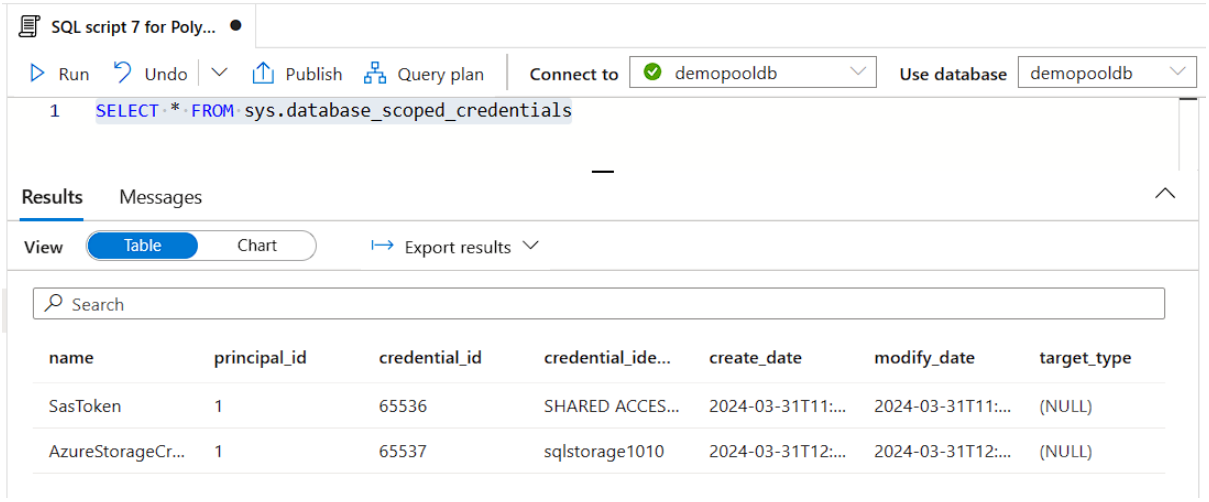
PolyBase is a feature in Azure SQL Data Warehouse and Azure Synapse Analytics (formerly Azure SQL Data Warehouse) that enables you to query and analyze data stored in external data sources such as Azure Blob Storage, Azure Data Lake Storage, and Hadoop Distributed File System (HDFS) without having to move or copy the data into your SQL pool.

Key features of PolyBase include:

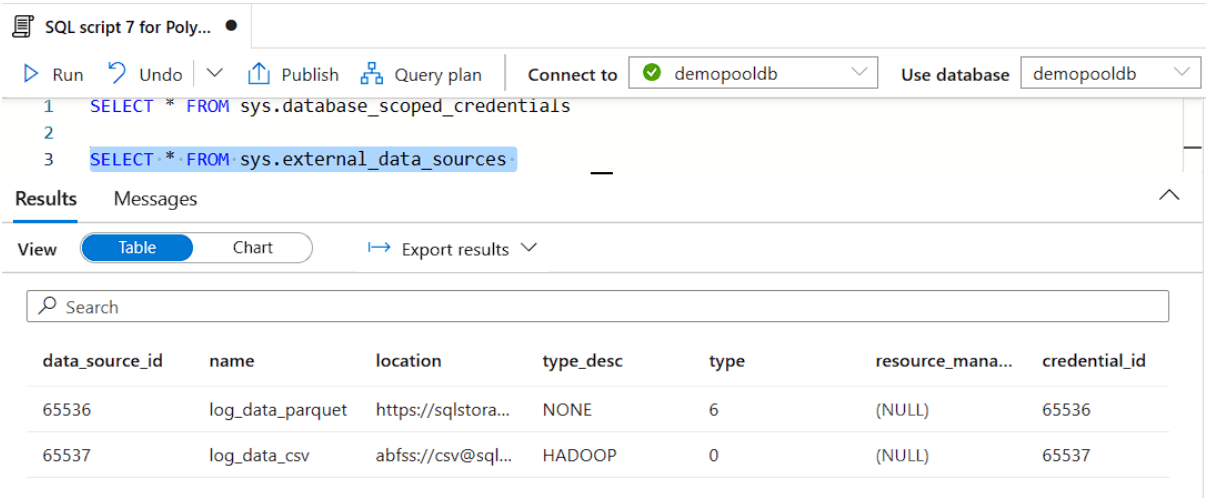
1. **External Tables:** PolyBase allows you to define external tables that reference data stored in external data sources. These external tables appear and behave like regular database tables, but the data resides outside of the SQL pool.
2. **T-SQL Queries:** You can use standard T-SQL queries to query data from both internal tables (stored in the SQL pool) and external tables (stored in external data sources) in the same query. PolyBase handles the data movement and query optimization transparently.
3. **Performance Optimization:** PolyBase optimizes query performance by pushing down query processing to the external data sources whenever possible. This minimizes data movement and maximizes query performance, especially for large-scale data analytics workloads.
4. **Parallel Data Movement:** PolyBase uses a distributed and parallel data movement architecture to efficiently transfer data between the SQL pool and external data sources. This ensures high throughput and scalability for data loading and querying operations.
5. **Integration with Azure Ecosystem:** PolyBase seamlessly integrates with other Azure services such as Azure Data Factory, Azure Databricks, and Azure Synapse Studio, enabling end-to-end data integration, analytics, and reporting workflows.
6. **Data Virtualization:** With PolyBase, you can virtually integrate and query data from multiple external data sources using a single SQL interface. This simplifies data access and analysis, especially in heterogeneous data environments with diverse data sources.
7. **Security and Compliance:** PolyBase provides robust security features including encryption, authentication, and authorization to ensure secure data access and compliance with regulatory requirements when accessing external data sources.

😄 To begin with the Lab:

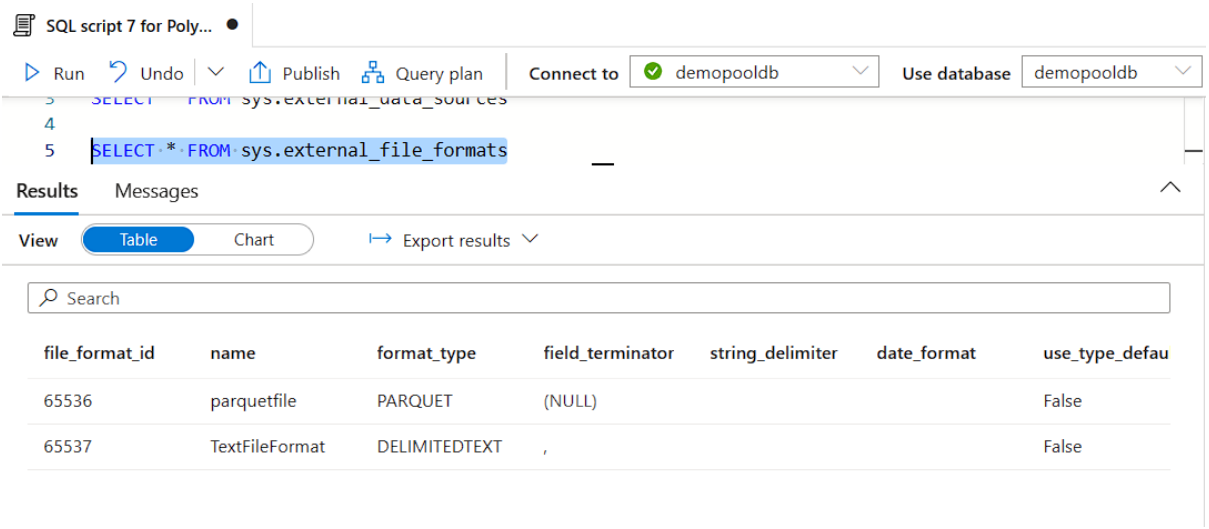
1. On your Azure Portal create a new SQL Script in your Dedicated SQL Pool.
2. Here you can see that we ran a query for scoped credentials and we got our results respectively.



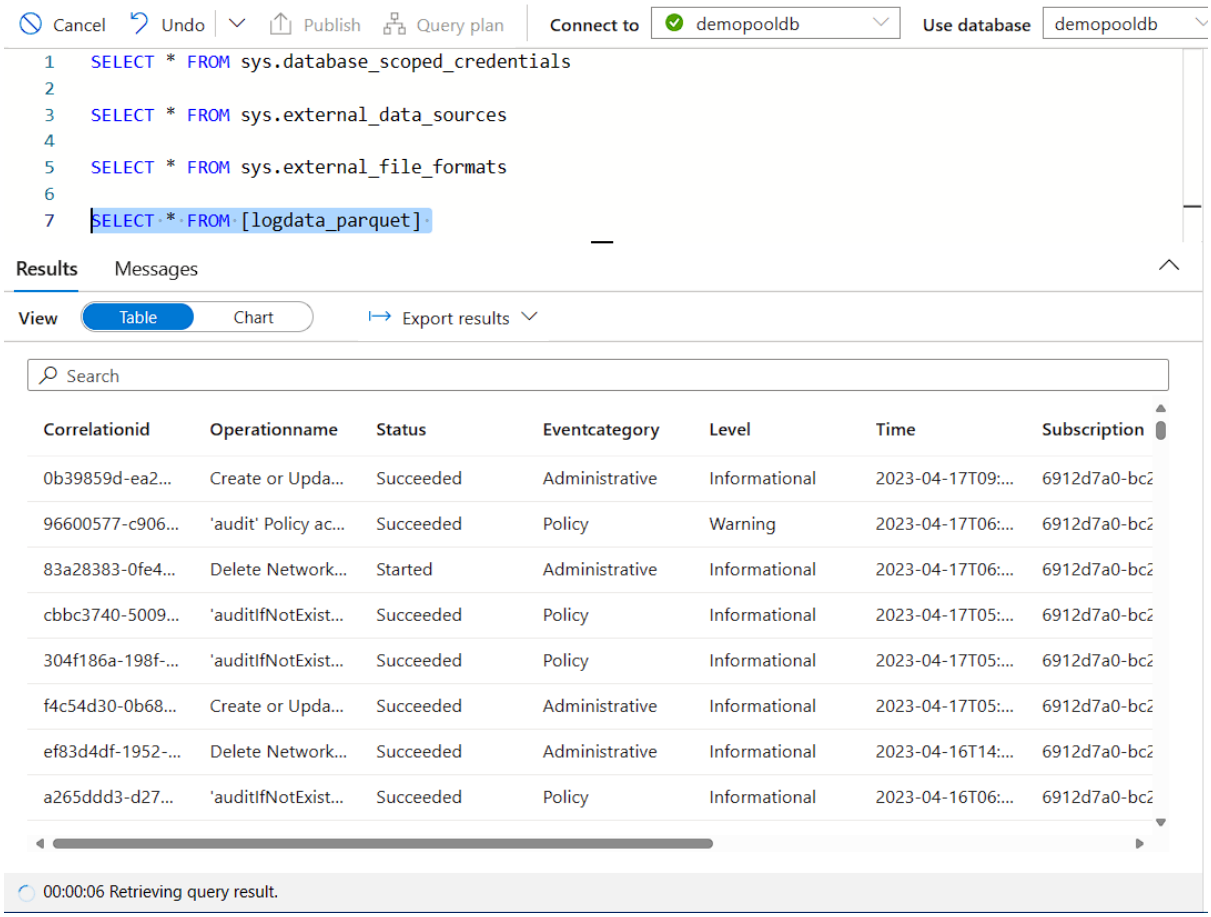
3. Below you can see that we have run a query for external data sources and we got the results respectively.



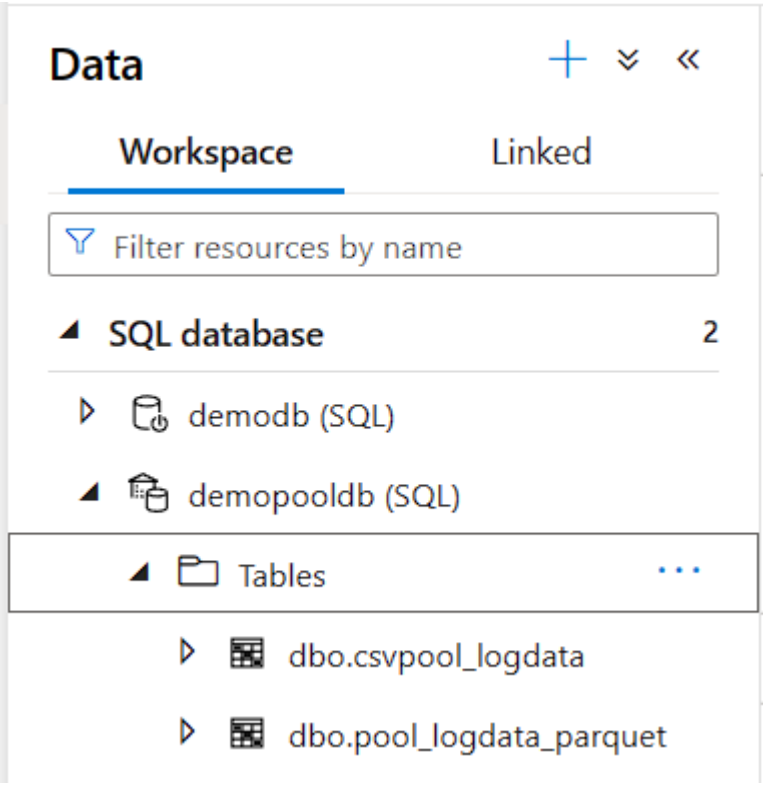
4. You can also run a query to look for the file formats which we'd seen earlier and we got the results.



5. If you want to use your existing database code credentials, your external data sources, and your existing external file formats, you can query for them from these system tables. So, the underlying system tables and use them accordingly.
6. Now that we already have the scope credentials, the external data sources, and the file formats in place, let's go ahead and reuse those existing artifacts that we already have. Also, let's use the existing external table that we have.



7. Now if go to data and then in workspace, refresh it and you will see two databases one is serverless and other is dedicated.
8. Expand the dedicated database and then you will have two options one for table and the other for external tables. Now you have to choose tables. Then you have to choose the Parque table and drop this. Then we'll recreate this table using Polybase.



9. After dropping the table, we will create a table command to copy data from one of the external tables that we have.
10. Note if you don't have any of these you can always go back to the previous labs and re-create everything.
11. Here in this table, we are using the distribution as round robin and we are selecting the data from our external table.

```
DROP TABLE [pool_logdata_parquet]

CREATE TABLE [pool_logdata_parquet]
WITH
(
DISTRIBUTION = ROUND_ROBIN
)
AS
SELECT  *
FROM  [logdata_parquet];
```

12. Now just run the statement. Once the query is successful. Then use the select command to view the data.
13. Below you can see that we are getting the data accordingly.
14. Once you are done just publish the data.

```
19
20  SELECT * FROM [pool_logdata_parquet]
```

Results | Messages

View: Table | Chart | Export results

Search

Correlationid	Operationname	Status	Eventcategory	Level	Time	Subscription
84fb2d66-c063...	'auditIfNotExist...	Succeeded	Policy	Informational	2023-04-17T15:...	6912d7a0-bc2
2127d1c8-ba1f...	'audit' Policy ac...	Succeeded	Policy	Warning	2023-04-17T15:...	6912d7a0-bc2
0dce6e10-870b...	Deletes An App...	Failed	Administrative	Error	2023-04-17T14:...	6912d7a0-bc2
c0d9e332-63ef...	'audit' Policy ac...	Succeeded	Policy	Warning	2023-04-17T14:...	6912d7a0-bc2
3536525e-4881...	Create or Upda...	Succeeded	Administrative	Informational	2023-04-17T14:...	6912d7a0-bc2
b9ac6192-3460...	Delete Virtual ...	Succeeded	Administrative	Informational	2023-04-17T14:...	6912d7a0-bc2

00:00:07 Query executed successfully.