**Azure Synapse Analytics** is a powerful, cloud-based analytics service that brings together big data and data warehousing. It enables organizations to ingest, prepare, manage, and serve data for immediate business intelligence and machine learning needs. With a unified experience for querying both relational and non-relational data, Synapse allows users to use SQL, Spark, or pipelines within a single workspace—streamlining data workflows and boosting productivity.

**Azure Synapse Storage Configuration**

* Azure Synapse Analytics uses **Azure Data Lake Storage Gen2** for storing its data and metadata.
* Each Synapse workspace links to a default Data Lake Storage account, but you can also connect external storage.
* This linked storage acts as a container for both system-managed and user-managed data.

**Output Destination Strategy**

* After processing data in Synapse, results are often stored in a **separate external storage location** for structured access, further analysis, or integration.

**Synapse Studio Interface Overview**

* **Home** – Quick access to recent items, resources, and common actions.
* **Data** – Browse, manage, and preview data from various sources.
* **Develop** – Create and edit SQL scripts, notebooks, and data flows.
* **Integrate** – Used for orchestrating ETL with Azure Data Factory
* **Monitor** – For pipeline and job tracking
* **Manage** – Admin settings, linked services, credentials, etc.

**Table Distribution in Synapse**

Azure Synapse distributes table data across **60 distributions** to parallelize processing.

**Distribution Strategies:**

* **Hash-distributed**: Ideal for large fact tables. Uses a specified column (e.g., numeric keys) to evenly distribute data and optimize joins.
* **Round-robin**: Distributes rows randomly. Best for staging or intermediate tables.
* **Replicated**: Stores full copies of the table on all distributions. Recommended for small dimension tables to reduce shuffle during joins.

**Example: Creating a Round-robin Table**

CREATE TABLE round\_table (

id INT,

name VARCHAR(4000),

salary INT

)

WITH (

DISTRIBUTION = ROUND\_ROBIN

);

INSERT INTO round\_table VALUES (1, 'girald', 10);

SELECT \* FROM round\_table;

**Example: Creating Replicated and Hash-distributed Tables**

CREATE SCHEMA gold;

CREATE TABLE gold.dim\_product (

dim\_key\_prod INT,

prod\_id INT,

prod\_name VARCHAR(4000)

)

WITH (

DISTRIBUTION = REPLICATE

);

CREATE TABLE gold.facttable (

dim\_key\_prod INT,

revenue INT,

cost VARCHAR(4000)

)

WITH (

DISTRIBUTION = HASH(dim\_key\_prod)

);

**Serverless SQL Pool Overview**

* A **logical metadata layer** that queries files directly from the data lake without requiring data movement.

**Table Types:**

* **Managed Tables**: Data and metadata managed by Synapse. Dropping the table deletes both.
* **External Tables**: Metadata managed by Synapse; data resides in external storage. Dropping the table removes only the metadata.

**Connecting Synapse to Azure Data Lake**

Use **Managed Identity** with Linked Services to securely connect to external storage.

**Steps:**

1. **Create a Master Key**

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'giraldpogi';

1. **Create Credential**

CREATE DATABASE SCOPED CREDENTIAL giraldcreds

WITH IDENTITY = 'Managed Identity';

1. **Create External Data Source**

CREATE EXTERNAL DATA SOURCE raw\_ext\_source

WITH (

LOCATION = 'https://giralddatalake.dfs.core.windows.net/raw',

CREDENTIAL = giraldcreds

);

**Querying with OPENROWSET (Serverless SQL)**

SELECT \* FROM OPENROWSET(

BULK 'revenue',

DATA\_SOURCE = 'raw\_ext\_source',

FORMAT = 'CSV',

PARSER\_VERSION = '2.0',

HEADER = TRUE

) AS query1;

**Creating External File Formats**

CREATE EXTERNAL FILE FORMAT csv\_format

WITH (

FORMAT\_TYPE = DELIMITEDTEXT,

FORMAT\_OPTIONS (

FIELD\_TERMINATOR = ',',

ENCODING = 'UTF-8'

)

);

CREATE EXTERNAL FILE FORMAT parquet\_format

WITH (

FORMAT\_TYPE = PARQUET,

DATA\_COMPRESSION = 'org.apache.hadoop.io.compress.SnappyCodec'

);

**Creating External Tables**

CREATE EXTERNAL TABLE revenue\_ext\_table (

Dealer\_ID VARCHAR(4000),

Model\_ID VARCHAR(4000),

Branch\_ID VARCHAR(4000),

Date\_ID VARCHAR(4000),

Unit\_Sold VARCHAR(4000),

Revenue VARCHAR(4000)

)

WITH (

LOCATION = 'revenue',

DATA\_SOURCE = 'raw\_ext\_source',

FILE\_FORMAT = csv\_format

);

**CETAS (Create External Table As Select)**

CREATE EXTERNAL TABLE revenue\_cetas

WITH (

LOCATION = 'cetas\_revenue',

DATA\_SOURCE = 'raw\_ext\_source',

FILE\_FORMAT = parquet\_format

)

AS

SELECT \* FROM OPENROWSET(

BULK 'revenue',

DATA\_SOURCE = 'raw\_ext\_source',

FORMAT = 'CSV',

PARSER\_VERSION = '2.0',

HEADER = TRUE

) AS query1;

**Creating a View with OPENROWSET**

CREATE VIEW revenue\_view AS

SELECT \* FROM OPENROWSET(

BULK 'revenue',

DATA\_SOURCE = 'raw\_ext\_source',

FORMAT = 'CSV',

PARSER\_VERSION = '2.0',

HEADER = TRUE

) AS query1;

**Linking to External Data Lake**

* Use **Linked Services** with **System-assigned Managed Identity** to authenticate.
* Define the storage account name and validate the connection.

**Loading Parquet Data into Dedicated SQL Pool using COPY INTO**

1. Set up credentials and file format.
2. Create destination table.
3. Use COPY INTO:

CREATE TABLE copy\_into\_table (

Dealer\_ID VARCHAR(4000),

Model\_ID VARCHAR(4000),

Branch\_ID VARCHAR(4000),

Date\_ID VARCHAR(4000),

Unit\_Sold BIGINT,

Revenue BIGINT

)

WITH (

DISTRIBUTION = ROUND\_ROBIN

);

COPY INTO copy\_into\_table (

Dealer\_ID 1,

Model\_ID 2,

Branch\_ID 3,

Date\_ID 4,

Unit\_Sold 5,

Revenue 6

)

FROM 'https://giralddatalake.dfs.core.windows.net/raw/cetas\_revenue/'

WITH (

FILE\_TYPE = 'PARQUET',

CREDENTIAL = (IDENTITY = 'Managed Identity')

);

SELECT \* FROM copy\_into\_table;

**Using POLYBASE with CTAS in Dedicated SQL Pool**

CREATE EXTERNAL DATA SOURCE raw\_ext\_source\_dfs

WITH (

LOCATION = 'abfs://raw@giralddatalake.dfs.core.windows.net/',

CREDENTIAL = giraldcreds

);

CREATE EXTERNAL TABLE parquet\_table (

Dealer\_ID VARCHAR(4000),

Model\_ID VARCHAR(4000),

Branch\_ID VARCHAR(4000),

Date\_ID VARCHAR(4000),

Unit\_Sold BIGINT,

Revenue BIGINT

)

WITH (

LOCATION = 'cetas\_revenue',

DATA\_SOURCE = raw\_ext\_source\_dfs,

FILE\_FORMAT = parquet\_format

);

CREATE TABLE poly\_table

WITH (

DISTRIBUTION = ROUND\_ROBIN

)

AS

SELECT \* FROM parquet\_table;

SELECT \* FROM poly\_table;

**Synapse Integration Capabilities**

* **Azure Data Factory (ADF)**: Build and orchestrate pipelines directly in Synapse.
* **Apache Spark Pools**: Run scalable big data jobs using PySpark or Scala.

**Summary**

* Azure Synapse Analytics is a **unified platform** for enterprise data warehousing, big data analytics, and data integration.
* It enables seamless querying, transformation, and orchestration across on-prem and cloud data using SQL, Spark, and Data Factory pipelines.