In this exercise, write the data from the event hub created in the previous exercise into a location in databricks. Then, stream the data from databricks into a table called youtube_videos in Snowflake.

Now, query the snowflake table, and you should be able to see the data.

**Title**: Streaming Data from Azure Event Hub to Snowflake via Databricks

**Objective**: Continuing fromt the previous exercise, the objective of this exercise is to stream data from the Azure Event Hub, write it into a location in Databricks, and then stream the data from Databricks into a Snowflake table called "youtube_videos." Finally, validate the data by querying the Snowflake table.
(SOL)

**Tasks**:

1. Configure Event Hub to Databricks Integration: Establish the connection between Azure Event Hub and Databricks. Set up the necessary credentials and configurations to access the Event Hub.

```
event_hub_connection_string = "<event_hub_connection_string>"
event_hub_name = "<event_hub_name>"
```

2. Stream Data to Databricks: Implement Spark code in Databricks to consume data from the Event Hub. Write the data to a specific location in Databricks, such as a Delta table or Parquet files.

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("EventHubToDatabricks").getOrCreate()

df = spark.readStream \
    .format("eventhubs") \
    .option("eventhubs.connectionString", event_hub_connection_string) \
    .option("eventhubs.consumerGroup", "$Default") \
    .option("eventhubs.startingPosition", "earliest") \
    .load()

# Apply necessary transformations to df

output_path = "/mnt/<mount_point>/eventhub_data"
df.writeStream \
```

```
    .format("delta") \
    .option("path", output_path) \
    .option("checkpointLocation", "/mnt/<mount_point>/eventhub_checkpoint") \
    .start()
```

3. Create Snowflake Table: Set up a table named "youtube_videos" in Snowflake with the appropriate schema that matches the data from the Event Hub. Define the column names and data types accordingly.

Create a Snowflake Table "youtube_videos" with the necessary schema as how it is goint to be written and defined in Databricks

4. Stream Data from Databricks to Snowflake: Configure a data streaming pipeline in Databricks to read the data from the location where it was written and stream it into the Snowflake table. Use Snowflake Connector to establish the connection and write data to the table.
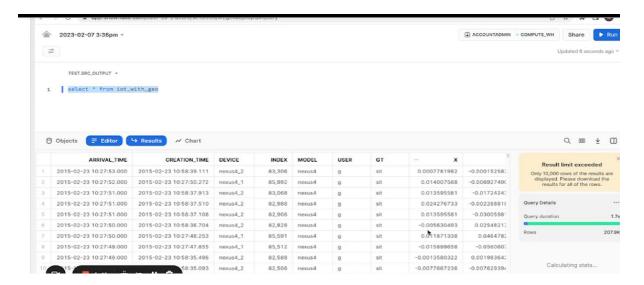
```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("DatabricksToSnowflake").getOrCreate()

df = spark.readStream \
    .format("delta") \
    .load(output_path)

df.writeStream \
    .format("snowflake") \
    .option("sfURL", "<snowflake_url>") \
    .option("sfDatabase", "<snowflake_database>") \
    .option("sfWarehouse", "<snowflake_warehouse>") \
    .option("sfSchema", "<snowflake_schema>") \
    .option("sfRole", "<snowflake_role>") \
    .option("dbtable", "youtube_videos") \
    .option("checkpointLocation", "/mnt/<mount_point>/snowflake_checkpoint") \
    .start()
```

5. Validate Data in Snowflake: Execute SQL queries in Snowflake to retrieve and examine the data stored in the "youtube_videos" table. Verify that the data streaming process from Databricks to Snowflake was successful.

Execute SQL Statements and validate the Data load and do analysis on top of it.



**Learning Outcomes:**

● Integrating Azure Event Hub with Databricks.
● Streaming data from Event Hub to Databricks.
● Writing data from Databricks to a specific location.
● Setting up a Snowflake table for data ingestion.
● Streaming data from Databricks to Snowflake using the Snowflake Connector.
● Querying and validating data in Snowflake.