

SNOWFLAKE ASSIGNMENT 1

Scenario: Migrate an on-premises SQL Server database to Snowflake via Azure Databricks.

Task: Extract data from SQL Server, transform it in Databricks using Snowpark, and load it into Snowflake.

Step 1: Using Azure Portal Query Editor

1. Open Azure Portal

- Go to <https://portal.azure.com>
- Navigate to your **SQL Database** (e.g., sqlsf) → Click **Query editor (preview)**

2. Login

- Use **SQL Authentication**
- Username: sa-user
- Password: Poojashree@307

3. Create Table

```
CREATE TABLE CUSTOMER (  
    id INT PRIMARY KEY,  
    customer VARCHAR(50),  
    region VARCHAR(50),  
    amount DECIMAL(10,2),  
);
```

4. Insert Data Directly

```
INSERT INTO CUSTOMER (id, customer, region, amount) VALUES  
(1, 'Amit Kumar', 'North', 12000),  
(2, 'Priya Sharma', 'South', 18000)  
(3, 'Rahul Mehta', 'East', 15000),  
(4, 'Sneha Rao', 'West', 22000),  
(5, 'Ravi Patel', 'Central', 17000);
```

5. Verify Data

```
SELECT * FROM CUSTOMER;
```

Microsoft Azure

Microsoft.SQLDatabase.newDatabaseNewServer_f72a8a62787e4bda94a08 | Overview

Your deployment is complete

Deployment name : Microsoft.SQLDatabase.newDatabaseNewServer_f72a8a62787e4bda94a08
Subscription : MML Learners
Resource group : rg-azuser4835_mml.local-gjNir

Start time : 10/15/2023, 9:55:28 PM
Correlation ID : a2ce60e3-b50f-40f2-a8bb-fd6d84cfaf5

Deployment details

Next steps

Go to resource

Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >

Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

Microsoft Azure

poojasql | Networking

Public Endpoints allow access to this resource through the internet using a public IP address. An application or resource that is granted access with the following network rules still requires proper authorization to access this resource. Learn more

Public network access

Selected networks

Connections from the IP addresses configured in the Firewall rules section below will have access to this database. By default, no public IP addresses are allowed. Learn more

Please save public network access value before adding new virtual networks.

Virtual networks

Allow virtual networks to connect to your resource using service endpoints. Learn more

Add a virtual network rule

Rule	Virtual network	Subnet	Address range	Endpoint status	Resource group	Subscription	State

Firewall rules

Allow certain public internet IP addresses to access your resource. Learn more

Add your client IPv4 address (81.3.193.247) Add a firewall rule

Rule name	Start IPv4 address	End IPv4 address
ClientIPAddress_2025-10-15_22-0-14	61.3.193.247	61.3.193.247

Exceptions

Allow more services and resources to access this server

Save Discard

Microsoft Azure

Home > Microsoft.SQLDatabase.newDatabaseNewServer_9cacfef5f634335b45ce | Overview

sqlsf (poojasql/sqlsf)

Show me performance metrics for this SQL database. Improve slowest/worst queries. Ensure disaster recovery readiness.

Copy Restore Export Set server firewall Delete Connect to... Feedback

Mirror databases in Microsoft Fabric. Easily replicate your existing databases in Fabric, and help your team achieve streamlined ETL and operational analytics goals. Learn more

Essentials

Resource group (move)	: rg-azuser4835_mml.local-gjNir	Ser	Server name	: poojasql.database.windows.net
Status	: Online	Connection strings	Show database connection strings	
Location	: East US	Pricing tier	General Purpose - Serverless: Gen5, 1 vCore	
Subscription (move)	: MML Learners	Auto-pause delay	1 hour	
Subscription ID	: 2a3c6418-97b9-4d96-a24b-2c2d763d375	Earliest restore point	2025-10-15 15:37 UTC	
Tags (edit)	: Add tags			

Getting started Monitoring Properties Features Notifications (0) Integrations Tutorials

Start working with your database

Connect to your database and start working with data with a few simple steps. Learn more

Configure access Configure network access to your SQL server. Learn more

Configure See connection strings

Connect to application Use connection strings to connect to your SQL database from your applications and favorite tools.

Open Azure Data Studio Open in Visual Studio Microsoft Store

Start developing Work in your database by using tools to add, modify and query data. Compare tools

Mirror database in Fabric Replicate existing databases in Fabric, and help your team achieve streamlined ETL and operational analytics goals. Learn more

Query 1 >

Run Cancel query Save query Export data as Show only Editor

```

1
2
3  -- Create a table
4  CREATE TABLE CUSTOMER (
5
6    id INT,
7    customer VARCHAR(30),
8    region VARCHAR(30),
9    amount INT
10 );
11 INSERT INTO CUSTOMER (id, customer, region, amount) VALUES
12 (1, 'Amit Kumar', 'North', 12000),
13 (2, 'Priya Sharma', 'South', 18000),
14 (3, 'Rahul Mehta', 'East', 15000),
15 (4, 'Sneha Rao', 'West', 22000),
16 (5, 'Ravi Patel', 'Central', 17000);
17
18 SELECT * FROM CUSTOMER;

```

17
18 SELECT * FROM CUSTOMER;

Results Messages

Search to filter items...

id	customer	region	amount
1	Amit Kumar	North	12000
2	Priya Sharma	South	18000
3	Rahul Mehta	East	15000
4	Sneha Rao	West	22000
5	Ravi Patel	Central	17000

Before transforming and connecting sql op in snowflake

The screenshot shows the Snowflake web interface. On the left, there's a sidebar with navigation links like 'Work with data' (Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace), 'Horizon Catalog', 'Catalog', and 'Data sharing'. A message at the bottom says '\$384 credits left' and 'Trial ends in 28 days' with a 'Upgrade' button. The main area has tabs for 'Data Ingestion and Transfo...', 'types of Views', 'hexa_day1_query', 'creating file formats in sf', 'snowpipe_azure', and a timestamp '2025-10-15 9:27pm'. Below these tabs is a search bar. The central part of the screen contains a code editor with a query and a results pane below it. The results pane shows a table with columns ID, CUSTOMER, REGION, AMOUNT, and LOAD_DATE. A message in the results pane says 'Query produced no results'. To the right of the results pane are 'Query Details' showing a duration of 51ms, 0 rows, and a query ID.

```

1 CREATE DATABASE MIGRATION_DB;
2 USE SCHEMA MIGRATION_DB.PUBLIC;
3
4 CREATE OR REPLACE TABLE CUSTOMER (
5   id INT,
6   customer STRING,
7   region STRING,
8   amount NUMBER(10,2),
9   load_date DATE
10 );
11
12

```

Results

ID	CUSTOMER	REGION	AMOUNT	LOAD_DATE

Query produced no results

Query Details

- Query duration: 51ms
- Rows: 0
- Query ID: 01fbacc-0001-6698-0...

Step 2: Prepare Azure Databricks workspace

Open:

Go to <https://portal.azure.com>

Actions:

1. In the search bar, type "Databricks"
2. Click "Azure Databricks" → Create

3. Choose:
 - o Subscription: your Azure subscription
 - o Resource group: create a new one (e.g., data-migration-rg)
 - o Workspace name: databricks-sql-migration
 - o Region: same as your storage or Snowflake region (e.g., East US)
4. Click **Review + Create → Create**
5. Once deployed, click “**Go to resource**” → “**Launch Workspace**”
6. It opens **Databricks Web UI** (in a new tab).

Step 3: Create a Databricks cluster

In Databricks Workspace:

1. Click **Compute → Create Cluster**
2. Name: migration-cluster
3. Runtime: choose “**11.x LTS (includes Apache Spark 3.5 + Scala 2.12)**”
4. Click **Create Cluster** (wait a few minutes until cluster = Running)

Step 4: Connect Databricks to SQL Server

We'll use **JDBC** connection.

In Databricks:

1. Click **Workspace → Create → Notebook**
 - o Name: SQL_to_Snowflake_Migration
 - o Language: Python
2. Attach to your cluster.

Step 5: Transform data in Databricks using Snowpark / PySpark

Step 6: Connect to Snowflake

Now, we'll write data into Snowflake.

Pre-setup (in Snowflake Web UI):

1. Login to Snowflake Snowsight

2. Create target database and schema:

```
CREATE DATABASE MIGRATION_DB;  
USE SCHEMA MIGRATION_DB.PUBLIC;
```

Create target table:

```
CREATE OR REPLACE TABLE EMPLOYEES_SNOW (  
    EMP_ID INT,  
    EMP_NAME STRING,  
    DEPT STRING,  
    LOAD_DATE DATE  
);
```

Step 7: Load data into Snowflake from Databricks

In your Databricks notebook, add:

```
sfOptions = {  
    "sfURL": "sqishot-fa68768.snowflakecomputing.com",  
    "sfDatabase": "MIGRATION_DB",  
    "sfSchema": "PUBLIC",  
    "sfWarehouse": "COMPUTE_WH",  
    "sfRole": "ACCOUNTADMIN",  
    "sfUser": "poojashree",  
    "sfPassword": "Poojashree@307"  
}
```

```
# Write transformed data into Snowflake  
transformed_df.write \  
.format("snowflake") \  
.options(**sfOptions) \  
.option("dbtable", "EMPLOYEES_SNOW") \  
.mode("overwrite") \  
.save()
```

```

Just now (20s)

jdbc_url = (
    "jdbc:sqlserver://poojasql.database.windows.net:1433;" 
    "databaseName=poojasql;" 
    "encrypt=true;" 
    "trustServerCertificate=false;" 
    "hostNameInCertificate=*.database.windows.net;" 
    "loginTimeout=30;" 
)

connection_properties = {
    "user": "sa-user@poojasql",
    "password": "Poojashree@387",
    "driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
}

customer_df = spark.read.jdbc(
    url=jdbc_url,
    table="dbo.CUSTOMER",
    properties=connection_properties
)

display(customer_df)
from pyspark.sql.functions import lit, current_date

transformed_df = customer_df.withColumn("Load_Date", current_date())
transformed_df.show(5)
sfOptions = {
    "sfURL": "sqishot-fa68768.snowflakecomputing.com",
    "sfDatabase": "MIGRATION_DB",
    "sfSchema": "PUBLIC",
    "sfWarehouse": "COMPUTE_WH",
    "sfRole": "ACCOUNTADMIN",
    "sfUser": "poojasheer",
    "sfPassword": "Poojashree@387"
}

# Write transformed data into Snowflake
transformed_df.write \
    .format("snowflake") \
    .options(**sfOptions) \
    .option("dbtable", "customer") \
    .mode("overwrite") \
    .save()
print("Data written to Snowflake")

```

(3) Spark Jobs

- customer df: pyspark.sql.dataframe.DataFrame = [id: integer, customer: string ... 2 more fields]
- transformed df: pyspark.sql.dataframe.DataFrame = [id: integer, customer: string ... 3 more fields]

	# id	# customer	# region	# amount
1	1	Amit Kumar	North	12000
2	2	Priya Sharma	South	18000
3	3	Rahul Mehta	East	15000

(3) Spark Jobs

- customer df: pyspark.sql.dataframe.DataFrame = [id: integer, customer: string ... 2 more fields]
- transformed df: pyspark.sql.dataframe.DataFrame = [id: integer, customer: string ... 3 more fields]

	# id	# customer	# region	# amount	
1	1	Amit Kumar	North	12000	
2	2	Priya Sharma	South	18000	
3	3	Rahul Mehta	East	15000	
4	4	Sneha Rao	West	22000	
5	5	Ravi Patel	Central	17000	

5 rows | 19.58s runtime

```

+---+---+---+---+
| id|customer|region|amount| Load_Date|
+---+---+---+---+
| 1| Amit Kumar| North| 12000| 2025-10-15|
| 2| Priya Sharma| South| 18000| 2025-10-15|
| 3| Rahul Mehta| East| 15000| 2025-10-15|
| 4| Sneha Rao| West| 22000| 2025-10-15|
| 5| Ravi Patel| Central| 17000| 2025-10-15|
+---+---+---+---+

```

Data written to Snowflake

Step 8: Verify in Snowflake

Go back to **Snowsight**, and run:

```
USE DATABASE MIGRATION_DB;  
SELECT * FROM CUSTOMER;
```

You'll see your migrated data now available in Snowflake.

The screenshot shows the Snowflake Snowsight interface. On the left, there's a sidebar with navigation links like 'Work with data' (Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace), 'Horizon Catalog', 'Catalog', and 'Data sharing'. A message at the bottom says '\$384 credits left' and 'Trial ends in 28 days' with an 'Upgrade' button. The main area has tabs for 'Data Ingestion and Transfo...', 'types of Views', 'hexa_day1_query', 'creating file formats in sf', 'snowpipe_azure', and a timestamp '2025-10-15 9:27pm'. Below these tabs, there's a code editor window with the following SQL code:

```
USE SCHEMA MIGRATION_DB.PUBLIC;  
CREATE OR REPLACE TABLE CUSTOMER (  
    id INT,  
    customer STRING,  
    region STRING,  
    amount NUMBER(10,2),  
    load_date DATE  
);  
SELECT * FROM CUSTOMER;
```

Below the code editor is a 'Results' tab showing a table with the following data:

ID	CUSTOMER	REGION	AMOUNT	LOAD_DATE
1	Amit Kumar	North	12000	2025-10-15
2	Priya Sharma	South	18000	2025-10-15
3	Rahul Mehta	East	15000	2025-10-15
4	Sneha Rao	West	22000	2025-10-15
5	Ravi Patel	Central	17000	2025-10-15

On the right side of the results table, there's a 'Query Details' panel with the following information:

- Query duration: 84ms
- Rows: 5
- Query ID: 01bfbaec-0001-6699-0...