# Lab 3 - Data Transformation and Querying in Amazon Redshift

1. Perform an ELT operation with materialized views and stored procedures.
2. Use Amazon Redshift scheduled queries.
3. Query data directly from the source using Amazon Redshift data sharing.

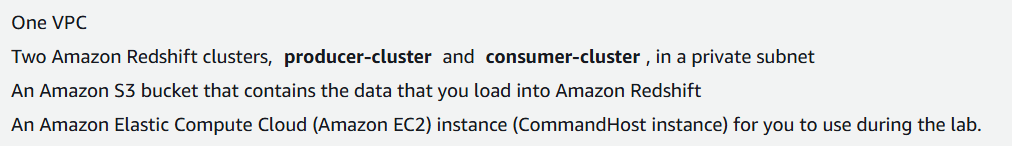
* **ETL (Extract, Transform, Load): First, we extract the data, transform it to a needed format, and then load it into Redshift.**

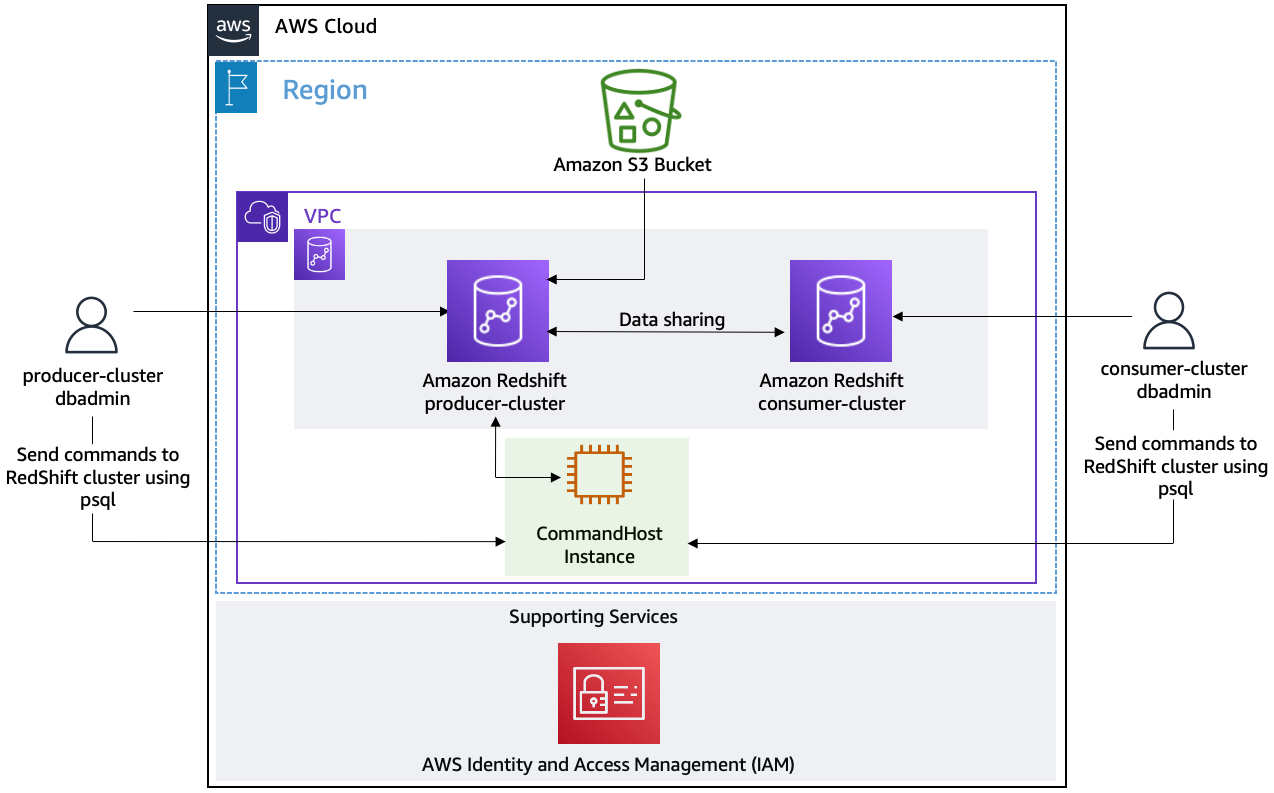
**to retain only the parts of the raw data that you need.**

* **ELT (Extract, Load, Transform): First, we extract the data, load it into Redshift, and then transform it using Redshift's capabilities.**

In this lab, you load stock market data (stored in Amazon S3) into Amazon Redshift. You first query the data as is. Then, you create a materialized view to transform the data (to better suit your needs) and query that view. Finally, you create a scheduled task to query the materialized view at a set interval, and you learn how to retrieve the results of a scheduled query.

Pre-created





Producer-Cluster: This is where the data is initially loaded and possibly transformed.

Consumer-Cluster: This is where the data is shared and used for analysis.

1. \*\*Load Data into Amazon Redshift\*\*: The producer-cluster dbadmin loads data from Amazon S3 into the producer-cluster.

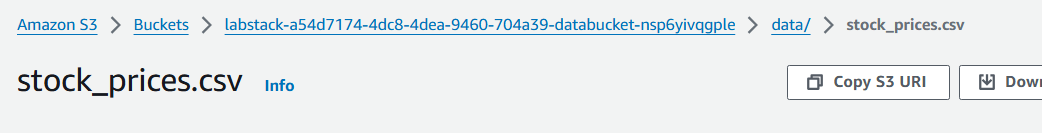
2. \*\*Data Sharing\*\*: The producer-cluster shares its data with the consumer-cluster without duplicating it.

3. \*\*Analyze Data in Consumer-Cluster\*\*: The consumer-cluster dbadmin queries and analyzes the shared data.

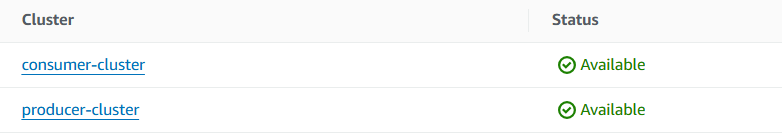
4. \*\*CommandHost Instance\*\*: Manages and sends commands to both the producer and consumer clusters.

**Task 1: Explore the lab environment**

1.1 open s3 location



1.2 in RS



1.3 copy both Endpoints

**Task 2: Create an external table**

2.0 open URL to CLI

2.1 connect to producer cluster

cd ~

export PGPASSWORD='<INSERT\_PASSWORD>'

psql -U dbadmin -h '<INSERT\_REDSHIFT\_CLUSTER\_ENDPOINT>' -d producer\_stocks -p 5439



2.2 in prod, create schema and table

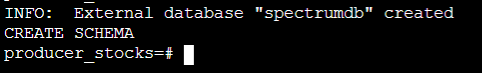
CREATE EXTERNAL SCHEMA spectrum

FROM DATA CATALOG

DATABASE spectrumdb

IAM\_ROLE 'INSERT\_REDSHIFT\_ROLE'

CREATE EXTERNAL DATABASE IF NOT EXISTS;



2.3 create external table

DROP TABLE IF EXISTS spectrum.stocksummary;

CREATE EXTERNAL TABLE spectrum.stocksummary(

Trade\_Date VARCHAR(15),

Ticker VARCHAR(5),

High DECIMAL(8,2),

Low DECIMAL(8,2),

Open\_value DECIMAL(8,2),

Close DECIMAL(8,2),

Volume DECIMAL(15),

Adj\_Close DECIMAL(8,2)

)

ROW FORMAT DELIMITED

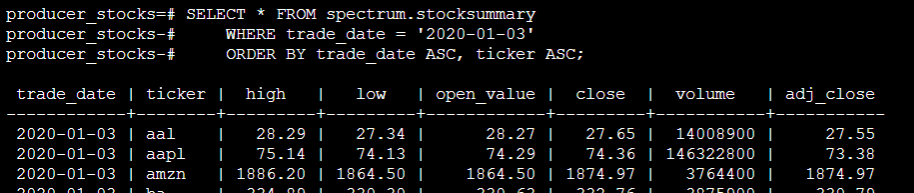
FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 's3://INSERT\_DATA\_BUCKET/data/';



2.4 works





**Task 3: Create and query a materialized view - T from ETL**

A materialized view (MV) is a database object that contains the results of a query, stored as a physical table, allowing for faster access to precomputed data compared to executing the query each time it is requested.

3.1

DROP MATERIALIZED VIEW IF EXISTS stocks\_mv;

CREATE MATERIALIZED VIEW stocks\_mv AS

SELECT trade\_date, ticker, volume FROM spectrum.stocksummary;

it only shows the selected columns

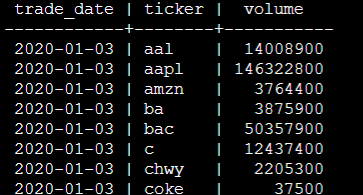
Like here, we create MV such that a query which only returns these three columns of the table

3.2 query

SELECT \* FROM stocks\_mv

WHERE trade\_date = '2020-01-03'

ORDER BY trade\_date ASC, ticker ASC;



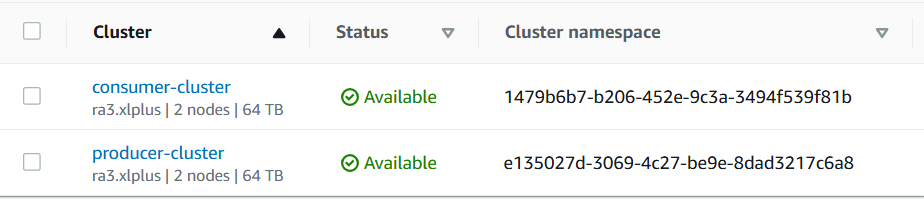


**Task 4: Use Amazon Redshift data sharing for faster data access between clusters**

With Amazon Redshift data sharing, you can grant access directly to specific data and objects in one Redshift cluster to entities in another cluster.

It is helpful in case you (consumer DB admin) want to see live data from producer DB, and dont’t have to wait for complete copy of data from prod to cons.

4.1 copy namespace values



4.2 in cli

CREATE DATASHARE stocks\_share;



4.3 add schema and MV to data share

ALTER DATASHARE stocks\_share ADD SCHEMA public;

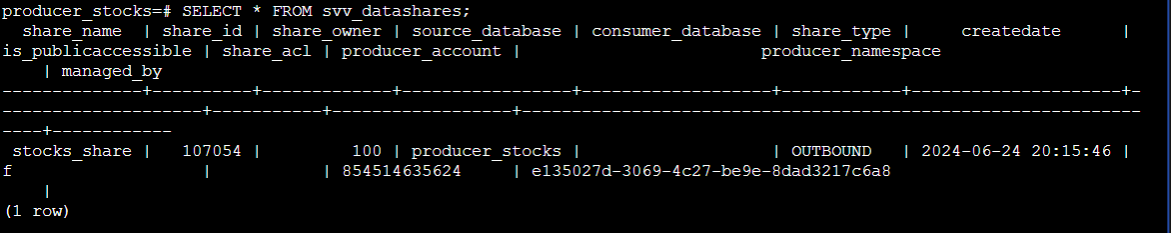
ALTER DATASHARE stocks\_share ADD TABLE public.stocks\_mv;

4.4 grant consumer access to producer via data share

GRANT USAGE ON DATASHARE stocks\_share TO NAMESPACE 'INSERT\_CONSUMER\_NAMESPACE\_ID';



4.5 verify



Source - producer\_stocks

Share type - OUTBOUND (prod -> cons)

4.6



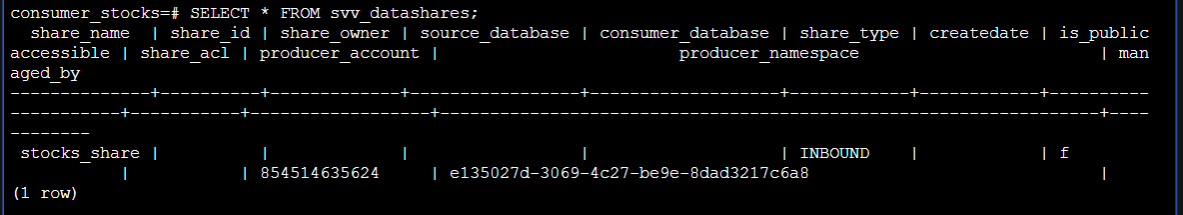
2 things outbound - schema and MV

4.7 verify from consumer side

Connect to consumer cluster using endpoint



4.8 view datashare



INBOUND

4.9 

4.10 query in consumer

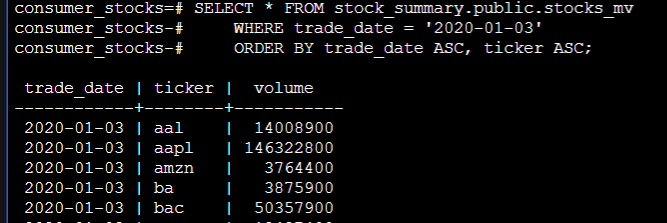
CREATE DATABASE stock\_summary FROM DATASHARE stocks\_share of NAMESPACE 'INSERT\_PRODUCER\_NAMESPACE\_ID';

4.11 now, get data from MV, which was shared from producer to consumer

SELECT \* FROM stock\_summary.public.stocks\_mv

WHERE trade\_date = '2020-01-03'

ORDER BY trade\_date ASC, ticker ASC;



To revoke data share, go to producer

REVOKE USAGE ON DATASHARE stocks\_share FROM NAMESPACE 'INSERT\_CONSUMER\_NAMESPACE\_ID';