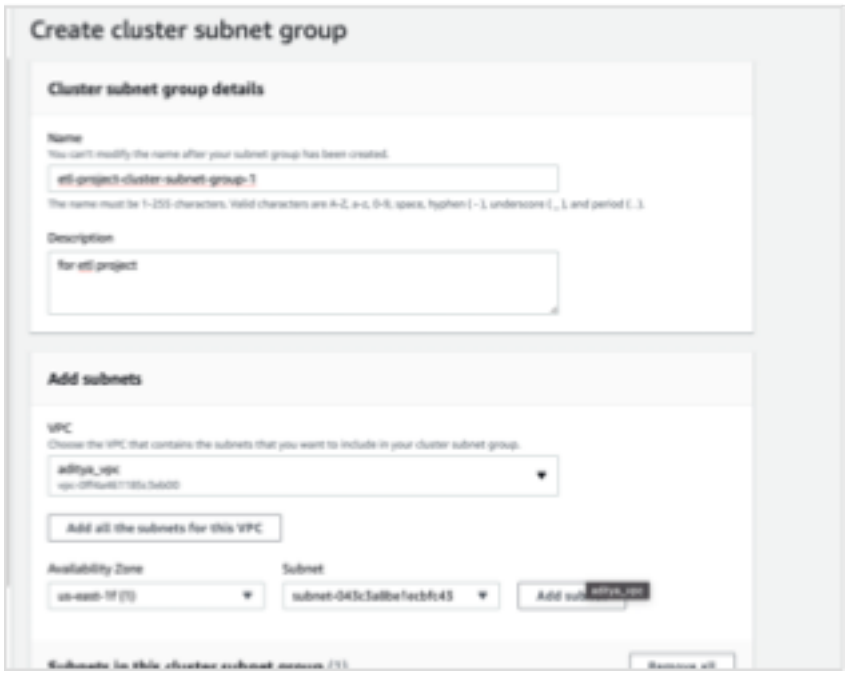
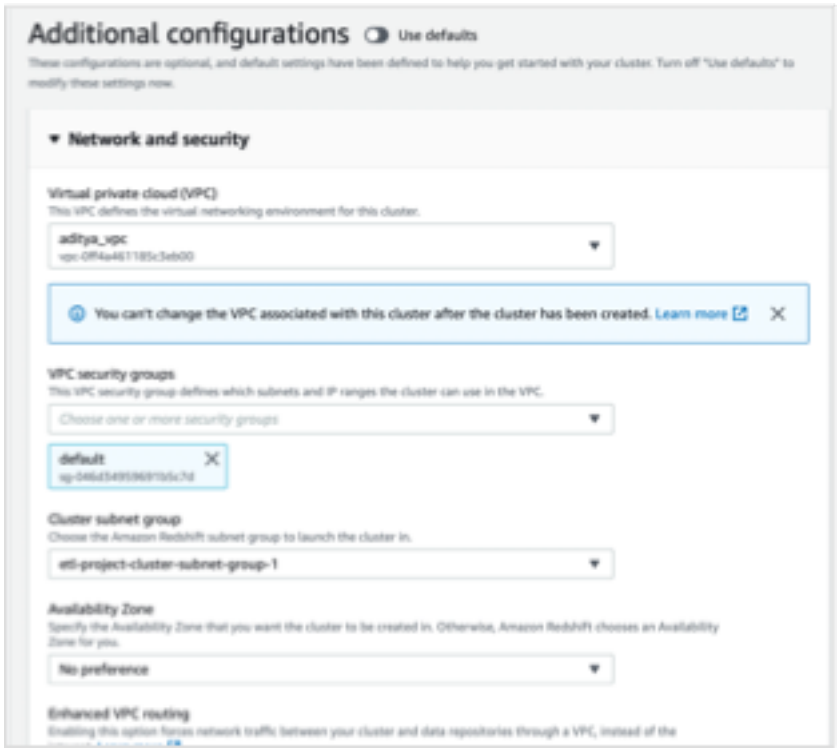
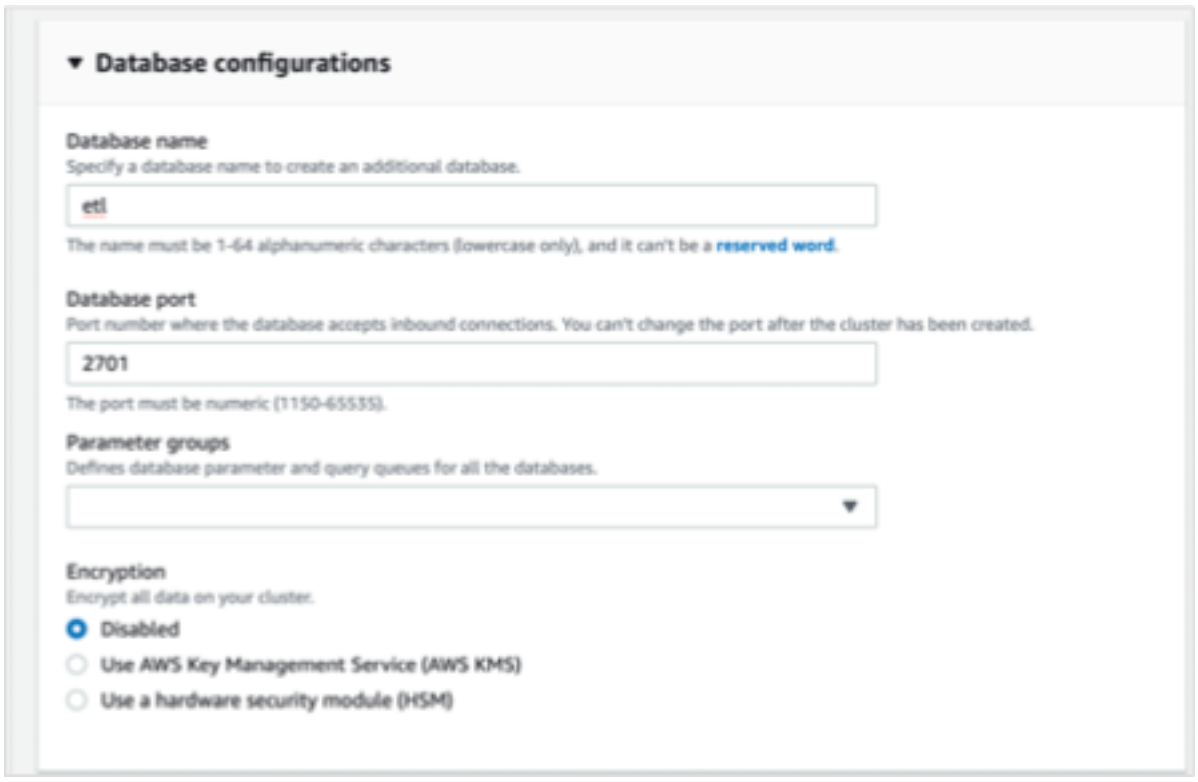
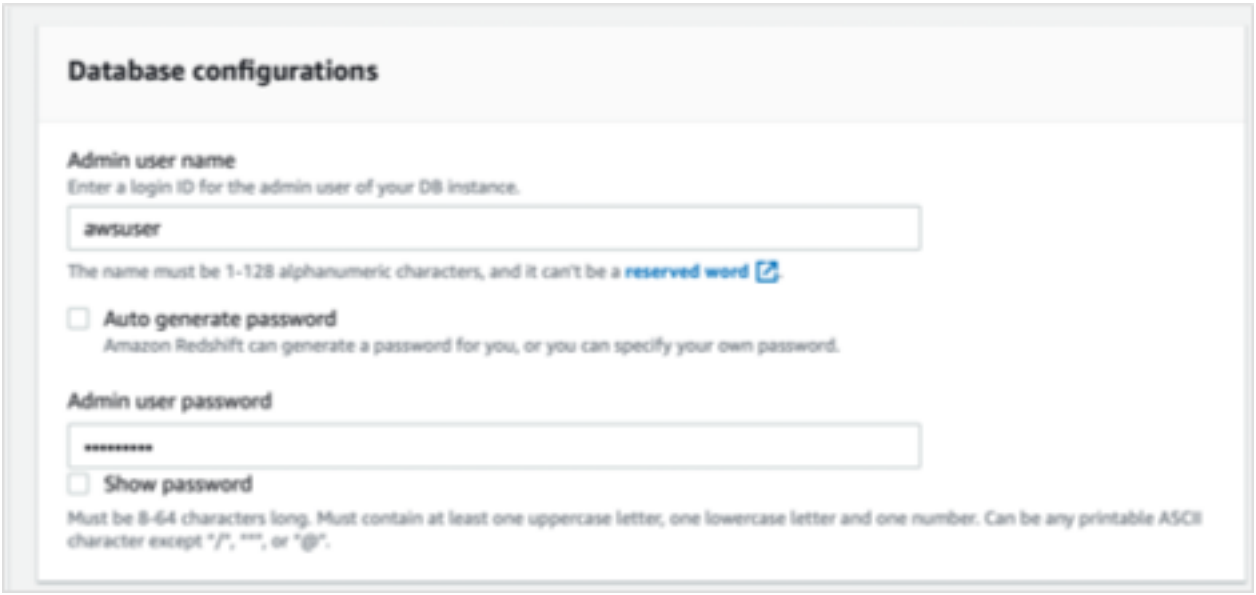
Creation of a Redshift Cluster

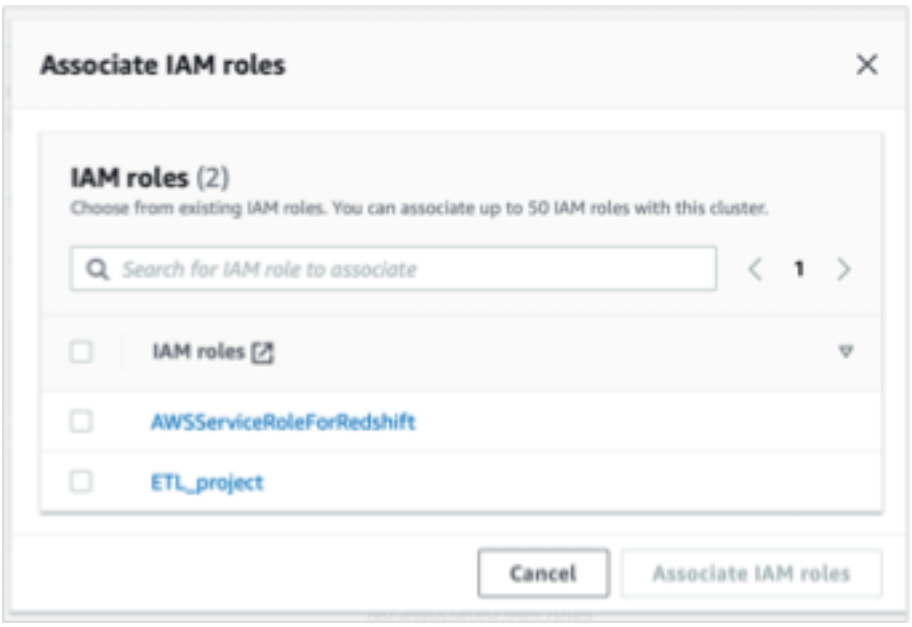
1. Screenshots of the configuration of the Redshift cluster that you have created:



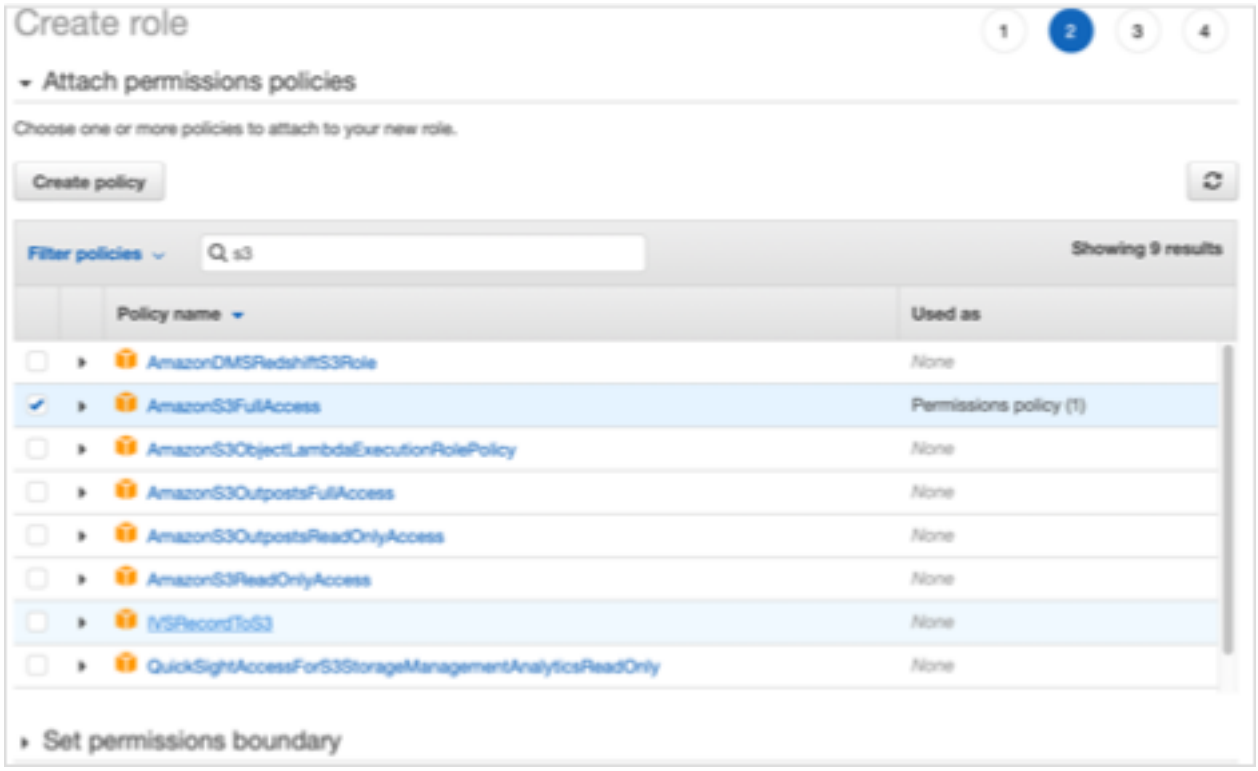

Additional configurations
Use defaults
These configurations are optional, and default settings have been defined to help you get started with your cluster. Turn off "Use defaults" to modify these settings now.
Network and security
Virtual private cloud (VPC)
This VPC defines the virtual networking environment for this cluster.
aditya_vpc
vpc-0ff4a461185c3eb00
You can't change the VPC associated with this cluster after the cluster has been created. Learn more
VPC security groups
This VPC security group defines which subnets and IP ranges the cluster can use in the VPC.
Choose one or more security groups
default
sg-046d54959691b5c7d
Cluster subnet group
Choose the Amazon Redshift subnet group to launch the cluster in.
etl-project-cluster-subnet-group-1
Availability Zone
Specify the Availability Zone that you want the cluster to be created in. Otherwise, Amazon Redshift chooses an Availability Zone for you.
No preference
Enhanced VPC routing


Database configurations
Database name
Specify a database name to create an additional database.
etl
The name must be 1-64 alphanumeric characters (lowercase only), and it can't be a reserved word.
Database port
Port number where the database accepts inbound connections. You can't change the port after the cluster has been created.
2701
The port must be numeric (1150-65535).
Parameter groups
Defines database parameter and query queues for all the databases.
Encryption
Encrypt all data on your cluster.
Disabled
Use AWS Key Management Service (AWS KMS)
Use a hardware security module (HSM)





Screenshots of the configuration of the IAM Role:



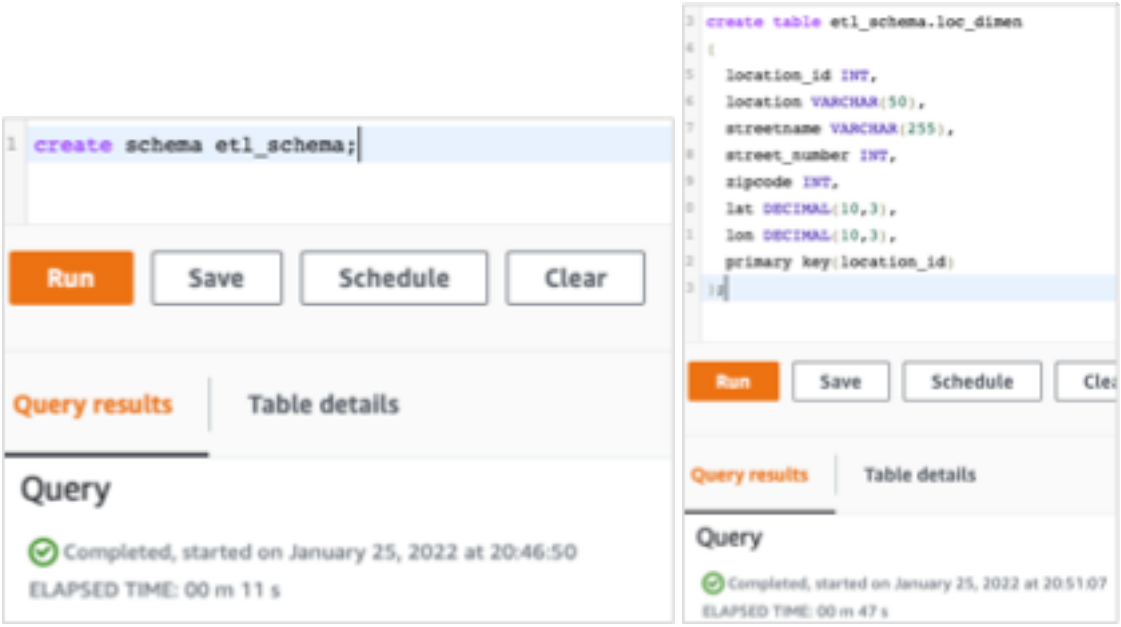
Final Configuration

2. Setting up a database in the Redshift cluster and running queries to create the dimension and fact tables

Queries to create the various dimension and fact tables with appropriate primary and foreign keys:

Query to create a schema:

create schema etl_schema;

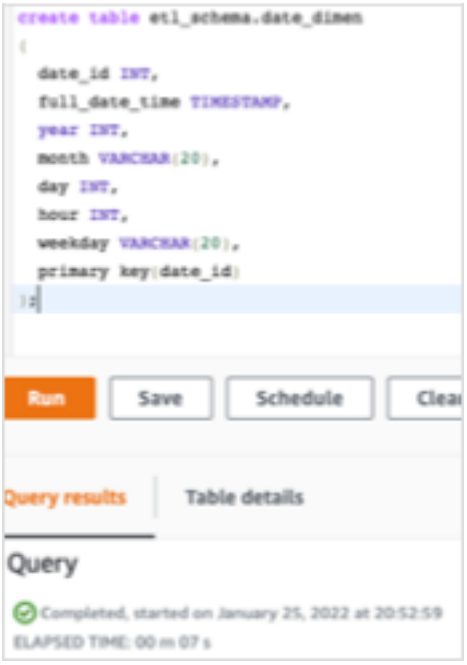


Query to create a location dimension table:

create table etl_schema.loc_dimen (

location_id INT,
location VARCHAR(50), streetname VARCHAR(255),
street_number INT,
zipcode INT,
lat DECIMAL(10,3),
lon DECIMAL(10,3),
primary key(location_id)

);



Query to create a date dimension table:

```
create table etl_schema.date_dimen (

date_id INT,
full_date_time TIMESTAMP, year INT,
month VARCHAR(20),
day INT,
hour INT,
weekday VARCHAR(20), primary key(date_id)

);
```

Query to create a card type dimension table:

```
create table etl_schema.card_type_dimen (

card_type_id INT, card_type VARCHAR(30), primary
key(card_type_id)

);
```

Query to create an atm dimension table:

```
create table etl_schema.atm_dimen (

atm_id INT,
atm_number VARCHAR(20), atm_manufacturer VARCHAR(50),
atm_location_id INT,
```

primary key(atm_id),
foreign key(atm_location_id)
references etl_schema.loc_dimen(location_id)

);

Query to create a transaction fact table:
create table etl_schema.txn_fact
(

trans_id BIGINT,
atm_id INT,
weather_loc_id INT,
date_id INT,
card_type_id INT,
atm_status VARCHAR(20), currency VARCHAR(10),
service VARCHAR(20), transaction_amount INT, message_code
VARCHAR(255), message_text VARCHAR(255), rain_3h
DECIMAL(10,3), clouds_all INT,

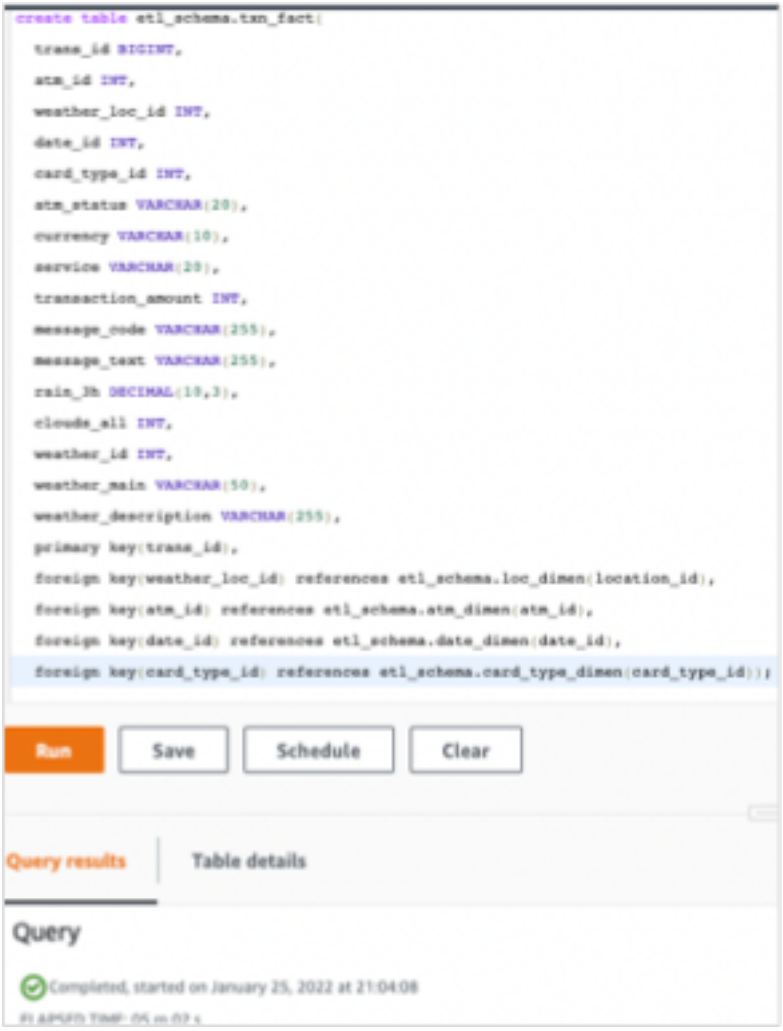
weather_id INT,
weather_main VARCHAR(50), weather_description
VARCHAR(255), primary key(trans_id),
foreign key(weather_loc_id) references

etl_schema.loc_dimen(location_id), foreign key(atm_id)
references

etl_schema.atm_dimen(atm_id), foreign key(date_id) references

etl_schema.date_dimen(date_id),
foreign key(card_type_id)
references etl_schema.card_type_dimen(card_type_id)

);



3. Loading data into a Redshift cluster from Amazon S3 bucket

Queries to copy the data from S3 buckets to the Redshift cluster in the appropriate tables

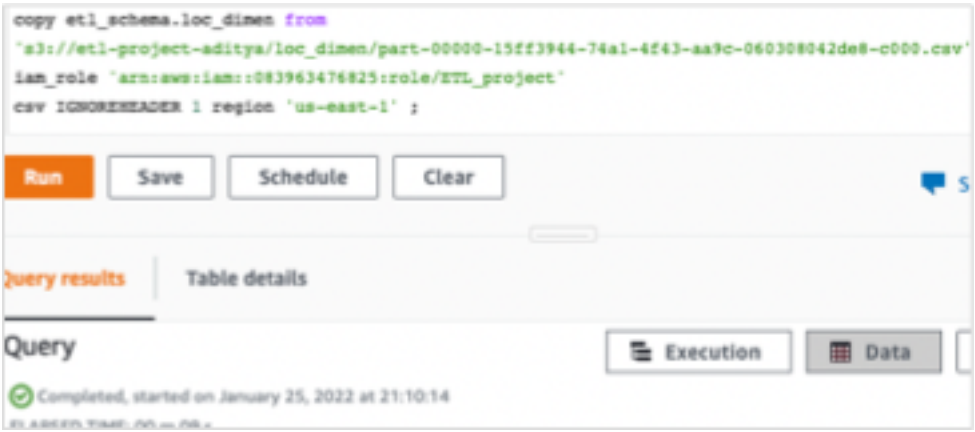
Query to load data into loc_dimen:

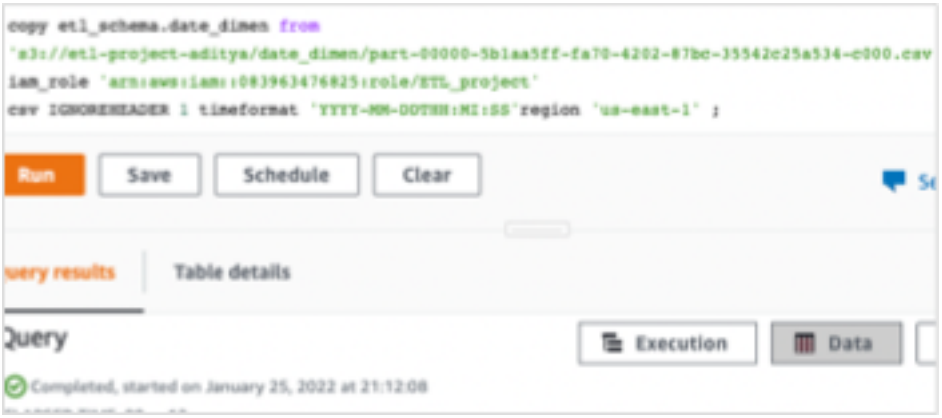
copy etl_schema.loc_dimen from 's3://etl-project-aditya/loc_dimen/part-00000-15ff3944-74a1-4f43-aa9c-060308042de8- c000.csv'

iam_role 'arn:aws:iam::083963476825:role/ETL_project'
csv IGNOREHEADER 1 region 'us-east-1' ;

Query to load data into date_dimen:

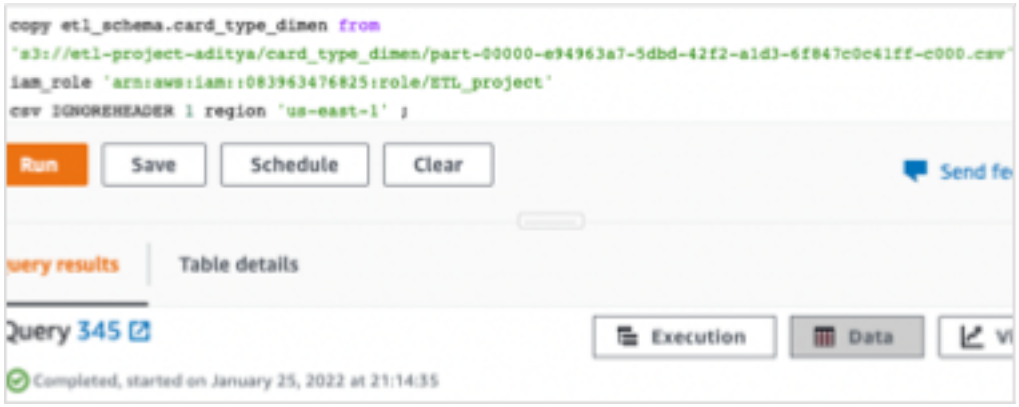
copy etl_schema.date_dimen from 's3://etl-project-aditya/
date_dimen/part-00000-5b1aa5ff-fa70-4202-87bc-35542c25a534-
c000.csv'
iam_role 'arn:aws:iam::083963476825:role/ETL_project'
csv IGNOREHEADER 1 timeformat 'YYYY-MM-
DDTHH:MI:SS'region 'us-east-1' ;





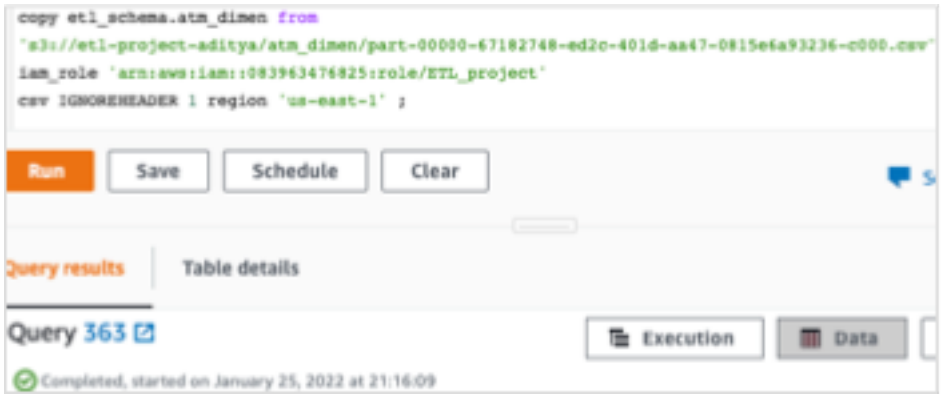
Query to load data into card_type_dimen:

copy etl_schema.card_type_dimen from 's3://etl-project-aditya/
card_type_dimen/part-00000-e94963a7-5dbd-42f2-
a1d3-6f847c0c41ff-c000.csv'
iam_role 'arn:aws:iam::083963476825:role/ETL_project'
csv IGNOREHEADER 1 region 'us-east-1' ;



Query to load data into atm_dimen:

copy etl_schema.atm_dimen from 's3://etl-project-aditya/ atm_dimen/part-00000-67182748-ed2c-401d- aa47-0815e6a93236-c000.csv'
iam_role 'arn:aws:iam::083963476825:role/ETL_project'
csv IGNOREHEADER 1 region 'us-east-1' ;

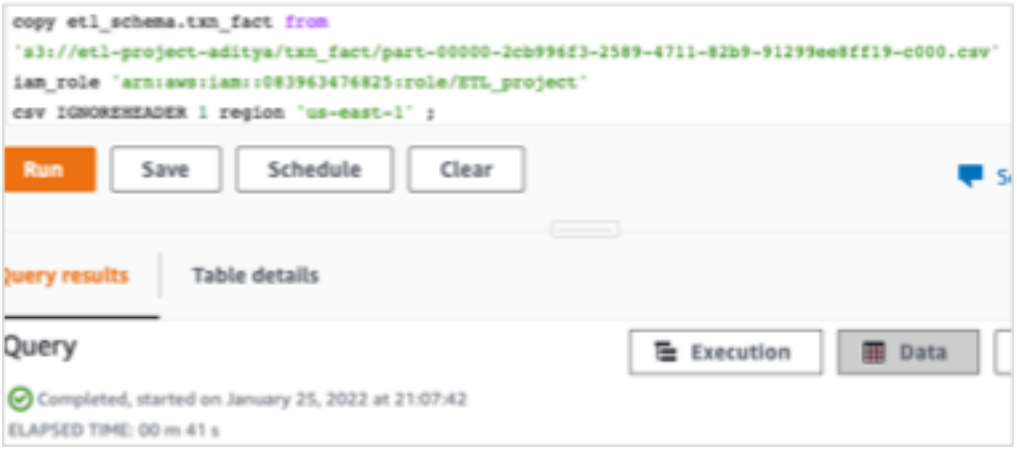


Query to load data into txn_fact:

copy etl_schema.txn_fact from 's3://etl-project-aditya/txn_fact/ part-00000-2cb996f3-2589-4711-82b9-91299ee8ff19- c000.csv'
iam_role 'arn:aws:iam::083963476825:role/ETL_project'
csv IGNOREHEADER 1 region 'us-east-1' ;

4. Query to check number of record in each table:

select count(*) from etl_schema.txn_fact;
select count(*) from etl_schema.loc_dimen;
select count(*) from etl_schema.date_dimen; select count(*) from etl_schema.atm_dimen; select count(*) from etl_schema.card_type_dimen;


copy etl_schema.txn_fact from
's3://etl-project-aditya/txn_fact/part-00000-2cb996f3-2589-4711-82b9-91299ee8ff19-c000.csv'
iam_role 'arn:aws:iam::083963476825:role/ETL_project'
csv IGNOREHEADER 1 region 'us-east-1' ;
Run
Save
Schedule
Clear
Query results
Table details
Query
Execution
Data
Completed, started on January 25, 2022 at 21:07:42
ELAPSED TIME: 00 m 41 s