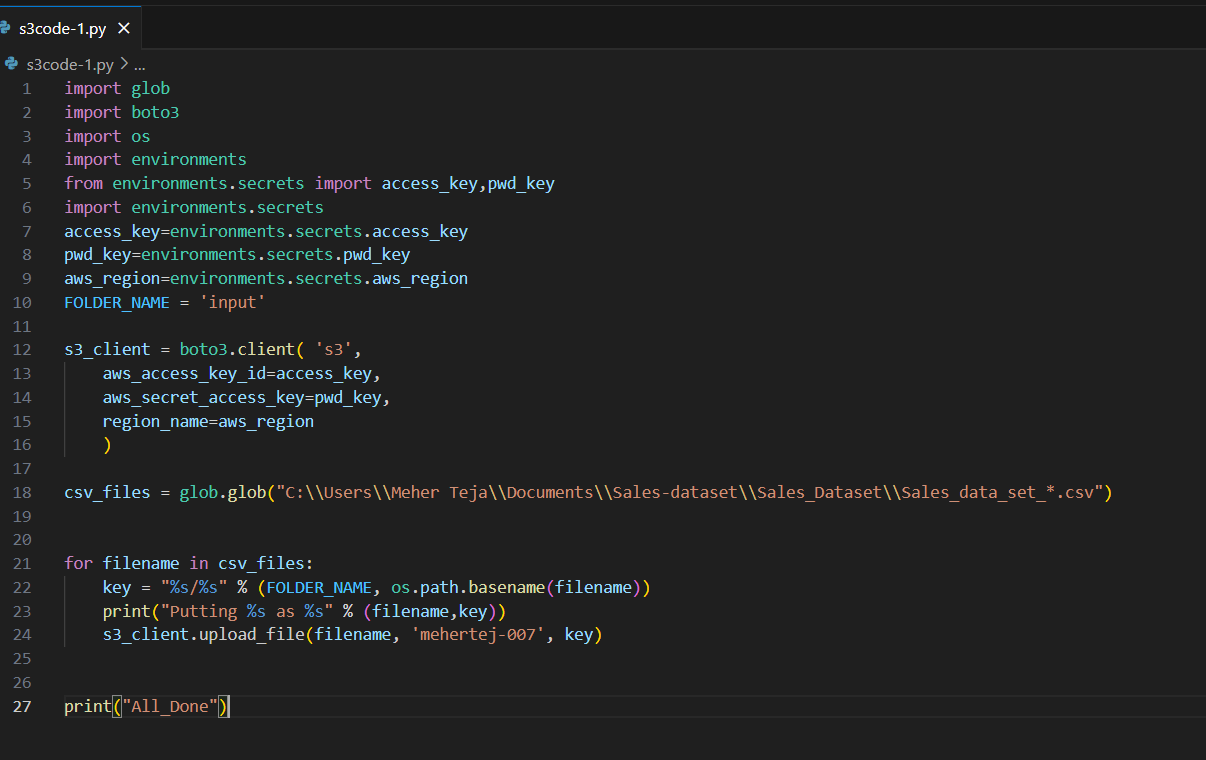
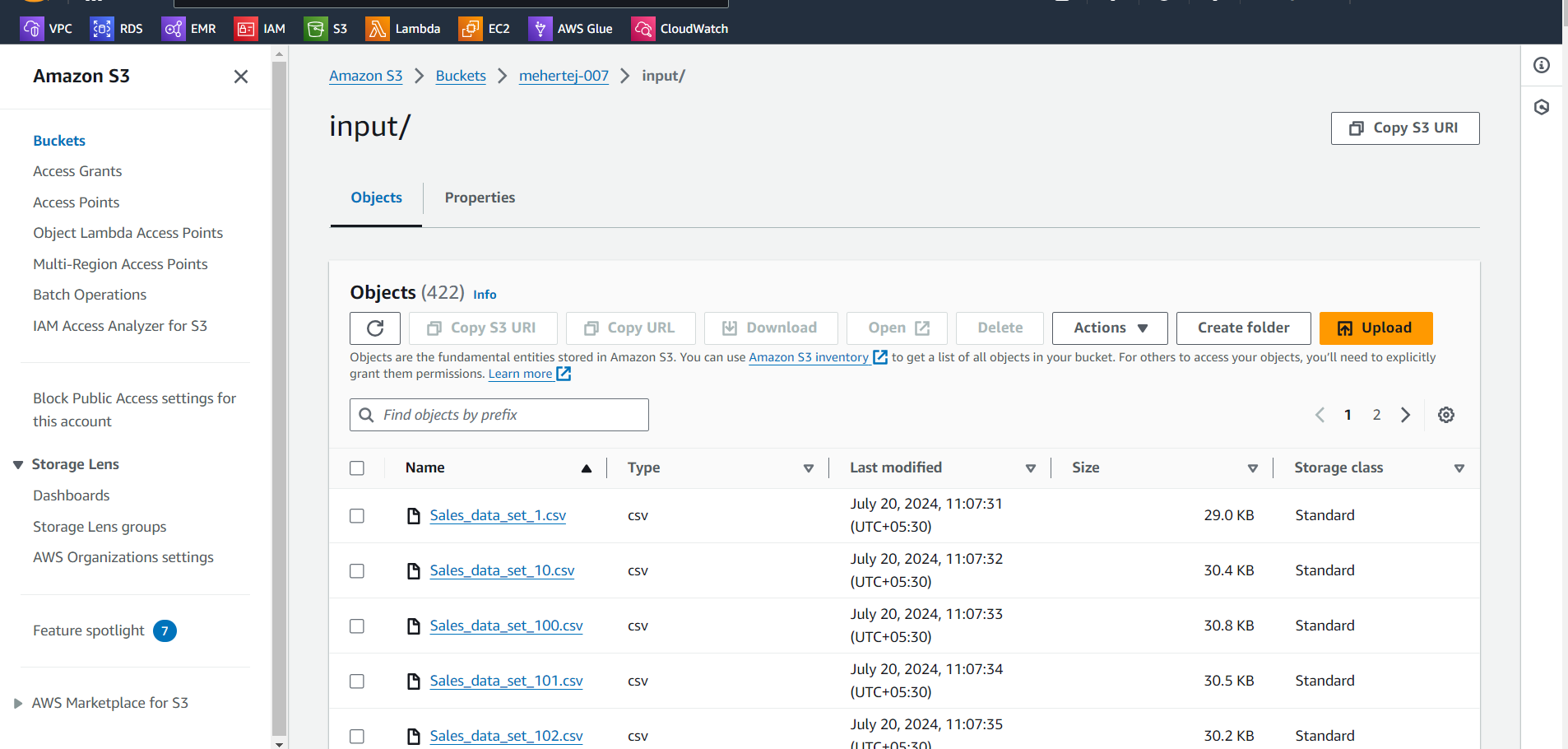
**Final Capstone Project: Retail Stream**

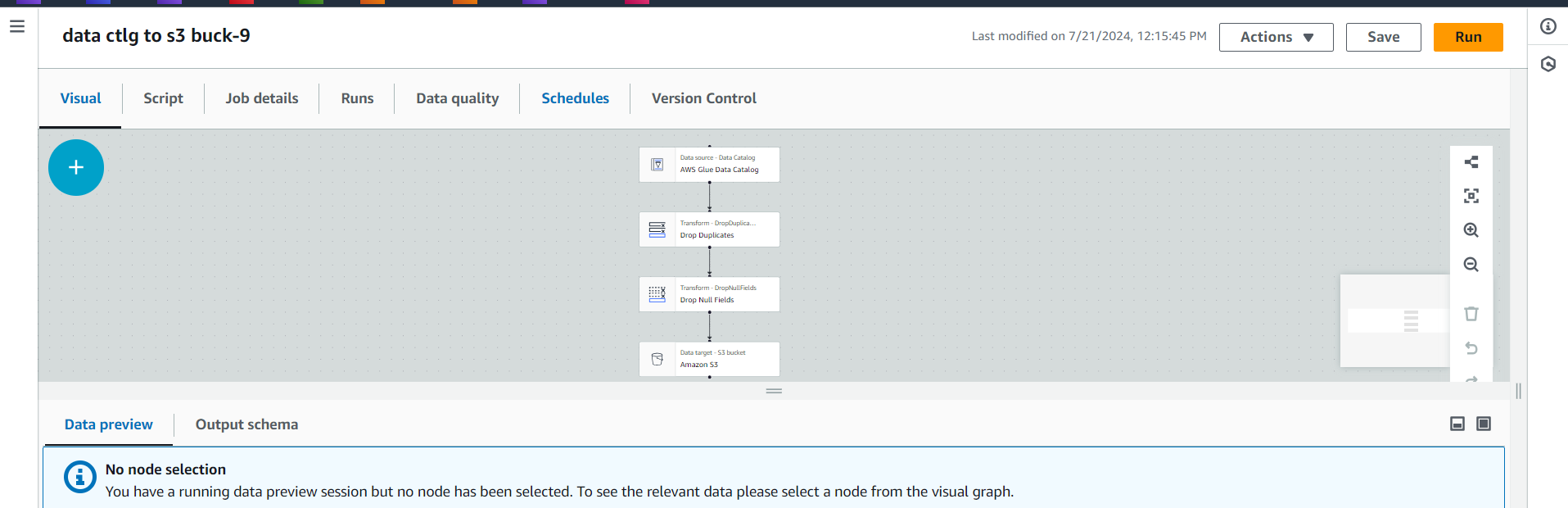
Stage 1:

Push sales data set from local to s3 bucket using python



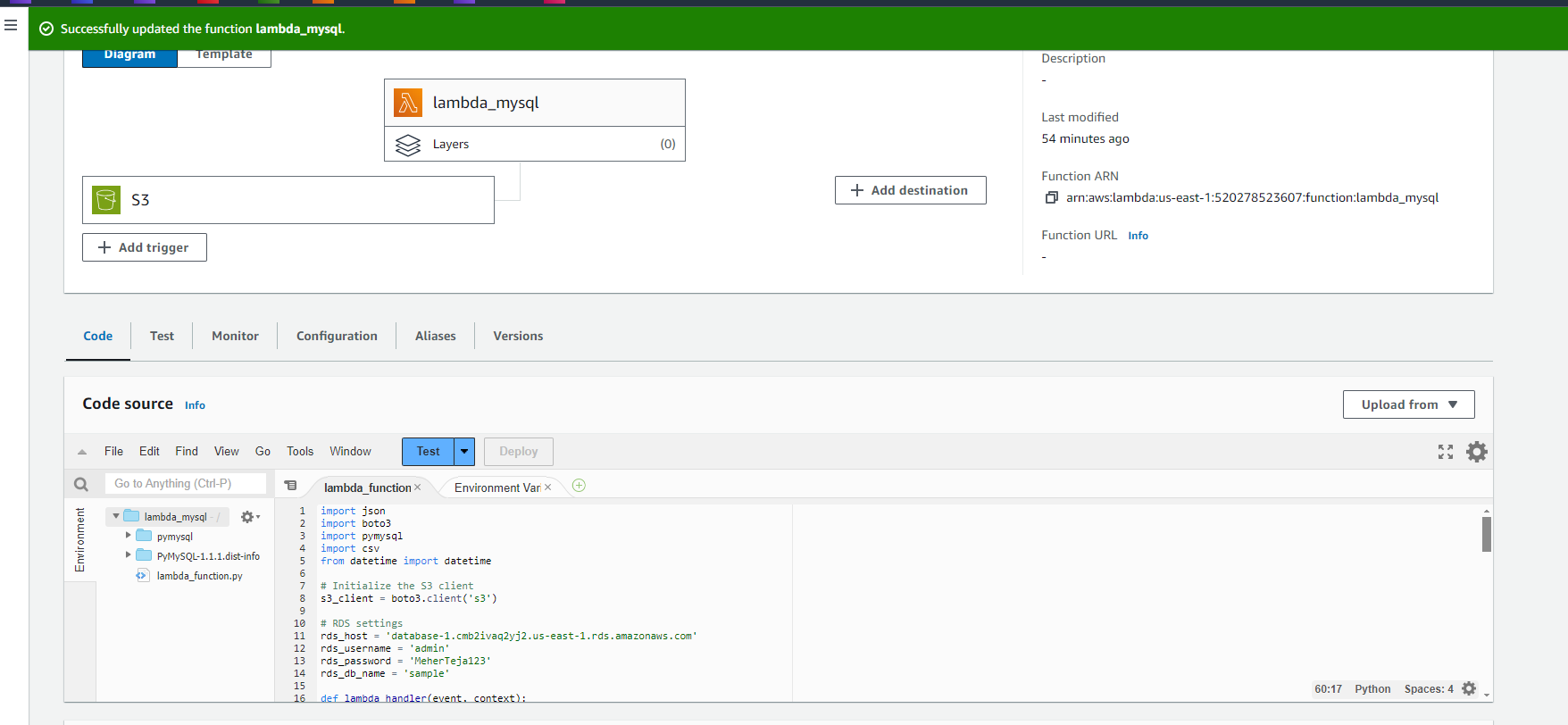


once data is available in S3 , do data cleaning and merge all files in S3 in another bucket (new bucket) using Glue with schedule

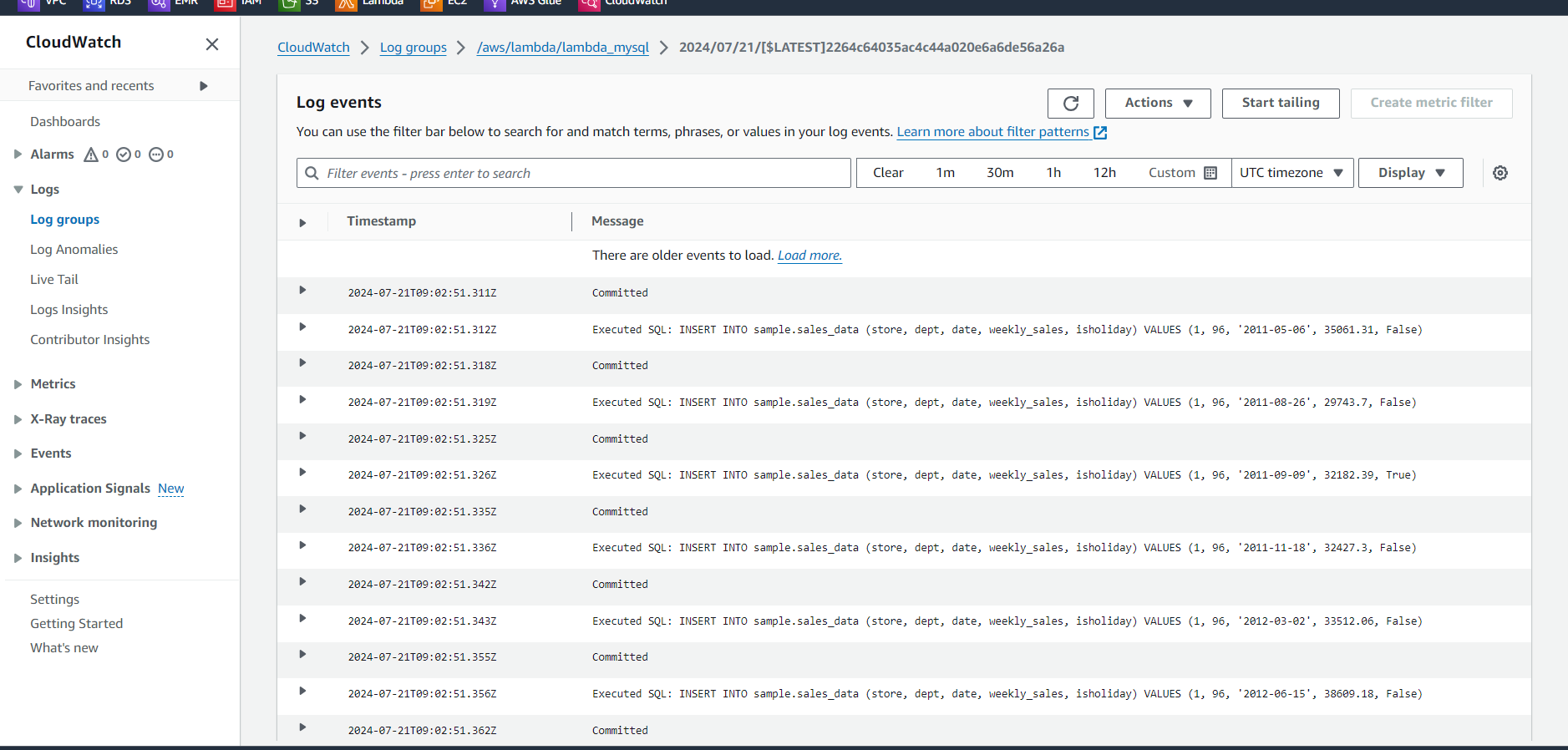


Schedule interval: 5 minutes

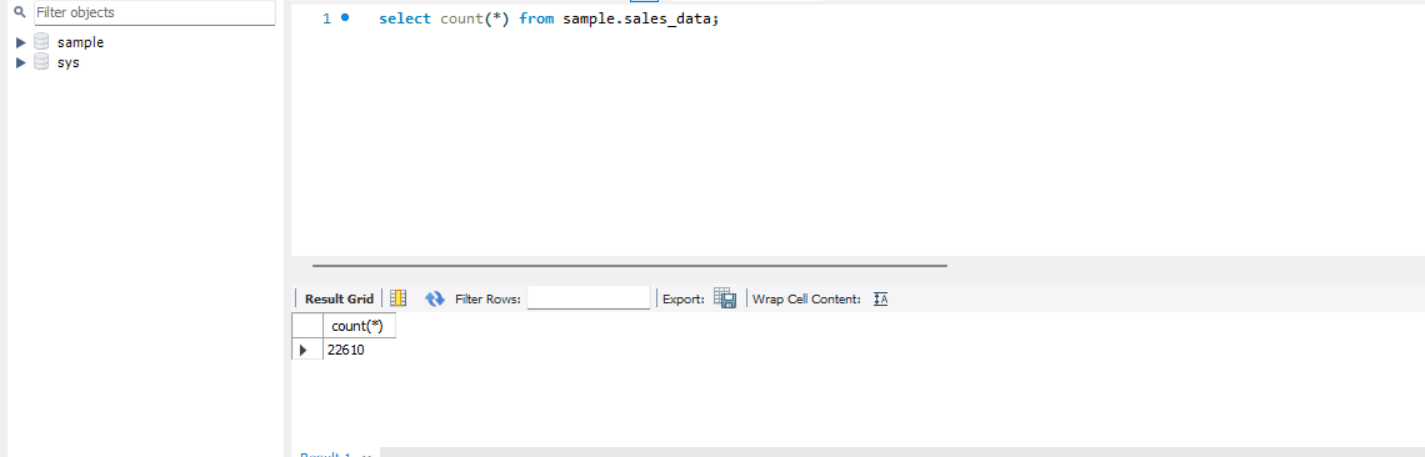
And store to mysql rds using aws lambda python



Cloudwatch logs:

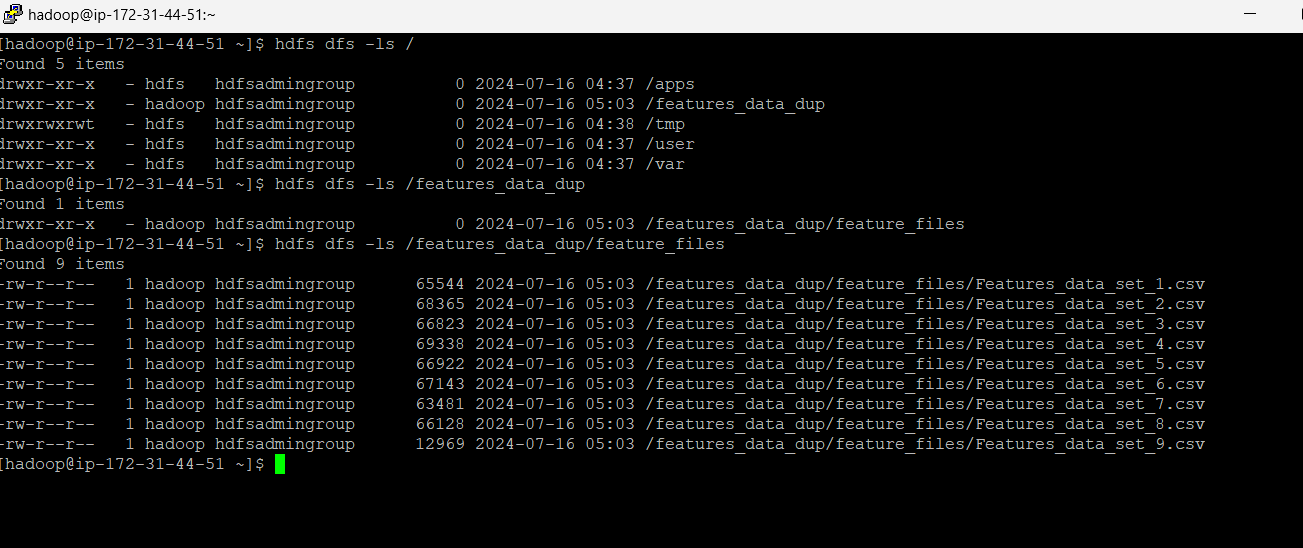


Verify the data in aws rds mysql

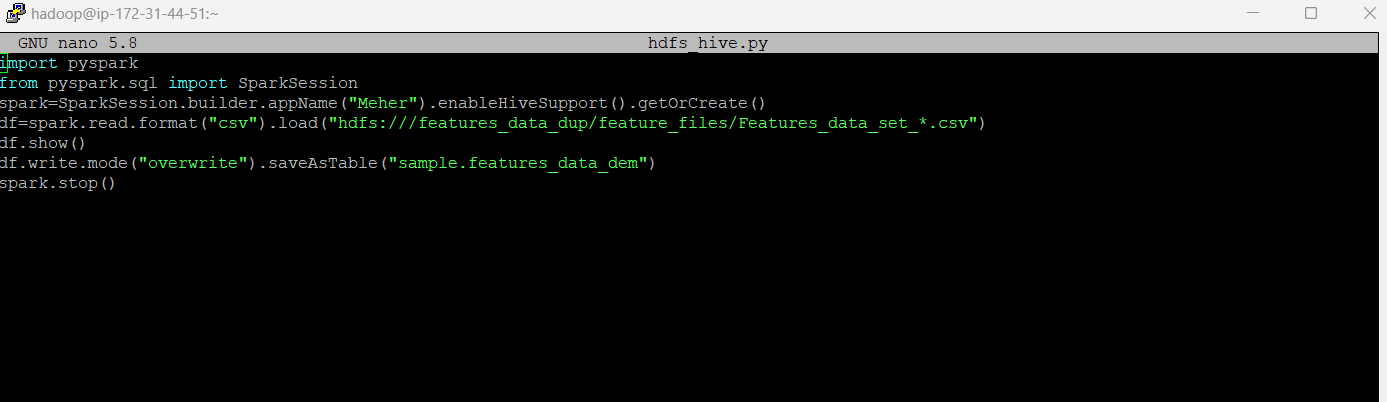


Stage 2:

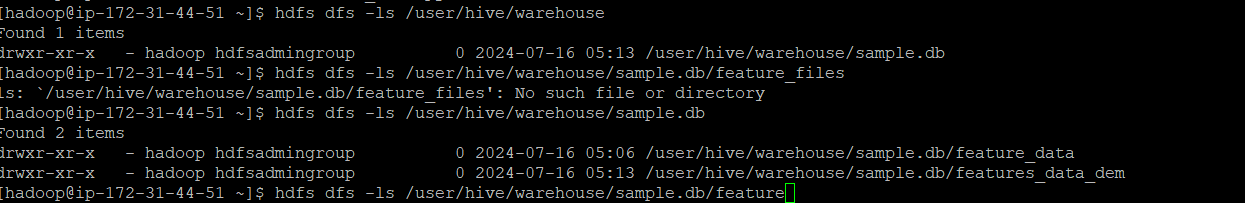
1. Pass all feature files and store it in HDFS

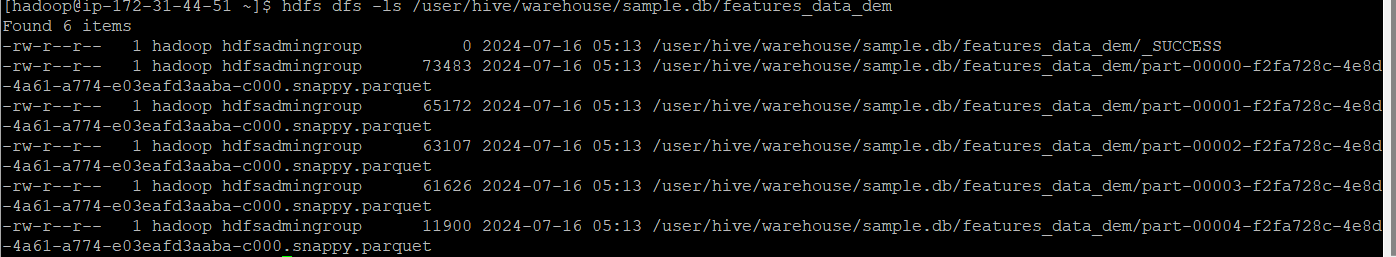


1. Pushing data from hdfs to hive using pyspark



1. Verify whether the data is available in hive in features\_data\_dem table in /user/warehouse/sample path in hdfs





Stage 3:

Push Store Data to AWS RDS

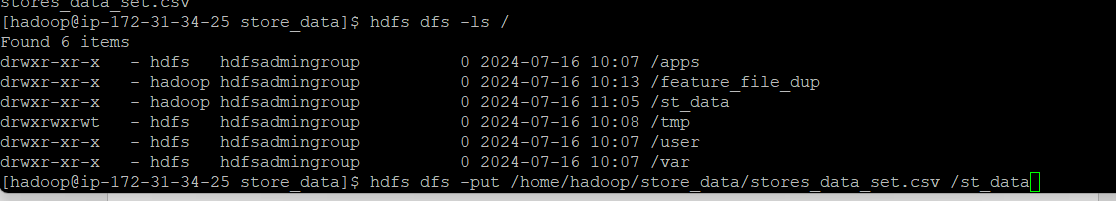
Step 1:

Copy file from s3 bucket



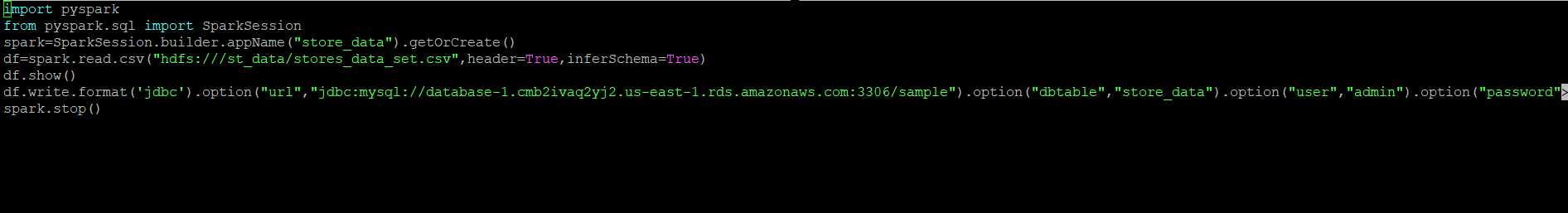
Step 2:

Push to Hdfs

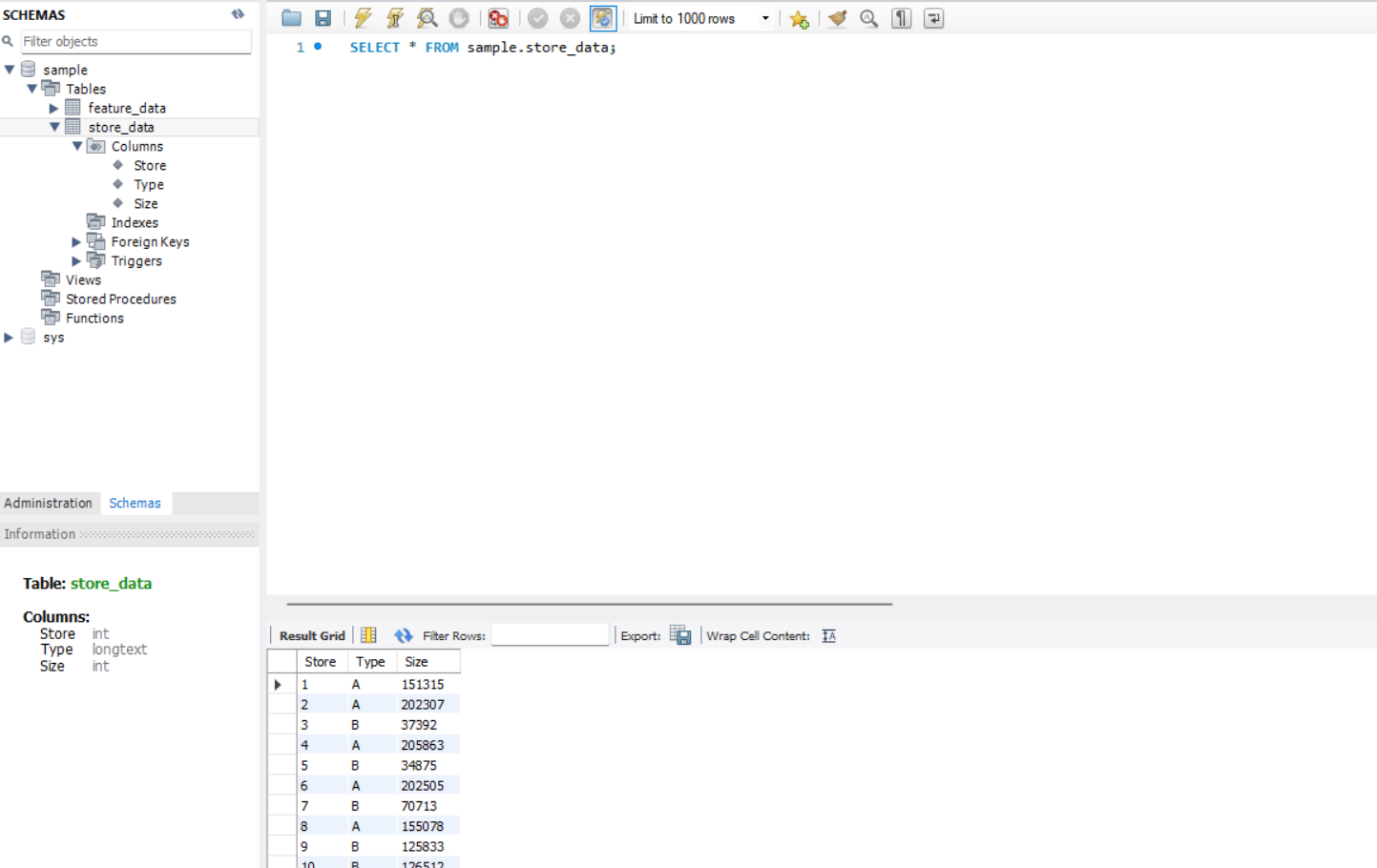


Step 3:

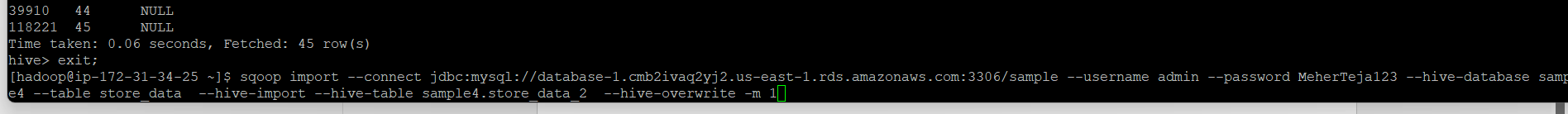
Push the content to RDS using pyspark



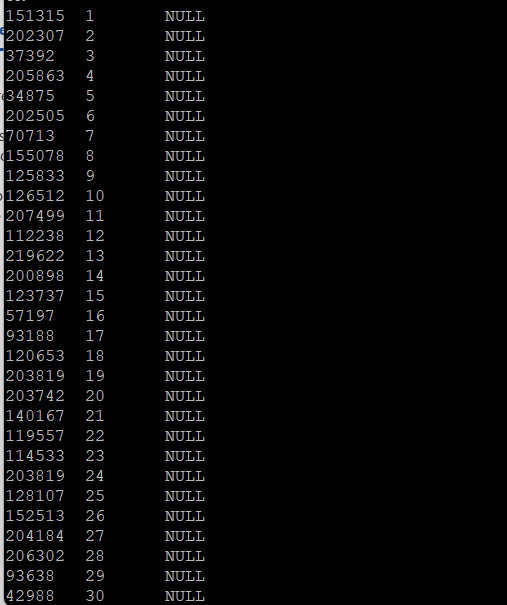
Check whether data is inserted or not in RDS Mysql



Push rds data to hive using sqoop

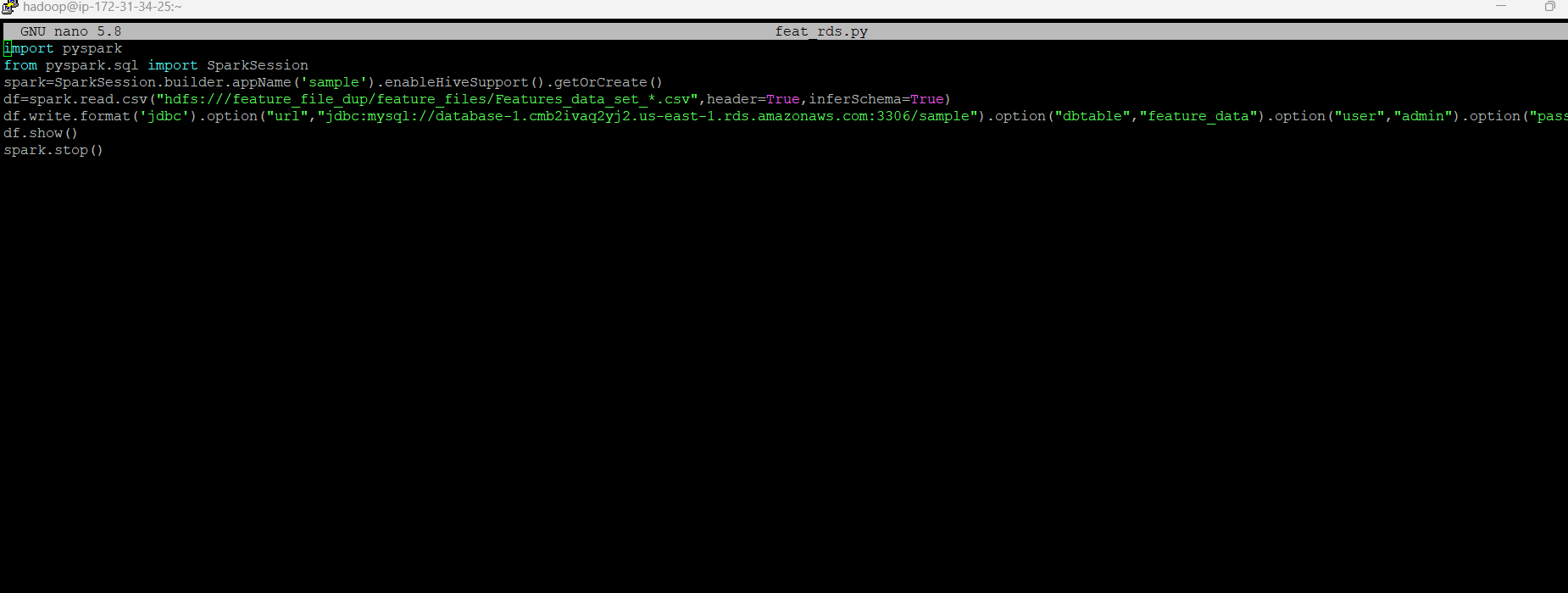


Result:

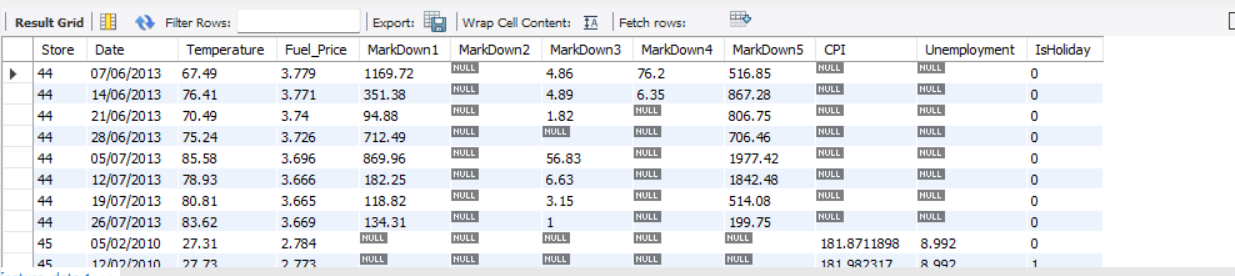


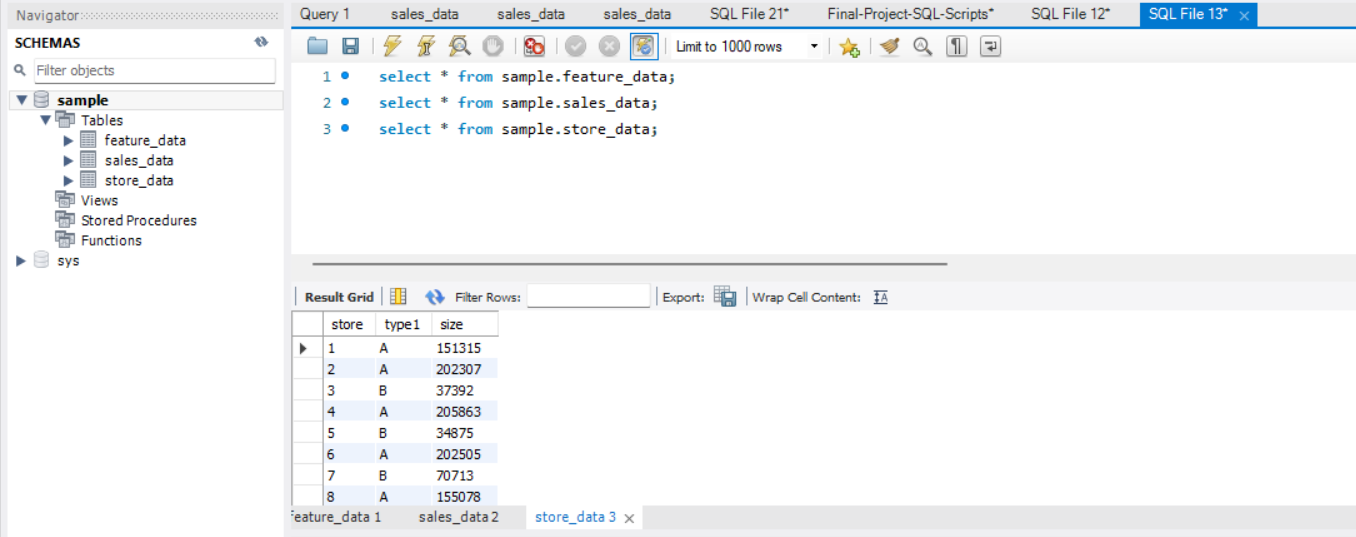
Stage 4

Pushing feature\_data from hdfs to rds using pyspark



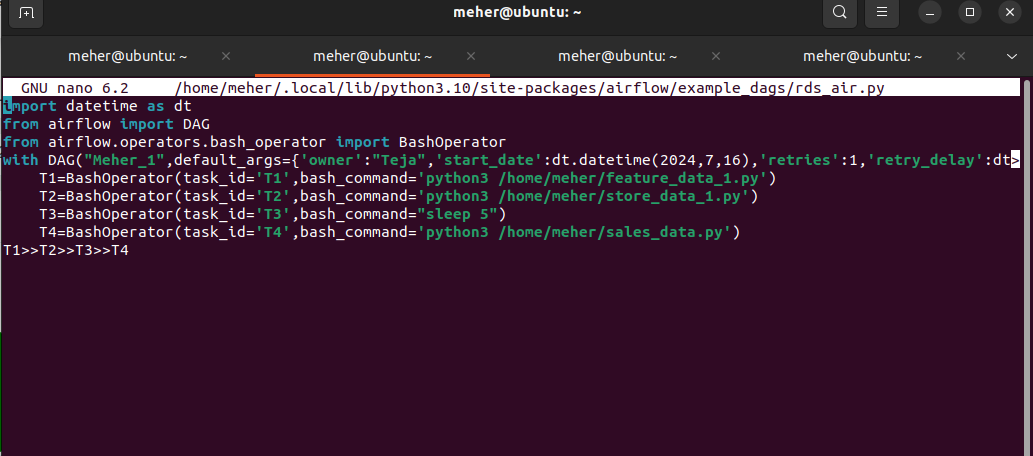
Verify the data in mysql rds

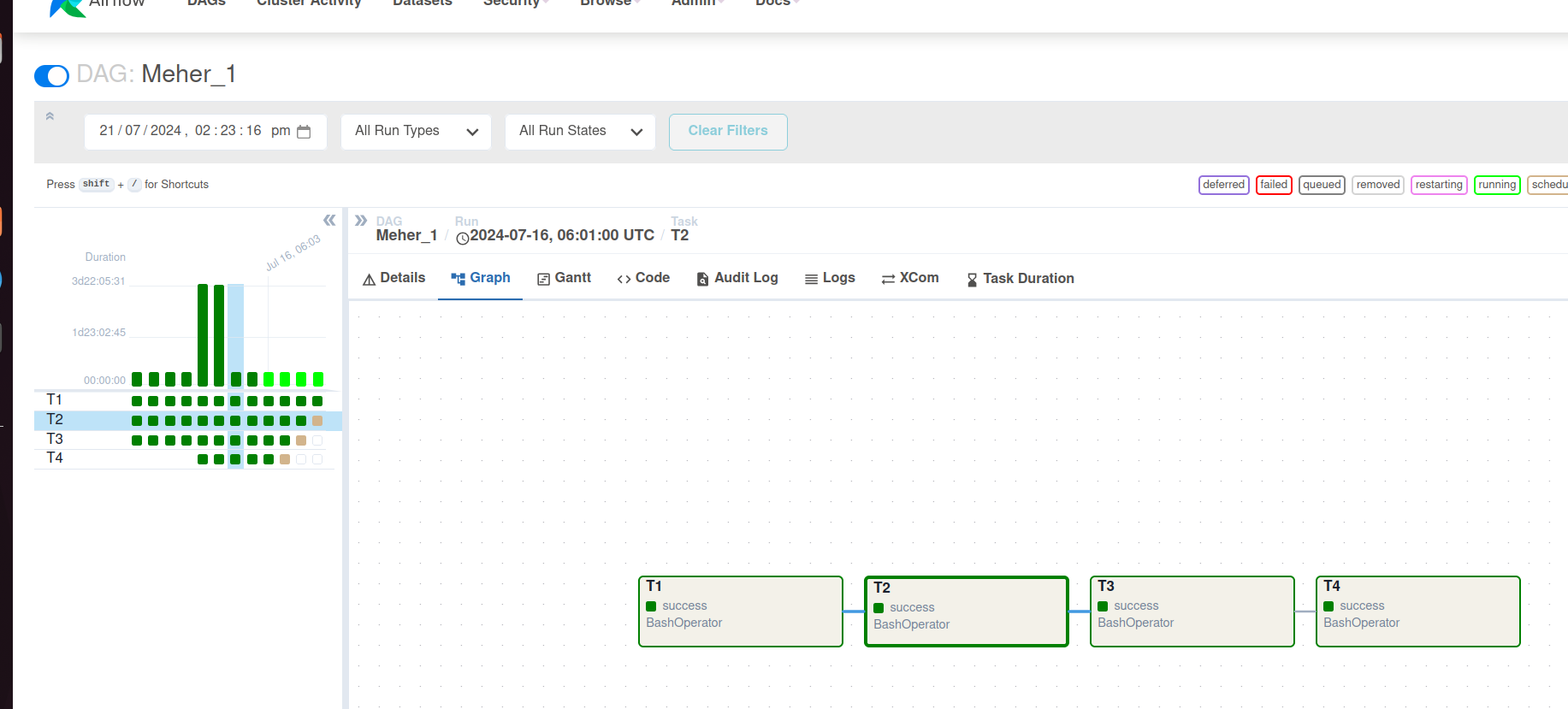




Stage 6:

Monitoring using airflow





DB Queries

**#Query 1:**

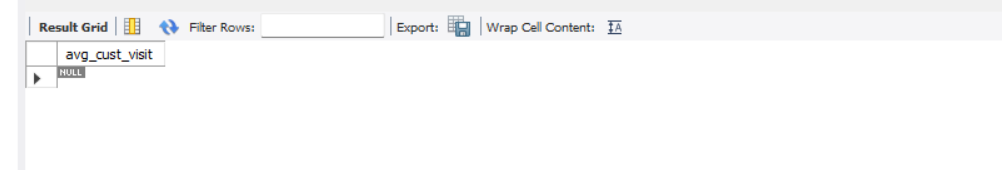
**# Identify average customer visit in the type B store in April Months**

**Query**:

select sum(sd1.weekly\_sales)/fd1.cpi as avg\_cust\_visit from sample.sales\_data sd1 inner join sample.feature\_data fd1 on sd1.store=fd1.store

inner join sample.store\_data std1 on fd1.store=std1.store where std1.type1='B' and Month(fd1.date1)=04;

**Output Snap**



#Query 2:

# Best sales in holiday week for all store types

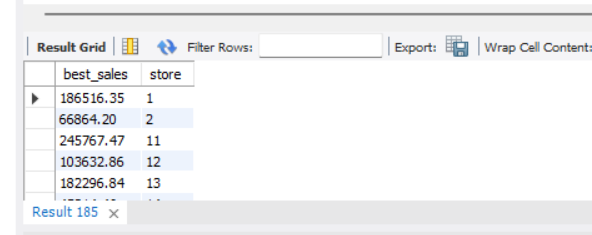
**Query**:

select max(sald1.weekly\_sales) as best\_sales,sald1.store from sample.sales\_data sald1 inner join sample.store\_data sd1 on sald1.store=sd1.store

where sald1.isholiday=1

group by sald1.store;

**Snap**:



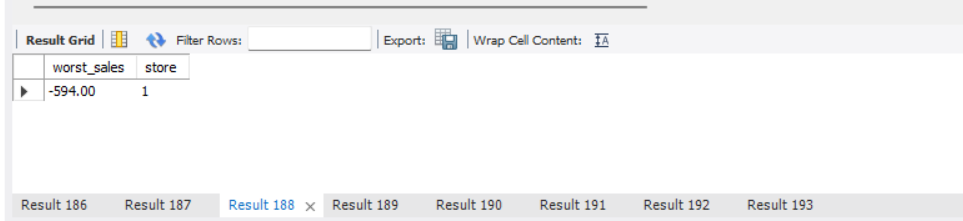
#Query 3:

# Store with worst sales in leap year

**Query:**

select min(sd1.weekly\_sales) as worst\_sales,sd1.store from sample.sales\_data sd1 left join sample.feature\_data sfd1 on sfd1.store=sd1.store where ((sfd1.date1%4=1 or sfd1.date1%400=1) and sfd1.date1%100!=1);

**Snap**:



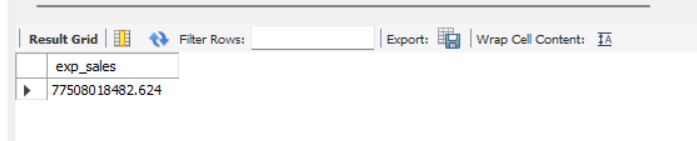
#Query 4:

# What is the expected sales of each department when unemployment factor is greater >8

**Query**:

select (sum(sd1.weekly\_sales)+sum(fd1.markdown1)+sum(fd1.markdown2)+sum(fd1.markdown3)+sum(fd1.markdown4)+sum(fd1.markdown5)) as exp\_sales from sample.sales\_data sd1 inner join sample.feature\_data fd1 on sd1.store=fd1.store where fd1.unemployment>8;

**Snap**:



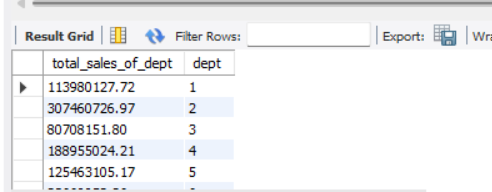
# Query 5:

#Total sales of each department month wise

**Query:**

select sum(weekly\_sales) as total\_sales\_of\_dept,dept from sample.sales\_data sd group by sd.dept;

**Snap**:



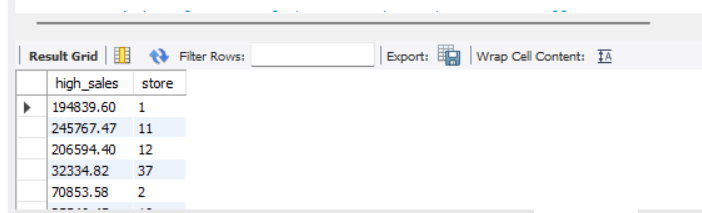
# Query 6

#store with high sales in week wise

Query:

select max(weekly\_sales) as high\_sales,store from sample.sales\_data group by store;

**snap**:



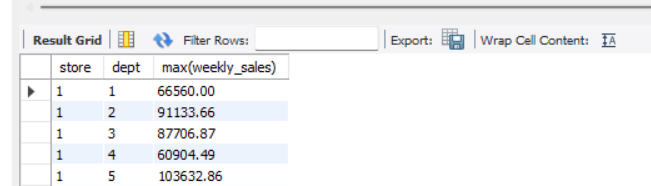
#Query 7

#better dept based performance on the store on all the week

**Query**:

select store,dept,max(weekly\_sales) from sample.sales\_data group by dept;

**Snap**:



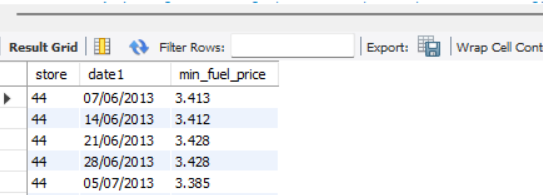
#Query 8

#Identify the store which has minimum fuel price based on the week

**Query:**

select store, date1,min(fuel\_price) as min\_fuel\_price from sample.feature\_data group by date1;

**Snap:**

****

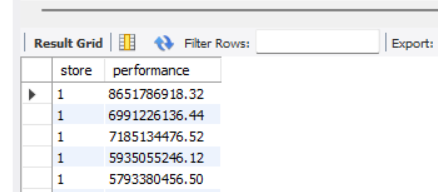
#Query 9

#overall performance of the store based on year wise

**Query**:

select sd1.store,sum(sd1.weekly\_sales) as performance from sample.sales\_data sd1 inner join sample.feature\_data fd1 on sd1.store=fd1.store group by sd1.date1;

**Snap**

****

# Query 10

#Find performance of the store where there are no offers

**Query:**

select sd1.store,sum(sd1.weekly\_sales) as performance from sample.sales\_data sd1 inner join sample.feature\_data fd1 on sd1.store=fd1.store where fd1.markdown1 is NULL and fd1.markdown2 is Null and fd1.markdown3 is Null and fd1.markdown4 is null and fd1.markdown5 is Null;

**Snap**

****