## FLIGHT DATA ANALYSIS USING SNOWFLAKE

## 

**GUIDANCE BY:**

**Mr. Pritam G**

**PRESENTED BY:**

**Vishwanath M**

**Mani Chandu V**

**Tirupathi CH**

**Naveen J**

**Naziya A**

## CONTENTS

## 1.INTRODUCTION

## 2.DATA SPLITTING

## 3.DATA LOADING

## 4.STORAGE INTEGRATION

## 5.EXTERNAL STAGE

## 6.TASK

## 7.SCD2

## 8.SQL QUERIES

## 9.ROW LEVEL SECURITY

## 10.GIT HUB

## 11.CONCLUSION

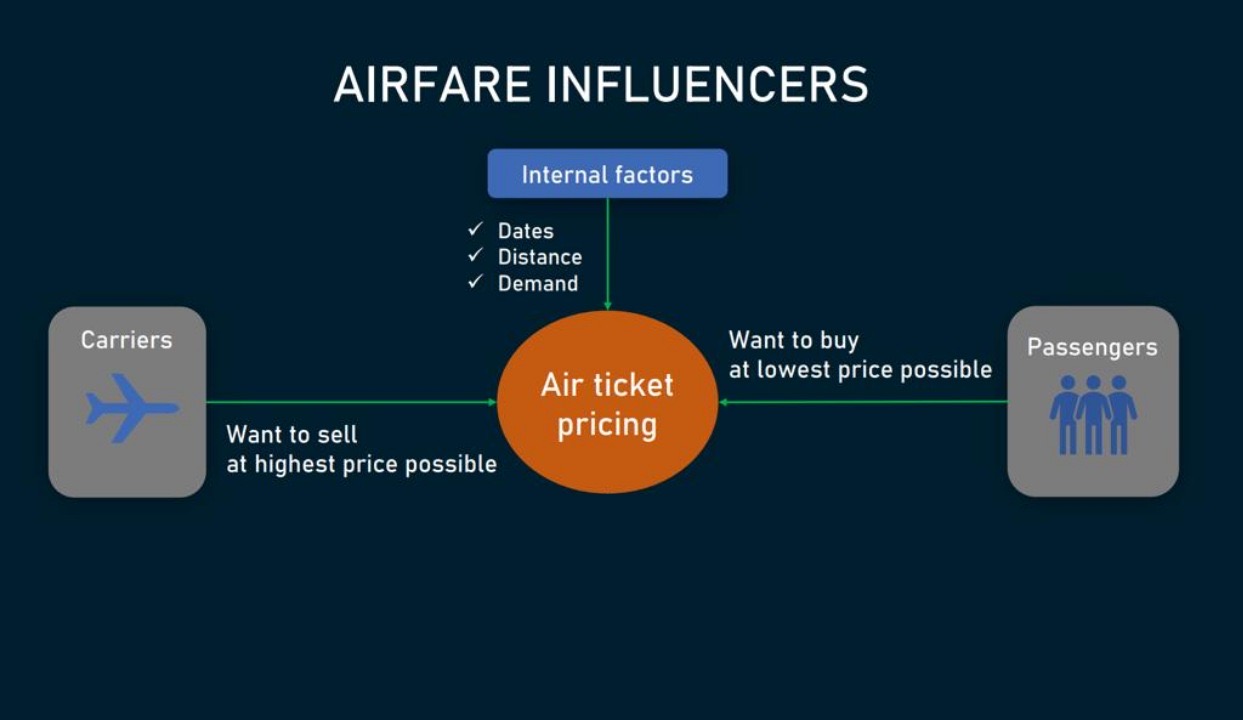
## FLIGHT DATA ANALYSIS USING SNOWFLAKE

## Introduction

* **AIM**: Implementing the snowflake methodologies for the given data extraction from AWS s3 Bucket.
* **SYSTEM** **REQUIRED**: Window OS, Snowflake, AWS, Kaggle.
* **PLATFORM’s** **USED**: Colab, Snowflake, AWS, GIT-HUB.
* **LANGUAGES** **USED**: PYTHON, SQL

**Overview**

In this project, **analyzing the flight price prediction dataset** using essential exploratory data analysis techniques is done then will **draw some predictions about the price of the flight based on some features** such as what type of airline it is, what is the arrival time, what is the departure time, what is the duration of the flight, source, destination and more.



## About the dataset

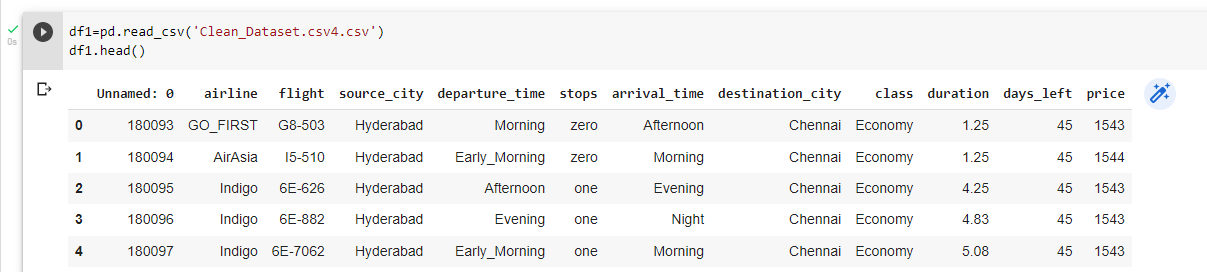
1. **Airline:** So this column will have all the types of airlines like Indigo, Jet Airways, Air India, and many more.
2. **Flight:** This column will let us know about the flight on which the passenger’s journey are about to travel.
3. **Source city:** This column holds the name of the place from where the passenger’s journey will start.
4. **Departure\_Time:** This column will let us know the departure time of the passenger.
5. **Total\_Stops:** This will let us know in how many places flights will stop there for the flight in the whole journey.
6. **Arrival\_Time:** Arrival time is when the passenger will reach his/her destination.
7. **Destination\_city:** This column holds the name of the place to where passengers wanted to travel.
8. **Class:** Here we can know about that what is the class in which passengers have opted to travel from his/her source to their destination.
9. **Duration:**Duration is the whole period that a flight will take to complete its journey from source to destination.
10. **Days\_left:** In this column, we will get information about the days left.
11. **Price:** Price of the flight for a complete journey including all the expenses before onboarding.

**STEP 1**: **DOWNLOAD AND SPLITTING DOCUMENT INTO CHUNKS**

* Importing pandas library.



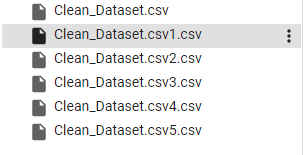
* Reading the dataset using pandas library and viewing it.



* Python Code to divide dataset into chunks.



* Divided dataset into five chunks



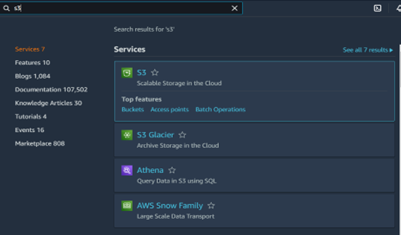
**SNOWFLAKE**

**Snowflake** is a fully managed SaaS (software as a service) that provides a single platform for data warehousing, data lakes, data engineering, data science, data application development, and secure sharing and consumption of real-time / shared data.

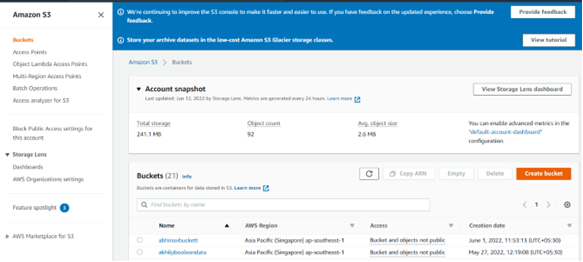
* It is one of MPP system (Missile Parallel Processing).
* Snowflake has high scalability.
* It is modern architecture. (separate storage and compute cost)
* Snowflake is pay per use.
* It loads the data is encrypted format.
* It has two stages external stage and internal stage.

**STEP 2**: **LOAD THE DATA INTO S3 BUCKET**

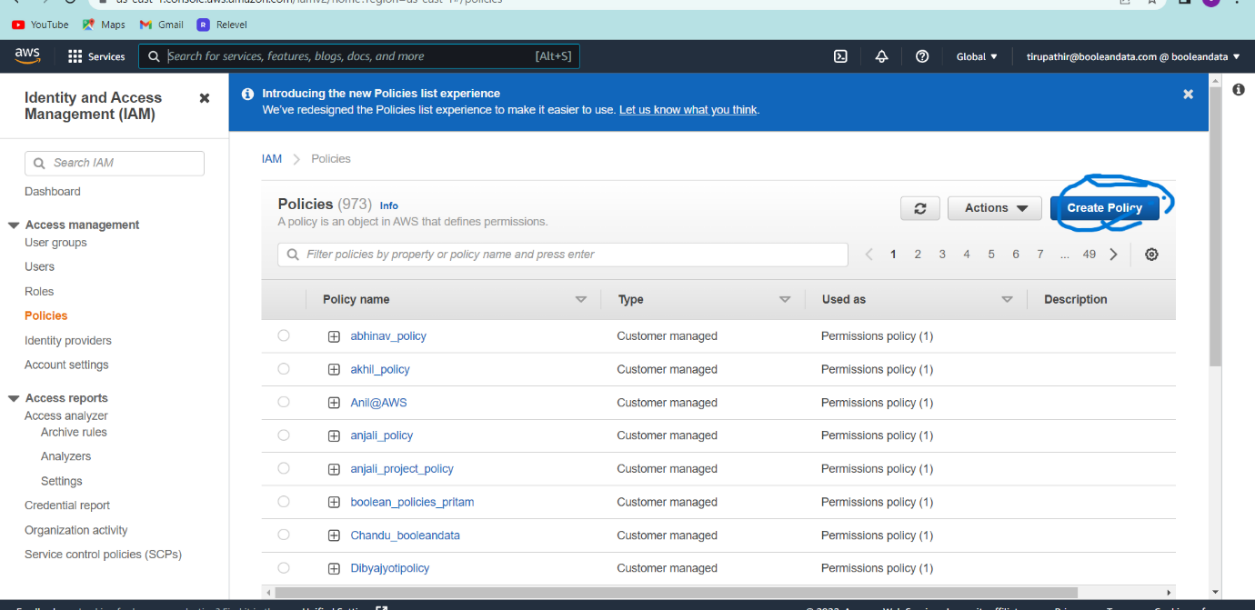
* Open AWS (Amazon Web Services).
* Search and Open S3 bucket.

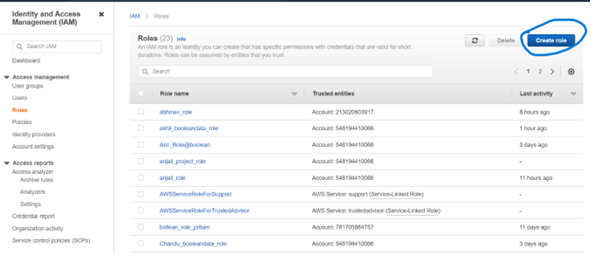


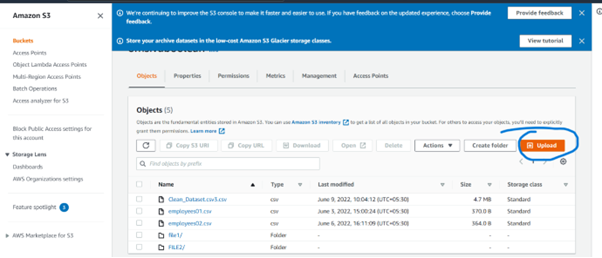
* Create S3 bucket.



* Create Policy.



* Create Role.
* Open the S3 bucket and upload the files and folders which we needed.



**STEP 3: CREATE STORAGE INTEGRATION IN SNOWFLAKE TO LOAD THE DATA CONTINUOUSLY.**

**External stage:**

An external (i.e., S3) stage specifies where data files are stored so that the data in the files can be loaded into a table.

In addition to loading directly from files in S3 buckets, Snowflake supports creating named external stages, which encapsulate all of the required information for staging files, including:

* The S3 bucket where the files are staged.
* The named storage integration object or S3 credentials for the bucket (if it is protected).
* An encryption key (if the files in the bucket have been encrypted).
* The cloud services are amazon web services, Microsoft azure, Google could platform.
* The cloud is used to store the data in different ways.
* The external stage loads the data in either public or private.

1.Create Storage Integration

2.Click on desc integration.

The following are the respected cases and their outputs in order to get a clear idea about the flow of project:

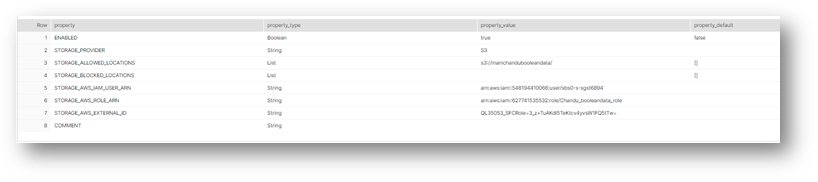
**CASE 1: Creating Integration**

Input:

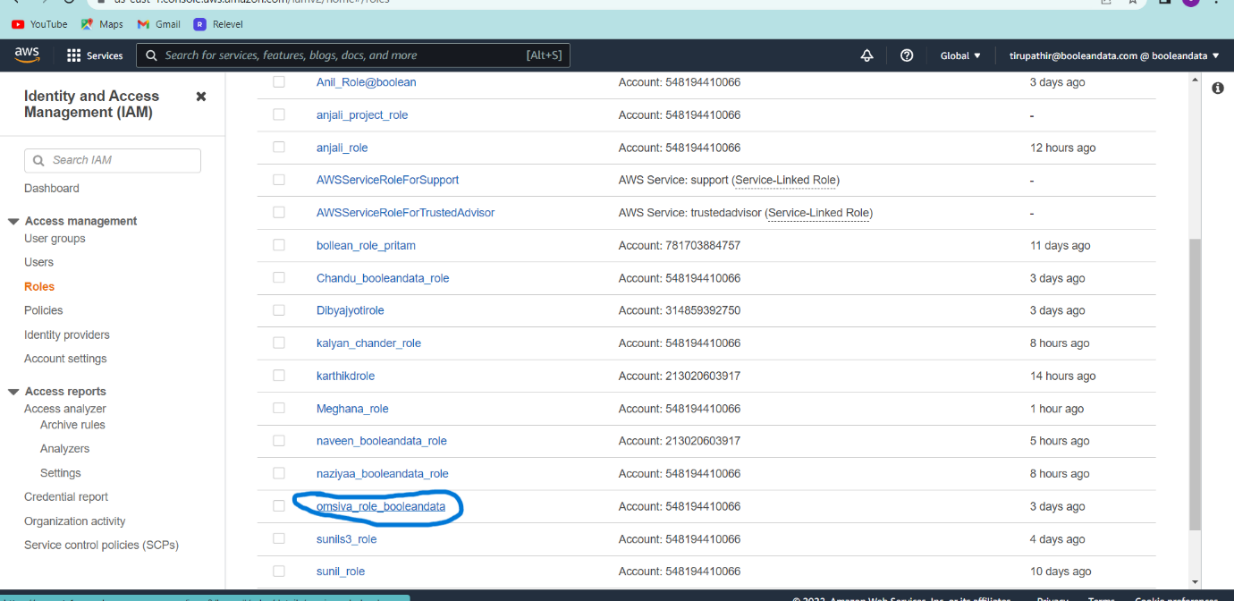


**Expected Output:**

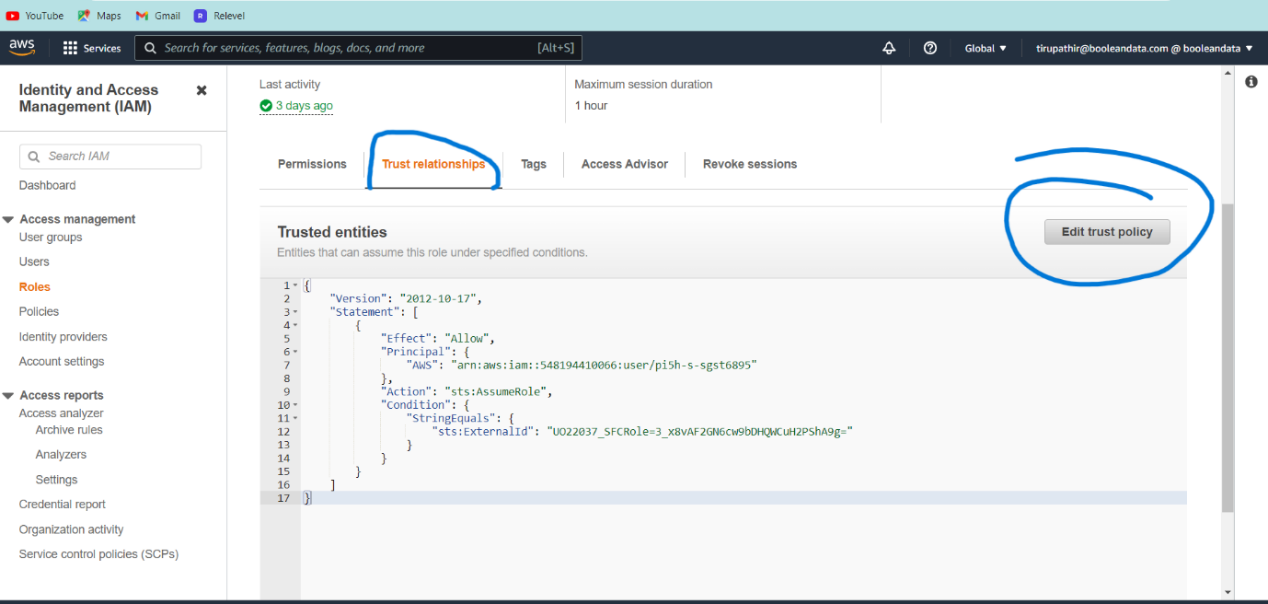
Here we are expecting to get ‘**IAM\_USER\_ARN’** and ‘**EXTERNAL\_ID’.**

**Output**: 

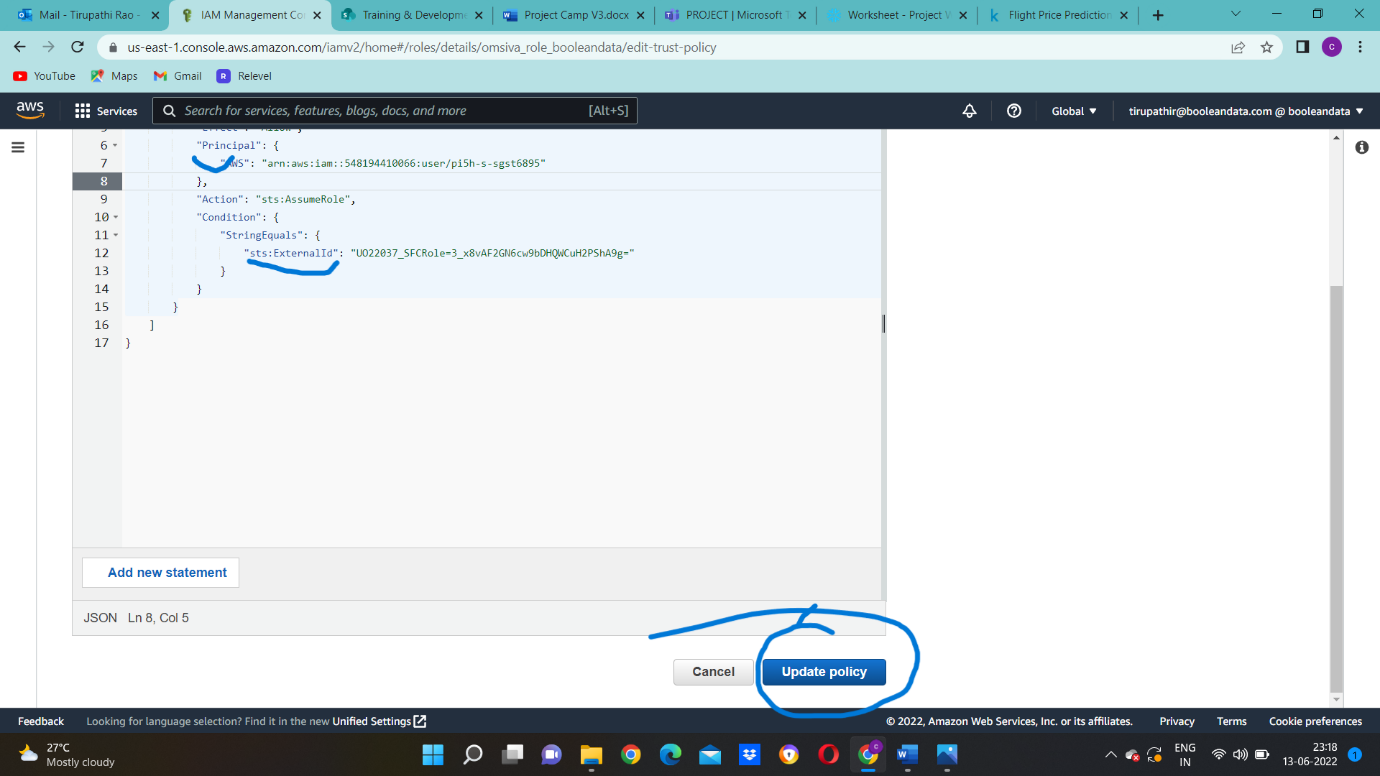
* Open roles and click on your particular role



* Click on trust relationships and click on Edit trust policy.



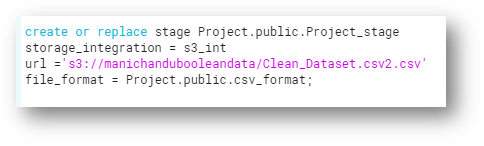
* Change the AWS and external id which are coming from snowflake integration.



**STEP 4: CREATING EXTERNAL STAGE AND LOADING DATA INTO IT.**

**CASE 2: Creating Stage And Loading Data Into It**

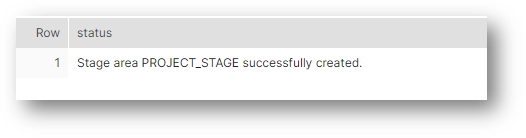
Input:



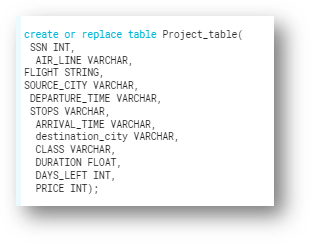
Expected Output:

Here we are expecting to extract data from s3 bucket to the stage in snowflake.

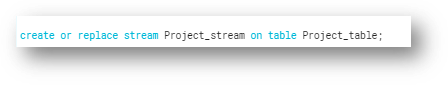
Output:



**CASE 3: Creating A Table And Stream And Loading Data Along With Task.**

Input:

Creating stream on a table:



**STEP 5: CREATE A SCHEDULER TO SCHEDULE THE JOB AT 12:00 AM IST HOURS EVERY THURSDAY TO PERFORM STEP 4**

**Cron:** Scheduling of a job in a specific time in a unique environment

# ┌───────────── minute (0 - 59)

# │ ┌───────────── hour (0 - 23)

# │ │ ┌───────────── day of the month (1 - 31)

# │ │ │ ┌───────────── month (1 - 12)

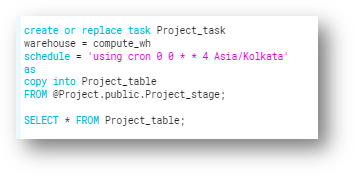
# │ │ │ │ ┌───────────── day of the week (0 - 6) (Sunday to Saturday;

# │ │ │ │ │ 7 is also Sunday on some systems)

# │ │ │ │ │

# │ │ │ │ │

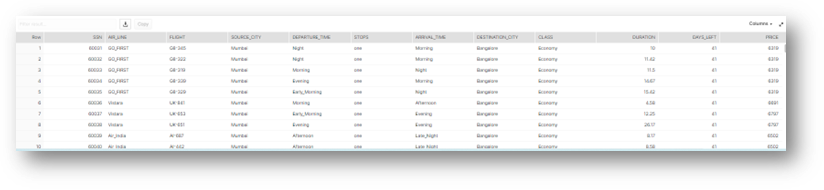
# \* \* \* \* \* <command to execute>



**Expected Output:**

Here we are expecting to insert data into the table along with the stream and task.

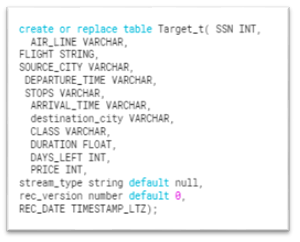
Output:

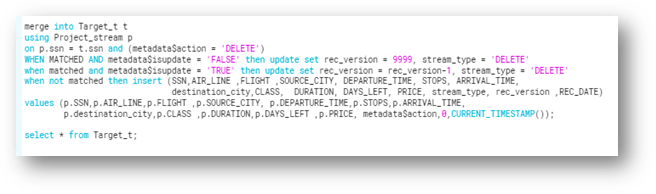


**STEP 6: IMPLEMENT SCD 2 FOR THE ABOVE DATA FLOW**

**CASE 4: Creating A Target Table And Merging Data Into It From Stream:**

**Input:**

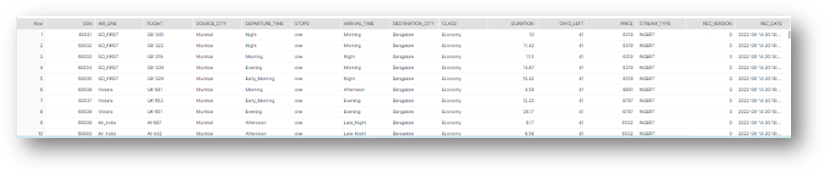




**Expected Output:**

Here we are expecting to extract data from the stream to target table i.e consumer table.

**Output**:

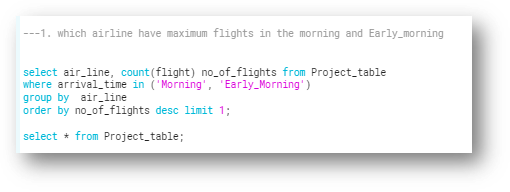


**STEP 7: PERFORMING THE ANALYSIS OF DATASET:**

**CASE 5: Executing SQL Queries:**

**1:** Which airlines have maximum flights in the Morning and Early morning.

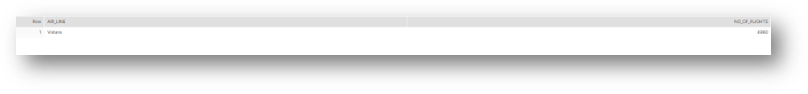
**Input:**

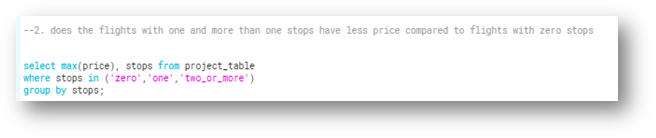


**Expected Output:**

The result should be the maximum flights in the morning and early morning.

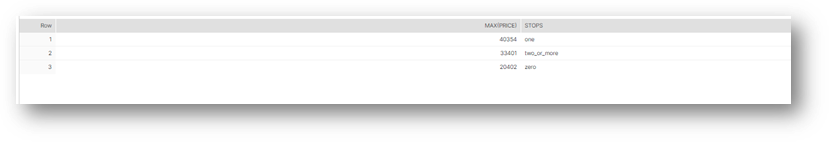
**Output:**

**2:** Does the flights with one and more than one stops have less price compared to flights with zero stops.

**Input:****Expected Output:**

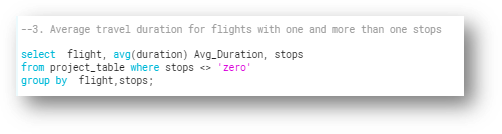
Expecting to get the maximum price of all stops to compare.

**Output:**



**3:** Average travel duration for flights with one and more than one stops.

**Input:**



**Expected Output:**

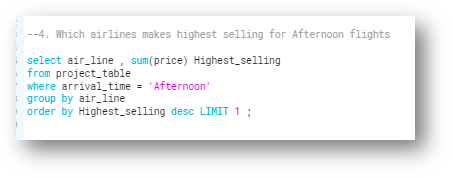
Expecting to get the average of travel duration for the flights with one and more than one stops.

**Output**:



**4:** Which airlines makes highest selling for afternoon flights.

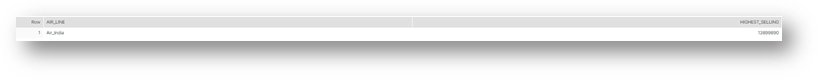
**Input:**



**Expected Output:**

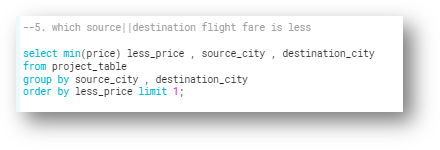
Expecting to get highest selling for afternoon flights.

**Output**:



**5:** Which source || destination flight fare is less

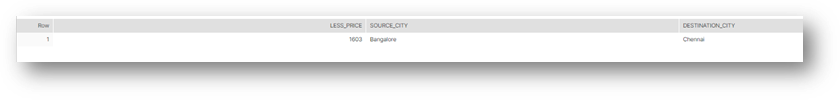
**Input:**



**Expected Output:**

Expecting to get which source || destination flight fare is less

**Output:**



**STEP 8: IMPLEMENTION OF ROW LEVEL SECURITY**

**CASE 6: Implementing Row level security and column level security**

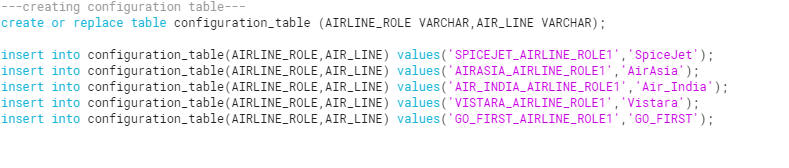
Row-level security, or row-based security, is a data access control concept in which access to data or row in a table is limited according to certain restrictions, and various users, groups or roles may have different permissions on certain rows, based on identities within the rows.

Column level security in snowflake allows application of masking policy to a column within a table or view. These are build to protect the sensitive data from unauthorized users while allowing authorized users to access sensitive data.

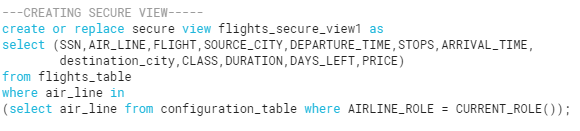
Steps to perform row-level and column level security

**Step-1**: create a configuration table using Account admin role.

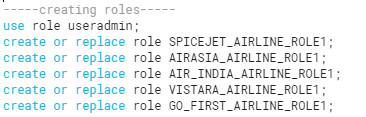
Configuration table is created for role-name and row-name.



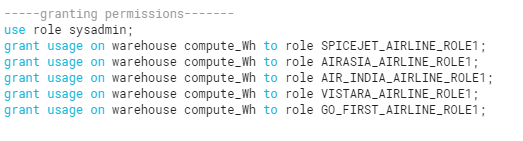
**Step-2**: create a secure view



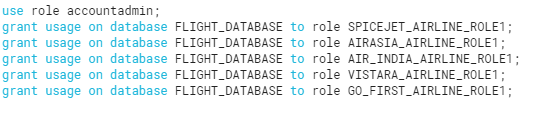
**Step-3**: create roles using user admin role



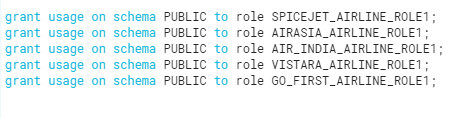
**Step-4**: grant warehouse permissions on roles using sysadmin role



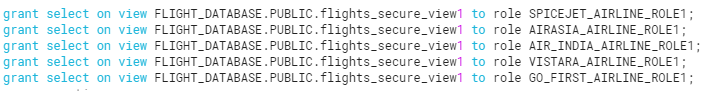
**Step-5**: grant database permissions on roles using Account admin role



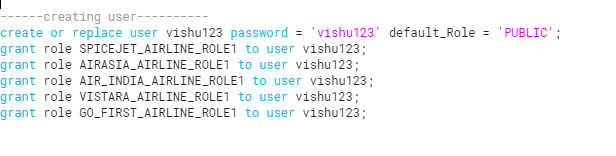
**Step-6**: grant schema permissions on roles using Account admin role



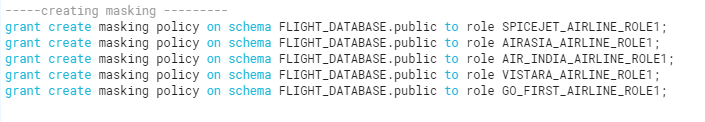
**Step-7**: grant select on secure-view to roles



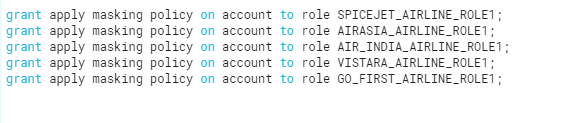
**Step-8**: create a user and assign roles to the user



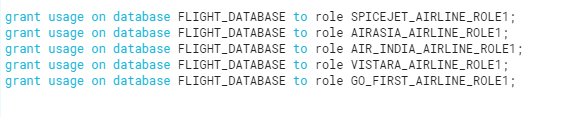
**Step-9**: grant privileges of policy to roles



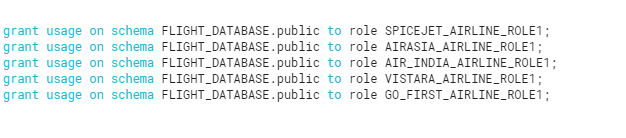
**Step-10**: apply masking policy to roles



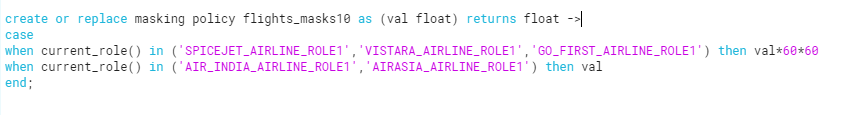
**Step-11**: grant database usage to roles



**Step-12**: grant schema usage to roles



**Step-13**: create masking policy



**Step-14**: check weather masking policy is created or not

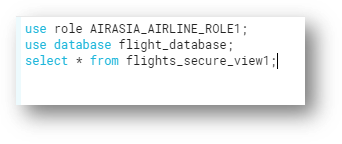


**Step-15**: apply masking policy to column.

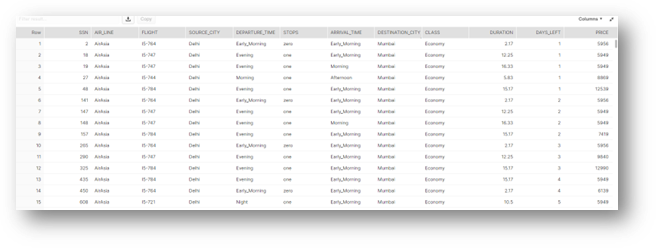


**Result 1:**

**Input:**

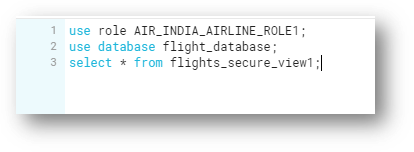


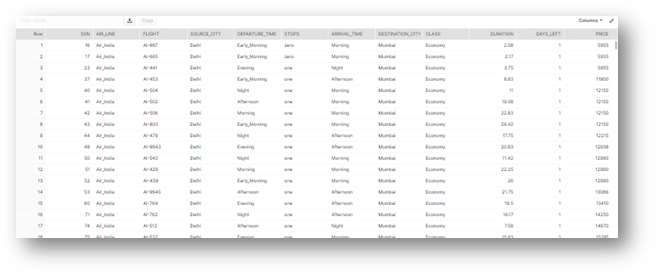
**Output:**



**Result 2:**

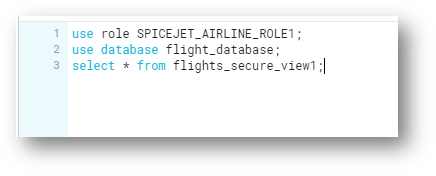
**Input:**



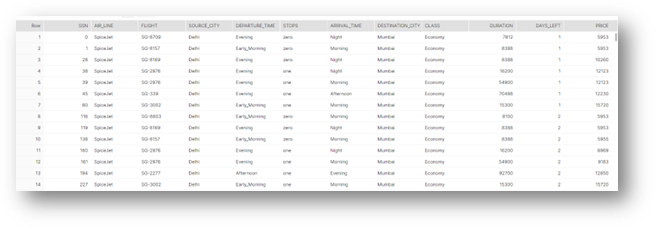
**Output:** 

**Result 3:**

**Input:**

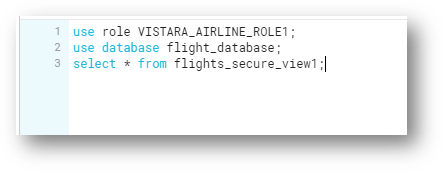


**Output:**

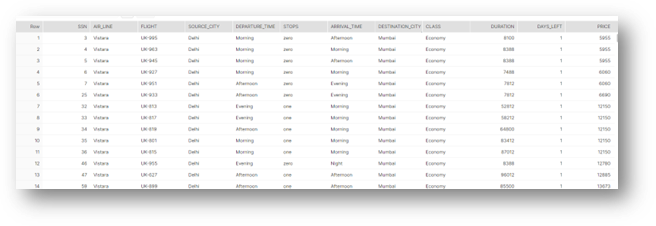


**Result 4:**

**Input:**

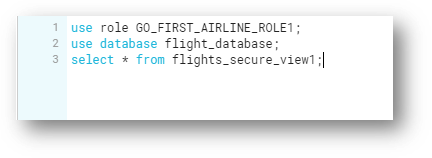


**Output:**

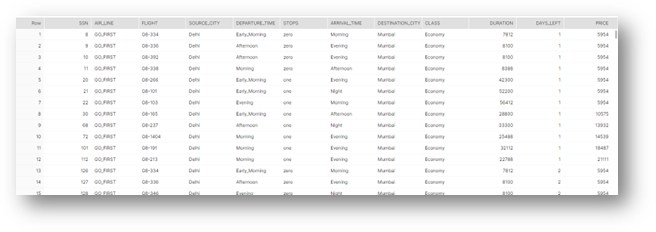


**Result 5:**

**Input:**



**Output:**



**CONCLUSION:**

* We have successfully predicted the price and solved the queries
* We have successfully implemented the updates being done in tables
* We have successfully implemented the row & column level security in our project
* We have observed the different airlines like SpiceJet, air Asia and many different airlines, stops, duration, destination, source city, departure, arrival time in our project
* We have tracked the data and seen the update being done in it
* We have successfully implemented the security policies