CASE STUDY ON AIRLINES ANALYSIS (SPRINT 2)

Batch: BI V7 with MS Azure PT Sep 6th Batch1

Group Members:

1. Harshit Mudgal
2. Abhaya Shankar
3. Armaandeep Sandhu
4. Ashitosh Joshi
5. Abhishek Baghel

INTRODUCTION

In our project scope we have created an end-to-end flow for executing the queries related to Airlines Analysis using Jupyter Notebook and generating visualizations.

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.

Workflow:

- ❖ Open SQL Server Management Studio.
- ❖ Connect to an instance of the SQL Server Database Engine or localhost.
- ❖ Expand Databases, right-click a database, point to Tasks, and click Import Flat File above Import Data.
- ❖ Adding data from your local machine to Jupyter Notebook
    - ✓ First, navigate to the Jupyter Notebook interface home page. You can do this by going to the URL <my-hub-url>/user/<my-username>/tree.
    - ✓ Click the "Upload" button to open the file chooser window.
    - ✓ Choose the file you wish to upload. You may select multiple files if you wish.
    - ✓ Click "Upload" for each file that you wish to upload.
    - ✓ Wait for the progress bar to finish for each file. These files will now be on your JupyterHub, your home user's home directory.
- ❖ Establish connection between Jupyter Notebook and on-premise database using pyodbc module
- ❖ Execute the queries stated in the Problem Statement.
- ❖ Create Visualization using matplotlib and seaborn libraries

PROBLEM STATEMENT

To create an end-to-end flow for executing the queries related to Airlines Analysis using Jupyter Notebook and generating visualizations.

Files attached here with all schema and data creation script:

Airlines Data.7z

Tasks to be performed:

- Run the script “Script.sql” attached in the zipped folder “Airlines Data.7z” to create following tables:
    - ➢ Hub_Airport,
    - ➢ Hub_Flight1,
    - ➢ Link_Flight_Airport1
    - ➢ Sat_Airport and
    - ➢ Sat_Flight1.
- Load that dataset into the SQL DB.
- Clean the dataset, remove any non-required columns, and import the data into the tabular model project.
- Load that data into the Jupyter Notebook.
- Execute the following queries:
    1. Total No. of different flights running.
    2. Flights going to particular country
    3. Flights going to state
    4. Flights going to a city.
    5. Perform calculation to Identify Regional/International Airport.
    6. Flights going to every State.
    7. Find Different Airlines Available
    8. Which Airline has the maximum running flights
    9. No of Airports in the state AZ, DE, and NY.
- Get possible Visualization of data.

PREREQUISITES

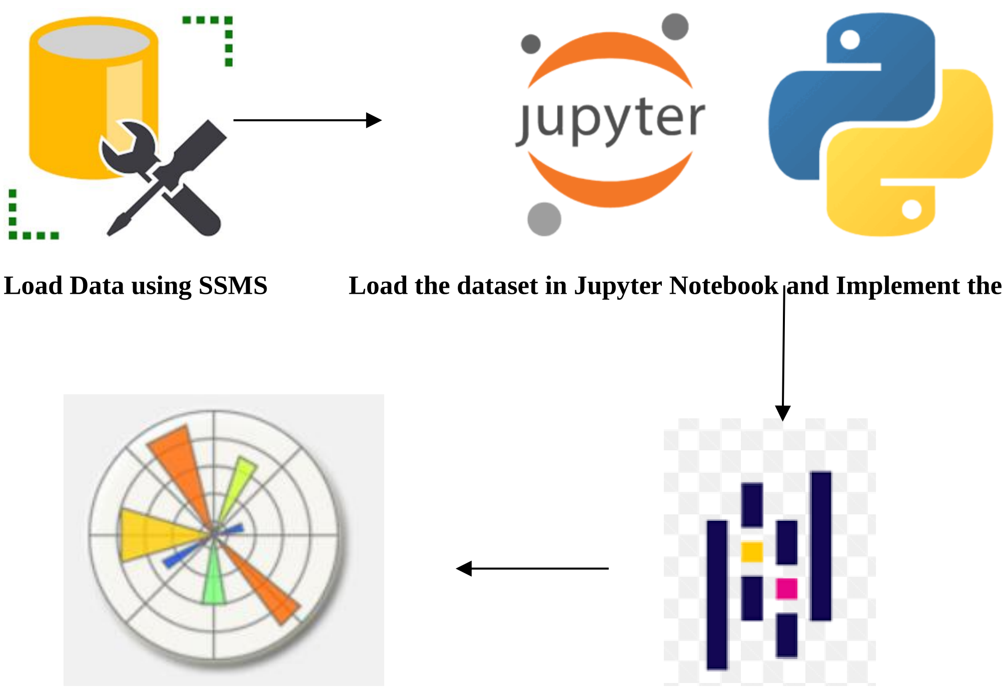
1. SQL Server 2017 (or higher) with SQL Server Analysis Services Implemented
2. Miniconda with Jupyter Notebook
3. Pyodbc module to create connection with database
4. Pandas, matplotlib and seaborn libraries to generate visualizations

PROJECT DESIGN

High Level Design:

$$\begin{tikzcd}
\boxed{\text{Data Loading}} \arrow[r] & \boxed{\text{Data Cleaning}} \arrow[r] & \boxed{\text{Deployment}} \arrow[d] \\
 & \boxed{\text{Data Visualisation}} & \boxed{\text{Implementing Queries}} \arrow[l]
\end{tikzcd}$$

Low Level Design:



Load Data using SSMS

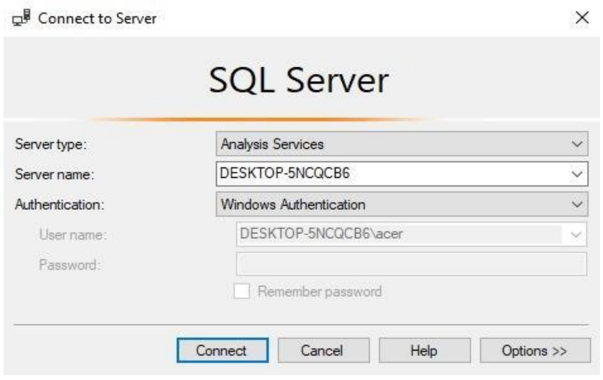
Load the dataset in Jupyter Notebook and Implement the queries

Visualization using matplotlib and seaborn

Create dataframes using pandas library

1.Run Script file (Data Source) in SSMS to create database.

2. Perform data cleaning .


Connect to Server

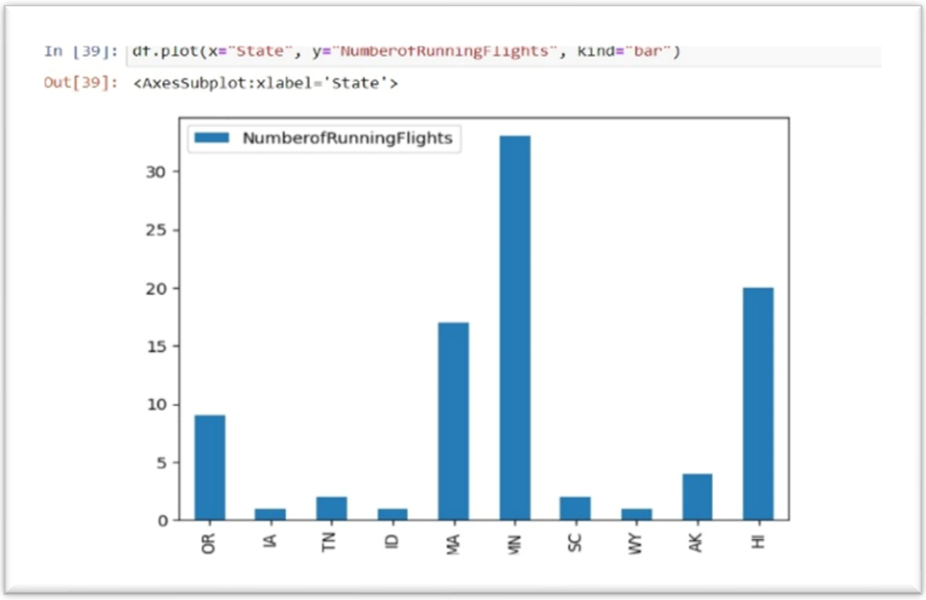
SQL Server

Server type:	Analysis Services
Server name:	DESKTOP-5NCQCB6
Authentication:	Windows Authentication
User name:	DESKTOP-5NCQCB6\acer
Password:	

Remember password

3. Cleaned data loaded to Jupiter notebook

4. Query implementation is done to produce the expected results and visualisation .

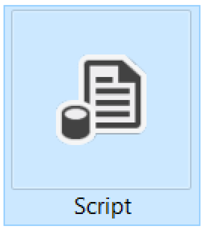


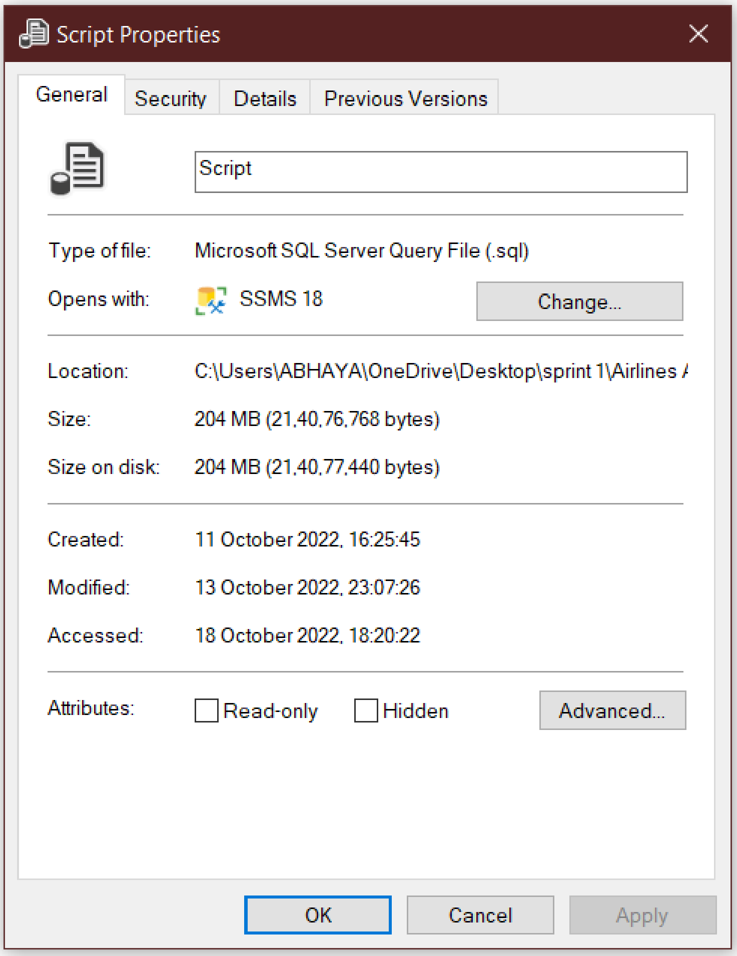
```
In [39]: df.plot(x="State", y="NumberofRunningFlights", kind="bar")
Out[39]: <AxesSubplot:xlabel='State'>
```

IMPLEMENTATION STEPS

LOADING DATA INTO SQL SERVER:

1. Run the script as given in AIRLINE SERVICES Sprint Case Study.





Script

Type of file: Microsoft SQL Server Query File (.sql)

Opens with: SSMS 18

Location: C:\Users\ABHAYA\OneDrive\Desktop\sprint 1\Airlines

Size: 204 MB (21,40,76,768 bytes)

Size on disk: 204 MB (21,40,77,440 bytes)

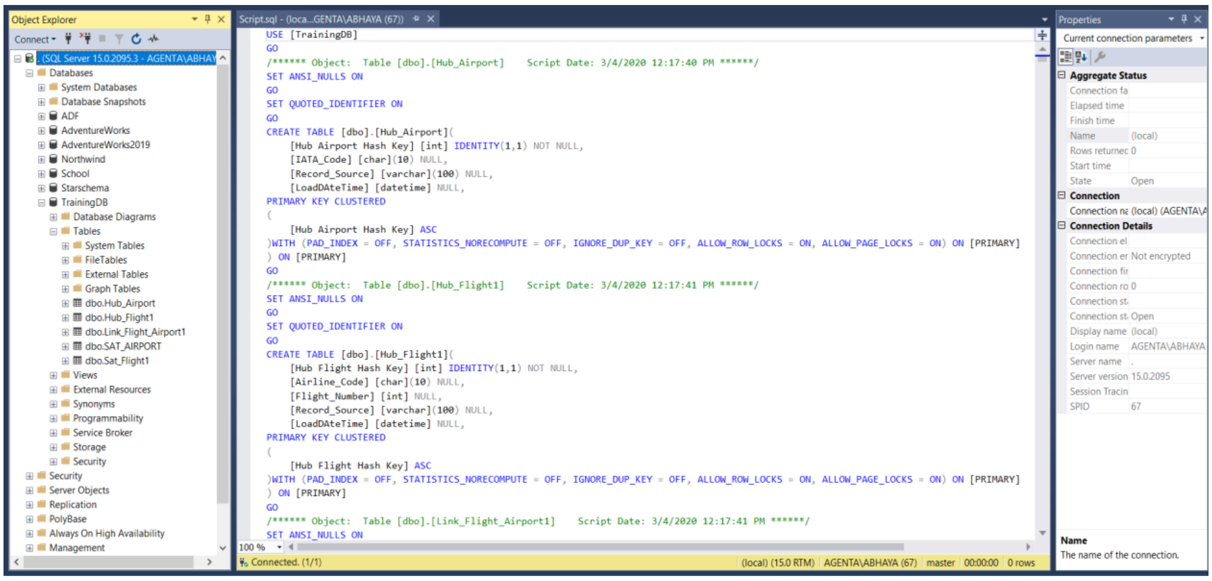
Created: 11 October 2022, 16:25:45

Modified: 13 October 2022, 23:07:26

Accessed: 18 October 2022, 18:20:22

Attributes: ☐ Read-only ☐ Hidden

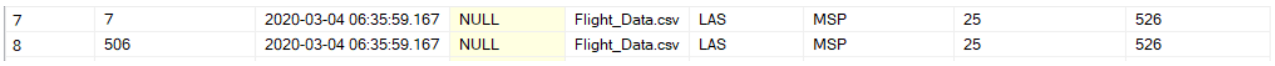
2. Create Database and run the code to apply tables and values into the database.



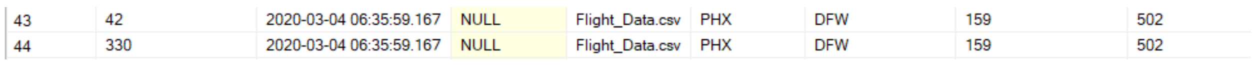
```sql
USE [TrainingDB]
GO
/****** Object:  Table [dbo].[Hub_Airport]    Script Date: 3/4/2020 12:17:40 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Hub_Airport](
    [Hub Airport Hash Key] [int] IDENTITY(1,1) NOT NULL,
    [IATA_Code] [char](10) NULL,
    [Record_Source] [varchar](100) NULL,
    [LoadDAteTime] [datetime] NULL,
PRIMARY KEY CLUSTERED
(
    [Hub Airport Hash Key] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Hub_Flight1]    Script Date: 3/4/2020 12:17:41 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Hub_Flight1](
    [Hub Flight Hash Key] [int] IDENTITY(1,1) NOT NULL,
    [Airline_Code] [char](10) NULL,
    [Flight_Number] [int] NULL,
    [Record_Source] [varchar](100) NULL,
    [LoadDAteTime] [datetime] NULL,
PRIMARY KEY CLUSTERED
(
    [Hub Flight Hash Key] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Link_Flight_Airport1]    Script Date: 3/4/2020 12:17:41 PM ******/
SET ANSI_NULLS ON
```

```
Connected. (1/1)    (local) (15.0 RTM)    AGENTA\ABHAYA (67)    master    00:00:00    0 rows
```

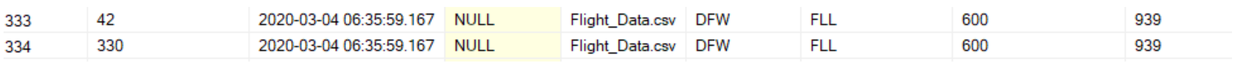
3. Clean the datasets for finding better performance and less errors. (Example: Removing duplicate and unwanted columns)



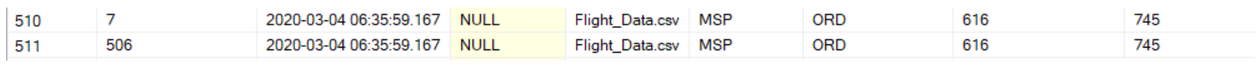
7	7	2020-03-04 06:35:59.167	NULL	Flight_Data.csv	LAS	MSP	25	526
8	506	2020-03-04 06:35:59.167	NULL	Flight_Data.csv	LAS	MSP	25	526



43	42	2020-03-04 06:35:59.167	NULL	Flight_Data.csv	PHX	DFW	159	502
44	330	2020-03-04 06:35:59.167	NULL	Flight_Data.csv	PHX	DFW	159	502



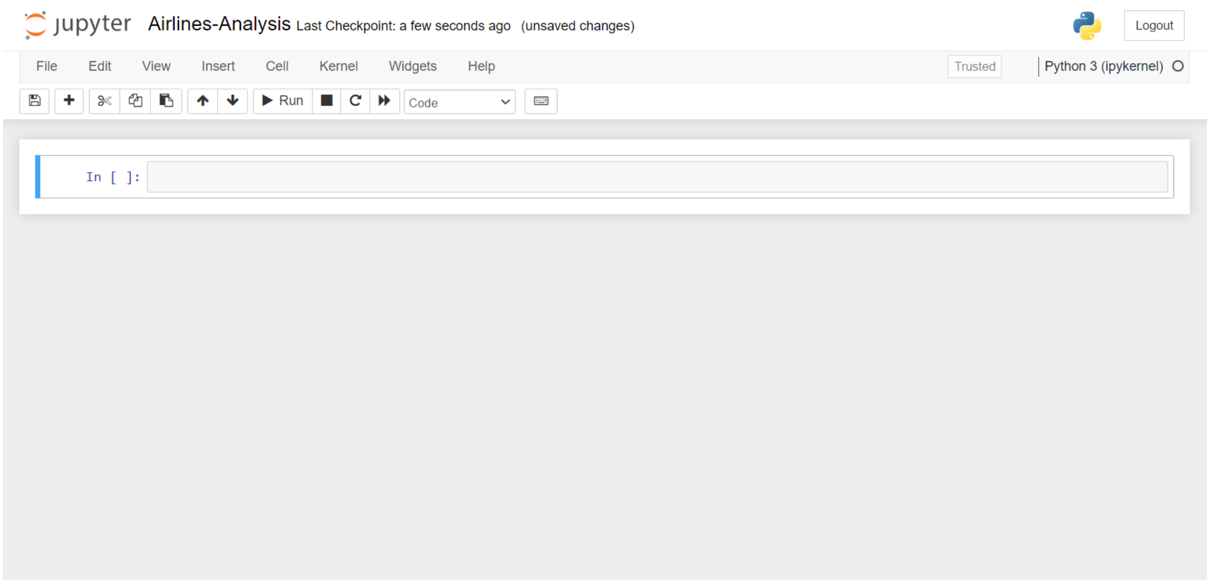
333	42	2020-03-04 06:35:59.167	NULL	Flight_Data.csv	DFW	FLL	600	939
334	330	2020-03-04 06:35:59.167	NULL	Flight_Data.csv	DFW	FLL	600	939



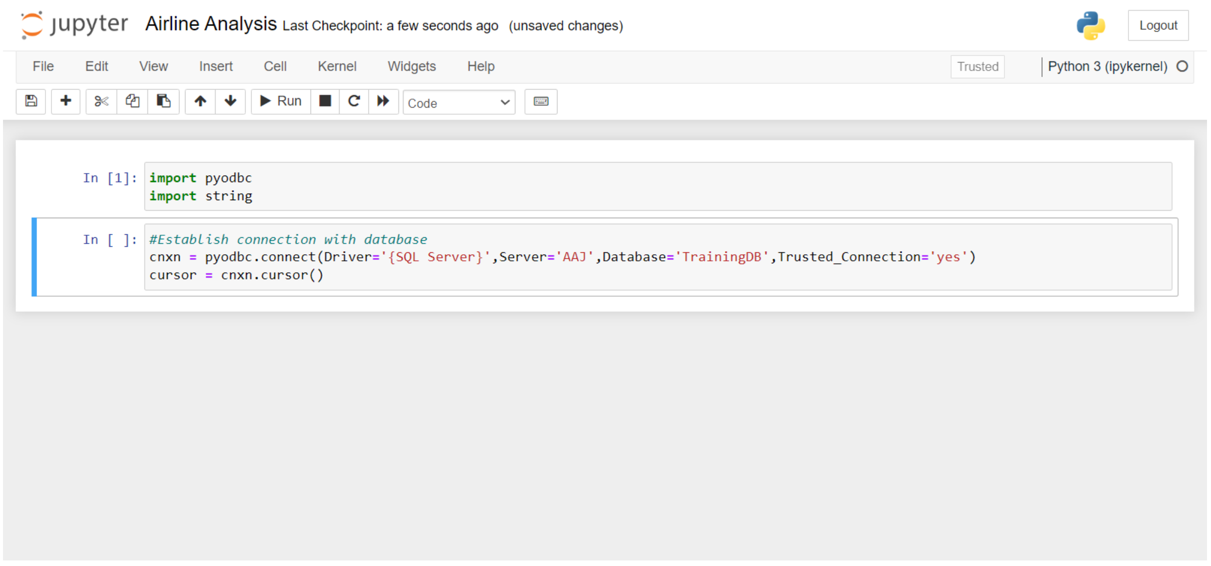
510	7	2020-03-04 06:35:59.167	NULL	Flight_Data.csv	MSP	ORD	616	745
511	506	2020-03-04 06:35:59.167	NULL	Flight_Data.csv	MSP	ORD	616	745

LOADING DATABASE INTO JUPYTER NOTEBOOK:

1. Open and run Jupyter Notebook


jupyter
Airlines-Analysis Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [ ]:

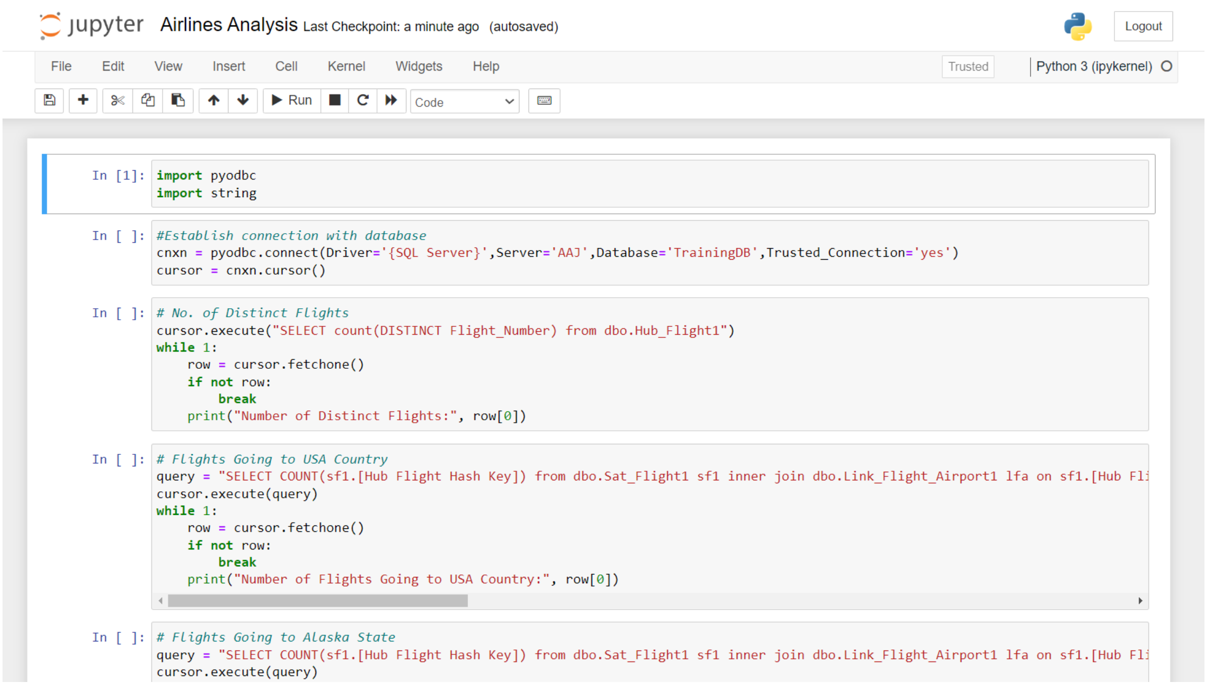
2. Establish the connection with SQL Server.


jupyter
Airline Analysis Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: import pyodbc
        import string
```

```
In [ ]: #Establish connection with database
        cnxn = pyodbc.connect(Driver='{SQL Server}',Server='AAJ',Database='TrainingDB',Trusted_Connection='yes')
        cursor = cnxn.cursor()
```

3. After successful connection, run the queries.

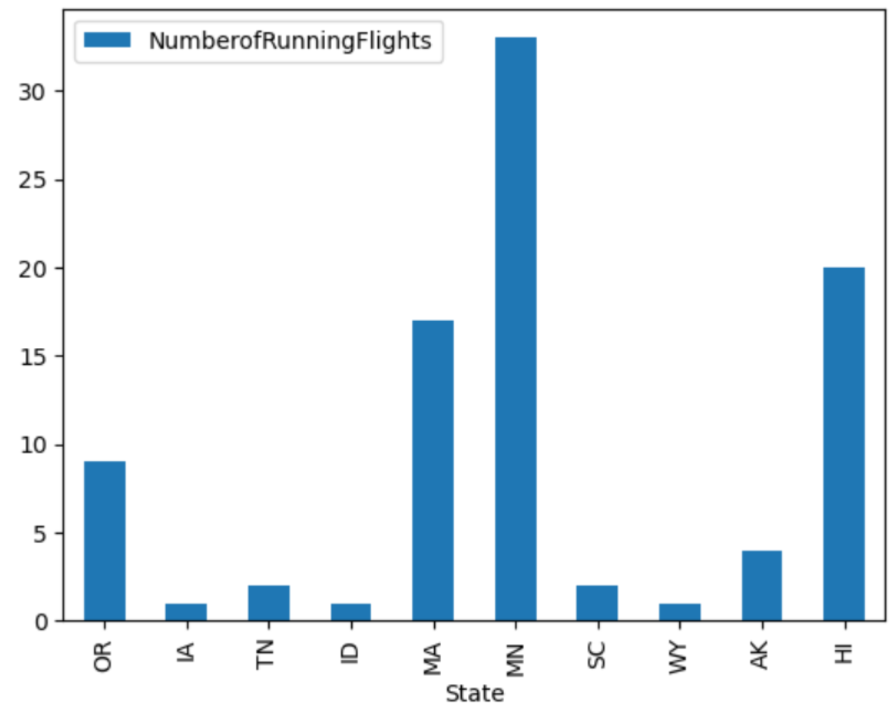

jupyter Airlines Analysis Last Checkpoint: a minute ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: import pyodbc
        import string

In [ ]: #Establish connection with database
        cnxn = pyodbc.connect(Driver='{SQL Server}',Server='AAJ',Database='TrainingDB',Trusted_Connection='yes')
        cursor = cnxn.cursor()

In [ ]: # No. of Distinct Flights
        cursor.execute("SELECT count(DISTINCT Flight_Number) from dbo.Hub_Flight1")
        while 1:
            row = cursor.fetchone()
            if not row:
                break
            print("Number of Distinct Flights:", row[0])

In [ ]: # Flights Going to USA Country
        query = "SELECT COUNT(sf1.[Hub Flight Hash Key]) from dbo.Sat_Flight1 sf1 inner join dbo.Link_Flight_Airport1 lfa on sf1.[Hub Fli
        cursor.execute(query)
        while 1:
            row = cursor.fetchone()
            if not row:
                break
            print("Number of Flights Going to USA Country:", row[0])

In [ ]: # Flights Going to Alaska State
        query = "SELECT COUNT(sf1.[Hub Flight Hash Key]) from dbo.Sat_Flight1 sf1 inner join dbo.Link_Flight_Airport1 lfa on sf1.[Hub Fli
        cursor.execute(query)
```

4. Build some visualizations using pandas, matplotlib and seaborn libraries.

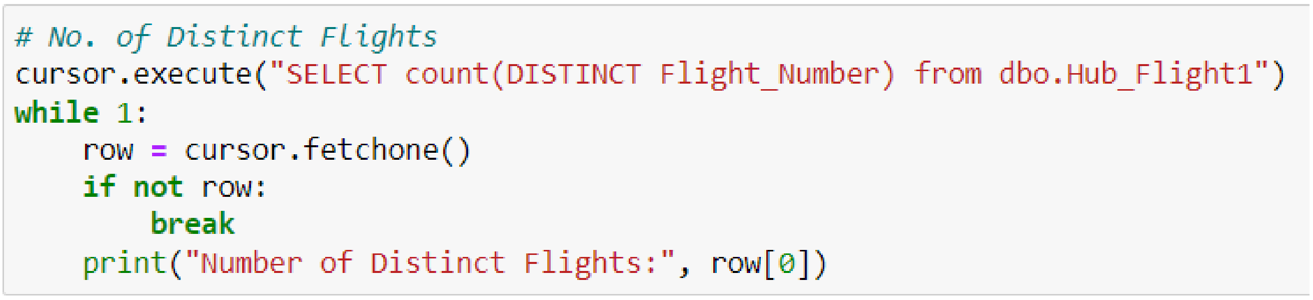
```
df.plot(x="State", y="NumberofRunningFlights", kind="bar")
```

```
<AxesSubplot:xlabel='State'>
```



QUERIES AND SCREENSHOTS

1) Total Number of Distinct Flights

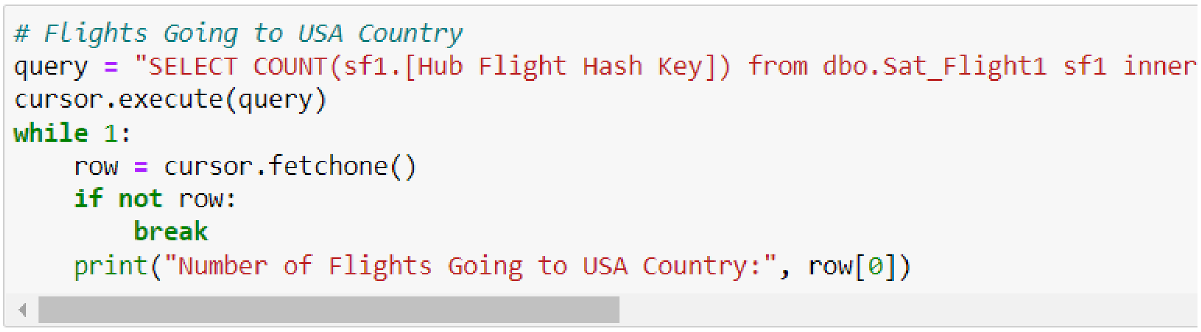


```
# No. of Distinct Flights
cursor.execute("SELECT count(DISTINCT Flight_Number) from dbo.Hub_Flight1")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("Number of Distinct Flights:", row[0])
```

```
Number of Distinct Flights: 911
```

2) Flights Going to USA Country

query = "SELECT COUNT(sf1.[Hub Flight Hash Key]) from dbo.Sat_Flight1 sf1 inner join dbo.Link_Flight_Airport1 lfa on sf1.[Hub Flight Hash Key] = lfa.[Hub Flight Hash Key] inner join dbo.SAT_AIRPORT sa on sa.[Hub Airport Hash Key] = lfa.[Hub Airport Hash Key] inner join dbo.Hub_Airport ha on ha.[Hub Airport Hash Key] = sa.[Hub Airport Hash Key] where sa.Country = 'USA'and ha.IATA_Code = sf1.Destination_Airport"

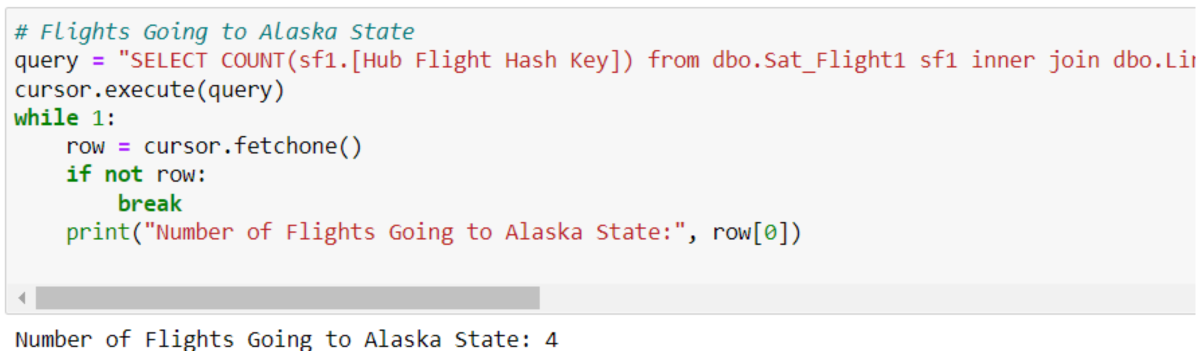


```
# Flights Going to USA Country
query = "SELECT COUNT(sf1.[Hub Flight Hash Key]) from dbo.Sat_Flight1 sf1 inner
cursor.execute(query)
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("Number of Flights Going to USA Country:", row[0])
```

```
Number of Flights Going to USA Country: 1006
```

3) Flights Going to Alaska State

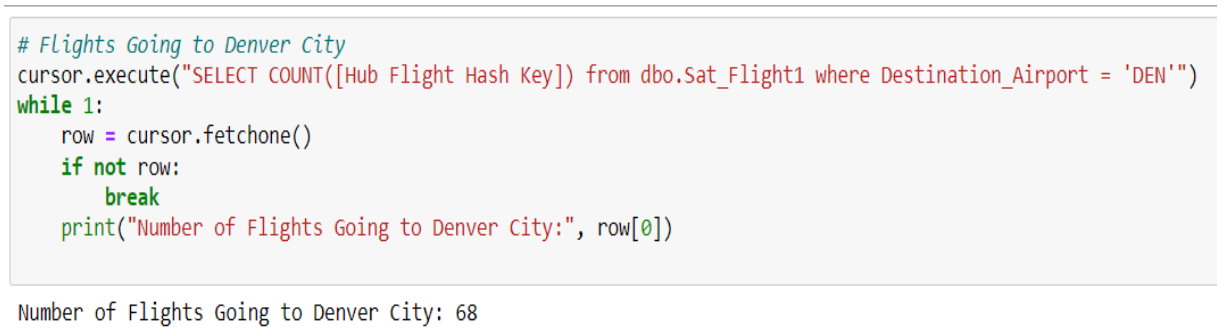
query = "SELECT COUNT(sf1.[Hub Flight Hash Key]) from dbo.Sat_Flight1 sf1 inner join dbo.Link_Flight_Airport1 lfa on sf1.[Hub Flight Hash Key] = lfa.[Hub Flight Hash Key] inner join dbo.SAT_AIRPORT sa on sa.[Hub Airport Hash Key] = lfa.[Hub Airport Hash Key] inner join dbo.Hub_Airport ha on ha.[Hub Airport Hash Key] = sa.[Hub Airport Hash Key] where sa.State = 'AK'and ha.IATA_Code = sf1.Destination_Airport"



```
# Flights Going to Alaska State
query = "SELECT COUNT(sf1.[Hub Flight Hash Key]) from dbo.Sat_Flight1 sf1 inner join dbo.Li
cursor.execute(query)
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("Number of Flights Going to Alaska State:", row[0])
```

```
Number of Flights Going to Alaska State: 4
```

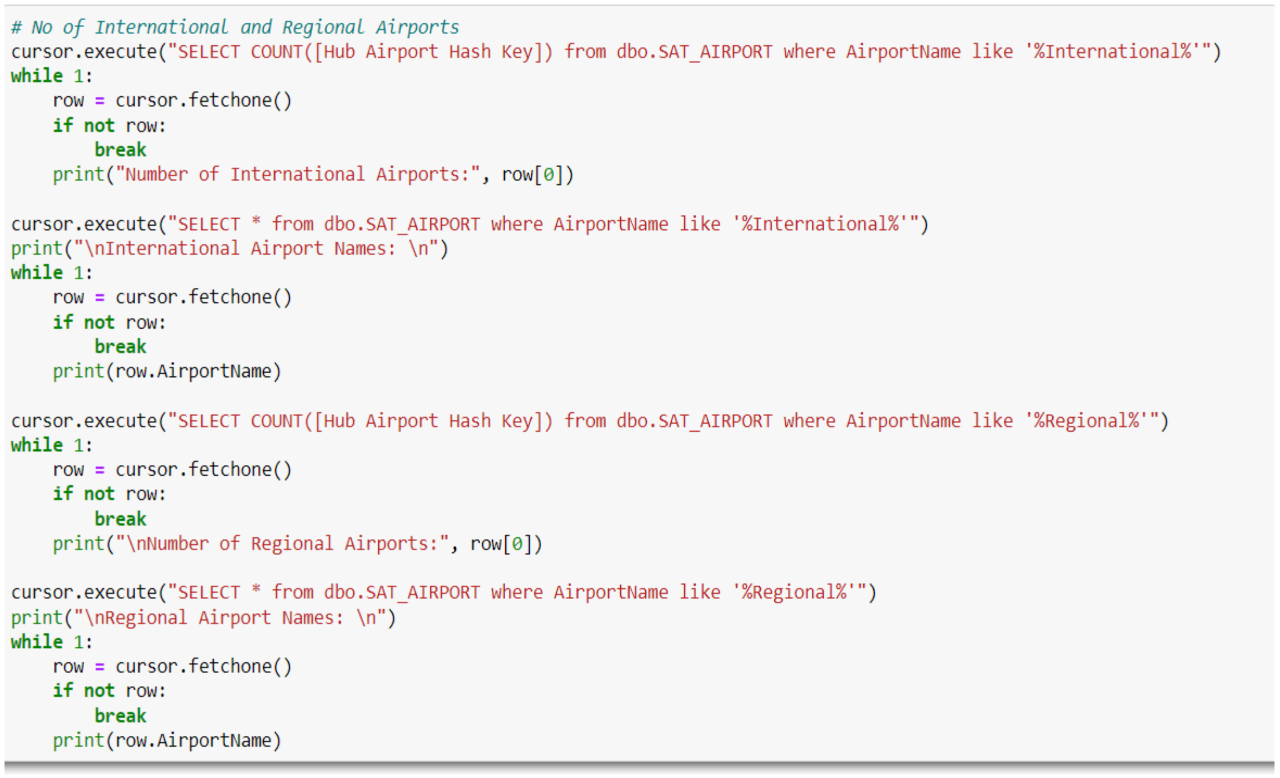
4) Flights Going to Denver City



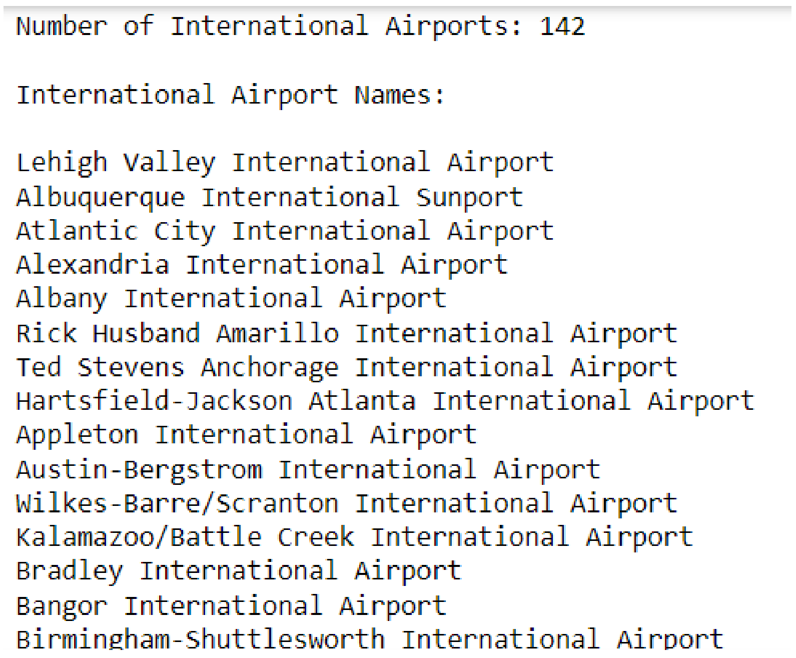
```
# Flights Going to Denver City
cursor.execute("SELECT COUNT([Hub Flight Hash Key]) from dbo.Sat_Flight1 where Destination_Airport = 'DEN'")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("Number of Flights Going to Denver City:", row[0])
```

```
Number of Flights Going to Denver City: 68
```

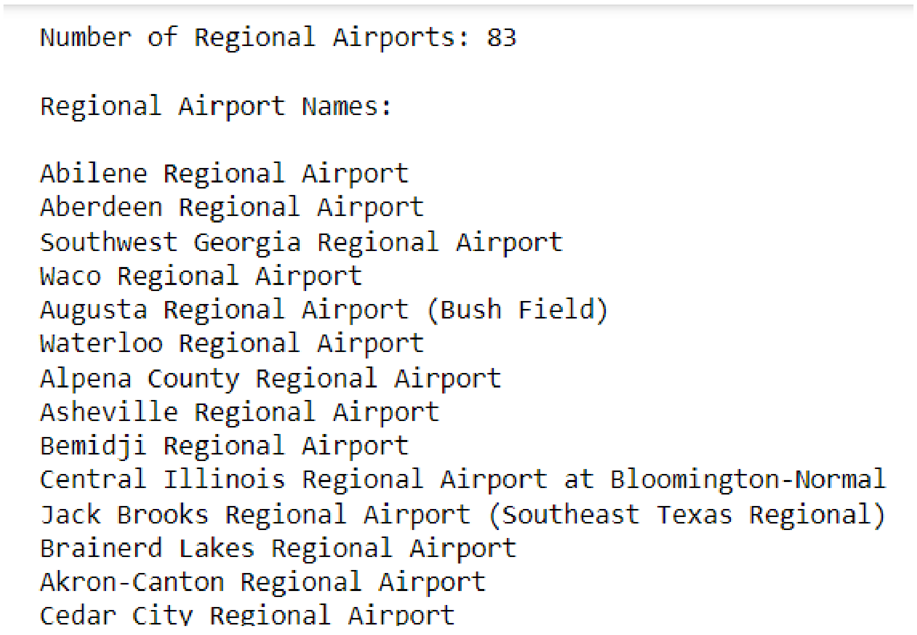
5) International and Regional Airports



```
# No of International and Regional Airports
cursor.execute("SELECT COUNT([Hub Airport Hash Key]) from dbo.SAT_AIRPORT where AirportName like '%International%'")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("Number of International Airports:", row[0])

cursor.execute("SELECT * from dbo.SAT_AIRPORT where AirportName like '%International%'")
print("\nInternational Airport Names: \n")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print(row.AirportName)

cursor.execute("SELECT COUNT([Hub Airport Hash Key]) from dbo.SAT_AIRPORT where AirportName like '%Regional%'")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("\nNumber of Regional Airports:", row[0])

cursor.execute("SELECT * from dbo.SAT_AIRPORT where AirportName like '%Regional%'")
print("\nRegional Airport Names: \n")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print(row.AirportName)
```



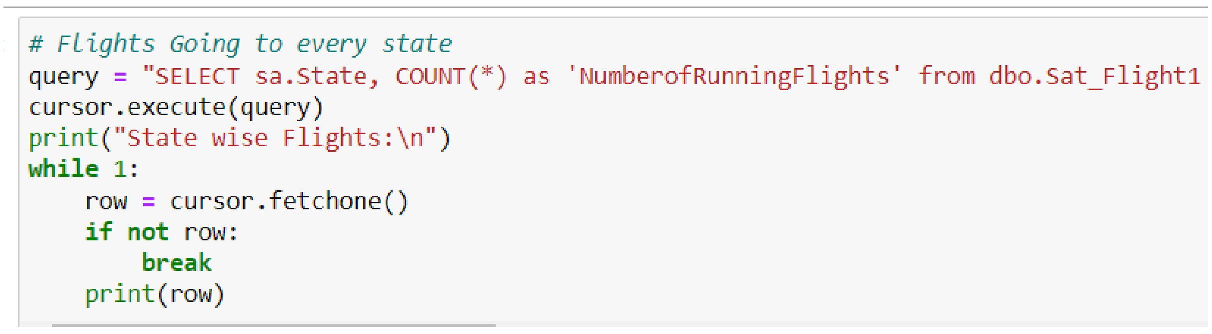
```
Number of International Airports: 142

International Airport Names:

Lehigh Valley International Airport
Albuquerque International Sunport
Atlantic City International Airport
Alexandria International Airport
Albany International Airport
Rick Husband Amarillo International Airport
Ted Stevens Anchorage International Airport
Hartsfield-Jackson Atlanta International Airport
Appleton International Airport
Austin-Bergstrom International Airport
Wilkes-Barre/Scranton International Airport
Kalamazoo/Battle Creek International Airport
Bradley International Airport
Bangor International Airport
Birmingham-Shuttlesworth International Airport
```



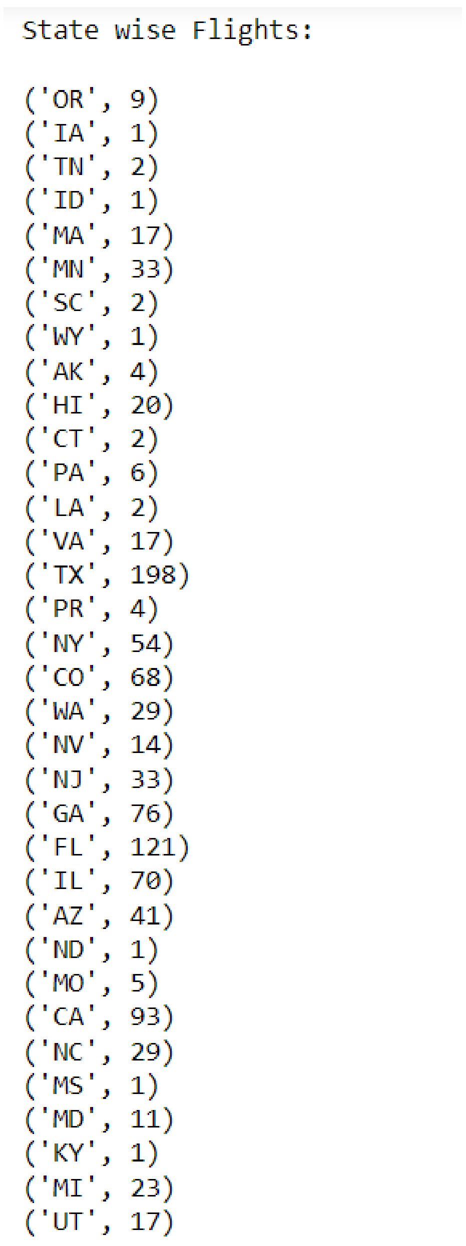
```
Number of Regional Airports: 83

Regional Airport Names:

Abilene Regional Airport
Aberdeen Regional Airport
Southwest Georgia Regional Airport
Waco Regional Airport
Augusta Regional Airport (Bush Field)
Waterloo Regional Airport
Alpena County Regional Airport
Asheville Regional Airport
Bemidji Regional Airport
Central Illinois Regional Airport at Bloomington-Normal
Jack Brooks Regional Airport (Southeast Texas Regional)
Brainerd Lakes Regional Airport
Akron-Canton Regional Airport
Cedar City Regional Airport
```

6) Flights Going to Every State

query = "SELECT sa.State, COUNT(*) as 'NumberofRunningFlights' from dbo.Sat_Flight1 sf1 inner join dbo.Link_Flight_Airport1 lfa on sf1.[Hub Flight Hash Key] = lfa.[Hub Flight Hash Key] inner join dbo.SAT_AIRPORT sa on sa.[Hub Airport Hash Key] = lfa.[Hub Airport Hash Key] inner join dbo.Hub_Airport ha on ha.[Hub Airport Hash Key] = sa.[Hub Airport Hash Key] inner join dbo.Hub_Flight1 hf1 on hf1.[Hub Flight Hash Key] = sf1.[Hub Flight Hash Key] where ha.IATA_Code = sf1.Destination_Airport Group by sa.State"

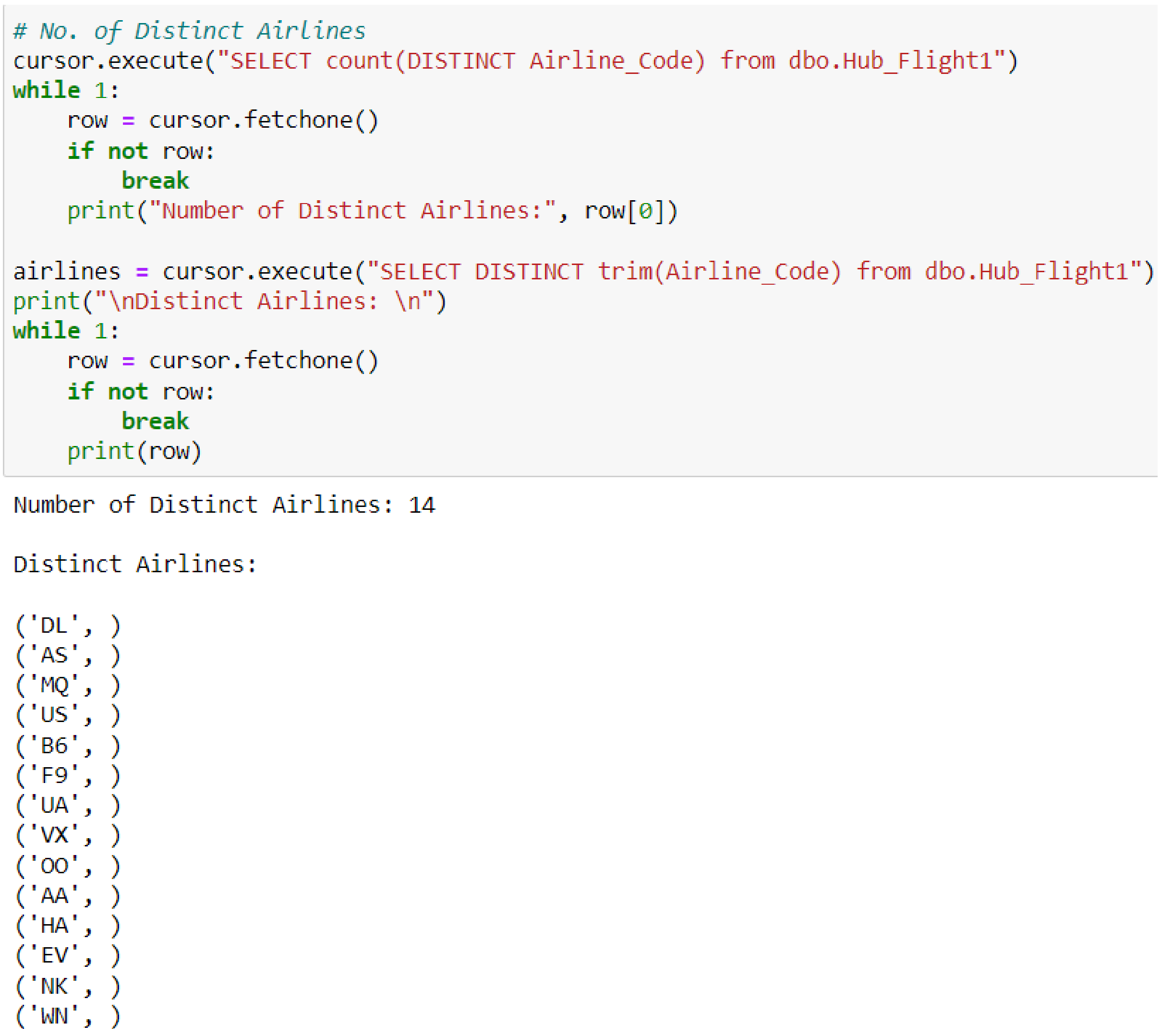


```python
# Flights Going to every state
query = "SELECT sa.State, COUNT(*) as 'NumberofRunningFlights' from dbo.Sat_Flight1
cursor.execute(query)
print("State wise Flights:\n")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print(row)
```



```
State wise Flights:

('OR', 9)
('IA', 1)
('TN', 2)
('ID', 1)
('MA', 17)
('MN', 33)
('SC', 2)
('WY', 1)
('AK', 4)
('HI', 20)
('CT', 2)
('PA', 6)
('LA', 2)
('VA', 17)
('TX', 198)
('PR', 4)
('NY', 54)
('CO', 68)
('WA', 29)
('NV', 14)
('NJ', 33)
('GA', 76)
('FL', 121)
('IL', 70)
('AZ', 41)
('ND', 1)
('MO', 5)
('CA', 93)
('NC', 29)
('MS', 1)
('MD', 11)
('KY', 1)
('MI', 23)
('UT', 17)
```

7) Distinct Airlines

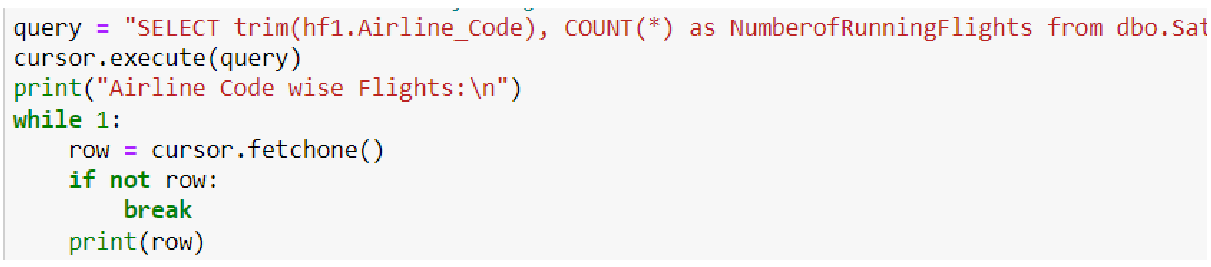


```
# No. of Distinct Airlines
cursor.execute("SELECT count(DISTINCT Airline_Code) from dbo.Hub_Flight1")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("Number of Distinct Airlines:", row[0])

airlines = cursor.execute("SELECT DISTINCT trim(Airline_Code) from dbo.Hub_Flight1")
print("\nDistinct Airlines: \n")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print(row)
```

```
Number of Distinct Airlines: 14

Distinct Airlines:

('DL', )
('AS', )
('MQ', )
('US', )
('B6', )
('F9', )
('UA', )
('VX', )
('OO', )
('AA', )
('HA', )
('EV', )
('NK', )
('WN', )
```

8) Airline Code Wise Flights

query = "SELECT trim(hf1.Airline_Code), COUNT(*) as NumberofRunningFlights from dbo.Sat_Flight1 sf1 inner join dbo.Link_Flight_Airport1 lfa on sf1.[Hub Flight Hash Key] = lfa.[Hub Flight Hash Key] inner join dbo.SAT_AIRPORT sa on sa.[Hub Airport Hash Key] = lfa.[Hub Airport Hash Key] inner join dbo.Hub_Airport ha on ha.[Hub Airport Hash Key] = sa.[Hub Airport Hash Key] inner join dbo.Hub_Flight1 hf1 on hf1.[Hub Flight Hash Key] = sf1.[Hub Flight Hash Key] where ha.IATA_Code = sf1.Destination_Airport Group by hf1.Airline_Code order by NumberofRunningFlights desc"



```
query = "SELECT trim(hf1.Airline_Code), COUNT(*) as NumberofRunningFlights from dbo.Sat
cursor.execute(query)
print("Airline Code wise Flights:\n")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print(row)
```

```
Airline Code wise Flights:

('AA', 135)
('OO', 121)
('DL', 118)
('UA', 103)
('B6', 91)
('EV', 80)
('WN', 72)
('MQ', 68)
('AS', 60)
('US', 59)
('NK', 44)
('F9', 34)
('HA', 17)
('VX', 4)
```

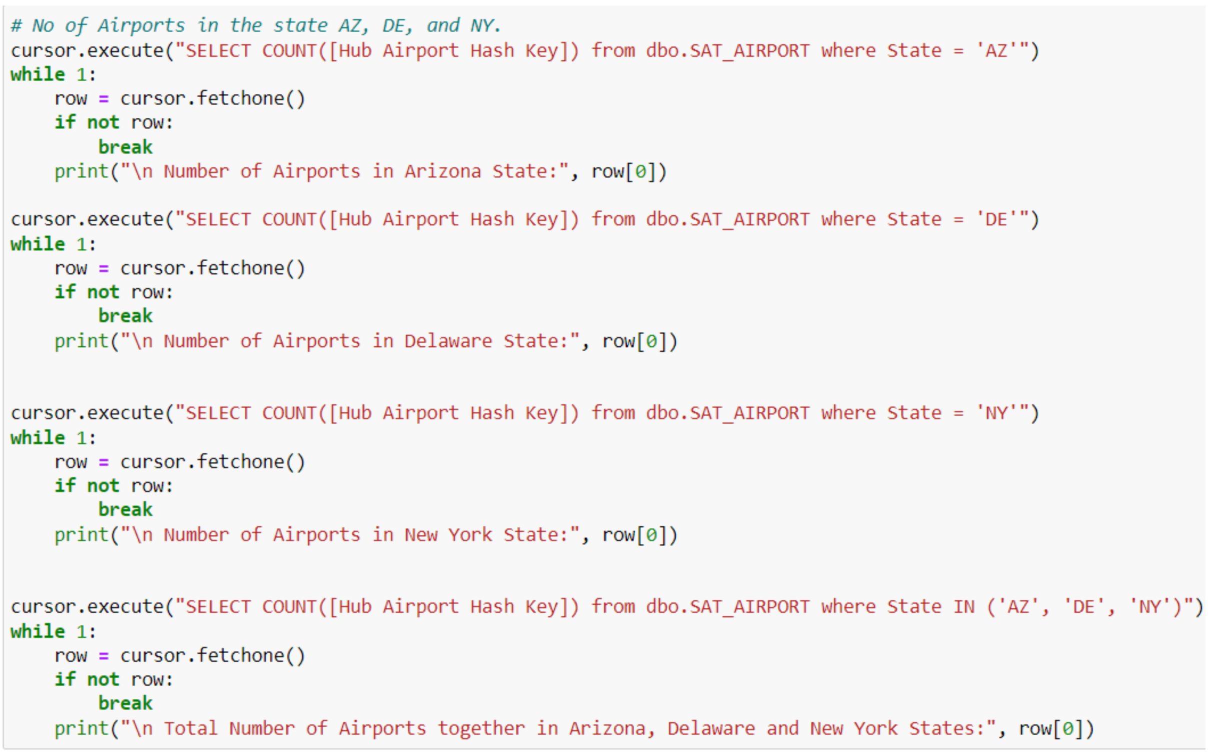
9) Airline with Maximum Flights

query = "SELECT top 1 trim(hf1.Airline_Code), COUNT(*) as NumberofRunningFlights from dbo.Sat_Flight1 sf1 inner join dbo.Link_Flight_Airport1 lfa on sf1.[Hub Flight Hash Key] = lfa.[Hub Flight Hash Key] inner join dbo.SAT_AIRPORT sa on sa.[Hub Airport Hash Key] = lfa.[Hub Airport Hash Key] inner join dbo.Hub_Airport ha on ha.[Hub Airport Hash Key] = sa.[Hub Airport Hash Key] inner join dbo.Hub_Flight1 hf1 on hf1.[Hub Flight Hash Key] = sf1.[Hub Flight Hash Key] where ha.IATA_Code = sf1.Destination_Airport Group by hf1.Airline_Code order by NumberofRunningFlights desc"

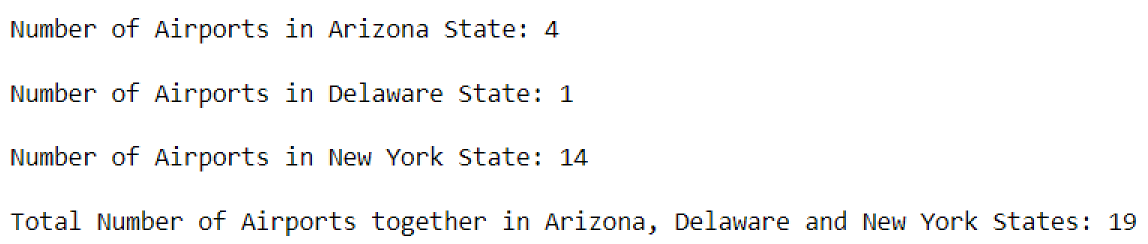
```
# Airlines with Maximum number of Flights
query = "SELECT top 1 trim(hf1.Airline_Code), COUNT(*) as NumberofRunningFlights fro
cursor.execute(query)
print("\nAirline Code with Maximum Number of Flights:\n")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print(row)
```

```
Airline Code with Maximum Number of Flights:

('AA', 135)
```

10) Number of Airports in Arizona, Delaware and New York States



```python
# No of Airports in the state AZ, DE, and NY.
cursor.execute("SELECT COUNT([Hub Airport Hash Key]) from dbo.SAT_AIRPORT where State = 'AZ'")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("\n Number of Airports in Arizona State:", row[0])

cursor.execute("SELECT COUNT([Hub Airport Hash Key]) from dbo.SAT_AIRPORT where State = 'DE'")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("\n Number of Airports in Delaware State:", row[0])


cursor.execute("SELECT COUNT([Hub Airport Hash Key]) from dbo.SAT_AIRPORT where State = 'NY'")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("\n Number of Airports in New York State:", row[0])


cursor.execute("SELECT COUNT([Hub Airport Hash Key]) from dbo.SAT_AIRPORT where State IN ('AZ', 'DE', 'NY')")
while 1:
    row = cursor.fetchone()
    if not row:
        break
    print("\n Total Number of Airports together in Arizona, Delaware and New York States:", row[0])
```



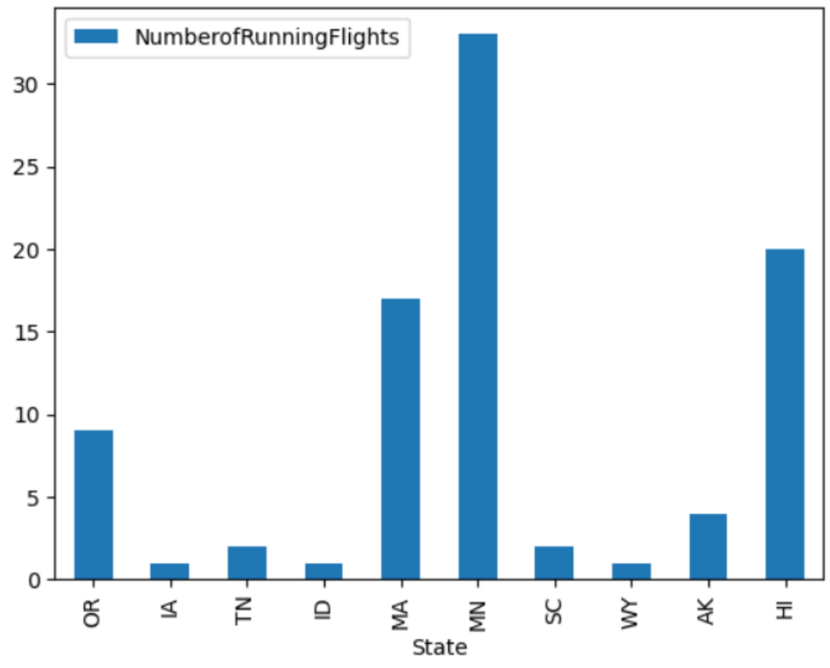
```
 Number of Airports in Arizona State: 4

 Number of Airports in Delaware State: 1

 Number of Airports in New York State: 14

 Total Number of Airports together in Arizona, Delaware and New York States: 19
```

VISUALIZATION OF DATA

1) State wise Number of Flights

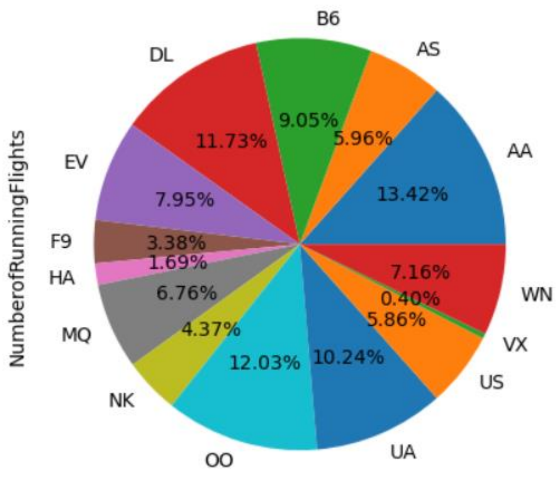
```
df.plot(x="State", y="NumberofRunningFlights", kind="bar")
```

```
<AxesSubplot:xlabel='State'>
```



2) Airline wise Number of Flights

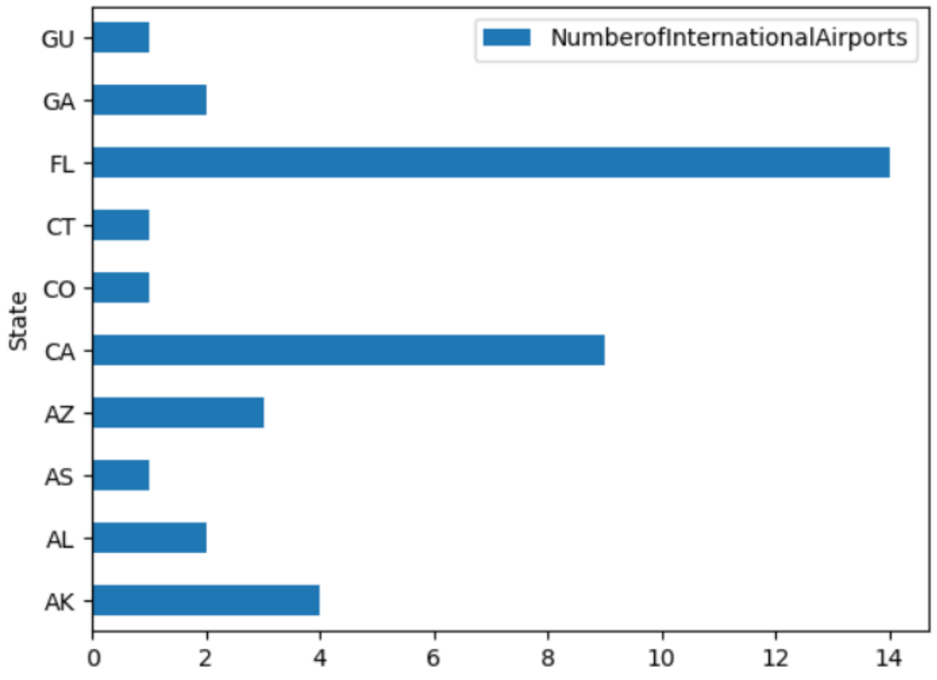
```
df1.groupby(['Airline_Code']).sum().plot(kind='pie',y='NumberofRunningFlights'
```



3) State wise Number of International Airports

```
df2.plot(x="State", y="NumberofInternationalAirports", kind="barh")
```

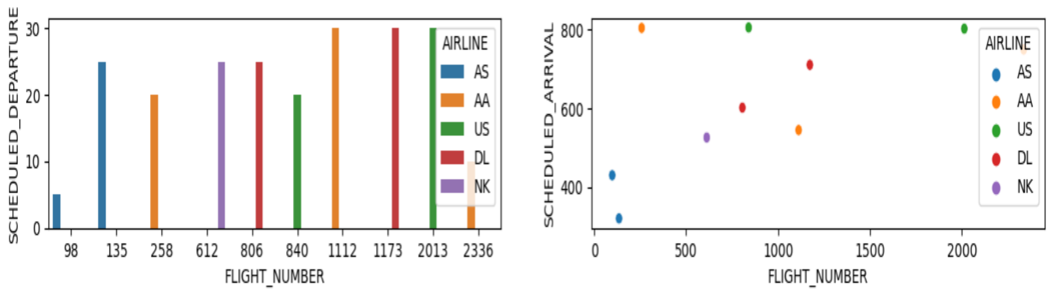
```
<AxesSubplot:ylabel='State'>
```



4) Scheduled Arrival and Departure for Flight Numbers with Airline Code

```
In [41]: fig, axs = plt.subplots(ncols=2, figsize=(15,2))
         sns.barplot(x="FLIGHT_NUMBER", y="SCHEDULED_DEPARTURE", hue="AIRLINE", data=data2, ax=axs[0])
         sns.scatterplot(x="FLIGHT_NUMBER", y="SCHEDULED_ARRIVAL", hue="AIRLINE", data=data2, ax=axs[1])
```

```
Out[41]: <AxesSubplot:xlabel='FLIGHT_NUMBER', ylabel='SCHEDULED_ARRIVAL'>
```



SUMMARY

The learning outcomes from this case study are:

1. Successfully created connection between Jupyter notebook and database using pyodbc module.
2. Implemented the queries as per the case study document.
3. Created dataframes for the obtained results using pandas library,
4. Created possible visualizations of data using matplotlib and seaborn libraries.

APPENDIX

Pyodbc Documentation:

https://learn.microsoft.com/en-us/sql/connect/python/pyodbc/step-3-proof-of-concept-connecting-to-sql-using-pyodbc?source=recommendations&view=sql-server-ver16

Matplotlib Documentation:

https://matplotlib.org/stable/plot_types/index.html

Seaborn Documentation:

https://seaborn.pydata.org/tutorial/function_overview.html

Github Link:

https://github.com/aj1497/Airlines_Analysis