DAMG 7290 Data Warehousing & Bus Intel SEC 01
Spring 2022

Team-4

Project Portfolio

Aviation Data Warehousing

Members

Siddhesh Aher

Bhavya Batra

Ronen Sengupta

Aishwarya Melige Suryanarayana

Table of Contents

1.0 Introduction

Aviation data is available everywhere but tracking the airport and airline history physically is not easy. The objective of this project is to build a Data Warehouse using airports, airlines, routes, and passenger datasets taken from various sources. The Data transformations will be done in the staging area and visualizations will be created using Tableau/Power BI to get some more general insights.

The seven datasets considered for this project are taken from various sources. These datasets provide information about airports, airlines operated in different countries along with their respective IATA, ICAO codes, and routes between two airports. It also gives information on the number of passengers by different airlines.

The Aviation Data Warehouse will not only give historical insights but also provides an important means to effectively support airport decision-making. It will allow users to access critical data from several sources in a single place. It can be used to understand traffic patterns of airlines for different years, track aircraft history, observe airport congestion, etc.

1.0.1 Business Scenario

To gather appropriate data, we start the process by identifying the input sources. We consider the flight dataset obtained from Kaggle and Flight.org as the main source of input. We then extract the data present in the CSV format and store it in a staging area. This will help in resolving inaccuracies and errors in the data and verifying the input based on the required conditions.

To showcase the trends and extract significant information from the Data warehouse we outline a dimension model containing measures describing the Number of Passengers and Airline Routes through attributes like Passenger_Count, Source Airport, Destination Airport.

In order to analyze the changes in the dataset, we incorporate Slowly Changing Dimensions on a few attributes present in the dimension tables to keep track of the variation and history of the data. This will help us analyze the data more precisely.

Here are a few scenarios we considered to understand the slowly changing dimensions.

- Keeping track of the changes in Flight and the Number of Airlines will help us understand the availability of the Airline and flight modification.
- Keeping track of the Passengers around the airport around the year can help us with planning proper facilities for the passenger.
- Keeping a track of the number of Airports in the Country can help us understand the Passenger_country over a particular period.

Through the OLAP cube analyzing this data with historical values can help us provide valuable information in order to improve the performance of the Flights. To display the visualizations, we create dashboards containing all the crucial information via Tableau/Power BI.

2.0 Data Description

The datasets are considered from Flights.org and Kaggle. The project has seven datasets. The description of the datasets are as follows:

AIRPORT DATASET: This Dataset gives information about airports spanning the globe. This dataset has a total of 14 attributes and 7699 datapoints.

Attributes	Data Types	Description
Airport_ID	Int	Unique identifier for the Airport. Primary Key(PK)
Name	Char	Name of airport
City	Char	Main city served by airport. May be spelled differently from Name
Country	Char	Country or territory where airport is located
IATA	Char	3-letter IATA code. Null if not assigned/unknown
ICAO	Char	4-letter ICAO code. Null if not assigned
Latitude	Float	Decimal degrees, usually to six significant digits. Negative is South, positive is North
Longitude	Float	Decimal degrees, usually to six significant digits. Negative is West, positive is East
Altitude	Int	In feet
Time zone	Int	Hours offset from UTC. Fractional hours are expressed as decimals, e.g., India is 5.5

DST	Char	Daylight savings time. One of E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None) or U (Unknown)

AIRLINE DATASET: It gives information about the airlines operating around the globe along with their unique IATA/ICAO codes.

It has a total of 8 attributes and 6163 records.

Attributes	Data Types	Description
Airline_ID	Int	Unique identifier for this airline. Primary Key(PK)
Name	Char	Name of the airline
IATA	Char	2-letter IATA code, if available
ICAO	Char	3-letter ICAO code, if available
Call sign	Char	Airline callsign
Country	Char	Country or territory where airport is located
Active	Boolean	“Y" if the airline is or has until recently been operational, "N" if it is defunct

ROUTE DATASET: It gives the information on the routes between two airports and the number of stops for that route taken. It has 10 attributes and 67664 records.

Attributes	Data Types	Description
Route_ID	Int	Primary Key(PK)
Airline	Char	2-letter (IATA) or 3-letter (ICAO) code of the airline
Airline_ID	Int	Foreign Key (FK)
Source Airport	Char	3-letter (IATA) or 4-letter (ICAO) code of the source airport
Source Airport_ID	Int	Unique identifier for source airport

Destination Airport	Char	3-letter (IATA) or 4-letter (ICAO) code of the destination airport
Destination Airport_ID	Int	Unique identifier for destination airport
Codeshare	Boolean	"Y" if this flight is a codeshare (that is, not operated by Airline, but another carrier), empty otherwise
Stops	Int	Number of stops on this flight ("0" for direct)
Equipment	Char	3-letter codes for plane type(s) generally used on this flight, separated by spaces

COUNTRIES DATASET: It has all the countries listed along with their ISO codes. It has 3 attributes and 262 records.

Attributes	Data Types	Description
Country_ID	Int	Primary Key (PK)
Name	Char	Full name of the country or territory
iso_code	Char	Unique two-letter ISO 3166-1 code for the country or territory

AIR_PASSENGER_STATISTICS DATASET: It gives the number of passengers for different domestic and international airlines. It has a total of 11 attributes and 22870 records.

Attributes	Data Types	Description
Passenger_Statistics_ID	Int	Primary Key(PK)
Activity Period	Int	The year and month at which passenger activity took place

Operating Airline	Char	Airline name for the operator of aircraft with passenger activity
Operating Airline IATA Code	Char	The International Air Transport Association (IATA) two-letter designation for the Operating Airline.
Published Airline	Char	Airline name that issues the ticket and books revenue for passenger activity
Published Airline IATA Code	Char	The International Air Transport Association (IATA) two-letter designation for the Published Airline
GEO Summary	Char	Designates whether the passenger activity in relation to SFO arrived from or departed to a
		location within the United States (“domestic”), or outside the United States (“international”)without stops
GEO Region	Char	Provides a more detailed breakdown of the GEO Summary field to designate the region in the world where activity in relation to SFO arrived from or departed to without stops
Activity Type Code	Char	A description of the physical action a passenger took in relation to a flight, which includes boarding a flight (“enplanements”), getting off a flight (“deplanements”) and transiting to another location (“in-transit”)

Price Category Code	Char	A categorization of whether a Published Airline is a low-cost carrier or not a low-cost carrier
Terminal	Char	The airport terminal designations at SFO where passenger activity took place
Boarding Area	Char	The airport boarding area designations at SFO where passenger activity took place
Passenger Count	Int	The number of monthly passengers associated with the above attribute fields.

INTERNATIONAL _REPORT_DEPARTURE: This dataset provides the information on a specific airline for a pair of airports along with the number of flights from those airports. Three main columns record that record the number of flights is: Scheduled, Charter, and Total. It has a total of 16 attributes and approx. 68000 records.

Attributes	Data Types	Description
Departure_ID	Int	Primary Key (PK)
date_dte	Date	Departure Date
Year	Date	Departure Year
Month	Date	Departure Month
Dept_usg_apt_id	Int	US Gateway airport ID
Dept_usg_apt	Char	US gateway IATA code
Dept_usg_wac	Int	US gateway world area code
Dept_fg_apt_id	Int	Foreign gateway airport ID
Dept_fg_apt	Char	Foreign gateway IATA code
Dept_fg_wac	int	Foreign gateway world area code
Dept_airlineid	Int	Airline ID

Carrier	Char	Airline IATA code
Type	Char	Type of the metric
Dept_scheduled	Int	Metric flown by scheduled service operations
Dept_Charter	Int	Metric flown by charter operations
Dept_Total	Int	Total Metric flown by scheduled service and charter operations

INTERNATIONAL_REPORT_PASSENGER: This dataset gives the information on the total number of passengers for each month and year between a pair of airports, as serviced by a particular airline.

Attributes	Data Types	Description
Passenger_ID	Int	Primary Key (PK)
date_dte	Date	Date
Year	Date	Year
Month	Date	Month
usg_apt_id	Int	US Gateway airport ID
usg_apt	Char	US gateway IATA code
usg_wac	Int	US gateway world area code
fg_apt_id	Int	Foreign gateway airport ID
fg_apt	Char	Foreign gateway IATA code
fg_wac	Int	Foreign gateway world area code
airlineid	Int	Airline ID
Carrier	Char	Airline IATA code
Type	Char	Type of the metric
Scheduled	Int	Passengers flown by scheduled service operations

Charter	Int	Passenger flown by charter operations
Total	Int	Total Passengers flown by scheduled service and charter operations

3.0 Selection of Facts and Dimension Tables

Facts tables:
The Facts tables considered are as follows:

Airport (FactAirport)

Airline (FactAirlines)

Dimensions tables:

Airport	Airlines
Countries (Dimcountries)	Routes (DimRoutes)
Passenger (Dimpassenger)	AirCarrier (DimAirCarrier)
Airport (DimAirport)	PassengerStats (DimPassengerStats)
International_ Departures(DimDepartures)	
International_Passengers(DimDPassengers)	

Types of measure:

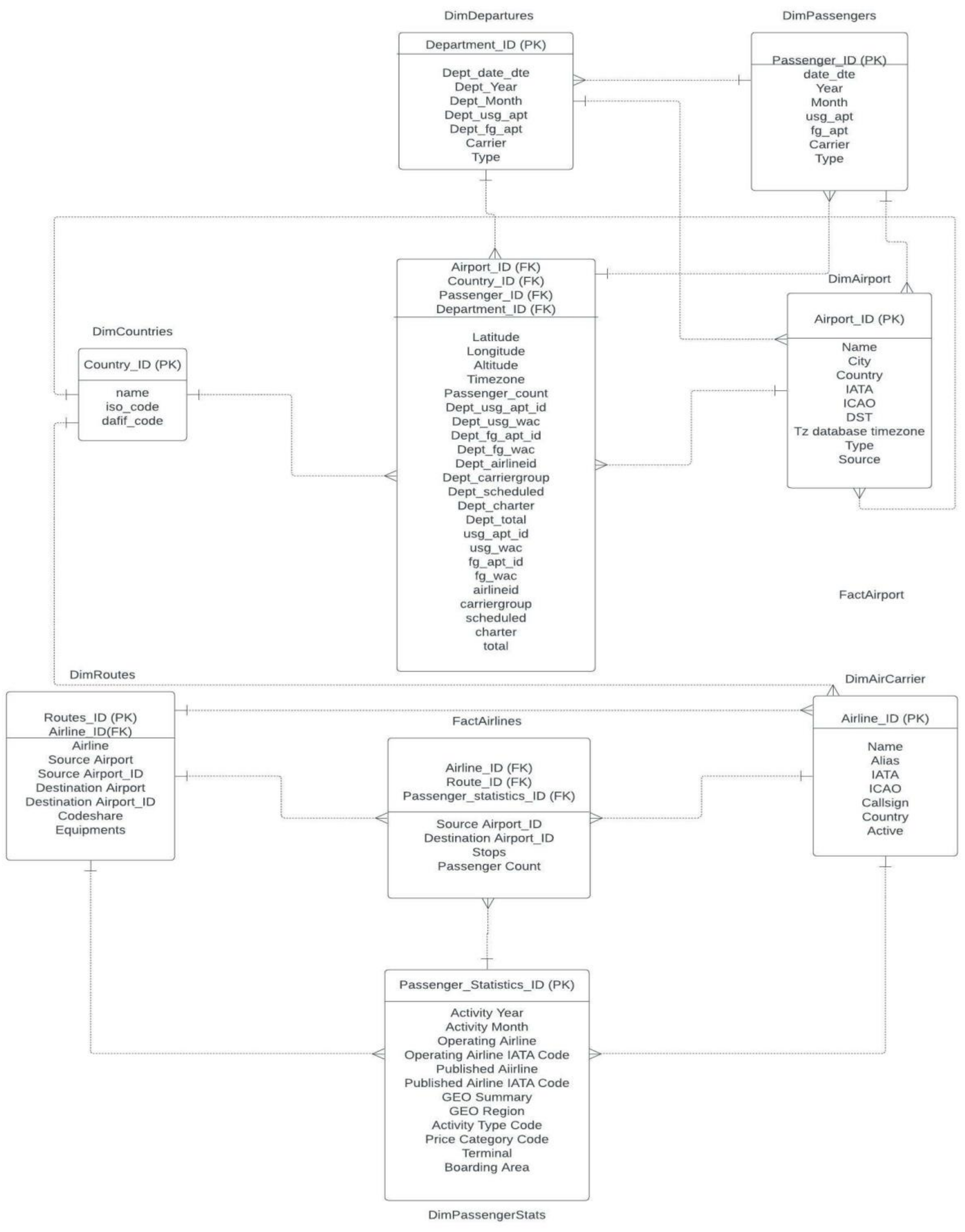
Airport

Fully Additive: Passenger Count

Airlines

Fully Additive: Passenger Count

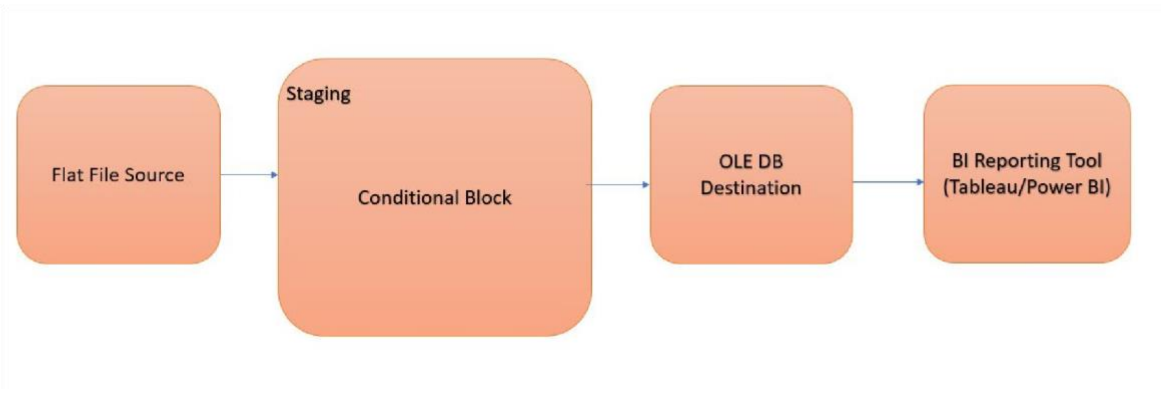
4.0 Data Model



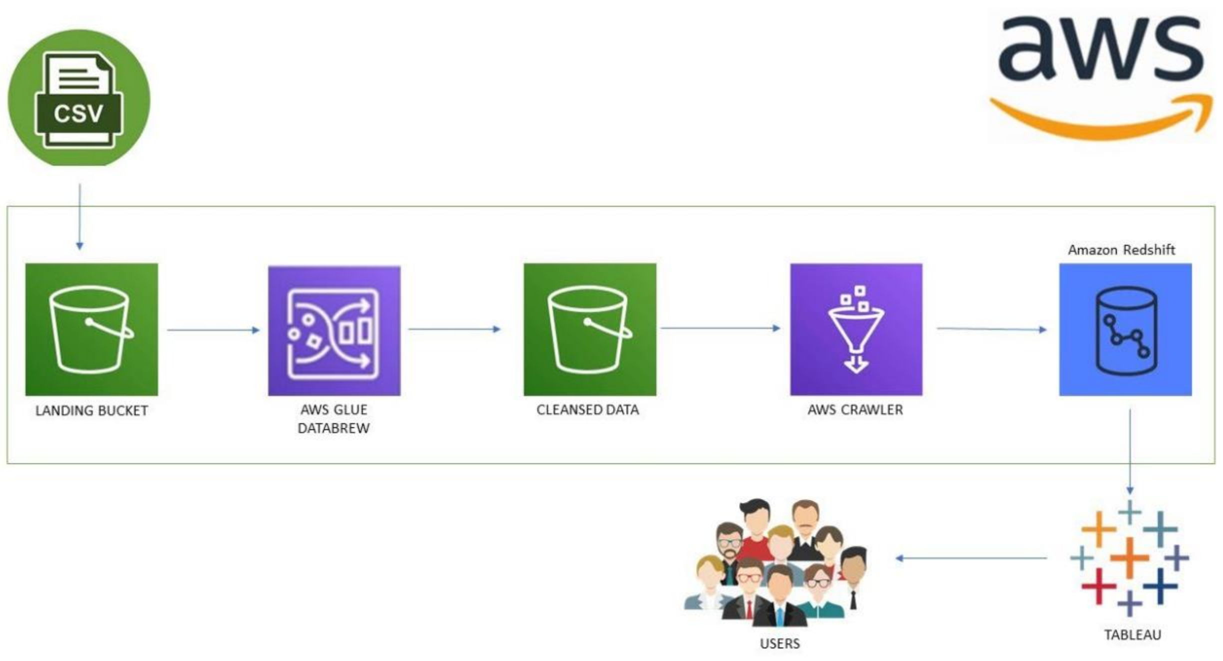
5.0 Data Flow:

We would Consider Flat File Source Connection, the source would be in CSV Format (Aviation Datasets from Kaggle and Flight.org.) In the staging area, we would have a condition block to load bulk data. After a bulk load, we will have the condition to load weekly data, till the rest of the data is transferred into our OLE Db Destination. In destination, we will create facts and dimension tables as mentioned below.

5.0.1 Basic Data Flow



5.0.2 High-level View



6.0 Loading and staging

As discussed in the previous sections, we will initially load the data from the source datasets, through the appropriate source tables.

We load the data initially into the Staging table.

We process the data by using tasks present in AWS Glue Databrew like Data Conversion to check and convert the data types appropriately, adding Slowly Changing Dimensions to add the historical values to attributes like Callsign, and PassengerCount.

With the changes reflected into the appropriate fields along with the StartDate, End Date and the Flag, we load the data into the Star/ Snowflake Schema of the destination tables.

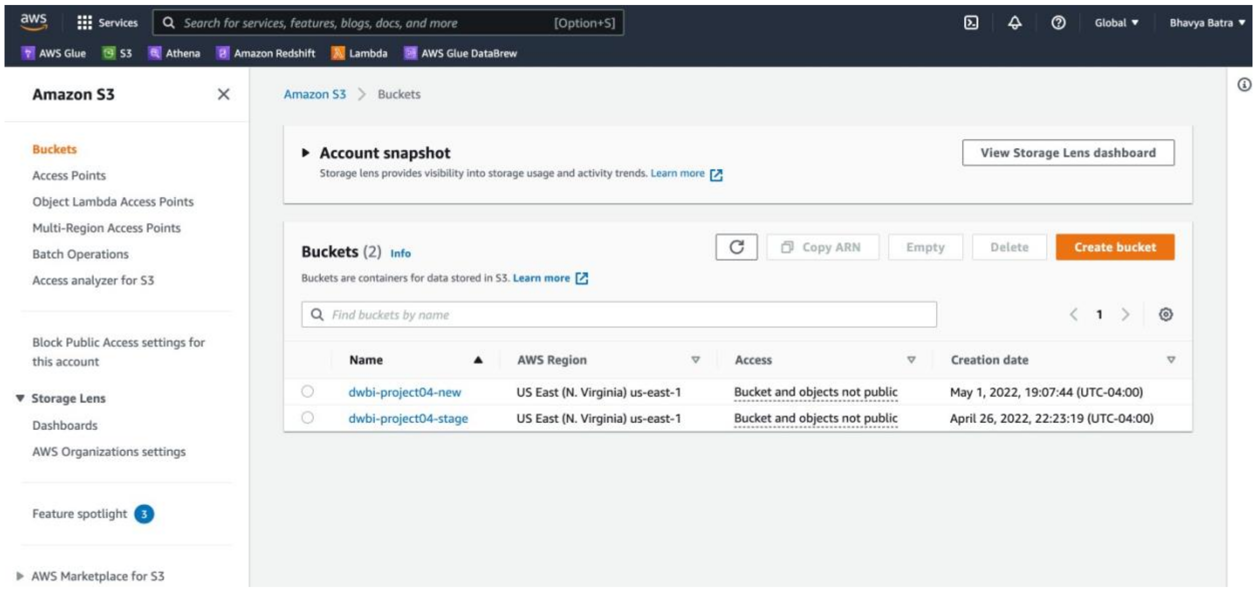
We consider look up task to probe the IATA and examine the modifications of data through Slowly Changing Dimension tasks.

S3 Bucket created for storing the CSV file: We created a two S3 buckets for storing the data.

The 1st bucket was created to store the raw data and the second bucket was created to store the cleaned the data.

1st S3 bucket: dwbi-project04-stage

2nd S3 bucket: dwbi-project04-new



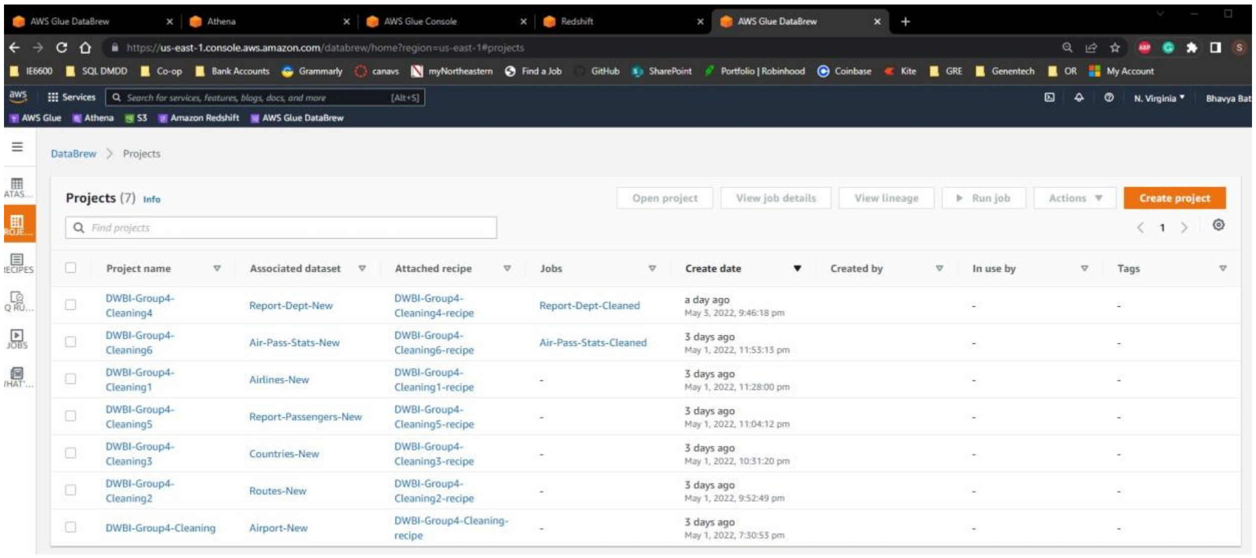
Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (2) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
dwbi-project04-new	US East (N. Virginia) us-east-1	Bucket and objects not public	May 1, 2022, 19:07:44 (UTC-04:00)
dwbi-project04-stage	US East (N. Virginia) us-east-1	Bucket and objects not public	April 26, 2022, 22:23:19 (UTC-04:00)

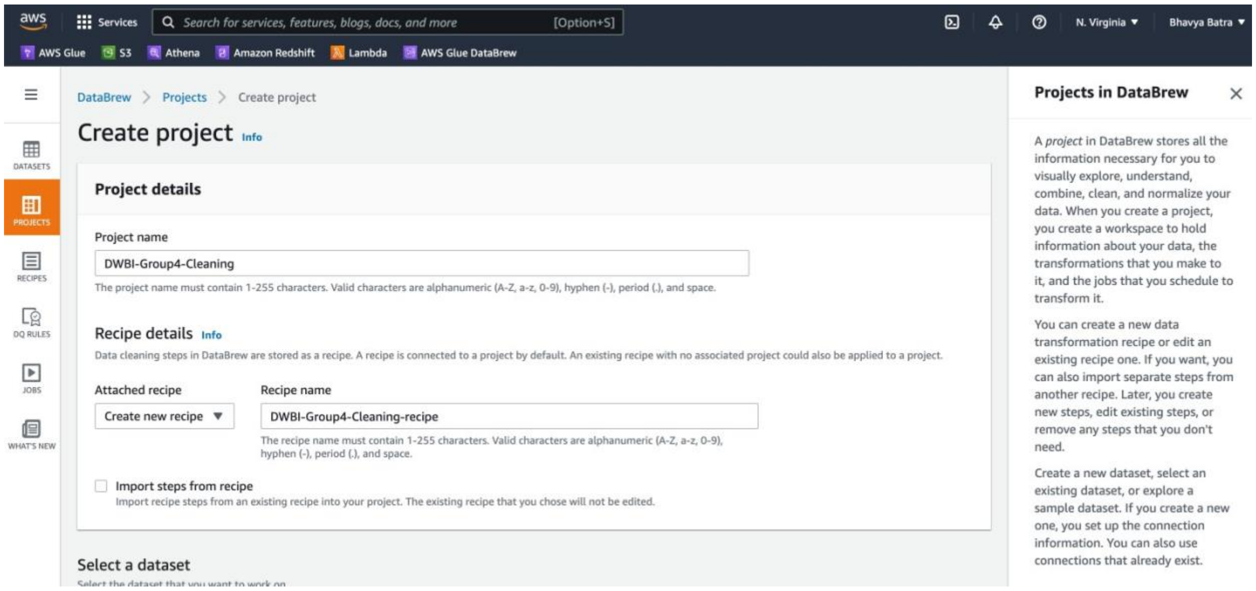


Projects (7)

Project name	Associated dataset	Attached recipe	Jobs	Create date	Created by	In use by	Tags
DWBI-Group4-Cleaning4	Report-Dept-New	DWBI-Group4-Cleaning4-recipe	Report-Dept-Cleaned	a day ago May 3, 2022, 9:46:18 pm		-	-
DWBI-Group4-Cleaning6	Air-Pass-Stats-New	DWBI-Group4-Cleaning6-recipe	Air-Pass-Stats-Cleaned	3 days ago May 1, 2022, 11:53:13 pm		-	-
DWBI-Group4-Cleaning1	Airlines-New	DWBI-Group4-Cleaning1-recipe	-	3 days ago May 1, 2022, 11:28:00 pm		-	-
DWBI-Group4-Cleaning5	Report-Passengers-New	DWBI-Group4-Cleaning5-recipe	-	3 days ago May 1, 2022, 11:04:12 pm		-	-
DWBI-Group4-Cleaning3	Countries-New	DWBI-Group4-Cleaning3-recipe	-	3 days ago May 1, 2022, 10:31:20 pm		-	-
DWBI-Group4-Cleaning2	Routes-New	DWBI-Group4-Cleaning2-recipe	-	3 days ago May 1, 2022, 9:52:49 pm		-	-
DWBI-Group4-Cleaning	Airport-New	DWBI-Group4-Cleaning-recipe	-	3 days ago May 1, 2022, 7:30:53 pm		-	-

After the raw data sets were stored into the S3 bucket, we cleaned the datasets to remove all the unnecessary data using AWS Glue Databrew.

In the above fig you can see the projects created for all the data sets. For each Dataset we were required to create separate projects and then we ran recopied on each of the projects.



Create project

Project details

Project name: DWBI-Group4-Cleaning

The project name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.

Recipe details

Data cleaning steps in DataBrew are stored as a recipe. A recipe is connected to a project by default. An existing recipe with no associated project could also be applied to a project.

Attached recipe: Create new recipe

Recipe name: DWBI-Group4-Cleaning-recipe

The recipe name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.

Import steps from recipe

Import recipe steps from an existing recipe into your project. The existing recipe that you chose will not be edited.

Select a dataset

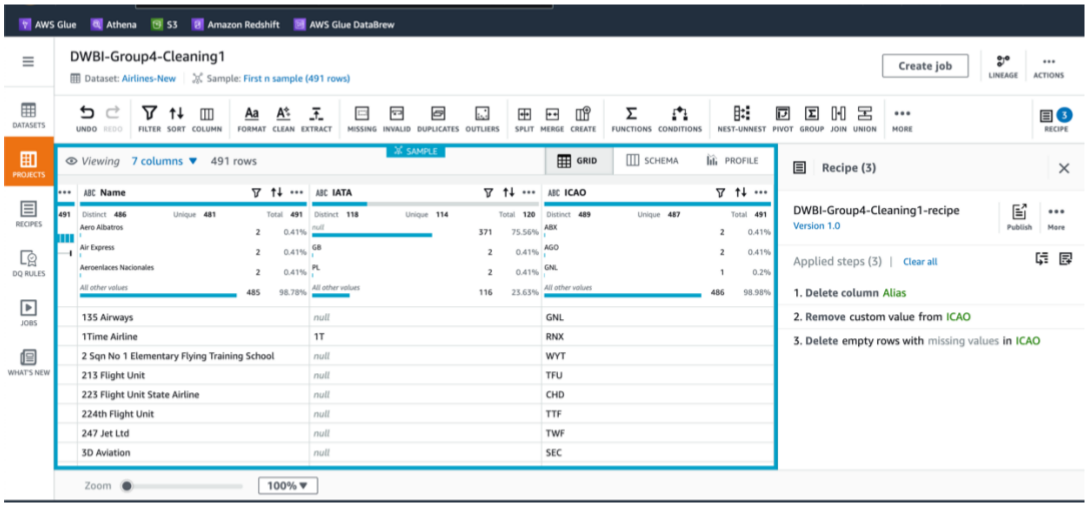
Projects in DataBrew

A *project* in DataBrew stores all the information necessary for you to visually explore, understand, combine, clean, and normalize your data. When you create a project, you create a workspace to hold information about your data, the transformations that you make to it, and the jobs that you schedule to transform it.

You can create a new data transformation recipe or edit an existing recipe one. If you want, you can also import separate steps from another recipe. Later, you create new steps, edit existing steps, or remove any steps that you don't need.

Create a new dataset, select an existing dataset, or explore a sample dataset. If you create a new one, you set up the connection information. You can also use connections that already exist.

After that we removed unnecessary columns for all the datasets and recipes were created for each datasets.


DWBI-Group4-Cleaning1

Dataset: Airlines-New | Sample: First n sample (491 rows)

Viewing 7 columns ▼ 491 rows

Name	IATA	ICAO
135 Airways	null	GNL
1Time Airline	1T	RNX
2 Sqn No 1 Elementary Flying Training School	null	WYT
213 Flight Unit	null	TFU
223 Flight Unit State Airline	null	CHD
224th Flight Unit	null	TTF
247 Jet Ltd	null	TWF
3D Aviation	null	SEC

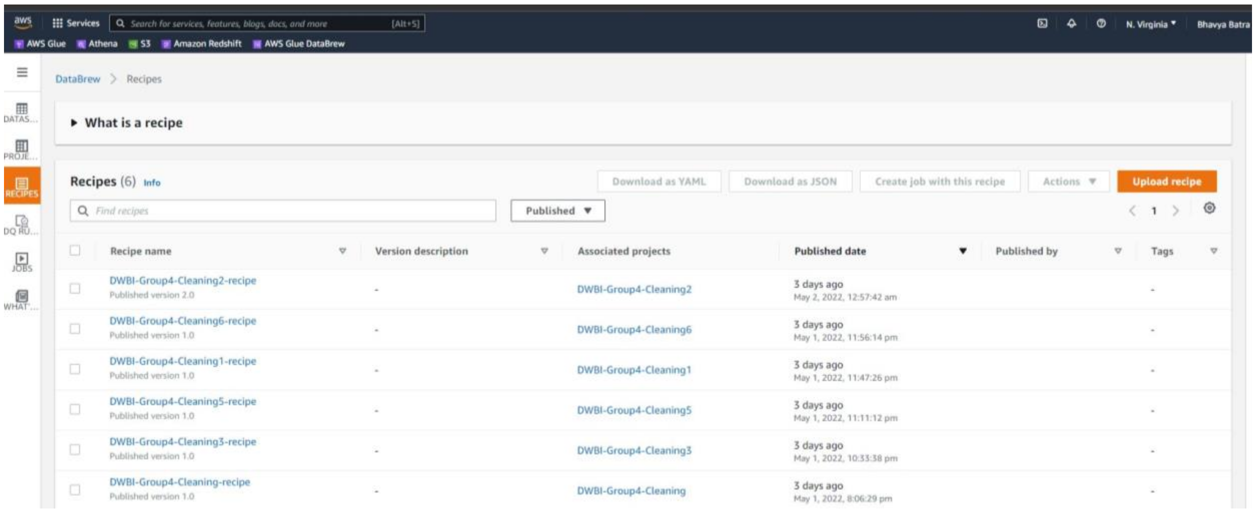
Recipe (3)

DWBI-Group4-Cleaning1-recipe
Version 1.0

Applied steps (3) | Clear all

1. Delete column Alias
2. Remove custom value from ICAO
3. Delete empty rows with missing values in ICAO

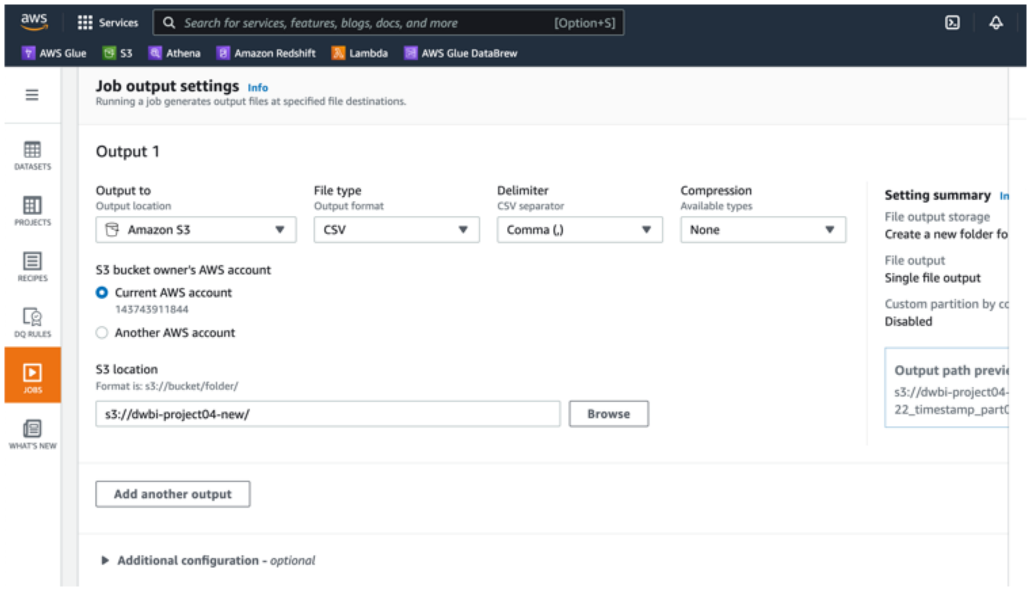
For each individual project different recipes were created. Recipes are nothing but the data cleaning process that were done to obtain clean data set. After each recipe a JOB was run. The JOB implemented the recipe to the whole dataset and then it was store to the new S3 bucket i.e., dwbi-project04-new.


DataBrew > Recipes

▸ What is a recipe

Recipes (6) Info

Recipe name	Version description	Associated projects	Published date	Published by	Tags
DWBI-Group4-Cleaning2-recipe Published version 2.0	-	DWBI-Group4-Cleaning2	3 days ago May 2, 2022, 12:57:42 am		-
DWBI-Group4-Cleaning6-recipe Published version 1.0	-	DWBI-Group4-Cleaning6	3 days ago May 1, 2022, 11:56:14 pm		-
DWBI-Group4-Cleaning1-recipe Published version 1.0	-	DWBI-Group4-Cleaning1	3 days ago May 1, 2022, 11:47:26 pm		-
DWBI-Group4-Cleaning5-recipe Published version 1.0	-	DWBI-Group4-Cleaning5	3 days ago May 1, 2022, 11:11:12 pm		-
DWBI-Group4-Cleaning3-recipe Published version 1.0	-	DWBI-Group4-Cleaning3	3 days ago May 1, 2022, 10:33:38 pm		-
DWBI-Group4-Cleaning-recipe Published version 1.0	-	DWBI-Group4-Cleaning	3 days ago May 1, 2022, 8:06:29 pm		-


Job output settings

Running a job generates output files at specified file destinations.

Output 1

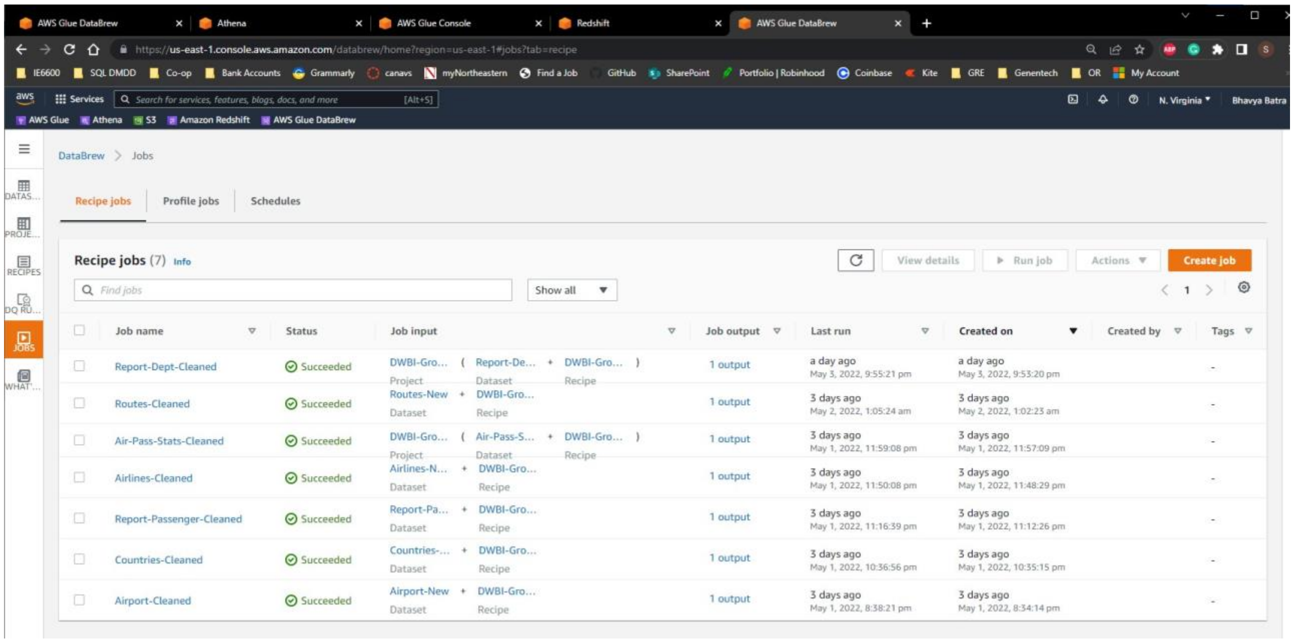
Output to	File type	Delimiter	Compression
Amazon S3	CSV	Comma (,)	None

S3 bucket owner's AWS account: Current AWS account 143743911844

S3 location: s3://dwbi-project04-new/

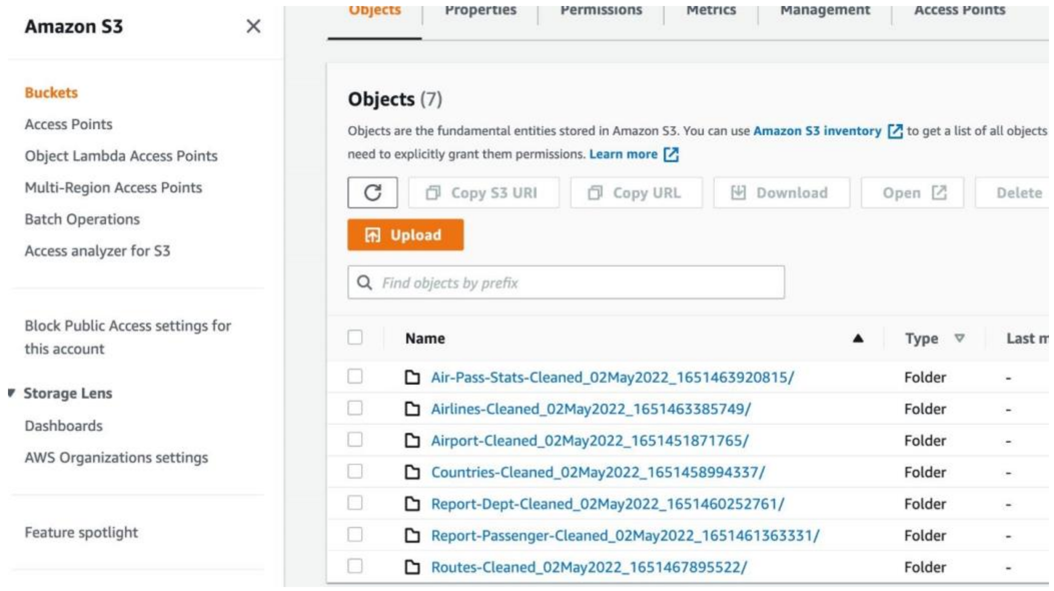
Setting summary
File output storage: Create a new folder fo
File output: Single file output
Custom partition by c: Disabled
Output path previ: s3://dwbi-project04- 22_timestamp_part0

Similarly, Jobs were run for all the recipes to store the cleaned data in the new bucket.


Recipe jobs (7)

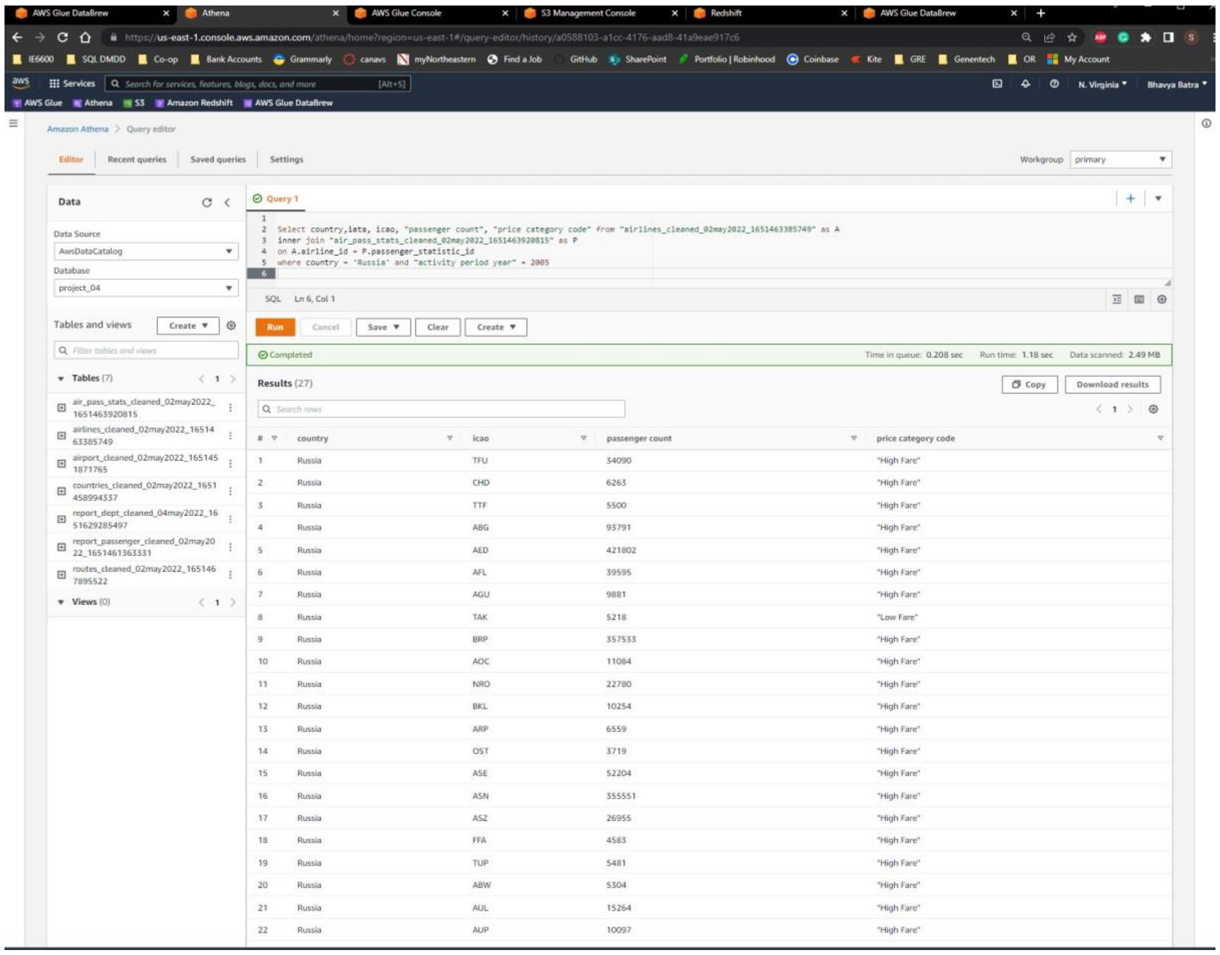
Job name	Status	Job input	Job output	Last run	Created on	Created by	Tags
Report-Dept-Cleaned	Succeeded	DWBI-Gro... Project (Report-De... Dataset + DWBI-Gro... Recipe)	1 output	a day ago May 3, 2022, 9:55:21 pm	a day ago May 3, 2022, 9:53:20 pm		-
Routes-Cleaned	Succeeded	Routes-New Dataset + DWBI-Gro... Recipe	1 output	3 days ago May 2, 2022, 1:05:24 am	3 days ago May 2, 2022, 1:02:23 am		-
Air-Pass-Stats-Cleaned	Succeeded	DWBI-Gro... Project (Air-Pass-S... Dataset + DWBI-Gro... Recipe)	1 output	3 days ago May 1, 2022, 11:59:08 pm	3 days ago May 1, 2022, 11:57:09 pm		-
Airlines-Cleaned	Succeeded	Airlines-N... Dataset + DWBI-Gro... Recipe	1 output	3 days ago May 1, 2022, 11:50:08 pm	3 days ago May 1, 2022, 11:48:29 pm		-
Report-Passenger-Cleaned	Succeeded	Report-Pa... Dataset + DWBI-Gro... Recipe	1 output	3 days ago May 1, 2022, 11:16:39 pm	3 days ago May 1, 2022, 11:12:26 pm		-
Countries-Cleaned	Succeeded	Countries-... Dataset + DWBI-Gro... Recipe	1 output	3 days ago May 1, 2022, 10:36:56 pm	3 days ago May 1, 2022, 10:35:15 pm		-
Airport-Cleaned	Succeeded	Airport-New Dataset + DWBI-Gro... Recipe	1 output	3 days ago May 1, 2022, 8:38:21 pm	3 days ago May 1, 2022, 8:34:14 pm		-

For each recipe different JOB was created and it was run to obtain all the clean datasets. The fig above shows 7 JOB's that were created for all the data sets.



	Name	Type	Last m
	Air-Pass-Stats-Cleaned_02May2022_1651463920815/	Folder	-
	Airlines-Cleaned_02May2022_1651463385749/	Folder	-
	Airport-Cleaned_02May2022_1651451871765/	Folder	-
	Countries-Cleaned_02May2022_1651458994337/	Folder	-
	Report-Dept-Cleaned_02May2022_1651460252761/	Folder	-
	Report-Passenger-Cleaned_02May2022_1651461363331/	Folder	-
	Routes-Cleaned_02May2022_1651467895522/	Folder	-

To check if the JOBs were successful, we pulled the clean data to Athena and we did a query to check if the data is being populated and if all the unwanted data are removed from the datasets.



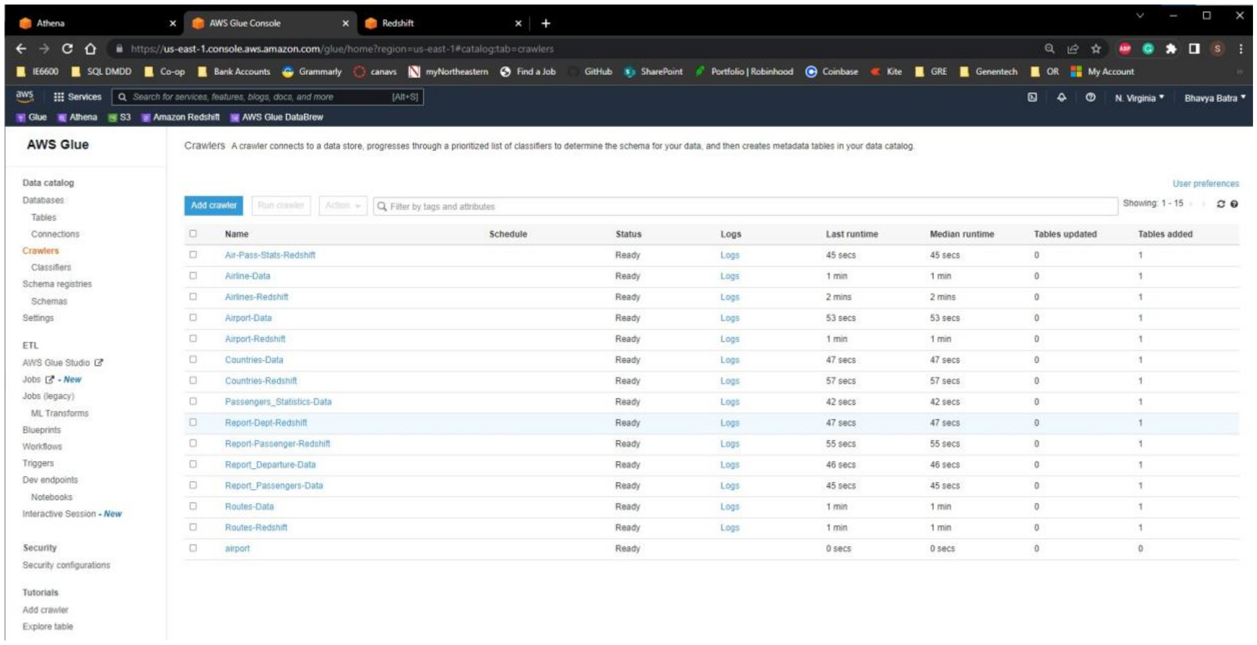
```
Select country,iata, icao, "passenger count", "price category code" from "airlines_cleaned_02may2022_1651463385749" as A
inner join "air_pass_stats_cleaned_02may2022_1651463920815" as P
on A.airline_id = P.passenger_statistic_id
where country = 'Russia' and "activity period year" = 2005
```

Completed — Time in queue: 0.208 sec — Run time: 1.18 sec — Data scanned: 2.49 MB

Results (27)

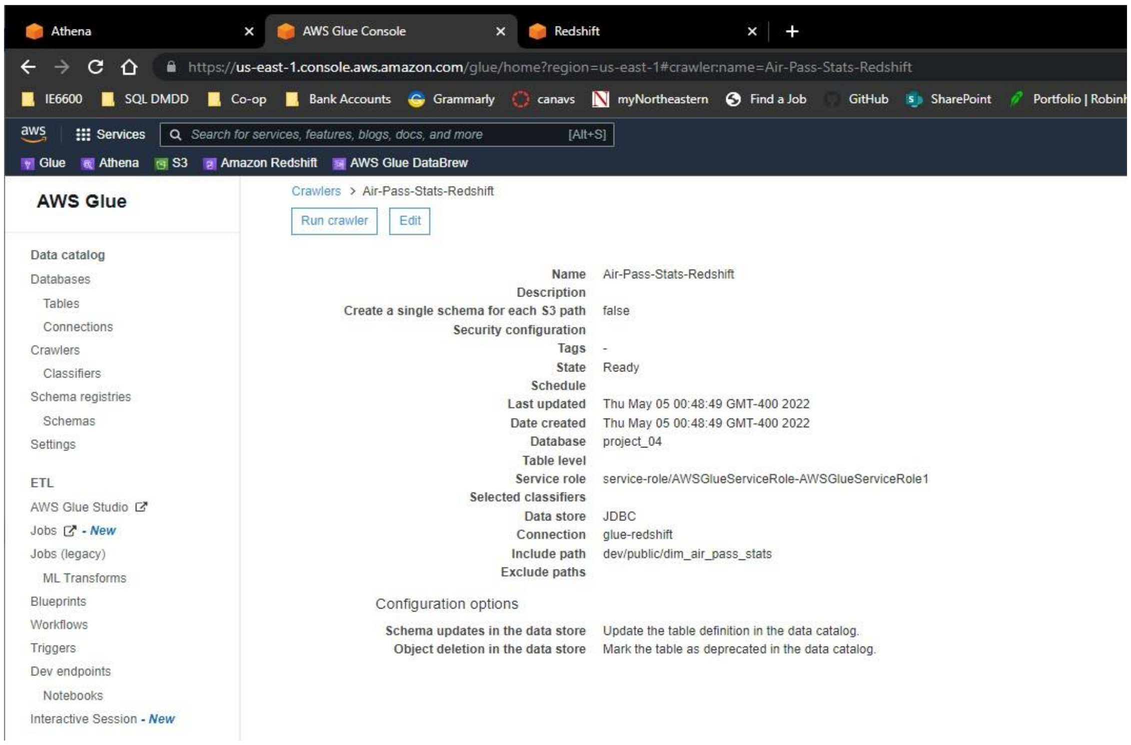
#	country	icao	passenger count	price category code
1	Russia	TFU	34090	"High Fare"
2	Russia	CHD	6263	"High Fare"
3	Russia	TTF	5500	"High Fare"
4	Russia	ABG	93791	"High Fare"
5	Russia	AED	421802	"High Fare"
6	Russia	AFL	39595	"High Fare"
7	Russia	AGU	9881	"High Fare"
8	Russia	TAK	5218	"Low Fare"
9	Russia	BRP	357533	"High Fare"
10	Russia	AOC	11084	"High Fare"
11	Russia	NRO	22780	"High Fare"
12	Russia	BKL	10254	"High Fare"
13	Russia	ARP	6559	"High Fare"
14	Russia	OST	3719	"High Fare"
15	Russia	ASE	52204	"High Fare"
16	Russia	ASN	355551	"High Fare"
17	Russia	ASZ	26955	"High Fare"
18	Russia	FFA	4583	"High Fare"
19	Russia	TUP	5481	"High Fare"
20	Russia	ABW	5304	"High Fare"
21	Russia	AUL	15264	"High Fare"
22	Russia	AUP	10097	"High Fare"

In order to load the datasets to the data warehouse we used the ETL tool data Glue. In Data glue we created different crawlers to add the tables for all the datasets. As we can see from the image below, 14 different crawlers were created. In those 14 crawlers, 7 crawlers were created to pull the data form the source table i.e., the S3 bucket and the other 7 crawlers were created in to connect it to the destination i.e Redshift.



Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
Air-Pass-Stats-Redshift		Ready	Logs	45 secs	45 secs	0	1
Airline-Data		Ready	Logs	1 min	1 min	0	1
Airlines-Redshift		Ready	Logs	2 mins	2 mins	0	1
Airport-Data		Ready	Logs	53 secs	53 secs	0	1
Airport-Redshift		Ready	Logs	1 min	1 min	0	1
Countries-Data		Ready	Logs	47 secs	47 secs	0	1
Countries-Redshift		Ready	Logs	57 secs	57 secs	0	1
Passengers_Statistics-Data		Ready	Logs	42 secs	42 secs	0	1
Report-Dept-Redshift		Ready	Logs	47 secs	47 secs	0	1
Report-Passenger-Redshift		Ready	Logs	55 secs	55 secs	0	1
Report_Departure-Data		Ready	Logs	46 secs	46 secs	0	1
Report_Passengers-Data		Ready	Logs	45 secs	45 secs	0	1
Routes-Data		Ready	Logs	1 min	1 min	0	1
Routes-Redshift		Ready	Logs	1 min	1 min	0	1
airport		Ready		0 secs	0 secs	0	0



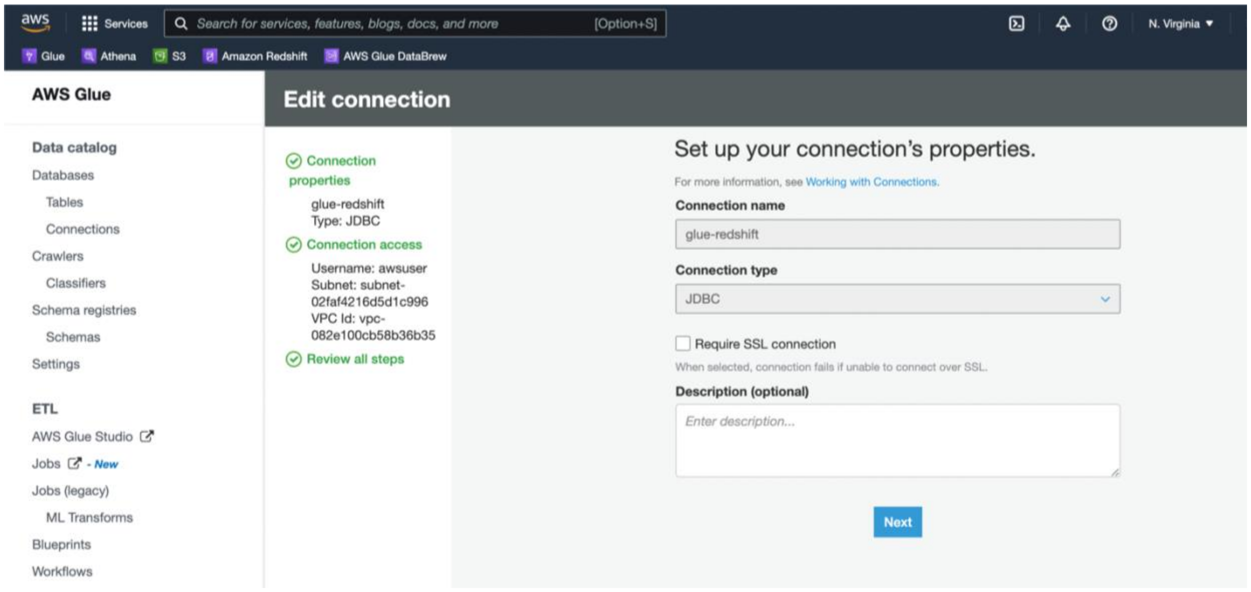
Crawlers > Air-Pass-Stats-Redshift

Name	Air-Pass-Stats-Redshift
Description	
Create a single schema for each S3 path	false
Security configuration	
Tags	-
State	Ready
Schedule	
Last updated	Thu May 05 00:48:49 GMT-400 2022
Date created	Thu May 05 00:48:49 GMT-400 2022
Database	project_04
Table level	
Service role	service-role/AWSGlueServiceRole-AWSGlueServiceRole1
Selected classifiers	
Data store	JDBC
Connection	glue-redshift
Include path	dev/public/dim_air_pass_stats
Exclude paths	

Configuration options

Schema updates in the data store	Update the table definition in the data catalog.
Object deletion in the data store	Mark the table as deprecated in the data catalog.

The above fig shows the properties of the Redshift data crawler that was created to connect the Air-Pass-Stats-Redshift table in the redshift to the source. For this we used the JDBC connection and provided the proper service role and included the path for the connection. The image below shows the connection to the Redshift.



Edit connection

Connection properties
glue-redshift
Type: JDBC
Connection access
Username: awsuser
Subnet: subnet-02faf4216d5d1c996
VPC Id: vpc-082e100cb58b36b35
Review all steps

Set up your connection's properties.

For more information, see Working with Connections.

Connection name: glue-redshift

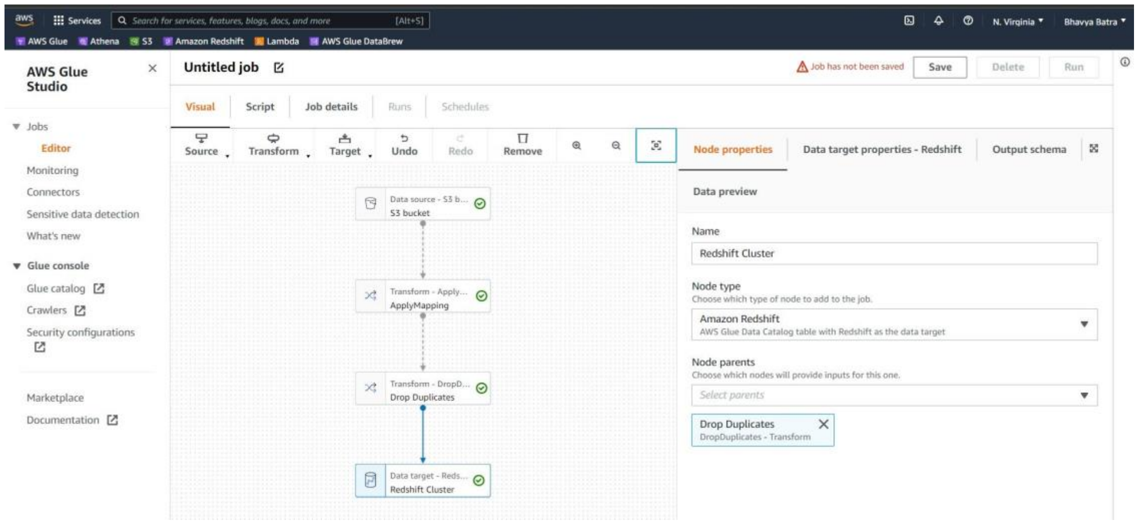
Connection type: JDBC

☐ Require SSL connection
When selected, connection fails if unable to connect over SSL.

Description (optional): Enter description...

7.0 Creation of Data Warehouse

After the crawlers were created, we were needed to create jobs. The Jobs were used in order to map the data table form the source to the data warehouse. The diagram below shows how the normal schema was created for a certain job. S3 bucket was connected to the redshift via mapping and drop duplicates transformation tools.



Untitled job

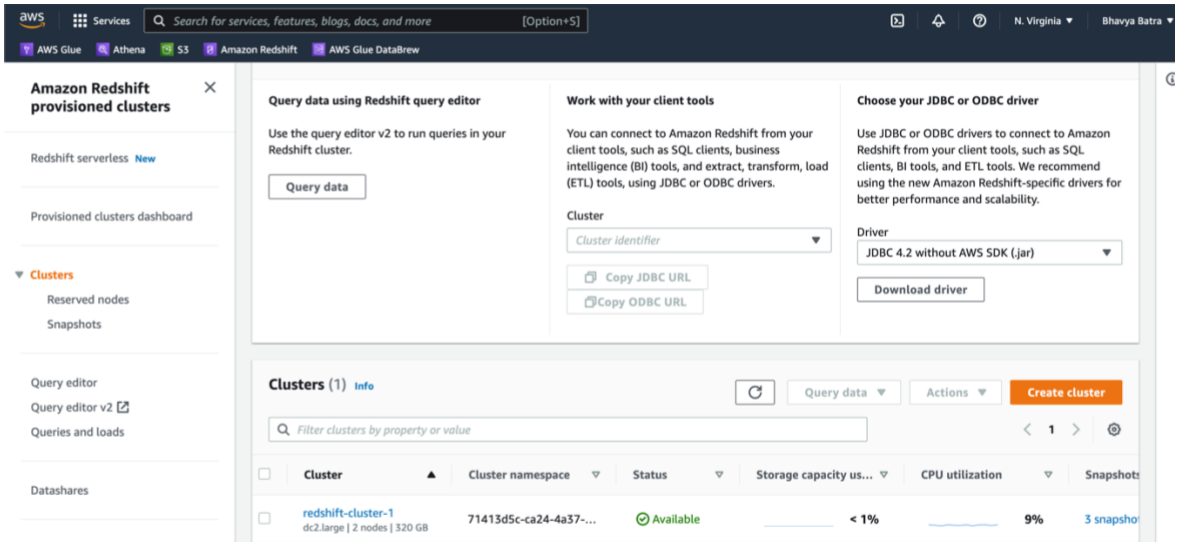
Job has not been saved

Data source - S3 b... S3 bucket
Transform - Apply... ApplyMapping
Transform - DropD... Drop Duplicates
Data target - Reds... Redshift Cluster

Name: Redshift Cluster

Node type
Choose which type of node to add to the job.
Amazon Redshift
AWS Glue Data Catalog table with Redshift as the data target

Node parents
Choose which nodes will provide inputs for this one.
Select parents
Drop Duplicates
DropDuplicates - Transform



Query data using Redshift query editor

Use the query editor v2 to run queries in your Redshift cluster.

Query data

Work with your client tools

You can connect to Amazon Redshift from your client tools, such as SQL clients, business intelligence (BI) tools, and extract, transform, load (ETL) tools, using JDBC or ODBC drivers.

Cluster: Cluster identifier

Copy JDBC URL | Copy ODBC URL

Choose your JDBC or ODBC driver

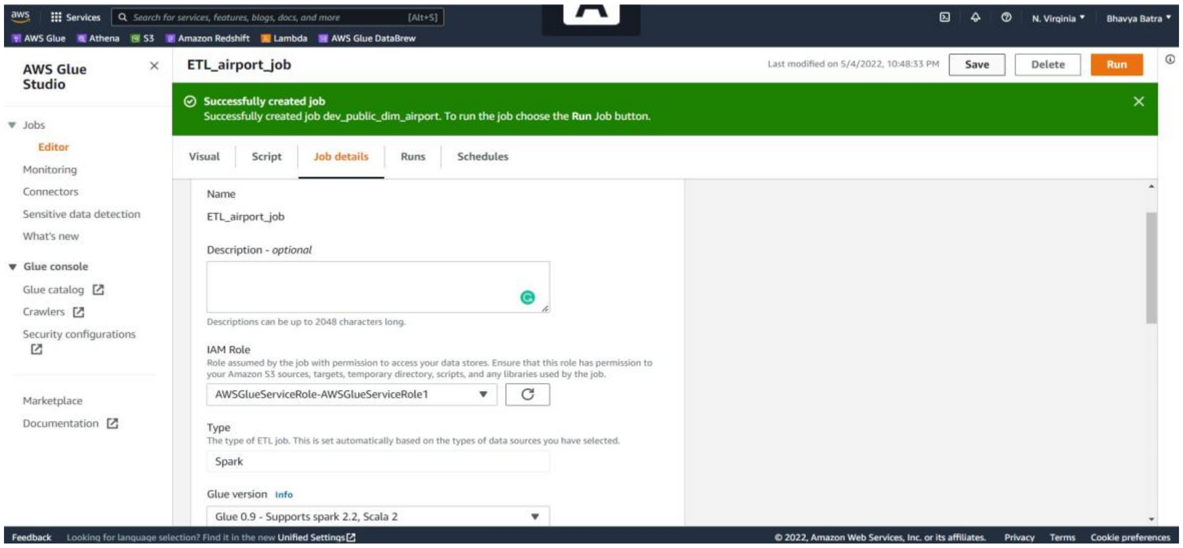
Use JDBC or ODBC drivers to connect to Amazon Redshift from your client tools, such as SQL clients, BI tools, and ETL tools. We recommend using the new Amazon Redshift-specific drivers for better performance and scalability.

Driver: JDBC 4.2 without AWS SDK (.jar)

Download driver

Clusters (1) Info

Cluster	Cluster namespace	Status	Storage capacity us...	CPU utilization	Snapshots
redshift-cluster-1 dc2.large | 2 nodes | 320 GB	71413d5c-ca24-4a37-...	Available	< 1%	9%	3 snapsho



ETL_airport_job

Last modified on 5/4/2022, 10:48:33 PM

Successfully created job
Successfully created job dev_public_dim_airport. To run the job choose the **Run** Job button.

Name: ETL_airport_job

Description - *optional*

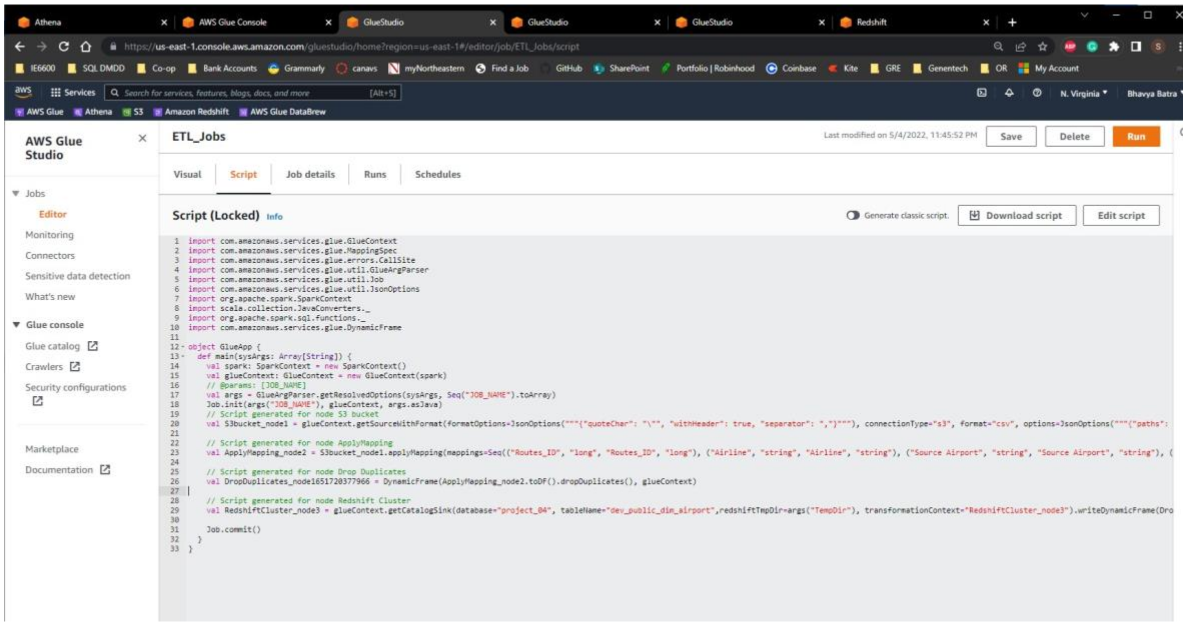
Descriptions can be up to 2048 characters long.

IAM Role
Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.
AWSGlueServiceRole-AWSGlueServiceRole1

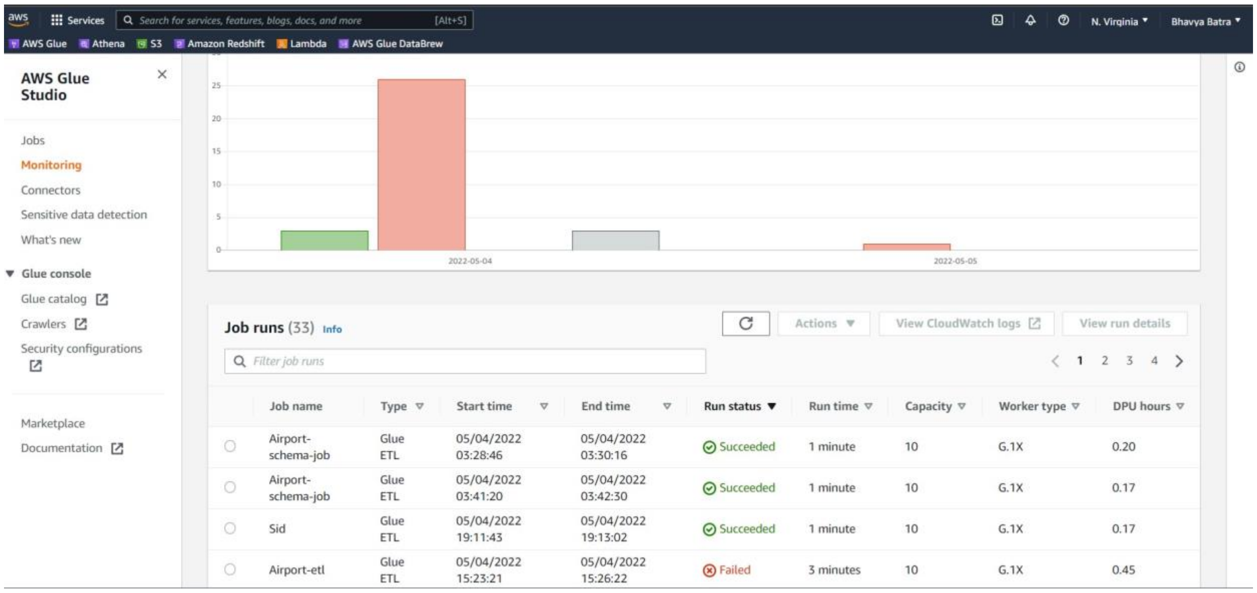
Type
The type of ETL job. This is set automatically based on the types of data sources you have selected.
Spark

Glue version Info
Glue 0.9 - Supports spark 2.2, Scala 2

Feedback Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

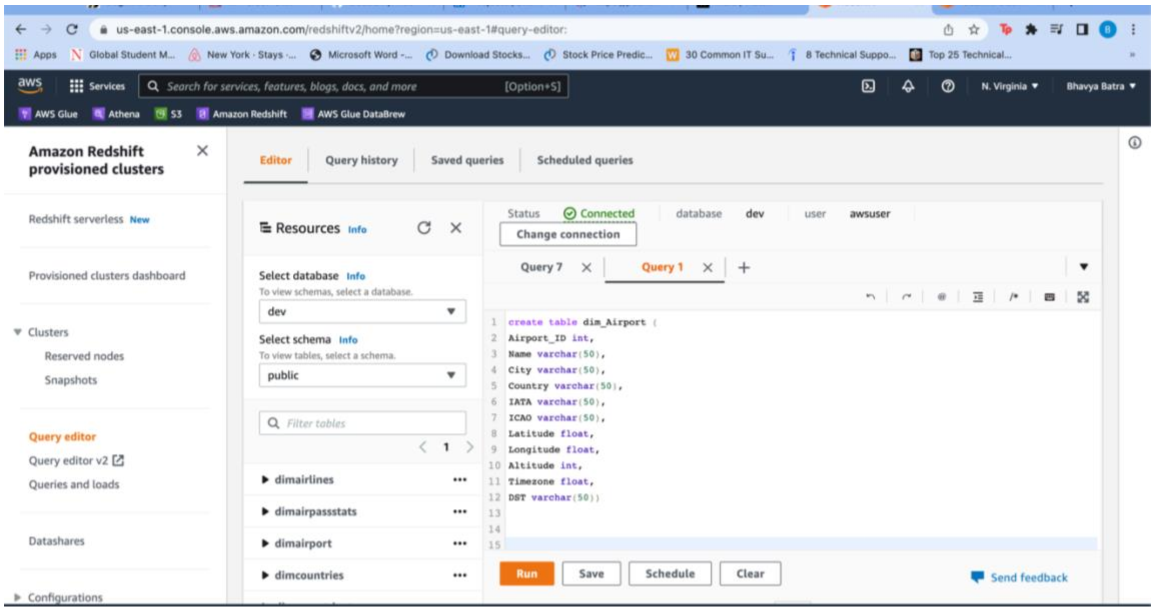


```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._
import org.apache.spark.sql.functions._
import com.amazonaws.services.glue.DynamicFrame

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    // Script generated for node S3 bucket
    val S3bucket_node1 = glueContext.getSourceWithFormat(formatOptions=JsonOptions("""{"quoteChar": "\"", "withHeader": true, "separator": ","}"""), connectionType="s3", format="csv", options=JsonOptions("""{"paths":

    // Script generated for node ApplyMapping
    val ApplyMapping_node2 = S3bucket_node1.applyMapping(mappings=Seq(("Routes_ID", "long", "Routes_ID", "long"), ("Airline", "string", "Airline", "string"), ("Source Airport", "string", "Source Airport", "string"), (

    // Script generated for node Drop Duplicates
    val DropDuplicates_node1651720377966 = DynamicFrame(ApplyMapping_node2.toDF().dropDuplicates(), glueContext)

    // Script generated for node Redshift Cluster
    val RedshiftCluster_node3 = glueContext.getCatalogSink(database="project_04", tableName="dev_public_dim_airport",redshiftTmpDir=args("TempDir"), transformationContext="RedshiftCluster_node3").writeDynamicFrame(Dro

    Job.commit()
  }
}
```

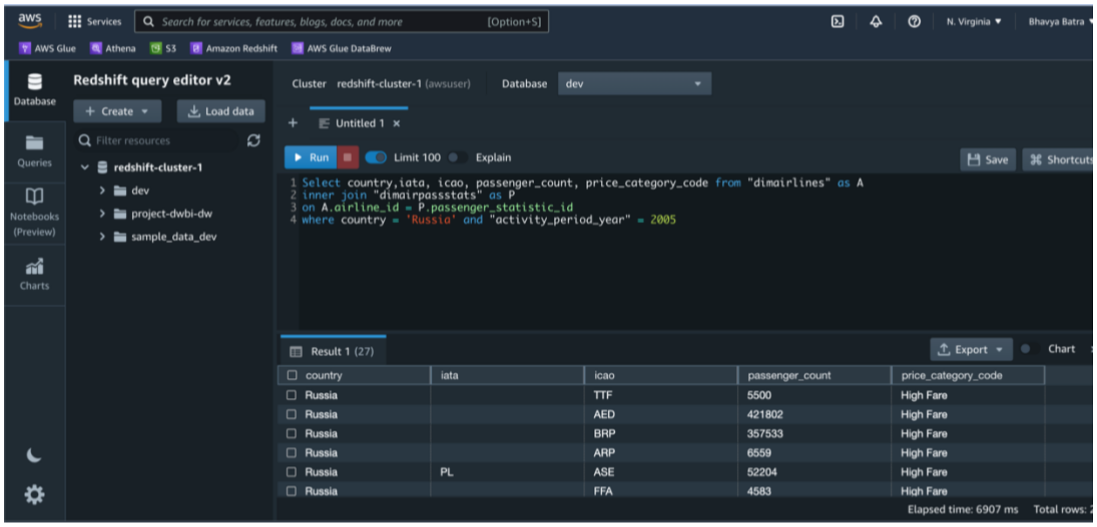


	Job name	Type	Start time	End time	Run status	Run time	Capacity	Worker type	DPU hours
○	Airport-schema-job	Glue ETL	05/04/2022 03:28:46	05/04/2022 03:30:16	Succeeded	1 minute	10	G.1X	0.20
○	Airport-schema-job	Glue ETL	05/04/2022 03:41:20	05/04/2022 03:42:30	Succeeded	1 minute	10	G.1X	0.17
○	Sid	Glue ETL	05/04/2022 19:11:43	05/04/2022 19:13:02	Succeeded	1 minute	10	G.1X	0.17
○	Airport-etl	Glue ETL	05/04/2022 15:23:21	05/04/2022 15:26:22	Failed	3 minutes	10	G.1X	0.45

Ran the queries in the query editor to check if the data has been loaded into the new tables.



```
create table dim_Airport (
Airport_ID int,
Name varchar(50),
City varchar(50),
Country varchar(50),
IATA varchar(50),
ICAO varchar(50),
Latitude float,
Longitude float,
Altitude int,
Timezone float,
DST varchar(50))
```



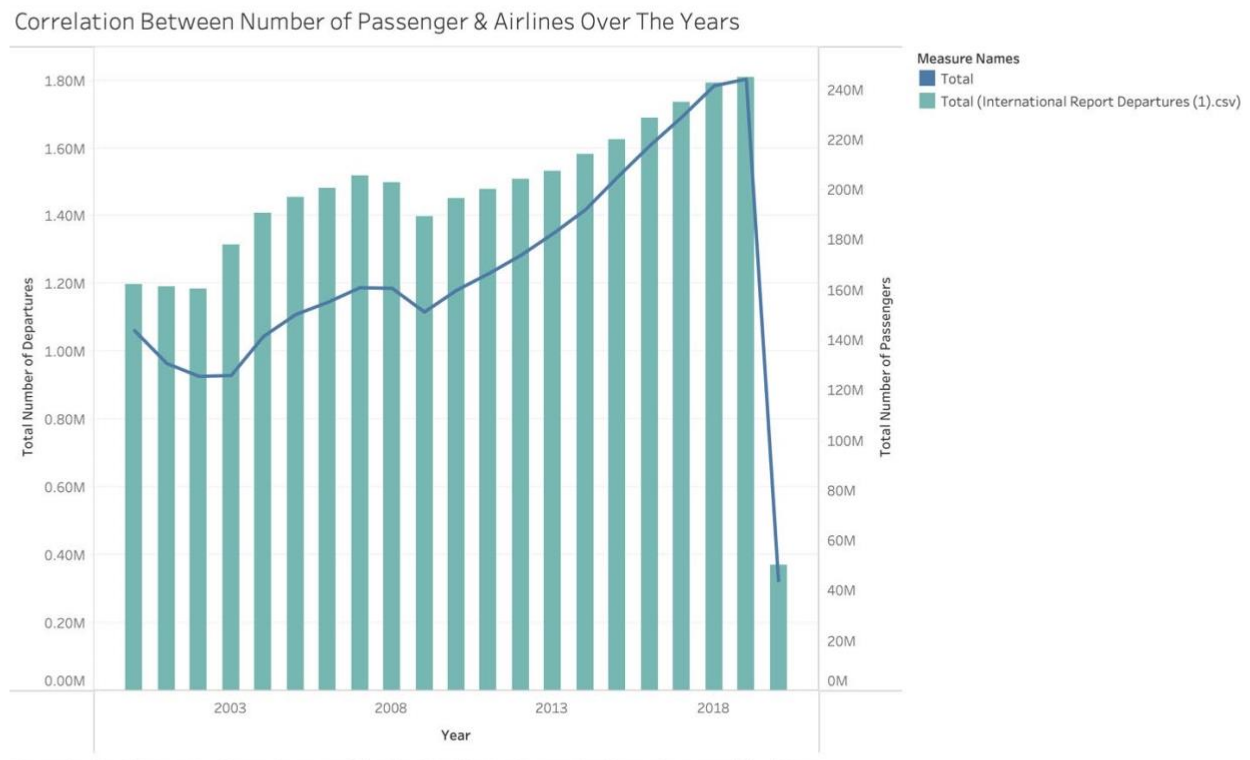
```
Select country,iata, icao, passenger_count, price_category_code from "dimairlines" as A
inner join "dimairpassstats" as P
on A.airline_id = P.passenger_statistic_id
where country = 'Russia' and "activity_period_year" = 2005
```

Result 1 (27)

country	iata	icao	passenger_count	price_category_code
Russia		TTF	5500	High Fare
Russia		AED	421802	High Fare
Russia		BRP	357533	High Fare
Russia		ARP	6559	High Fare
Russia	PL	ASE	52204	High Fare
Russia		FFA	4583	High Fare

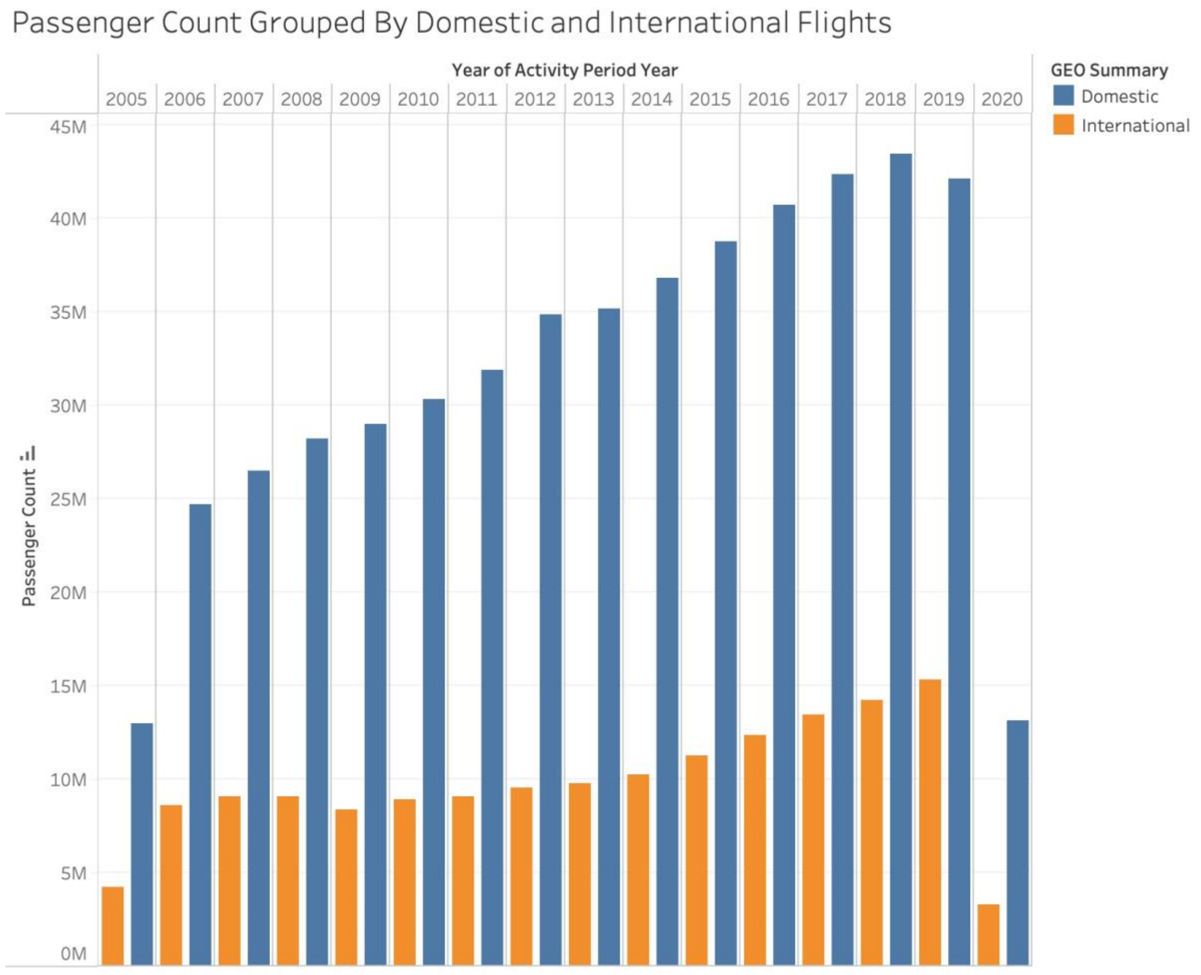
Elapsed time: 6907 ms

8.0 Visualization Expected:



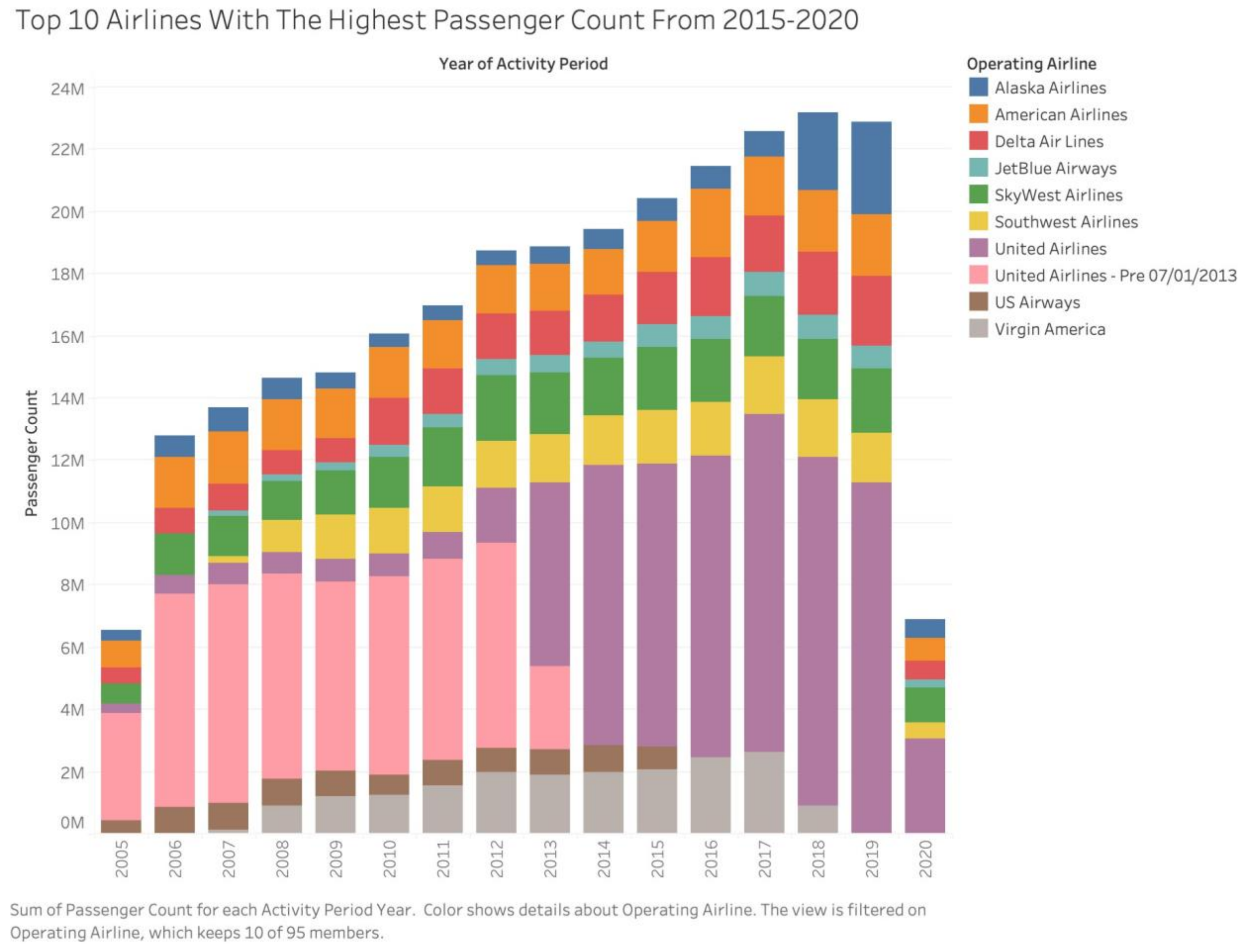
The trends of Total (International Report Departures (1).csv) and Total for Year (International Report Departures (1).csv) Year. Color shows details about Total (International Report Departures (1).csv) and Total. The data is filtered on Year (International Report Departures (1).csv), which ranges from 01/01/2000 to 01/01/2020.

- This visualization shows the relationship between the number of passengers and the number of flights departed from the year 2000 to 2020.
- It shows a strong positive correlation between the two attributes analyzed here.
- There is a sharp decrease in the number of departures and passengers in the year 2020 that can be speculated because of covid-19.



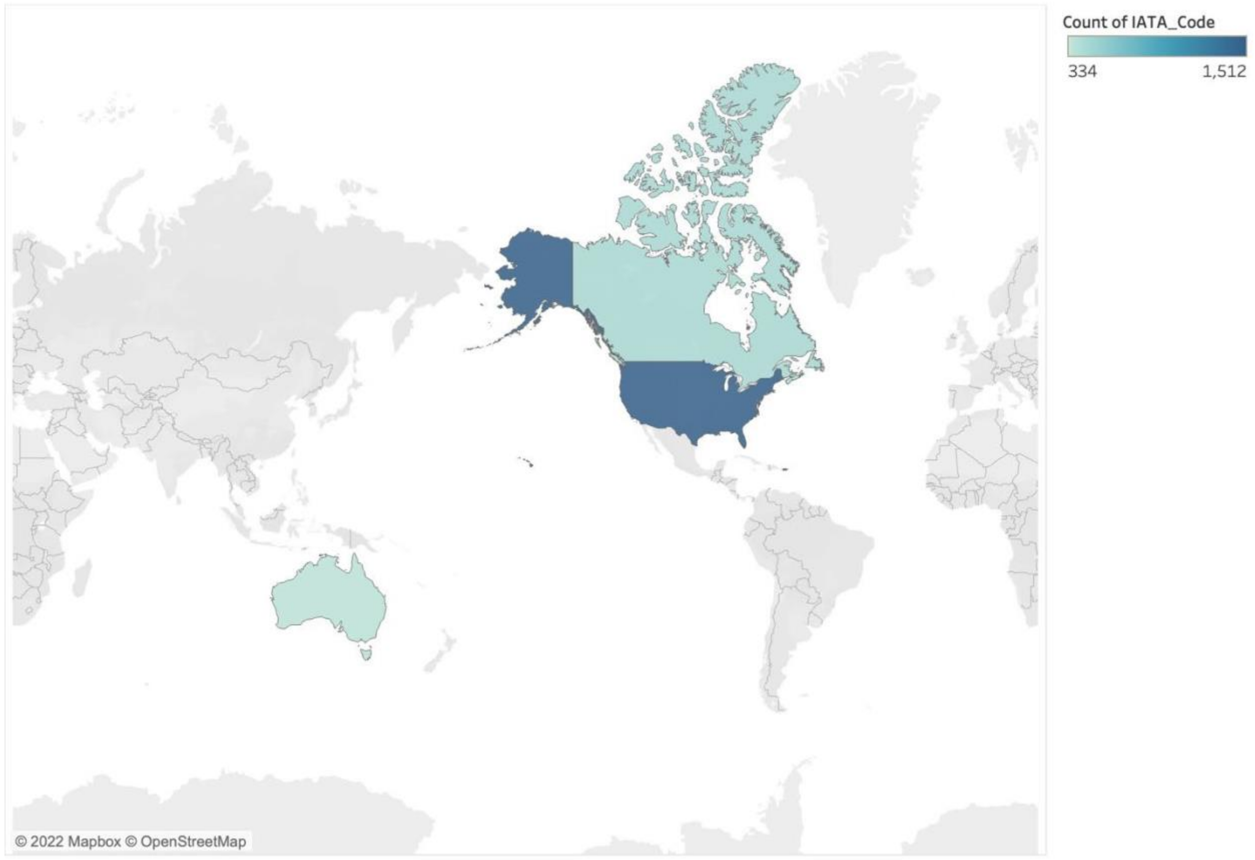
Sum of Passenger Count for each GEO Summary broken down by Activity Period Year Year. Color shows details about GEO Summary.

- This visualization shows the comparison between the number of passengers in domestic flights and international flights departing from the United States between the years 2005 to 2020.
- The number of passengers in international flights is more than twice the number of passengers in domestic flights every year. .
- Each year, the number of passengers increases for both domestic flights and international flights.
- We can conclude that the economic growth in a country influences the growth in air travel



- This visualization shows the top 10 highest performing airports in USA based on the passenger count between 2005 and 2020.
- As we can see, the visualization shows that United Airlines has monopolized the industry with almost half of all the total passengers choosing United Airlines as their preferred airlines.
- The other airlines have been performing consistently since the past two decades, except for Virgin America having a sudden increase and decrease between 2008 and 2018.

Countries With The Largest Number Of Airports



Map based on Longitude (generated) and Latitude (generated). Color shows count of IATA_Code. Details are shown for Country. The view is filtered on count of IATA_Code, which includes values greater than or equal to 302.

- This visualization shows the highest number of airports in the world based on our sample of airports dataset.
- It counts the number of unique IATA codes and shows which country has the highest number of airports(between 300 and 1512).
- As we can see, the highest number of airports are located in ISA, Canada and Australia. We can predict that this has to do with the country’s economy, size, tourism, etc. Although a lot of factors influence the number of airports in a country, airports are a symbol of growth and prosperity of the country.
- This visualization does not state the absolute facts, but only represents the major ROI of our dataset.

9.0 Modification History:

Version	Date	Author	Description of Edits
1	03/23	Team	Project Proposal (Draft)
2	04/06	Team	Second Draft
2.1	03/31	Bhavya	Data source
2.2	04/01	Ronen	Facts and Dimension
2.3	04/02	Aishwarya	ER Diagram
2.4	04/03	Siddhesh	Word Document
3	04/19	Team	Final Draft
3.1	04/19	Ronen	Data Set Description and Cleaning, Word Document
3.2	04/20	Siddhesh	SCD Types, Word Document
3.3	04/21	Aishwarya	ER Diagram updated, Data set Description
3.4	04/21	Bhavya	Data set Description, Word Document
4	04/28	Team	Loading Data to Data Warehouse
4.1	04/28	Bhavya	Loading data into AWS S3 and cleaning the data using AWS Databrew
4.2	04/29	Ronen	Loading the cleaned data into S3
4.3	04/30	Siddhesh	Loading Data from S3 to Redshift
4.4	05/02	Aishwarya	Creating Visualization
5	05/04	Team	Creating the Report and recording the Video