

|  |
| --- |
| TECHNICAL REPORT ON DESIGN AND DEVELOPMENT OF DATA STORAGE SOLUTIONS FOR ANALYSIS OF AIRLINE DELAY |
|  |
| April 13,2020  Team B1  Authored by: BILLY JACOB Student No: 10540532 |

Table of Contents

DATAMART …………………………………………………………………………………………………..

Business Vision……………………………………………………………………………………

Subject area selection………………………………………………………………………...

Key stakeholders…………………………………………………………………………….…..

DATA WAREHOUSE SCHEMA…………………………………………………………………….…..

ETL - EXTRACT TRANSFORM LOAD…………………………………………………………….….

ETL Architecture for Airline Delay Data Warehouse……………………….……

ETL processes for importing data into Dimension and Fact Table…..……

RELATIONAL DATABASE AND GRAPH DATABASE ………………………………………….

Comparison between Realtion and Graph Database queries…………….…

REPORTING ANAD VISUALISATION USING SSRS AND TABLEAU………..……………

 SSRS……………………………………………………………………………………………………

TABLEAU……………………………………………………………………………………………..

APPENDIX …………………………………………………………………………………………………….

Appendix A : Data Mart Table Creation Scripts……………………………….…..

Appendix B : Stored procedures for ssrs reports…………………………….…..

Appendix C : Create Table for Neo4j Database……………………………….…..

Appendix D: Create Table for Relational Database ………………………….….

REFERENCES AND BIBILIOGRAPHY…….………………………………………………………….

## 1.**DATAMART**

A data warehouse is used to collect and store large amounts of data from multiple data sources. It can be used for analysis by organizations to make business decisions through valuable insights. Through collection of data for a long time the historical records accumulated can be used for detailed analysis by data scientists and analysts. The Data Mart is a subset of the data warehouse and is specifically oriented towards a domain or department.

# 

“What makes data warehouse unique is its four main characteristics viz., Subject oriented, Integrated, Non-Volatile, Time-variant.

* Subject Oriented – specific subject area can be analysed (Data Mart)
* Integrated – Even if collected from different data sources they will be categorized under the same datatype in the DataMart.
* Non-Volatile – Its stable once its brought inside the DataMart and cannot be overridden by all users.
* Time Variant – It provides data with respect to the change in time”.

Normally organizations come up with different types of queries like tactical query, strategy query and update query. The strategic query provides the essential information to determine the decision to be made on a long term. A strategic query is something which tends to answer questions about what and why an event happened in the past and what may have happened in the future. A tactical query provides information to rank and file elements in the field that need to respond quickly to a set of unfolding events. Tactical queries tend to produce a very small result set.

The Bureau of Transportation Statistics, which is a part of the United States Department of Transportation, makes accessible information on the nation's transportation systems. We are using the dataset that contains information about various Airlines on Time performance for the year 2019.

This data is chosen from [www.transtats.bts.gov](http://www.transtats.bts.gov) and the airlines operating from the following east American states viz., New York, Florida, Alabama, Pennsylvania and North Carolina are taken into consideration for analysis.

## Subject area selection

The subject area selected is the Airline delay in all airports of the following five East American States.  
 1. Alabama  
 2. Florida  
 3. New York  
 4. North Carolina  
 5. Pennsylvania  
The reason for the selection of this subject area is that the flight delay is a serious issue which causes significant loss and strain to the airlines, passengers and to the society as well at many billion dollars per year in the US Economy. So, a proper approach in the design of the Data Mart for this dataset will help us visualize the different delays in airports and help us formulate solutions to overcome them.

## Business Vision

While designing data Marts it is of prime importance to make the data accessible and highly consistent. The security must be proper and the data Mart should have better access controls. It must have a low operating cost as well as highly scalable. The maintainability of the data Mart should be also be high. Here our data Mart will be helpful in analysing the Delay data with detailed information on the individual delay component as well as the aggregate delay data. Our key motive is to analyse the delay metrics for all airlines operating in the five major states in the east of US and find the actual delay reason which can be reduced by taking proper measures.



The vision is to reduce the loss for all the stakeholders at their own terms. The airlines can reduce the spending on fuel, air fleet cost, Maintenance cost and crew cost. The passengers can save time which is crucial, and the airport authorities also will find it easy to handle passengers if the delays are reduced considerably.

## **Key Stakeholders**

The Key stake holders here are the

* Airlines operating in US
* The Passengers, Crew members
* airport. officials.

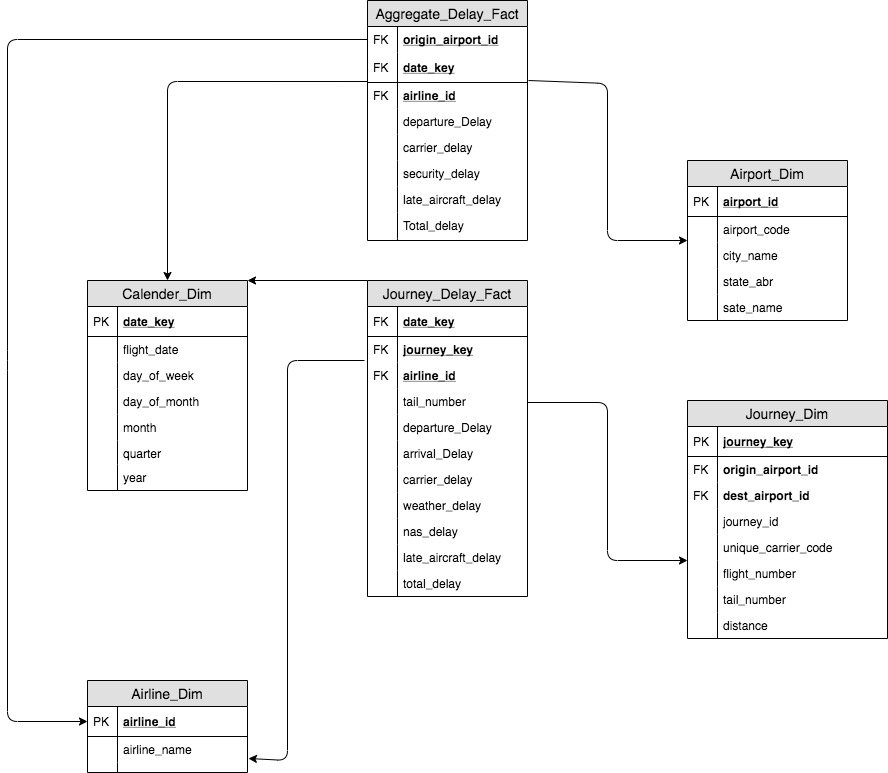


The delay of aircraft affects the passengers by getting them late for their scheduled tasks. Even though the passengers are not affected by cost directly they lose significant amount of time in the delay process.

The flight delay against schedule will result in various costs for the airport officials including the need to accommodate disrupted passengers and the cost of aircraft repositioning

It’s impact on airline can be measured using two variables viz., Scheduled buffer and flight delay against schedule. The scheduled buffer impacts the airline by resulting in the need to pay more for the pilots and crew staffs since they have to work more than the blocked time. So, it results in the need to change the fleet plan and eventually results in poorer aircraft utilization and larger fleet.

## 2. Data Warehouse Schema:



Data warehouse modelling can be done using many different techniques like ER modelling, Relational modelling and Dimensional modelling. The ER modelling is used for the conceptual modelling of operational databases while the Relational modelling is used for the logical modeling of operational databases. The dimensional modelling is used specifically for the analytical database design and is the most followed method of modelling for data warehouse and data Mart.

Here for our dataset we have used dimensional modelling since it helps in designing of subject oriented analytical databases and we have exploited this feature in designing a data mart for calculating the delay in airlines. The dimensional modelling in addition to the conventional usage of primary and foreign keys and constraints distinguishes itself through the usage of two types of tables:

* Dimensions
* Facts

Dimension table usually contains information about the detail of the subject area of analysis. They form the basis for analysis from the Fact table. The information in the dimension table is usually textual and contains more information than the fact table.

The Fact table contains foreign keys from different dimension tables and the fact measure corresponding to the area of analysis. Here in our data set the fact being analyzed is the delays in the airline.

Here while designing schema for our dataset we adhered to the Fact constellation or the Galaxy Schema. The major reason for implementing the galaxy schema is the need to have a detailed fact table to identify the individual delay for flights which represents every single fact and an aggregate fact table which helps in summarizing the delay fact of multiple flights for every airline operating. The slowly changing attributes are included in the dimension table.

Advantages of Galaxy Schema:

* The aggregated fact table has the advantage of providing quicker query details compared to the detailed fact table.
* The major advantage of this schema is that the aggregated table can be utilized for faster query performance while at the same time the detailed fact table retains the ability to do depth analysis by capturing every single fact. **Please refer Appendix A for the Data Mart Fact table and Dimension table create scripts**

# 3. ETL – Extract Transform and Load

# ETL is the process of Extracting, Transforming and Loading Data. “Everyone understands the three letters: You get the data out of its original source location (E), you do something to it (T), and then you load it (L) into a final set of tables for the business users to query”(Ralph,Kimball and Margy,Ross, 2013, p. 443).

Extraction:

The Extraction layer retrieves the data from the operational databases.

Transformation:

The Transformation layer transforms the data by creating new columns, derived columns, aggregated columns, changing data types etc. so that the data adheres to the target Data warehouse.

Loading:

The Loading layer loads the transformed data into the target data warehouse.

Why we Need ETL Infrastructure?

The ETL process is a recurring activity of the Data Warehouse Infrastructure. The ETL activity may place once in a week, month or a year based on the business requirements of the organization so it is vital to have a proper infrastructure to handle this recurring activity. Instead of having a changing pipeline to extract transform and load data if we have a fixed and agile ETL infrastructure which adheres to the current object model of the Data Warehouse it will be easy to bring data each and every time a data import is required.

# 

ETL Tools

Some of the prevalent ETL tools that are prevalently used by the organisations are.

* AWS Glu
* Blend
* Talend
* Pentaho
* IBM Infosphere
* Microsoft SSIS etc.

A picture containing drawing

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA picture containing drawing

Description automatically generated

A close up of a logo

Description automatically generated

A picture containing artifact, building, brick, drawing

Description automatically generated

A picture containing drawing, table

Description automatically generated

ETL Tool Used for the Airline Delay Data Warehouse

We have used the Microsoft SQL Server Integration Services and Visual Studio as the GUI to load data into the Data Warehouse. ****SQL Server Integration Service (SSIS)**** is a component of the Microsoft SQL Server database software that can be used to execute a wide range of data migration tasks. SSIS is a fast & flexible data warehousing tool used for data extraction, loading and transformation like cleaning, aggregating, merging data, etc (*SSIS Tutorial for Beginners: What is, Architecture, Best Practices*, para. 1) .

ETL Architecture for Airline Delay Data Warehouse

A close up of a sign

Description automatically generatedA drawing of a person

Description automatically generated

A close up of a logo

Description automatically generatedA close up of a sign

Description automatically generatedA picture containing drawing, table

Description automatically generated

Transformation

Loading

Extraction

ETL Processes for Importing Data into Dimension and Fact Tables

Populating the Airlines\_Dim

A screenshot of a cell phone

Description automatically generated

Source: airline\_data.csv

Destination: Airlines\_Dim

There are 2 columns in the input file and there are 2 columns in the destination table, so transformation was not required for this scenario.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedPopulating the Airport\_Dim

Source: dbo.delays

Destination: Airport\_Dim

There are 38 columns in the source table and there are 5 columns in the destination table, Data transformation was not required because the values in the source and the destination table are of the same date.

A screenshot of a cell phone

Description automatically generated

Populating the Journey\_Dim

Source: dbo.delays

Destination: Journey\_Dim

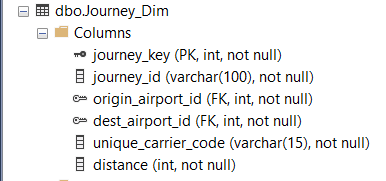
There are 38 columns in the source table and there are 6 columns in the destination table, in which journey Id was a derived column

Created by concatenating three other columns.

Derived Column: journey\_id

Derived Column Formula: (unique\_carrier\_code + "-" + (DT\_STR,30,1252) origin\_airport\_id + "-" + (DT\_STR,30,1252)

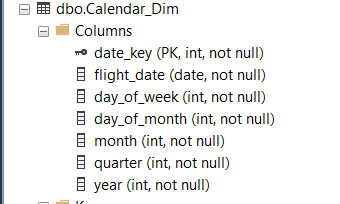
dest\_airport\_id).

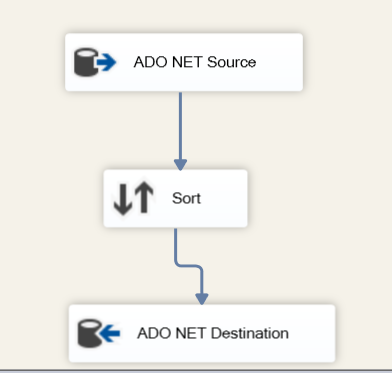


A screenshot of a cell phone

Description automatically generated

Populating the Calendar\_Dim





Source: dbo. delays

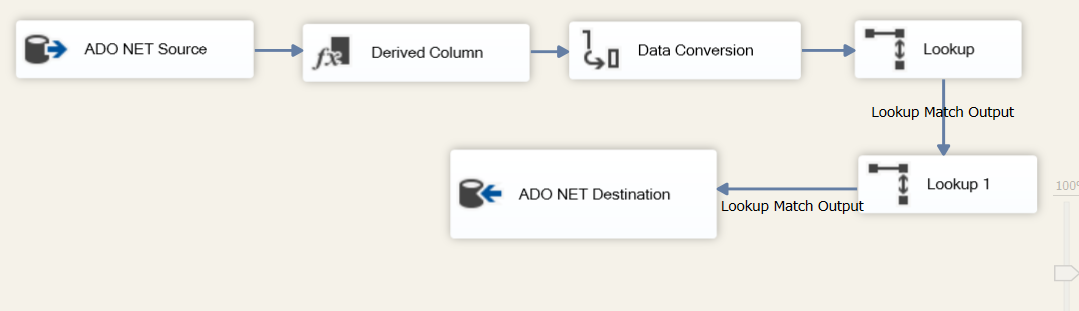
Destination: Calendar\_Dim

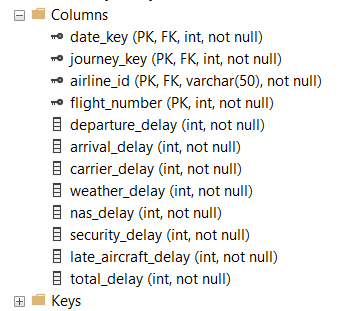
There are 38 columns in the source table and there are 7 columns in the destination table.

Sort Column: flight\_date

The flight date column was sorted before populating it into the Calendar\_dim so the dates would be in the ascending order.

Populating the Journey\_Delay\_Fact





Source: dbo. delays

Destination: Journey\_Delay\_Fact

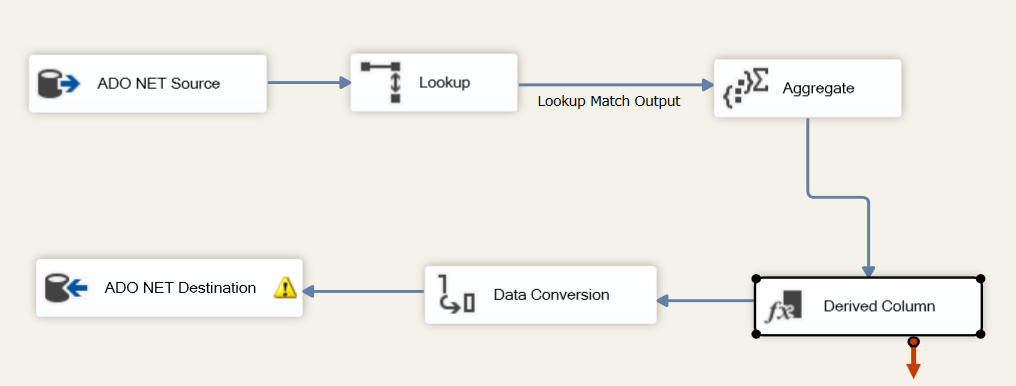
There are 38 columns in the source table and there are 12 columns in the destination table.

Derived Column: journey\_id, total\_delay

Data Conversion: journey\_id converted from DTWSTR to DTSTR.

Lookup: Lookup for journey\_Key

Lookup1; Lookup for Date Key

Populating the Aggregate Delay Fact

Source: dbo. delays

Destination: Aggregate\_Delay\_Fact

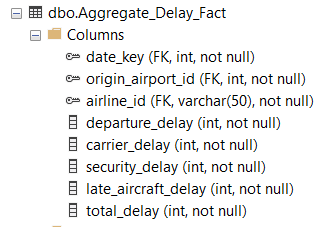
There are 38 columns in the source table and there are 8columns in the destination table.

Derived Column: to calculate the total delay

Data Conversion: values of all delays converted from DT\_18 to DT\_14

Aggregate: Aggregate data grouped by origin airport and airline

Lookup; Lookup for Date Key



4. Relational Database vs Graph Database

Relational database

A relational database is a type of [database](https://www.oracle.com/ie/database/what-is-database.html) that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points (*What is a relational database?*, para. 1). Chris Date says that “The reason that such systems are called ‘relational’ is that the term ‘relation’ is essentially just a mathematical term for a table.” (An Introduction to Database Systems, 6th ed., Addison-Wesley, 1995, p. 22).

.

A picture containing drawing

Description automatically generated

Graph database

A Graph is a collection of vertices and indexes. A Graph database is a type of [database](https://www.oracle.com/ie/database/what-is-database.html) that is mainly composed of two elements or building blocks called as a node or a relationship. Each node in the Graph database represents or denotes an entity or a class and each relationship denote how two nodes are associated. Here Each node represents different things. Each nodes and relationships in a Graph Database are identified by a unique identifier. A Graph database management system is an online Database management system with CRUD methods that expose a graph data model. (Robinson, Webber and Eifrem, 2015, p. 5)

.



Comparison between Relational Database and Graph database queries

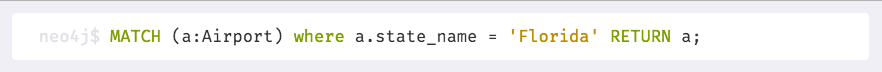
Relational databases use Structured Query Language whereas Graph databases use Cypher Query Language. We have executed seven queries to describe the differences of storing/retrieving data in/from relational and graph database. Please refer Appendix C and Appendix D for the table creation scripts for Relational and Graph Database.

Query 1: All details of the airports in State Florida.

A picture containing bird

Description automatically generated

SQL QUERY:

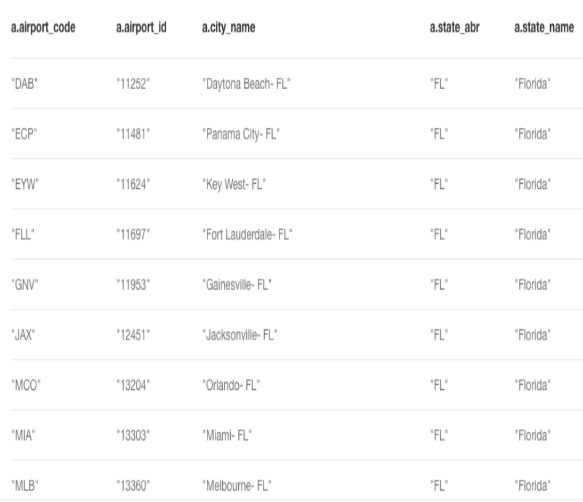


CQL QUERY:

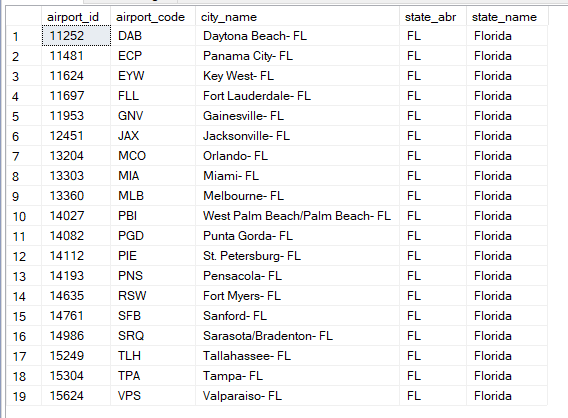
SQL Output

CQL Graph Output

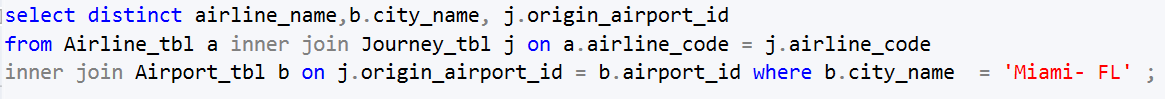
CQL Table Output

A picture containing food

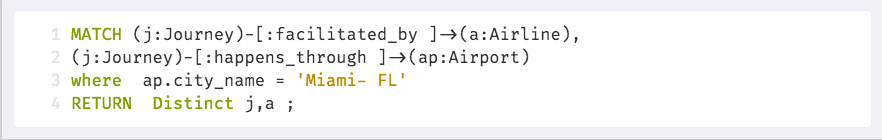
Description automatically generated



Query 2: Names of all airlines operating from Miami-FL city (origin city as Miami-FL).



SQL QUERY:



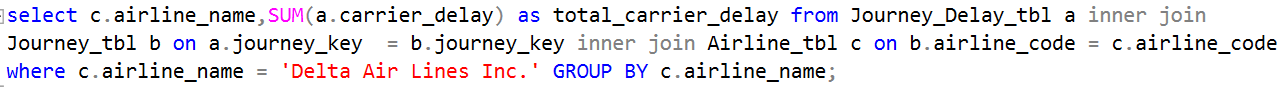
CQL QUERY:

CQL Graph Output

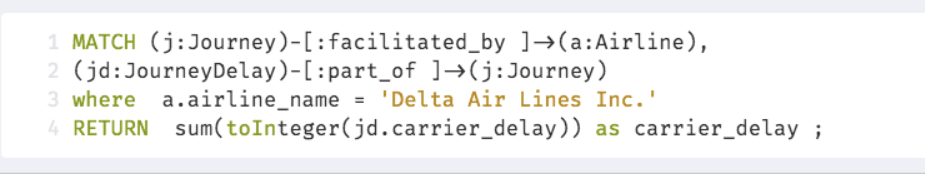
CQL Table Output

SQL Output

Query 3: Total carrier delay of Delta Air Lines Inc. in the year 2019.



SQL QUERY:



CQL QUERY:

CQL Graph Output

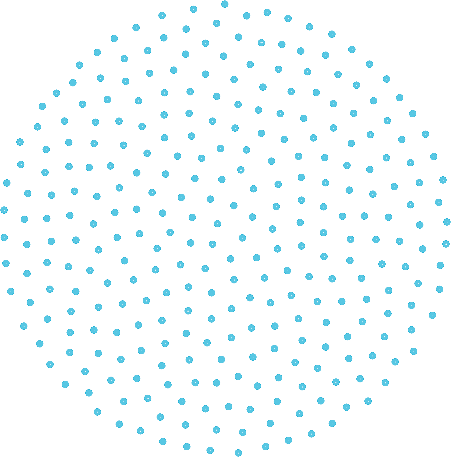
CQL Table Output

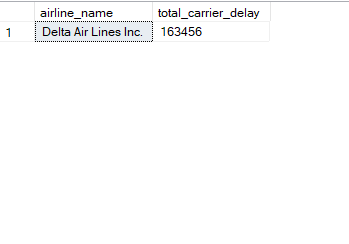
SQL Output

CQL Graph Output

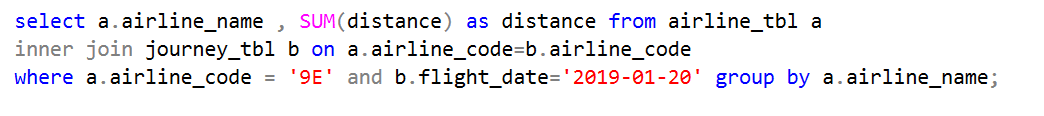
CQL Table Output

SQL Output

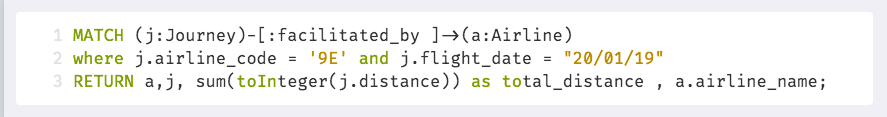




Query 4: Distances travelled by airline with airline code 9E ON 2019-01-20.



SQLQUERY:

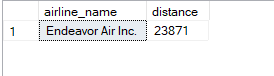


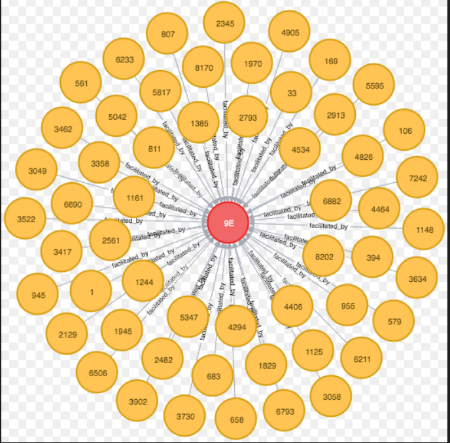
CQL QUERY:

CQL Graph Output

CQL Table Output

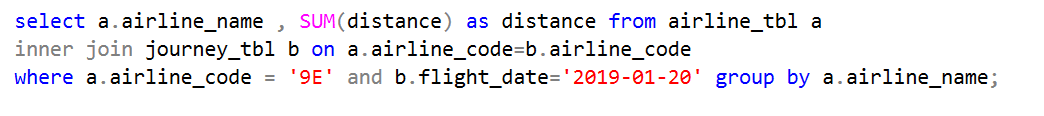
SQL Output

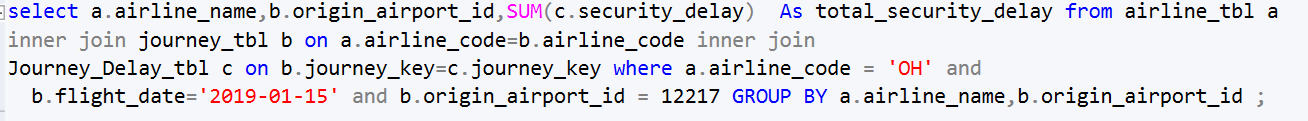




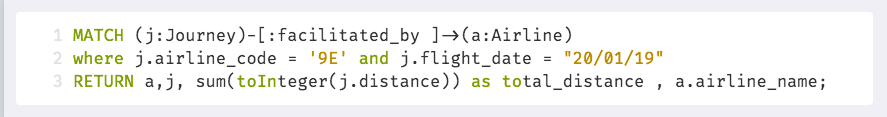
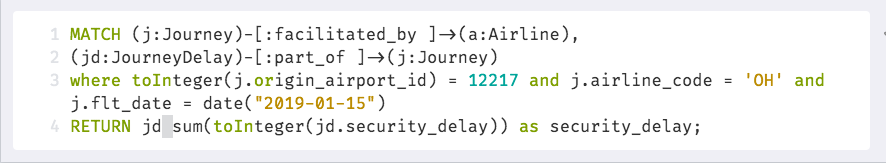


Query 5: security delay for airline code OH in airport 12217 on 15/01/2019





SQLQUERY:

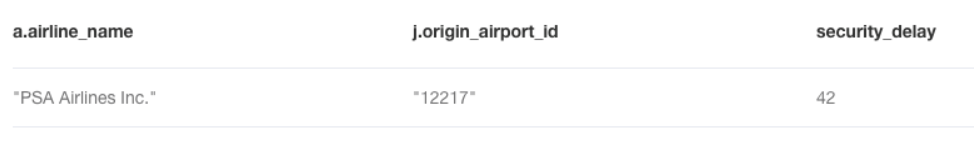
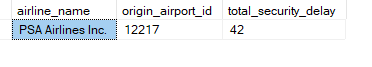
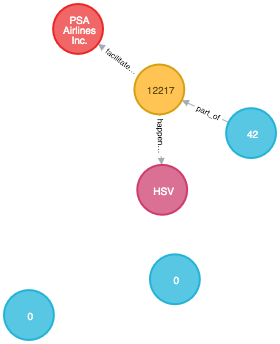


CQL QUERY:

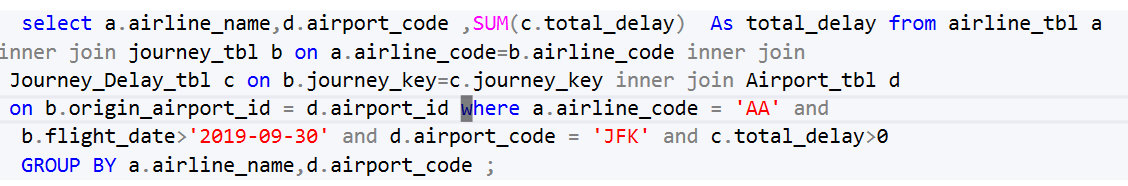
CQL Graph Output

CQL Table Output

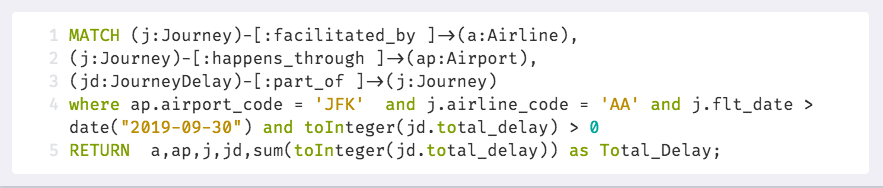
SQL Output



Query 6: total delay for airline code AA from origin airport code JFK on for the fourth quarter of 2019



SQLQUERY:

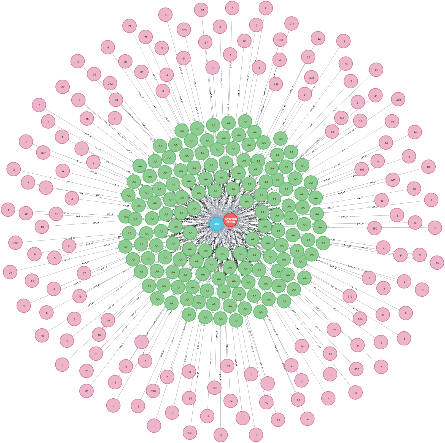
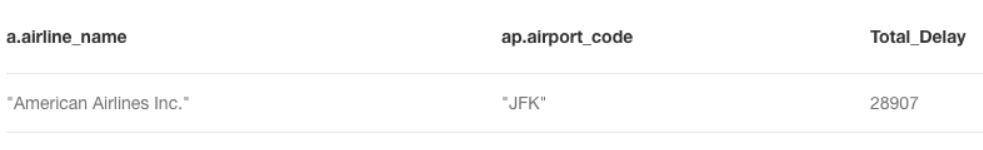


CQL QUERY:

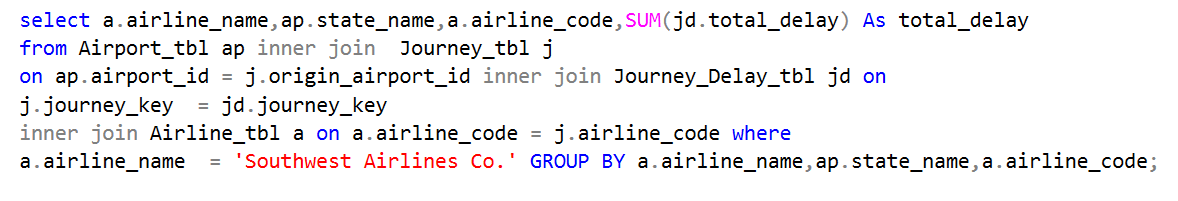
SQL Output

CQL Table Output

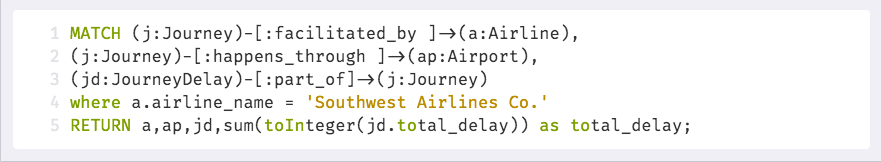
CQL Graph Output



Query 7: total delay of Southwest Airlines Co. operating from each state as origin



SQLQUERY:

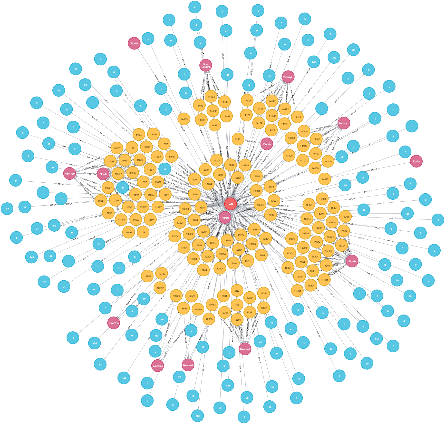


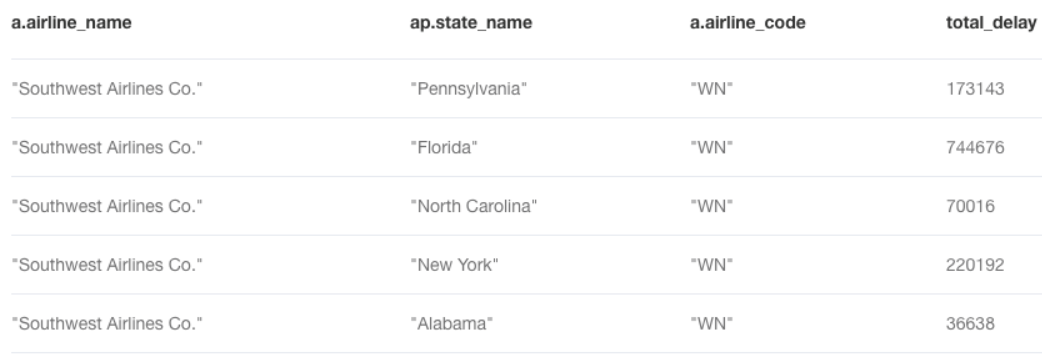
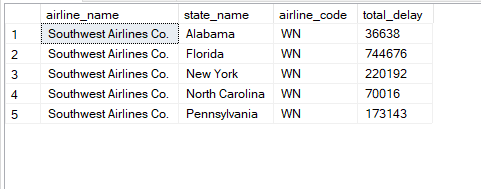
CQL QUERY:

CQL Graph Output

CQL Table Output

SQL Output





After Executing all the seven queries in both the Relational and the Graph databases we have found some differences between both the Databases and we have provided the Inference below.

Inferences made after comparing Graph and Relation Databases

Storage of Data

|  |  |
| --- | --- |
| GRAPH | RELATIONAL |
| Data is stored in the form of Nodes and Vertices. | Data is stored in the form of Table structures. |
| Nodes, Relationships and Properties are physically stored in three Different Files. | Every Table created in the Relational Database is Physically stored in a individual Files. |
| Uses More Storage space as all the relations must be stored. | A picture containing drawing  Description automatically generated  Uses Less storage space compared to Graph database. |
| Shuffling of data is complex as each data has many relations pointing to it. | Shuffling of data is easy compared to the Graph Databases. |

Retrieval of Data.

|  |  |
| --- | --- |
| GRAPH | Relational |
| Cypher queries are implemented to retrieve data | A picture containing drawing  Description automatically generatedStructured queries are implemented to retrieve data |
| Implementation of Join queries are much easier | Implementation of Join queries are complex if the  Number of tables from which the data needs to be retrieved is more. |
| Graph databases use the paths and indexes to reach the criteria much faster than the relational database. | Relational databases search all the data present in the table to find anything that meets the search criteria. |

5.REPORTING AND VISUALISATION USING SSRS AND TABLEAU.

SSRS

SSRS is a report generating software developed by Microsoft. It is a server-based reporting tool. The SSRS can be accessed using the Microsoft Visual Studio for developing complex reports. It is ideal for generating paginated reports. Using Visual Studio, it can be connected to the Microsoft SQL Server sohat both users and DB administrators can generate Reports. SSRS is used when we must generate a report which displays lot of textual and numerical data. It is more commonly used in the operational databases.

TABLEAU

Tableau is a visualization, Business Intelligence and analytics platform. It is inspired from the Excel Architecture. It can be used by everyday people to generate reports, visualizations and other insights without much training or knowledge. It works on the concept of drag and drop data. It can be used to solve everyday problems and to create value by easy analysis of data. In this project we have created four visualizations to show the different insights on the airline delay that is happening on a day to day basis..

Airline Delay Reports using SSRS

Report 1: Total Departure Delay by flight origin State

The following Reports show the location-based data by denoting the location of the states in the United State Map. This report is dynamically generated by providing the state name as the input value.

A close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA picture containing text, map, black, sitting

Description automatically generated

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Report 2: Late Aircraft Delay per Airline Per Month

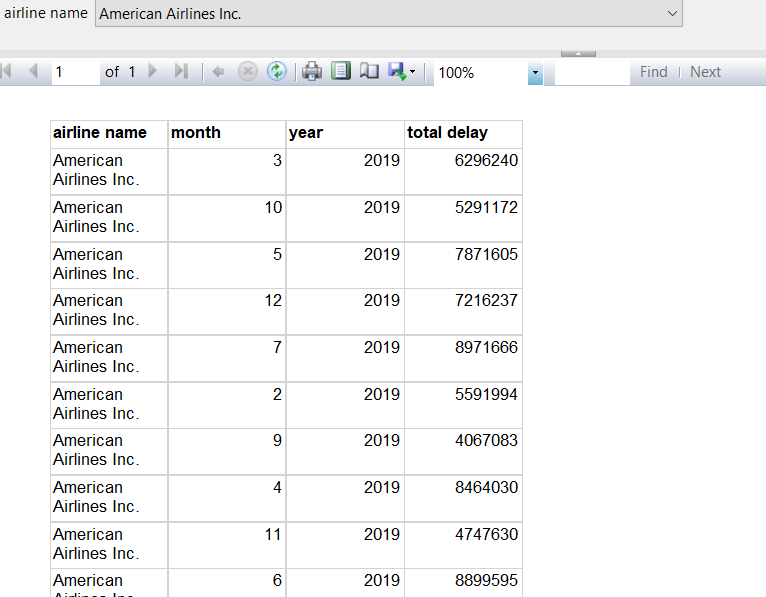
The following Reports show the aggregated value of the late aircraft delay per airline per month for the entire year. We have attached the Stored Procedure which was used to generate this report. This report can be used to get a detailed information about all the airlines operating within the five states. Alabama, New York, Florida, Pennsylvania and North Carolina. It also gives the information about the total distances operated by each airline based on the month and the year.

A screenshot of a cell phone

Description automatically generatedReport 3: Total Delay for a Particular Airline based on the input parameter.

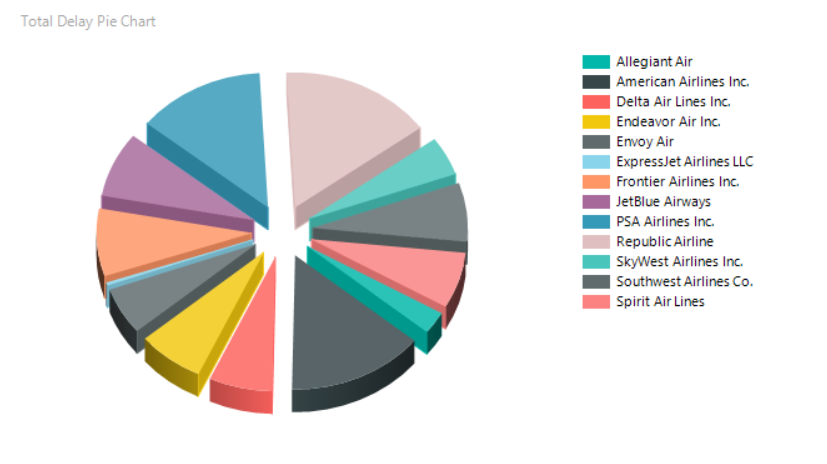
This Report is generated for providing the information about total delays happening in different airlines. The difference between Report 1 and Report 3 is that here we can provide an input parameter. In this Report we have taken the details about the American Airlines.

The input parameter is provided through a drop down which is populated dynamically by a sql query which is executed on the Airline table.



Report 4: Pie Chart showing Total Delay of the Different Airlines

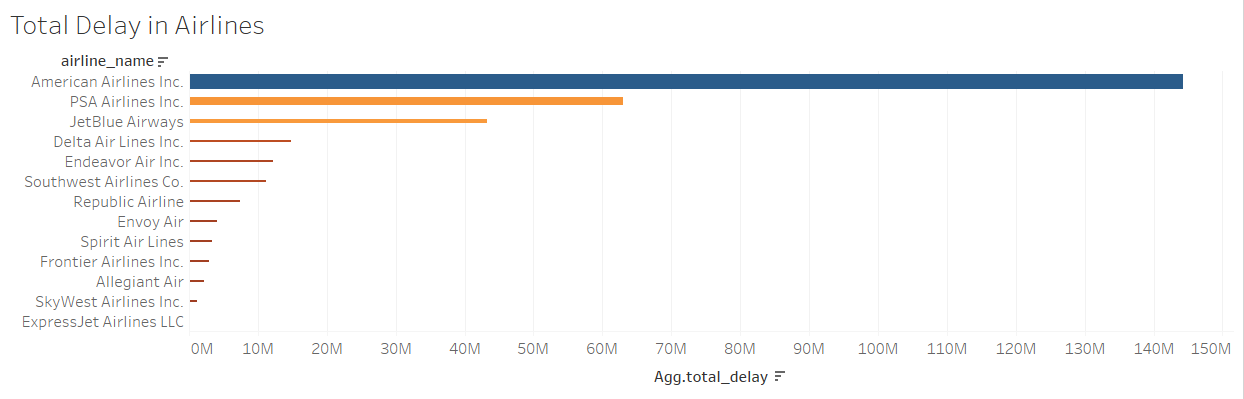
This Report was created to give a very easy pictorial representation for the total delays incureed by the different airline.



Airline Delay Reports using TABLEAU

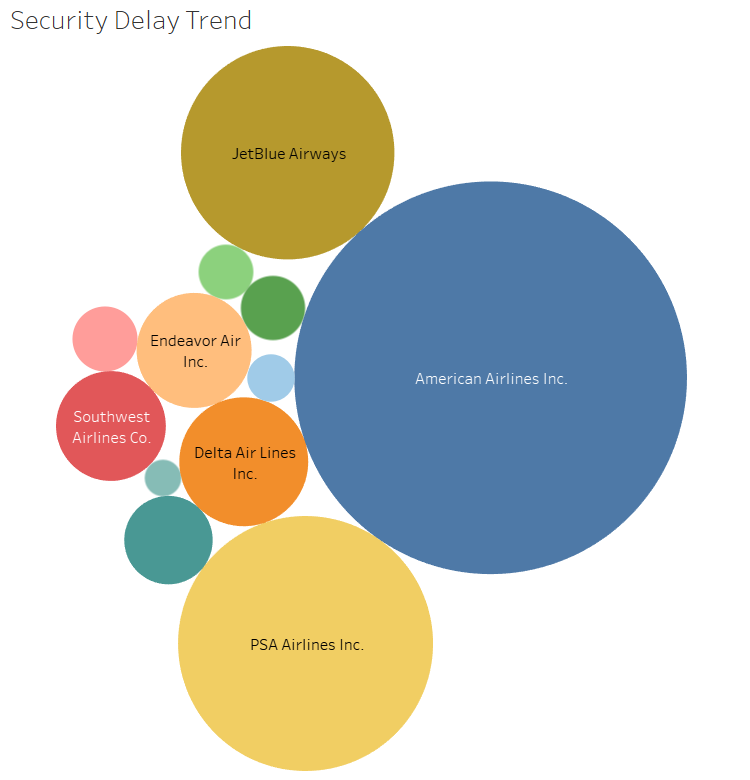
Report 1: Total Delay in Airlines

The total delay for the top airlines were visualized using a bar chart and sorted by Descending order. The width of the bar represents the late aircraft delay while the color of the bar represents the carrier delay. And from this chart it can be inferred that the American Airlines Inc. has the maximum delay in all the categories of delay across all the airports within our scope.

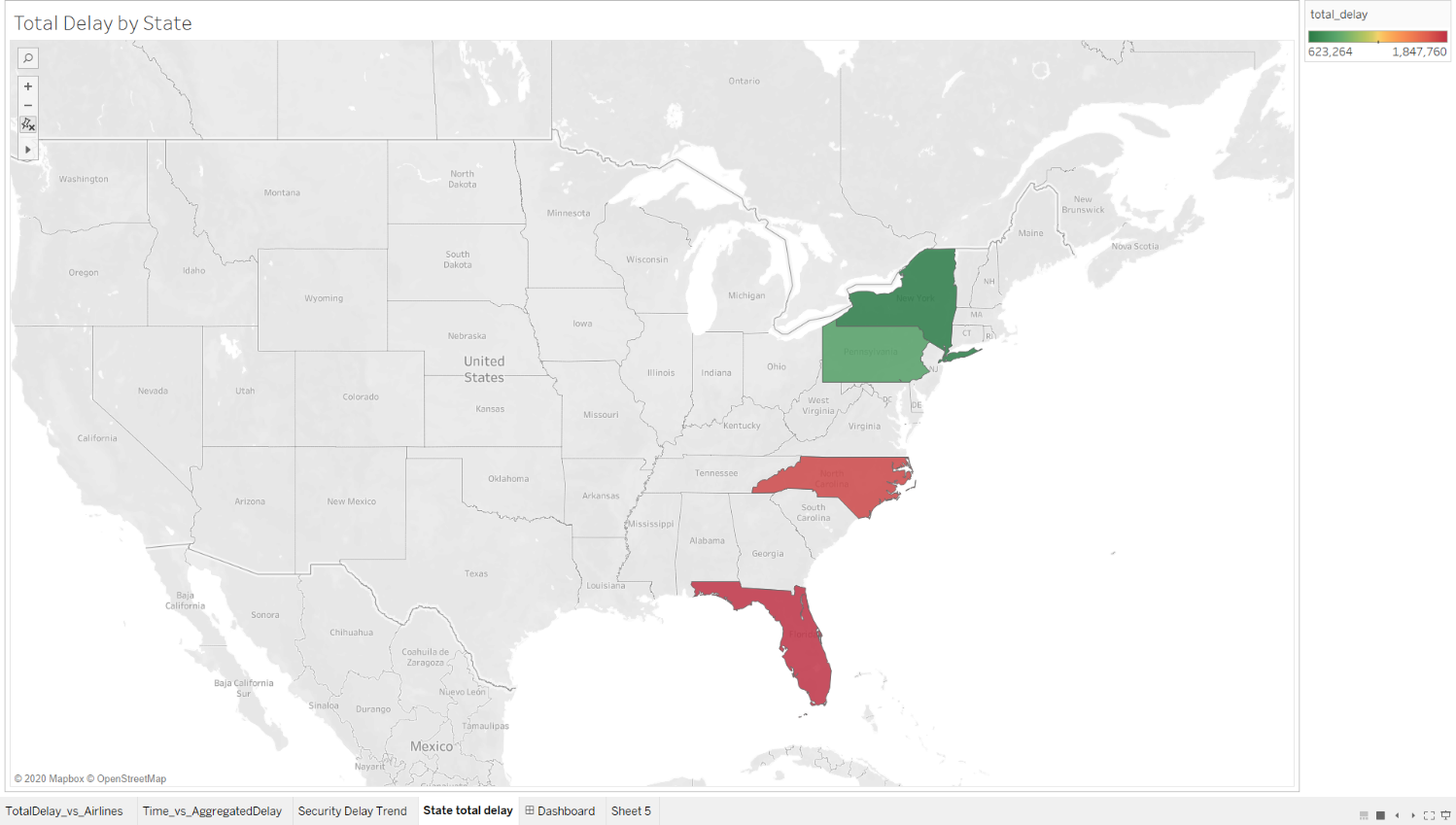


Report 2: Security Delay Trend

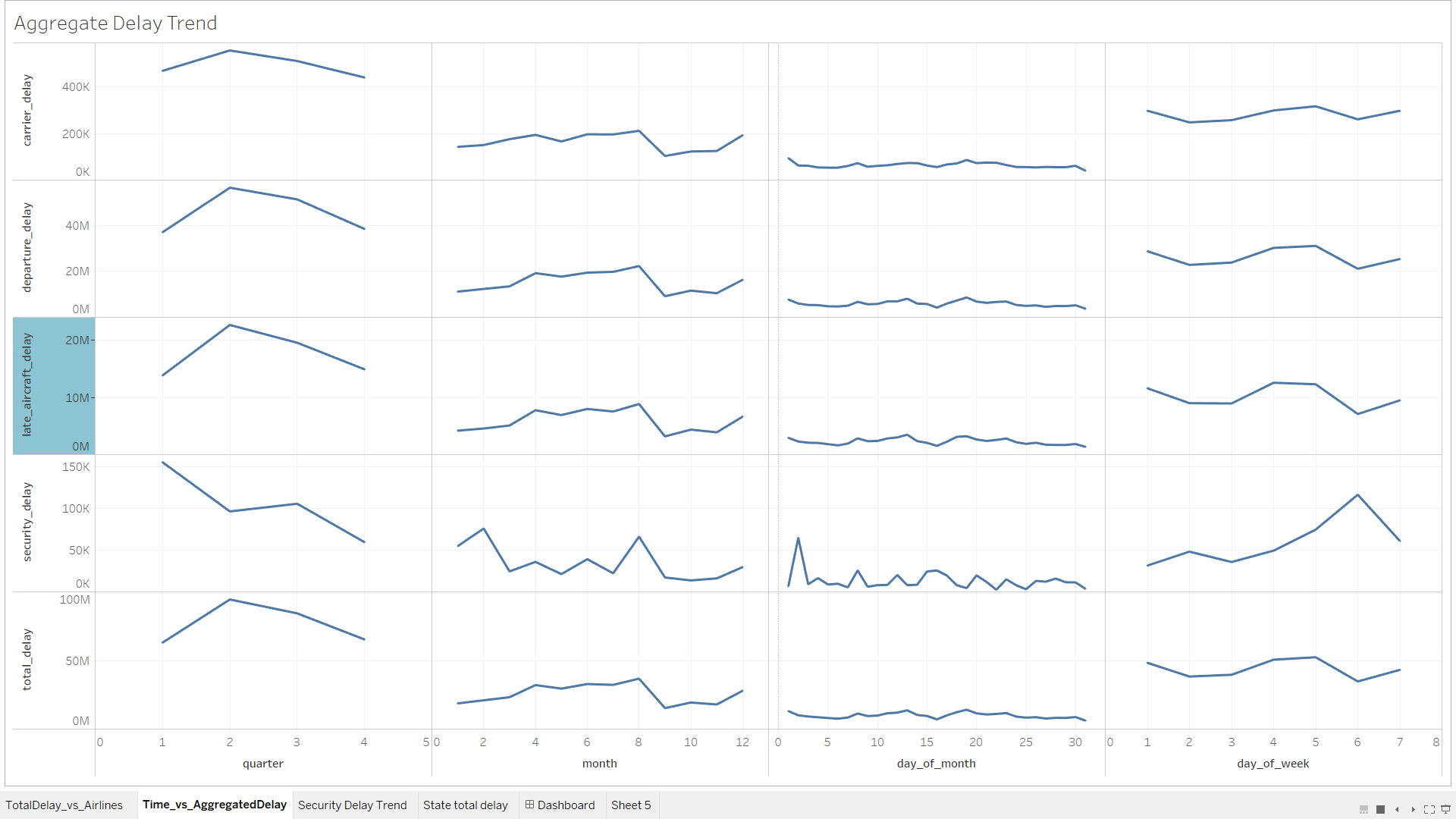
The airlines with more security delay were listed to identify the airlines which are most affected by them. Here also the American Airlines Inc. is the most affected airline. The second and the third Airline carrier which are affected by the security Delays are the PSA Airlines Inc. The advantage of this report is that we can find the data or the categories with the highest count or the highest aggregated sum with a single glance of the Report.



Report 3: Total delay by origin state:  
  
This heat map chart was implemented to visualize the total delay across the states where the airlines had their origin and identify the state wherein the overall delay is very high. Here it is evident that Florida is the state with highest total delay. From this we can understand that the airlines which are operating their flights with Florida as their origin state are having the highest delays..



Report 4: Aggregate delay Trend  
  
 In order to identify a pattern in the delay for all airlines across all the airports in our scope this chart was implemented at all possible time range. The following delay metrics were the observations across all the airlines and airports within our scope.  
  All the delays show a decline trend during the final quarter and have reached the maximum during the second quarter while the security delay alone had declined during the second quarterThe security delay is more at the weekend  
Their is an increasing trend in all the delays during the mid of the month.



6. APPENDIX

APPENDIX A : Data Mart Table Creation Scripts

Source Table.

CREATE TABLE [dbo].[delays](

[Id] [int] NOT NULL IDENTITY(1,1),

[year] [int] NOT NULL,

[quarter] [int] NOT NULL,

[month] [int] NOT NULL,

[day\_of\_month] [int] NOT NULL,

[day\_of\_week] [int] NOT NULL,

[flight\_date] [date] NOT NULL,

[unique\_carrier\_code] [varchar](15) NOT NULL,

[tail\_number] [varchar](15) NOT NULL,

[flight\_number] [int] NOT NULL,

[origin\_airport\_id] [int] NOT NULL,

[origin] [varchar](15) NOT NULL,

[origin\_city\_name] [varchar](50) NOT NULL,

[origin\_state\_abr] [varchar](15) NOT NULL,

[origin\_state\_name] [varchar](50) NOT NULL,

[dest\_airport\_id] [int] NOT NULL,

[dest] [varchar](15) NOT NULL,

[dest\_city\_name] [varchar](50) NOT NULL,

[dest\_state\_abr] [varchar](15) NOT NULL,

[dest\_state\_name] [varchar](50) NOT NULL,

[crs\_dept\_time] [varchar](15) NOT NULL,

[dept\_time] [varchar](15) NOT NULL,

[dept\_delay] [int] NOT NULL,

[taxi\_out] [int] NOT NULL,

[wheels\_off] [varchar](15) NOT NULL,

[wheels\_on] [varchar](15) NOT NULL,

[taxi\_in] [int] NOT NULL,

[crs\_arr\_time] [varchar](15) NOT NULL,

[arr\_time] [varchar](15) NOT NULL,

[arr\_delay] [int] NOT NULL,

[actual\_elapsed\_time] [int] NOT NULL,

[air\_time] [int] NOT NULL,

[distance] [int] NOT NULL,

[carrier\_delay] [int] DEFAULT 0,

[weather\_delay] [int] DEFAULT 0,

[nas\_delay] [int] DEFAULT 0,

[security\_delay] [int] DEFAULT 0,

[late\_aircraft\_delay] [int] DEFAULT 0

) ON [PRIMARY]

GO

Airline Dimension.

CREATE TABLE [dbo].[Airline\_Dim](

[airline\_code] [varchar](15) NOT NULL,

[airline\_name] [varchar](8000) NOT NULL

) ON [PRIMARY]

GO

Calendar Dimension.

CREATE TABLE [dbo].[Calendar\_Dim](

[date\_key] [int] NOT NULL IDENTITY(1,1),

[flight\_date] [date] NOT NULL,

[day\_of\_week] [int] NOT NULL,

[day\_of\_month] [int] NOT NULL,

[month] [int] NOT NULL,

[quarter] [int] NOT NULL,

[year] [int] NOT NULL

) ON [PRIMARY]

GO

Airport Dimension.

CREATE TABLE [dbo].[Airport\_Dim](

[airport\_id] [int] NOT NULL,

[airport\_code] [varchar](15) NOT NULL,

[city\_name] [varchar](50) NOT NULL,

[state\_abr] [varchar](15) NOT NULL,

[state\_name] [varchar](50) NOT NULL

) ON [PRIMARY]

GO

Journey Dimension.

CREATE TABLE [dbo].[Journey\_Dim](

[journey\_key] [int] NOT NULL IDENTITY(1,1),

[journey\_id] [varchar](100) NOT NULL,

[origin\_airport\_id] [int] NOT NULL,

[dest\_airport\_id] [int] NOT NULL,

[unique\_carrier\_code] [varchar](15) NOT NULL,

[distance] [int] NOT NULL

) ON [PRIMARY]

GO

Journey Delay Fact Table.

CREATE TABLE [dbo].[Journey\_Delay\_Fact](

[date\_key] [int] NOT NULL,

[journey\_key] [int] NOT NULL,

[airline\_id] [varchar](50) NOT NULL,

[flight\_number] [int] NOT NULL,

[departure\_delay] [int] NOT NULL,

[arrival\_delay] [int] NOT NULL,

[carrier\_delay] [int] NOT NULL,

[weather\_delay] [int] NOT NULL,

[nas\_delay] [int] NOT NULL,

[security\_delay] [int] NOT NULL,

[late\_aircraft\_delay] [int] NOT NULL,

[total\_delay] [int] NOT NULL

) ON [PRIMARY]

Aggregate Delay Fact Table.

CREATE TABLE [dbo].[Aggregate\_Delay\_Fact](

[date\_key] [int] NOT NULL,

[origin\_airport\_id] [int] NOT NULL,

[airline\_id] [varchar](50) NOT NULL,

[departure\_delay] [int] NOT NULL,

[carrier\_delay] [int] NOT NULL,

[security\_delay] [int] NOT NULL,

[late\_aircraft\_delay] [int] NOT NULL,

[total\_delay] [int] NOT NULL

) ON [PRIMARY]

GO

APPENDIX B: Stored procedures for SSRS Reports

Delay for every KM Report

CREATE PROC [dbo].[Delay\_per\_Distance\_Report]

AS

SELECT a.airline\_name, c.month, c.year,

SUM(ad.total\_delay) as total\_delay, SUM(ad.late\_aircraft\_delay) as late\_aircraft\_delay ,

SUM(j.distance) AS [total\_distance]

FROM dbo.Journey\_Dim j INNER JOIN Airline\_Dim a

ON j.unique\_carrier\_code = a.airline\_code INNER JOIN Aggregate\_Delay\_Fact ad

ON j.origin\_airport\_id = ad.origin\_airport\_id

INNER JOIN Calendar\_Dim c

ON ad.date\_key = c.date\_key

GROUP BY a.airline\_name, c.month, c.year

GO

Delay for every Airline

Drop Delay\_per\_Airline\_based\_on\_Delay\_Type\_Report

CREATE PROC [dbo].[Delay\_per\_Airline\_based\_on\_Delay\_Type\_Report]

@airline\_name varchar(8000)

AS

SELECT a.airline\_name, c.month, c.year,

SUM(ad.total\_delay) as total\_delay, SUM(ad.late\_aircraft\_delay) as late\_aircraft\_delay ,

SUM(security\_delay)as security\_delay,SUM(departure\_delay) as departure\_delay,

SUM(carrier\_delay) as carrier\_delay,SUM(j.distance) AS [total\_distance]

FROM dbo.Journey\_Dim j INNER JOIN Airline\_Dim a

ON j.unique\_carrier\_code = a.airline\_code INNER JOIN Aggregate\_Delay\_Fact ad

ON j.origin\_airport\_id = ad.origin\_airport\_id

INNER JOIN Calendar\_Dim c

ON ad.date\_key = c.date\_key where a.airline\_name = @airline\_name

GROUP BY a.airline\_name, c.month, c.year

GO

Delay Grouped by different cities.

CREATE PROC [dbo].[Departure\_Delay\_per\_city]

@city\_name varchar(8000)

AS

SELECT b.city\_name,a.departure\_delay from Aggregate\_Delay\_Fact a inner join Airport\_Dim b

on a.origin\_airport\_id = b.airport\_id where b.city\_name LIKE @city\_name

APPENDIX C : Create Table for neo4j Database.

Airline Node

LOAD CSV WITH HEADERS FROM "file:///Airline\_tbl.csv"  as row CREATE (a:Airline) SET a = row{airline\_code:row.airline\_code,airline\_name:row.airline\_name} return a

**Airport Node**

LOAD CSV WITH HEADERS FROM “file:///Airport\_tbl.csv”  as row CREATE (a:Airport) SET a = row{airport\_id:row.airport\_id,airport\_code:row.airport\_code,city\_name:row.city\_name,state\_abr:row.state\_abr,state\_name:row.state\_name} return a

**Journey Node**

LOAD CSV WITH HEADERS FROM “file:///Journey\_tbl.csv”  as row CREATE (j:Journey) SET j = row{journey\_key:row.journey\_key, flight\_date:row.flight\_date, flight\_number:row.flight\_number, tail\_number:row.tail\_number, origin\_airport\_id:row.origin\_airport\_id, dest\_airport\_id:row.dest\_aiport\_id, airline\_code:row.airline\_code, distance:row.distance} return j

**Journey Delay Node**

LOAD CSV WITH HEADERS FROM "file:///Journey\_delay\_tbl.csv" as row CREATE (j:JourneyDelay) SET j = row{journey\_key:row.journey\_key, departure\_delay:row.departure\_delay, arrival\_delay:row.arrival\_delay, carrier\_delay:row.carrier\_delay, weather\_delay:row.weather\_delay, nas\_delay:row.nas\_delay, security\_delay:row.security\_delay, late\_aircraft\_delay:row.late\_aircraft\_delay, total\_delay:row.total\_delay} return j

**CONSTRAINTS**

CREATE CONSTRAINT ON (a:Airline) ASSERT a.airline\_code is UNIQUE

CREATE CONSTRAINT ON (a:Airport) ASSERT a.airport\_id  is UNIQUE

CREATE CONSTRAINT ON (j:Journey) ASSERT j.journey\_key  is UNIQUE

CREATE CONSTRAINT ON (j:JourneyDelay) ASSERT j.journey\_key  is UNIQUE

**RELATIONSHIPS**

Relation between journey and journey delay

MATCH (j:Journey),(jd:JourneyDelay) where j.journey\_key = jd.journey\_key create (jd) -[r:part\_of]->(j)  return jd,j,r

Relation between journey and airline

MATCH (j:Journey),(a:airline) where j.airline\_code = a.airline\_code create (j) -[r:facilitated\_by]->(a)  return j,a,r

Relation between journey and airport

MATCH (j:Journey),(a:Airport) where j.origin\_airport\_id = a.airport\_id create (j) -[r:happens\_through]->(a)  return j,a,r

APPENDIX D : Create Table for Relational Database.

USE [flight\_delay\_relation]

GO

Create Airline Table

CREATE TABLE [dbo].[Airline\_tbl](

[airline\_code] [varchar](15) NOT NULL,

[airline\_name] [varchar](8000) NOT NULL,

PRIMARY KEY (airline\_code)

) ON [PRIMARY]

GO

Create Airport Table

CREATE TABLE [dbo].[Airport\_tbl](

[airport\_id] [int] NOT NULL,

[airport\_code] [varchar](15) NOT NULL,

[city\_name] [varchar](50) NOT NULL,

[state\_abr] [varchar](15) NOT NULL,

[state\_name] [varchar](50) NOT NULL,

PRIMARY KEY (airport\_id)

) ON [PRIMARY]

GO

Create Journey Table

CREATE TABLE [dbo].[Journey\_tbl](

[journey\_key] [int] IDENTITY(1,1) NOT NULL,

[flight\_date] [date] NOT NULL,

[flight\_number] [int] NOT NULL,

[tail\_number] [varchar](15) NOT NULL,

[origin\_airport\_id] [int] NOT NULL,

[dest\_airport\_id] [int] NOT NULL,

[airline\_code] [varchar](15) NOT NULL,

[distance] [int] NOT NULL,

PRIMARY KEY (journey\_key),

CONSTRAINT dt\_origin\_airport\_id\_fk FOREIGN KEY (origin\_airport\_id) REFERENCES Airport\_tbl (airport\_id),

CONSTRAINT dt\_dest\_airport\_id\_fk FOREIGN KEY (dest\_airport\_id) REFERENCES Airport\_tbl (airport\_id),

CONSTRAINT dt\_airline\_code\_fk FOREIGN KEY (airline\_code) REFERENCES Airline\_tbl (airline\_code)

) ON [PRIMARY]

GO

Create Journey Delay Table

drop table dbo.Journey\_Delay\_tbl

CREATE TABLE [dbo].[Journey\_Delay\_tbl](

[journey\_key] [int] IDENTITY(1,1) NOT NULL,

[departure\_delay] [int] NOT NULL,

[arrival\_delay] [int] NOT NULL,

[carrier\_delay] [int] NOT NULL,

[weather\_delay] [int] NOT NULL,

[nas\_delay] [int] NOT NULL,

[security\_delay] [int] NOT NULL,

[late\_aircraft\_delay] [int] NOT NULL,

[total\_delay] [int] NOT NULL

PRIMARY KEY (journey\_key),

CONSTRAINT dt\_journey\_key\_fk FOREIGN KEY (journey\_key) REFERENCES Journey\_tbl (journey\_key)

) ON [PRIMARY]

GO

7. **Referencing and Bibliography**

*ETL (Extract, Transform, and Load) Process* (no date). Available at: https://www.guru99.com/etl-extract-load-process.html#8 (Accessed: 12 April 2020).

Ralph,Kimball and Margy,Ross (2013) *The Data Warehouse Toolkit: 9781118530771*. Wiley. Available at: https://www.dawsonera.com:443/abstract/9781118530771.

Robinson, I., Webber, J. and Eifrem, E. (2015) *Graph Databases: New Opportunities for Connected Data*. O’Reilly Media. Available at: https://www.dawsonera.com:443/abstract/9781491930861.

Srivastava, A. (2019) *Dimensional Data Modeling*, *Medium*. Available at: https://towardsdatascience.com/dimensional-data-modeling-49038b96d95a (Accessed: 13 April 2020).

Thakrani, S. (2019) *FLIGHT DELAY ANALYSIS*, *Medium*. Available at: https://medium.com/@saurabhthakrani/analyzing-flight-delays-d8cba808bcd3 (Accessed: 13 April 2020).

*The True Value of a Business Data Warehouse* (no date) *Business 2 Community*. Available at: https://www.business2community.com/business-intelligence/true-value-business-data-warehouse-01810395 (Accessed: 13 April 2020).