Advanced Data Management

Project Report

Jyoti Dhiman(4853961)

Ramish Majeed(4853334)

Owais Ghafoor Malik (4871516)

1. PART -1

1.1. Dataset

The dataset contains performance of the airlines operating in the US for the period January 2016. The datasetis available here. The dataset exists as a relational database with:

- Size: 455.5 MB
- Count of tables: 19
- Count of rows: 448,156

1.2. Nature of proposed application

The data consists of a lot of attributes related to the airline like flight number, destination, flight delay time, arrival delay time, departure delay time, etc. We have to do batch processing to load the data and it will be read intensive as the given data is complex and we have to keep in mind that it is dynamic, and it keeps updating.

1.3. Partitioning and replication

As we are using Neo4j we do not need partitioning, we can work with application-level partitioning.

Replication: Neo4j Graph database system relies on Core nodes &read replicas which ensures Causal consistency.

1.4. The NO-SQL System

We will work on the graph database using Neo4J, because the topology of the graph database is more dynamic, it would help with analysis applications. As we will see, the workload queries require a lot of join operation to get the required results, so the best way to do it is to work with graph-oriented system.
Also, the selected data set contains complex relationships between its entities and these relations have semantic and meaningful values, which is preserved by the graph database

1.5. Workload

1. Average Arrival Delay
2. Average Departure Delay
3. Distinct Airlines in Dataset with Occurrence

4. Delay Performance by Airline
5. Cancellations
6. Flight Cancellation Rate
7. Flight Cancellation Rate (by Airline)
8. Unique Destinations (by Airline)
9. Busiest Destinations daily Flight count
10. Unique Destinations (by Airline)
11. Busiest Destinations
12. Daily Flight Count
13. Most Visited Destination (Specific)
14. Most Visited Destinations (by Airlines)
15. Longest Flight
16. Top 10 Destinations for Airline (Specific)
17. Most Delayed Flight by Airline
18. Delay Counts for Flights
19. Most flight delays for Destination
20. Weather delayed flights for specific airline on specific date
21. Delayed flights for airlines on specific date
22. Weather delayed flights for specific airline on specific date
23. Airline Connectedness
24. Destination Connectedness

1.6. Main entities and relationships

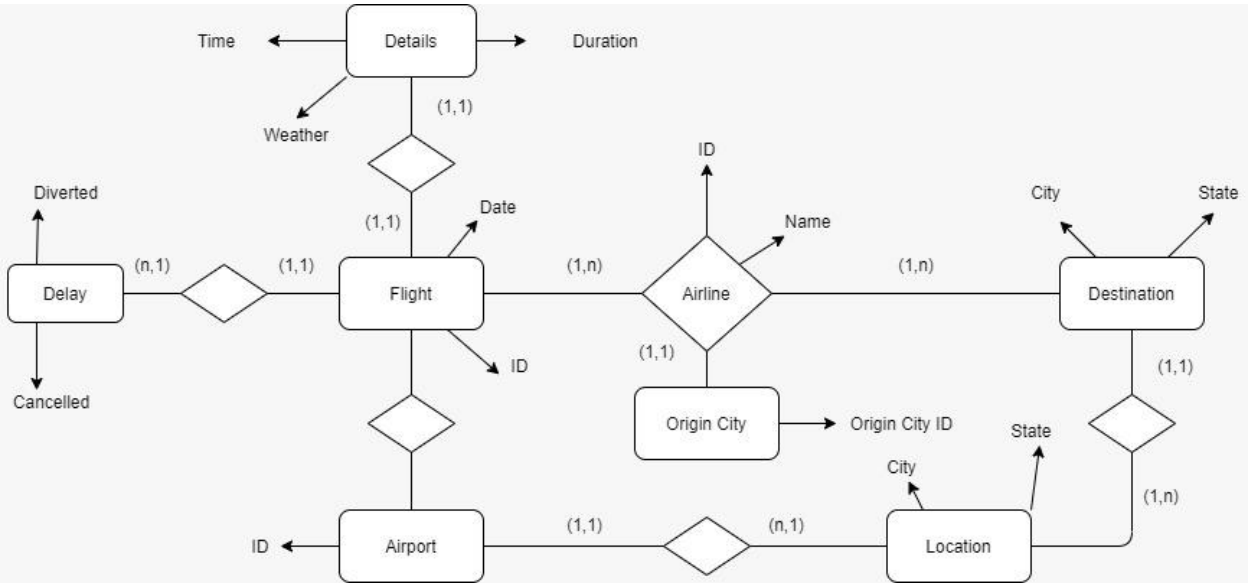
- Flight number
- Airlines
- Arrival Delay
- Departure Delay
- Date of a specific flight
- Weather
- Destination
- Location

Relationship between them:

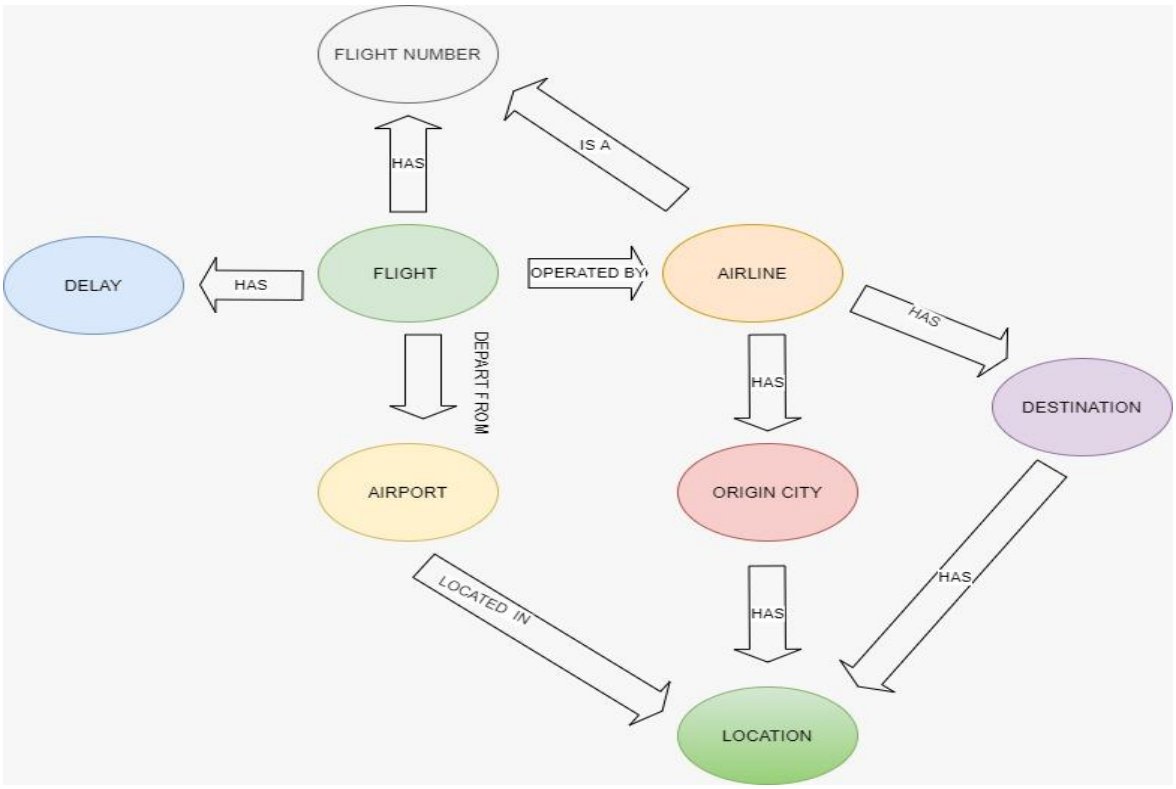
- Airline has a departure city
- Flight has an arrival delay
- Flight has a departure delay
- Weather affecting flights
- Duration of a flight
- Description of airline

2. PART -2

2.1.Conceptual Schema



2.2.Logical Schema



2.3.Database Setup

Installation

Neo4j desktop client was used for this project. After installing the software, A DBMS needs to be created where the database would reside. Ideally, Neo4j's ETL tool could have been used to import the database fromRDBMS to our new DBMS, but the ETL tool was buggy and did not work as intended. As a workaround, we exported all the tables of the database from the RDBMS as CSV files. Before the files are imported, they needto be made accessible by placing them in DBMS' import folder.

Constraints

Applying constraints before importing the data ensures that no duplicate values are entered. Additionally, an index is also created automatically which improves query speed. We applied constraints to all tables using thecode below (only shown for one table)

```
CREATE CONSTRAINT ON (airlineID:AirlineID) ASSERT airlineID.code IS UNIQUE
```

Data Import

Data import was straightforward via the LOAD CSV command. All the rows were converted to nodes withlabels assigned using the create command.
Since the files were MS-Excel version of CSV, field terminators need to be assigned properly beforeimporting.
The dataset we used was originally a relational database with a relatively large transaction table. The importcommand would run out of memory. We used periodic commit to ensure memory issues were resolved.

```
:auto USING PERIODIC COMMIT 10000

LOAD CSV WITH HEADERS FROM 'file:///airlineID.csv' AS row
FIELDTERMINATOR ';'

CREATE(airlineID:AirlineID)SET
airlineID += row
```

Create New Node Labels

Since there was a monolithic transaction table, it proved helpful to create new nodes using selective columnsas fields.

```
MATCH (p:Performance)
with DISTINCT p.AirlineID AS AirlineID, p.Dest AS dst
CREATE (d:AirlineRouteDst)
SET d.airlineID = AirlineID,d.dst = dst
Return COUNT(d)

MATCH (p:Performance)
```

```
with DISTINCT p.AirlineID AS AirlineID, p.Origin AS src
CREATE (d:AirlineRouteSrc)
SET d.airlineID = AirlineID,d.src = src
Return COUNT(d)
```

Create Relationships

As mentioned earlier, the dataset was originally stored in an RDBMS. Therefore, to make sense of the foreignkeys, we defined relationship between the nodes based on the foreign keys.

```
MATCH
  (a:AirlineID),
  (b:Performance)
WHERE a.code = b.AirlineID
CREATE (b)-[r:hasAirlineID]->(a)
RETURN type(r)
```

2.4.Queries

Once the database was set up, we were ready to answer some interesting questions pertaining to delays,cancellations, performance and connectedness.

1. Average Arrival Delay

```
MATCH (n:Performance)
WITH toInteger(n.ArrDelayMinutes) as val
WHERE val <> 0
RETURN toString(round(avg(val),2))+" minutes"
```

```
32.15 minutes
```

2. Average Departure Delay

```
MATCH (n:Performance)
WITH toInteger(n.DepDelayMinutes) as val
WHERE val <> 0
RETURN toString(round(avg(val),2))+" minutes"
```

```
31.38 minutes
```

3. Distinct Airlines in Dataset with Occurrence

```
MATCH p=(n:Performance)-[r:withAirline]->(x)
RETURN x.description AS AIRLINE, count(*) AS COUNT
ORDER BY COUNT DESC
```

"AIRLINE"	"COUNT"
"Southwest Airlines Co.: WN"	104154
"American Airlines Inc.: AA"	75580
"Delta Air Lines Inc.: DL"	69711
"SkyWest Airlines Inc.: OO"	47619
"ExpressJet Airlines Inc.: EV"	41970
"United Air Lines Inc.: UA"	39761
"JetBlue Airways: B6"	23018
"Alaska Airlines Inc.: AS"	14205
"Spirit Air Lines: NK"	11047
"Frontier Airlines Inc.: F9"	7099
"Hawaiian Airlines Inc.: HA"	6279
"Virgin America: VX"	5384

4. Delay Performance by Airline

```
MATCH p=(n)-[r:withAirline]->(x)
WITH toInteger(n.ArrDelayMinutes) AS arrDelay,toInteger(n.DepDelayMinutes) AS
depDelay, x.description AS des
RETURN des as Airline, round(avg(arrDelay),2) as ArrivalDelay,
round(avg(depDelay),2) as DepartureDelay
ORDER BY Airline ASC
```

"Airline"	"ArrivalDelay"	"DepartureDelay"	"TotalFlights"
"Southwest Airlines Co.: WN"	7.57	8.64	104154
"American Airlines Inc.: AA"	10.7	10.28	75580
"Delta Air Lines Inc.: DL"	9.46	9.8	69711
"SkyWest Airlines Inc.: OO"	15.17	14.03	47619
"ExpressJet Airlines Inc.: EV"	10.92	10.38	41970
"United Air Lines Inc.: UA"	9.67	11.94	39761
"JetBlue Airways: B6"	17.28	17.37	23018
"Alaska Airlines Inc.: AS"	6.98	6.57	14205
"Spirit Air Lines: NK"	18.63	17.57	11047
"Frontier Airlines Inc.: F9"	10.65	11.23	7099
"Hawaiian Airlines Inc.: HA"	4.18	2.7	6279
"Virgin America: VX"	15.66	15.87	5384

5. Cancellations

```
MATCH p = (a)-[r:hasCancellation]-(b)
WHERE a.CancellationCode IS NOT NULL
RETURN DISTINCT b.description AS Cancellation, count(*) AS Count
```

"Cancellation"	"Count"
"Carrier"	1423
"Weather"	9054
"National Air System"	1188

6. Flight Cancellation Rate

```
MATCH q = (c:Performance)
WITH toFloat(COUNT(c)) AS TOTAL

MATCH p = (a)-[r:hasCancellation]-(b)
WHERE a.CancellationCode IS NOT NULL

RETURN toString("Flight cancellation rate is
"+ROUND(toFloat(count(*)*100)/TOTAL,2)+"%")
```

```
Flight cancellation rate is 2.62%
```

7. Flight Cancellation Rate (by Airline)

```
MATCH q = (d)-[s:withAirline]->(e)
WITH toFloat(COUNT(q)) as TOTAL, e.description AS AL

MATCH p = (a)<-[r:hasCancellation]-(b)-[s:withAirline]->(c)
WITH toFloat(COUNT(*)) AS SUBTOTAL,c.description AS Airline,TOTAL, AL
WHERE AL = Airline

RETURN Airline,SUBTOTAL,TOTAL,ROUND(SUBTOTAL*100/TOTAL,3) AS CancellationRate
ORDER BY CancellationRate
```

"Airline"	"SUBTOTAL"	"TOTAL"	"CancellationRate"
"Hawaiian Airlines Inc.: HA"	4.0	6279.0	0.064
"Alaska Airlines Inc.: AS"	139.0	14205.0	0.979
"Frontier Airlines Inc.: F9"	77.0	7099.0	1.085
"Delta Air Lines Inc.: DL"	974.0	69711.0	1.397
"SkyWest Airlines Inc.: OO"	984.0	47619.0	2.066
"Southwest Airlines Co.: WN"	2640.0	104154.0	2.535
"Spirit Air Lines: NK"	308.0	11047.0	2.788
"Virgin America: VX"	159.0	5384.0	2.953
"United Air Lines Inc.: UA"	1336.0	39761.0	3.36
"ExpressJet Airlines Inc.: EV"	1427.0	41970.0	3.4
"American Airlines Inc.: AA"	2720.0	75580.0	3.599
"JetBlue Airways: B6"	897.0	23018.0	3.897

8. Unique Destinations (by Airline)

```
MATCH (p:Performance)-[r:withAirline]-(q:AirlineID)
RETURN q.description AS Airline,COUNT(DISTINCT p.Dest) AS Destinations
ORDER BY Destinations DESC
```

"Airline"	"Destinations"
"SkyWest Airlines Inc.: OO"	183
"ExpressJet Airlines Inc.: EV"	163
"Delta Air Lines Inc.: DL"	148
"American Airlines Inc.: AA"	95
"Southwest Airlines Co.: WN"	86
"United Air Lines Inc.: UA"	86
"Alaska Airlines Inc.: AS"	64
"JetBlue Airways: B6"	62
"Frontier Airlines Inc.: F9"	50
"Spirit Air Lines: NK"	33
"Virgin America: VX"	20
"Hawaiian Airlines Inc.: HA"	17

9. Busiest Destinations

```
MATCH (p:Performance)
WITH p.Dest AS Destination, p.FlightDate AS Date, COUNT(*) AS FlightsPerDay
RETURN Destination, ROUND(AVG(FlightsPerDay)) AS AvgFlightsPerDay
ORDER BY AvgFlightsPerDay DESC LIMIT 10
```

"Destination"	"AvgFlightsPerDay"
"ATL"	963.0
"ORD"	600.0
"DEN"	565.0
"DFW"	535.0
"LAX"	529.0
"SFO"	426.0
"PHX"	420.0
"LAS"	395.0
"IAH"	376.0
"MCO"	346.0

10. Daily Flight Count

```
MATCH (p:Performance)
RETURN date(p.FlightDate) AS Date, COUNT(*) AS FlightsPerDay
ORDER BY Date ASC
```

"Date"	"FlightsPerDay"
"2016-01-01"	13019
"2016-01-02"	14869
"2016-01-03"	15878
"2016-01-04"	15570
"2016-01-05"	14582
"2016-01-06"	14683
"2016-01-07"	15193
"2016-01-08"	15228
"2016-01-09"	11791
"2016-01-10"	13988
"2016-01-11"	15174
"2016-01-12"	14566
"2016-01-13"	14800
"2016-01-14"	15295
"2016-01-15"	15308
"2016-01-16"	11563
"2016-01-17"	12970
"2016-01-18"	15107
"2016-01-19"	14580
"2016-01-20"	14790
"2016-01-21"	15285
"2016-01-22"	15290

"2016-01-23"	11732
"2016-01-24"	14001
"2016-01-25"	15177
"2016-01-26"	14545
"2016-01-27"	14763
"2016-01-28"	15271
"2016-01-29"	15293
"2016-01-30"	11699
"2016-01-31"	13817

11. Most Visited Destination (Specific)

```
MATCH (p:Performance)-[r:withAirline]-(a:AirlineID)
WHERE a.description = "SkyWest Airlines Inc.: OO"
RETURN  a.description AS Airline, p.DestCityName AS Destination, COUNT(*) AS Count
ORDER BY Count DESC LIMIT 1
```

"Airline"	"Destination"	"Count"
"SkyWest Airlines Inc.: OO"	"Denver, CO"	4236

12. Most Visited Destination (By Airlines)

```
MATCH (p:Performance)-[r:withAirline]-(a:AirlineID)
WITH  a, p.DestCityName AS Destination, COUNT(*) AS Count
ORDER BY Count DESC
RETURN a.description AS Airline, collect(Destination)[0] AS Destination ,
MAX(Count)
```

"Airline"	"Destination"	"max(Count)"
"Delta Air Lines Inc.: DL"	"Atlanta, GA"	19115
"American Airlines Inc.: AA"	"Dallas/Fort Worth, TX"	11956
"Southwest Airlines Co.: WN"	"Chicago, IL"	6334
"United Air Lines Inc.: UA"	"Houston, TX"	4770
"ExpressJet Airlines Inc.: EV"	"Atlanta, GA"	4690
"Alaska Airlines Inc.: AS"	"Seattle, WA"	4444
"SkyWest Airlines Inc.: OO"	"Denver, CO"	4236
"JetBlue Airways: B6"	"New York, NY"	4203
"Hawaiian Airlines Inc.: HA"	"Honolulu, HI"	2706
"Frontier Airlines Inc.: F9"	"Denver, CO"	1753
"Virgin America: VX"	"San Francisco, CA"	1542
"Spirit Air Lines: NK"	"Fort Lauderdale, FL"	1178

13. Longest Flight

```
MATCH (p:Performance)-[r:withAirline]-(a:AirlineID)
RETURN p.OriginCityName AS Origin,p.DestCityName AS Destination,a.description AS
Airline,MAX(p.Distance) AS Distance
ORDER BY Distance DESC LIMIT 1
```

```
╒═══════════════════════╤═══════════════╤═══════════════════════════════╤══════════╕
│"Origin"               │"Destination"  │"Airline"                      │"Distance"│
╞═══════════════════════╪═══════════════╪═══════════════════════════════╪══════════╡
│"Salt Lake City, UT"   │"Dallas, TX"   │"Southwest Airlines Co.: WN"   │"999"     │
└───────────────────────┴───────────────┴───────────────────────────────┴──────────┘
```

14. Top 10 Destinations for Airline (Specific)

```
MATCH (p:Performance)-[r:withAirline]-(a:AirlineID)
WHERE a.description = "Southwest Airlines Co.: WN"
RETURN a.description AS Airline,p.DestCityName AS Destination,Count(*) AS Count
ORDER BY Count DESC LIMIT 10
```

"Airline"	"Destination"	"Count"
"Southwest Airlines Co.: WN"	"Chicago, IL"	6334
"Southwest Airlines Co.: WN"	"Las Vegas, NV"	6098
"Southwest Airlines Co.: WN"	"Baltimore, MD"	5395
"Southwest Airlines Co.: WN"	"Dallas, TX"	5205
"Southwest Airlines Co.: WN"	"Denver, CO"	5072
"Southwest Airlines Co.: WN"	"Phoenix, AZ"	4857
"Southwest Airlines Co.: WN"	"Houston, TX"	4248
"Southwest Airlines Co.: WN"	"Atlanta, GA"	3547
"Southwest Airlines Co.: WN"	"Orlando, FL"	3426
"Southwest Airlines Co.: WN"	"Los Angeles, CA"	3379

15. Most Delayed Flight by Airline

```
MATCH (p:Performance)-[r:withAirline]-(a:AirlineID)
WHERE p.DepDelayMinutes <> "\N" AND p.ArrDelayMinutes <> "\N"
RETURN a.description AS Airline,toInteger(p.DepDelayMinutes) +
toInteger(p.ArrDelayMinutes) AS MaxDelay
ORDER BY MaxDelay DESC LIMIT 1
```

"Airline"	"MaxDelay"
"American Airlines Inc.: AA"	3322

16. Delay Counts for Flights

```
MATCH (p:Performance)-[r:withAirline]-(a:AirlineID)
WHERE p.DepDelayMinutes <> "\N" AND p.ArrDelayMinutes <> "\N"
WITH a.description AS Airline,toInteger(p.DepDelayMinutes) +
toInteger(p.ArrDelayMinutes) AS Delay
WHERE Delay > 0
RETURN Airline,COUNT(Delay) as DelayCount
ORDER By DelayCount DESC
```

"Airline"	"DelayCount"
"Southwest Airlines Co.: WN"	46749
"American Airlines Inc.: AA"	32278
"Delta Air Lines Inc.: DL"	28113
"SkyWest Airlines Inc.: OO"	20425
"United Air Lines Inc.: UA"	17419
"ExpressJet Airlines Inc.: EV"	14912
"JetBlue Airways: B6"	12491
"Spirit Air Lines: NK"	6303
"Alaska Airlines Inc.: AS"	5225
"Virgin America: VX"	2753
"Frontier Airlines Inc.: F9"	2409
"Hawaiian Airlines Inc.: HA"	2167

17. Most flight delays for Destination

```
MATCH (p:Performance)-[r:withAirline]-(a:AirlineID)
WHERE p.DepDelayMinutes <> "\N" AND p.ArrDelayMinutes <> "\N"
WITH p.DestCityName AS Destination,toInteger(p.DepDelayMinutes) +
toInteger(p.ArrDelayMinutes) AS Delay
WHERE Delay > 0
RETURN Destination,COUNT(Delay) as DelayCount
ORDER By DelayCount DESC LIMIT 1
```

```
╒═══════════════╤═══════════════╕
│"Destination"  │"DelayCount"   │
╞═══════════════╪═══════════════╡
│"Atlanta, GA"  │11741          │
└───────────────┴───────────────┘
```

18. Weather delayed flights for specific airline on specific date

```
MATCH (n:Performance)-[r:withAirline]-(m)
WHERE n.WeatherDelay <> "\N" AND
n.FlightDate = "2016-01-10" AND
m.description = "American Airlines Inc.: AA"
RETURN m.description AS Airline, n.FlightDate AS Date,COUNT(n) AS Count
```

```
╒═══════════════════════════════╤═══════════════╤═══════╕
│"Airline"                      │"Date"         │"Count"│
╞═══════════════════════════════╪═══════════════╪═══════╡
│"American Airlines Inc.: AA"   │"2016-01-10"   │536    │
└───────────────────────────────┴───────────────┴───────┘
```

19. Delayed flights for airlines on specific date

```
MATCH (p:Performance)-[r:withAirline]-(m)
WHERE p.DepDelayMinutes <> "\N" AND p.ArrDelayMinutes <> "\N" AND p.FlightDate =
"2016-01-31"
```

```
WITH m.description as Airline,DATE(p.FlightDate) AS Date,COUNT(*) AS Count
ORDER BY Count DESC
RETURN Airline,Date, Count
```

"Airline"	"Date"	"Count"
"Southwest Airlines Co.: WN"	"2016-01-31"	3001
"American Airlines Inc.: AA"	"2016-01-31"	2413
"Delta Air Lines Inc.: DL"	"2016-01-31"	2146
"SkyWest Airlines Inc.: OO"	"2016-01-31"	1490
"ExpressJet Airlines Inc.: EV"	"2016-01-31"	1289
"United Air Lines Inc.: UA"	"2016-01-31"	1219
"JetBlue Airways: B6"	"2016-01-31"	728
"Alaska Airlines Inc.: AS"	"2016-01-31"	452
"Spirit Air Lines: NK"	"2016-01-31"	354
"Frontier Airlines Inc.: F9"	"2016-01-31"	217
"Hawaiian Airlines Inc.: HA"	"2016-01-31"	201
"Virgin America: VX"	"2016-01-31"	176

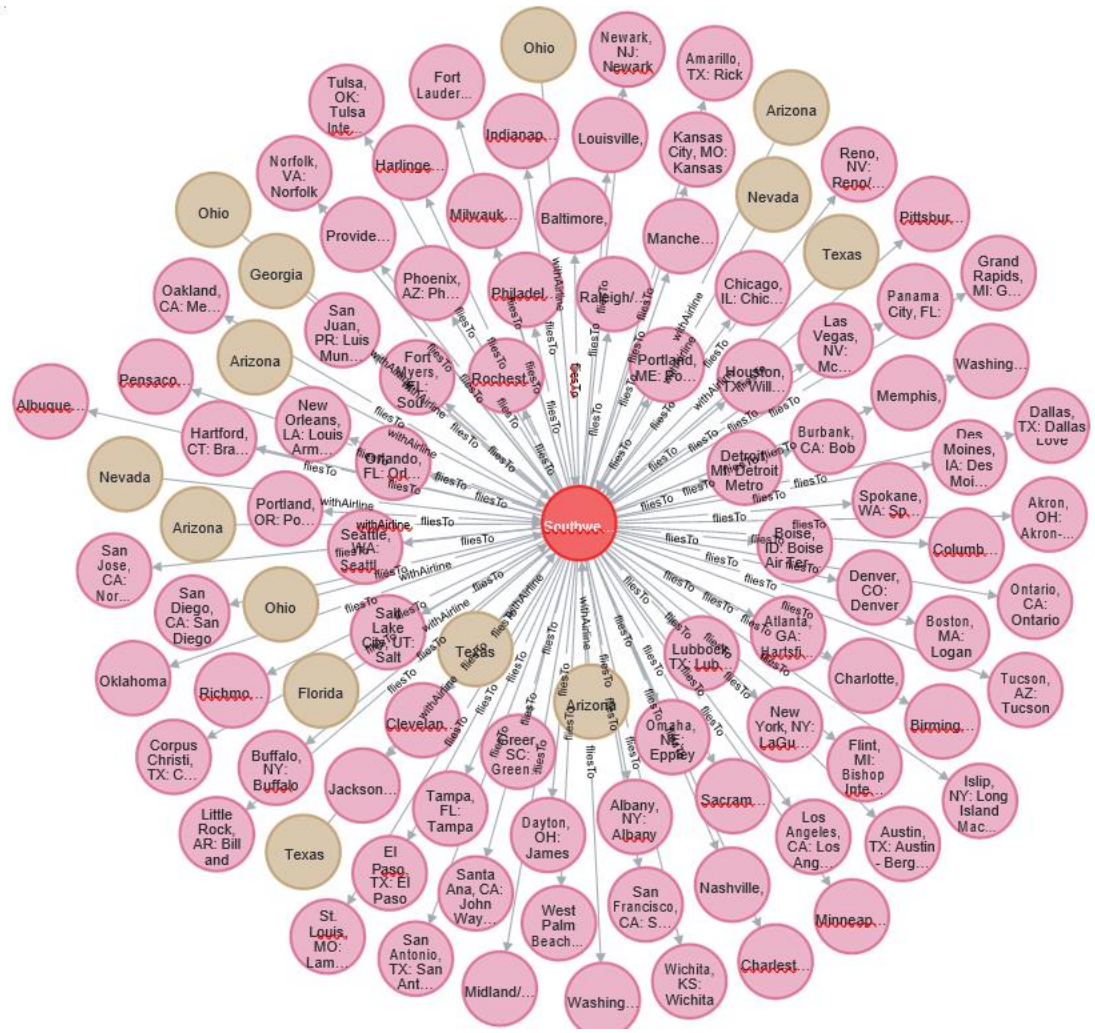
20. Weather delayed flights for specific airline on specific date

```
MATCH (n:Performance)-[r:withAirline]-(m)
WHERE n.WeatherDelay <> "\N" AND
n.FlightDate = "2016-01-10" AND
m.description = "American Airlines Inc.: AA"
RETURN m.description AS Airline, n.FlightDate AS Date,COUNT(n) AS Count
```

"Airline"	"Date"	"Count"
"American Airlines Inc.: AA"	"2016-01-10"	536

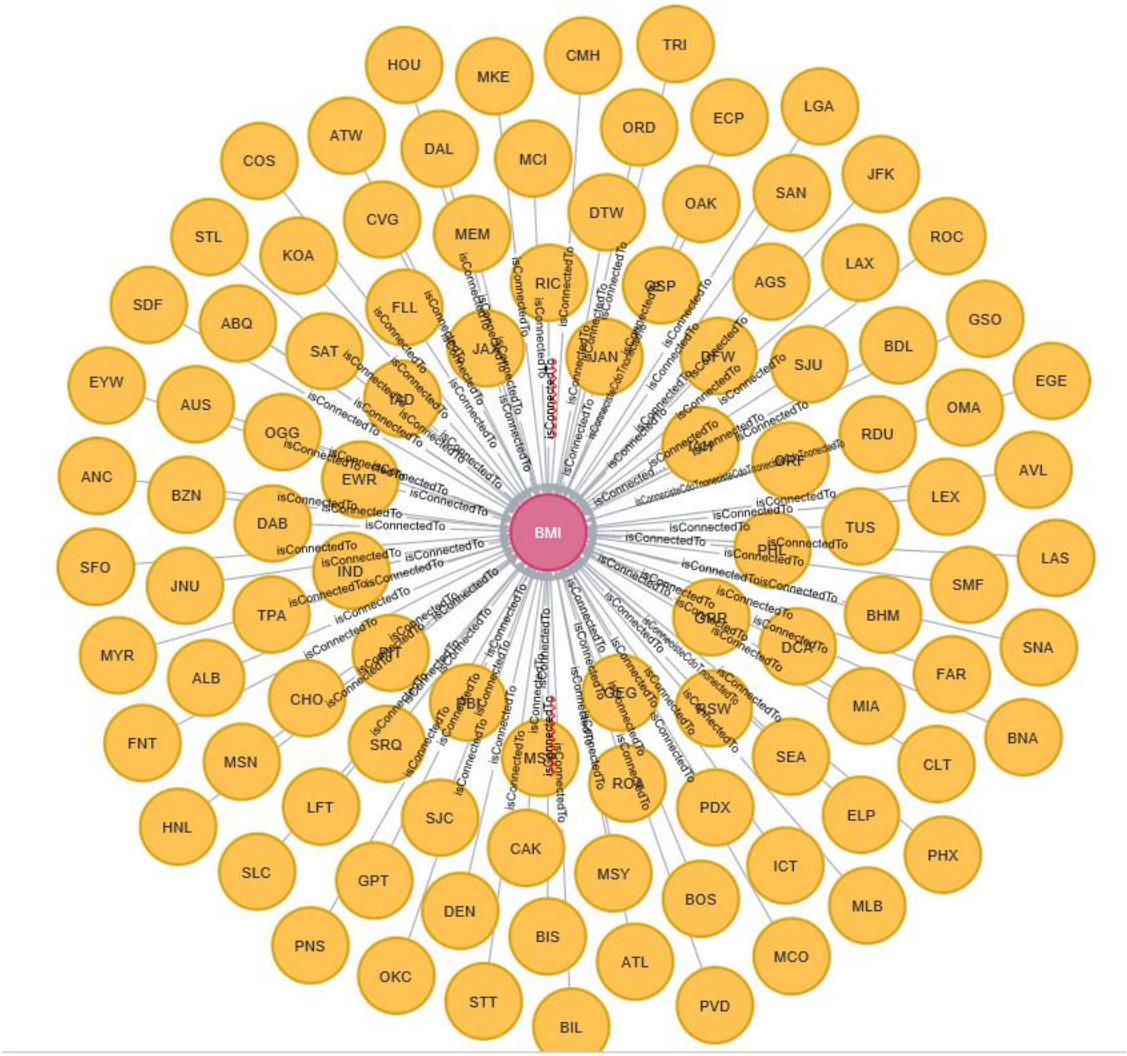
21. Airline Connectedness

```
MATCH (a:AirlineRouteSrc),
(b:AirlineRouteDst),
(c:Airports),
(d:AirlineID)
WHERE a.src = c.code AND a.airlineID = d.code AND b.dst = c.code AND b.airlineID =
d.code
CREATE (d)-[r2:fliesTo]->(c)
RETURN COUNT(*)

MATCH p= ()-[r:fliesTo]->() RETURN p LIMIT 1
```



22. Destination Connectedness

```
MATCH (a:AirlineRouteSrc),
(b:AirlineRouteDst)
WHERE a.airlineID = b.airlineID
CREATE (a)-[r2:isConnectedTo]->(b)
RETURN COUNT(*)

MATCH p= ()-[r:isConnectedTo]->() RETURN p LIMIT 1
```



Video link for the presentation of second part

https://drive.google.com/file/d/1_rCLAfuWA_JOwSEoUiqwU2Wiw4jwW3AO/view?usp=sharing

3. PART – 3(RDF, RDFS, and OWL)

```
# Namespace declarations

@prefix ex: <http://www.example.org/#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.

Properties explained with domain and range

ex:hasBeenClimbedBy a rdf:Property.
ex:firstClimbedBy rdfs:SubPropertyOf ex:hasBeenClimbedBy.

ex:firstClimbedBy rdfs:domain ex:Mountain; rdfs:range ex:Person.

ex:hasClimbed a rdf:Property.

ex:firstClimbedBy rdfs:domain ex:Person; rdfs:range ex:Mountain.

ex:hasClimbed owl:inverseOf ex:hasBeenClimbedBy.

ex:LocatedIn a rdf:Property.

ex:locatedInRegion rdfs:SubPropertyOf ex:LocatedIn.

ex:locatedInRegion rdfs:domain ex:Mountain; rdfs:range ex:Region.

ex:RegLocatedInC rdfs:SubPropertyOf ex:LocatedIn.

ex:RegLocatedInC rdfs:domain ex:Region; rdfs:range ex:Country.

ex:CapitalOf rdfs:SubPropertyOf ex:LocatedIn.

ex:CapitalOf rdfs:domain ex:City; rdfs:range ex:Country.ex:CapitalOf a owl:InverseFunctionalProperty

ex:HasCapital a rdfs:Property.

ex:HasCapital rdfs:domain ex:Country; rdfs:range ex:City.

ex:HasCapital a owl:FunctionalProperty.

ex:CapitalOf owl:inverseOf ex:HasCapital

ex:hasPopulation a owl:DataTypeProperty.

ex:hasPopulation a owl:InverseFunctionalProperty
.

ex:hasPopulation rdfs:domain ex:Country.

ex:hasPopulation rdfs:range rdfs:Literal.

Classes

ex:Country a rdfs:Class.
ex:City a rdfs:Class.
ex:City rdfs:subClassOf ex:Country.
ex:Region a rdfs:Class.
ex:Region owl:equivalentClass ex:Country.

ex:Mountain a rdfs:Class.

ex:firstClimbed a owl:ObjectProperty.
ex:firstClimbedBy a owl:ObjectProperty.
ex:firstClimbedBy rdfs:subPropertyOf ex:firstClimbed.
ex:firstClimbedBy rdfs:domain ex:Mountain.
ex:firstClimbedBy rdfs:range ex:Person.

ex:Mountain owl:disjointWith ex:Country.
ex:Mountain owl:disjointWith ex:City.
ex:Mountain owl:disjointWith ex:Region.
ex:Mountain owl:disjointWith ex:Person.
ex:Country owl:disjointWith ex:City.
ex:Country owl:disjointWith ex:Region.
ex:Country owl:disjointWith ex:Person.
ex:City owl:disjointWith ex:Region.
ex:City owl:disjointWith ex:Person.
ex:Region owl:disjointWith ex:Person.

-- RDF data --

ex:MountEverest rdf:type ex:Mountain .

ex:MountEverest ex:locatedInRegion ex:Khumbu .

ex:MountEverest ex:locatedInRegion ex:Xigaze .

ex:MountEverest ex:hasHeightInMeter 8848 .

```
ex:MountEverest ex:firstClimbed [
  ex:firstClimbedBy ex:TenzingNorgay ;
  ex:firstClimbedBy ex:EdmundHillary ;
  ex:firstClimbedInYear 1953 ;
] .
ex:MountEverest ex:hasName "Mount Everest"@en .
ex:MountEverest ex:hasAlternativeName "Sagarmāthā"@ne .
ex:MountEverest ex:hasAlternativeName "Chomolungma"@bo .

ex:Lhotse rdf:type ex:Mountain .
ex:Lhotse ex:locatedInRegion ex:Khumbu .
ex:Lhotse ex:locatedInRegion ex:Xigaze .
ex:Lhotse ex:hasHeightInMeter 8516 .
ex:Lhotse ex:firstClimbed [
  ex:firstClimbedBy ex:ErnstReiss  ;
  ex:firstClimbedBy ex:FritzLuchsinger ;
  ex:firstClimbedInYear 1956 ;
] .
ex:Lhotse ex:hasName "Lhotse" .

ex:AmaDablam rdf:type ex:Mountain .
ex:AmaDablam ex:locatedInRegion ex:Khumbu .
ex:AmaDablam ex:hasHeightInMeter 6170 .
ex:AmaDablam ex:firstClimbed [
  ex:firstClimbedBy ex:MikeGill ;
  ex:firstClimbedBy ex:BarryBishop ;
  ex:firstClimbedBy ex:MikeWard ;
  ex:firstClimbedBy ex:WallyRomanes ;
```

```
] .
ex:AmaDablam ex:hasName "Ama Dablam"@en .

ex:AnnapurnaI rdf:type ex:Mountain .
ex:AnnapurnaI ex:locatedInRegion ex:AnnapurnaConservationArea .
ex:AnnapurnaI ex:hasHeightInMeter 8091 .
ex:AnnapurnaI ex:hasName "AnnapurnaI"@en .

ex:MountKilimanjaro rdf:type ex:Mountain .
ex:MountKilimanjaro ex:locatedInRegion ex:KilimanjaroRegion .
ex:MountKilimanjaro ex:hasHeightInMeter 5895 .
ex:MountKilimanjaro ex:firstClimbed [
  ex:firstClimbedBy ex:HansMeyer ;
  ex:firstClimbedBy ex:LudwigPurtscheller ;
  ex:firstClimbedInYear 1889 ;
] .
ex:MountKilimanjaro ex:hasName "Mount Kilimanjaro"@en .
```

For each considered country, add information about the number of inhabitants and the capital

For each region, add information about the country in which it is located

```
ex:KilimanjaroRegion rdf:type ex:TanzanianRegion .
ex:KilimanjaroRegion ex:RegLocatedInC ex:Tanzania.
ex:Tanzania ex:InhabitedWith "59734218 ".
ex:Tanzania ex:HasCapital ex:Dodoma.
```

ex:Dodoma ex:CapitalOf ex:Tanzania.

ex:AnnapurnaConservationArea ex:RegLocatedInC ex:Nepal.

ex:AnnapurnaConservationArea rdf:type ex:NepaleseRegion .

ex:Nepal ex:InhabitedWith "29646700".

ex:Nepal ex:HasCapital ex:Kathmandu.

ex:Kathmandu ex:CapitalOf ex:Nepal.

ex:Xigaze rdf:type ex:TibetanRegion .

ex:Xigaze ex:hasName "Xigaze" .

ex:Xigaze ex:RegLocatedInC ex:China.

ex:China ex:InhabitedWith "1439323776".

ex:China ex:HasCapital ex:Beijing.

ex:Beijing ex:CapitalOf ex:China.

For each climber, add information about his/her birth date and dead date (if known) and their current occupation

ex:TenzingNorgay rdf:type ex:Person .

ex:TenzingNorgay ex:hasName "Tenzing Norgay" .

ex:TenzingNorgay ex:hasClimbed ex:MountEverest .

ex:TenzingNorgay ex:BirthDate "1914-05-29T00:00:00".

ex:TenzingNorgay ex:DeathDate"1986-05-09T00:00:00".

ex:EdmundHillary rdf:type ex:Person .

ex:EdmundHillary ex:hasName "Edmund Hillary" .
ex:EdmundHillary ex:BirthDate "1919-07-20T00:00:00".
ex:EdmundHillary ex:DeathDate "2008-01-11T00:00:00".

ex:ReinholdMessner ex:hasName "Reinhold Messner".
ex:ReinholdMessner ex:BirthDate "1944-09-17T00:00:00".
ex:ReinholdMessner ex:hasClimbed ex:MountEverest .
ex:ReinholdMessner ex:hasClimbed ex:Lhotse .
ex:AnnapurnaI ex:hasBeenClimbedBy ex:ReinholdMessner .
ex:ReinholdMessner ex:currentOccupation ex:Writer .

ex:ErnstReiss ex:hasName "Ernst Reiss".
ex:ErnstReiss ex:BirthDate"1920-02-24T00:00:00".
ex:ErnstReiss ex:DeathDate "2010-08-03T00:00:00".

ex:FritzLuchsinger ex:hasName "Fritz Luchsinger".
ex:FritzLuchsinger ex: BirthDate "1921-03-08T00:00:00".
ex:FritzLuchsinger ex:DeathDate "1983-04-28T00:00:00".

ex:BarryBishop ex:hasName "Barry Bishop".
ex:BarryBishop ex:BirthDate"1932-01-13T00:00:00".
ex:BarryBishop ex:DeathDate"1994-09-24T00:00:00".

ex:MikeGill ex:hasName "Mike Gill".
ex:MikeGill ex:BirthDate "1921-03-08T00:00:00"^^.
ex:MikeGill ex:DeathDate"1983-04-28T00:00:00"^^.

ex:MikeWard ex:hasName "Michael John Ward".

ex:LudwigPurtscheller ex:hasName "Ludwig Purtscheller".
ex:LudwigPurtscheller ex:BirthDate"1849-10-06T00:00:00".
ex:LudwigPurtscheller ex:DeathDate "1900-03-03T00:00:00".

ex:HansMeyer ex:hasName "Hans Meyer".

ex:WallyRomanes ex:hasName "Wally Romanes".