Athens University of Economics and Business
M.Sc. in Information Systems Development and Security

Assignment 2: Neo4j Graph database

Professor: Y.Kotidis (kotidis@aueb.gr)
Assistant responsible for this assignment: I.Filippidou (filippidoui@aueb.gr)

Λυδία Αθανασίου f3312102
Σοφία Δρούγκα f3312105
5/1/2022

Α) Παρακάτω αναλύουμε διεξοδικά τον Γράφο που χρησιμοποιήσαμε για την υλοποίηση της εργασίας. Χρησιμοποιούμε και ένα διάγραμμα ώστε να επικοινωνήσουμε καλύτερα την σκέψη μας και την λογική που χρησιμοποιήσαμε.

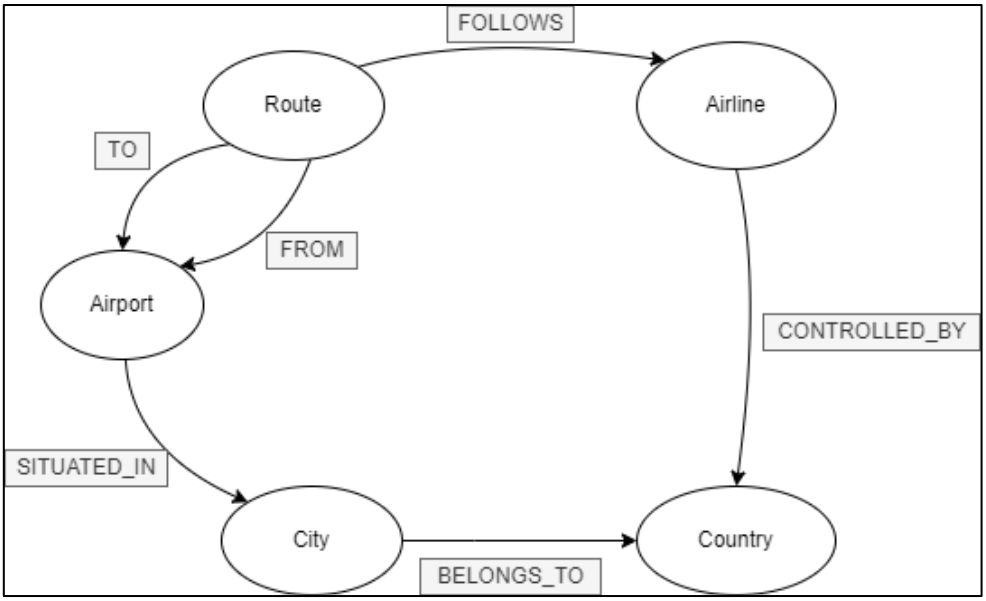


Figure 1: Graph model chart by draw.io

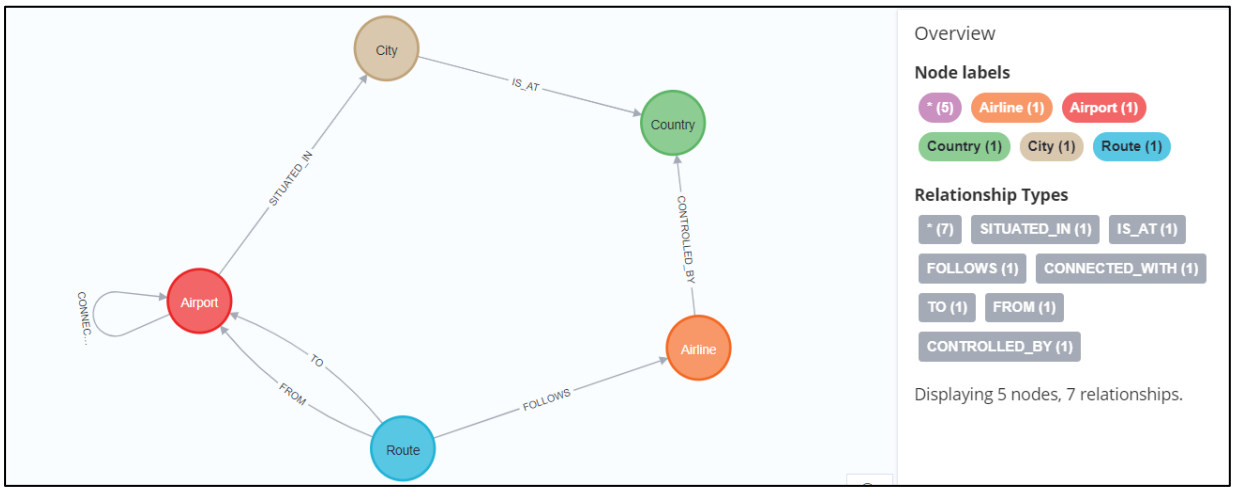


Figure2: Output graph by Neo4j tool

Για την επίτευξη της εργασίας σκεφτήκαμε να δημιουργήσουμε τον Γράφο που φαίνεται στην παραπάνω εικόνα (*Figure 1, Figure 2*) ύστερα από εκτενή μελέτη των excel αρχείων και της περιγραφής των στοιχείων που περιείχαν. Πιο συγκεκριμένα, θεωρήσαμε πως οι σημαντικότερες έννοιες οι οποίες πρέπει οπωσδήποτε να αναπαρασταθούν ως κόμβοι στον Γράφο είναι οι κόμβοι: Route, Airport, Airline, City και Country. Οι έννοιες αυτές θεωρήθηκαν σημαντικές καθώς προσδιορίζουν τα κύρια στοιχεία ενός αεροδρομίου. Τόσο το ίδιο το αεροδρόμιο (airport), όσο και οι αερογραμμές (airline), οι διαδρομές που ακολουθούν αυτές οι αερογραμμές (πτήσεις-routes) και η χώρα (Country) – πόλη (City) προορισμός είναι ιδιάζουσας σημασίας. Προφανώς, κάθε ένας από τους παραπάνω κόμβους έχει και τα αντίστοιχα properties τα οποία φαίνονται στον *Πίνακα 1*. Οι

ιδιότητες αυτές προκύπτουν από τα χαρακτηριστικά κάθε κόμβου που υπάρχουν στα excel/csv αρχεία. Προφανώς οι κόμβοι αυτοί πρέπει να συσχετίζονται με ακμές σύμφωνα με την θεωρία των γράφων και παρατηρώντας τη λογική συσχέτιση των εννοιών σκεφτήκαμε να δημιουργήσουμε τις εξής συσχετίσεις:

- TO → Δηλώνει τον ο συγκεκριμένο προορισμό μιας πτήσης-Route.
- FROM → Δηλώνει το σημείο εκκίνησης μιας πτήσης-Route.
- FOLLOWS → Δηλώνει ότι μία πτήση-διαδρομή ακολουθεί μία συγκεκριμένη αερογραμμή και δεν δρα αυτόνομα.
- SITUATED_IN → Δηλώνει ότι ένα αεροδρόμιο στεγάζεται σε μια συγκεκριμένη πόλη. Άρα, αν ένα συγκεκριμένο αεροδρόμιο είναι ο προορισμός μιας πτήσης – route, τότε ο προορισμός της συγκεκριμένης πτήσης είναι η πόλη στην οποία στεγάζεται το αεροδρόμιο.
- BELONGS_TO → Δηλώνει ότι μια πόλη ανήκει σε μια συγκεκριμένη χώρα.
- CONTROLLED_BY → Δηλώνει την σχέση ανάμεσα στην αερογραμμή και την συγκεκριμένη χώρα.

Στον παρακάτω πίνακα φαίνονται αναλυτικά οι ιδιότητες κάθε κόμβου του γράφου που δημιουργήσαμε για την υλοποίηση της συγκεκριμένης εργασίας. Οι εντολές με τις οποίες εισήχθησαν οι ιδιότητες αυτούς στους κόμβους αναγράφονται παρακάτω στη συγκεκριμένη αναφορά.

Nodes	Attributes
Airline	AirlineID, Name, IATA, ICAO, Country, Active
Airport	AirportID, Name, City, Country, IATA, ICAO, Type, Source
Flight	Airline, AirlineID, source, sourceID, Destination, DstinationID
Country	CountryName
City	CityName

Πίνακας 1: Nodes and Attributes

Β) Οι εντολές που χρησιμοποιήσαμε για να εισάγουμε τα αρχεία στη βάση, για να δημιουργήσουμε τους διαφορετικούς κόμβους, για να ορίσουμε τις συγκεκριμένες ιδιότητες για κάθε κόμβο και τις διάφορες συσχετίσεις μεταξύ τους φαίνεται στη συνέχεια της παρούσας αναφοράς.

Για να εισάγουμε το αρχείο «airports.csv» στην βάση χρησιμοποιήσαμε την εξής εντολή:

- ```
  load csv with headers from "file:///airports.csv" as csvline
  CREATE (airport:Airport { airportId: toInteger(csvline.AirportID),
  country: csvline.Country,city:csvline.City,name:csvline.Name,
  DST:csvline.DST,ICAO:csvline.ICAO,IATA:csvline.IATA,
  source:csvline.Source, type:csvline.Type, tz:csvline.Tz,
  latitude:csvline.Latitude, longitude:csvline.Longitude,
  altitude:csvline.Altitude })
  ```

Χρησιμοποιώντας την ίδια εντολή, την επεκτίναμε ώστε κατά το φόρτωμα του αρχείου να δημιουργούμε απευθείας τους κόμβους που έχουμε ορίσει σε παραπάνω βήμα πως θα περιέχει η βάση μας καθώς και τα κατάλληλα Properties που θα αποδοθούν σε αυτόν. Οπότε χρησιμοποιώντας την παραπάνω εντολή φορτώνουμε το αρχείο, δημιουργούμε τον κόμβο Airport και αποδίδουμε σε αυτόν τα κατάλληλα Attributes από το csv αρχείο.

Ομοίως, για να εισάγουμε το αρχείο «airlines.csv» στην βάση χρησιμοποιήσαμε και να δημιουργήσουμε τον κόμβο Airline με τα κατάλληλα attributes χρησιμοποιήσαμε την εξής εντολή:

- ```
  load csv with headers from "file:///airlines.csv" as csvline
  CREATE (airline:Airline { id: toInteger(csvline.AirlineID), country:
  csvline.Country,ICAO:csvline.ICAO,IATA:csvline.IATA,
  name:csvline.Name, alias:csvline.Alias, callsing:csvline.Callsign,
  active:csvline.Active})
  ```

Ομοίως, για να εισάγουμε το αρχείο «routes.csv» στην βάση χρησιμοποιήσαμε και να δημιουργήσουμε τον κόμβο Route με τα κατάλληλα attributes χρησιμοποιήσαμε την εξής εντολή:

- ```
  load csv with headers from "file:///routes.csv" as csvline
  CREATE (route:Route { airLine:csvline.Airline,
  airlineId:toInteger(csvline.AirlineID),source:csvline.Source,sourceI
  D:toInteger(csvline.SourceID), destination:csvline.Destination,
  destinationId:toInteger(csvline.DestinationID),codeshare:csvline.Cod
  eshare, stops:csvline.Stops,equipment:csvline.Equipment})
  ```
  ```

Για την δημιουργία των εναπομείναντων 2 κόμβων (Contry και City) χρησιμοποιήσαμε τις παρακάτω εντολές:

- ```
  load csv with headers from "file:///airports.csv" as csvline
  CREATE (city:City { cityId: toInteger(csvline.AirportID), country:
  csvline.Country,cityName:csvline.City,airportName:csvline.Name,
  timezone:csvline.Tz})
  ```
- ```
  load csv with headers from "file:///airports.csv" as csvline
  MERGE(country: Country{countryName:csvline.Country})
  CREATE (city)-[:BELONGS_TO]->(country)
  CREATE INDEX ON:Country(countryName)
  CREATE INDEX ON:Airport(name)
  ```

Αξίζει να σημιωθεί πως με την παραπάνω εντολή ταυτοχρόνος δημιουργούμε τον κόμβο Country και την συσχέτιση που υπάρχει ανάμεσα στο Country και το City (BELONGS_TO). Επίσης, σημαντικό είναι το γεγονός πως χωρίς την δημιουργία Index η εντολή έκανε να τρέξει πολύ ώρα και οριακά δεν ολοκληρωνόταν ποτέ. Για να προχωρήσει η εργασία, λοιπόν, δημιουργήσαμε και τα κατάλληλα Indexes όπως φαίνεται και στην παραπάνω εντολή, στους κόμβους Country και Airport.

Για την δημιουργία των υπόλοιπων συσχετίσεων χρησιμοποιήσαμε τις παρακάτω εντολές:

- ```
  MATCH (route:Route ), (airline:Airline) WHERE route.airlineId =
  airline.airlineId
  CREATE (route)-[:FOLLOWS]->(airline)
  ```
- ```
  MATCH (route:Route), (airport:Airport) WHERE route.destinationId =
  airport.airportId
  CREATE (route)-[:TO]->(airport)
  ```
- ```
  MATCH (route:Route), (airport:Airport) WHERE route.sourceID =
  airport.airportId
  CREATE (route)-[:FROM]->(airport)
  ```
- ```
  MATCH (airport:Airport), (city:City) WHERE city.cityId =
  airport.airportId
  CREATE (airport)-[:SITUATED_IN]->(city)
  ```
- ```
  MATCH (airline: Airline), (country: Country)
  WHERE airline.country = country.countryName
  CREATE (airline)-[:CONTROLLED_BY]->(country)
  ```
  ```

C) Στο Τρίτο σκέλος του report έχουμε παραθέσει μια σειρά από screenshots και συγκεκριμένα ένα για κάθε ερώτημα της εργασίας. Στις παρακάτω εικόνες φαίνεται ο κώδικας που χρησιμοποιήσαμε για να ανασύρουμε από τη βάση τις σωστές απαντήσεις στα διάφορα ερωτήματα που θέτονται στην εργασία. Επίσης, στις εικόνες φαίνονται ενδεικτικά και κάποια αποτελέσματα που λάβαμε για κάθε query που υλοποιήσαμε. Αναλυτικά και σε μορφή Cypher ο κώδικας των ερωτημάτων βρίσκεται στο αρχείο queries.cy μέσα στον συμπιεσμένο φάκελο της εργασίας μας.

Query 1: Which are the top 5 airports with the most flights. Return airport name and number of flights.

Query 2: Which are the top 5 countries with the most airports. Return country name and number of airports.

```
1 //Question2
2 MATCH (airport:Airport)
3 RETURN airport.country, COUNT(*) as num_of_airports
4 ORDER BY num_of_airports DESC
5 LIMIT 5;
```

neo4j$ // Question2 MATCH (airport:Airport) RETURN airport.country, COUNT(*) as num_of_airports…

Table | Text | Warn | Code

	airport.country	num_of_airports
1	"United States"	1512
2	"Canada"	430
3	"Australia"	334
4	"Brazil"	264
5	"Russia"	264

Started streaming 5 records after 8 ms and completed after 32 ms.

Query 3: Which are the top 5 airlines with international flights from/to 'Greece'. Return airline name and number of flights.

```
1  // Question 3 Updated
2  MATCH (airport1:Airport), (route:Route), (airport2:Airport), (airline:Airline)
3  WHERE route.source=airport1.IATA
4  AND route.destination=airport2.IATA
5  AND route.airlineId = airline.airlineId
6  AND (airport1.country='Greece' OR airport2.country='Greece')
7  RETURN airline.name AS AIRLINE, COUNT(*) AS INTERNATIONAL_FLIGHTS_TO_FROM_GREECE
8  ORDER BY INTERNATIONAL_FLIGHTS_TO_FROM_GREECE DESC
9  LIMIT 5
10
```

Table | Text | Warn | Code

	AIRLINE	INTERNATIONAL_FLIGHTS_TO_FROM_GREECE
1	"Aegean Airlines"	220
2	"Ryanair"	158
3	"Air Berlin"	120
4	"Olympic Airlines"	110
5	"easyJet"	80

Started streaming 5 records in less than 1 ms and completed after 124 ms.

Query 4: Which are the top 5 airlines with local flights inside ‘Germany’. Return airline name and number of flights.

```
//Question 4 updated
MATCH (airport1:Airport), (route:Route), (airport2:Airport), (airline:Airline)
WHERE route.source=airport1.IATA
AND route.destination=airport2.IATA
AND route.airlineId = airline.airlineId
AND airport1.country='Germany'
AND airport2.country='Germany'
RETURN airline.name AS AIRLINE, COUNT(*) AS FLIGHTS_INSIDE_GERMANY
ORDER BY  FLIGHTS_INSIDE_GERMANY DESC
LIMIT 5;
```

	AIRLINE	FLIGHTS_INSIDE_GERMANY
1	"Lufthansa"	64
2	"Germanwings"	54
3	"Air Berlin"	44
4	"Hainan Airlines"	16
5	"Maastricht Airlines"	6
6	"Intersky"	6

Query 5: Which are the top 10 countries with flights to Greece. Return country name and number of flights.

```
//Question 5
MATCH (airline:Airline), (route:Route), (airport:Airport)
WHERE airline.airlineId=route.airlineId
AND route.destination=airport.IATA
AND airport.country="Greece"
//AND airline.country<>"Greece"
RETURN airline.country AS COUNTRY, COUNT(*) AS NUMBER_OF_FLIIGHTS_TO_GREECE
ORDER BY NUMBER_OF_FLIIGHTS_TO_GREECE DESC
LIMIT 10
```

	COUNTRY	NUMBER_OF_FLIIGHTS_TO_GREECE
1	"Greece"	248
2	"Germany"	162
3	"Ireland"	85
4	"United Kingdom"	61
5	"United States"	59
6	"France"	35

Query 6: Find the percentage of air traffic (inbound and outbound) for every city in Greece. Return city name and the corresponding traffic percentage in descending order.

```
//Question 6
MATCH (route:Route)-[:TO]→(airport:Airport {country:'Greece'})
OPTIONAL MATCH (route:Route)-[:FROM]→(airport:Airport  {country:'Greece'})
WITH COUNT(*) AS total
MATCH (route:Route)-[:TO]→(airport:Airport {country:'Greece'})
OPTIONAL MATCH (route:Route)-[:FROM]→(airport:Airport  {country:'Greece'})
RETURN airport.city AS CITY, 100*COUNT(*)/total AS percentage
ORDER BY percentage DESC
```

neo4j$ //Question 6 NEW MATCH (route:Route)-[:TO]→(airport:Airport {country:'Greece'}) OPTIONA…

Table | Text | Warn | Code

	CITY	percentage
1	"Athens"	25
2	"Heraklion"	13
3	"Rhodos"	10
4	"Thessaloniki"	10
5	"Kerkyra/corfu"	6
6	"Kos"	5

Query 7: Find the number of international flights to Greece with plane types '738' and '320'. Return for each plane type the number of flights.

```
// Question 7
MATCH (airline:Airline),(route:Route), (airport:Airport)
WHERE airline.airlineId=route.airlineId
AND(route.destination=airport.IATA OR route.source=airport.IATA)
AND airport.country="Greece"
AND (route.equipment="738" OR route.equipment="320")
RETURN route.equipment AS PLANE_TYPE, COUNT(*) AS NUMBER_OF_FLIIGHTS
ORDER BY NUMBER_OF_FLIIGHTS DESC
```

Table | Text | Warn | Code

	PLANE_TYPE	NUMBER_OF_FLIIGHTS
1	"320"	331
2	"738"	233

Started streaming 2 records after 1 ms and completed after 389 ms.

Query 8: Which are the top 5 flights that cover the biggest distance between two airports (use function point({ longitude: s1.longitude, latitude: s1.latitude }) and function distance(point1, point2)). Return From (airport), To (airport) and distance in km.

```
// Question 8
MATCH (airport1:Airport), (route:Route), (airport2:Airport)
WHERE route.source=airport1.IATA
AND route.destination=airport2.IATA
WITH airport1 AS a1, airport2 AS a2
WITH a1, a2, point({ longitude: toFloat(a1.longitude), latitude: toFloat(a1.latitude) })
AS source, point({ longitude: toFloat(a2.longitude), latitude: toFloat(a2.latitude) }) AS
destination
RETURN  a1.city AS SOURCE, a2.city AS DESTINATION, round(distance(source,
destination))/1000 AS travelDistance
ORDER BY travelDistance DESC
LIMIT 10
```

Table | Text | Warn | Code

	SOURCE	DESTINATION	travelDistance
1	"Sydney"	"Dallas-Fort Worth"	13823.653
2	"Sydney"	"Dallas-Fort Worth"	13823.653
3	"Johannesburg"	"Atlanta"	13597.81
4	"Atlanta"	"Johannesburg"	13597.81
5	"Los Angeles"	"Dubai"	13415.095
6	"Dubai"	"Los Angeles"	13415.095

Query 9: Find 5 cities that are not connected with direct flights to 'Berlin'. Score the cities in descending order with the total number of flights to other destinations. Return city name and score.

```
match (a1:Airport)-[:CONNECTED_WITH]→(a2:Airport )
WHERE a2.city◇"Berlin"  AND a1.city◇"Berlin"
AND EXISTS ((a2)-[:CONNECTED_WITH]→(:Airport {city:"Berlin"}))
WITH COUNT(a1.city) as total, a1
RETURN a1.city, count(*)/total AS FLIGHTS
ORDER BY FLIGHTS DESC
LIMIT 5
```

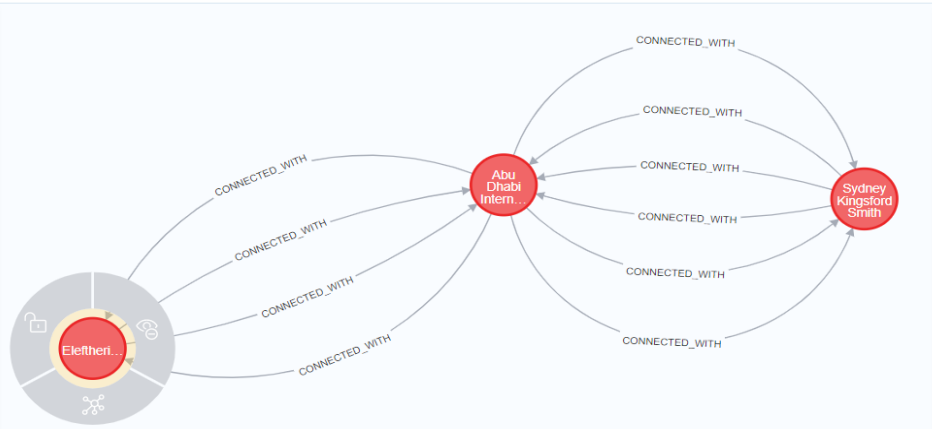
Table | Text | Warn | Code

	a1.city	FLIGHTS
1	"Zanzibar"	1
2	"Windhoek"	1
3	"Godthaab"	1
4	"Eskissehir"	1
5	"Zonguldak"	1

Started streaming 5 records after 1 ms and completed after 136 ms.

Query 10: Find all shortest paths from ‘Athens’ to ‘Sydney’. Use only relations between flights and city airports.

```
MATCH (route:Route)-[:TO]→(airport:Airport)
MATCH (route:Route)-[:FROM]→(airport:Airport)
MATCH (airportSource:Airport {city: 'Athens', country:'Greece'}),(airportDest:Airport
{city: 'Sydney', country:'Australia'}),
p = shortestPath((airportSource)-[*]-(airportDest))
RETURN p
```

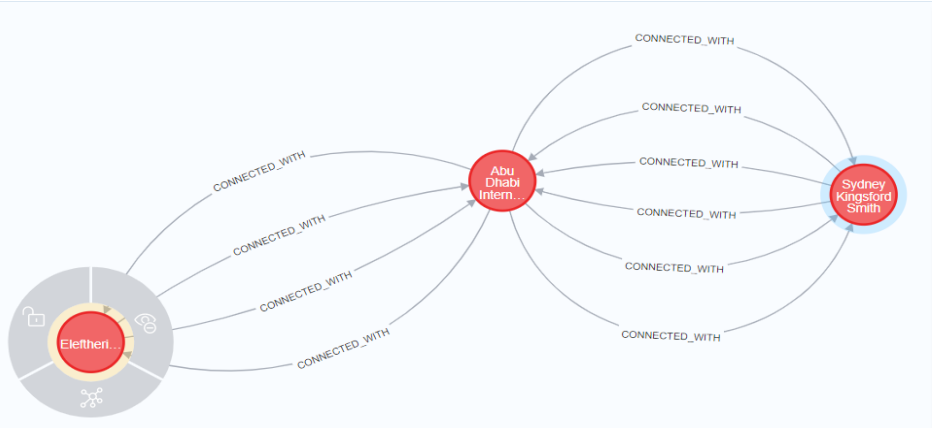


Node Properties

Airport

<id>	3735
DST	E
IATA	ATH
ICAO	LGAV
airportId	3941
altitude	308
city	Athens
country	Greece
latitude	37.9364013672
longitude	23.9444999695
name	Eleftherios Venizelos International Airport
source	OurAirports
type	airport
tz	Europe/Athens

```
MATCH (route:Route)-[:TO]→(airport:Airport)
MATCH (route:Route)-[:FROM]→(airport:Airport)
MATCH (airportSource:Airport {city: 'Athens', country:'Greece'}),(airportDest:Airport
{city: 'Sydney', country:'Australia'}),
p = shortestPath((airportSource)-[*]-(airportDest))
RETURN p
```



Node Properties

Airport

<id>	3167
DST	O
IATA	SYD
ICAO	YSSY
airportId	3361
altitude	21
city	Sydney
country	Australia
latitude	-33.94609832763672
longitude	151.177001953125
name	Sydney Kingsford Smith International Airport
source	OurAirports
type	airport
tz	Australia/Sydney