



## **Εργασία Neo4j**

### **Συστήματα Ανάλυσης & Διαχείρισης Μεγάλων Δεδομένων**

Διδάσκων καθηγητής: Κωτίδης Ιωάννης

Συντελεστές εργασίας: 1) Καρλής Κωνσταντίνος (f3312403)

2) Κούρσος Αναστάσιος(f3312404)

#### **a) Περιγραφή του Μοντέλου Γράφου:**

Το μοντέλο γράφου που υλοποιήθηκε για το σύστημα αεροπορικών πτήσεων αποτελείται από:

#### **Κόμβοι (Nodes):**

##### **1. Airport**

- Ιδιότητες: id (UNIQUE), name, iata, icao, latitude, longitude, altitude, timezone
- Περιγραφή: Αντιπροσωπεύει τα αεροδρόμια με τα βασικά χαρακτηριστικά τους όπως τοποθεσία και ζώνη ώρας

##### **2. City**

- Ιδιότητες: name, country\_name (UNIQUE compound key)
- Περιγραφή: Αντιπροσωπεύει τις πόλεις όπου βρίσκονται τα αεροδρόμια

##### **3. Country**

- Ιδιότητες: name (UNIQUE)

- Περιγραφή: Αντιπροσωπεύει τις χώρες στις οποίες ανήκουν οι πόλεις

#### 4. Airline

- Ιδιότητες: id (UNIQUE), name, alias, iata, icao, callsign, active
- Περιγραφή: Αντιπροσωπεύει τις αεροπορικές εταιρείες και τα χαρακτηριστικά τους

#### 5. Route

- Ιδιότητες: id (UNIQUE - συνδυασμός AirlineID\_SourceID\_DestinationID), stops, codeshare, equipment
- Περιγραφή: Αντιπροσωπεύει τις πτήσεις μεταξύ αεροδρομίων με πληροφορίες για στάσεις και τύπο αεροσκάφους

#### Σχέσεις (Relationships):

##### 1. LOCATED\_IN (Airport → City)

- Περιγραφή: Συνδέει ένα αεροδρόμιο με την πόλη στην οποία βρίσκεται

##### 2. BELONGS\_TO (City → Country)

- Περιγραφή: Συνδέει μια πόλη με τη χώρα στην οποία ανήκει

##### 3. BASED\_IN (Airline → Country)

- Περιγραφή: Συνδέει μια αεροπορική εταιρεία με τη χώρα βάσης της

##### 4. OPERATES (Airline → Route)

- Περιγραφή: Συνδέει μια αεροπορική εταιρεία με τις πτήσεις που εκτελεί

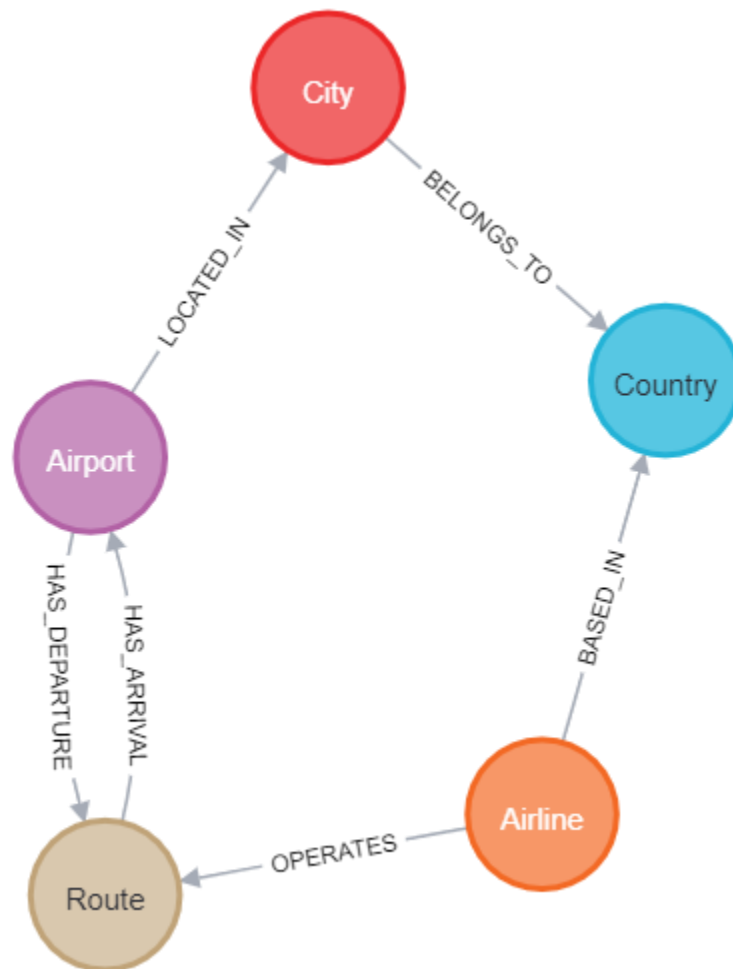
##### 5. HAS\_DEPARTURE (Airport → Route)

- Περιγραφή: Συνδέει ένα αεροδρόμιο με τις πτήσεις που αναχωρούν από αυτό

##### 6. HAS\_ARRIVAL (Route → Airport)

- Περιγραφή: Συνδέει μια πτήση με το αεροδρόμιο προορισμού

**Διαγραμματική απεικόνιση:**



**b) Εισαγωγή αρχείων στην βάση:**

**1. Καθαρισμός και προετοιμασία της Βάσης.**

```
2. // Καθαρισμός υπαρχόντων δεδομένων
3. MATCH (n) DETACH DELETE n;
4.
5. // Δημιουργία περιορισμών
6. CREATE CONSTRAINT airport_id IF NOT EXISTS FOR (a:Airport) REQUIRE a.id
   IS UNIQUE;
7. CREATE CONSTRAINT airline_id IF NOT EXISTS FOR (al:Airline) REQUIRE
   al.id IS UNIQUE;
8. CREATE CONSTRAINT city_compound_key IF NOT EXISTS FOR (c:City) REQUIRE
   (c.name, c.country_name) IS UNIQUE;
9. CREATE CONSTRAINT country_name IF NOT EXISTS FOR (c:Country) REQUIRE
   c.name IS UNIQUE;
```

```
10. CREATE CONSTRAINT route_id IF NOT EXISTS FOR (r:Route) REQUIRE r.id IS UNIQUE;
```

**Επεξήγηση:** Αρχικά καθαρίζουμε τη βάση από τυχόν υπάρχοντα δεδομένα και ορίζουμε τους απαραίτητους περιορισμούς μοναδικότητας (constraints). Αυτό εξασφαλίζει την ακεραιότητα των δεδομένων μας, αποτρέποντας διπλότυπες εγγραφές και εξασφαλίζοντας μοναδικά αναγνωριστικά για κάθε οντότητα.

## 2. Εισαγωγή Βασικών Οντοτήτων

```
3. // Εισαγωγή Countries
4. LOAD CSV WITH HEADERS FROM 'file:///airports.csv' AS row
5. MERGE (c:Country {name: row.Country});
6.
7. // Εισαγωγή Cities
8. LOAD CSV WITH HEADERS FROM 'file:///airports.csv' AS row
9. MERGE (city:City {name: row.City, country_name: row.Country})
10. WITH city, row
11. MATCH (country:Country {name: row.Country})
12. MERGE (city)-[:BELONGS_TO]->(country);
```

**Επεξήγηση:** Εισάγουμε πρώτα τις βασικές οντότητες (χώρες και πόλεις) από το airports.csv. Χρησιμοποιούμε την εντολή MERGE αντί για CREATE για να αποφύγουμε διπλότυπες εγγραφές. Δημιουργούμε επίσης τις σχέσεις BELONGS\_TO μεταξύ πόλεων και χωρών.

## 3. Εισαγωγή Αεροδρομίων

```
4. // Εισαγωγή Airports
5. LOAD CSV WITH HEADERS FROM 'file:///airports.csv' AS row
6. MERGE (a:Airport {id: toInteger(row.AirportID)})
7. SET a.name = row.Name,
8.     a.iata = row.IATA,
9.     a.icao = row.ICAO,
10.    a.latitude = toFloat(row.Latitude),
11.    a.longitude = toFloat(row.Longitude),
12.    a.altitude = toInteger(row.Altitude),
13.    a.timezone = toFloat(row.Timezone)
14. WITH a, row
15. MATCH (city:City {name: row.City, country_name: row.Country})
16. MERGE (a)-[:LOCATED_IN]->(city);
```

**Επεξήγηση:** Εισάγουμε τα αεροδρόμια με όλες τις γεωγραφικές και λειτουργικές πληροφορίες τους. Μετατρέπουμε τους τύπους των δεδομένων όπου χρειάζεται (π.χ. toInteger, toFloat) και συνδέουμε

κάθε αεροδρόμιο με την αντίστοιχη πόλη μέσω της σχέσης LOCATED\_IN.

#### 4. Εισαγωγή Αεροπορικών Εταιριών

```
5. // Εισαγωγή Airlines
6. LOAD CSV WITH HEADERS FROM 'file:///airlines.csv' AS row
7. MERGE (al:Airline {id: toInteger(row.AirlineID)})
8. SET al.name = row.Name,
9.     al.alias = row.Alias,
10.    al.iata = row.IATA,
11.    al.icao = row.ICAO,
12.    al.callsign = row.Callsign,
13.    al.active = row.Active
14. WITH al, row
15. MATCH (country:Country {name: row.Country})
16. MERGE (al)-[:BASED_IN]->(country);
```

**Επεξήγηση:** Εισάγουμε τις αεροπορικές εταιρείες από το airlines.csv με όλα τα χαρακτηριστικά τους. Κάθε εταιρεία συνδέεται με τη χώρα βάσης της μέσω της σχέσης BASED\_IN.

#### 5. Εισαγωγή δρομολογίων

```
6. // Εισαγωγή Routes
7. LOAD CSV WITH HEADERS FROM 'file:///routes.csv' AS row
8. WITH row, (row.AirlineID + '_' + row.SourceID + '_' +
  row.DestinationID) AS routeId
9. MERGE (route:Route {id: routeId})
10. SET route.stops = toInteger(CASE row.Stops WHEN null THEN '0' ELSE
  row.Stops END),
11.    route.codeshare = CASE row.Codeshare WHEN null THEN '' ELSE
  row.Codeshare END,
12.    route.equipment = CASE row.Equipment WHEN null THEN '' ELSE
  row.Equipment END
13. WITH route, row
14. MATCH (al:Airline {id: toInteger(row.AirlineID)})
15. MATCH (src:Airport {id: toInteger(row.SourceID)})
16. MATCH (dst:Airport {id: toInteger(row.DestinationID)})
17. MERGE (al)-[:OPERATES]->(route)
18. MERGE (src)-[:HAS_DEPARTURE]->(route)
19. MERGE (route)-[:HAS_ARRIVAL]->(dst);
20.
```

**Επεξήγηση:** Τέλος, εισάγουμε τα δρομολόγια από το routes.csv. Δημιουργούμε ένα μοναδικό αναγνωριστικό για κάθε δρομολόγιο συνδυάζοντας τα ID της αεροπορικής εταιρείας και των αεροδρομίων αναχώρησης και προορισμού. Δημιουργούμε επίσης όλες τις απαραίτητες σχέσεις: OPERATES για τη σύνδεση με την αεροπορική εταιρεία, HAS\_DEPARTURE για το αεροδρόμιο αναχώρησης και HAS\_ARRIVAL για το αεροδρόμιο προορισμού.

### c) Cypher Queries και Αποτελέσματα

#### 1. Top 5 αεροδρόμια με τις περισσότερες πτήσεις

```
2. MATCH (a:Airport)-[:HAS_DEPARTURE|HAS_ARRIVAL]-(r:Route)
3. WITH a.name as airport_name, COUNT(r) as total_flights
4. RETURN airport_name, total_flights
5. ORDER BY total_flights DESC
6. LIMIT 5;
```

airport_name	total_flights
"Hartsfield Jackson Atlanta International Airport"	1826
"Chicago O'Hare International Airport"	1108
"Beijing Capital International Airport"	1061
"London Heathrow Airport"	1047
"Charles de Gaulle International Airport"	1041

#### 2. Top 5 χώρες με τα περισσότερα αεροδρόμια

```
3. MATCH (c:Country)<-[:BELONGS_TO]-(city:City)<-[:LOCATED_IN]-(a:Airport)
4. WITH c.name as country_name, COUNT(DISTINCT a.id) as airport_count
5. WHERE country_name IS NOT NULL
6. RETURN country_name, airport_count
7. ORDER BY airport_count DESC
```

8. `LIMIT 5;`

country_name	airport_count
"United States"	1512
"Canada"	430
"Australia"	334
"Russia"	264
"Brazil"	264

### 3. Top 5 αεροπορικές με διεθνείς πτήσεις από/προς Ελλάδα

```
4. MATCH (a1:Airline)-[:OPERATES]->(r:Route),
5.      (src:Airport)-[:HAS_DEPARTURE]->(r),
6.      (r)-[:HAS_ARRIVAL]->(dst:Airport),
7.      (src)-[:LOCATED_IN]->(srcCity:City)-[:BELONGS_TO]-
    >(srcCountry:Country),
8.      (dst)-[:LOCATED_IN]->(dstCity:City)-[:BELONGS_TO]-
    >(dstCountry:Country)
9. WHERE (srcCountry.name = 'Greece' AND dstCountry.name <> 'Greece')
10. OR (dstCountry.name = 'Greece' AND srcCountry.name <> 'Greece')
11. WITH a1.name as airline_name, COUNT(DISTINCT r.id) as total_flights
12. WHERE airline_name IS NOT NULL
13. RETURN airline_name, total_flights
14. ORDER BY total_flights DESC
15. LIMIT 5;
```

airline_name	total_flights
"Ryanair"	150
"Aegean Airlines"	136
"Air Berlin"	120
"easyJet"	80
"TUIfly"	64

#### 4. Top 5 αεροπορικές με τοπικές πτήσεις εντός Γερμανίας

```

5. MATCH (al:Airline)-[:OPERATES]->(r:Route),
6.     (src:Airport)-[:HAS_DEPARTURE]->(r),
7.     (r)-[:HAS_ARRIVAL]->(dst:Airport),
8.     (src)-[:LOCATED_IN]->(srcCity:City)-[:BELONGS_TO]-
    >(srcCountry:Country),
9.     (dst)-[:LOCATED_IN]->(dstCity:City)-[:BELONGS_TO]-
    >(dstCountry:Country)
10. WHERE srcCountry.name = 'Germany'
11. AND dstCountry.name = 'Germany'
12. WITH al.name as airline_name, COUNT(DISTINCT r.id) as total_flights
13. WHERE airline_name IS NOT NULL
14. RETURN airline_name, total_flights
15. ORDER BY total_flights DESC
16. LIMIT 5;

```

airline_name	total_flights
"Lufthansa"	64
"Germanwings"	54
"Air Berlin"	44
"Hainan Airlines"	16
"Ethiopian Airlines"	6



## 5. Top 10 χώρες με πτήσεις προς Ελλάδα

```
6. //Εδω εμφανίζουμε 11 κατηγορίες καθώς μετράμε για πτήσεις προς την
   Ελλάδα και τις εσωτερικές πτήσεις. Δεν ξέραμε αν ήταν απαραίτητο για
   αυτο βάλαμε 11.
7. MATCH (a1:Airline)-[:OPERATES]->(r:Route),
8.       (src:Airport)-[:HAS_DEPARTURE]->(r),
9.       (r)-[:HAS_ARRIVAL]->(dst:Airport),
10.      (src)-[:LOCATED_IN]->(srcCity:City)-[:BELONGS_TO]-
    >(srcCountry:Country),
11.      (dst)-[:LOCATED_IN]->(dstCity:City)-[:BELONGS_TO]-
    >(dstCountry:Country)
12.WHERE dstCountry.name = 'Greece'
13.WITH srcCountry.name as country_name, COUNT(DISTINCT r.id) as
    total_flights
14.WHERE country_name IS NOT NULL
15.RETURN country_name, total_flights
16.ORDER BY total_flights DESC
17.LIMIT 11;
```

country_name	total_flights
"Greece"	235
"Germany"	176
"United Kingdom"	105
"Austria"	36
"France"	32
"Russia"	28
"Italy"	25
"Netherlands"	21
"Belgium"	20
"Cyprus"	12
"Norway"	11

## 6. Ποσοστό αεροπορικής κίνησης για κάθε πόλη στην Ελλάδα

```
7. //στο screenshot δεν χωρουν ολες οι χωρες.  
8. MATCH (city:City)-[:BELONGS_TO]->(country:Country {name: 'Greece'})  
9. WITH city  
10.  
11. OPTIONAL MATCH (city)<-[:LOCATED_IN]-(airport:Airport)  
12. OPTIONAL MATCH (airport)-[:HAS_DEPARTURE|HAS_ARRIVAL]-(route:Route)  
13.  
14. WITH city.name as city_name, COUNT(DISTINCT route) as city_traffic  
15.  
16. WITH COLLECT({city: city_name, traffic: city_traffic}) as cities,  
17.      SUM(city_traffic) as total_traffic  
18.  
19. UNWIND cities as city_data  
20. RETURN city_data.city as city_name,  
21.        CASE  
22.          WHEN total_traffic > 0 THEN round(100.0 * city_data.traffic /  
            total_traffic, 2)  
23.          ELSE 0  
24.        END as traffic_percentage  
25. ORDER BY traffic_percentage DESC;
```

city_name	traffic_percentage
"Athens"	25.17
"Heraklion"	13.68
"Thessaloniki"	10.87
"Rhodos"	10.49
"Kerkyra/corfu"	6.43
"Kos"	5.62
"Chania"	5.62
"Zakynthos"	2.06
"Thira"	1.94
"Mykonos"	1.62
"Preveza"	1.37
"Kalamata"	1.25
"Mytilini"	1.25
"Samos"	1.12
"Keffallinia"	0.81

## 7. Διεθνείς πτήσεις προς Ελλάδα με αεροσκάφη τύπου '738' και '320'

```
8. // Βρίσκουμε πτήσεις που καταλήγουν σε ελληνικά αεροδρόμια
9. MATCH (sourceAirport:Airport)-[:LOCATED_IN]->(sourceCity:City)-
   [:BELONGS_TO]->(sourceCountry:Country),
10.      (destAirport:Airport)-[:LOCATED_IN]->(City)-[:BELONGS_TO]-
   >(destCountry:Country {name: 'Greece'}),
11.      (sourceAirport)-[:HAS_DEPARTURE]->(route:Route)-[:HAS_ARRIVAL]-
   >(destAirport)
12.// Βεβαιωνόμαστε ότι η πτήση είναι διεθνής (διαφορετικές χώρες)
13.WHERE sourceCountry.name <> 'Greece'
14. AND route.equipment IN ['738', '320']
15.// Επιστρέφουμε τον τύπο αεροπλάνου και το πλήθος πτήσεων
16.RETURN route.equipment as aircraft_type,
17.      COUNT(route) as number_of_flights
18.ORDER BY aircraft_type;
```

aircraft_type	number_of_flights
"320"	147
"738"	110

## 8. Top 5 πτήσεις με τη μεγαλύτερη απόσταση

```
9. MATCH (src:Airport)-[:HAS_DEPARTURE]->(r:Route)-[:HAS_ARRIVAL]-
   >(dst:Airport)
10.WITH src, dst,
11.point({ longitude: src.longitude, latitude: src.latitude }) as p1,
12.point({ longitude: dst.longitude, latitude: dst.latitude }) as p2
13.WITH src.name as from_airport,
14.      dst.name as to_airport,
15.      round(point.distance(p1, p2)/1000*100)/100 as distance_km
16.RETURN from_airport, to_airport, distance_km
17.ORDER BY distance_km DESC
18.LIMIT 5;
```

from_airport	to_airport	distance_km
"Los Alamitos Army Air Field"	"Kenneth Kaunda International Airport Lusaka"	16100.28
"Kenneth Kaunda International Airport Lusaka"	"Los Alamitos Army Air Field"	16100.28
"Simon Mwansa Kapwepwe International Airport"	"Los Alamitos Army Air Field"	15954.98
"Los Alamitos Army Air Field"	"Simon Mwansa Kapwepwe International Airport"	15954.98
"Sydney Kingsford Smith International Airport"	"Dallas Fort Worth International Airport"	13823.65

## 9. 5 πόλεις χωρίς απευθείας πτήσεις προς Βερολίνο

```
10.MATCH (c:City)
11.WHERE NOT EXISTS {
12.    // Ελέγχουμε ΜΟΝΟ για απευθείας πτήσεις προς Βερολίνο
13.    MATCH (c)-[:LOCATED_IN]-(a:Airport)
14.    -[:HAS_DEPARTURE]->(:Route)-[:HAS_ARRIVAL]->
15.    (:Airport)-[:LOCATED_IN]->(:City {name: 'Berlin'})
16.}
17.WITH c
18.MATCH (c)-[:LOCATED_IN]-(airport:Airport)
19.-[:HAS_DEPARTURE]->(route:Route)
20.WITH c.name as city_name, COUNT(DISTINCT route) as total_flights
21.WHERE total_flights > 0
22.RETURN city_name, total_flights
23.ORDER BY total_flights DESC
24.LIMIT 5;
```

city_name	total_flights
"Atlanta"	915
"Shanghai"	610
"Los Angeles"	489
"Dallas-Fort Worth"	469
"Tokyo"	443

## 10.Συντομότερα μονοπάτια από Αθήνα προς Σίδνεϊ

```
11.MATCH (start:Airport)-[:LOCATED_IN]->(startCity:City {name: 'Athens'}),
12.    (end:Airport)-[:LOCATED_IN]->(endCity:City {name: 'Sydney'})
13.MATCH p = shortestPath(
14.    (start)-[:HAS_DEPARTURE|HAS_ARRIVAL*]-(end)
15.)
16.RETURN [node in nodes(p) WHERE node:Airport | node.name] as
    ShortestPath,
17.    length(p)/2 as PathLength
18.ORDER BY PathLength;
```

ShortestPath	PathLength
["Eleftherios Venizelos International Airport", "Lester B. Pearson International Airport", "Sydney / J.A. Douglas McCurdy Airport"]	2
["Eleftherios Venizelos International Airport", "Abu Dhabi International Airport", "Sydney Kingsford Smith International Airport"]	2
["Athens Ben Epps Airport", "Nashville International Airport", "Lester B. Pearson International Airport", "Sydney / J.A. Douglas McCurdy Airport"]	3
["Athens Ben Epps Airport", "Nashville International Airport", "Los Angeles International Airport", "Sydney Kingsford Smith International Airport"]	3