

Neo4j airport project

Създаване на nodes:

LOAD CSV WITH HEADERS FROM

"https://raw.githubusercontent.com/YordanTop/airport_routes_project/prototype/airports.csv"

AS row

CREATE (:_Airports {`Airport ID`:row.`Airport ID`, City:row.City, Country:row.Country, IATA:row.IATA, ICAO:row.ICAO, Latitude:row.Latitude, Longitude:row.Longitude, Altitude:row.Altitude, Timezone:row.Timezone, DST:row.DST, `TZ database timezone`:row.`TZ database timezone`, Type:row.Type, Source:row.Source});

Създаване на връзки:

LOAD CSV WITH HEADERS FROM

"https://raw.githubusercontent.com/YordanTop/airport_routes_project/prototype/routes.csv"

AS row

WITH row

MATCH (_a:_Airports {IATA: row.`Source airport`})

WITH row, _a

MATCH (_a2:_Airports {IATA: row.`Destination airport`})

CREATE (_a)-[:ROUTE{Stops:row.Stops}]->(_a2);

Създаване на Уникален ключ.

CREATE CONSTRAINT FOR (a:_Airports) REQUIRE a.IATA IS UNIQUE;

1.Кои летища можем да посетим от София и които са на 1 дъга разстояние?

MATCH (a:_Airports)-[r:ROUTE]->(b:_Airports) WHERE a.IATA = 'SOF' RETURN a,b,r

2.А на 2 дъги разстояние?

MATCH (a:_Airports)-[r:ROUTE]-(b:_Airports)-[r2:ROUTE]->(c:_Airports) WHERE a.IATA = 'SOF' RETURN a,b,c,r,r2

3.Същото за Пловдив

MATCH (a:_Airports)-[r:ROUTE]->(b:_Airports) WHERE a.IATA = 'PVD' RETURN a,b,r

MATCH (a:_Airports)-[r:ROUTE]-(b:_Airports)-[r2:ROUTE]->(c:_Airports) WHERE a.IATA = 'PVD' RETURN a,b,c,r,r2

4.Най-краткият път от Пловдив до Сидни?

MATCH p = shortestPath((a:_Airports {IATA:'PVD'})-[*]->(b:_Airports {IATA:'SYD'})) return p

5. А най-бързият?

```
MATCH (a: `_Airports` {IATA: "PVD"}), (b: `_Airports` {IATA: "SYD"}) CALL  
apoc.algo.dijkstra(a, b, 'ROUTE', "SYD") YIELD path, weight RETURN path, weight
```

6. Същото, но за Лондон

```
MATCH p = shortestPath((a: `_Airports` {IATA: "PVD"})-[*]->(b: `_Airports` {IATA: "LCY"})) RETURN p
```

```
MATCH (a: `_Airports` {IATA: "PVD"}), (b: `_Airports` {IATA: "LCY"}) CALL  
apoc.algo.dijkstra(a, b, 'ROUTE', "LCY") YIELD path, weight RETURN path, weight
```

7. Същото от Монреал до София

```
MATCH p = shortestPath((a: `_Airports` {IATA: "YUL"})-[*]->(b: `_Airports` {IATA: "SOF"})) RETURN p
```

```
MATCH (a: `_Airports` {IATA: "YUL"}), (b: `_Airports` {IATA: "SOF"}) CALL  
apoc.algo.dijkstra(a, b, 'ROUTE', "SOF") YIELD path, weight RETURN path, weight
```

8. Най-натовареното летище?

```
MATCH (a: `_Airports`) <- [r: ROUTE] - (b: `_Airports`) RETURN a, count(*) ORDER BY count(*)  
DESC LIMIT 1
```