Coding Practice Week 10

Aaryan Agarwal fall 2023

# Part A

LOAD CSV WITH HEADERS FROM

'file:///D:/Programs/Personal/MSDS/ADT/HospitalGeneralInformation.csv' AS line

WITH line

LIMIT 1

RETURN line

A screenshot of a computer

Description automatically generated

CREATE INDEX index\_name For (n: County) ON (n.name);

CREATE INDEX index\_name1 For (n:City) ON (n.name);

CREATE INDEX index\_name2 For (n: ZipCode) ON (n.name);

CREATE INDEX index\_name3 For (n:Address) ON (n.name);

A screenshot of a computer

Description automatically generated

SHOW INDEXES

A screenshot of a computer

Description automatically generated

CREATE CONSTRAINT FOR (h:Hospital) REQUIRE h.id IS UNIQUE;

CREATE CONSTRAINT FOR (s:State) REQUIRE s.name IS UNIQUE;

A screenshot of a computer

Description automatically generated

SHOW CONSTRAINTS

A screenshot of a computer

Description automatically generated

:auto LOAD CSV WITH HEADERS FROM

'file:///D:/Programs/Personal/MSDS/ADT/HospitalGeneralInformation.csv' AS row

WITH row

WHERE row.State = 'NY'

// state name is unique

MERGE (state:State{name:row.State})

// merge by pattern with their parents

MERGE (state)<-[:IS\_IN]-(county:County{name:row.County\_Name})

MERGE (county)<-[:IS\_IN]-(city:City{name:row.City})

MERGE (city)<-[:IS\_IN]-(zip:ZipCode{name:row.ZIPCode})

MERGE (zip)<-[:IS\_IN]-(address:Address{name:row.Address})

// for entities, it is best to have an id system

MERGE (h:Hospital{id:row.Provider\_ID})

MERGE (h)-[:IS\_IN]->(address)

A screenshot of a computer

Description automatically generated

:auto LOAD CSV WITH HEADERS FROM

'file:///D:/Programs/Personal/MSDS/ADT/HospitalGeneralInformation.csv' AS row

WITH row

WHERE row.State = 'NY'

MATCH (h:Hospital{id:row.Provider\_ID})

SET h.phone=row.Phone\_Number,

h.emergency\_services = row.Emergency\_Services,

h.name= row.Hospital\_Name,

h.mortality = row.Mortality\_national\_comparison,

h.safety = row.Safety\_of\_care\_national\_comparison,

h.timeliness = row.Timeliness\_of\_care\_national\_comparison,

h.experience = row.Patient\_experience\_national\_comparison,

h.effectiveness = row.Effectiveness\_of\_care\_national\_comparison

MERGE (type:HospitalType{name:row.Hospital\_Type})

MERGE (h)-[:HAS\_TYPE]->(type)

MERGE (ownership:Ownership{name: row.Hospital\_Ownership})

MERGE (h)-[:HAS\_OWNERSHIP]->(ownership)

MERGE (rating:Rating{name:row.Hospital\_overall\_rating})

MERGE (h)-[:HAS\_RATING]->(rating)

A screenshot of a computer

Description automatically generated

LOAD CSV WITH HEADERS FROM 'file:///D:/Programs/Personal/MSDS/ADT/gpsinfo.csv' AS row

WITH row

MATCH (hospital:Hospital {id:row.id})

SET hospital.latitude = toFloat(row.latitude),

hospital.longitude = toFloat(row.longitude)

A screenshot of a computer

Description automatically generated

call db.schema.visualization()

A computer screen shot of a network

Description automatically generated

MATCH (n) RETURN n LIMIT 14

A screenshot of a computer

Description automatically generated

MATCH (n:City) RETURN n LIMIT 10

A screen shot of a computer

Description automatically generated

Query 1

MATCH (h:Hospital)-[:IS\_IN\*3..3]->(city)

RETURN city.name AS city, count(h) AS NumberOfHospitals

ORDER BY NumberOfHospitals DESC LIMIT 15

A screenshot of a computer

Description automatically generated

Query 2

MATCH (h:Hospital)-[:IS\_IN\*2..2]->(city)

RETURN city.name AS zip, count(h) AS NumberOfHospitals

ORDER BY NumberOfHospitals DESC LIMIT 15

A screenshot of a computer

Description automatically generated

Query 3

MATCH (h:Hospital)-[:IS\_IN\*2..2]->(city)

RETURN city.name AS street, count(h) AS NumberOfHospitals

ORDER BY NumberOfHospitals DESC LIMIT 15

A screenshot of a computer

Description automatically generated

Query 4

MATCH (h:Hospital)-[:IS\_IN\*3..3]->(City)

WHERE h.mortality = 'Below the National average'

RETURN City.name AS city, count(h) AS NumberOfHospitals

ORDER BY NumberOfHospitals DESC LIMIT 15

A screenshot of a computer

Description automatically generated

Query 5

MATCH (p) WHERE p.emergency\_services = 'Yes'

return p

limit 10

A screenshot of a computer

Description automatically generated

# Part 2

LOAD CSV WITH HEADERS FROM "file:///full\_path/flights\_10k.csv" AS row

RETURN row

LIMIT 5

A screen shot of a computer

Description automatically generated

CREATE CONSTRAINT FOR (a:Airport)

REQUIRE a.code IS UNIQUE;

CREATE CONSTRAINT FOR (f:Flight)

REQUIRE f.id IS UNIQUE;

A screenshot of a computer

Description automatically generated

LOAD CSV WITH HEADERS FROM "file:///your\_full\_path/flights\_10k.csv" AS row

MERGE (origin:Airport {code: row.Origin})

MERGE (destination:Airport {code: row.Dest})

WITH row.UniqueCarrier + row.FlightNum + "\_" + row.Year + "-" + row.Month + "-" + row.DayofMonth + "\_" + row.Origin + "\_" + row.Dest AS flightIdentifier, row, origin, destination

MERGE (flight:Flight { id: flightIdentifier })

ON CREATE SET flight.date = row.Year + "-" + row.Month + "-" + row.DayofMonth,

flight.airline = row.UniqueCarrier, flight.number = row.FlightNum, flight.departure = row.CRSDepTime,

flight.arrival = row.CRSArrTime, flight.distance = row.Distance, flight.cancelled = row.Cancelled

MERGE (flight)-[:ORIGIN]->(origin)

MERGE (flight)-[:DESTINATION]->(destination)

A screen shot of a computer

Description automatically generated

MATCH (:Flight)

RETURN count(\*)

A screen shot of a computer

Description automatically generated

call db.schema.visualization()

A screenshot of a computer

Description automatically generated

MATCH (flight:Flight)

SET flight.distance = toInteger(flight.distance)

A screenshot of a computer

Description automatically generated

MATCH (flight:Flight)

WHERE flight.distance > 500

RETURN flight

A screenshot of a computer

Description automatically generated

MATCH (flight:Flight)

SET flight.cancelled = CASE WHEN flight.cancelled = "1" THEN true ELSE false END

A screen shot of a computer

Description automatically generated

MATCH (flight:Flight)

WHERE flight.cancelled

RETURN flight

A screenshot of a computer

Description automatically generated

Query 1

PROFILE

MATCH path = (origin:Airport {code: "LAS"})<-[:ORIGIN]-(flight:Flight)-[:DESTINATION]->(destination:Airport {code: "MDW"})

WHERE flight.date = "2008-1-3"

RETURN path

A screenshot of a computer

Description automatically generated

Query 2

:auto LOAD CSV WITH HEADERS FROM "file:///you\_path/flights\_10k.csv" AS row

MERGE (origin)-[connection:CONNECTED\_TO {

airline: row.UniqueCarrier,

flightNumber: row.FlightNum,

date: toInteger(row.Year) + "-" + toInteger(row.Month) + "-" + toInteger(row.DayofMonth)}]->(destination)

ON CREATE SET connection.departure = toInteger(row.CRSDepTime), connection.arrival = toInteger(row.CRSArrTime)

A screenshot of a computer

Description automatically generated

MATCH ()-[:CONNECTED\_TO]->()

RETURN COUNT(\*)

A screenshot of a computer

Description automatically generated

Query 3

MATCH path = (origin:Airport {code: "LAS"})<-[:ORIGIN]-(flight:Flight)-[:DESTINATION]->(destination:Airport {code: "MDW"})

WHERE flight.date = "2008-1-3"

RETURN path

A screenshot of a computer

Description automatically generated