Universitat Autònoma de Barcelona
Facultat de Ciències

Pràctica NEO4J

Autors:
Manuel Arnau & Sofia Di Capua & Gerard Lahuerta & Ona Sánchez
1597487 — 1603685 — 1601350 — 1601181

18 de Juny de 2023

Contents

1 Introducció

En aquest projecte es treballa el disseny, la implementació i la consulta a una base de dades en NEO4J. A partir dels requisits i les dades que s'han subministrat, s'ha implementat un script en Cypher que processa i insereix les dades en una base de dades de NEO4J. Seguidament, s'han implementat les consultes demanades per corroborar l'ús correcte de la base de dades.

L'objectiu d'aquest projecte és afermar els conceptes ensenyats a classe mitjançant una aplicació diversa i realista d'una base de dades.

Podeu veure tots els codis a la següent pàgina de github:

https://github.com/Gerard-Lahuerta/Projecte_NEO4J.git

2 Exercicis

2.1 Exercici 1

Per tal de poder treballar amb NEO4J s'ha hagut d'importar les dades que s'hi disposen en els fitxers (*FAMILIA*, *HABITATGES*, *INDIVIDUAL*, *SAME_AS* i *VIU*)[1] en format *CSV* al format Cypher.

Per tal de fer la correcta importació de les dades amb els tractaments de no duplicitat de dades, no importació de dades amb codis identificatius *null* i tipus de variables correctes, s'ha creat un document en Cypher encarregat de fer tots aquests processos: .

Mostrem a continuació el contingut del fitxer i els processos que fa.

2.1.1 Importació i creació del node Habitatge

```
LOAD CSV WITH HEADERS FROM
 "https://docs.google.com/spreadsheets/d/e/2PACX-
1vT0ZhR6BSO_M72JEmxXKs6GLuOwxm_Oy0UruLJeX8_R04KA
cICuvrwn2OENQhtuvddU5RSJSclHRJf/puboutput=csv" as row
with toInteger(row.Id_Llar) as LlarID, row.Municipi as
Municipi, toInteger(row.Any_Padro) as AnyPadro, row.Carrer as
 Carrer, toInteger(row.Numero) as Numero

WHERE LlarID is not NULL and Municipi<>'null'

CREATE (h:Habitatge {LlarID: LlarID, Municipi: Municipi,
 AnyPadro: AnyPadro})

SET h.Carrer = Carrer,  h.Num = Numero

CREATE CONSTRAINT FOR (h:Habitatge) REQUIRE (h.LlarID,
h.Municipi, h.AnyPadro) IS NODE KEY
```

[1] Aquets fitxer per a un desembolupament adient i escalable del treball en el nostre grup s'han decidit suministrar-los en el fitxer mitjançant un url a un directori drive on són descargat quan es generen les dades al NEO4J.

2.1.2 Importació i creació del node Individu

```
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/
2PACX-1vTfU6oJBZhmhzzkV_0-avABPzHTdXy8851ySDbn2gq32WwaNmYxfiBtCGJGOZ
sMgCWjzlEGX4Zh1wqe/pub?output=csv" as row
WITH row
WHERE row.Id is not NULL
CREATE (i:Individu {IndividuID: row.Id})
SET i.Year = toInteger(row.Year), i.Name = row.name,
    i.Surname = row.surname, i.SecondSurname = row.second_surname

CREATE CONSTRAINT UniqueIndividuID FOR (i:Individu)
REQUIRE i.IndividuID IS UNIQUE;
```

2.1.3 Importació i creació de la relació Familia

```
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-1vRVOoMAMoxHiGboT
jCIHo2yT30CCWgVHgocGnVJxiCTgyurtmqCfAFahHajobVzwXFLwhqajz1fqA8d/
pub?output=csv" AS row
WITH row.ID_1 AS Id_1, row.ID_2 AS Id_2, row.Relacio AS relacio,
    row.Relacio_Harmonitzada AS relacio_harmonitzada
MATCH (p:Individu {IndividuID:Id_1})
MATCH (o:Individu {IndividuID:Id_2})
MERGE (o)-[rel:Relacio_Familiar {relacio: relacio,
    relacio_harmonitzada:relacio_harmonitzada}]->(p)
RETURN count(rel);
```

2.1.4 Importació i creació de la relació SAME_AS

```
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-1vTgC8TBmdXhjUOPK
JxyiZSpetPYjaRC34gmxHj6H2AWvXTGbg7MLKVdJnwuh5bIeer7WLUi0OigI6wc/
pub?output=csv" AS row
WITH row.Id_A AS Id_A, row.Id_B AS Id_B
MATCH (p:Individu {IndividuID:Id_A})
MATCH (o:Individu {IndividuID:Id_B})
MERGE (o)-[rel: SAME_AS]-(p)
RETURN count(rel);
```

2.1.5 Importació i creació de la relació VIU

```
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/
2PACX-1vRM4DPeqFmv7w6kLH5msNk6_Hdh1wuExRirgysZKO_Q70L21MKBkDISIyjvd
m8shVixl5Tcw_5zCfdg/pub?output=csv' AS vivencia
WITH vivencia
MATCH (p:Individu {IndividuID:vivencia.IND})
MATCH(h:Habitatge{LlarID:toInteger(vivencia.HOUSE_ID),
                  Municipi:vivencia.Location,
                  AnyPadro:toInteger(vivencia.Year)})
MERGE (p)-[rel:VIU_A]-(h)
return count(rel)
```

2.2 Exercici 2

2.3 Apartat 1

```
MATCH (h:Habitatge)<-[v:VIU_A]-(i:Individu)
WHERE h.Municipi='CR'
RETURN count(i.IndividuID) AS N_habitants,
       collect(i.Surname) AS Cognoms
```

2.4 Apartat 2

```
match (h:Habitatge)<-[v:VIU_A]-(i:Individu)
where h.Municipi='SFLL' and i.Surname<>'nan'
    and i.Surname<>'?' is not null
return distinct h.YearPadro as Any_Padro,
    count(i.IndividuID) as N_habitants,
    collect(DISTINCT i.Surname) as Cognoms
```

2.4.1 Apartat 3

```
match (h:Habitatge {Municipi: "SFLL"})
where h.AnyPadro < 1845 and h.AnyPadro > 1800
return h.Municipi as `Població`,
       h.AnyPadro as `Any_Padro`,
       collect(h.LlarID) as `Identificador_LLar`
```

2.4.2 Apartat 4

```
MATCH (h:Habitatge)<-[v:VIU_A]-(i:Individu)
WHERE h.Municipi='SFLL' AND
      i.Surname<>'nan' AND
      i.Surname<>'?' IS NOT NULL
RETURN DISTINCT h.YearPadro AS Any_Padro,
       count(i.IndividuID) AS N_habitants,
       collect(DISTINCT i.Surname) AS Cognoms
```

2.5 Apartat 5

```
MATCH (i:Individu {Name:"miguel",
                   Surname:"estape",
                   SecondSurname:"bofill"})
      -[r:SAME_AS]->(l:Individu)
return i,l
```

2.6 Apartat 6

```
MATCH (i:Individu {Name:"miguel",
                   Surname:"estape",
                   SecondSurname:"bofill"})
      -[r:SAME_AS]->(l:Individu)
return l.Name,
       collect(distinct l.Surname),
       collect(distinct l.SecondSurname)
```

2.7 Apartat 7

```
match (p:Individu {Name:"benito", Surname: "julivert"})
       -[rel]- (r:Individu)
where p.IndividuID <> r.IndividuID
return r.Name+"␣"+r.Surname+"␣"+r.SecondSurname as `Individu`,
       type(rel) as `relació`
```

2.8 Apartat 8

```
match (p:Individu {Name:"benito", Surname: "julivert"})
      -[rel]- (r:Individu)
where rel.relacio_harmonitzada = "fill" or
      rel.relacio_harmonitzada = "filla"
return r.Name+"␣"+r.Surname+"␣"+r.SecondSurname as `Individu`,
       type(rel) as `relació`
order by r.Name
```

2.9 Apartat 9

```
match (i:Individu)-[f:Relacio_Familiar]-(i2:Individu)
where f.relacio<>'null'
return distinct f.relacio
```

2.10 Apartat 10

```
match (h:Habitatge)<-[v:VIU_A]-(i:Individu)
where h.Municipi='SFLL' and h.Carrer IS NOT NULL
    and h.Num is not null
return h.Carrer as Carrer, h.Num as Numero,
    count(h.YearPadro) as TotalPadrons,
    collect(distinct h.YearPadro) as AnysPadrons,
    collect(distinct h.LlarID) as Id_Llars
order by TotalPadrons DESC limit 15
```

2.11 Apartat 11

```
MATCH (h:Habitatge{Municipi:"CR"})
      -[:VIU_A]-(cap:Individu)<-[rel:Relacio_Familiar]-
      (fill:Individu)
WHERE rel.relacio = "hijo" or
      rel.relacio = "hija" and
      cap.IndividuID <> fill.IndividuID
WITH  collect(distinct fill.IndividuID) as Fills,
      cap.IndividuID as Pare,
      collect(fill.Name) as NomsFills,
      cap.Name as NomPare
WHERE size(Fills) >= 3
RETURN NomsFills, NomPare, size(NomsFills)
ORDER BY size(NomsFills) DESC LIMIT 20
```

2.12 Apartat 12

```
CALL {
    MATCH(h:Habitatge{Municipi:"SFLL", AnyPadro:1881})
    RETURN count(h.LlarID) as NombreLlars
}

MATCH (h:Habitatge{Municipi:"SFLL", AnyPadro:1881})
      -[:VIU_A]-(cap:Individu)<-[rel:Relacio_Familiar]-
      (fill:Individu)
WHERE rel.relacio_harmonitzada = "fill" or
      rel.relacio_harmonitzada = "filla" and
      cap.IndividuID <> fill.IndividuID
WITH  distinct fill.IndividuID as Fills, NombreLlars
RETURN count(Fills), NombreLlars, count(Fills)/NombreLlars as Mitja
```

2.13 Apartat 13

```
call {
    match (i:Individu)-[v:VIU_A]->(h:Habitatge)
    where h.Municipi='SFLL'
    return count(i) as habitants,
           h.Carrer as carrer,
           h.AnyPadro as any
}
with habitants, carrer, any
order by habitants ASC
return collect(carrer)[0] as carrer,
       min(habitants) as Num_Habitants,
       any
order by any
```

2.14 Exercici 3

2.14.1 Apartat a

L'objectiu d'aquest exercici és fer un estudi de les components connexes (cc) i de l'estructura de les components en funció de la seva mida.

Abans de començar hem de fer una projecció del graf en memoria en la qual executarem algorismes de la GDS. Per fer-ho, usem la següent comanda a Neo4j:

```
CALL gds.graph.project('ex3a',['Individu','Habitatge'],
                              ['VIU_A','SAME_AS','Relacio_Familiar'])
```

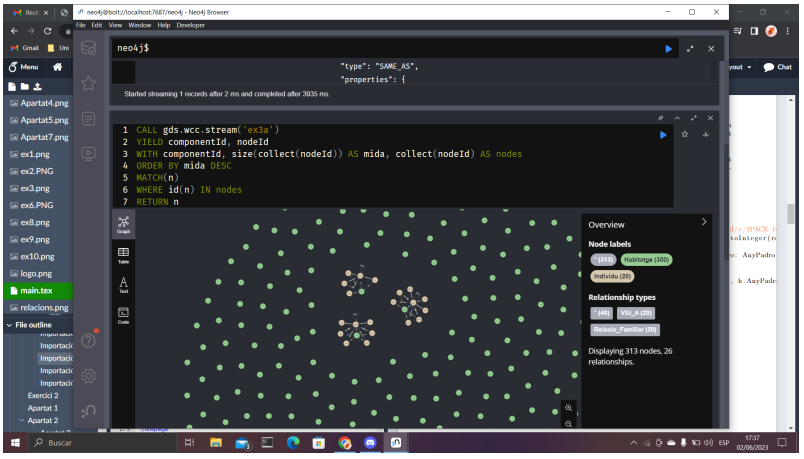
A continuació, indiquem els motius, les consultes i l'explicació dels resultats que hem fet per explorar les dades:

- **Taula agrupant els resultats segons la mida de la cc.**

  ```
  CALL gds.wcc.stream('ex3a')
  YIELD componentId, nodeId
  WITH componentId,
       size(collect(nodeId)) AS mida,
       collect(nodeId) AS nodes
  ORDER BY mida DESC
  MATCH (n)
  WHERE id(n) IN nodes
  RETURN n
  ```

  Amb aquesta cerca bàsica voliem saber les mides de les components creades per l'algorisme WCC i, a més a més, tenir un codi base per a les altres cerques.

  Els resultats que vem obtenir són:

  A partir d'aquests resultats podem veure clarament que no hi ha relació entre els habitatges. La qual cosa és obvia ja que no hi ha cap relació que conecti dos habitatges de forma directa a les dades.

- **Per cada municipi i any el nombre de parelles del tipus: (Individu)—(Habitatge).**
  Reutilitcem la projecció del graf feta al punt anterior per tal de fer la cerca del nombre de parelles (Individu)—(Habitatge) existéixen.

  La comanda en *Cypher* utilitzada per aquesta taasca és la següent:

```
CALL gds.wcc.stream('ex3a')
YIELD componentId, nodeId
WITH componentId, size(collect(nodeId)) AS mida,
     collect(nodeId) AS nodes
     ORDER BY mida DESC
MATCH(m:Individu) -[rel:VIU_A]-> (n:Habitatge)
WHERE id(n) IN nodes
return n.Municipi, n.AnyPadro, count(rel)
```

  Aquesta comanda retorna el nombre de parelles comentades abans per municipi i per any; de forma que l'ouput que genera és:

- **Quantes components connexes no estan connectades a cap node de tipus 'Habitatge'.**
  De forma similar, reutilitzarem la projecció del graf *ex3a* i utilitcem la següent comanda per a cerca la informació:

```
CALL gds.wcc.stream('ex3a')
YIELD componentId, nodeId
WITH componentId as totalCC,
     componentId,
     size(collect(nodeId)) AS mida,
     collect(nodeId) AS nodes
MATCH(h:Habitatge)
WHERE id(h) in nodes
WITH componentId, nodes, totalCC
return count(totalCC)-count(distinct componentId)
       as CCNotConnected
```

  D'aquesta forma, obtenim l'ouput que es mostra a continuació:

2.14.2 Apartat b

L'objectiu d'aquest exercici és fer un estudi de les similituds entre els nodes. Ens interessa saber quins nodes són semblants per a identificar els individus que són el mateix.
Per a fer-ho, seguim els següents passos:

1. Determineu els habitatges que són els mateixos al llarg dels anys. Afegiu una aresta amb nom "MATEIX_HAB" entre aquests habitatges. Per evitar arestes duplicades feu que la aresta apunti al habitatge amb any de padró més petit.

```
MATCH (h1:Habitatge), (h2:Habitatge)
WHERE h1 <> h2 AND h1.LlarID = h2.LlarID
                AND h1.AnyPadro < h2.AnyPadro
MERGE (h1)<-[:MATEIX_HAB]-(h2)
```

2. Creeu un graf en memòria que inclogui els nodes Individu i Habitatge i les relacions VIU, FAMILIA, MATEIX_HAB que acabeu de crear.

```
CALL gds.graph.project('ex3b',['Individu','Habitatge'],
    ['VIU_A', 'MATEIX_HAB', 'Relacio_Familiar'])
```

3. Calculeu la similaritat entre els nodes del graf que acabeu de crear, escriviu el resultat de nou a la base de dades i interpreteu els resultats obtinguts.

```
CALL gds.nodeSimilarity.stats('ex3b')
YIELD nodesCompared, similarityDistribution
```

```
CALL gds.nodeSimilarity.write('ex3b',{
writeRelationshipType:'SIMILAR',
writeProperty:'score',
similarityCutoff:0.0,
topK:3 })
YIELD nodesCompared, relationshipsWritten
```

```
CALL gds.nodeSimilarity.write('ex3b',{
writeRelationshipType:'SIMILAR',
writeProperty:'score',
similarityCutoff:0.0,
topK:3 })
YIELD nodesCompared, relationshipsWritten
```

```
match (i:Individu)-[r:SIMILAR]->(p:Individu)
return r.score, i.IndividuID, p.IndividuID
```

Cal comentar l'utilització del diversos fragments: el primer per a cercar els *stats*, el segon per a crear la relació de similitud entre nodes; i el tercer i quart per a mirar aquesta similitud.

Anàlisi dels resultats

aklñdkasldfjag

3 Treball en equip

Per tal de treballar de forma més eficient s'ha decidit separar les diverses tasques entre el grup.
D'aquesta forma, cada individu s'ha encarregat d'una part.

Cada integrant del grup s'ha encarregat de fer els següents treballs:

- Ona Sánchez:
- Manuel Arnau:
- Sofia Di Capua:
- Gerard Lahuerta:

Cada integrant del grup es va encarregar de descriure en aquest informe la part que havia realitzat.

Comentar a més, que tot el treball ha sigut documentat i treballat mitjançant la plataforma *GitHub*; per la qual cosa es pot accedir a la tasca a través de l'enllaç on també es pot observar el desenvolupament cronològic de l'entrega.