Bases De Dades No Relacionals

GRAU EN ENGINYERIA DE DADES

Projecte Neo4j

Introducció

Aquest informe es centra en explicar i documentar el disseny, la implementació i les consultes de un projecte de bases de dades no relacional utilitzant Neo4j. El projecte parteix de la transformació d'un model Entitat-Relació (E-R) en un graf de Neo4j, seguit de consultes a la base de dades en Cypher i un anàlisi extens del graf.

Part 1

La base de dades no relacional conté nodes de tipus habitatges i individus, i tres tipus de relacions:

- viu: representen el lloc on viu cada individu.
- família: relacions de parentesc entre individus que conviuen al mateix habitatge.
- same_as: els nodes que representen el mateix individu al llarg del temps.

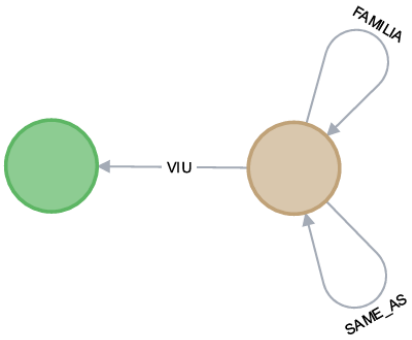


Figura 3. Esquema del graf de padrons

Exercici 1: Càrrega de Dades

Importeu les dades en la BD de Neo4j del projecte. Genera un script en cypher que carregui totes les dades, generi tots els nodes, relacions i afegeixi les característiques allà on toqui.

```
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vTfU6oJBZhmhzzkV_0-
avABPzHTdXy8851ySDbn2gq32WwaNmYxfiBtCGJGOZsMgCWjzlEGX4Zh1wqe/pub?output=csv'
as row
with(row.Id) as id, toInteger(row.Year) as born, (row.name) as name,
(row.surname) as surname1, (row.second_surname) as surname2
merge (i: Individu {id: id, born: born, name: name, surname1: surname1,
surname2: surname2});

LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/e/2PACX-
1vT0ZhR6BSO_M72JEmxXKs6GLuOwxm_Oy-
0UruLJeX8_R04KAcICuvrwn2OENQhtuvddU5RSJSclHRJf/pub?output=csv" as row
with (row.Municipi) as municipi, toInteger(row.Id_Llar) as id_llar,
toInteger(row.Any_Padro) as any_padro, (row.Carrer) as carrer, (row.Numero)
as numero
where municipi <> "null"
merge(h: Habitatge{municipi: municipi, id_llar: id_llar,
any_padro:any_padro, carrer: carrer, numero:numero});

LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vRVOoMAMoxHiGboTjCIHo2yT30CCWgVHgocGnVJxiCTgyurtmqCfAFahHajobVzwXFLwhqajz1f
qA8d/pub?output=csv' AS row
with(row.ID_1) as a, (row.Relacio_Harmonitzada) as relacio,(row.ID_2) as b
where relacio <> "null"
match (primer:Individu {id: a})
match (segon:Individu{id: b})
merge (primer)-[r:Familia {tipus: relacio}]->(segon);

LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vRM4DPeqFmv7w6kLH5msNk6_Hdh1wuExRirgysZKO_Q70L21MKBkDISIyjvdm8shVixl5Tcw_5z
Cfdg/pub?output=csv' AS row
with (row.IND) as indi, (row.Location) as muni, toInteger(row.Year) as year,
toInteger(row.HOUSE_ID) as house_id
match (i:Individu {id: indi})
match (h:Habitatge {municipi: muni, id_llar: house_id, any_padro: year})
merge (i)-[r:Viu]->(h);

LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vTgC8TBmdXhjUOPKJxyiZSpetPYjaRC34gmxHj6H2AWvXTGbg7MLKVdJnwuh5bIeer7WLUi0Oig
I6wc/pub?output=csv' AS row
with (row.Id_A) as a, (row.Id_B) as b
match (n:Individu{id: a})
```

```
match (m:Individu{id: b})
merge (n)-[r:Same_as]->(m);


create constraint UniqueIndividuId for (m:Individu) require m.id is unique;
create constraint ExistFamiliaTipus for ()-[rel:Familia]->() require
rel.tipus is not null;
create constraint UniqueHabitatgeMuniIdAny for (h:Habitatge) require
(h.municipi, h.id_llar, h.any_padro) is node key;
```

Aquest script Cypher importa dades en tres passos. Primer, crea els nodes bàsics: individus amb les seves dades personals (ID, nom, cognoms, any naixement) i llars amb la seva ubicació (municipi, carrer, número), identificats per una clau composta única.
Després, estableix les relacions: familiars entre persones, de residència entre persones i llars, i d'equivalència per a registres duplicats de la mateixa persona.
Finalment, aplica restriccions per garantir la qualitat de les dades: IDs únics per a persones, tipus obligatoris per a relacions familiars i validació de l'estructura dels llars. L'script llegeix els CSV directament des de Google Sheets, filtra dades incompletes i evita duplicats amb MERGE.

Exercici 2: Consultes

a) **Per a cada padró (any) de Castellví de Rosanes, retorna l'any de padró, el número d'habitants, i la llista de cognoms. Elimina duplicats i "nan".**

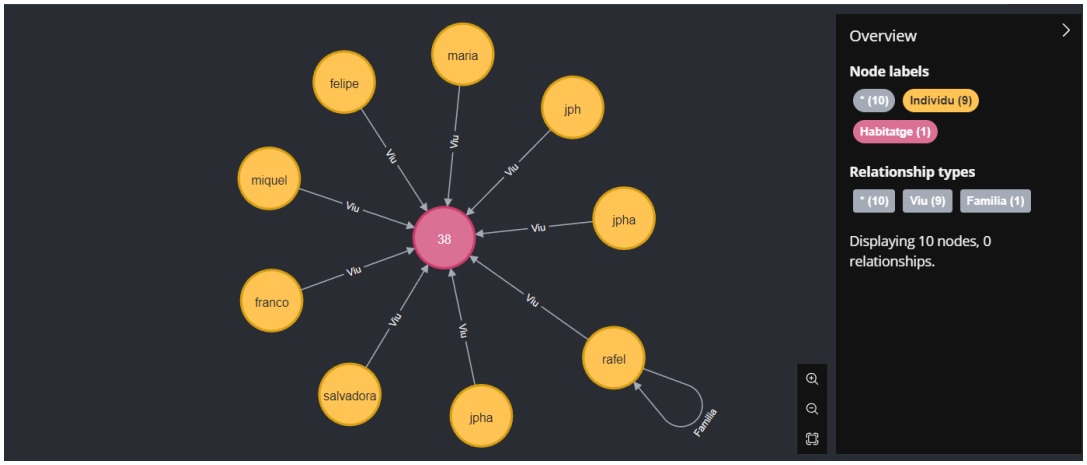
```
MATCH (i:Individu)-[:Viu]->(h:Habitatge)
WHERE h.municipi = "CR"
  AND i.surname1 IS NOT NULL
  AND i.surname1 <> "nan"
RETURN
  h.any_padro AS AnyPadro,
  COUNT(DISTINCT i.id) AS NombreHabitants,
  COLLECT(DISTINCT i.surname1) AS Cognoms;
```

	AnyPadro	NombreHabitants	Cognoms
1	1866	336	["galceran", "olle", "suñol", "rusell", "julibert", "bargallo", "anglada", "ros", "julia", "vila", "julivert", "parera", "rumeu", "gaset"

Aquesta consulta Cypher resol el problema analitzant els habitants de Castellví de Rosanes (identificat per "CR") a través de les seves relacions de residència amb els llars. Primer filtra els individus que viuen en aquest municipi, descartant els que tenen cognoms nuls o amb valor "nan". Per a cada any de padró, calcula el nombre d'habitants únics mitjançant el recompte d'identificadors distints i recull la llista de cognoms diferents presents aquell any.

b) Retorna el nom de les persones que vivien al mateix habitatge que "rafel marti" (no té segon cognom) segons el padró de 1838 de Sant Feliu de Llobregat (SFLL). Retorna la informació en mode graf i mode llista.

```
MATCH (p:Individu) -[:Viu] ->(h:Habitatge{any_padro:1838,
municipi:"SFLL"}) <-[:Viu]- (s:Individu{name:"rafel",
surname1:"marti"})
RETURN p, h, s;
```




```
MATCH (p:Individu) -[:Viu] ->(h:Habitatge{any_padro:1838,
municipi:"SFLL"}) <-[:Viu]- (s:Individu{name:"rafel",
surname1:"marti"})
RETURN h.numero, collect(p.name) as familiares, s.name as jefe;
```

	h.numero	familiares	jefe
1	"38"	["salvadora", "jpha", "jph", "maria", "felipe", "miquel", "franco", "jpha"]	"rafel"

La consulta parteix de l'individu específic (Rafel Martí, identificat pel nom "rafel" i cognom "marti") i troba tots els altres residents que apareixen vinculats al mateix node d'habitatge. La consulta filtra precisament per any (1838) i municipi (SFLL), garantint que els resultats corresponen exactament a la situació demogràfica sol·licitada. Tant el node de Rafel Martí com els dels seus veïns queden clarament identificats en ambdós modes de visualització.

c) **Retorna totes les aparicions de "miguel estape bofill". Fes servir la relació SAME_AS per poder retornar totes les instancies, independentment de si hi ha variacions lèxiques (ex. diferents formes d'escriure el seu nom/cognoms). Mostra la informació en forma de taula: el nom, la llista de cognoms i la llista de segon cognom (elimina duplicats).**

```
match (n:Individu{name:"miguel", surname1:"estape",
surname2:"bofill"}) -[:Same_as]- (p:Individu)
return count(n), collect(distinct p.name) as noms, collect(distinct
p.surname1) as cognoms, collect(distinct p.surname2) as
segons_cognoms;
```

	count(n)	noms	cognoms	segons_cognoms
1	9	["miguel"]	["estape"]	["bofill", "bufill"]

Aquesta consulta localitza totes les variants de "Miguel Estapé Bofill" mitjançant la relació SAME_AS que enllaça registres duplicats o amb variants del nom. El DISTINCT elimina repeticions, mostrant només les variants úniques de cada camp. La relació SAME_AS assegura que s'inclouen totes les instàncies, fins i tot amb escriptures diferents.

d) **Mitja de fills a Sant Feliu de Llobregat l'any 1881 per família. Mostreu el total de fills, el nombre d'habitatges i la mitja de fills per habitatge. Fes servir CALL per obtenir el nombre de llars.**

```
MATCH (h:Habitatge {municipi: "SFLL", any_padro: 1881})<-[:Viu]-
(p:Individu)<-[:Familia {tipus: "jefe"}]-(j:Individu)-[r:Familia]-
(f:Individu)
WHERE r.tipus in ["fill", "filla"]
WITH h, count(f) as fills_per_llar
RETURN count(h) as llars,sum(fills_per_llar) as total_fills,
avg(fills_per_llar) as fills_mitjans_per_habitatge;
```

	llars	total_fills	fills_mitjans_per_habitatge
1	465	1363	2.931182795698926

La consulta comença localitzant tots els habitatges del municipi aquell any concret, després identifica els caps de família a través de la relació "jefe" i els seus fills mitjançant les relacions "fill" o "filla". Filtra exclusivament les dades de Sant Feliu de Llobregat i l'any 1881.

e) Mostreu les famílies de Castellví de Rosanes amb més de 3 fills. Mostreu el nom i cognoms del cap de família i el nombre de fills. Ordeneu-les pel nombre de fills fins a un límit de 20, de més a menys.

```
match (h:Habitatge {municipi: "CR"})<-[:Viu]-(p:Individu)<-[:Familia
{tipus: "jefe"}]-(j:Individu)-[r:Familia]-(f:Individu)
where r.tipus in ["fill", "filla"]
with count(f) as fills, j, collect(f.name) as nombres where fills >3
return j.name, j.surname1, j.surname2, nombres, fills order by fills
desc limit 20;
```

	j.name	j.surname1	j.surname2	nombres	fills
1	"pablo"	"astruch"	"julia"	["pedro", "maria", "ramon", "teresa", "francisco", "rosa", "carmen"]	7
2	"jose"	"olle"	"domenech"	["antonio", "rosa", "elisa", "manuel", "miguel", "emilia"]	6
3	"benito"	"julivert"	"parera"	["jose", "magdalena", "joaquina", "juan", "dolores", "martin"]	6
4	"jose"	"canals"	"olle"	["jose", "teresa", "pedro", "catalina", "antonia", "dolores"]	6
5	"pedro"	"bargallo"	"ilegible"	["jose", "maria", "isidro", "filomena", "catalina", "francisco"]	6
6	"jose"	"canals"	"mila"	["jaime", "nan", "juan", "miguel", "lorenzo", "dolores"]	6
7					

La cerca parteix dels caps de família (marcats amb la relació "jefe") i els seus fills biològics (etiquetats com "fill" o "filla"), tot filtrant exclusivament els residents d'aquest municipi. El resultat mostra el nom complet dels pares, la llista de noms dels fills i el recompte total, ordenant les famílies de més a menys fills i limitant la sortida als 20 casos més destacats.

f) Per cada padró/any de Sant Feliu de Llobregat, mostra el carrer amb menys habitants i el nombre d'habitants en aquell carrer. Fes servir la funció min() i CALL per obtenir el nombre mínim d'habitants. Ordena els resultats per any de forma ascendent

```
MATCH (h:Habitatge {municipi: 'SFLL'})<-[:Viu]-(i:Individu)
WITH h.any_padro AS any, h.carrer AS carrer, COUNT(i) AS habitants

CALL {
  MATCH (h2:Habitatge {municipi: 'SFLL'})<-[:Viu]-(i2:Individu)
  WITH h2.any_padro AS any_sub, h2.carrer AS carrer_sub, COUNT(i2) AS
habitants_sub
  RETURN any_sub, MIN(habitants_sub) AS min_habitants
}

WITH any, carrer, habitants, any_sub, min_habitants
WHERE any = any_sub AND habitants = min_habitants

RETURN any AS AnyPadro, carrer AS CarrerAmbMenysHabitants, habitants
AS NombreHabitants
ORDER BY AnyPadro ASC
```

	AnyPadro	CarrerAmbMenysHabitants	NombreHabitants
1	1833	"carrtera de la part de molins de rey"	5
2	1838	"carretera de barna"	30
3	1839	"casas del 3onmany"	3
4	1878	"carrretera"	2
5	1881	"Carretera"	5
6	1889	"s n antonio"	1

La cerca comença agrupant els habitants per carrer i any, calculant per a cada combinació el nombre total de residents. Mitjançant una subconsulta, determina el mínim d'habitants registrat en qualsevol carrer per cada any concret. La consulta utilitza funcions d'agregació com MIN() per trobar els valors mínims i CALL per gestionar la subconsulta, assegurant que només es retornin els carrers amb la població més baixa de cada any.

Exercici 3: Analítica de Grafs

a) Estudi de les components connexes (cc) i de l'estructura de les components en funció de la seva mida. Feu servir el mode stream. Un cop calculades les components connexes (nodes individu, habitatge i relació VIU), feu dues consultes per explorar les dades. Per exemple (podeu fer-ne d'altres):

```
CALL gds.graph.project(
  'individusHabitatges',
  ['Individu', 'Habitatge'],
  {
    VIU: {
      type: 'Viu',
      orientation: 'UNDIRECTED'
    }
  }
)
```

i) Mostra, en forma de taula, les 10 components connexes més grans (ids i mida).

```
CALL gds.wcc.stream('individusHabitatges')
YIELD componentId, nodeId
WITH componentId, count(*) AS size
ORDER BY size DESC
LIMIT 10
RETURN componentId, size
```

	componentId	size
1	17402	20
2	13271	19
3	12679	19
4	12911	17
5	14716	17
6	5168	17

En aquesta consulta s'estan mostrant els components connexos més grans, en aquest cas les components corresponents a les diferents famílies pel que s'estan mostrant les famílies més grans

ii) Per cada municipi i any, mostra el nombre de parelles del tipus: (Individu)— (Habitatge).

```
MATCH (i:Individual)-[:LIVES_IN]->(h:House)
RETURN
 h.municipality AS municipi,
 h.year_padron AS any,
 COUNT(*) AS num_parelles
ORDER BY municipi, any;
```

	municipi	any	num_parelles
1	"CR"	1866	337
2	"SFLL"	1833	1433
3	"SFLL"	1838	287
4	"SFLL"	1839	1946
5	"SFLL"	1878	2745
6	"SFLL"	1881	3000

El conjunt de parelles (individu) — (habitatge) correspon al total de individus d'un padró.

iii) Quantes components connexes no estan connectades a cap node de tipus 'Habitatge', és a dir, els individus sense casa.

```
CALL gds.wcc.stream('individusHabitatges')
YIELD nodeId, componentId
WITH gds.util.asNode(nodeId) AS node, componentId
WITH componentId, collect(DISTINCT labels(node)) AS tipusNodes
WHERE NOT any(etiqueta IN tipusNodes WHERE etiqueta = 'Habitatge')
RETURN count(DISTINCT componentId) AS componentsSenseHabitatge;
```

	componentsSenseHabitatge
1	7449

Mostrant el número de components connexes d'un node, és a dir, que siguin aïllats i no tinguin habitatge.

b) Semblança entre els nodes. Ens interessa saber quins nodes són semblants com a pas previ a identificar els individus que són el mateix (i unirem amb una aresta de tipus SAME_AS). Abans de fer aquesta anàlisi:

i) 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 l'aresta apunti al habitatge amb any de padró més petit.

```
MATCH (h1:House), (h2:House)
WHERE h1.house_id = h2.house_id AND h1.year_padron < h2.year_padron
MERGE (h1)-[:MATEIX_HAB]->(h2);
```

```
Created 3955 relationships, completed after 483 ms.
```

Estem connectant els mateixos habitatges de diferents anys entre ells.

ii) 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(
  'graf_individu_habitatge',
  ['Individual', 'House'],
  {
    LIVES_IN: {
      type: 'LIVES_IN',
      orientation: 'UNDIRECTED'
    },
    IS_FAMILY: {
      type: 'IS_FAMILY',
      orientation: 'UNDIRECTED'
    },
    MATEIX_HAB: {
      type: 'MATEIX_HAB',
      orientation: 'UNDIRECTED'
    }
  }
);
```

nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	projectMillis
1					


```
{
  "Individu": {
    "label": "Individu",
    "properties": {
    }
  },
  "Habitatge": {
    "label": "Habitatge",
    "properties": {
    }
  }
}
```

```
{
  "MATEIX_HAB": {
    "aggregation": "DEFAULT",
    "orientation": "UNDIRECTED",
    "indexInverse": false,
    "properties": {
    },
    "type": "MATEIX_HAB"
  },
  "Viu": {
    "aggregation": "DEFAULT",
    "orientation":
```


	graphName	nodeCount	relationshipCount	projectMillis
1	"graf_individus_habitatges"	20314	57684	94

Estem emmagatzemant en memòria un graf amb els individus, habitatges i les seves relacions més la relació que hem creat en la consulta anterior.

iii) Calculeu la similitud 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.write('graf_individus_habitatges', {
  writeRelationshipType: 'SIMILAR_TO',
  writeProperty: 'similarityScore'
})
YIELD nodesCompared, relationshipsWritten, similarityDistribution
RETURN nodesCompared, relationshipsWritten, similarityDistribution;
```

	nodesCompared	relationshipsWritten	similarityDistribution
1	18020	137593	


```
{
  "min": 0.035714149475097656,
  "p5": 0.08333325386047363,
  "max": 1.000007629394531,
  "p99": 1.0000073909759521,
  "p1": 0.07142853736877441,
  "p10": 0.10000014305114746,
  "p90": 1.0000073909759521,
  "p50": 0.20000052452087402,
  "p25": 0.1250007152557373,
  "p75": 1.0000073909759521,
  "p95": 1.0000073909759521,
  "mean": 0.47279399869802285,
  "p100": 1.0000073909759521,
  "stdDev": 0.40693032784311345
}
```


Per calcular la similitud, el mètode compara cada parell de nodes possibles i les seves connexions. La consulta ens retorna: la quantitat de nodes comparats, les relacions de similitud entre elles i estadístiques sobre la distribució d'aquesta similitud. Llegint aquestes estadístiques podem veure que el mínim és 0.0357, el màxim 1 i la mitjana és

de 0.4727 amb una desviació estàndard de 0.4069. A més, podem veure gràcies als percentils que hi ha molts nodes amb una similitud menor o igual a 1 (p90). Amb aquesta informació podem concloure que hi ha molts nodes que són la mateixa persona al llarg del temps, per la qual cosa es podrien unir aquests nodes o crear una aresta SAME_AS entre elles.

Treball en equip

La distribució de les tasques ha sigut:

- Daniela Lou ha realitzat el primer exercici.
- David Liu ha realitzat la majoria de les consultes de l'exercici 2 y una de l'exercici 3.
- Lucia Garrido ha realitzat la majoria de les consultes de l'exercici 3 y una de l'exercici 2.
- Sergio Martínez ha realitzat el informe i documentat la càrrega de dades i les consultes de l'exercici 2.

També s'adjunta el link del repositori amb tots els scripts creats i utilitzats.