PROJECTE NEO4J : Padrons

Bases de Dades no Relacionals

Cèlia Martínez
Marta Matute
Goretti Pena
Judit Ugas

Introducció

Durant aquest projecte hem treballat la càrrega de dades i les consultes en una base de dades de grafs fent servir el software Neo4j, així com l'ús d'algorismes d'analítica de grafs a partir d'un subconjunt de dades de padrons.

Les dades han estat proporcionades pel Centre de Visió per Computació, extretes del projecte XARXES. Aquestes dades formen part del registres demogràfics de dues ciutats catalanes. Dins la base de dades hi trobem habitatges i persones, i les relacions, tant familiars com d'habitatge, que hi ha.

Podem dividir en tres parts la feina feta al llarg del projecte: la importació de les dades, la creació de consultes, i l'anàlisi del graf. Cadascuna d'aquestes parts queda explicada en detall al llarg del document.

Tota la documentació i codi relacionats amb aquest projecte es troba al repositori de GitHub Projecte-Neo4j-Grup2. Les dades utilitzades estan dins la carpeta "Data", el codi per importar les dades esta dins la carpeta "Code" i les queries i exercicis corresponents estan al repositori github.

Repartició volum de feina

En general totes hem fet una mica de tot, com es pot veure amb tots els commits del repositori.

El codi per carregar les dades va ser una tasca conjunta entre la Marta, la Cèlia i la Judit. La implementació de les queries de l'exercici 2 va ser feina, principalment, de la Marta, la Judit, la Goretti, mentre que la Cèlia va acabar de repassar aquelles queries que donaven més problemes. La implementació de l'exercici 3 es va dur a terme, generalment, per la Cèlia i la Judit. Finalment, l'informe va ser escrit per totes en general i la Marta va penjar la pràctica a la Màquina Virtual.

Exercici 1: importació de dades

La importació de les dades per a crear el graf que farem servir per a les quèries es pot trobar al GitHub, dins la carpeta "Code". Hem optat per escriure tota la informació en anglès (noms de les propietats, relacions, etc). En total, la importació de dades està formada per 8 comandes executades una darrera l'altra:

Una comanda per a assegurar-nos que la base de dades està buida:

```
match (n) detach delete n;
```

Dues comandes per a carregar tots els nodes:

```
// LOAD PEOPLE DATA
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vTfU6oJBZhmhzzkV_0-
avABPzHTdXy8851ySDbn2gq32WwaNmYxfiBtCGJGOZsMgCWjzlEGX4Zh1wqe/pub?output=csv'
AS row
CREATE (p:PERSON {Id: toInteger(row.Id), Year: toInteger(row.Year), Name: row
.name, Last_Name: row.surname, Second_Last_Name: row.second_surname});

// LOAD HOUSE DATA
```

```
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vT0ZhR6BSO_M72JEmxXKs6GLuOwxm_Oy-
0UruLJeX8_R04KAcICuvrwn2OENQhtuvddU5RSJSclHRJf/pub?output=csv' AS row
CREATE (:HOUSE {Municipality: row.Municipi, Home_Id: toInteger(row.Id_Llar),
Registry_Year: toInteger(row.Any_Padro), Street: row.Carrer, Number: toIntege
r(row.Numero)});
```

Dues comandes per a crear índexs en algunes de les propietats (això ens servirà per accelerar el temps de creació de les relacions i les quèries), una per a esborrar qualsevol índex anterior, i l'altre per a crear els que ens calen:

```
// DELTE ANY EXISTING INDEXES
CALL apoc.schema.assert({},{},true) YIELD label, key
RETURN *;

// CREATE INDEXES (FOR FASTER DATA CREATION)
create index for (p:PERSON) on (p.Id);
create index for (p:PERSON) on (p.Year);
create index for (h:HOUSE) on (h.Home_Id);
create index for (h:HOUSE) on (h.Municipality);
create index for (h:HOUSE) on (h.Registry_Year);
```

I per últim tres comandes que creen les tres relacions entre nodes que necessitem:

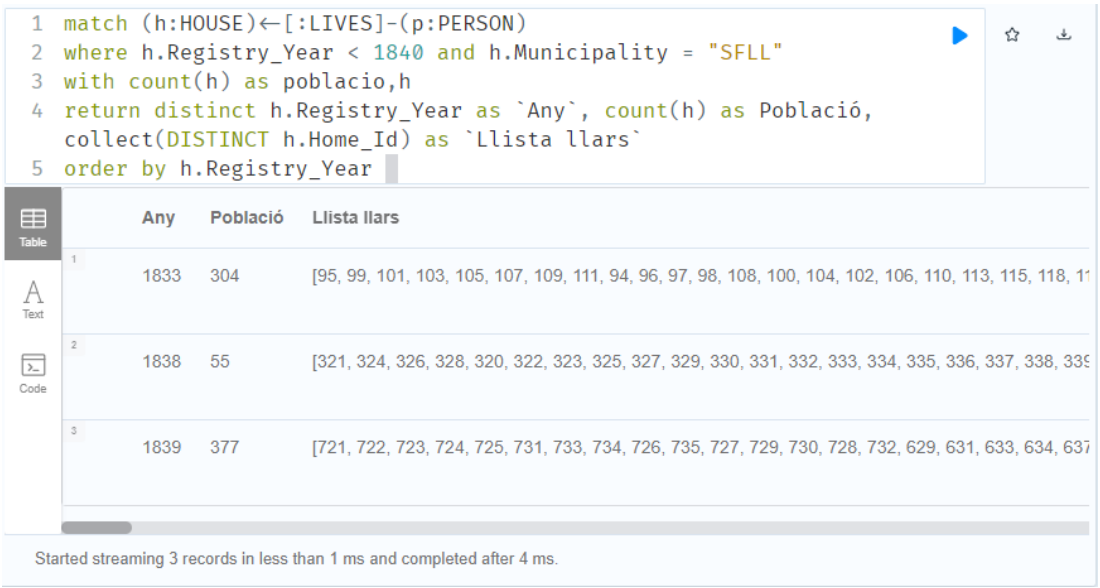
```
// CREATION OF "LIVES" RELATIONSHIP BETWEEN A PERSON AND A HOUSE
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vRM4DPeqFmv7w6kLH5msNk6_Hdh1wuExRirgysZKO_Q70L21MKBkDISIyjvdm8shVixl5Tcw_5zC
fdg/pub?output=csv' AS row
MATCH (p:PERSON {Id: toInteger(row.IND), Year: toInteger(row.Year)})
MATCH (h:HOUSE {Municipality: row.Location, Home_Id: toInteger(row.HOUSE_ID),
 Registry_Year: toInteger(row.Year)})
MERGE (p)-[:LIVES {Year: toInteger(row.Year)}]->(h);

// CREATION OF "SAME_AS" BETWEEN TWO PEOPLE
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vTgC8TBmdXhjUOPKJxyiZSpetPYjaRC34gmxHj6H2AWvXTGbg7MLKVdJnwuh5bIeer7WLUi0OigI
6wc/pub?output=csv' AS row
MATCH (p:PERSON {Id: toInteger(row.Id_A)})
MATCH (q:PERSON {Id: toInteger(row.Id_B)})
MERGE (p)-[:SAME_AS]->(q);

// CREATION OF "FAMILY" RELATIONSHIP BETWEEN TWO PEOPLE
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-
1vRVOoMAMoxHiGboTjCIHo2yT30CCWgVHgocGnVJxiCTgyurtmqCfAFahHajobVzwXFLwhqajz1fq
A8d/pub?output=csv' AS row
MATCH (p:PERSON {Id: toInteger(row.ID_1)})
MATCH (q:PERSON {Id: toInteger(row.ID_2)})
MERGE (p)-
[rel:FAMILY {Relation: row.Relacio, Harmonized_Relation: row.Relacio_Harmonit
zada}]->(q);
```

Exercici 2: Queries

Un cop s'han carregat les dades, l'exercici 2 consistia en resoldre 10 queries[1]:

1. Dels padró de 1866 de Castellví de Rosanes (CR), retorna el número d'habitants i la llista de noms. Elimina duplicats i nan.

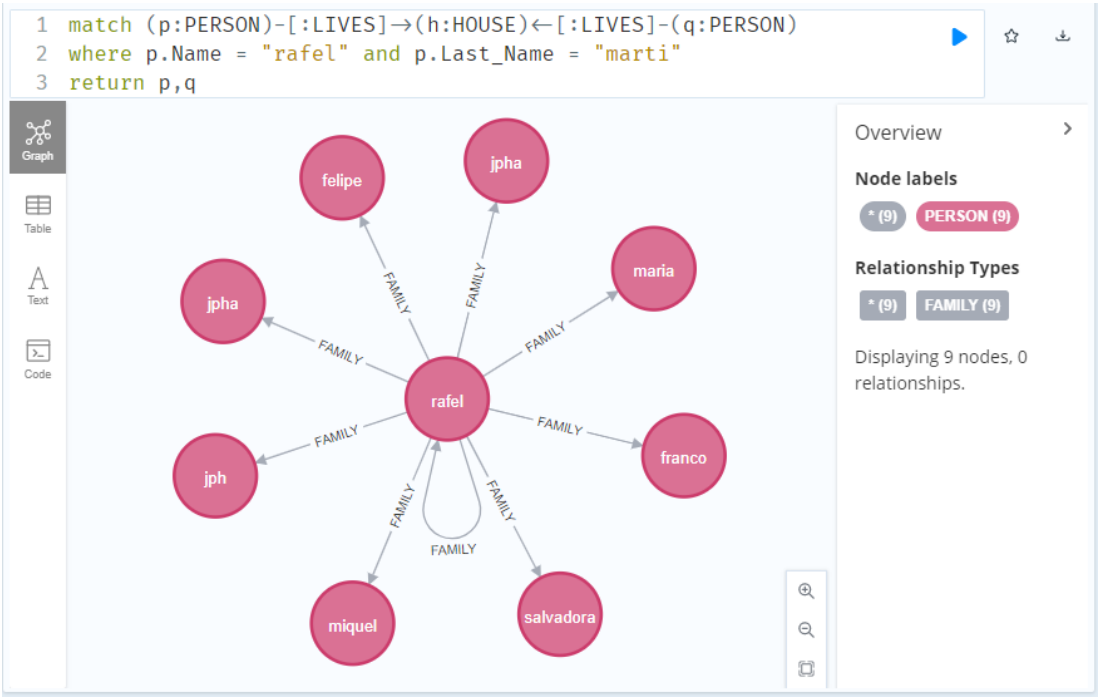
2. Dels padrons de Sant Feliu de Llobregat (SFLL) d'abans de l'any 1840 (no inclòs), retorna la població, l'any del padró i la llista d'identificadors dels habitatges de cada padró. Ordena els resultats per l'any de padró.



[1] Algunes queries s'han modificat per tal de coincidir amb els resultats del joc de proves tot mantenint l'objectiu principal d'aquestes

3. 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.

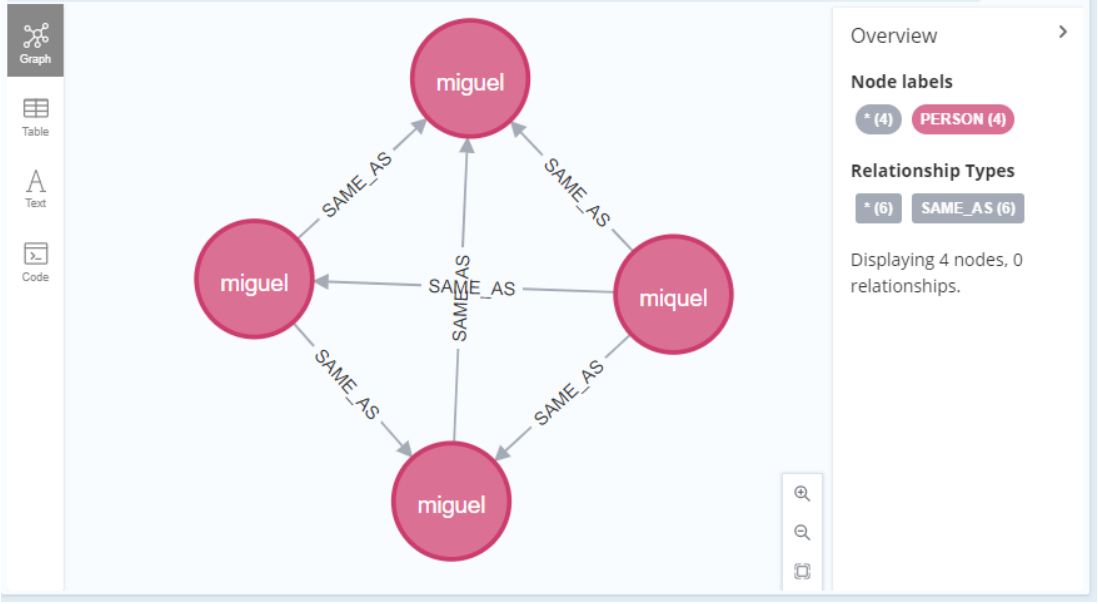
En mode de graf:



I en mode de llista:

4. Retorna totes les aparicions de "Miguel ballester". 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 subgraf.

```
match (p:PERSON)-[:SAME_AS]-(q:PERSON)
where p.Name = "miguel" and p.Last_Name = "ballester"
return p,q;
```



5. Mostra totes les persones relacionades amb "antonio farran". Mostra la informació en forma de taula: el nom, cognom1, cognom2, i tipus de relació.

```
match (p:PERSON)-[rel]-(q:PERSON)
where p.Name = "antonio" and p.Last_Name = "farran"
return q.Name as Nom, q.Last_Name as Cognom1, q.Second_Last_Name
as Cognom2, type(rel) as Tipus;
```

Table | Text | Code

	Nom	Cognom1	Cognom2	Tipus
1	"isidro"	"farran"	"colet"	"FAMILY"
2	"catalina"	"farran"	"colet"	"FAMILY"
3	"francisco"	"farran"	"colet"	"FAMILY"
4	"esperanza"	"farran"	"colet"	"FAMILY"
5	"esperanza"	"colet"	"gavarro"	"FAMILY"
6	"antonio"	"farran"	"sole"	"FAMILY"
7	"antonio"	"ferran"	"sele"	"SAME_AS"
8	"antonio"	"ferran"	"sole"	"SAME_AS"

Started streaming 8 records after 4 ms and completed after 11 ms.

6. Llisteu totes les relacions familiars que hi ha.

```
match (p:PERSON)-[r:FAMILY]-(q:PERSON)
where r.Harmonized_Relation <> 'null'
return distinct r.Harmonized_Relation as `Relacions Familiars`;
```

	Relacions Familiars
1	"esposa"
2	"fill"
3	"filla"
4	"jefe"
5	"net"
6	"gendre"
7	

Started streaming 47 records after 3 ms and completed after 35 ms.

7. Identifiqueu els nodes que representen el mateix habitatge (carrer i numero) al llarg dels anys de Sant Feliu del Llobregat (SFLL). Mostreu el resultat dels habitatges que tingueu totes dues informacions (carrer i numero), el nombre total d'habitatges, el llistat d'anys dels padrons i el llistat de les Ids de les llars. Ordeneu de més a menys segons el total d'habitatges i mostreu-ne els 10 primers.

```
match (h1:HOUSE) match (h2:HOUSE)
where h1.Street = h2.Street and h1.Number = h2.Number and
h1.Home_Id <> h2.Home_Id and h1.Municipality = "SFLL" and
h2.Municipality = "SFLL" and h1.Street <> "null" and h1.Number
<> "nan"
return h1.Street as Carrer, h1.Number as Numero, count(distinct
h2) as Total, collect(distinct h2.Registry_Year) as any,
collect(DISTINCT h2.Home_Id) as Ids
order by Total desc limit 10
```

	Carrer	Numero	Total	any	Ids
1	"falguera"	5	9	[1833, 1878, 1881, 1889]	[247, 359, 360, 361, 416, 398, 414, 415, 417]
2	"falguera"	3	8	[1833, 1878, 1881, 1889]	[245, 357, 358, 668, 411, 397, 412, 413]
3	"san antonio"	1	7	[1889]	[562, 563, 564, 565, 566, 567, 568]
4	"falguera"	22	7	[1833, 1878, 1881, 1889]	[269, 376, 377, 378, 429, 430]
5	"carretera"	28	6	[1838, 1878, 1881, 1889]	[346, 347, 34, 35, 26, 128]
6	"iglesia"	3	6	[1833, 1839, 1878, 1881, 1889]	[97, 723, 724, 516, 511, 484]

Started streaming 10 records after 1 ms and completed after 36 ms.

8. 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.

```
1 match (h:HOUSE)-[:LIVES]-(p:PERSON)-[r:FAMILY]→(q:PERSON)
2 where r.Harmonized_Relation =~ "f.*" and r.Harmonized_Relation
  <> "familiar" and h.Municipality = "CR"
3 with count(distinct q) as total, p.Name as nom, p.Last_Name as
  `1er cognom`, p.Second_Last_Name as `2n Cognom`
4 where total>3
5 return nom, `1er cognom`,`2n Cognom`, total
6 order by total desc
7 limit 20
```

Table | Text | Code

	nom	1er cognom	2n Cognom	total
1	"pablo"	"astruch"	"julia"	7
2	"jose"	"olle"	"domenech"	6
3	"benito"	"julivert"	"parera"	6
4	"jose"	"canals"	"olle"	6
5	"pedro"	"bargallo"	"ilegible"	6
6	"jose"	"canals"	"mila"	6
7				

Started streaming 20 records in less than 1 ms and completed after 2 ms.

9. Mitja de fills a Sant Feliu del Llobregat l'any 1881 per família. Mostreu el total de fills, el nombre d'habitatges i la mitja.

```
call{
    match (l:HOUSE)
    where l.Municipality='SFLL' and l.Registry_Year=1881
    return count(l.Home_Id) as NumLlars
}
match (h:HOUSE)<-[:LIVES]-(p:PERSON)-[r:FAMILY]->(q:PERSON)
where (r.Harmonized_Relation =~ "f.*" and r.Harmonized_Relation
<> "familiar") and p.Year = 1881 and h.Municipality = "SFLL"
with p as persones, count(distinct q) as nombre_fills, NumLlars
return sum(nombre_fills) as total_fills, NumLlars as num_llars,
(sum(nombre_fills)*1.0)/NumLlars as mitja;
```

Table | Text | Code

	total_fills	num_llars	mitja
1	1292	596	2.1677852348993287

Started streaming 1 records after 54 ms and completed after 68 ms.

10. Per cada any que hi ha a la base de dades, quin és el carrer amb menys habitants de Sant Feliu de Llobregat?

```
MATCH (p:PERSON)-[r:LIVES]->(h:HOUSE {Municipality: 'SFLL'})
WITH COLLECT(r.Year) AS year, p, h
UNWIND h.Street as st
WITH count(p) as per, year, st
ORDER BY per
RETURN year, COLLECT(st)[0], min(per)
ORDER BY year
```

Table | Text | Code

	year	COLLECT(st)[0]	min(per)
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

Started streaming 6 records in less than 1 ms and completed after 32 ms.

Exercici 3: Anàlisis del graf

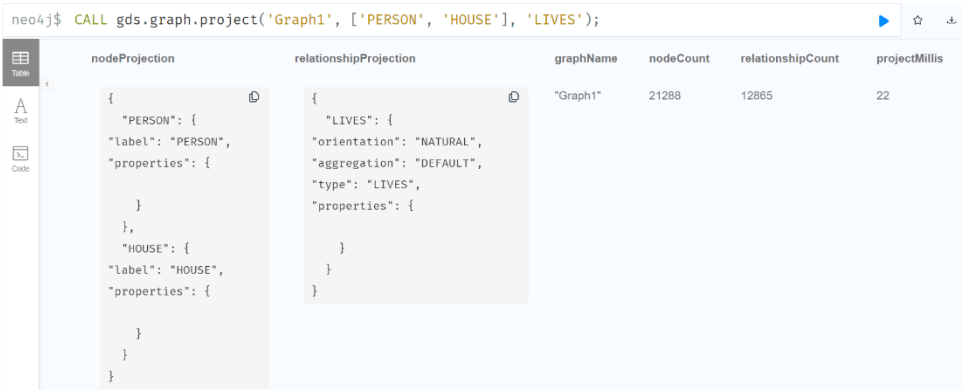
A continuació analitzarem les dades en funció de les components connexes i de les similituds entre nodes.

Apartat A

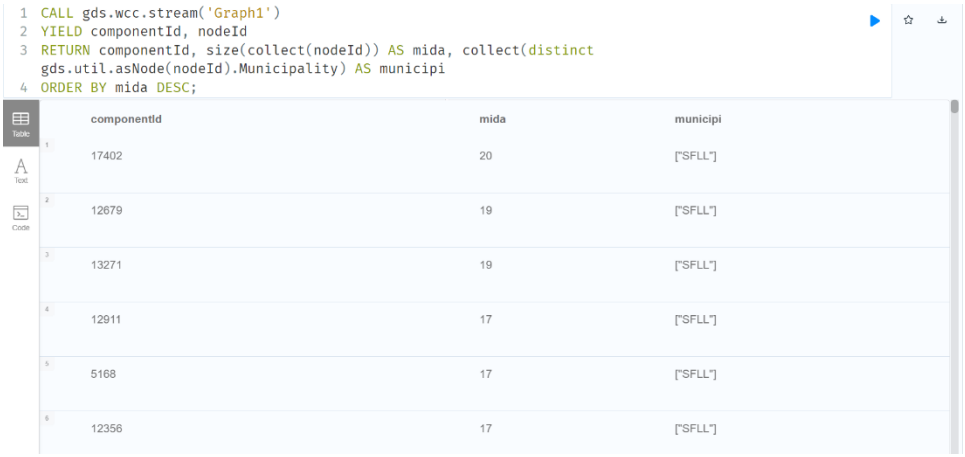
Per fer l'anàlisi sobre les components connexes ens hem plantejat una sèrie de qüestions.

La primera és sobre la mida de les components connexes, ja que ens pot donar indicacions sobre quines son les mides més habituals i per donar-nos una visió més global pels següents exercicis.

Primer hem fet un graf de connexions de la família, per tant de les propietats person I house amb la relació lives:

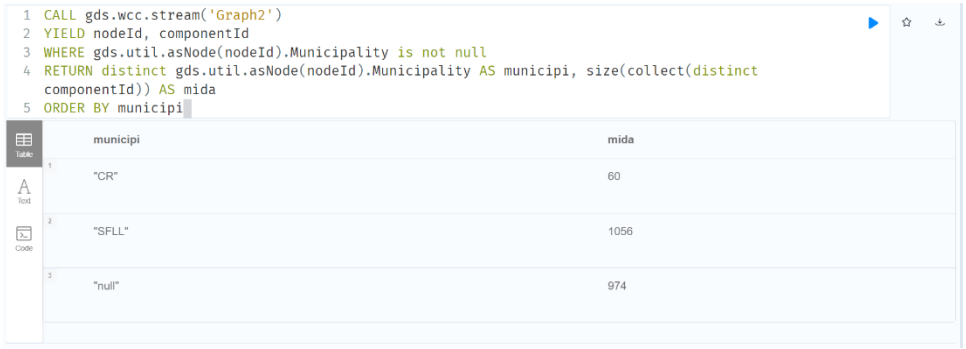


I després hem mirat el nombre de connexions de les persones, és a dir, les famílies I hem visualitzat d'on són. Es pot veure com la família més gran (les connexions entre les persones) és de 20 I és de Sant Feliu de Llobregat.

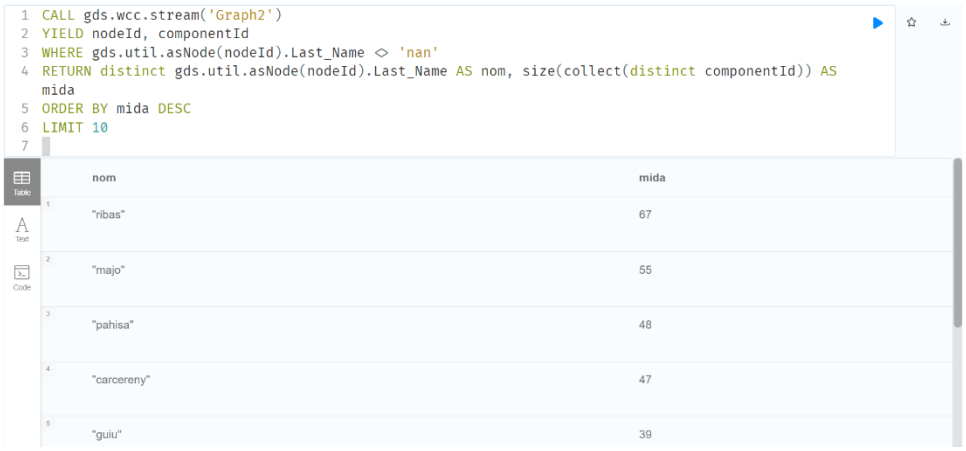


Hem creat un altre graph per veure altres connexions relacionades amb Habitatge i Persona:

Hem mirat la distribució de tipus de nodes segons la mida de la component connexa. Primer hem mirat segons el municipi de l'habitatge per tal de veure on hi ha més connexions. Es pot observar com el municipi amb més connexions és sant Feliu de Llobregat.



També hem mirat els 10 cognoms amb més connexions per tal de veure els més abundants I quines són les famílies més extenses dins els municipis del graf.



Hem mirat per cada municipi i any el nombre de parelles del tipus (Individu)-(Habitatge) per tal de veure quants habitants hi ha per cada any.

```
CALL gds.wcc.stream('Graph')
YIELD componentId, nodeId
WITH componentId, collect(nodeId) AS nodes, size(collect(nodeId)) AS mida
ORDER BY mida DESC
MATCH path=((n)⟶(h:HOUSE))
WHERE id(n) IN nodes
RETURN h.Municipality As Municipi, h.Registry_Year AS `Any`, size(collect(n.Id)) AS `Parelles
totals`
ORDER BY `Any`, Municipi ASC;
```

	Municipi	Any	Parelles totals
1	"SFLL"	1833	2713
2	"SFLL"	1838	1039
3	"SFLL"	1839	2636
4	"CR"	1866	337
5	"SFLL"	1878	8411

Finalment hem mirat quantes components connexes no estan connectades a cap node per tal de veure si cal actualitzar les dades o hi ha algun problema...

```
CALL gds.wcc.stream('Graph2')
YIELD componentId, nodeId
WITH componentId, collect(nodeId) AS nodes
MATCH (n)
WHERE (NOT (n)⟶(:HOUSE)) AND (id(n) IN nodes)
RETURN count(DISTINCT componentId) AS NumComponents
```

	NumComponents
1	4226

Apartat B

D'altra banda, hem fet l'anàlisi de similitud entre nodes. Abans de començar amb l'anàlisi hem fet uns passos previs. Primer hem creat una nova aresta entre aquells nodes 'HOUSE' que representen el mateix habitatge tot i tenir identificacions diferents. Per fer-ho hem trobat els nodes que es troben en aquesta situació I hem fet ús de la comanda 'merge', per tal de crear la relació.

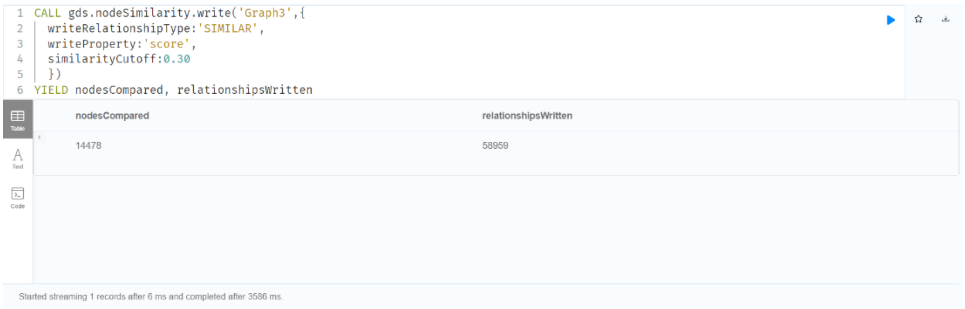
```
MATCH (h1:HOUSE),(h2:HOUSE)
WHERE h1.Home_Id <> h2.Home_Id AND h1.Number=h2.Number AND h1.Municipality=h2.Municipality AND
h1.Street=h2.Street AND h1.Registry_Year < h2.Registry_Year
MERGE (h2)-[:MATEIX_HAB]→(h1)
RETURN h1.Home_Id, h1.Number, h1.Municipality, h1.Street, h1.Registry_Year, h2.Home_Id,
h2.Number, h2.Municipality, h2.Street, h2.Registry_Year;
```

	h1.Home_Id	h1.Number	h1.Municipality	h1.Street	h1.Registry_Year	h2.Home_Id	h2.Number	h2.Municipality	h2.S
1	2	2	"SFLL"	"carretera"	1881	118	2	"SFLL"	"carr
2	321	7	"SFLL"	"carretera"	1838	5	7	"SFLL"	"carr
3	324	10	"SFLL"	"carretera"	1838	7	10	"SFLL"	"carr
4	324	10	"SFLL"	"carretera"	1838	119	10	"SFLL"	"carr
5	7	10	"SFLL"	"carretera"	1881	119	10	"SFLL"	"carr

A continuació hem creat un graf a memòria amb els nodes ‘PERSON’ i ‘HOUSE’, i les relacions ‘FAMILY’, ‘LIVES’ i ‘MATEIX_HAB’ (la nova aresta).

Un cop fet aquest procediment podem fer l’anàlisi de similitud de nodes, en aquest cas es tracta de similitud entre individus i habitatges.

Per fer-ho primer definim la relació ‘SIMILAR’ mitjançant la comanda ‘gds.nodeSimilarity.write’ per poder escriure el resultat al graf en memòria que hem creat.



Finalment visualitzem el graf en memòria que hem modificat.

(Hem especificat HOUSE com a node target, ja que d’aquesta manera ens retorna tots els tipus de relació que hi ha al graf)

Si s’executa la comanda per visualitzar les dades en forma de graf veurem que la puntuació de similitud entre els nodes és diferent quan es tracta de nodes entre individus o bé si es tracta d’una relació entre un individu I un habitatge. En el primer cas, la similitud entre individus és de 1, ja que segurament es

tracta de relacions dins d'una mateixa família. D'altra banda, en el segon cas, la similitud és de 0.5, és més baixa ja que es tracta de nodes de diferent tipus.

- Explicació raonada de cada exercici
- Treball en equip
-