

# CAS D'ÚS NEO4J: PADRONS

Resolució d'exercicis del cas pràctic

**Cruz Escalera, Cèlia (1565123)**

**Fernández Fernández, Mireia (1562636)**

**Janer Oliver, Noa (1562752)**

**Vázquez Guerrero, Marina (1563735)**

Bases de Dades no Relacionals, III MatCAD

Juny, 2022

## Índex

<i>Organització Prèvia: Treball en Equip</i>	<b>3</b>
<b>1r Exercici</b>	<b>4</b>
<b>2n Exercici</b>	<b>7</b>
<b>3r Exercici</b>	<b>12</b>
<i>Apèndix</i>	<b>19</b>

## ***Organització Prèvia: Treball en Equip***

Abans de començar a organitzar el treball cada una de nosaltres ens vam mirar separatament les instruccions del treball i les dades que se'ns proporcionen per a crear la base de dades.

- **Sobre l'exercici 1**

Aquí se'ns demana que, a partir dels fitxers de dades que se'ns ha proporcionat, instanciem la bases de dades. A l'enunciat del projecte ja hi ha una clara explicació sobre com s'han de relacionar les diferents dades entre elles, per tant, no hem hagut de pensar nosaltres com organitzar les dades.

En conseqüència, nosaltres hem considerat que era més pràctic repartir-nos les diferents instruccions de creació dels nodes, relacions i restriccions entre nosaltres, atès que les instruccions són quelcom repetitives però cada tipus és diferent entre si. Finalment ho hem ajuntat tot.

Per descomptat que, tot i que cada una feia una part, si en algun moment alguna tenia un problema les altres ens ho miràvem també per a ajudar-la a solucionar-ho.

- **Sobre l'exercici 2**

El segon exercici consisteix en resoldre un seguit de consultes que se'ns proposen. En aquest cas, hem trobat més eficient dividir-nos les consultes equitativament, sempre comentant entre nosaltres els problemes que sorgien i ajudant-nos quan a alguna no li funcionava bé.

- **Sobre l'exercici 3**

Aquest últim tros ha estat una mica diferent a la resta, atès que hi ha hagut una primera etapa de familiarització amb l'anàlisi de grafs, sobretot per al segon apartat. Seguidament, ens hem dividit en dos subgrups per a treballar en el codi demanat i, finalment, ho hem posat tot en comú per a extreure les conclusions totes juntes.

## 1r Exercici

Com que en el fitxer *Requeriments.pdf* ja se'ns fa una descripció molt detallada de com s'han de relacionar entre elles les dades que se'ns proporcionen als fitxers *csv*, nosaltres hem hagut de pensar com han de ser les instruccions *cypher* per a inserir les dades a la base de dades:

- Creem una restricció que s'asseguri que, quan creem un nou node de tipus *Individu*, el seu atribut *Id* sigui únic.

```
CREATE CONSTRAINT IndividuUnic ON (p:Individu)
ASSERT p.Id IS UNIQUE
```

- Creem tots els nodes de tipus *Individu* que podem trobar al fitxer *INDIVIDUAL.csv*. Aquests nodes tindran com a atributs el nom de la persona, el seu primer i segon cognom, un identificador *id* i l'any de naixement.

```
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/
spreadsheets/d/e/2PACX-1vTfU6oJBZhnhzzkV_0-avABPzHTdX
y8851ySDbn2gq32WwaNmYxfiBtCGJGOZsMgCWjzlEGX4Zh1wqe/
pub?output=csv' AS row
WITH toInteger(row.Id) AS id_individu, toInteger(row.Year)
AS Year, row.name AS name, row.surname AS surname,
row.second_surname AS second_surname
MERGE (p:Individu {Id: id_individu}) SET p.Year=Year,
p.second_surname=second_surname, p.surname=surname, p.name=name
RETURN count(p)
```

- Creem totes les relacions de tipus *SAME\_AS* que trobem al fitxer *SAME\_AS.csv*. Aquestes relacionen dos nodes que representen el mateix individu al llarg del temps.

```
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/
spreadsheets/d/e/2PACX-1vTgC8TBmdXhjUOPKJxyiZSpetPYj
aRC34gmxHj6H2AWvXTGbg7MLKVdJnwh5bIeer7WLUi00igI6wc/
pub?output=csv' AS row
WITH toInteger(row.Id_B) AS id_B, toInteger(row.Id_A) AS id_A
MATCH (p1:Individu), (p2:Individu)
```

```

WHERE p1.Id = id_A AND p2.Id = id_B
MERGE (p1)-[r:SAME_AS]->(p2)
RETURN count(r)

```

- Creem totes les relacions de tipus *FAMILIA* que trobem al fitxer *FAMILIA.csv*. Aquestes expliciten les relacions de parentesc entre individus que conviuen al mateix habitatge. Tenen com a atributs el nom de la relació en diferents idiomes -en concret català i castellà-.

```

LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-1vRVOoMAMoxHiGboTjCIHo2yT30CCWgVHgocGnVJxiCTgyurtmqCfAFahHajobVzwXFLwhqajz1fqA8d/pub?output=csv' AS row
WITH toInteger(row.ID_1) AS id_1, toInteger(row.ID_2) AS id_2, row.Relacio AS Relacio, row.Relacio_Harmonitzada AS Relacio_harmonitzada
MATCH (p1:Individu), (p2:Individu)
WHERE p1.Id = id_1 AND p2.Id = id_2
MERGE (p1)-[r:FAMILIA]->(p2)
SET r.relacio = Relacio,
    r.relacio_harmonitzada = Relacio_harmonitzada
RETURN count(r)

```

- Creem una altra restricció que s'asseguri que, quan creem un nou node de tipus *Habitatge*, el conjunt dels seus atributs *Id\_Llar*, *Municipi*, i *Any\_Padro* siguin únics.

```

CREATE CONSTRAINT HabitatgeUnic
ON (h:Habitatge) ASSERT (h.Id_Llar, h.Municipi, h.Any_Padro)
IS UNIQUE

```

- Creem tots els nodes de tipus *Habitatge* que podem trobar al fitxer *HABITATGES.csv*. Aquests nodes tindran com a atributs l'identificador de l'habitatge, l'any del padró i el carrer, el número i el municipi on es troba.

```

LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-1vT0Zhr6BSO_M72JEmxXKs6GLu0wx

```

```

m_Oy-0UruLJeX8_R04KAcICuvrwn20ENQhtuvddU5RSJSclHRJf/
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, toInteger(row.Numero) AS Numero
MERGE (h:Habitatge {Id_Llar : Id_Llar, Municipi : Municipi,
Any_Padro: Any_Padro }) SET h.Carrer=Carrer, h.Numero = Numero
RETURN count(h)

```

- Creem totes les relacions de tipus *VIU* que trobem al fitxer *VIU.csv*. Aquestes relacionen cada habitatge amb els individus que hi viuen segons l'any del padró.

```

LOAD CSV WITH HEADERS FROM 'https://docs.google.com/
spreadsheets/d/e/2PACX-1vRM4DPeqFmv7w6kLH5msNk6_Hdh1
wuExRirgysZK0_Q70L21MKBkDISIyjvdm8shVixl5Tcw_5zCfdg/
pub?output=csv' AS row
WITH toInteger(row.Year) AS Year, row.Location As
Location, toInteger(row.HOUSE_ID) AS house_id,
toInteger(row.IND) AS individu_id
MATCH (h:Habitatge), (p:Individu)
WHERE p.Id = individu_id AND h.Any_Padro = Year
AND h.Municipi = Location AND h.Id_Llar = house_id
MERGE (p)-[r:VIU]->(h)
RETURN count(r)

```

## 2n Exercici

En aquesta secció es mostren les consultes implementades per al nostre disseny de la base de dades. Cadascuna inclou el codi i una breu explicació de la idea principal de la resolució. A més, a l'Apèndix es troben les sortides per consola que ens ha retornat el codi per tal de mostrar que el seu funcionament és correcte.

1. Número d'habitants i llista de noms del padró de 1866 de Castellví de Rosanes (CR).

```
MATCH (h:Habitatge)-[:VIU]-(p:Individu)
WHERE h.Any_Padro = 1866 AND h.Municipi = "CR" AND
p.surname <> "nan"
RETURN count(p.surname) AS `Número d'habitants`,
collect(distinct p.surname) AS Habitants
```

Aquesta primera consulta és molt senzilla, filtrem l'any del padró, el municipi i també pels noms que no tenen cognoms.

2. Població, any del padró i llista d'identificadors dels padrons de Sant Feliu de Llobregat d'abans del 1840 (no inclòs).

```
MATCH (h:Habitatge)
WHERE h.Any_Padro < 1840 AND h.Municipi = "SFLL"
RETURN h.Municipi AS `Població`, h.Any_Padro As
`Any del padró`, collect(h.Id_Llar) AS `Identificador de
l'habitatge`
ORDER BY `Identificador de l'habitatge`
```

La consulta és molt directa, filtrem per l'any de padró i pel municipi i després retornem el que ens demana l'enunciat.

3. Persones que vivien al mateix habitatge que *rafel marti* segons el padró de 1838 de Sant Feliu de Llobregat (SFLl).

Resultat com a taula:

```
MATCH (p1:Individu)-[:VIU]-(h:Habitatge)-[:VIU]-(p2:Individu)
WHERE p1.name = "rafel" AND p1.surname = "marti" AND
h.Any_Padro = 1838 AND h.Municipi = "SFLl"
RETURN collect(p2.name) AS Noms
```

Resultat com a graf:

```
MATCH path = (p1:Individu)-[:VIU]-(h:Habitatge)
-[:VIU]-(p2:Individu)
WHERE p1.name = "rafel" AND p1.surname = "marti"
AND h.Any_Padro = 1838
AND h.Municipi = "SFLl"
RETURN path
```

En ambdós casos el procediment és el mateix, és basa en filtrar amb el *WHERE* el nom *rafel marti* a partir d'un any i un municipi determinat.

4. Retorna totes les aparicions de *miguel ballester*. Fes servir la relació *SAME\_AS* per poder retornar totes les instàncies, independentment de si hi ha variacions lèxiques. Mostra la informació en forma de subgraf.

```
MATCH (p1:Individu{})-[:SAME_AS]->(p2:Individu)
WHERE toLower(p1.name) STARTS WITH "mi"
AND toLower(p1.surname) STARTS WITH "ballest"
RETURN p1,p2
```

La idea principal és trobar aquells nodes que contenen informació sobre en Miguel Ballester, amb totes les variacions en el nom possibles. Per això hem fet un filtratge on busquem nodes que comencin amb la informació donada, atès que podríem tenir *Miguel*, *Miquel*, etc. i similar amb el cognom.



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 (p1:Individu)-[r]->(p2:Individu)
WHERE p1.name= "antonio" AND p1.surname = "farran"
RETURN p2.name AS nom, p2.surname AS cognom1,
       p2.second_surname AS cognom2,
       type(r) AS tipus_relacio
```

Aquí simplement hem cercat tots els individus relacionats amb la persona demanada i hem retornat la informació amb els noms assignats.

6. Llisteu totes les relacions familiars que hi ha.

```
MATCH (:Individu)-[r:FAMILIA]-(:Individu)
WHERE NOT r.relacio_harmonitzada = "null"
RETURN collect(DISTINCT r.relacio_harmonitzada)
       AS relacions_familiars
```

Aquí fem un llistat de relacions familiars combinant les comandes *collect* i *distinct*. Abans però hem tret la relació *null* perquè això només significa que no se sap la relació real.

7. Nodes que representen el mateix habitatge (carrer i numero) al llarg dels anys de Sant Feliu del Llobregat (SFLl). Ordenats de més a menys segons el total d'habitatges, només es se'n mostren 10.

```
MATCH (h:Habitatge {Municipi:'SFLl'}), (h1: Habitatge
{Municipi: 'SFLl'})
WHERE h.Carrer = h1.Carrer AND h.Numero = h1.Numero
RETURN h.Carrer as Carrer, h.Numero as Número,
       size(collect(DISTINCT(h.Any_Padro))) as Número_total_habitatges,
       collect(DISTINCT(h.Any_Padro)) as llistat_anys,
       collect(DISTINCT(h.Id_Llar)) as llistat_Ids
ORDER BY Número_total_habitatges DESC,h.Carrer LIMIT 10
```

Per a identificar dos habitatges iguals no podem fer-ho amb la *Id.Llar*, ja que aquest es crea a partir del Municipi on es troba i de l'any del padró, per a fer-ho hem de tenir en compte el municipi, el carrer i el número de l'habitatge.

8. Famílies de Castellví de Rosanes amb més de 3 fills ordenades de més a menys segons el nombre de fills, se'n mostren 20.

```
MATCH (h:Habitatge{Municipi:'CR'})<-[:VIU]-(p:Individu)
-[fcap:FAMILIA]-(p)-[ffill:FAMILIA]->(p2:Individu)
WHERE (ffill.relacio_harmonitzada="fill" OR
ffill.relacio_harmonitzada="filla") AND
fcap.relacio_harmonitzada="jefe"
WITH p, size(collect(distinct p2)) AS numfills
WHERE numfills > 3
RETURN p.name AS nom, p.surname AS primer_cognom,
p.second_surname AS segon_cognom, numfills AS total
ORDER BY total DESC LIMIT 20
```

Per a saber el número de fills de les famílies n'hi ha prou amb identificar el cap de la família i contar quants individus estan relacionats amb ell, segons el tipus de relació harmonitzada fill o filla.

9. Mitja de fills a Sant Feliu del Llobregat l'any 1881 per família.

```
MATCH (h: Habitatge{Municipi:"SFLL", Any_Padro:1881})-
[:VIU]-(p1:Individu)
CALL {
  WITH p1
  OPTIONAL MATCH (p1)-[f:FAMILIA
    {relacio_harmonitzada:"fill"}]-> (p2:Individu)
  RETURN p2
}
UNION
  WITH p1
  OPTIONAL MATCH (p1)-[f:FAMILIA
    {relacio_harmonitzada:"filla"}]-> (p2:Individu)
```

```
RETURN p2}
```

```
RETURN count(distinct(h)) AS total_llars, count(p2) AS total_fills,  
toFloat(count(distinct(p1)))/count(distinct(h)) AS mitja
```

Necessitem fer un *OPTIONAL MATCH* per a poder no fer el filtratge de les famílies que no tenen fills ni filles, ja que volem fer la mitja incloent aquelles famílies que no en tenen. Amb la *UNION* tenim en compte tant els fills com les filles. D'aquesta manera, aconseguim que quan fem el *RETURN*, tenim les diferents llars i els fills/es.

10. Carrer amb menys habitants de Sant Feliu de Llobregat per cada any de la base de dades.

```
MATCH (p:Individu)-[:VIU]-(h:Habitatge{Municipi: "SFL"})  
WITH count(DISTINCT(p)) AS pers, h.Any_Padro AS any_padro,  
h.Carrer AS carrer  
WITH min([pers, carrer]) AS lst, any_padro  
RETURN any_padro, lst[1] AS carrer, lst[0] AS num_habitants
```

Atès que hem de trobar el mínim de persones a cada carrer, ho hem fet a partir del *WITH*; creant com una matriu a partir de la funció *min*, per tal d'obtenir el desitjat. Finalment, accedim als resultats de cadascuna de les dues columnes de la matriu per a obtenir els carrers i el nombre mínim dels habitants per a cada carrer.

### 3r Exercici

En aquest exercici analitzarem les dades del graf per entendre millor l'estructura de les dades. Consta de dos apartats que ens han servit per orientar-nos en l'ús de les eines que ofereix Neo4j.

- (a) Per a començar, farem un petit estudi de les components connexes i de la seva estructura en funció de la mida. Hem realitzat 4 consultes diferents utilitzant la llibreria *Graph Data Science*.

La primera consulta, per tal d'entendre les dades, buscarà ordenar les unitats familiars segons el nombre d'individus que les componen. Primer, però, cal que creem del graf per a poder agrupar els nodes per famílies:

```
CALL gds.graph.project.cypher('graf_num_families',
  'MATCH (p:Individu) RETURN id(p) AS id',
  'MATCH (p1:Individu)-[f:FAMILIA{relacio_harmonitzada:"jefe"}]
  -(p1)-[:FAMILIA]-(p2:Individu) RETURN id(p1) AS source,
  id(p2) AS target, "familiars" as type, count(*) AS weight')
```

Pensarem en una unitat familiar com a el cap de família i els seus familiars, cosa que ens permet comptabilitzar fàcilment el número de familiars que hi ha en les famílies, agrupant les dades en una taula, havent fet ús de l'algorisme *Weakly Connected Components*.

```
CALL gds.wcc.stream('graf_num_families')
YIELD componentId, nodeId
RETURN componentId, size(collect(nodeId)) AS mida,
  collect(nodeId) AS nodes ORDER BY mida DESC LIMIT 3
```

Per la gran quantitat de famílies que emparen les dades, hem decidit que com a mostra de resultats limitariem la sortida a les tres majors, és a dir, les tres famílies que estan formades per més persones. La sortida que obtenim és la següent:

	componentId	mida	nodes
1	7752	24	[7752, 7753, 7754, 7755, 7756, 7757, 7758, 7759, 7760, 7761, 7762, 7763, 7764, 7765, 7766, 7767, 7768, 7769, 7770, 7771, 7772, 7773, 7774, 7775]
2	17402	19	[17402, 17403, 17404, 17405, 17406, 17407, 17408, 17409, 17410, 17411, 17412, 17413, 17414, 17415, 17416, 17417, 17418, 17419, 17420]
3	7281	18	[7281, 7282, 7283, 7284, 7285, 7286, 7287, 7288, 7289, 7290, 7291, 7292, 7293, 7294, 7295, 7296, 7297, 7298]

Figura 1: Sortida de l'anàlisi de les components connexes

Veiem que les famílies més grans estan formades per 24, 19 i 18 individus, respectivament. Per a entendre com pot ser que hi hagués famílies tan grans, cal analitzar una mica el [context històric](#).

Resumidament, podem concloure que donat el retard en el creixement de l'economia, respecte a la resta del continent, va fer que es tingués encara una visió dels fills com a eines d'ajuda per econòmica i de mà d'obra, sobretot donada la fomentació de la indústria tèxtil, on treballaven sobretot dones i nens, en la zona de Barcelona i arreu.

Ara, a partir del graf de la consulta anterior, podem afegir un *MATCH* amb els habitatges per a poder veure les connexions entre els nodes.

```
CALL gds.wcc.stream('graf_num_families')
YIELD nodeId, componentId
WITH componentId, collect(nodeId) AS nodes, size(collect(nodeId))
    AS mida ORDER BY mida DESC LIMIT 3
MATCH (n)--(h:Habitatge)
WHERE id(n) IN nodes
RETURN n, h
```

Obtenim per sortida el graf de la figura 2. Allà es pot veure la composició d'una família on hi ha un cap de família, ja que hi ha una relació recíproca i on hi ha una sèrie de diferents persones unides al mateix habitatge i també entre ells.

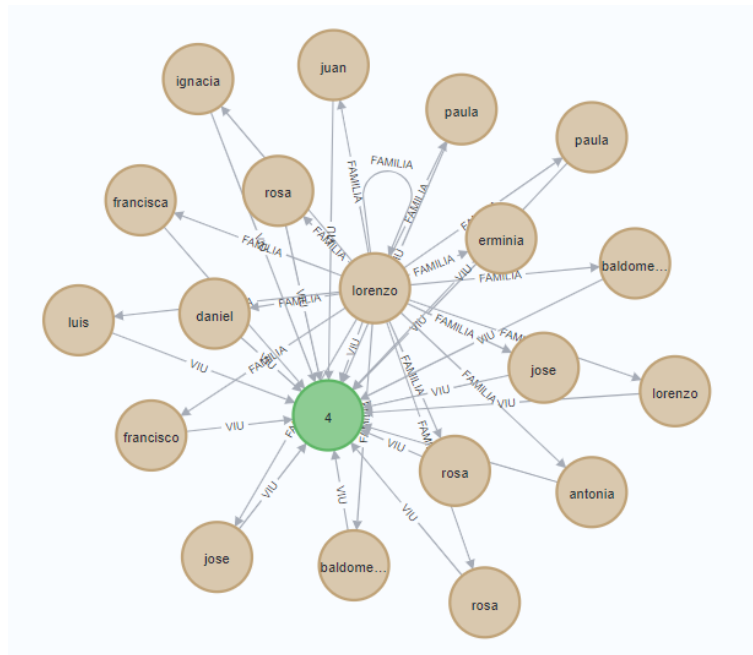


Figura 2: Sortida de d'un exemple d'anàlisi de les unitats familiars - per a treballar les components connexes- de la base de dades

Per a la tercera consulta volem buscar el número de relacions (Individu)--(Habitatge) per cada municipi i any de padró, així podrem entendre millor com estava distribuïda la població en aquella època. Abans de res, creem un graf que relacioni els individus amb els habitatges i, seguidament, fem la consulta desitjada.

```
CALL gds.graph.project.cypher('graf_num_relacions',
  'MATCH (p) RETURN id(p) AS id',
  'MATCH (p1:Individu)--(h:Habitatge) RETURN h.Id_Llar AS source,
  id(p1) AS target, "relacio" as type, count(*) AS weight')

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

```
RETURN n.Municipi AS municipi, n.Any_Padro AS any_padro,
COUNT(*) AS num_relacions ORDER BY any_padro
```

Obtenim la següent taula de resultats:

"municipi"	"any_padro"	"num_relacions"
"null"	1828	426
"SFLl"	1833	304
"null"	1833	9
"SFLl"	1838	55
"SFLl"	1839	377
"null"	1857	533
"CR"	1866	60
"null"	1866	3
"SFLl"	1878	671
"null"	1878	2
"SFLl"	1881	596
"null"	1881	1
"SFLl"	1889	645

Figura 3: Agrupació de relacions existents per municipi, segons l'any del padró

És pot observar que a mesura que passen els anys hi ha zones on disminueix la seva població en algun dels períodes. Per exemple, Sant Feliu del Llobregat va disminuir de 304 a 55 en un període de 5 anys. Aquest canvi segurament va ser causat per l'increment en el flux de migració cap a zones més industrials.

Cal destacar, però, que hi ha molta falta d'informació en la nostra base de dades i això produeix que hi hagi una sèrie de *nulls* en la nostra sortida que són significatius per a valorar quina informació manca més: a tall d'exemple,

podem observar que de l'any 1828 no es conserva informació sobre en quin municipi vivia la gent, que podria venir donat pel fet que llavors els arxius fossin purament locals i no fos necessari registrar el municipi de residència.

L'últim petit anàlisi que farem respecte a les components connexes, té com a objectiu mirar si hi ha rellevància de sexe entre els caps de família, funció de l'any d'enregistrament del padró.

Per tal de poder fer-ho, necessitem un graf simple que uneixi l'individu amb un habitatge i que es compleixi la condició que sigui *jefe*, atribut que nombra a una persona com al cap de la família.

```
CALL gds.graph.project.cypher('graf_cap3',  
  'MATCH (p:Individu) RETURN id(p) AS id',  
  'MATCH (p:Individu)-[f:FAMILIA{relacio_harmonitzada:"jefe"}]  
  -(p)-[:VIU]-(h:Habitatge) RETURN id(p) AS source,  
  h.Id_Llar AS target, "cap" as type, count(*) AS weight')
```

Seguidament, executem la consulta que ens retornarà les dades que necessitem:

```
CALL gds.wcc.stream('graf_cap3')  
YIELD componentId, nodeId  
WITH componentId, collect(nodeId) AS nodes, size(collect(nodeId))  
  AS mida  
MATCH (n:Individu) WHERE id(n) IN nodes  
RETURN size(collect(n.name)) AS num_caps, collect(DISTINCT(n.name))  
  AS caps, n.Year AS any ORDER BY any
```

Un exemple de sortida d'aquesta consulta es troba a l'apèndix, on es mostren els 287 noms dels caps de família de 1838.

Amb la seva sortida, podem veure que hi ha força dones que eren caps de família, cosa que ens ha sorprès per l'època. També hem vist que a l'últim any del qual disposem informació, hi ha més caps de família anteriorment. Això podria ser degut a que ens falten o s'han perdut dades.



- (b) Per últim, volem fer un anàlisi de la semblança entre els nodes de la base de dades. Per a aconseguir-ho, però, caldrà primer preparar la base de dades i inicialitzar un graf a memòria que ens permeti treballar-hi a sobre.

En primer lloc, afegim una nova relació entre habitatges, anomenada *MATEIX\_HAB* que indiqui quins són el mateix al llarg del temps. Per a evitar arestes duplicades farem que l'aresta apunti a l'habitatge amb any de padró més petit:

```
MATCH (h1:Habitatge), (h2:Habitatge)
WHERE NOT h2.Carrer = "null" AND NOT h2.Numero = 'null'
      AND h1.Carrer = h2.Carrer AND h1.Numero = h2.Numero AND
      NOT h1.Id_Llar = h2.Id_Llar AND h1.Any_Padro < h2.Any_Padro
MERGE (h1)<-[:MATEIX_HAB]-(h2)
RETURN h1, h2
```

Ara creem un graf en memòria que ens inclogui els nodes de tipus *Individu* i *Habitatge*, juntament amb les relacions *VIU*, *FAMILIA*, *MATEIX\_HAB*:

```
CALL gds.graph.project('3b', ['Individu', 'Habitatge'],
  ['FAMILIA', 'MATEIX_HAB', 'VIU'])
YIELD graphName AS Nom_Graph, nodeProjection, nodeCount AS Nodes,
      relationshipCount AS Relacions
```

Un cop hem obtingut aquest graf, ara podem treballar-hi a sobre per a analitzar el comportament de les dades. Procedim a calcular la similitud entre els nodes, on ens interesaran els els nodes amb una semblança que arribi mínim al percentil 75 de la *similitud de Jaccard*:

```
CALL gds.nodeSimilarity.write("3b", {similarityCutoff: 0.75,
  degreeCutoff: 1, writeProperty: "similitud",
  writeRelationshipType: "SAME_AS"})
```

On ahora hem inserit el resultat de la mètrica calculada de nou a la base de dades. Això ens permet els resultats usant aquesta comanda:

```
MATCH (p1:Individu)-[r:SAME_AS]-(p2:Individu)
WHERE exists(r.similitud)
```

```
RETURN p1.name AS nom_persona1, p2.name AS nom_persona2,
       r.similitud AS similitud ORDER BY similitud DESC
```

Que ordenant descendentment se'ns retorna la següent sortida:

"nom_persona1"	"nom_persona2"	"similitud"
"rosa"	"emilia"	1.0
"rosa"	"antonio"	1.0
"rosa"	"miguel"	1.0
"rosa"	"miguel"	1.0
"rosa"	"elisa"	1.0
"rosa"	"rosa"	1.0
"rosa"	"manuel"	1.0
"rosa"	"manuel"	1.0
"rosa"	"elisa"	1.0
"rosa"	"rosa"	1.0

Figura 4: 10 primers resultats per a la similitud entre nodes.

Es pot deduir fàcilment que tots aquells nodes que representen a la mateixa persona, seran intrínsecament iguals.

Pel mateix raonament, aquells nodes que representen individus que pertanyen a la mateixa unitat familiar, seran considerats perfectament similars, segons l'escala de *Jaccard*, ja que comparteixen la mateixa informació respecte el seu habitatge, i la seva família.

Vist l'arc temporal del qual està composta la base de dades -tenim informació de 1828 a 1889, és a dir, informació de diversos moments temporals compresos en 61 anys-, creiem que si poguéssim disposar de més dades, buscant similituds podríem obtenir resultats més informatius, atès que les dades actuals tenen una estructura molt rígida que, majoritàriament, conté tota l'estona les mateixes unitats familiars.

## Apèndix

Tot el codi del treball es troba en aquest [repositori de Github](#).

### Exemple de sortides solució per a l'Exercici 3a)

```
[ "jose", "llorenzo", "franco", "juan", "rosa", "patronilla", "narsis",
  "eulalia", "çonxita", "anton", "jph", "çecilia", "çelestino",
  "silvestre", "jaume", "cristina", "maria", "jacinto", "madalena",
  "teresa", "anna", "pere", "beatris", "isidro", "rafel", "raymunda",
  "tomas", "gracia", "jpha", "paula", "dolores", "franca", "ramon",
  "domingo", "mariano", "luis", "llorens", "çarme", "lucia", "rarimi",
  "miquel", "rufino", "benet", "gertrudias", "margarida", "narcisa",
  "antonia", "sebastia", "juaquim", "paco", "çivina", "l·lucia",
  "çliment", "ilari", "quirse", "pau", "nan", "damia", "mariana",
  "jph anton", "ignasia", "manuel", "isabel", "rita", "blay", "isavel",
  "jacinta", "sarafina", "alejandro", "madrona", "rarimio", "bartomeu",
  "miguel", "leonor", "ana", "mar", "feliu", "l·luis", "sadurni", "angela",
  "felipa", "salvadora", "felipe", "barvara", "vicenta", "ramona",
  "selma", "andreu", "çliment", "justino"]
```

### Sortides solució de les consultes

"Número d'habitants"	"Habitants"
336	["olle", "galceran", "rusell", "suñol", "julibert", "bargallo", "anglada", "vila", "ros", "julia", "julivert", "gaset", "rumeu", "parera", "canales", "dalmases", "canald", "rigol", "calaf", "llimona", "astruch", "rafuls", "astrade", "esparchy", "volta", "farre", "llopert", "gallofre", "masana", "fqrre", "farres", "casanovas", "amat", "anducas", "capellades", "plana", "cadafaly", "farreny", "jarrey", "ilegible", "ventura", "domenech", "aregay", "gibert", "vives", "marcade", "cardus", "pañella", "tovella", "mitjans", "rios", "galofre", "calafi", "nicolau", "pujol", "anmatller", "rabantos", "alegre", "bley", "armengol", "bogoña", "gol", "roca", "salto", "pujadas", "pascual", "valls", "esteva", "juntanet", "garriaga", "cartro", "rafols", "claramunt", "escayol"]

Figura 5: Sortida de la consulta 1

"Població"	"Any del padró"	"Identificador de l'habitatge"
"SFLL"	1833	[95,99,101,103,105,107,109,111,94,96,97,98,108,100,104,102,106,110,113,115,118,119,120,122,124,126,128,129,130,132,134,135,136,137,140,142,144,112,114,116,117,121,123,125,127,131,133,138,139,141,143,145,10,11,15,16,18,20,22,24,26,28,32,34,35,36,38,39,40,44,46,47,50,51,52,56,57,58,59,12,14,17,19,21,23,25,27,28,30,31,33,37,41,42,43,45,48,49,53,54,55,13,60,61,64,65,66,67,68,72,73,74,75,76,77,78,80,81,82,83,86,88,89,91,93,62,63,69,70,71,76,84,85,87,90,92,146,147,148,156,149,151,152,154,155,157,159,161,162,164,165,167,150,153,158,160,163,166,168,169,170,171,173,174,176,178,180,172,175,179,177,181,182,184,186,188,190,192,193,194,197,198,200,204,205,207,208,209,210,183,185,187,189,191,195,196,199,201,202,203,206,211,212,213,214,215,220,222,216,217,218,221,230,231,219,223,224,225,226,227,228,233,239,232,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,253,258,260,261,250,251,252,254,255,256,257,259,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,278,284,277,279,280,281,282,283,285,286,287,288,289,290,291,292,293,294,295,296,302,309,310,311,312,313,297,298,299,300,301,303,304,305,306,307,308]
"SFLL"	1838	[321,324,326,328,320,322,323,325,327,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,314,315,316,317,318,319]
"SFLL"	1839	[721,722,723,724,725,731,733,734,726,735,727,729,730,728,732,629,631,633,634,637,638,640,642,645,647,648,649,651,653,630,632,635,636,639,641,643,644,646,650,652,565,473,474,567,569,475,477,571,480,482,484,485,486,487,488,490,491,496,497,498,500,503,504,506,507,460,554,461,553,462,555,463,556,464,557,465,558,466,559,467,560,468,561,469,562,470,563,471,564,472,566,476,478,479,481,483,489,492,493,494,495,499,501,502,505,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,550,551,552,568,570,602,603,604,610,608,616,618,620,624,605,606,607,609,611,612,613,614,615,617,619,621,622,623,625,626,627,628,736,739,745,743,737,738,740,741,742,744,654,655,656,657,661,658,659,660,662,663,664,666,671,668,670,672,667,669,572,573,574,575,576,577,578,579,580,581,582,583,585,586,587,588,589,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,554,590,591,592,593,594,595,596,597,598,599,600,601,665,673,674,676,677,678,679,675,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720]

Figura 6: Sortida de la consulta 2

```
"collect (p2.name) "
```

```
["jpha","felipe","maria","jpha","franco","jph","miquel","salvadora"]
```

Figura 7: Sortida de la consulta 3 com a taula

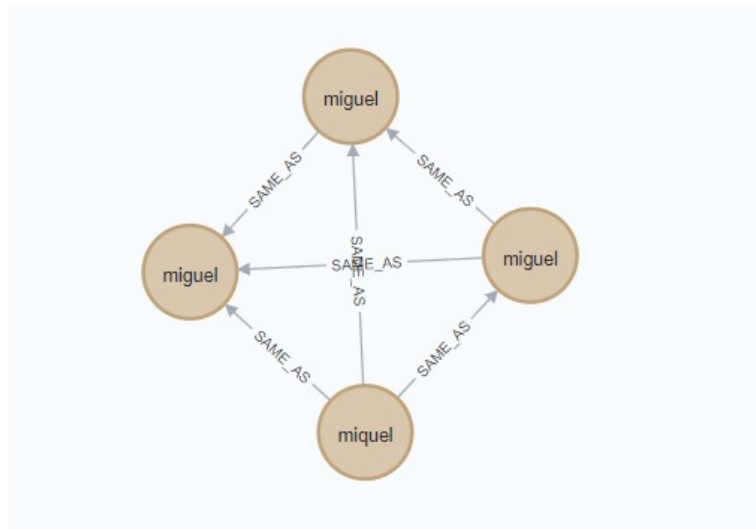


Figura 8: Sortida de la consulta 4

"nom"	"cognom1"	"cognom2"	"tipus_relacio"
"esperanza"	"colet"	"gavarro"	"FAMILIA"
"catalina"	"farran"	"colet"	"FAMILIA"
"esperanza"	"farran"	"colet"	"FAMILIA"
"antonio"	"farran"	"sole"	"FAMILIA"
"francisco"	"farran"	"colet"	"FAMILIA"
"isidro"	"farran"	"colet"	"FAMILIA"
"antonio"	"ferran"	"sele"	"SAME_AS"
"antonio"	"ferran"	"sole"	"SAME_AS"

Figura 9: Sortida de la consulta 5

"relacions_familiars"
["esposa","fill","filla","jefe","net","gendre","jove","neta","germa","mare","germana","cunyat","cunyada","sogre","altres","familiar","servents","parents","afillat","ala","al","fl1","nebot","neboda","ne","marit","pare","mae","besnet","consogre","besneta","fill_2","jefe_2","esposa_2","afillada","religios","tiet","fill_3","jefe_3","esposa_3","fill_1","jefe_1","esposa_1","filla_2","esposa_4","fill_5","fill_8"]

Figura 10: Sortida de la consulta 6

"nom"	"primer_cognom"	"segon_cognom"	"total"
"pablo"	"astruch"	"julia"	7
"jose"	"olle"	"domenech"	6
"benito"	"julivert"	"parera"	6
"jose"	"canals"	"olle"	6
"pedro"	"bargallo"	"ilegible"	6
"jose"	"canals"	"mila"	6
"jose"	"rafuls"	"mila"	5
"jaime"	"jarrey"	"ilegible"	5
"pablo"	"bargallo"	"armangol"	5
"francisco"	"aregay"	"rigol"	5
"pablo"	"canals"	"llimona"	4
"ramon"	"canals"	"amat"	4
"jaime"	"gallofre"	"bartran"	4
"tomas"	"parera"	"roig"	4
"juan"	"julibert"	"parera"	4
"estevan"	"gallofre"	"bertran"	4
"cristobal"	"olle"	"rabantos"	4
"pedro"	"farres"	"rigol"	4
"jose"	"llopart"	"pareyada"	4
"jose"	"suñol"	"henan"	4

Figura 11: Sortida de la consulta 7

"Carrer"	"Número"	"Número_total_habitatges"	"llistat_anys"	"llistat_ids"
"creus"	9	5	[1833,1839,1878,1881,1889]	[150,610,467,367,347]
"creus"	16	5	[1833,1839,1878,1881,1889]	[158,616,473,372,388]
"creus"	23	5	[1833,1839,1878,1881,1889]	[166,624,480,380,353]
"creus"	18	5	[1833,1839,1878,1881,1889]	[160,618,475,374,389]
"iglesia"	11	5	[1833,1839,1878,1881,1889]	[106,732,523,520,488]
"iglesia"	9	5	[1833,1839,1878,1881,1889]	[104,729,730,521,517,487]
"iglesia"	2	5	[1833,1839,1878,1881,1889]	[96,722,515,510,498]
"iglesia"	1	5	[1833,1839,1878,1881,1889]	[94,721,514,509,483]
"iglesia"	4	5	[1833,1839,1878,1881,1889]	[98,725,517,512,499]
"iglesia"	3	5	[1833,1839,1878,1881,1889]	[97,723,724,516,511,484]

Figura 12: Sortida de la consulta 8

total_llars	total_fills	mitja
596	1289	5.033557046979865

Figura 13: Sortida de la consulta 9

"any_padro"	"carrer"	"num_habitants"
1881	"Carretera"	5
1878	"carrretera"	2
1889	"s n antonio"	1
1838	"carretera de barna"	30
1833	"carrtera de la part de molins de rey"	5
1839	"casas del 3onmany"	3

Figura 14: Sortida de la consulta 10