**PROJECTE NEO4J**

Bases de Dades No Relacionals

**Introducció**

Aquest informe documenta la resolució d'un projecte utilitzant Neo4j per gestionar dades de padrons municipals. Es descriuen les tasques realitzades, la distribució de tasques entre els membres de l'equip, així com els scripts i consultes en Cypher per importació de dades, resolució d'exercicis i anàlisi de grafs.

**Adreça del Repositori**

L'adreça del repositori que conté tot el material generat per aquest informe és: https://github.com/exemple/proyecto-neo4j.

**Treball en Equip**

| David Morillo | * **Importació de les dades** |
| --- | --- |
| Albert Guillaumet | * **Consultes de Cypher** |
| Adrià Muro | * **Redacció de l’informe** |
| Lucía Garrido | * **Analítica de Grafs** |

Cada membre ha treballat en la seva tasca assignada, reflectint-se en els commits del repositori del projecte.

# Exercici 1: Importació de Dades en la BD de Neo4j

Hem importat dades en la base de dades de Neo4j mitjançant un script en Cypher que genera nodes, relacions i afegeix les característiques corresponents. Ens hem assegurat que l'execució de l'script dues vegades no dupliqui les dades, mitjançant l'ús de la clàusula “MERGE” i indexos. A continuació s’explica cada fase de la creació de la db.

***match (a) -[r] -> () delete a, r;***

***match (a) delete a;***

Aquesta fase esborra tots els nodes i relacions existents a la base de dades. S'utilitzen dues instruccions MATCH seguides de DELETE per assegurar-se que primer s'eliminen totes les relacions i després tots els nodes.

***CREATE INDEX house\_id\_index IF NOT EXISTS FOR (n:House) ON (n.id);***

***CREATE INDEX individual\_id\_index IF NOT EXISTS FOR (n:Individual) ON (n.id);***

Es creen indexos per als nodes House i Individual basats en el camp id per millorar el rendiment de les consultes. Es fa ús de IF NOT EXISTS per evitar errors si els indexos ja existeixen.

***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, row.Municipi + "\_" + row.Any\_Padro + "\_" + row.Id\_Llar AS house\_id***

***MERGE (n:House {id: house\_id})***

***SET n.year\_padron = row.Any\_Padro,***

***n.number = row.Numero,***

***n.municipality = row.Municipi,***

***n.house\_id = row.Id\_Llar,***

***n.street = row.Carrer;***

Aquesta fase carrega les dades de les cases des d'un fitxer CSV. Es genera un identificador únic per a cada casa combinant el municipi, l'any del padró i l'ID de la llar (house\_id). Es fa servir MERGE per evitar duplicats i SET per establir les propietats dels nodes House.

***LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-1vTfU6oJBZhmhzzkV\_0-avABPzHTdXy8851ySDbn2gq32WwaNmYxfiBtCGJGOZsMgCWjzlEGX4Zh1wqe/pub?output=csv' AS row***

***MERGE (n:Individual {id: row.Id})***

***SET n.year = row.Year,***

***n.name = row.name,***

***n.surname = row.surname,***

***n.second\_surname = row.second\_surname;***

Aquesta fase carrega les dades dels individus des d'un fitxer CSV. Utilitza MERGE per crear nodes Individual evitant duplicats, i SET per establir les seves propietats.

***LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-1vRVOoMAMoxHiGboTjCIHo2yT30CCWgVHgocGnVJxiCTgyurtmqCfAFahHajobVzwXFLwhqajz1fqA8d/pub?output=csv' AS row***

***MATCH (n1:Individual {id: row.ID\_1}), (n2:Individual {id: row.ID\_2})***

***MERGE (n1)-[f:IS\_FAMILY]->(n2)***

***SET f.relationship = row.Relacio,***

***f.harmonized\_relationship = row.Relacio\_Harmonitzada;***

Aquesta fase carrega les relacions familiars entre individus des d'un fitxer CSV. Els nodes d'individus es troben amb MATCH i es crea una relació IS\_FAMILY utilitzant MERGE per evitar duplicats. Les propietats de la relació es configuren amb SET.

***LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-1vRM4DPeqFmv7w6kLH5msNk6\_Hdh1wuExRirgysZKO\_Q70L21MKBkDISIyjvdm8shVixl5Tcw\_5zCfdg/pub?output=csv' AS row***

***WITH row, row.Location + "\_" + row.Year + "\_" + row.HOUSE\_ID AS house\_id***

***MATCH (n:Individual {id: row.IND}), (h:House {id: house\_id})***

***MERGE (n)-[l:LIVES\_IN]->(h)***

***SET l.lives = row.VIU,***

***l.location = row.Location,***

***l.year = row.Year;***

Aquesta fase carrega les relacions de residència entre individus i cases des d'un fitxer CSV. Es genera un house\_id únic, es troben els nodes Individual i House amb MATCH, i es crea la relació LIVES\_IN utilitzant MERGE. Les propietats de la relació es configuren amb SET.

***LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-1vTgC8TBmdXhjUOPKJxyiZSpetPYjaRC34gmxHj6H2AWvXTGbg7MLKVdJnwuh5bIeer7WLUi0OigI6wc/pub?output=csv' AS row***

***MATCH (n1:Individual {id: row.Id\_A}), (n2:Individual {id: row.Id\_B})***

***MERGE (n1)-[:SAME\_AS]->(n2);***

Aquesta fase carrega les relacions SAME\_AS entre individus des d'un fitxer CSV. Els nodes d'individus es troben amb MATCH i es crea la relació SAME\_AS utilitzant MERGE per evitar duplicats.

# Exercici 2: Consultes Cypher

Hem resolt diverses consultes Cypher per obtenir informació específica dels padrons municipals. A continuació es motraran les diferents consultes juntament amb una captura de pantalla del resultat al executar-les.

**1- Per a cada padró (any) de Sant Feliu de Llobregat (SFLL), retorna l’any de padró, el número d'habitants, i la llista de cognoms. Elimina duplicats i “nan”.**

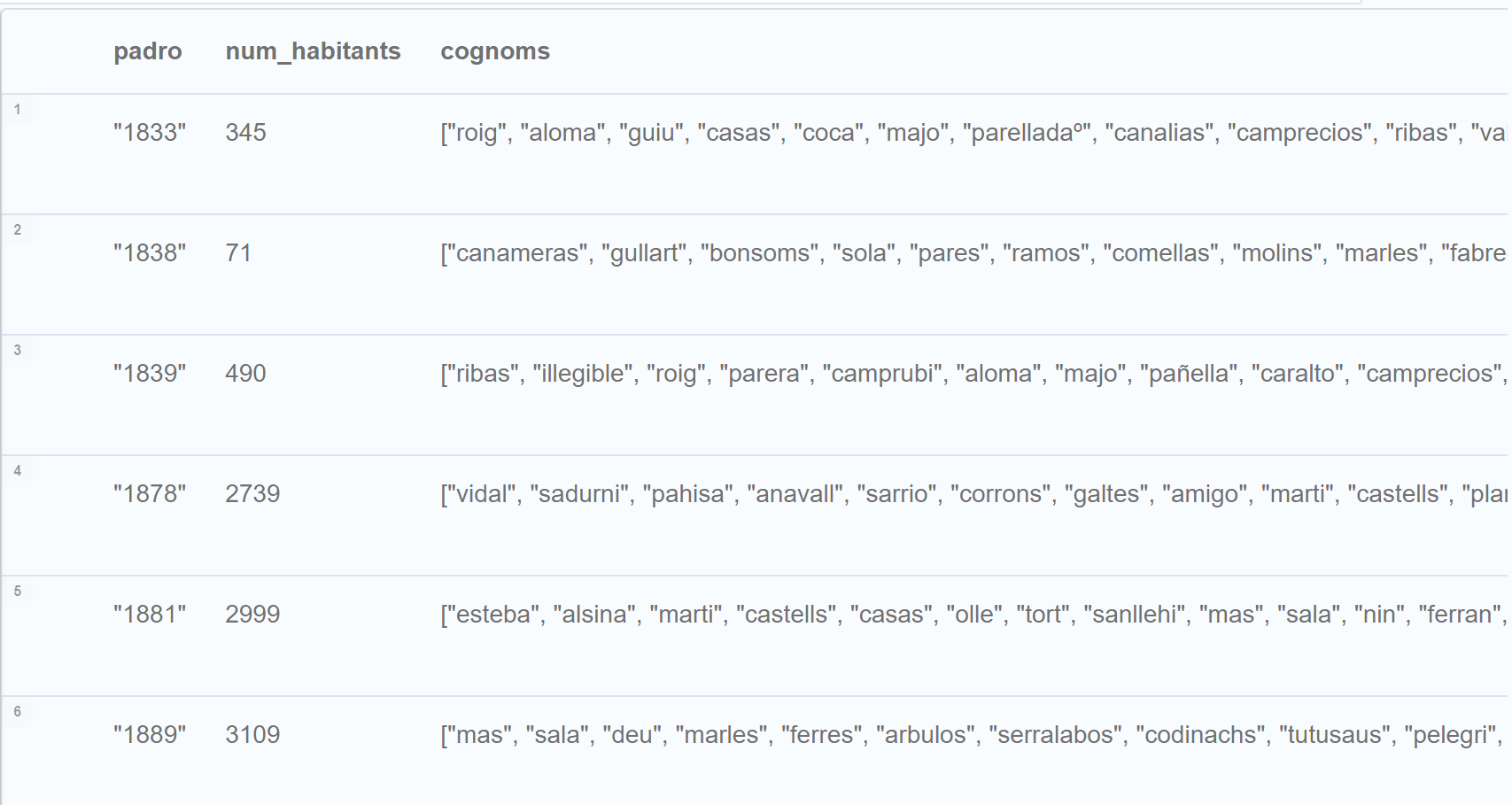
MATCH (h:House {municipality: 'SFLL'})<-[:LIVES\_IN]-(i:Individual)

WITH h.year\_padron as padro, i

WHERE i.surname IS NOT NULL AND i.surname <> "nan"

RETURN padro as padro, COUNT(DISTINCT i) AS num\_habitants, COLLECT(DISTINCT i.surname) as cognoms

ORDER BY padro



**2- 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 (i:Individual)-[:SAME\_AS]->(in:Individual)

WHERE (in.name = "miguel" AND in.surname = "estape" AND in.second\_surname = "bofill")

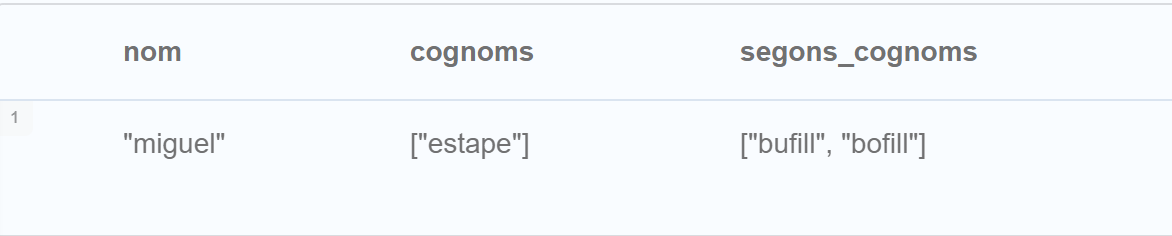
WITH i

RETURN i.name as nom,

COLLECT(DISTINCT i.surname) as cognoms,

COLLECT(DISTINCT i.second\_surname) as segons\_cognoms

ORDER BY nom



**3- Mostra els fills o filles(només) de "benito julivert". Mostra la informació en forma de taula: el nom, cognom1, cognom2, i tipus de relació. Ordena els resultats alfabèticament per nom.**

MATCH(pare:Individual{name: "benito", surname:"julivert"})-[r:IS\_FAMILY]->(fill:Individual)

WHERE r.harmonized\_relationship IN ["fill","filla"]

WITH fill,r

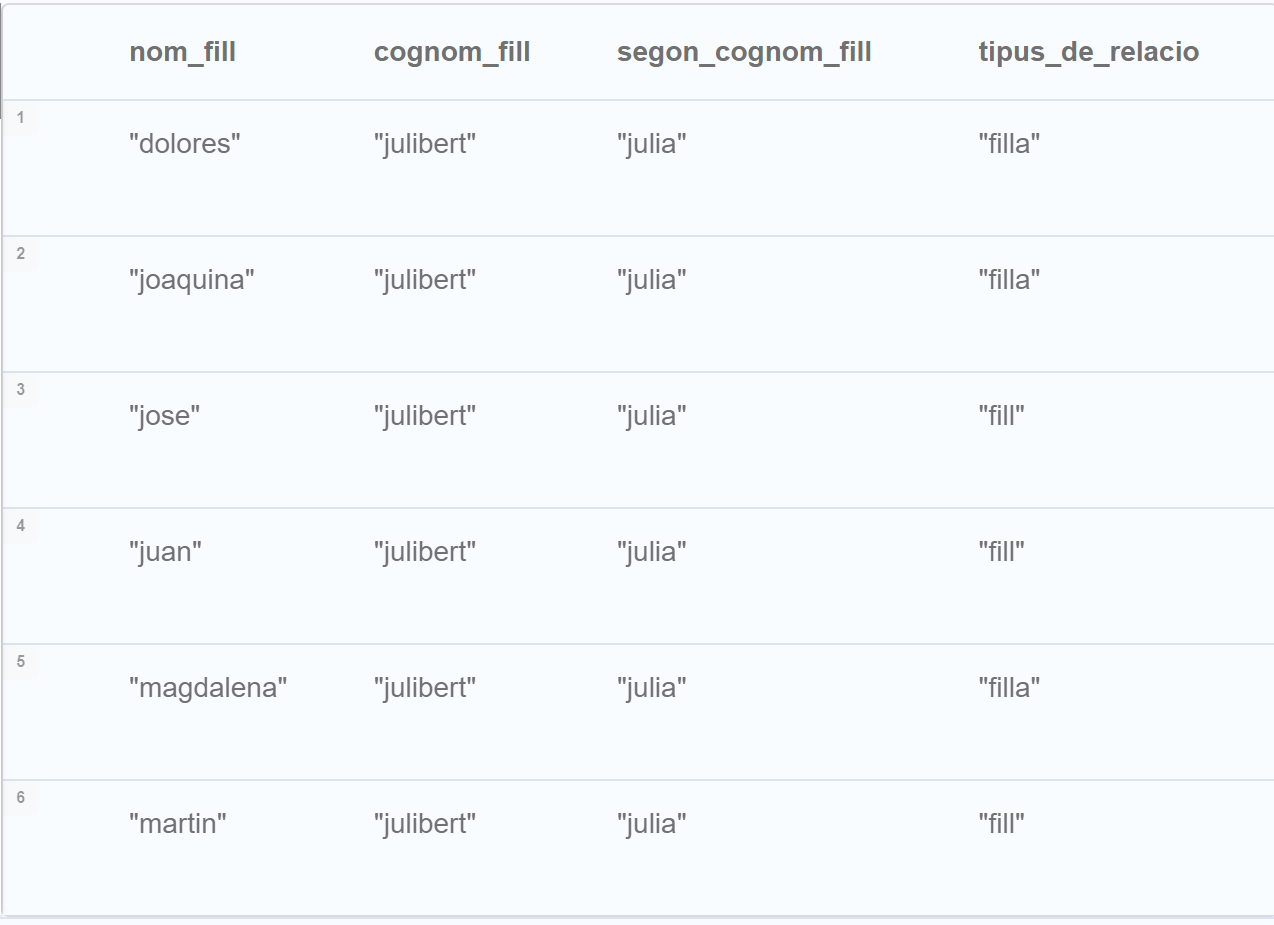
RETURN fill.name as nom\_fill,

fill.surname as cognom\_fill,

fill.second\_surname as segon\_cognom\_fill,

r.harmonized\_relationship as tipus\_de\_relacio

ORDER BY nom\_fill



**4- 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:House{municipality:"CR"})<-[:LIVES\_IN]-(p:Individual)

WITH p, h

MATCH (p)-[r:IS\_FAMILY]->(fill:Individual)

WHERE r.harmonized\_relationship IN ["fill","filla"]

WITH p,COUNT(fill) AS numero\_fills

WHERE numero\_fills > 3

RETURN p.name as nom\_cap,

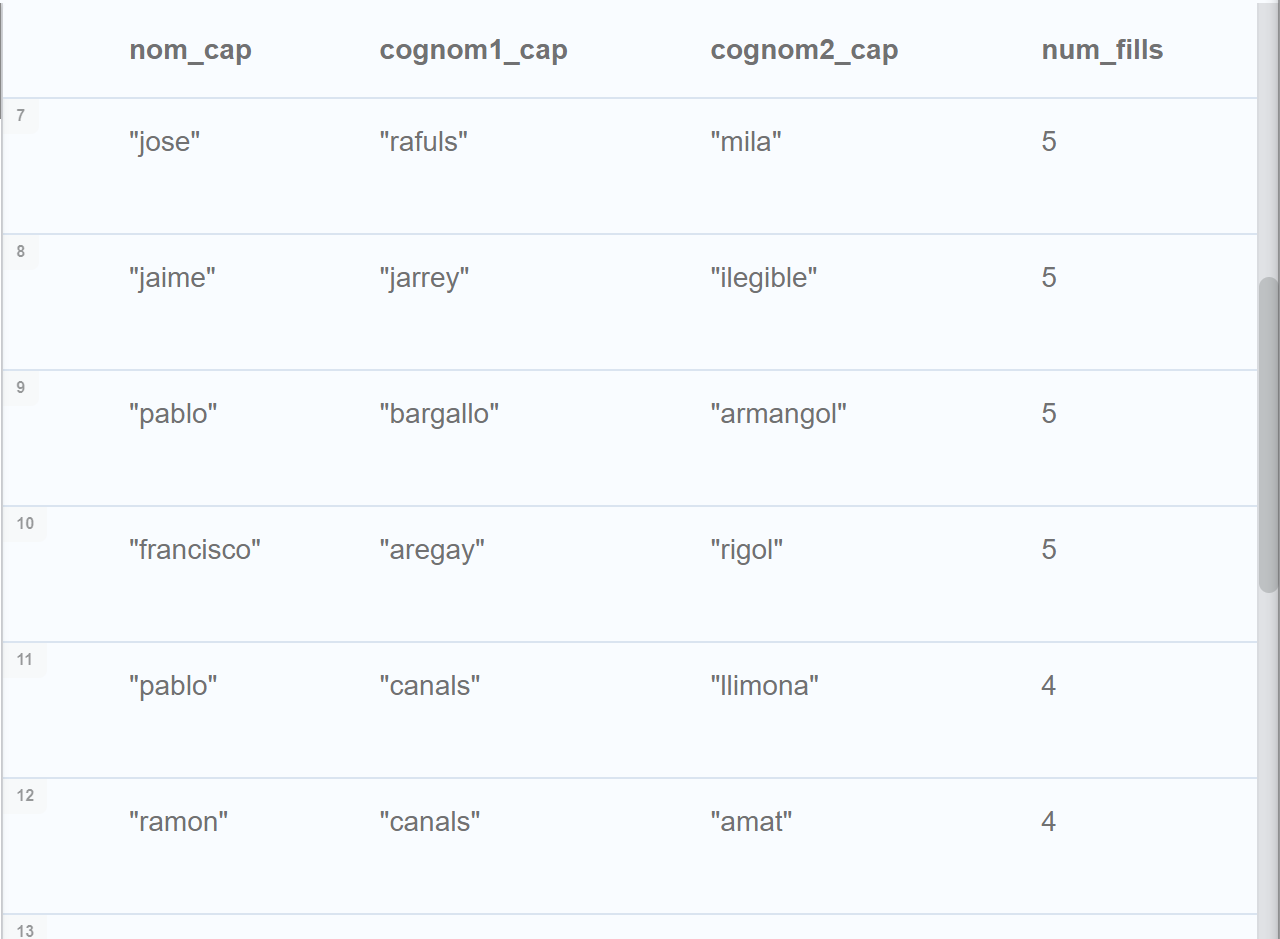
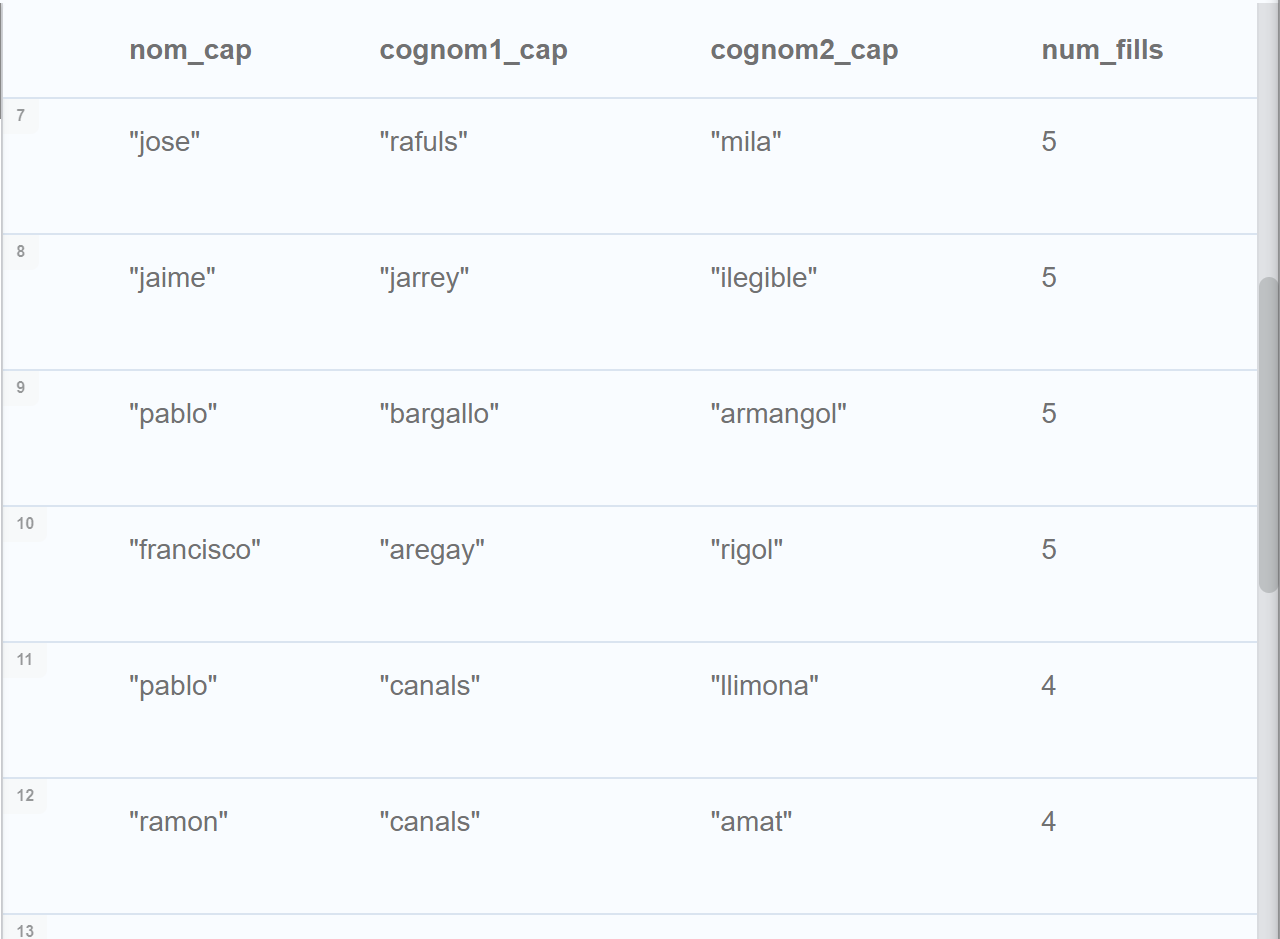
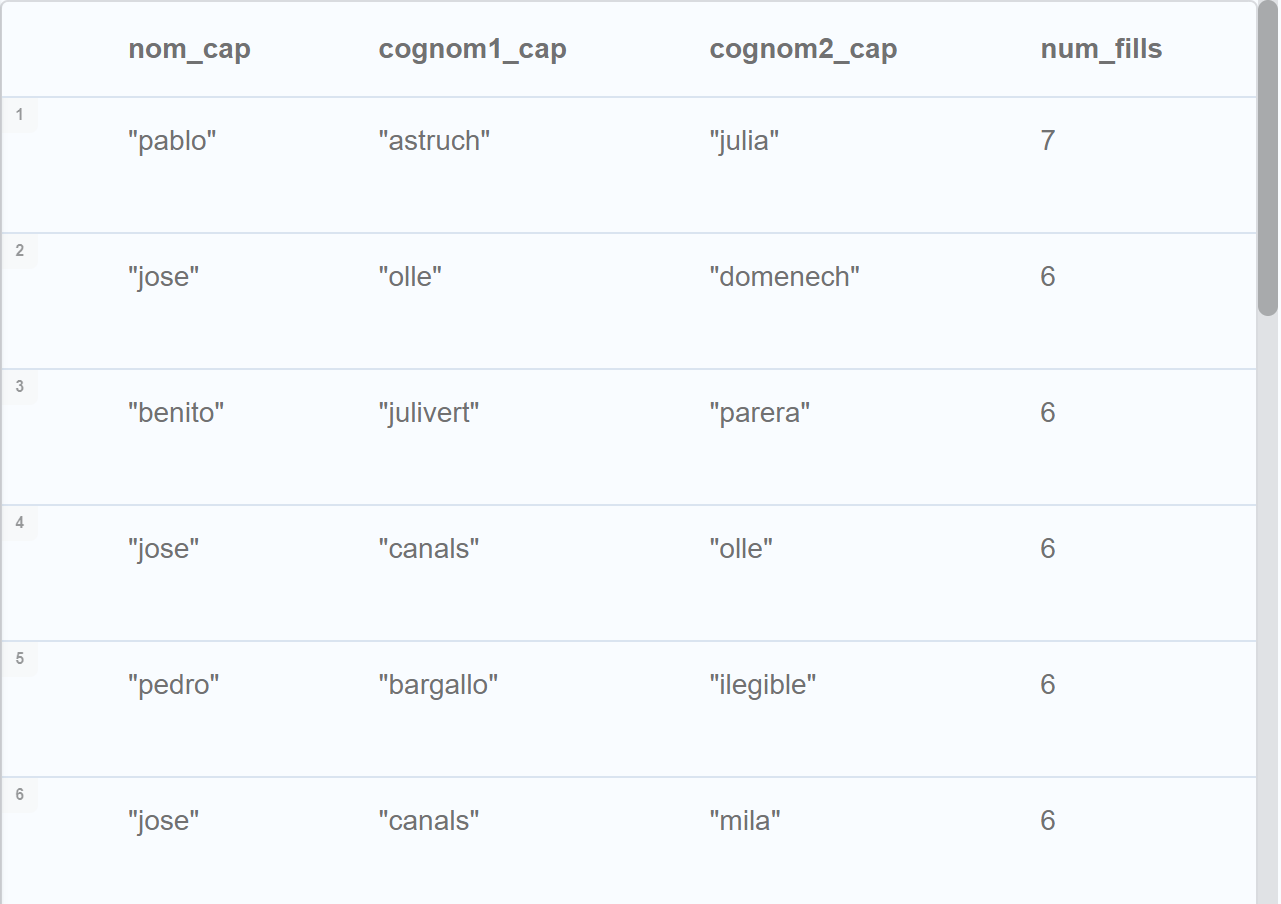
p.surname as cognom1\_cap,

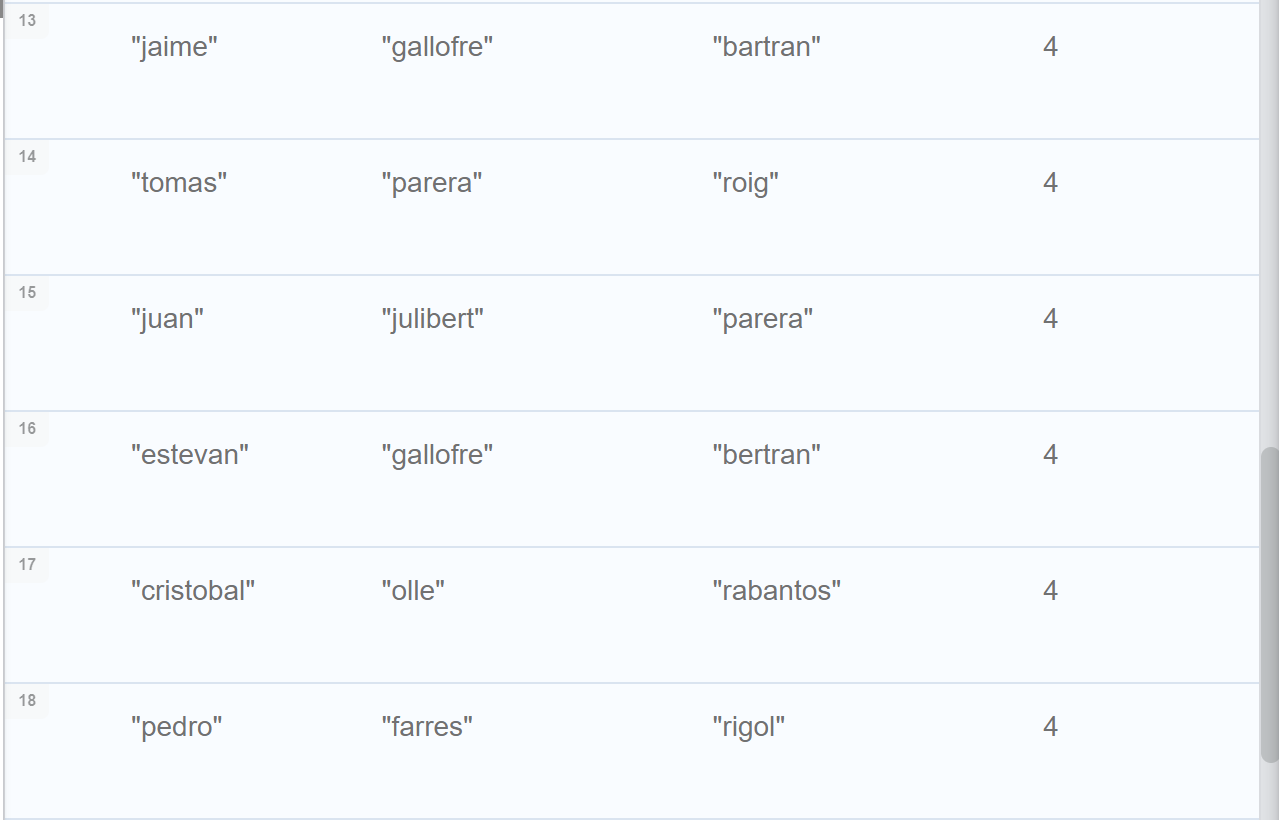
p.second\_surname as cognom2\_cap,

numero\_fills as num\_fills

ORDER BY num\_fills DESC

LIMIT 20





**5- 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:House {municipality: 'SFLL'})<-[:LIVES\_IN]-(i:Individual)

WITH h.year\_padron AS any, h.street AS carrer, COUNT(i) AS num\_habitants

WITH any, carrer, num\_habitants, min(num\_habitants) OVER (PARTITION BY any) AS min\_habitants

WHERE num\_habitants = min\_habitants

RETURN any, carrer, num\_habitants

ORDER BY any ASC;

# Exercici 3: Analítica de Grafs

Hem realitzat una anàlisi de grafs per entendre millor l'estructura de les dades, incloent l'estudi de components connexes i la semblança entre nodes.

**ESTUDI DE LES COMPONENTS CONNEXES**

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

CALL gds.wcc.stream({

nodeProjection: ['Individual', 'House'],

relationshipProjection: {

LIVES\_IN: {

type: 'LIVES\_IN',

orientation: 'UNDIRECTED'

}

}

})

YIELD nodeId, componentId

WITH componentId, COUNT(nodeId) AS size

RETURN componentId, size

ORDER BY size DESC

LIMIT 10;

**2 - Per cada municipi i any 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;

**SEMBLANÇA ENTRE NODES**

**1 - Determinar els habitatges que són els mateixos al llarg dels anys i afegir una aresta "MATEIX\_HAB"**

MATCH (h1:House), (h2:House)

WHERE h1.house\_id = h2.house\_id AND h1.year\_padron < h2.year\_padron

MERGE (h1)-[:MATEIX\_HAB]->(h2);

**2 - Crear un graf en memòria amb els nodes Individu i Habitatge i les relacions VIU, FAMILIA, MATEIX\_HAB**

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'

}

}

);

**3 - Calcular la similaritat entre els nodes del graf creat**

CALL gds.nodeSimilarity.write({

nodeProjection: ['Individual', 'House'],

relationshipProjection: {

LIVES\_IN: {

type: 'LIVES\_IN',

orientation: 'UNDIRECTED'

},

IS\_FAMILY: {

type: 'IS\_FAMILY',

orientation: 'UNDIRECTED'

},

MATEIX\_HAB: {

type: 'MATEIX\_HAB',

orientation: 'UNDIRECTED'

}

},

writeRelationshipType: 'SIMILAR\_TO',

writeProperty: 'similarityScore'

});