

# Grau en Matemàtica Computacional i Analítica de Dades

## Pràctica Neo4j

### Bases de Dades no Relacionals



Judit Yebra Valencia (1603614)  
Pau Fuentes Hernández (1600158)  
Andrea González Aguilera (1603921)  
Xavier Seminario Monllaó (1603853)

---

## Índex

1	Distribució de les tasques	2
2	Introducció	2
3	Link al repositori de github	3
4	Exercici 1	4
5	Exercici 2	5
6	Exercici 3	13

---

## 1 Distribució de les tasques

Per a la realització del treball, no s'ha establert una distribució de tasques des de l'inici. En canvi, ens hem coordinat de manera conjunta, aportant cadascú el que era necessari en cada moment. Donat que cadascú de nosaltres té horaris i compromisos laborals diferents, no sempre hem pogut treballar tots junts al mateix temps. A continuació, resumim com s'han repartit les tasques:

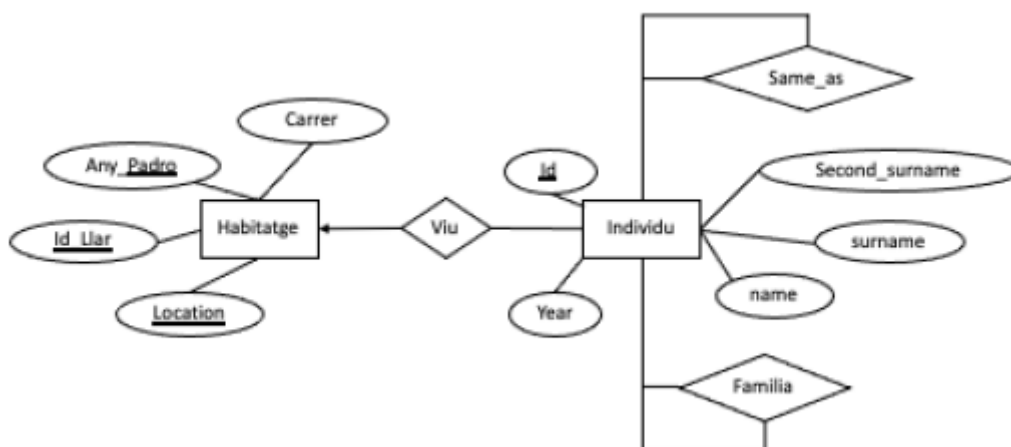
- S'ha realitzat l'script d'importació de les dades de forma conjunta inicialment en un document compartit. Després, l'Andrea s'ha encarregat de traslladar-lo a un document Cypher.
- Les consultes han estat realitzades entre el Xavi, la Judit i el Pau. No obstant això, l'Andrea i la Judit han estat les principals responsables d'incloure-les en el document, a més de corregir algunes consultes que no funcionaven correctament.
- El tercer exercici s'ha abordat mitjançant una conjunta. L'Andrea va fer una primera aproximació que no acabava de funcionar correctament, i posteriorment el Pau, el Xavi i la Judit van fer canvis i modificacions perquè funcionés de manera correcta.
- La Judit i l'Andrea han estat les principals encarregades de crear el document en LaTeX i el README.

En tot moment s'ha treballat de forma conjunta, tots els membres del grup han estat d'acord amb la tasca duta a terme per cadascun dels integrants i sempre ha existit un ambient de treball i coordinació excel·lents.

Podem concloure que hem treballat molt correctament com a equip col·laboratiu.

## 2 Introducció

L'objectiu principal d'aquest projecte és explorar com utilitzar Neo4J. Per dur-ho a terme es farà servir una base de dades sobre uns padrons els quals són els llistats d'habitants que elabora un municipi on figura la informació la seva informació com noms, cognoms, edat i altres dades personals. Aquest padró està organitzat segons el següent disseny Entitat-Relació:



Aquesta base de dades està conformada per cinc arxius els quals corresponen a les entitats i les relacions del diagrama anterior. Aquests arxius contenen la diferent informació que hi havia en el padró.

---

### **3 Link al repositori de github**

Tot seguit es disposa el link, per poder accedir al repositori de github on es troba tota la informació la qual s'ha usat per la realització d'aquest treball.

link repositori grup08.

---

## 4 Exercici 1

Importa 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. Consideracions:

- Feu servir *constraints* i *indexos* quan sigui necessari.
- Assegureu-vos que en executar el script dues vegades no es dupliquin les dades.
- No carregueu files *null* del fitxer CSV (Id de municipi, llar o individu = *null*).
- Feu les conversions de tipus que siguin necessàries.

Per aquest exercici es proporcionarà un document amb els passos documentats per importar dades amb èxit a un projecte per utilitzar-lo amb Neo4j.

## 5 Exercici 2

Resoleu les següents consultes:

1. Del padró de 1866 de Castellví de Rosanes (CR), retorna el número d'habitants i la llista de cognoms, sense eliminar duplicats.

*Codi:*

```
MATCH (p:Individu)-[:VIU]->(h:Habitatge)
WHERE h.Any_Padro=1866 AND h.Municipi="CR"

RETURN count(p) AS Habitants, collect(p.Cognom) AS Cognoms
```

<div><div></div><div>Tabelle</div></div>	HabitantsCognoms
<div><div>A</div><div>Text</div></div>	337   "olle", "olle", "olle", "olle", "olle", "galcecran", "olle", "olle", "rusell", "suñol", "suñol", "olle", "olle", "suñol", "julibert", "julibers", "julibert", "julibert", "barsallo", "anglada", "anglada", "vila", "ros", "suñol", "ros", "julia", "canals", "canals", "canals", "canals", "canals", "astrade", "esparchry", "gaset", "gaset", "gaset", "gaset", "julibert", "rumeu", "julibert", "perera", "julibert", "ajulibert", "julibert", "julibert", "calaf", "gaset", "gaset", "gaset", "julibert", "julivert", "julibert", "julibert", "julibert", "julibert", "julia", "julibert", "canals", "adalmases", "canald", "canals", "rigol", "canals", "canals", "canals", "canals", "canals", "canals", "pujadas", "pasqual", "aregay", "pasqual", "pasqual", "pasqual", "pujadas", "pasqual", "pasqual", "rafuls", "rafuls", "rafuls", "rafuls", "rafuls", "rafuls", "liopart", "canals", "volta", "farre", "volta", "farre", "canals", "calaf", "canals", "canals", "limona", "canals", "canals", "astruch", "astruch", "astruch", "astruch", "rafuls", "astruch", "astruch", "astruch", "astruch", "gaset", "gaset", "rafuls", "gaset", "gaset", "olle", "farre", "canals", "canals", "canals", "olle", "olle", "olle", "ollie", "liopart", "gallofre", "gallofre", "gallofre", "gallofre", "gallofre", "olle", "masana", "olle", "olle", "figue", "farre", "far

2. Per a cada padró de Sant Feliu de Llobregat (SFLL), retorna l'any de padró, el número d'habitants, i la llista de cognoms. Elimina duplicats i "nan".

*Codi:*

```
MATCH (i:Individu)-[:VIU]->(h:Habitatge)
WHERE h.Municipi = "SFLL" AND i.Cognom <> 'nan'
WITH h.Id AS ID, i.Any_Padro AS Any_Padro,
      count(h.Id) AS Recompte,
      collect(DISTINCT(i.Cognom)) AS Cognoms

RETURN Any_Padro, Recompte, Cognoms
ORDER BY Any_Padro ASC
```

Any_Padro	Recompte	Cognoms
1833	345	["dordal", "canals", "condeminas", "jove", "carcere
1838	71	["rovira", "julia", "vidal", "valles", "camperderros"
1839	492	["rovira", "julia", "farrer", "bros", "camperderros", ares", "andres", "casanovas", "llordella", "amat",
1878	2739	["galtes", "amigo", "corrons", "castells", "marti", arch", "campana", "forns", "borras", "padrosa", "ma t", "camats", "oliva", "carreras", "aragay", "pierr
1881	2999	["tort", "sanllehi", "nin", "mas", "sala", "castell sanchez", "marsell", "subrirachs", "viñas", "pale ntich", "aguilar", "barres", "marquet", "carnice",
1889	3109	["tort", "sanllehi", "olive", "ribas", "carcereny", ", "verdagner", "albareda", "taulet", "ruiz", "buro ", "llopis", "rafols", "gistaro", "masons", "domingo

*Codi:*

Municipio	Any_Padro	Llista_Id
"SFLL"	1833	[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 11, 72, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 4, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254,
"SFLL"	1838	[314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 3
"SFLL"	1839	[369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 3, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 45, 13, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 5, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 1, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 7

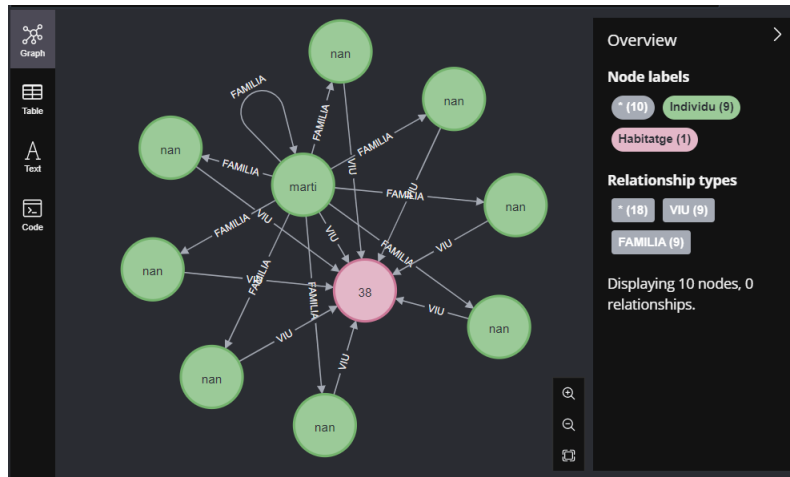
*Codi Mode Llista:*

Table	Nom
Text	"jpha"
Code	"felipe"
	"maria"
	"jpha"
	"franco"
	"jph"
	"miquel"
	"salvadora"

*Codi Mode Graf:*

```
MATCH (p:Individu{Nom:'rafel',Cognom:'marti'})-[:VIU]->
      (h:Habitatge{Municipi:"SFL",Any_Padro:1838})<-[:VIU]-(p2:Individu)

RETURN p, h, p2
```



5. Retorna totes les aparicions de "miguel estape bofill". Fes servir la relació SAME\_AS per poder retornar totes les instàncies, independentment de si hi ha variacions lèxiques (ex. diferents formes d'escriure el seu nom/cognoms). Mostra la informació en forma de subgraf.

*Codi:*

```
MATCH (p:Individu{Nom:'miguel',Cognom:'estape',Segon_Cognom:'bofill'})
      <-[:SAME_AS]->(p2:Individu)

RETURN p,p2
```

p	p2
(:Individu {Cognom: "estape",Segon_Cognom: "bofill",Any_Padr o: 1878,Id: 18780897,Nom: "miguel"})	(:Individu {Cognom: "estape",Segon_Cognom: "bofill",Any_Padr o: 1881,Id: 18810920,Nom: "miguel"})
(:Individu {Cognom: "estape",Segon_Cognom: "bofill",Any_Padr o: 1878,Id: 18780897,Nom: "miguel"})	(:Individu {Cognom: "estape",Segon_Cognom: "bofill",Any_Padr o: 1889,Id: 18891089,Nom: "miguel"})
(:Individu {Cognom: "estape",Segon_Cognom: "bofill",Any_Padr o: 1878,Id: 18780897,Nom: "miguel"})	(:Individu {Cognom: "estape",Segon_Cognom: "bofill",Any_Padr o: 1857,Id: 18570295,Nom: "miguel"})
(:Individu {Cognom: "estape",Segon_Cognom: "bofill",Any_Padr o: 1881,Id: 18810920,Nom: "miguel"})	(:Individu {Cognom: "estape",Segon_Cognom: "bofill",Any_Padr o: 1878,Id: 18780897,Nom: "miguel"})



6. De la consulta anterior, retorna la informació en forma de taula: el nom, la llista de cognoms i la llista de segon cognom (elimina duplicats).

*Codi:*

```
MATCH (p:Individu{Nom:'miguel',Cognom:'estape',Segon_Cognom:'bofill'})
<-[:SAME_AS]->(p2:Individu)

RETURN p.Nom AS Nom, collect(distinct p2.Cognom)AS Cognom1,
collect(distinct p2.Segon_Cognom) AS Cognom2
```

Nom	Cognom1	Cognom2
"miguel"	["estape"]	["bofill", "bufill"]

7. Mostra totes les persones relacionades amb "benito julivert". Mostra la informació en forma de taula: el nom, cognom1, cognom2, i tipus de relació.

*Codi:*

```
MATCH (p:Individu
{Nom:'benito',Cognom:'julivert'})-[:FAMILIA]-(otraPersona:Individu)

RETURN otraPersona.Nom AS Nom,
otraPersona.Cognom AS Cognom1,
otraPersona.Segon_Cognom AS Cognom2,
f.Relacio_Harmonitzada AS Tipus_Relacio
```

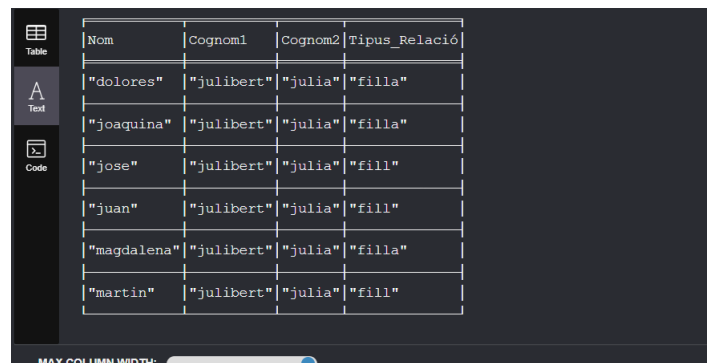
Nom	Cognom1	Cognom2	Tipus_Relació
"benito"	"julivert"	"parera"	"jefe"
"josefa"	"julia"	"jisa"	"esposa"
"jose"	"julibert"	"julia"	"fill"
"magdalena"	"julibert"	"julia"	"filla"
"joaquina"	"julibert"	"julia"	"filla"
"juan"	"julibert"	"julia"	"fill"
"dolores"	"julibert"	"julia"	"filla"
"martin"	"julibert"	"julia"	"fill"

8. De la consulta anterior, mostra ara només els fills o filles de "benito julivert".  
 Ordena els resultats alfabèticament per nom.

*Codi:*

```
MATCH (p:Individu {Nom:'benito',
Cognom:'julivert'})-[f:FAMILIA]-(fill:Individu)
WHERE f.Relacio_Harmonitzada IN ['fill', 'filla']

RETURN fill.Nom AS Nom,
       fill.Cognom AS Cognom1,
       fill.Segon_Cognom AS Cognom2,
       f.Relacio_Harmonitzada AS Tipus_Relacio
ORDER BY Nom ASC
```

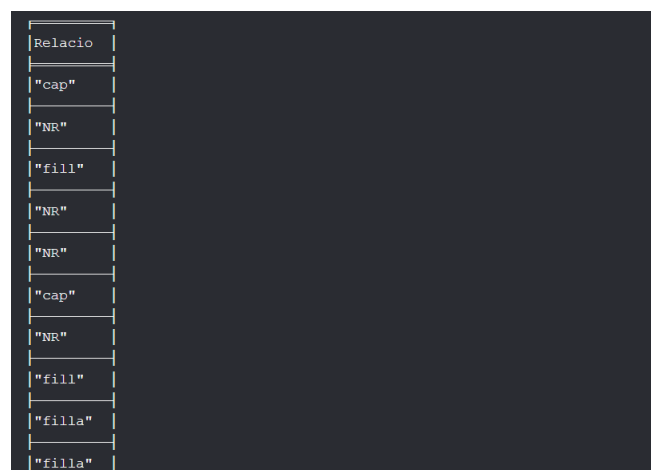


Nom	Cognom1	Cognom2	Tipus_Relació
"dolores"	"julibert"	"julia"	"filla"
"joaquina"	"julibert"	"julia"	"filla"
"jose"	"julibert"	"julia"	"fill"
"juan"	"julibert"	"julia"	"fill"
"magdalena"	"julibert"	"julia"	"filla"
"martin"	"julibert"	"julia"	"fill"

9. Llisteu totes les relacions familiars que hi ha.

*Codi:*

```
MATCH (:Individu)-[f:FAMILIA]->(:Individu)
RETURN f.Relacio AS Relacio
```



Relacio
"cap"
"NR"
"fill"
"NR"
"NR"
"cap"
"NR"
"fill"
"filla"
"filla"

10. Identifiqueu els nodes que representen el mateix habitatge (carrer i numero) al llarg dels padrons de Sant Feliu del Llobregat (SFLL). Seleccioneu només els habitatges que tinguin totes dues informacions (carrer i numero). Per a cada habitatge, retorneu el carrer i número, el nombre total de padrons on apareix, el llistat d'anys dels padrons i el llistat de les Ids de les llars (eviteu duplicats). Ordeneu de més a menys segons el total de padrons i mostreu-ne els 15 primers.

*Codi:*

```
MATCH (h:Habitatge)
WHERE h.Municipi = "SFLL" AND h.Carrer <> 'null' AND h.Numero <> -1

RETURN h.Carrer AS Carrer, h.Numero AS Numero, count(h.Id) AS Aparicions,
       collect(DISTINCT(h.Any_Padro)) AS Anys,
       collect(DISTINCT(h.Id)) AS Llista_Ids
ORDER BY Aparicions DESC
LIMIT 15
```

Carrer	Numero	Aparicions	Anys	Llista_Ids
"falguera"	5	9	[1833, 1878, 1889, 1881]	[247, 359, 360, 361, 398, 414, 415, 416, 417]
"falguera"	3	8	[1833, 1878, 1889, 1881]	[245, 357, 358, 397, 411, 412, 413, 668]
"falguera"	22	7	[1833, 1878, 1881, 1889]	[269, 376, 377, 378, 429, 430]
"san antonio"	1	7	[1889]	[562, 563, 564, 565, 566, 567, 568]
"carretera"	85	6	[1889, 1881]	[41, 83, 84, 85, 86, 87]
"carretera"	141	6	[1889, 1881, 1889]	[73, 152, 172, 173, 174, 673]

11. 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.

*Codi:*

```
CALL{
  MATCH (i:Individu)-[f:FAMILIA]->(:Individu)-[:VIU]
    ->(h:Habitatge)
  WHERE h.Municipi = "CR" AND (f.Relacio = "hijo" OR
    f.Relacio = "hija")

  RETURN i
}

MATCH (i)<-[f:FAMILIA]-(p:Individu)
WITH count(i) AS Nombre_Fills, f AS familia, p AS CAP
WHERE familia.Relacio = "cabeza" AND Nombre_Fills > 3

RETURN CAP.Nom AS Nom, CAP.Cognom AS Cognom,
  CAP.Segon_Cognom AS Segon_Cognom,
  Nombre_Fills AS Nombre_Fills
ORDER BY Nombre_Fills DESC
LIMIT 20
```

Nom	Cognom	Segon_Cognom	Nombre_Fills
"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

12. 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 de fills per habitatge. Fes servir CALL per obtenir el nombre de llars.

Codi:

```
CALL{
  MATCH (p2:Individu)-[:VIU{Any_Habitatge:1881}]->
    (h:Habitatge{Municipi:'SFL' })<-[:VIU]-(p:Individu)

  RETURN p,p2,h
}

MATCH (p)<-[r:FAMILIA]-(p2)
WHERE r.Relacio_Harmonitzada="fill" or
      r.Relacio_Harmonitzada="filla"

RETURN count(distinct p.Id) AS TotalFills,
       count(distinct h.Id) AS NombreHabitatges,
       toFloat(count(distinct p.Id))/toFloat(count(distinct h.Id))
       AS MitjaFills
```

TotalFills	NombreHabitatges	MitjaFills
1289	465	2.772043010752688

13. 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.

Codi:

```
CALL {
  MATCH (i:Individu)-[:VIU]->(h:Habitatge)
  WHERE h.Municipi = 'SFL'
  WITH h.Any_Padro AS Any, h.Carrer AS Carrer, count(i) AS Habitants

  RETURN Any, Carrer, min(Habitants) AS Min_Habitants
}

RETURN Any, Carrer, Min_Habitants
ORDER BY Any ASC
```

Any	Carrer	Min_Habitants
1833	"carretera de la part de molins"	9
1833	"carretera de la part de molins de rey"	237
1833	"carretera de la part de molins de rey"	5
1833	"masovenou"	141
1833	"iglesia"	92
1833	"rectoria"	168
1833	"creus"	119
1833	"armenteras"	63
1833	"baix"	157

## 6 Exercici 3

En aquest exercici analitzarem les dades del graf per entendre millor l'estructura de les dades.

a) Estudi de les components connexes (cc) i de l'estructura d'aquestes en funció de la seva mida. Consultes per explorar les dades:

### 1. Taula agrupant els resultats segons la mida de la cc.

Motivació:

La motivació d'aquesta primera consulta és poder entendre la distribució de les components connexes segons la seva mida per identificar si hi ha una gran component connexa dominant o si hi ha diverses petites components aïllades. Podent saber quants grups independents hi ha i la mida de cada grup.

Codi:

```
CALL gds.graph.project('myGraph1', ['Habitatge', 'Individu'],
    ['VIU', 'FAMILIA', 'SAME_AS'])

CALL gds.wcc.stream('myGraph1')
YIELD nodeId, componentId
WITH componentId, count(*) AS componentSize
RETURN componentSize, count(*) AS count
ORDER BY componentSize ASC
```



The screenshot shows a table with two columns: 'componentSize' and 'count'. The table is sorted by 'componentSize' in ascending order. The data is as follows:

	componentSize	count
1	1	3222
2	2	117
3	3	143
4	4	185
5	5	170
6	6	154
7		

Started streaming 56 records after 3 ms and completed after 54 ms

Explicació dels resultats obtinguts:

Els resultats obtinguts mostren els nuclis de veïns (o gent que comparteix alguna relació) que hi ha al padró ordenat del nucli conformat per més persones al més petit. A diferència de les consultes anteriors en aquest cas es tenen en compte totes les relacions i no només els nuclis familiars.

## 2. Distribució de tipus de nodes (Individu o Habitatge) segons la mida de la cc.

Primerament, crearem una projecció del graf original i seleccionarem només els nodes i relacions que ens interessin.

```
CALL gds.graph.project('myGraph2', ['Habitatge', 'Individu'], ['VIU'])
```

### 1. Proporció Individus/Habitatge

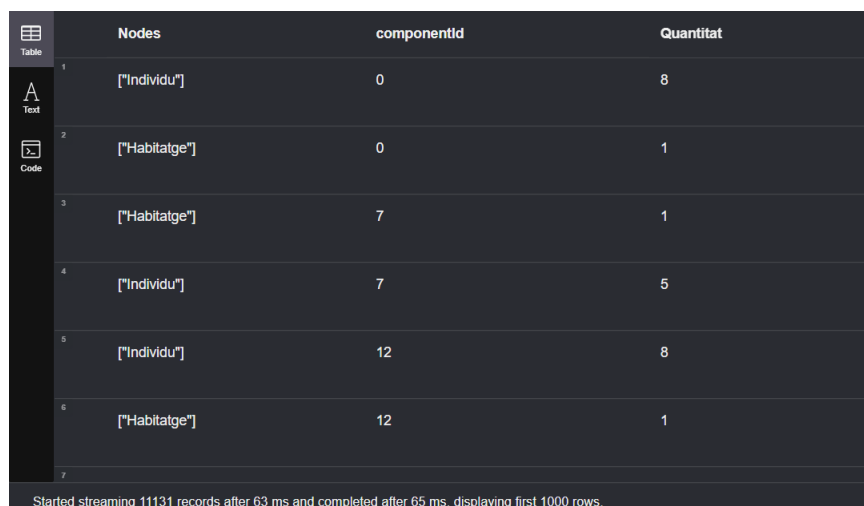
#### Motivació:

L'objectiu d'aquesta consulta és observar quantes persones vivien en un mateix habitatge de mitja per conèixer com eren els cases antigament.

#### Codi:

```
CALL gds.wcc.stream('myGraph2')
YIELD componentId, nodeId
WITH labels(gds.util.asNode(nodeId)) AS Nodes,
     componentId, count(*) AS Quantitat

RETURN Nodes, componentId, Quantitat
ORDER BY componentId
```



	Nodes	componentId	Quantitat
1	["Individu"]	0	8
2	["Habitatge"]	0	1
3	["Habitatge"]	7	1
4	["Individu"]	7	5
5	["Individu"]	12	8
6	["Habitatge"]	12	1
7			

Started streaming 11131 records after 63 ms and completed after 65 ms, displaying first 1000 rows.

#### Explicació dels resultats obtinguts:

En aquesta primera aproximació es pot observar, per cada nucli familiar, quantes persones vivien en un mateix habitatge, es pot observar com es veuen valors prou elevats, per poder extreure millors conclusions s'ha calculat la mitjana.

```
CALL gds.wcc.stream('myGraph2')
YIELD componentId, nodeId
WITH labels(gds.util.asNode(nodeId)) AS Nodes, componentId, count(*)
AS Quantitat
WHERE 'Individu' IN Nodes

RETURN Nodes, AVG(Quantitat) AS Mitjana
```

Nodes	Mitjana
["Individu"]	2.3635387300308777

Explicació dels resultats obtinguts:

Es pot observar com la mitjana és sorprenentment baixa, com el que es vol observar és el comportament de les famílies, s'ha provat de fer la mitjana d'aquells nuclis familiars que tinguessin com a mínim dues persones, ja que una persona que viu sola no es considera una família.

```
CALL gds.wcc.stream('myGraph2')
YIELD componentId, nodeId
WITH labels(gds.util.asNode(nodeId)) AS Nodes, componentId, count(*)
AS Quantitat
WHERE 'Individu' IN Nodes AND Quantitat > 1

RETURN Nodes, AVG(Quantitat) AS Mitjana
```

Nodes	Mitjana
["Individu"]	4.942934782608709

Explicació dels resultats obtinguts:

Es pot observar clarament com ha pujat la mitjana respecte del resultat anterior, i és que antigament les famílies solien ser bastant nombroses. Aprofitant el resultat anterior cal destacar també que hi havia bastantes persones que vivien soles en aquella època.



## 2. Màxim nombre d'individus en un mateix habitatge

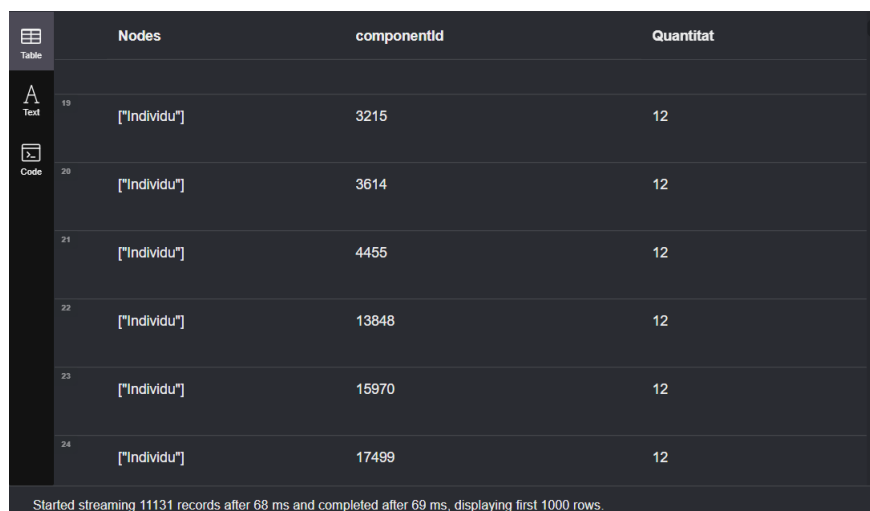
### Motivació:

En aquesta consulta, el que es vol és poder conèixer com vivien i s'organitzaven les famílies al segle XIX. Així com tenir informació sobre l'estructura familiar i la composició de les llars en aquella època.

### Codi:

```
CALL gds.wcc.stream('myGraph2')
YIELD componentId, nodeId
WITH labels(gds.util.asNode(nodeId)) AS Nodes, componentId,
      count(*) AS Quantitat

RETURN Nodes, componentId, Quantitat
ORDER BY Quantitat DESC
```



	Nodes	componentid	Quantitat
19	["Individu"]	3215	12
20	["Individu"]	3614	12
21	["Individu"]	4455	12
22	["Individu"]	13848	12
23	["Individu"]	15970	12
24	["Individu"]	17499	12

Started streaming 11131 records after 68 ms and completed after 69 ms, displaying first 1000 rows.

### Explicació dels resultats obtinguts:

En aquests resultats es pot observar quines són les famílies més nombroses, que en aquest cas són de 12 persones.

---

### 3. No hi ha cap comunitat amb més d'un habitatge

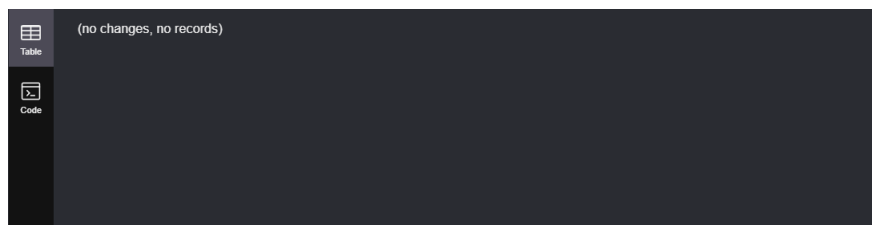
#### Motivació:

La comprensió de la dinàmica i la densitat de la població al segle XIX es pot aconseguir analitzant el desglossament de les llars per comunitat de la base de dades. Cap comunitat amb més d'un habitatge pot assenyalar un patró específic de dispersió de la població o un nivell de desenvolupament socioeconòmic baix en aquell moment, segons les circumstàncies.

#### Codi:

```
CALL gds.wcc.stream('myGraph2')
YIELD componentId, nodeId
WITH labels(gds.util.asNode(nodeId)) AS Nodes, componentId,
      count(*) AS Quantitat
WHERE 'Habitatge' IN Nodes AND Quantitat > 1

RETURN Nodes, componentId, Quantitat
ORDER BY componentId
```



#### Explicació dels resultats obtinguts:

Es pot observar clarament com no hi ha cap persona que estigui empadronada a dos habitatges diferents, per tant, no hi ha cap error d'aquest tipus a la base de dades.

### 3. Variació dels habitants per nucli familiar al llarg del temps

Primerament declarem el graf que s'usara per aquest tercer punt.

```
CALL gds.graph.project('myGraph4',
  {Individu:{label: 'Individu',properties:['Any_Padro']}}, ['FAMILIA'])
```

#### Motivació:

Un factor molt important que ajuda a comprendre com era la societat del segle XIX és la família, ja que antigament la vida intrafamiliar era probablement la prioritat més gran de les persones. Analitzant com canvien els nuclis familiars al llarg dels anys podem deduir com eren de freqüent els naixements i les defuncions per les famílies que conformaven la societat catalana abans dels descobriments medicinals dels quals fem ús sovint avui en dia.

#### Codi:

```
CALL gds.wcc.stream('myGraph4')
YIELD nodeId, componentId
WITH collect(nodeId) as Nodes, componentId
MATCH(i:Individu)
WHERE id(i) IN Nodes
RETURN i.Any_Padro AS Any, componentId, count(*) AS Nombre_Persones,
collect(DISTINCT i.Cognom) AS Cognoms
ORDER BY Cognoms[0], Any ASC)
```

Any	componentId	Nombre_Persones	Cognoms
1839	6184	4	["?", "nan"]
1878	9705	2	["abezuela", "aparicio"]
1889	17205	7	["adan", "moliner"]
1839	4839	4	["aguada", "nan"]
1857	7105	4	["aguade", "carcereny"]
1878	9501	5	["aguade", "carcereny", "morato"]
1881	12240	5	["aguade", "carcereny", "moreto"]
1889	15509	4	["aguade", "carcereny", "morato"]
1878	9854	9	["aguilar", "espany", "barres"]
1881	13442	8	["aguilar", "barres"]

#### Explicació dels resultats obtinguts:

El resultat d'aquesta consulta és una taula on es pot observar que els seus components són els que s'han demanat al return (l'any de padró, la ID del nucli familiar, el nombre de persones que el conformen i els cognoms que el caracteritzen).

Com que està ordenat per cognoms, es pot apreciar el canvi que han anat fent els nuclis familiars al llarg dels anys (no estan tots els anys consecutius, ja que no ho miraven cada any).

Es pot veure com varien el nombre de persones a mesura que passen els anys a causa de naixements i defuncions i fins i tot com augmenta o es redueix el nombre de cognoms que conforma cada nucli, perquè noves persones s'integraven al nucli (per exemple, mitjançant el matrimoni), i d'altres cognoms es perdien (per exemple, si tota la família tenia només filles).

#### 4. Quantes components connexes no estan connectades a cap node de tipus 'Habitatge'.

Primerament, crearem una projecció del graf original i seleccionarem només els nodes i relacions que ens interessin.

```
CALL gds.graph.project('myGraph4', ['Habitatge', 'Individu'], ['VIU'])
```

##### 1. Habitatges sense individus

###### Motivació:

L'anàlisi d'habitatges buits pot revelar tendències en la migració i la demografia d'aquest període. Podria mostrar regions on hi va haver un èxode de població, moviments migratoris significatius o canvis en l'estructura familiar.

###### Codi:

```
CALL gds.wcc.stream('myGraph4')
YIELD componentId, nodeId
WITH collect(DISTINCT labels(gds.util.asNode(nodeId))) AS
    Tipus_Nodes, componentId, collect(nodeId) AS nodes
WHERE size(Tipus_Nodes) = 1 AND NOT 'Individu' IN Tipus_Nodes[0]

MATCH (n)
WHERE id(n) IN nodes
RETURN n.Any_Padro, count(n)
```



	n.Any_Padro	count(n)
1	1866	3
2	1828	426
3	1833	9
4	1857	533
5	1878	2
6	1881	1

Started streaming 6 records after 97 ms and completed after 200 ms.

###### Explicació dels resultats obtinguts:

Amb el resultat obtingut podem observar la quantitat d'habitatges buits que hi havia a cada any. Això pot tenir correlació amb successos que van passar en aquell moment que van obligar la gent a migrar deixant-ho tot enrere.

## 2. Individus sense habitatges

### Motivació:

La informació sobre les persones sense llar pot donar una imatge més completa de les circumstàncies de vida de l'època, així com de les qüestions socials i econòmiques que els grups demogràfics van haver d'afrontar.

### Codi:

```
CALL gds.wcc.stream('myGraph4')
YIELD componentId, nodeId
WITH collect(DISTINCT labels(gds.util.asNode(nodeId))) AS Tipus_Nodes,
componentId, collect(nodeId) AS nodes
WHERE size(Tipus_Nodes) = 1 AND NOT 'Individu' IN Tipus_Nodes[0]

MATCH (n)
WHERE id(n) IN nodes

RETURN n.Any_Padro, count(n)
ORDER BY n.Any_Padro ASC
```



	n.Any_Padro	count(n)
1	1828	426
2	1833	9
3	1857	533
4	1866	3
5	1878	2
6	1881	1

Started streaming 6 records after 79 ms and completed after 134 ms.

### Explicació dels resultats obtinguts:

En aquest cas obtenim la informació contrària a l'anterior, és a dir, la quantitat de gent que no tenia casa. Això indica que probablement hi havia una situació econòmica molt dura.

Per exemple, comparant els dos primers anys amb els quals ens trobem veiem una diferència molt gran: l'any 1828, hi havia molta gent sense llar probablement a causa que s'acabava d'acabar la Guerra dels Malcontents. En canvi, l'any 1833, ja hi havia un ressorgiment econòmic el que va generar que més gent es pogués permetre una casa on viure.

---

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 aquest anàlisi:

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.

*Codi:*

```
MATCH (h1:Habitatge), (h2:Habitatge)
WHERE h1.Municipi=h2.Municipi AND
      apoc.text.sorensenDiceSimilarity(h1.Carrer,h2.Carrer)>0.5 AND
      h1.Carrer IS NOT NULL AND h1.Numero=h2.Numero AND
      h1.Numero<>-1 AND h1.Any_Padro>h2.Any_Padro

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.

*Codi:*

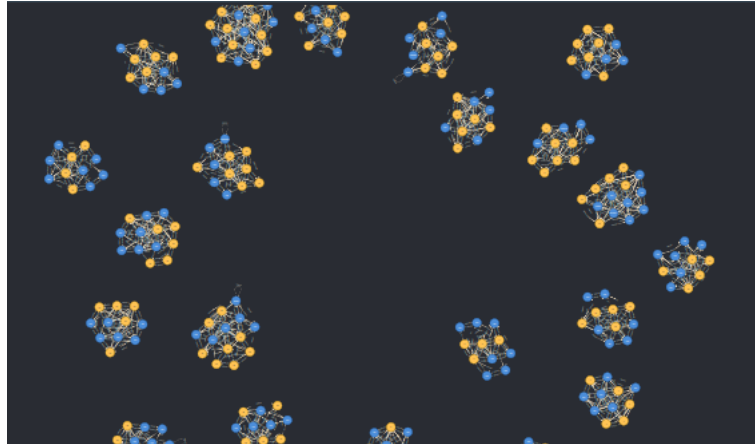
```
CALL gds.graph.project('myGraphs', {
  Individu: {
    label: 'Individu',
    properties: 'Id'
  },
  Habitatge: {
    label: 'Habitatge',
    properties: 'Id'
  }
}, {
  VIU: {
    type: 'VIU',
    orientation: 'NATURAL'
  },
  FAMILIA: {
    type: 'FAMILIA',
    orientation: 'NATURAL'
  },
  MATEIX_HAB: {
    type: 'MATEIX_HAB',
    orientation: 'NATURAL'
  }
})
```

---

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.

*Codi:*

```
CALL gds.nodeSimilarity.write('myGraphs', {  
  writeRelationshipType: 'SIMILAR',  
  writeProperty: 'score'  
}) yield nodesCompared, relationshipsWritten  
MATCH (i)-[:SIMILAR]->(o)  
RETURN i, o
```



*Explicació dels resultats obtinguts:*

Després de crear la relació s'ha provat de mostrar quines 'comunitats' s'han format mitjançant l'algorisme. L'algorisme utilitzat per trobar la similitud entre nodes considera que dos nodes són similars si aquests comparteixen diversos veïns iguals. De forma que el resultat obtingut és l'evolució de les famílies al llarg del temps, es poden veure els nuclis familiars al voltant de la mateixa casa tot i que estigui repetida, el que abans es veien com dues famílies de dos anys diferents i dues cases diferents, ara és un mateix conjunt d'habitatges i individus. En conclusió, el resultat aconseguit són els nuclis familiars independentment de l'any de padró.