

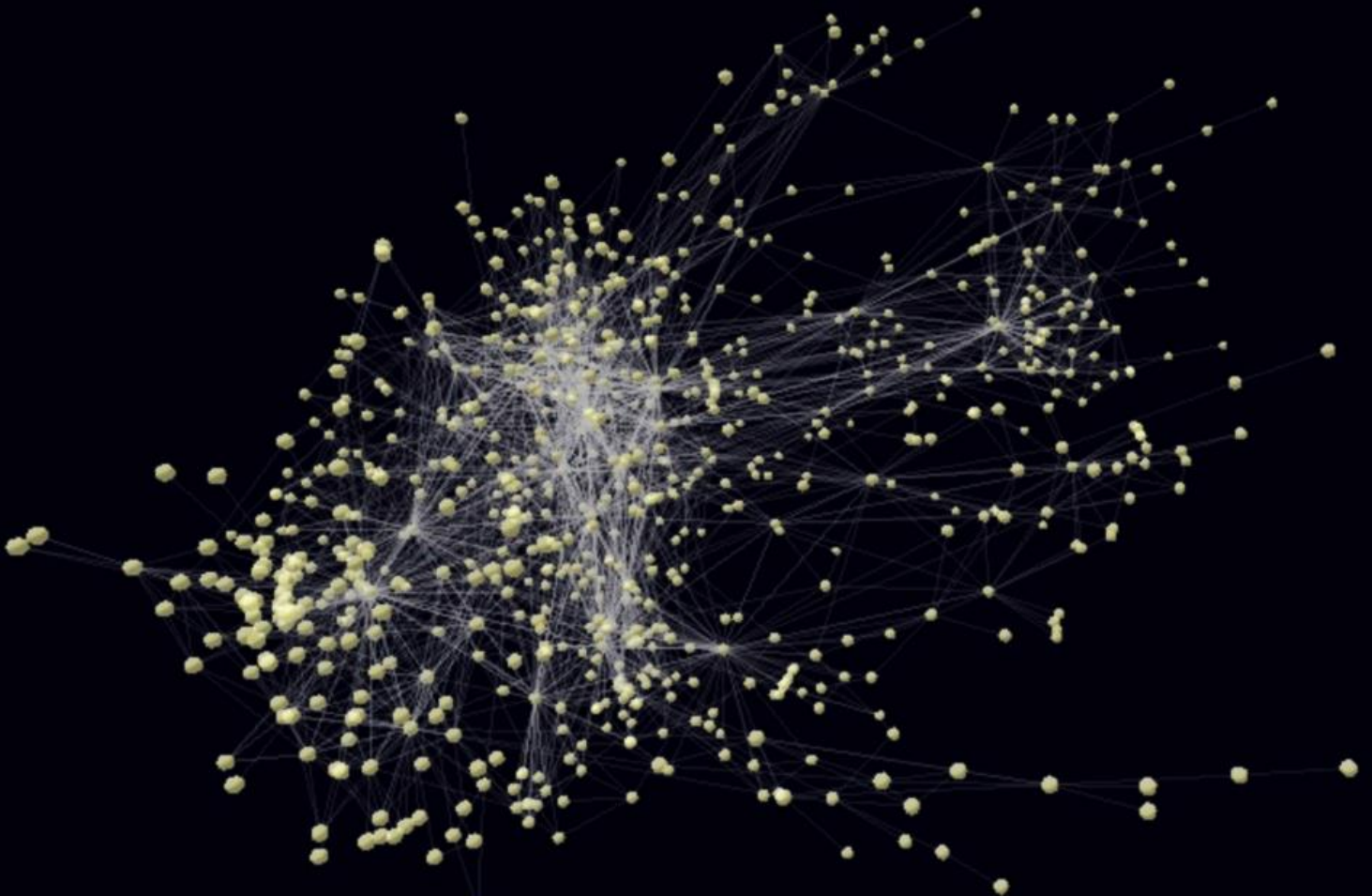
# neo4j

Dídac Alvarez - 1601114

Beatriz Salgado - 1605534

Carla Martínez - 1606635

Joan Paz - 1598851



# INTRODUCCIÓ

Durant la segona part del semestre de l'assignatura de Bases de Dades no Relacionals, hem après com funciona Neo4j. En aquest projecte mostrem el que hem après fent una serie d'exercicis que englove

## 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 afegixi les característiques allà on toqui.**

L'objectiu principal en aquest primer exercici és explicar com hem fet tots els imports i la creació de totes les relacions que hem creat en el nostre graf. Tot i així, també afegirem el codi utilitzat en cada apartat.

### 1. IMPORT INDIVIDUAL

Primerament, importarem les dades d'habitatges del csv "INDIVIDUAL.csv". En aquest csv trobarem les dades de cada persona/individu, per cada fila llegirem:

- El id (diferents per cada individu)
- Year, any de naixement que ho passarem a int
- El nom
- Els dos cognoms

Crearem un node Individu amb la id diferent de cada individu, després afegirem l'any, el nom i els cognoms. Finalment farem un recompte.

```
1
2 LOAD CSV WITH HEADERS FROM "file:///INDIVIDUAL.csv" AS fila
3 WITH toInteger(fila.Id) AS id,
4 toInteger(fila.Year) AS year, fila.name AS name, fila.surname AS surname,
5 fila.second_surname AS second_surname
6 MERGE (i:Individu{personId: id})
7 set i.year = year, i.name = name,
8 i.surname = surname, i.second_surname = second_surname
9 return count(i)
```

### 2. IMPORT HABITATGES

En segon lloc, importarem les dades d'habitatges del csv "HABITATGES.csv". En aquest cas per cada habitatge(fila) agafarem:

- El municipi
- La id de la llar, la qual ho convertirem a int
- L'any del Padró també com a int
- El carrer
- El numero

Per finalitzar crearem un node Habitatge amb totes aquestes característiques i farem un recompte.

```

1
2 LOAD CSV WITH HEADERS FROM "file:///HABITATGES.csv" AS fila
3 WITH fila.Municipi AS municipi, toInteger(fila.Id_Llar) AS id_llar,
4 toInteger(fila.Any_Padro) AS any_padro, fila.Carrer AS carrer,
5 fila.Numero AS numero
6 MERGE (h:Habitatge{id_llar: id_llar, municipi:municipi, any_padro:any_padro,carrer:carrer,
  numero:numero})
7 return count(h)

```

### 3. IMPORT SAME AS

En tercer lloc, importarem les dades d'habitatges del csv "SAME\_AS.csv".

Per cada fila llegirem els dos id, id\_A i id\_B, seguidament amb un match agafarem el nodes Individu amb aquest id respectivament, per últim crearem una relació SAME\_AS amb aquests nodes i farem el recompte de les relacions creades.

```

1
2 LOAD CSV WITH HEADERS FROM "file:///SAME_AS.csv" AS fila
3 WITH toInteger(fila.Id_A) AS id_A, toInteger(fila.Id_B) AS id_B
4 MATCH (i:Individu), (i2:Individu)
5 WHERE i.personId = id_A AND i2.personId = id_B
6 MERGE (i2)-[rel:SAME_AS]->(i)
7 RETURN count(rel)

```

### 4. IMPOR VIU

Seguidament, importarem les dades d'habitatges del csv "VIU.csv". En aquest import llegirem les dades de cada fila:

- Id del individu com ind
- L'any com year
- La id del habitatge com house\_id
- La location

Amb aquestes dades farem un match per trobar l'Individu amb la id llegida i l'habitatge que compleixi amb l'id de la llar y l'any del padró. Aleshores crearem una relació VIU amb l'Individu i l'Habitatge que hem localitzat amb el math, finalment farem un recompte de les relacions creades.

```

1
2 LOAD CSV WITH HEADERS FROM 'file:///VIU.csv' AS fila
3 WITH toInteger(fila.IND) as ind,
4 toInteger(fila.Year) as year,
5 toInteger(fila.HOUSE_ID) as house_id,
6 fila.Location as location
7 MATCH (i : Individu{personId:ind}),(h: Habitatge{id_llar:house_id,any_padro:year })
8 MERGE (i)-[rel:VIU]->(h)
9 RETURN COUNT(rel)

```

## 5. IMPORT FAMÍLIA

Per últim, importarem les dades d'habitatges del csv "FAMILIA.csv". Llegirem les id de cada Individu i la relació que tenen de cada fila del csv, seguidament farem un math per trobar aquest individus per finalment crear la relació FAMÍLIA que tindrà com a característica afegida quin tipus de relació tenen ( net, fill, esposa, jefe...).

No crearem relacions de cada tipus de relació ja que el que importa es que si cada parella de nodes estan relacionats o no, el tipus de relació només serà un atribut dins la relació. I com sempre per últim farem el recompte de les relacions.

```
1
2 LOAD CSV WITH HEADERS FROM 'file:///FAMILIA.csv' AS fila
3 WITH toInteger(fila.ID_1) as id_1, toInteger(fila.ID_2) as id_2,
   fila.Relacio_Harmonitzada as
4 relacio_harmonitzada
5 MATCH (i1:Individu{personId: id_1}),(i2:Individu{personId: id_2})
6 MERGE (i1)-[rel:FAMILIA {relacio_harmonitzada: relacio_harmonitzada}]- (i2)
7 RETURN count(rel)
```

## EXERCICI 2

Resoleu les següents consultes:

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

Codi:

```
1
2 match (h:Habitatge)-[:VIU]-(i:Individu)
3 where h.any_padro = 1866 and toUpper(h.municipi) = 'CR' and i.name <> 'null'
4 return count(i.name) as c, collect(distinct(i.name)) as names
```

Resultat :

```
c      names
337  ["miguel", "emilia", "rosa", "antonio", "elisa", "jose", "manuel", "teresa", "magdalena", "juan", "francisco", "salvador", "francisca", "pedro", "nan", "luis", "madrona",
```

Busquem un math entre individus amb una relació VIU amb un habitatge on l'any sigui 1866 i el municipi Castellví de Rosanes (CR) e individus que no siguin null, fem un count i creem una llista amb els noms.

2. Dels padrons de Sant Feliu de Llobregat (SFL) 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.

Codi:

```
1
2 match (i:Individu)-[:VIU]->(h:Habitatge)
3 where toUpper(h.municipi) = 'SFL' and h.any_padro < 1840
4 return h.any_padro as any, collect(distinct(h.id_llibre)) as id_llibres, h.any_padro
5 order by h.any_padro asc
```

Resultat:

any	id_IIars
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]
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, 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, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500]
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, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700]

Busquem un math entre individus amb una relació VIU amb un habitatge on l'any sigui menor que 1840 i el mucipi sigui Sant Feliu de Llobregat (SFLl). Crearem una llista de id diferents i farem el return.

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.

#### 3.1. Format de Llista:

##### 3.1.1.Consulta:

```
1
2 match(i:Individu)-[:VIU]->(h:Habitatge)<-[:VIU]-(i2:Individu)
3 where toLower(i.name) = 'rafel' and toLower(i.surname) = 'martí' and
4 toUpper(h.municipi) = 'SFLl' and h.any_padro = 1838
5 return i.name as nom, collect(i2.name) as convivents
```

##### 3.1.2.Subconsulta:

```
1
2 match(i:Individu)-[:VIU]->(h:Habitatge)
3 where toLower(i.name) = 'rafel' and toLower(i.surname) = 'martí' and
4 toUpper(h.municipi) = 'SFLl' and h.any_padro = 1838
5 match (i2:Individu)-[:VIU]->(h)
6 where i2.personId <> i.personId
7 with i.name as name, collect(i2.name) as names
8 return name, names
```

Resultat:

nom	convivents
"rafel"	["salvadora", "maria", "franco", "jph", "felipe", "jpha", "jpha", "miquel"]

Busquem dos individus que viuen en un mateix habitatge on un d'ells (i1) el seu nom es Rafael i el cognom marti, on el municipi del habitatge sigui SFLl i l'any del padró 1838 i retornem tots els individus i2 que compleixin això en format llista anomenada convivents.

En subconsulta busquem primer l'habitatge d'en Rafael i després busquem, amb un altre math, individus que viuen en aquell habitatge diferents d'en Rafael, i fem una llista anomenada names.

### 3.2 Format de Graf:

#### 3.2.1 Consulta:

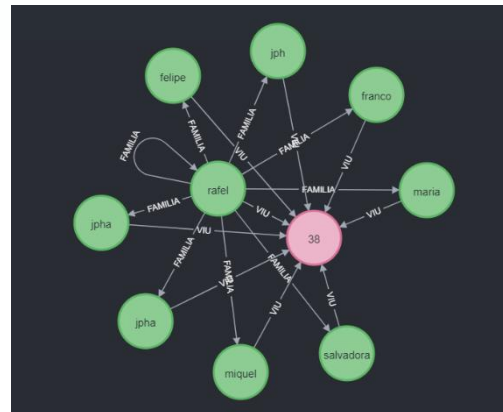
```
1
2 match(i:Individu)-[:VIU]->(h:Habitatge)<-[:VIU]-(i2:Individu)
3 where toLower(i.name) = 'rafel' and
4 toLower(i.surname) = 'marti' and
5 toUpper(h.municipi) = 'SFL' and h.any_padro = 1838
6 return i,i2,h
```

#### 3.2.2 Subconsulta:

```
1
2 match(i:Individu)-[:VIU]->(h:Habitatge)<-[:VIU]-(i2:Individu)
3 where toLower(i.name) = 'rafel' and
4 toLower(i.surname) = 'marti' and
5 toUpper(i.location) = 'SFL' and
6 i.year = 1838 and i2.year = 1838 and h.any_padro = 1838
7 return i,i2,h
```

Resultat:

Busquem dos individus que viuen en un mateix habitatge on un d'ells (i1) el seu nom es Rafael i el cognom marti, on el municipi del habitatge sigui SFL i l'any del padró 1838 i retornem tots els individus i2, l'habitatge i el node de Rafael.



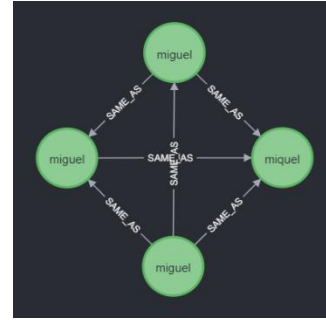
4. Retorna totes les aparicions de "Miguel ballester". Fes servir la relació SAME\_AS per poder retornar totes les instàncies, independentment de si hi ha variacions lèxiques (ex. diferents formes d'escriure el seu nom/cognoms). Mostra la informació en forma de subgraf.

Codi:

```
1
2 match (i:Individu)-[:SAME_AS]->(i2:Individu)
3 where (toLower(i.name) = 'miguel' and toLower(i.surname) = 'ballester') or
4 (toLower(i2.name) = 'miguel' and toLower(i2.surname) = 'ballester')
5 return i,i2
```

Resultat:

Busquem dos individus que tinguin una relació SAME\_AS on el nom i els cognoms siguin iguals transformant-los en minúscules i els retornem aquestes nodes.



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

Codi:

```
1
2 match (i:Individu)-[rel]->(i2:Individu)
3 where toLower(i.name) = 'antonio' and toLower(i.surname) = 'farran'
4 return i2.name as nom, i2.surname as cognom1, i2.second_surname as cognom2,
5 type(rel) as relacio
```

Resultat:

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

Busquem dos individus que tinguin una relació on un d'ells tingui com a nom Antonio i de cognom farran i retornem tots els nodes relacionats amb ell amb el nom, els dos cognoms i el tipus de relació.

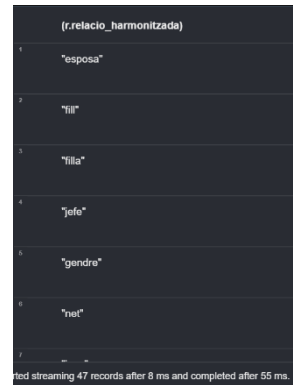
6. Llisteu totes les relacions familiars que hi ha.

Codi:

```
1
2 match (i:Individu)-[r:FAMILIA]->(i2:Individu)
3 where r.relacio_harmonitzada <> 'null'
4 return distinct(r.relacio_harmonitzada)
```

Resultat:

Busquem dos individus que tinguin una relació de tipus FAMILIA i que aquesta relació no sigui nula, retornarem els tipus de relació. Hi ha 47 resultats, tal i com es pot veure a la part inferior de la imatge.



	(r.relacio_harmonitzada)
1	"esposa"
2	"fill"
3	"filla"
4	"jefe"
5	"gendre"
6	"net"
7	

Started streaming 47 records after 8 ms and completed after 55 ms.

- 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.

Codi:

```
1
2 match (h:Habitatge)
3 where toUpper(h.municipi) = 'SFLL'
4 return h.carrer as carrer, h.numero as numero,
5 size(collect(distinct(h.any_padro))) as compt,
6 collect(distinct(h.any_padro)) as anys, collect(h.id_llibre) as ids
7 order by size(collect(distinct(h.any_padro))) desc
8 limit 10
```

Resultat:

carrer	numero	compt	anys	ids
"iglesia"	"8"	5	[1833, 1839, 1878, 1881, 1889]	[102, 728, 520, 516, 500]
"creus"	"16"	5	[1833, 1839, 1878, 1881, 1889]	[158, 616, 473, 372, 388]
"iglesia"	"3"	5	[1833, 1839, 1878, 1881, 1889]	[97, 723, 724, 516, 511, 484]
"iglesia"	"9"	5	[1833, 1839, 1878, 1881, 1889]	[104, 729, 730, 521, 517, 487]
"iglesia"	"11"	5	[1833, 1839, 1878, 1881, 1889]	[106, 732, 523, 520, 488]
"creus"	"9"	5	[1833, 1839, 1878, 1881, 1889]	[150, 610, 467, 367, 347]
"iglesia"	"1"	5	[1833, 1839, 1878, 1881, 1889]	[94, 721, 514, 509, 483]
"iglesia"	"2"	5	[1833, 1839, 1878, 1881, 1889]	[96, 722, 515, 510, 498]
"iglesia"	"4"	5	[1833, 1839, 1878, 1881, 1889]	[98, 725, 517, 512, 499]
"creus"	"18"	5	[1833, 1839, 1878, 1881, 1889]	[160, 618, 475, 374, 389]

Busquem habitatges del municipi SFLL, de cada habitatge retornarem, el carrer el numero, el numero de padrons( per fer això farem un size d'una llista on estan els anys de padro diferents), la llista dels anys i per ultim retornarem una llista amb el ids de la llar.



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.

Codi:

```
1
2 match (h:Habitatge{municipi:"CR"})-[:VIU]-(i:Individu)-[r:FAMILIA]-(i2:Individu)
3 where toLower(r.relacio_harmonitzada) = "fill" or toLower(r.relacio_harmonitzada) =
   "filla"
4 With collect(i2) as fills, i
5 where size(fills)>3
6 return i.name, i.surname, i.second_surname, size(fills) as num_fills
7 order by num_fills desc limit 20
```

Resultat:

i.name	i.surname	i.second_surname	num_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
"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

Buscarem individus amb una relació de FAMILIA que tinguin una relació VIU amb una habitatge de CR(Castellví de Rosanes). Voldrem els nodes i2 que tinguin una relació FAMILIA dse tipus fill/filla amb i, crearem una llista amb tots aquest nodes i2. Retornarem només el cap de família (i) on aquesta llista tingui un size mayor de 3.

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.

Codi:

```
1
2 match(h:Habitatge)
3 where h.any_padro = 1881 and h.municipi = 'SFL'
4 with count(distinct(h)) as num
5 match(h1:Habitatge)-[:VIU]-(i:Individu)-[r:FAMILIA]->(:Individu)
6 where h1.any_padro = 1881 and h1.municipi = 'SFL' and r.relacio_harmonitzada
   starts with 'fill'
7 return count(i) as total_fills, num as num_llars, toFloat(count(i))/toFloat(num)
   as Mitjana
```

Resultat:

total_fills	num_llars	Mitjana
1292	596	2.1677852348993287

Farem una subconsulta, en primer lloc buscarem els habitatges amb any de padró igual a 1881 i el municipi SFL (Sant feliu de Llobregat ) i farem un count dels distins nodes habitatges.

A continuació buscarem una relació de FAMILIA entre dos nodes de tipus fill/filla utilitzant la comanda starts with per trobar el tipus fill com filla. Aquests nodes han de viure a un habitatge amb l'any de padró igual a 1881 i el municipi SFL (Sant feliu de Llobregat ). Farem un recompte de total de fills i per fer la Mitjana el dividirem amb el count fet a la primera consulta amb el numero de habitatges.

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

Codi:

```
1
2 match (h:Habitatge)-[:VIU]-(i:Individu)
3 where toUpper(h.municipi) = "SFL"
4 with h.carrer as carrer, count(*) as num, h.any_padro as any_padro order by
5 any_padro,num
6 return any_padro,collect(carrer)[0] as carrer,collect(num)[0] as num_habitants
```

Resultat:

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

Buscarem els habitatges i els individus que visquen en ell on el municipi sigui SFL (Sant feliu de Llobregat) per cada habitatge guardarem el nom del carrer i el numero de individus en dos llistes.

## EXERCICI 3

- a) Estudi de les components connexes (cc) i de l'estructura de les component en funció de la seva mida. A continuació uns indiquem algunes consultes que podeu fer per explorar les dades:

Per aquest primer apartat treballarem amb aquests grafs en comptes del graf que tenim guardat en la base de dades del Neo4j.

```
1
2 call gds.graph.create('graph_3a_ind',['Individu'],'VIU')
3
4 call gds.graph.create('graph_3a_ind_hab',['Individu','Habitatge'],'VIU')
5
6 call gds.graph.create.cypher('graph_3a_1889_d',
7   'match (h:Habitatge) where h.any_padro = 1889 return h.id_llar as id',
8   'match (i:Individu)-[:VIU]->(h) return h.id_llar as source, i.personId as
   target',
9   {validateRelationships:False})
10
```

-Taula que mostra les comunitats entre individus dins la mateixa bombolla familiar. (mostrem una petita mostra de les comunitats)

**Objectiu:** Estem en una situació de pandèmia i s'ha donat el cas que certs individus han agafat el virus. Per evitar els contagis, suposant que els contactes més propers de l'infectat (familiars) són propensos a ser contagiats, es volen detectar els grup bombolla de famiars. D'aquesta manera, només s'haurien d'aïllar aquells grup bombolla que tinguin un infectat entre ells, evitant fer una quarentena de tots els habitants d'un municipi, per exemple.

```
1
2 CALL gds.wcc.stream('graph_3a_ind')
3 YIELD componentId, nodeId
4 RETURN componentId, collect(nodeId) AS nodes
```

**Exemple:** si l'individu amb id 7 agafa el virus, els individus amb id 8, 9, 10 i 11 s'haurien d'aïllar també, ja que són els més propensos de contagiar-se.

componentid	nodes
0	[0, 1, 2, 3, 4, 5, 6, 52]
7	[7, 8, 9, 10, 11]
12	[12, 13, 14, 15, 16, 17, 18, 19]
20	[20, 21, 22, 23, 24]
25	[25, 26, 27, 295, 296, 297]
28	[28, 29, 30, 31, 32, 33, 323]

**-Taula agrupant els resultats segons si la mida de la cc és 2.**

**Objectiu:** Es vol saber quins individus han viscut sols en algun moment de la seva vida per fer una enquesta .

nodes
[121, 17635]
[277, 17646]
[312, 17648]
[340, 17651]
[2855, 19100]
[2913, 19171]

```

1
2 CALL gds.wcc.stream('graph_3a_ind_hab')
3 YIELD componentId, nodeId
4 with componentId, collect(nodeId) AS nodes, size(collect(nodeId)) as mida
5 where mida = 2
6 return nodes

```

**Resultat:**

Tindríem una sortida de les dades, com es veu just a l'esquerra, on es projecta un node Individu i un node Habitatge. El resultat ha estat que, amb tota la informació dels padrons al llarg dels anys que ens han facilitat, han arribat a haver 132 individus que han viscut sols en algun moment de la seva vida (pot incloure una mateixa persona però en altres anys).

**-Saber quants habitatges ha hagut a l'any 1889.**

**Objectiu:** Es vol saber la quantitat d'habitatges que ha hagut el 1889 per comparar la mateixa informació però d'altres anys.

```

1
2 CALL gds.wcc.stream('graph_3a_1889_d')
3 YIELD componentId, nodeId
4 with componentId, collect(nodeId) AS nodes
5 return count(nodes)

```

**count(nodes)**

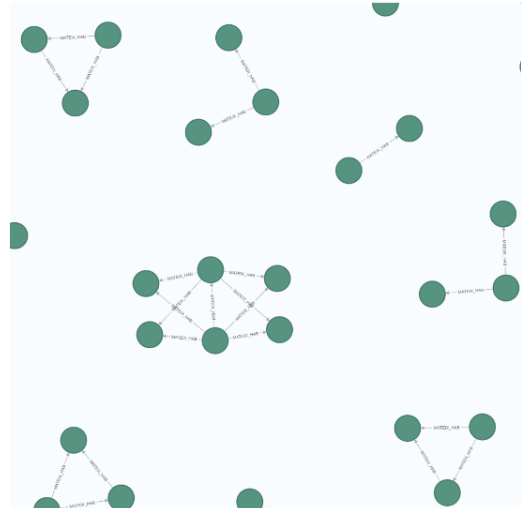
645

**Resultat:** es pot veure amb l'imatge de l'esquerra com en l'any 1886 es van registrar 645 habitatges.

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:

- 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.

```
match (h1:Habitatge),(h2:Habitatge)
where h1.carrer = h2.carrer and h1.numero =
h2.numero and h1.municipi = h2.municipi and
h1.carrer <> "null" and h2.carrer <> "null" and
h1.numero <> "null" and h2.numero <> "null" and
h1.any_padro < h2.any_padro
WITH h2,h1 ORDER BY h1.AnyPadro
merge(h1)-[:MATEIX_HAB]-(h2)
```



S'han creat 1752 arestes.

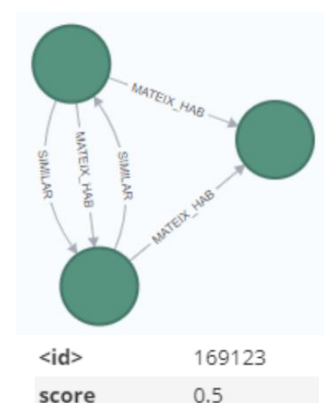
- Creeu un graf en memòria que inclogui els nodes Individu i Habitatge i les relacions VIU, FAMILIA, MATEIX\_HAB que acabeu de crear.

```
CALL gds.graph.create("habitatges_graph", ["Habitatge", "Individu"], ["FAMILIA", "VIU",
"MATEIX_HAB"])
```

nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	createMillis
<pre>{   "Habitatge": {     "label": "Habitatge",     "properties": {       }     },     "Individu": {       "label": "Individu",       "properties": {         }       }     }   }</pre>	<pre>{   "MATEIX_HAB": {     "orientation": "NATURAL",     "aggregation": "DEFAULT",     "type": "MATEIX_HAB",     "properties": {       }     },     "FAMILIA": {       "orientation": "NATURAL",       "aggregation": "DEFAULT",       "type": "FAMILIA",       "properties": {         }       }     }   }</pre>	"habitatges_graph"	21288	29200	68

- 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.

```
call gds.nodeSimilarity.write('habitatges_graph', {
writeRelationshipType: 'SIMILAR',
writeProperty: 'score'
})
yield nodesCompared, relationshipsWritten
```



## REPARTICIÓ

### **S'ha repartit la feina de la següent manera:**

Perquè cadascun dels integrants del grup participés en cada exercici, es va repartir el contingut de tal manera de que tothom toqués una mica de tot.

En la part d'importació, cada integrant va fer una importació diferent i un d'aquests en va fer la que faltava.

En la part de les consultes es va aplicar el mateix. A cada membre li va tocar fer un parell o 3 consultes.

En el tercer exercici, es va decidir fer-lo conjuntament, ja que és una part del treball que no és tan objectiva com les altres.

Per altra banda, la realització de les tres parts del treball ha estat feta en un drive. L'entrega corresponent al github s'ha fet un cop el projecte va estar acabat.