



**Universitat Autònoma
de Barcelona**

BASES DE DADES NO RELACIONALS

PROJECTE **Neo4j**

GRUP: 82_4

NIU: 1641111 - **Alumne:** Mar Blázquez

NIU: 1636713 - **Alumne:** Laura Marín

NIU: 1636465 - **Alumne:** Marta Monsó

NIU: 1637908 - **Alumne:** Mercè De la Torre

ÍNDEX

0. Organització de la feina	2
1. Importació de les dades en la BD de Neo4j del projecte	3
Constrains i Indexos	3
2. Resolució de consultes	5
3. Anàlisi de les dades del graf	14

0. Organització de la feina

Per tenir un lloc on guardem els codis generats i veure-hi les actualitzacions, hem utilitzat GitHub. El nostre repositori és:

https://github.com/1636465/Projecte_Neo4j_BDNR

I per poder treballar remotament al mateix temps en l'informe, hem fet ús de l'opció de compartir document en el Google.

A més, per tenir un seguiment de l'evolució del projecte, hem declarat dates per posar tot el que s'ha fet fins en aquell moment en comú. Llavors, ens hem obligat a tenir certes parts acabades, definint un data límit, per poder anar avançant al mateix moment.

Pel que fa l'organització de la feina, el nostre grup ha optat per repartir els diferents exercicis equitativament, de manera que no quedés descompensat i que una de nosaltres tingués una càrrega de treball més gran.

Així doncs, el primer exercici l'hem repartit entre dues persones i la resta, totes hem contribuït realitzant una part de la feina en qüestió. A continuació adjuntem una taula reflectint aquesta distribució.

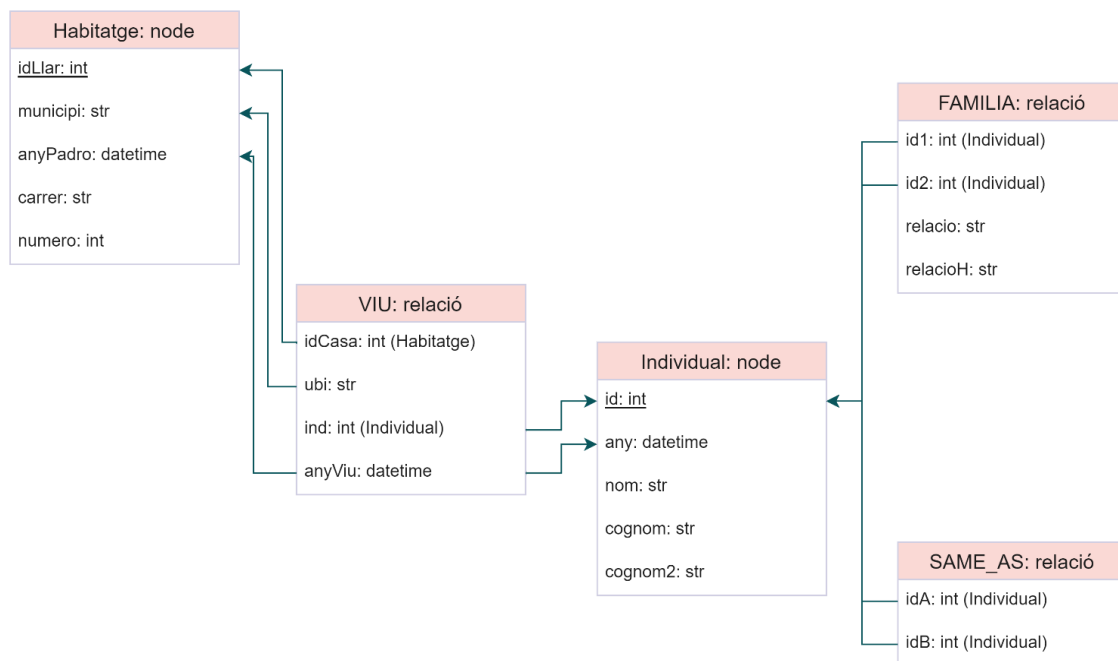
Exercici 1	Exercici 2	Exercici 3
<ul style="list-style-type: none"> • Mercè De la Torre • Marta Monsó 	<ul style="list-style-type: none"> • Marta Monsó Consultes: 4, 8, 12 • Laura Marin Consultes: 2, 7, 10 • Mar Blazquez Consultes: 3, 6, 9, 13 • Mercè De la Torre Consultes: 1, 5, 11 	<ul style="list-style-type: none"> • Mar Blázquez i Laura Marín <p>Apartat: a)</p> <ul style="list-style-type: none"> • Marta Monsó i Mercè De la Torre <p>Apartat: b)</p>

Taula 1: Organització del grup

1. Importació de les dades en la BD de Neo4j del projecte

Per llegir els csv, hem utilitzat les funcions LOAD i MERGE, la primera per carregar el fitxer amb les dades al Neo4j i la segona ens ha servit per crear els nodes o arestes al nostre graf. Evidentment, n'hem fet servir més, tanmateix aquestes dues són les més rellevants.

De manera que, com deia a l'enunciat, hem obviat i per tant, no hem afegit, aquells nodes que tenen NULL al camp de id de Municipi, Habitatge o Individual. I per tenir un esquema que ens ajudi a saber el nom dels atributs que hem carregat a la nostra base de dades, hem confeccionat un diagrama de relació indicant els nous noms atribuïts.



Gràfic 1: Diagrama de Relació de la Base de Dades

Constrains i Indexos

Pels nodes o relacions creats pels CSV següents hem creat alguns constraints que hem vist adients per la implementació del treball:

CSV Individual

En aquest cas, afegim un *uniqueness constraint* per garantir que el valor Id sigui únic per a tots els nodes de tipus Individual:

```
CREATE CONSTRAINT UniqueidIndividualConstraint
FOR (i:Individual) REQUIRE i.id IS UNIQUE;
```

També, hem creat una *existence constraint* per la propietat Id de Individual per assegurar que si un node es modifica o es crea tingui el Id que necessitem:

```
CREATE CONSTRAINT ExistsIndId FOR (i:Individual) REQUIRE i.id IS NOT NULL;
```

CSV Habitatges

Hem afegit un *existence constraint* per garantir que els valors IdLlar i municipi existeixin, és a dir, no siguin NULL quan creem o modifiquem un node de tipus Habitatge.

```
CREATE CONSTRAINT ExistsIdhabitatge
FOR (h:Habitatge) REQUIRE h.idLlar IS NOT NULL;
```

```
CREATE CONSTRAINT Existsmunicipihabitatge
FOR (h:Habitatge) REQUIRE h.municipi IS NOT NULL;
```

A més a més, hem inicialitzat un *node key* que garanteix que un conjunt de valors per les propietats, en aquest cas idLlar, municipi i anyPadro per un node d'habitatge sigui únic:

```
CREATE CONSTRAINT UniqueIdMunicipiConstraint
FOR (h:Habitatge) REQUIRE (h.idLlar, h.municipi, h.anyPadro) IS NODE KEY;
```

Aquí ens hem adonat que el idLlar no era únic per cada node habitatge diferent, sinó que el que no podia passar era que un mateix id i municipi es repetís per un mateix anyPadro. Aleshores, hem decidit crear per les tres propietats en conjunt un *node key*.

En el cas de la relació família com que acceptem que una relació pugui tenir la propietat relacio i/o relacioH a Null, no és precís generar un Constraint ja que no podem assegurar que existeix un valor i en els casos de la relació de VIU i SAME_AS no tenim cap propietat en la relació, llavors tampoc és possible generar un Constraint per aquests casos.

Per crear els índexs ens hem fixat amb les consultes posteriors que hem de realitzar i llavors, hem intentat buscar-ne alguns que fossin eficients a l'hora d'executar les consultes. Hem realitzat 3 *single-property Index* per a les propietats de nom, any i anyPadro i 2 *composite-property Index* pel conjunt de anyPadro-municipi i carrer-numero.

```
CREATE INDEX nomIndividu FOR (i:Individu) ON (i.nom);
```

```
CREATE INDEX anymunicipiHabitatges FOR (h:Habitatge) ON (h.anyPadro,h.municipi);
```

```
CREATE INDEX anyIndividu FOR (i:Individu) ON (i.any);
```

```
CREATE INDEX anyPadro FOR (h:Habitatge) ON (h.anyPadro);
```

```
CREATE INDEX carrernumeroHabitatges FOR (h:Habitatge) ON (h.carrer, h.numero);
```

2. Resolució de 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 (h:Habitatge {anyPadro: datetime("1866"), municipi: "CR"}) <— [:VIU] — (i:Individual)
```

```
RETURN COUNT(i), collect(i.cognom)
```

RESPOSTA:

Número d'habitants = 337

Llista = ["rusell", "olle", "suñol", "olle", "suñol", "suñol", "julibert", "julibert"]

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 (p:Individual)-[:VIU]->(h:Habitatge {municipi: 'SFLL'})
```

```
WHERE p.cognom IS NOT NULL AND p.cognom <> 'nan'
```

```
RETURN h.anyPadro AS anyPadro, count(p) AS numHabitants, collect(DISTINCT p.cognom) AS cognoms
```

RESPOSTA:

"anyPadro"	"numHabitants"	"cognoms"
"1833-01-01T00:00:00Z"	345	["roig", "aloma", "guiu", "casas", "coca", "majo", "parellada", "canalias", "camprecios", "ribas", "ribot", "v
"1838-01-01T00:00:00Z"	71	["canameras", "gullart", "bonsoms", "sola", "ramos", "pares", "comellas", "molins", "marles", "fabregas", "fe
"1839-01-01T00:00:00Z"	492	["ribas", "illegible", "roig", "parera", "camprubi", "aloma", "majo", "pañella", "?", "caralto", "camprecios",
"1878-01-01T00:00:00Z"	2739	["pahisa", "sadurni", "vidal", "sarrio", "anavall", "corrns", "amigo", "galtes", "castells", "marti", "planas
"1881-01-01T00:00:00Z"	2999	["alsina", "esteba", "casas", "marti", "olle", "castells", "tort", "sanllehi", "nin", "mas", "sala", "ferran", "
"1889-01-01T00:00:00Z"	3109	["mas", "sala", "deu", "ferres", "marles", "arbulos", "serralabos", "codinachs", "pelegri", "tutusaus", "carls

3. Dels padrons de Sant Feliu de Llobregat (SFL) d'entre 1800 i 1845 (no inclosos), 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:

MATCH (h:Habitatge{municipi:"SFL"})

WHERE datetime(h.anyPadro).year > 1800 **AND** datetime(h.anyPadro).year < 1845

RETURN h.municipi **AS** Poblacio, datetime(h.anyPadro).year **AS** anyPadro, COLLECT(h.idLlar) **AS** identificadorsHabitatges

ORDER BY anyPadro

RESPOSTA:

"Poblacio"	"anyPadro"	"identificadorsHabitatges"
"SFL"	1833	[10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000]
"SFL"	1838	[314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000]
"SFL"	1839	[369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000]

4. 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 (SFL). Retorna la informació en mode graf i mode llista.

CODI:

MATCH (p:Individual)-[:VIU]->(h:Habitatge)

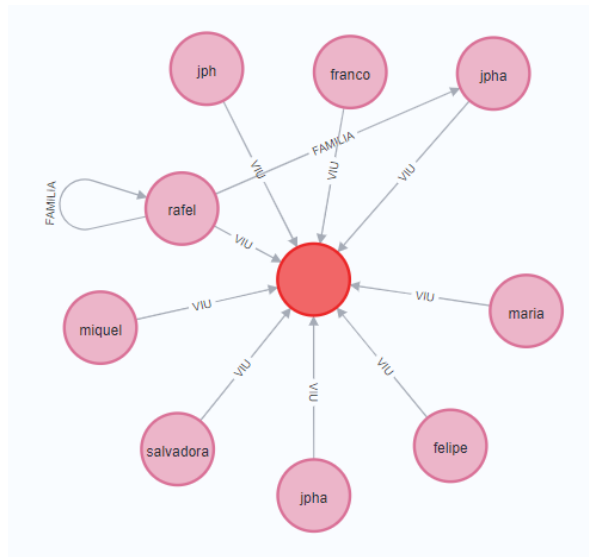
WHERE p.name="rafel" **AND** p.surname = "marti" **AND** p.second_surname= "nan" **AND** h.municipi = "SFL" **AND** datetime(h.anyPadro).year = 1838

MATCH (h)-[:VIU]-(other:Individual)

RETURN other.name

RESPOSTA:

Llista → ["jpha", "felipe", "maria", "jpha", "franco", "jph", "miquel", "salvadora"]

Graf

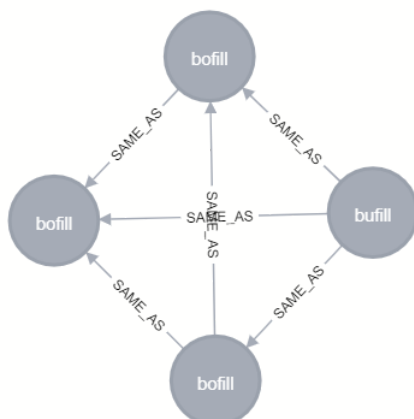
5. Retorna totes les aparicions de "miguel estape bofill". Fes servir la relació SAME_AS per poder retornar totes les instàncies, independentment de s'hi ha variacions lèxiques (ex. diferents formes d'escriure el seu nom/cognoms). Mostra la informació en forma de subgraf.

CODI:

```
MATCH (i:Individual)-[:SAME_AS]-(i2: Individual)
```

```
WHERE i.nom = "miguel" AND i.cognom = "estape" AND i.cognom2 = "bofill"
```

```
RETURN i, i2
```

RESPOSTA:

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 (i:Individual)-[:SAME_AS]-(i2:Individual)`

`WHERE i.nom = "miguel" AND i.cognom = "estape" AND i.cognom2 = "bofill"`

`RETURN DISTINCT i.nom AS nom, COLLECT(DISTINCT i2.cognom) AS cognoms,
COLLECT(DISTINCT i2.cognom2) AS cognoms2`

RESPOSTA:

"nom"	"cognoms"	"cognoms2"
"miguel"	["estape"]	["bofill", "bucfill"]

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:Individual)-[r:FAMILIA]->(rel:Individual)`

`WHERE p.nom = 'benito' AND p.cognom = 'julivert'`

`RETURN rel.nom AS Nom, rel.cognom AS Cognom1, rel.cognom2 AS Cognom2, r.relacioH AS
TipusRelacio`

RESPOSTA:

"Nom"	"Cognom1"	"Cognom2"	"TipusRelacio"
"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:Individual)-[r:FAMILIA]->(rel:Individual)
```

```
WHERE p.nom = 'benito' AND p.cognom = 'julivert' AND (r.relacioH = 'fill' OR r.relacioH = 'filla')
```

```
RETURN rel.nom AS Nom, rel.cognom AS Cognom1, rel.cognom2 AS Cognom2, r.relacioH AS TipusRelacio
```

```
ORDER BY Nom
```

RESPOSTA:

Nom	Cognom1	Cognom2	TipusRelacio
"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 (p1)-[rel:FAMILIA]->(p2)
```

```
WHERE rel.relacioH <> "null"
```

```
RETURN collect(DISTINCT(rel.relacioH)) AS Relacions
```

RESPOSTA:

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

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 número). 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.carrer **IS NOT NULL** AND h.numero **IS NOT NULL** AND h.municipi = "SFLL"

WITH h.carrer **AS** carrer, h.numero **AS** numero, **count**(*) **AS** totalPadrons, **collect**(**DISTINCT** h.anyPadro) **AS** anysPadro, **collect**(**DISTINCT** h.idLlar) **AS** idsLlar

RETURN carrer, numero, totalPadrons, anysPadro, idsLlar

ORDER BY totalPadrons **DESC**

LIMIT 15

RESPOSTA:

"carrer"	"numero"	"totalPadrons"	"anysPadro"	"idsLlar"
"falguera"	5	9	["1833-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[247,
"falguera"	3	8	["1833-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[245,
"falguera"	22	7	["1833-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[269,
"san antonio"	1	7	["1889-01-01T00:00:00Z"]	[562,
"carretera"	98	6	["1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[112,
"carretera"	141	6	["1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[172,
"falguera"	7	6	["1833-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[251,
"carretera"	36	6	["1838-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[354,
"creus"	21	6	["1839-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[621,
"creus"	12	6	["1839-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[613,
"carretera"	194	6	["1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[229,
"carretera"	85	6	["1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[83, 8
"carretera"	28	6	["1838-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[346,
"carretera"	100	6	["1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[115,
"creus"	6	6	["1839-01-01T00:00:00Z", "1878-01-01T00:00:00Z", "1881-01-01T00:00:00Z", "1889-01-01T00:00:00Z"]	[607,

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:

```
MATCH (h:Habitatge)-[:VIU]-(i:Individual)-[fam:FAMILIA]->(f:Individual)
```

```
WHERE h.municipi = "CR" AND (fam.relacio CONTAINS "hij" OR fam.relacioH CONTAINS "fill")
```

```
WITH i, count(f) AS num_fills
```

```
WHERE num_fills > 3
```

```
RETURN i.nom AS nom_cap_familia, i.cognom AS cognom_cap_familia, num_fills
```

```
ORDER BY num_fills DESC
```

```
LIMIT 20
```

RESPOSTA:

"nom_cap_familia"	"cognom_cap_familia"	"num_fills"
"pablo"	"astruch"	7
"benito"	"julivert"	6
"jose"	"canals"	6
"pedro"	"bargallo"	6
"jose"	"canals"	6
"jose"	"olle"	6
"jose"	"rafuls"	5
"jaime"	"jarrey"	5
"pablo"	"bargallo"	5
"francisco"	"aregay"	5
"pablo"	"canals"	4
"ramon"	"canals"	4
"jaime"	"gallofre"	4
"tomas"	"parera"	4
"juan"	"julibert"	4
"estevan"	"gallofre"	4

"cristobal"	"olle"	4
"pedro"	"farres"	4
"jose"	"llopart"	4
"jose"	"suñol"	4

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 (h:Habitatge {municipi: "SFLL", anyPadro: datetime("1881-01-01T00:00:00Z")})

WITH COUNT(DISTINCT h) AS nombreHabitatges

RETURN nombreHabitatges

}

MATCH (h:Habitatge {municipi: "SFLL", anyPadro:

datetime("1881-01-01T00:00:00Z"))<-[:VIU]-(i:Individual)

WHERE (((i)-[:FAMILIA {relacio: "hijo"}]->) OR (i)-[:FAMILIA {relacio: "hija"}]->) OR ((i)-[:FAMILIA {relacioH: "fill"}]->) OR (i)-[:FAMILIA {relacioH: "filla"}]->)) OR (((i)-[:FAMILIA {relacio: "hijo"}]->) OR (i)-[:FAMILIA {relacio: "hija"}]->) AND ((i)-[:FAMILIA {relacioH: "fill"}]->) OR (i)-[:FAMILIA {relacioH: "filla"}]->))

WITH COUNT(i) AS totalFillsPerHabitatge, nombreHabitatges

RETURN totalFillsPerHabitatge, nombreHabitatges,

toFloat(totalFillsPerHabitatge)/toFloat(nombreHabitatges) AS mitjanaFillsPerHabitatge

RESPOSTA:

totalFillsPerHabitatge	nombreHabitatges	mitjanaFillsPerHabitatge
465	596	0.7802013422818792

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:

```
MATCH (i:Individual)-[:VIU]->(h:Habitatge)
```

```
WHERE h.municipi = 'SFL'
```

```
WITH h.anyPadro AS anyPadro, h.carrer AS carrer, COUNT(i) AS numHabitantes
```

```
ORDER BY anyPadro ASC, numHabitantes ASC
```

```
WITH anyPadro, COLLECT({carrer: carrer, numHabitantes: numHabitantes})[0] AS carrerMenosHabitantes
```

```
CALL {
```

```
    WITH carrerMenosHabitantes
```

```
RETURN carrerMenosHabitantes.carrer AS Carrer, carrerMenosHabitantes.numHabitantes AS NumHabitantes
```

```
    ORDER BY carrerMenosHabitantes.carrer ASC
```

```
}
```

```
RETURN datetime(anyPadro).year as Year, Carrer, NumHabitantes
```

```
ORDER BY anyPadro ASC
```

RESPOSTA:

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

3. Anàlisi de les dades del graf

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

Creem el graf amb la següent comanda:

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

La comanda `CALL gds.graph.project('graf',['Habitatge','Individual'],['VIU', 'FAMILIA', 'SAME_AS'])` a Neo4j utilitza el procediment `gds.graph.project` de la biblioteca Graph Data Science (GDS) per crear un subgraf projectat a partir d'un graf existent.

Aquesta és l'explicació dels paràmetres utilitzats en aquesta comanda:

- `'graf'` és el nom del graf original del qual es vol projectar un subgraf.
- `['Habitatge','Individual']` és una llista d'etiquetes de nodes que s'inclouran al subgraf projectat. En aquest cas, els nodes etiquetats com a `'Habitatge'` i `'Individual'` seran inclosos.
- `['VIU', 'FAMILIA', 'SAME_AS']` és una llista de tipus de relació que s'inclouran al subgraf projectat. Aquí s'hi inclouen els tipus de relació `'VIU'`, `'FAMILIA'` i `'SAME_AS'`.

El procediment `gds.graph.project` pren aquests paràmetres i crea un nou graf que conté només els nodes i les relacions que compleixen les etiquetes i els tipus de relació especificats.

- Taula agrupant els resultats segons la mida de la cc.

```
CALL gds.wcc.stream('graf')
```

```
YIELD componentId, nodeId
```

```
RETURN componentId, size(collect(nodeId)) AS mida, collect(nodeId) AS nodes
```

```
ORDER BY mida DESC
```

"componentId"	"mida"	"nodes"
12	7613	[12,61,62,63,66,67,68,72,75,78,81,85,86,88,89,103,104,106,107,110,118,119,121,122,123,125,126,127,130,131,132,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,1038,1039,1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055,1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071,1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087,1088,1089,1090,1091,1092,1093,1094,10

L'execució d'aquesta consulta ens proporciona una taula que mostra el componentId, la mida de cada component (mida) i la llista de nodes que pertanyen a cada component (nodes). Això ens permet analitzar com es distribueixen les components connexes en termes de la seva mida.

- **Distribució de tipus de nodes (Individu o Habitatge) segons la mida de la cc.**

```
CALL gds.wcc.stream('graf')
```

YIELD nodeId, componentId

```
WITH componentId, size(collect(DISTINCT(nodeId))) AS mida, collect(nodeId) AS nodes
```

MATCH(i:Individual),(h:Habitatge)

WHERE id(i) IN nodes AND id(h) IN nodes

```
RETURN componentId, count(DISTINCT(i)) AS Individu, count(DISTINCT(h)) AS
Habitatge
```

ORDER BY Individu DESC

"componentId"	"Individu"	"Habitatge"
12	6526	1087
350	118	16
76	117	21
312	116	18
379	108	22
105	90	18

610	84	15
282	82	15
644	79	12
1377	63	10
363	61	6
259	53	6
689	53	8
262	44	6
708	44	6
2161	42	4

etc.

- Per cada municipi i any el nombre de parelles del tipus: (Individu)—(Habitatge)

MATCH (i:Individual)-[:VIU]->(h:Habitatge)

RETURN h.municipi **AS** municipi, datetime(h.anyPadro).year **AS** any, count(*) **AS** numParelles

ORDER BY municipi, any

"municipi"	"any"	"numParelles"
"CR"	1866	337
"SFLL"	1833	1433
"SFLL"	1838	287
"SFLL"	1839	1946
"SFLL"	1878	2745
"SFLL"	1881	3000
"SFLL"	1889	3117

Aquesta consulta recupera les parelles (Individu)—(Habitatge) mitjançant la relació VIU i retorna el municipi (municipi) i l'any de padró (anyPadro) de cada Habitatge juntament amb el nombre de parelles (nombre_parelles) existents per a cada combinació municipi-any. Això ens dóna una idea del nombre de parelles en funció del municipi i l'any de padró associat, útil per comprendre la relació entre individus i habitatges.

- **Quantes components connexes no estan connectades a cap node de tipus 'Habitatge'.**

```
CALL gds.wcc.stream('graf')
```

```
YIELD nodeId, componentId
```

```
WITH gds.util.asNode(nodeId) AS node, componentId
```

```
WHERE NOT (node)--(:Habitatge)
```

```
RETURN count(DISTINCT componentId) AS numComponentsDisconnected
```

"numComponentsDisconnected"
3532

L'execució d'aquesta consulta ens proporciona el nombre de components connexes al nostre graf que no estan connectades a cap node de tipus "Habitatge". Això ens permet entendre quantes components no tenen relació amb habitatges i poden ser considerades aïllades o desconnectades en termes d'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:

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.

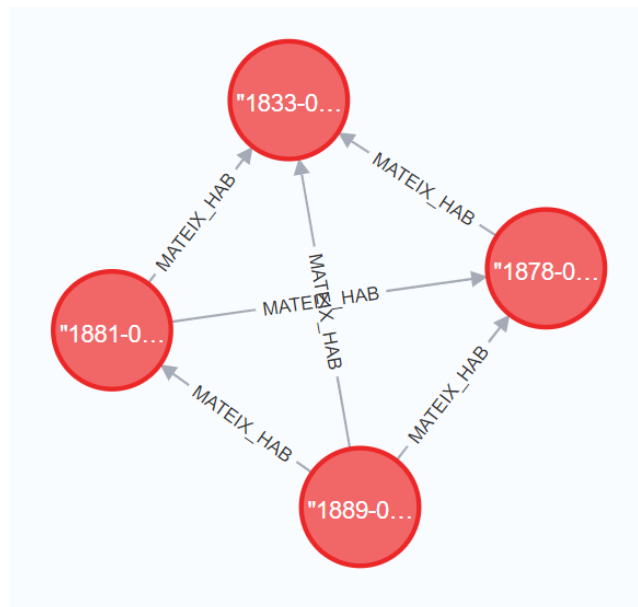
```
MATCH (h1:Habitatge), (h2:Habitatge)
```

```

WHERE h1.municipi = h2.municipi AND h1.idLlar = h2.idLlar AND h1.anyPadro <>
h2.anyPadro
WITH h1, h2, CASE WHEN h1.anyPadro > h2.anyPadro THEN h2 ELSE h1 END AS
nodeApuntat, CASE WHEN h1.anyPadro > h2.anyPadro THEN h1 ELSE h2 END AS
nodeApunta
MERGE (nodeApunta)-[r:MATEIX_HAB]->(nodeApuntat)
RETURN nodeApunta.municipi, nodeApunta.idLlar, nodeApunta.anyPadro,
nodeApuntat.municipi, nodeApuntat.idLlar, nodeApuntat.anyPadro

```

Exemple de la visualització de un subgraf de 4 nodes del graf generat en els quals l'habitatge és el mateix però amb any de padró diferent:



2. 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.project('graphEx3b', ['Individual', 'Habitatge'], ['VIU', 'FAMILIA',
'MATEIX_HAB'])

```

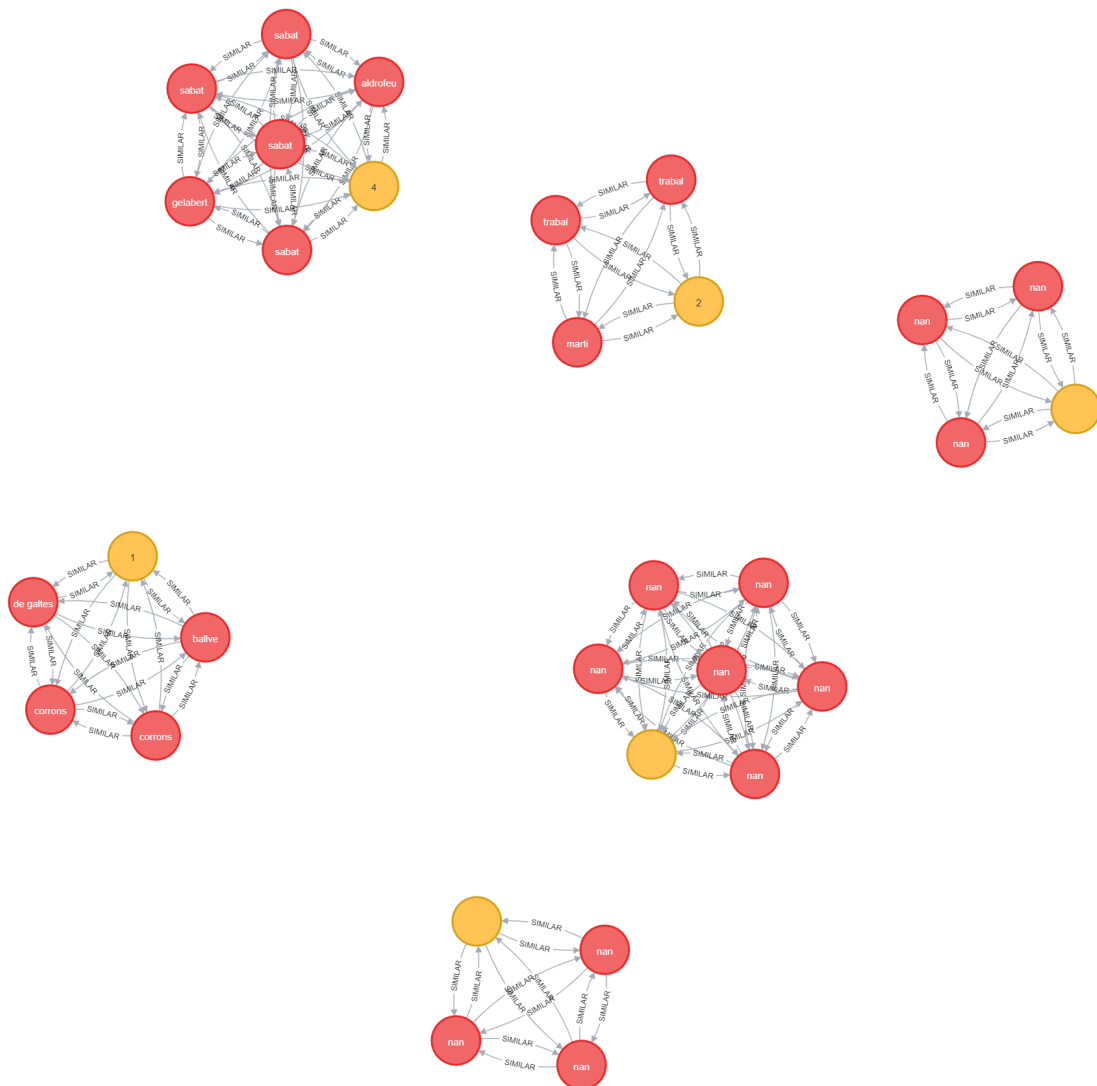
3. Calculeu la similaritat entre els nodes del graf que acabeu de crear, escriviu el resultat de nou a la base de dades i interpreteu els resultats obtinguts.

Per fer-ho, utilitzem `gds.nodeSimilarity`, que implementa la similaritat de Jaccard. De manera que es considera que dos nodes són semblants quan comparteixen molts veïns.

```
CALL      gds.nodeSimilarity.write('graphEx3b',      {writeRelationshipType:'SIMILAR',
writeProperty:'score'})
YIELD nodesCompared, relationshipsWritten
```

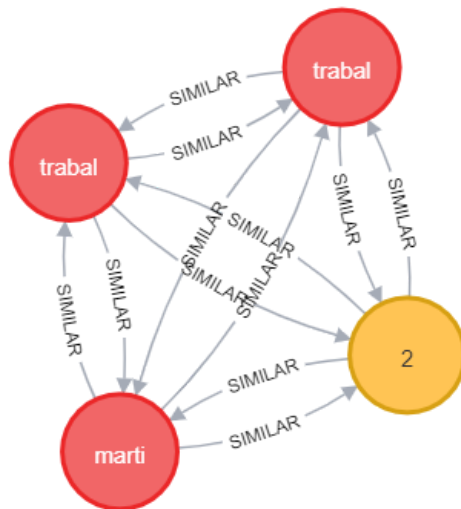
```
MATCH (n1)-[r:SIMILAR]->(n2)
RETURN n1, r.score, n2;
ORDER BY r.score DESC
LIMIT 25
```

El resultat que obtenim:



A partir dels resultats i tenint en compte la definició de *Jaccard Similarity*, podem deduir que els nodes més semblants són familiars i per tant, comparteixen llar i nodes amb que tenen una connexió familiar.

Si ens fixem amb un cas en particular, tenim aquest subgraf:



Els nodes individuals (de color vermell) corresponen a:

- francisca martí trabal
- enrica martí trabal
- antònia castells martí

Veiem que clarament són tenen un parentesc entre elles. Per aprofundir, fem la *query*:

MATCH (p:Individual)

WHERE p.cognom = "trabal" OR p.cognom2 = "trabal" OR p.cognom = "castells"

RETURN p

Fem un estudi i trobem que l'Antònia Castells és filla d'en Josep Castells Figueras. Aquest home té com esposa, l'Enrica Martí Trabal. I Francisca Martí Trabal és la cunyada de Jose. En aquest cas veiem que els nodes són família entre sí però en altres exemples amb un score de similitud igual d'alt (1) ens hem trobat que els nodes no són família encara que sí comparteixen habitatge. D'aquesta manera veiem que dos nodes no són similars basant-se només en una propietat sinó que és el conjunt d'elles que els fa més o menys semblants entre si.