Projecte de Neo4j: Padrons

Introducció: Padrons

Un grup d'investigadores en demografia històrica i processament de documents vol estudiar l'evolució socio-econòmica d'una població a partir de la informació dels padrons de poblacions. Els padrons són els llistats d'habitants que elabora un municipi on figura la informació la seva informació com noms, cognoms, edat i altres dades personals. En l'actualitat aquesta informació es manté actualitzada constantment gràcies als sistemes informàtics però antigament es recopilava manualment cada 3-5 anys aproximadament. Aquesta informació es registrava en llibres com el que es pot veure a la Figura 1.

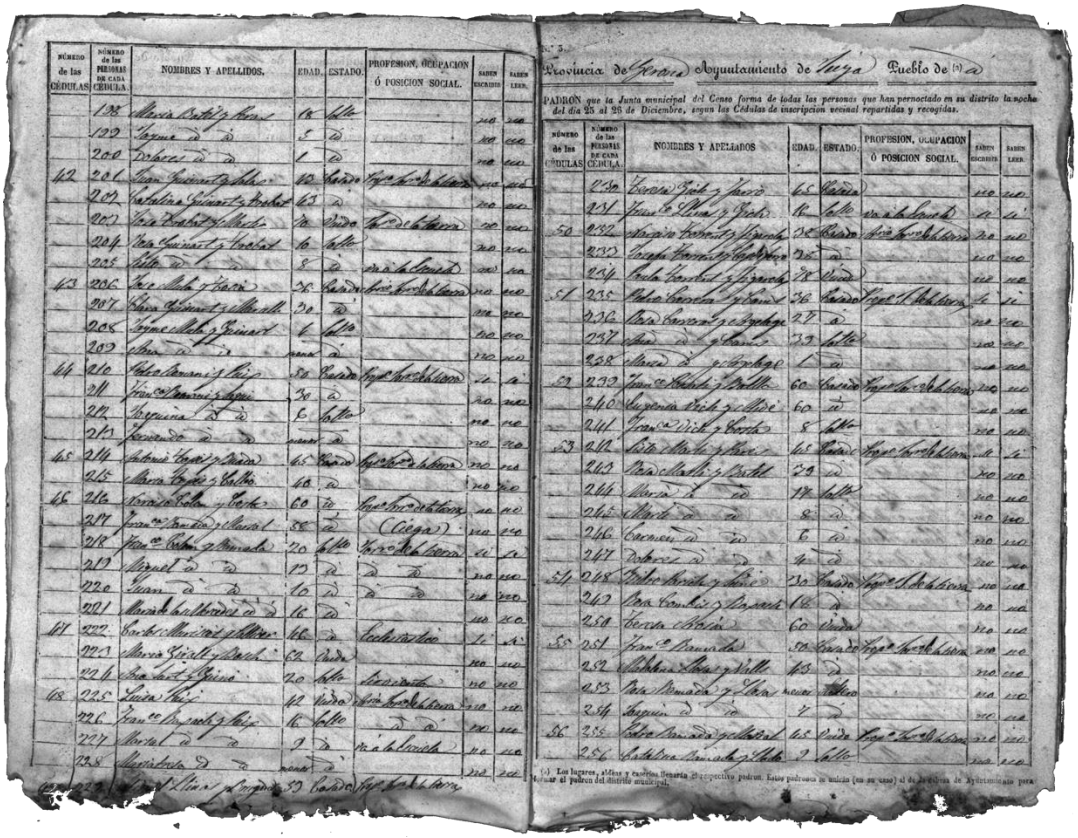

Figura 1. Padró del Cens de població que conté totes les persones que pernoctaren al municipi de Julià la nit del 25 de desembre de 1860.

La informació d'aquests documents, un cop digitalitzats, es processen amb tècniques de visió per computador i s'organitzen en una base de dades relacional. Per construir la base de dades, s'ha extret, de cada padró, la informació de cada habitatge, les persones que hi viuen i la relació de parentesc que hi ha entre ells. De cada habitatge, guarden la adreça completa (carrer, numero, pis), codi postal, barri i població. De cada persona que viu a l'habitatge, es guarda quina relació de parentesc el vincula amb el/la cap de família. A més, necessiten saber la ocupació (si treballa), una estimació dels ingressos (bruts) anuals de cadascun d'ells i l'estat civil. A més, com que els enregistraments poblacionals es repeteixen cada 3-5 anys la informació que es recopila per cada habitatge es va repetint per aquelles famílies que viuen en els mateixos habitatges.

Interessa identificar la informació dels individus al llarg del temps. Aquesta identificació es fa essencialment a partir de les dades personals (nom i cognoms) però en ocasions no és suficient. En aquests casos s'utilitza altres dades, com la data de naixement (inferida a partir de l'edat i l'any de padró), l'ofici o les relacions familiars. Tota aquesta informació s'ha organitzat segons el disseny Entitat-Relació de la Figura 2.

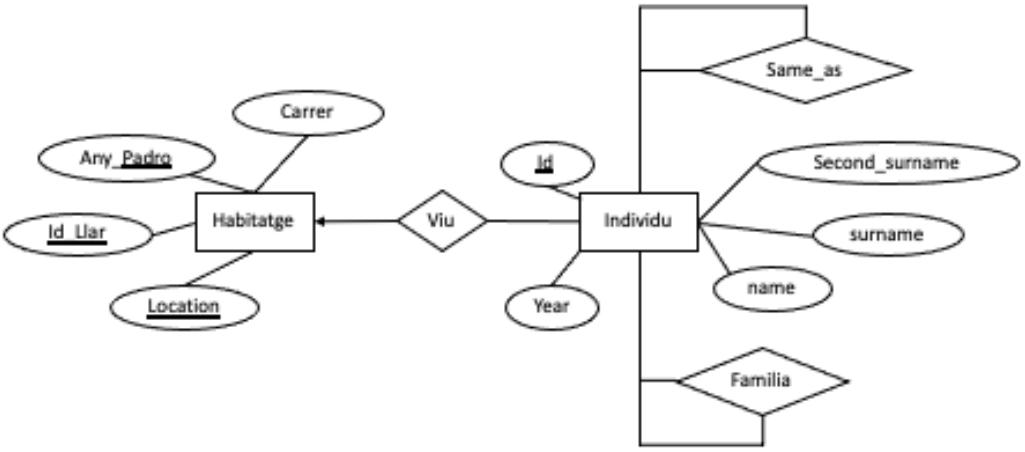

Figura 2. Disseny Entitat-Relació

Aquesta representació permet fer cerques i mostrar-les via una interfície web (http://dag.cvc.uab.es/xarxes/), però no permet analitzar fàcilment les relacions entre els individus al llarg del temps en els diferents habitatges en el que han viscut. Per això es vol representar la informació mitjançant una base de dades no relacional basada en grafs (veure Figura 3), que conté nodes de tipus *habitatges* i *individus*, i tres tipus de relacions:

- *viu*: representen el lloc on viu cada individu.
- *família:* relacions de parentesc entre individus que conviuen al mateix habitatge.
- *same_as*: els nodes que representen el mateix individu al llarg del temps.

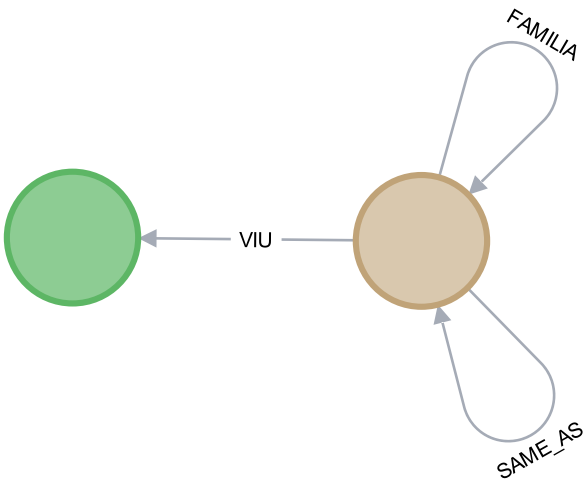

Figura 3. Esquema del graf de padrons

Projecte

En aquest projecte es treballarà, a partir d'un subconjunt de dades de padrons, la càrrega de dades i consultes en una base de dades de grafs (Neo4J) i es practicarà l'ús d'algorismes d'analítica de grafs.

Material

Disposeu del següent material per a realitzar el projecte:

- Fitxers de dades (CSVs).
- Script per visualitzar-les.

Treball previ

Abans d'iniciar el projecte caldrà llegir i visualitzar tota la documentació del projecte. A més cada grup haurà de crear un repositori privat de codi a **github** i donar accés al professorat que tutoritza el treball.

Exercicis

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 *constrains* i *indexos* quan sigui necessari.
- Assegureu-vos que en executar el script dues vegades no es dupliquin les dades (https://neo4j.com/docs/cypher-manual/current/constraints/).
- No carregueu files *null* del fitxer CSV (*Id* de municipi, llar o individu = *null*).
- Feu les conversions de tipus que siguin necessàries.

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.
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".
3. Dels padrons de Sant Feliu de Llobregat (SFLL) 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.
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 (SFLL). Retorna la informació en mode graf i mode llista.
5. 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 subgraf.

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).
7. Mostra totes les persones relacionades amb "benito julivert". Mostra la informació en forma de taula: el nom, cognom1, cognom2, i tipus de relació.
8. De la consulta anterior, mostra ara només els fills o filles de "benito julivert". Ordena els resultats alfabèticament per nom.
9. Llisteu totes les relacions familiars que hi ha.
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.
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.
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.
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.

Exercici 3. En aquest exercici analitzarem les dades del graf per entendre millor l'estructura de les dades. Els següents apartats tenen com objectiu orientar-vos respecte a com utilitzar algunes de les eines que ofereix Neo4J.

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:

   - Taula agrupant els resultats segons la mida de la cc.
   - Distribució de tipus de nodes (Individu o Habitatge) segons la mida de la cc.
   - Per cada municipi i any el nombre de parelles del tipus: (Individu)—(Habitatge)
   - quantes components connexes no estan connectades a cap node de tipus 'Habitatge'.

   Aquestes consultes són només una orientació del tipus d'exploració de dades que podeu fer. Tant si feu aquestes com d'altres, heu d'acompanyar-les d'una motivació (que preteneu saber amb la consulta) i d'una explicació dels resultats. Acompanyeu aquesta explicació amb el codi de la consulta i el resultat obtingut. (Indicació: utilitzeu la funció wcc en mode 'stream')

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.
2. Creeu un graf en memòria que inclogui els nodes Individu i Habitatge i les relacions VIU, FAMILIA, MATEIX_HAB que acabeu de crear.
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.

Material a lliurar i puntuació

Haureu de lliurar un informe on s'expliqui raonadament la resolució de cada exercici. Aquest informe haurà de ser auto contingut i contenir una secció: treball en equip, on s'expliqui la distribució de tasques entre els components de l'equip i qui s'ha responsabilitzat de cadascuna. Cada membre s'haurà de responsabilitzar d'almenys d'una tasca. La distribució de tasques haurà de ser consistent amb els commits al repositori del projecte. En l'informe s'haurà d'indicar l'adreça del repositori que haurà de contenir tot el material generat per l'informe. Això inclou tant els scripts en Cypher per importar les dades com els scripts per resoldre els altres exercicis.
La puntuació de cada exercici és la següent:

Exercici	**Puntuació**
Exercici 1	2
Exercici 2	4
Exercici 3	4

Factors multiplicatius de la nota.

A més de la puntuació de cada exercici hi haurà dos factors multiplicatius que s'aplicarà globalment a la nota final de cada exercici que pot afectar a la nota final del projecte. Els factors multiplicatius són:

- Treball en equip i ús del repositori de codi. Aquest factor s'aplicarà per avaluar el treball en equip i es podrà aplicar factors diferents als membres de l'equip. Normalment s'aplicarà un factor de 1 si el grup ha funcionat normalment. En cas que hagi evidències que algun membre de l'equip no faci la seva part se li podrà aplicar fins a un factor de 0 a nivell individual.
- Qualitat Informe. Aquest factor s'aplicarà per avaluar aspectes globals de la presentació de l'informe. Una presentació molt deficient de la feina feta podrà ser motiu suficient per suspendre el projecte.