**Exercici 1. Importeu 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/**](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.**

//IMPORTAR FINAL

//NETEJAR DATABASE

MATCH (n)

DETACH DELETE n;

En primer lloc, per a realitzar una importació correcta de les dades i assegurar-nos que no obtenim duplicats al executar l’script dues vegades, el que fem es netejar la base de dades. Per fer-ho, seleccionem tots els nodes de la base de dades (MATCH (n)) i els eliminem juntament amb totes les relacions existents associades als nodes (DETACH DELETE n)

//CONSTRAINTS

CREATE CONSTRAINT UNIQUE\_ID IF NOT EXISTS FOR (i:INDIVIDUAL) REQUIRE i.Id is UNIQUE;

CREATE CONSTRAINT ID\_INDIVIDUAL IF NOT EXISTS FOR (i:INDIVIDUAL) REQUIRE i.Id is NOT NULL;

CREATE CONSTRAINT ID\_HABITATGE IF NOT EXISTS FOR (h:HABITATGE) REQUIRE h.Id\_Habitatge is NOT NULL;

Pel que fa a les restriccions (CREATE CONSTRAINT), n’hem creat tres: la primera restricció per garantir que l’atribut ID dels nodes de tipus INDIVIDUAL sigui únic. Les altres dues restriccions creades fan referència al identificador (Id) dels nodes INDIVIDU (Id) i HABITATGE (Id\_Habitatge), aquestes restriccions ens asseguren que l’atribut ID no sigui nul tant per als nodes INDIVIDUAL com pels nodes HABITATGE.

//INDEX

CREATE INDEX AnyPadroIndex IF NOT EXISTS FOR ()-[r:VIU]-() ON (r.Year);

CREATE INDEX RelacioFamilia IF NOT EXISTS FOR ()-[r:FAMILIA]-() ON (r.Relacio\_Harmonitzada);

Per a poder realitzar consultes de forma més eficient creem índexs mitjançant CREATE INDEX per a l’atribut Any (YEAR) en les relacions del tipus VIU, i l’atribut Relacio\_Harmonitzada en les relacions del tipus FAMILIA.

//HABITATGES

LOAD CSV WITH HEADERS FROM 'file:///HABITATGES.csv' AS row

WITH row WHERE row.Municipi <> 'null' AND row.Id\_Llar <> 'null'

MERGE (h:HABITATGE {Municipi: row.Municipi, Id\_Habitatge: row.Id\_Llar, Any\_Padro: toInteger(row.Any\_Padro), Carrer: row.Carrer,

Numero: CASE WHEN row.Numero <> "null" THEN toInteger(row.Numero) ELSE row.Numero END});

Utilitzem la comanda LOAD CSV WITH HEADERS per carregar dades des de l’arxiu csv HABITATGES.csv i filtrem les files on el camp Municipi o el camp Id\_Llar sigui null. Utilitzem la clàusula MERGE per a la creació dels nodes de tipus HABITATGE amb els atributs Municipi, Id\_Habitatge, Any\_Padro, Carrer i Numero (int, sempre que hi hagi número del carrer)

//INDIVIDUAL

LOAD CSV WITH HEADERS FROM 'file:///INDIVIDUAL.csv' AS row

WITH row WHERE row.Id <> "null"

MERGE (i:INDIVIDUAL {Id: row.Id})

SET i.Year = toInteger(row.Year), i.name = row.name, i.surname = row.surname, i.second\_surname = row.second\_surname;

Utilitzem la comanda LOAD CSV WITH HEADERS per carregar dades des de l’arxiu csv INDIVIDUAL.csv i filtrem les files on el camp Id sigui nul. Utilitzem la clàusula MERGE per a la creació dels nodes de tipus INDIVIDUAL amb l’atribut Id. Usem la clàusula SET per definir els atributs Year (tipus enter), name, surname i second\_surname per al nom i cognoms de cada individu i l’any de naixement.

//VIU

LOAD CSV WITH HEADERS FROM 'file:///VIU.csv' AS row

WITH row WHERE row.IND <> 'null' AND row.HOUSE\_ID <> 'null'

// WITH row.IND AS IND, toInteger(row.HOUSE\_ID) AS HOUSE\_ID, row.Location AS Location, toInteger(row.Year) AS Year

MATCH (i:INDIVIDUAL), (h:HABITATGE)

WHERE row.IND = i.Id AND toInteger(row.HOUSE\_ID) = toInteger(h.Id\_Habitatge) AND toInteger(row.Year) = toInteger(h.Any\_Padro)

MERGE (i)-[:VIU {Location: row.Location, Year: row.Year}]->(h);

Utilitzem la comanda LOAD CSV WITH HEADERS per carregar dades des de l’arxiu csv VIU.csv i filtrem les files on els camps IND i HOUSE\_ID siguin nuls. Utilitzem la clàusula MATCH per trobar tots els nodes de tipus INDIVIDUAL i HABITATGE que compleixin amb els criteris especificats per a l’Id de l’individu, l’Id de l’habitatge i l’any de padró. Finalment utilitzem MERGE per a crear la relació VIU entre els nodes INDIVIDUAL i HABITATGE amb els atributs Location i Year.

//SAME\_AS

LOAD CSV WITH HEADERS FROM 'file:///SAME\_AS.csv' AS row

WITH row WHERE row.Id\_A <> "null" AND row.Id\_B <> "null"

MATCH (p:INDIVIDUAL {Id: row.Id\_A}), (s:INDIVIDUAL {Id: row.Id\_B})

MERGE (p)-[:SAME\_AS]->(s);

Fem servir la comanda LOAD CSV WITH HEADERS per carregar dades des de l’arxiu csv SAME\_AS.csv i filtrem les files on els camps Id\_A i Id\_B siguin nuls. Utilitzem la clàusula MATCH per trobar tots els nodes de tipus INDIVIDUAL i HABITATGE que compleixin amb els criteris especificats per a l’Id de cada l’individu (Id\_A i Id\_B). Amb la clàusula MERGE creem la relació SAME\_AS entre els nodes INDIVIDUAL i HABITATGE.

//FAMILIA

LOAD CSV WITH HEADERS FROM 'file:///FAMILIA.csv' AS row

WITH row WHERE row.ID\_1 <> "null" AND row.ID\_2 <> "null"

MATCH (p:INDIVIDUAL), (s:INDIVIDUAL)

WHERE row.ID\_1 = p.Id AND row.ID\_2 = s.Id

CREATE (p)-[:FAMILIA {Relacio: row.Relacio, Relacio\_Harmonitzada: row.Relacio\_Harmonitzada}]->(s);

Usem la comanda LOAD CSV WITH HEADERS per carregar dades des de l’arxiu csv FAMILIA.csv i filtrem les files on els camps Id\_1 i Id\_2 siguin nuls. Utilitzem la clàusula MATCH per trobar tots els nodes de tipus INDIVIDUAL que compleixin amb els criteris especificats per a l’Id de cada l’individu (Id\_1 i Id\_2). Amb la clàusula MERGE creem la relació FAMILIA entre nodes de tipus INDIVIDUAL amb els atributs Relacio i Relacio\_Harmonitzada.