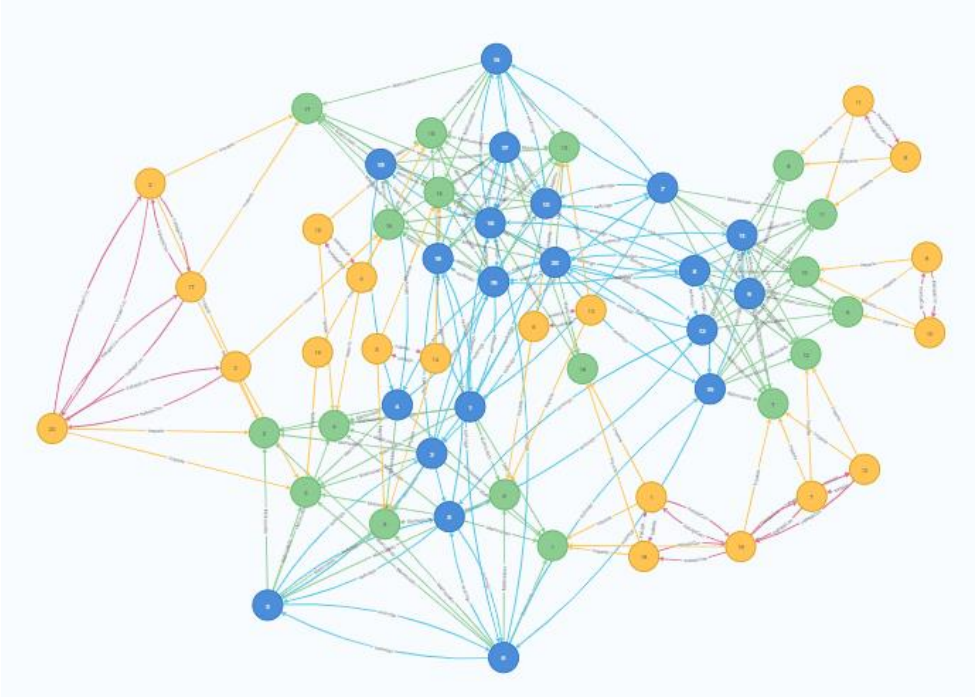
Practical work Neo4j



Máster en Bioinformática y Biología Computacional

Procesado y Manejo de Datos Masivos

Elena Aguayo Jara

Germán Vallejo Palma

18/11/2022

1. Querys for loading data from CSV:

- LOAD CSV WITH HEADERS FROM "file:/estudiantes.csv" AS csvLine CREATE (e:Estudiante { id: toInteger(csvLine.id), nombre: csvLine.nombre})
- LOAD CSV WITH HEADERS FROM " file:/profesores.csv" AS csvLine CREATE (p:Profesor { id: toInteger(csvLine.id), nombre: csvLine.nombre})
- LOAD CSV WITH HEADERS FROM "file:/esamigo.csv" AS csvLine MATCH(e1:Estudiante),(e2:Estudiante) where e1.id= toInteger(csvLine.id1) and e2.id= toInteger(csvLine.id2) create (e1)-[m:esAmigo]->(e2) return m
- LOAD CSV WITH HEADERS FROM "file:/asignaturas.csv" AS csvLine CREATE (e:Asignatura { id: toInteger(csvLine.id), nombre: csvLine.nombre, curso: csvLine.curso, semestre: csvLine.semestre, creditos: toInteger(csvLine.creditos)})
- LOAD CSV WITH HEADERS FROM " file:/imparte.csv" AS csvLine MATCH(p:Profesor),(a:Asignatura) where p.id= toInteger(csvLine.id1) and a.id= toInteger(csvLine.id2) create (p)-[m:Imparte]->(a) return m
- LOAD CSV WITH HEADERS FROM "file:/matriculada.csv" AS csvLine match(e:Estudiante),(a:Asignatura) where e.id= toInteger(csvLine.id1) and a.id= toInteger(csvLine.id2) create (e)-[m:Matriculado{ nota: toInteger(csvLine.nota) }]->(a) return m
- LOAD CSV WITH HEADERS FROM " file:/trabajaCon.csv" AS csvLine match(p1:Profesor),(p2:Profesor) where p1.id= toInteger(csvLine.id1) and p2.id= toInteger(csvLine.id2) create (p1)-[m:trabajaCon]->(p2) return m

2. Querys:

A. Names of subjects taught by Professor Jose Dorronsoro:
Match (p:Profesor)-[r:Imparte]->(a:Asignatura) Where p.nombre = "Jose_Dorronsoro" Return a

B. Names of 4th ESO subjects:
Match (a:Asignatura) Where a.curso = "4_ESO" Return a

C. Names of subjects arranged alphabetically:
Match (a:Asignatura) Return a Order by a.nombre

D. Names of 1º Bachillerato or 2º Bachillerato subjects:
Match (a:Asignatura) Where a.curso = "1_Bachillerato" or a.curso = "2_Bachillerato" Return a

E. **Eliminate the semester property of the nodes Subject:**
Match (a:Asignatura) Remove a.semestre Return a

F. **Change the name of one of the subjects (e.g French instead of English):**
Match (a:Asignatura) Where a.nombre="Ingles" set a.nombre = "Frances" Return a

G. **Number of subjects in which the student Germán is enrolled:**
Match (e:Estudiante)-[m:Matriculado]->(a:Asignatura) Where e.nombre = "German" return a

H. **Names of Elena's friends:**
Match (e:Estudiante)-[m:esAmigo]->(a:Estudiante) Where e.nombre = "Elena" return a

I. **Names of friends and friends of Elena's friends:**
Match (e:Estudiante)-[m:esAmigo*1..2]->(a:Estudiante) Where e.nombre = "Elena" return a

J. **Names of subjects in which one of the friends of the student Álvaro is enrolled:**
Match (e:Estudiante)-[m:esAmigo]->(a:Estudiante),(e)-[i:Matriculado]->(u:Asignatura) Where e.nombre="Alvaro" return u.nombre

K. **Names of students who are enrolled in any of the subjects taught by Luis del Peso:**
Match (p:Profesor)-[i:Imparte]->(a:Asignatura),(e:Estudiante)-[m:Matriculado]->(a) where p.nombre="Luis_Peso" return e.nombre, a.nombre

L. **Names of students who are enrolled in any of the subjects taught by any of the teachers who work with Karol Tezka:**
Match (p:Profesor) - [t:trabajaCon]->(r:Profesor), (r:Profesor) - [i:Imparte] -> (a:Asignatura), (e:Estudiante)- [m:Matriculado] -> (a) where p.nombre="Karol_Tezka" return e.nombre, a.nombre, r.nombre

3. <u>Proposed querys</u>:

A) This query returns a table with the name of the students of 4th ESO, their average grade and the number of friends of each one, sorted according to the average grade (from lowest to highest):

match (e:Estudiante)-[m:Matriculado]->(a:Asignatura), (e)-[am:esAmigo]->(es:Estudiante) where a.curso="4_ESO" return e.nombre as NOMBRE,count(DISTINCT es) as NUMERO_AMIGOS, AVG(m.nota) as NOTA_MEDIA ORDER BY NOTA_MEDIA

B) The following query returns a table with the names of the teachers sorted according to the number of different students they have (from the largest to the smallest number):

match (p:Profesor)-[i:Imparte]->(a:Asignatura),(e:Estudiante)-[m:Matriculado]->(a) RETURN p.nombre as NOMBRE ,(count(DISTINCT e)) as NUMERO_ALUMNOS_DISTINTOS ORDER BY NUMERO_ALUMNOS_DISTINTOS DESC

-