

Bases de datos NoSQL

Práctica 2: Neo4j

Daniel Pérez Efremova

2. Crea la base de datos en Neo4j (nodos y relaciones) utilizando el comando LOAD CSV.

Para construir la BD descrita en la práctica (ver imagen) se requieren dos pasos. En primer lugar definir los nodos y en segundo lugar definir las relaciones entre los nodos.

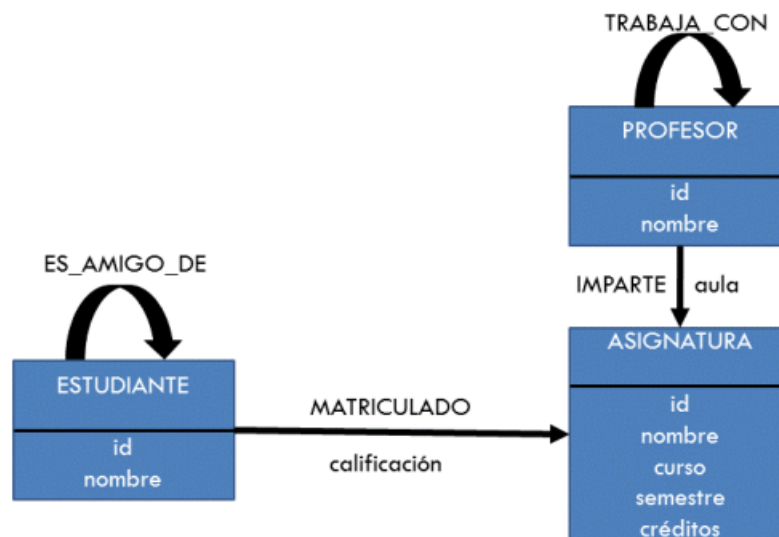


Figura 1: Esquema ERD de la BD a implementar

Se crean los nodos con las instrucciones del Código 1.

Código 1: Creación de los nodos de la BD

```
load csv with headers from 'file:///asignatura.csv' as asignatura
create (a:Asignatura {curso: asignatura.curso,
    creditos: asignatura.creditos,
    id: asignatura.id,
    semestre: asignatura.semestre,
    nombre: asignatura.nombre})

load csv with headers from 'file:///estudiante.csv' as estudiante
create (e:Estudiante {id:toInteger(estudiante.id),
    nombre: estudiante.nombre})

load csv with headers from 'file:///profesores.csv' as profesores
create (p:Profesores {id: toInteger(profesores.id),
    nombre: profesores.nombre})
```

Después se crean las relaciones con las instrucciones del Código 2. Las instrucciones mas importantes son:

- LOAD para cargar los ficheros que contienen nodos y relaciones
- CREATE para definir la estructura o schema de los objetos que se crean.

Código 2: Creación de las relaciones de la BD

```
load csv with headers from 'file:///imparte.csv' as file
match (a:Profesores {id: toInteger(file.id1)}), (b:Asignatura {id: file.id2})
create (a)-[:IMPARTE {aula: file.aula}]->(b)

load csv with headers from 'file:///matriculado.csv' as file
match (a:Estudiante {id: toInteger(file.id1)}), (b:Asignatura {id: file.id2})
create (a)-[:MATRICULADO {calificacion: file.calificacion}]->(b)

load csv with headers from 'file:///esAmigo.csv' as file
match (a:Estudiante {id: toInteger(file.id1)}), (b:Estudiante {id: toInteger(file.id2)})
create (a)-[:ES_AMIGO_DE]-(b)

load csv with headers from "file:///trabajaCon.csv" AS file
match (c:Profesores {id: toInteger(file.id1)}), (d:Profesores {id: toInteger(file.id2)})
create (c)-[:TRABAJA_CON]->(d)
```

A continuación se añaden imágenes dentro del servidor local Neo4j con la BD creada y el resultado de las instrucciones anteriores.

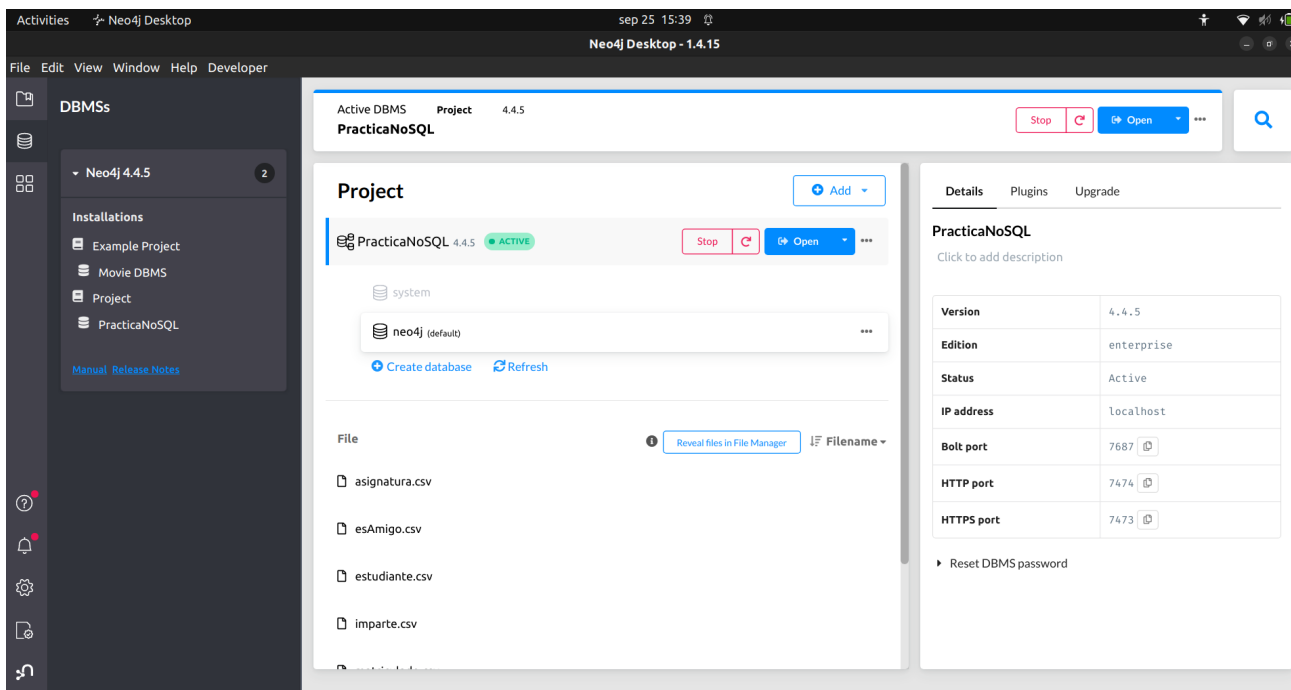


Figura 2: BD creada en la aplicación

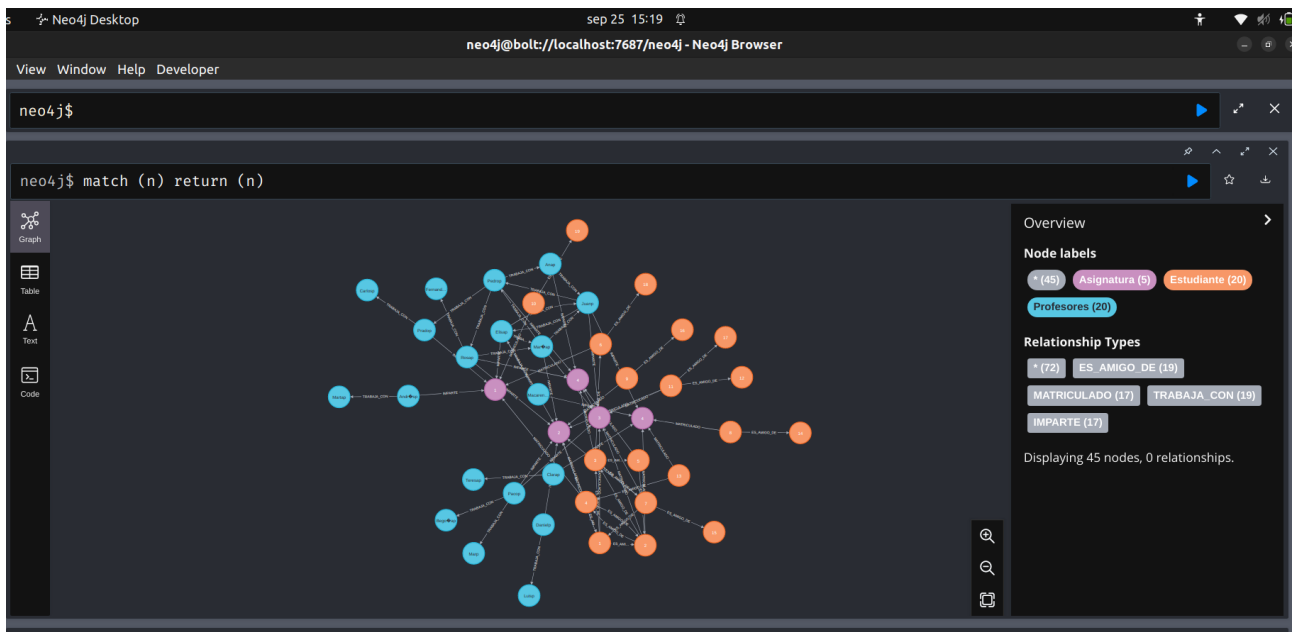


Figura 3: BD implementada siguiendo las instrucciones propuestas

3. Ejecuta las siguientes consultas en Neo4j.

En los siguientes apartados se explica la idea detrás de la consulta propuesta y una imagen que prueba su ejecución.

1. Nombres de asignaturas impartidas por la profesora Clarap en el aula 5

En la base de datos, no figura que Clarap imparta clases en el aula 5. Se plantea la consulta para el aula 6, que sí figura en la BD. Para encontrar la/s asignatura/s, se filtran los profesores por Clarap talque en la relación IMPARTE tenga valor de aula = 6.

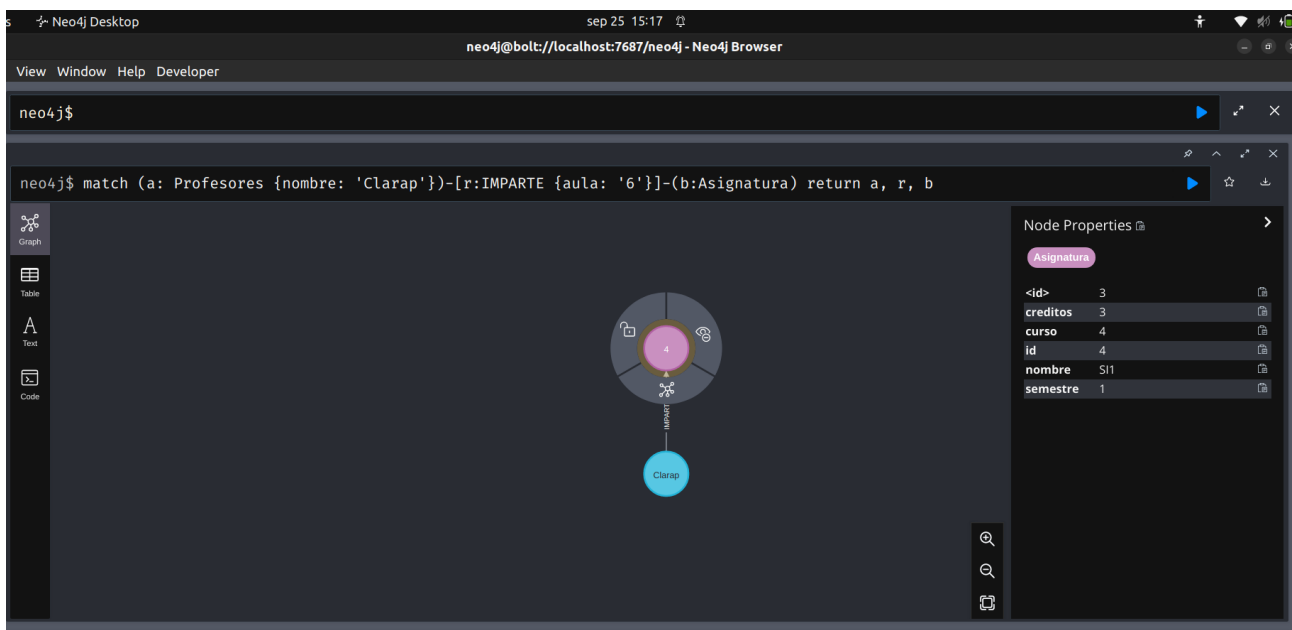


Figura 4: Asignaturas que imparte Clarap en el aula 6

Se observa que solo imparte una asignatura: SI1.

2. Calificación obtenida por el estudiante Pérez en la asignatura de Física

No existe un estudiante Pérez ni la asignatura Física en la BD. Se hace la consulta para Pedro y la asignatura Prog2. Se filtran los estudiantes por nombre igual a Pedro y que, con la relacion MATRICULADO, esté conectado a la asignatura Prog2. Se consulta la propiedad calificacion de la relación.

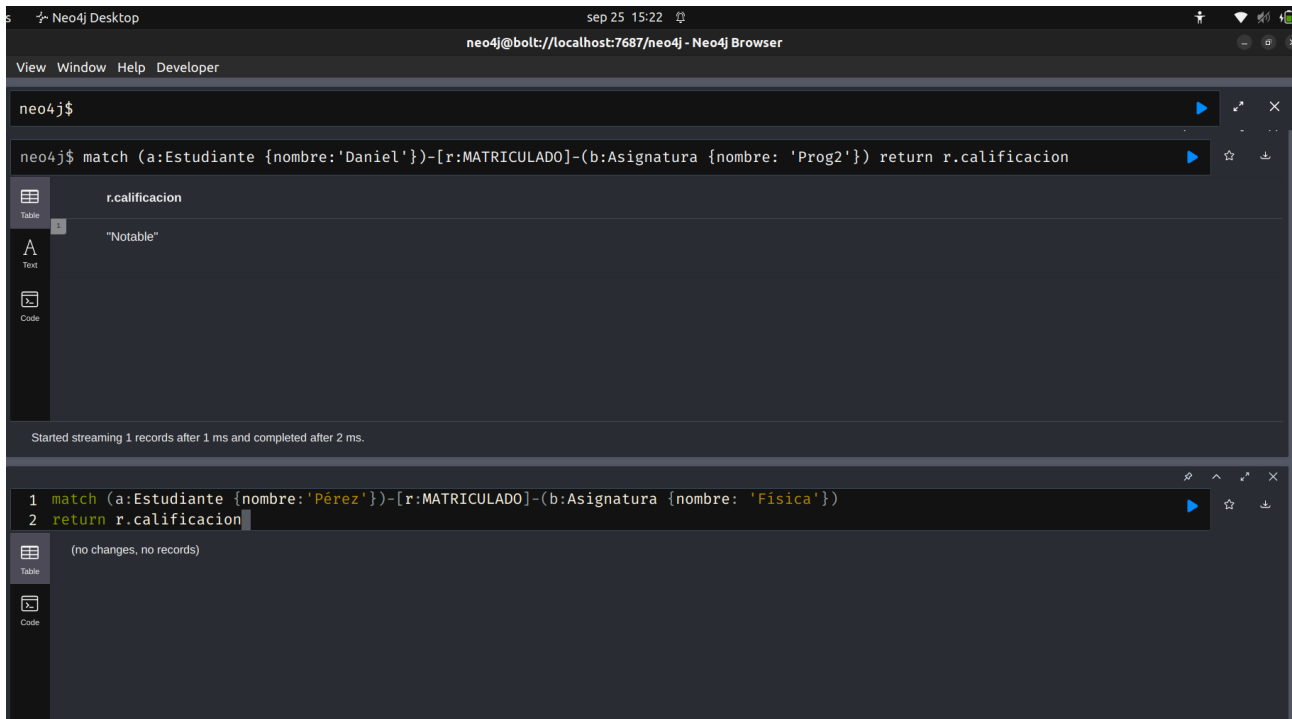


Figura 5: Calificacion del estudiante Pedro en la asignatura Prog2

3. Número de créditos de los que está matriculado la estudiante IsabelJ

IsabelJ no existe en la base de datos. Se hace la misma consulta para el estudiante Daniel. Se filtran los estudiantes por nombre igual a Daniel, despues, mediante la relación MATRICULADO, se obtienen los nodos Asignatura. Para terminar se agrega la suma de la propiedad creditos.

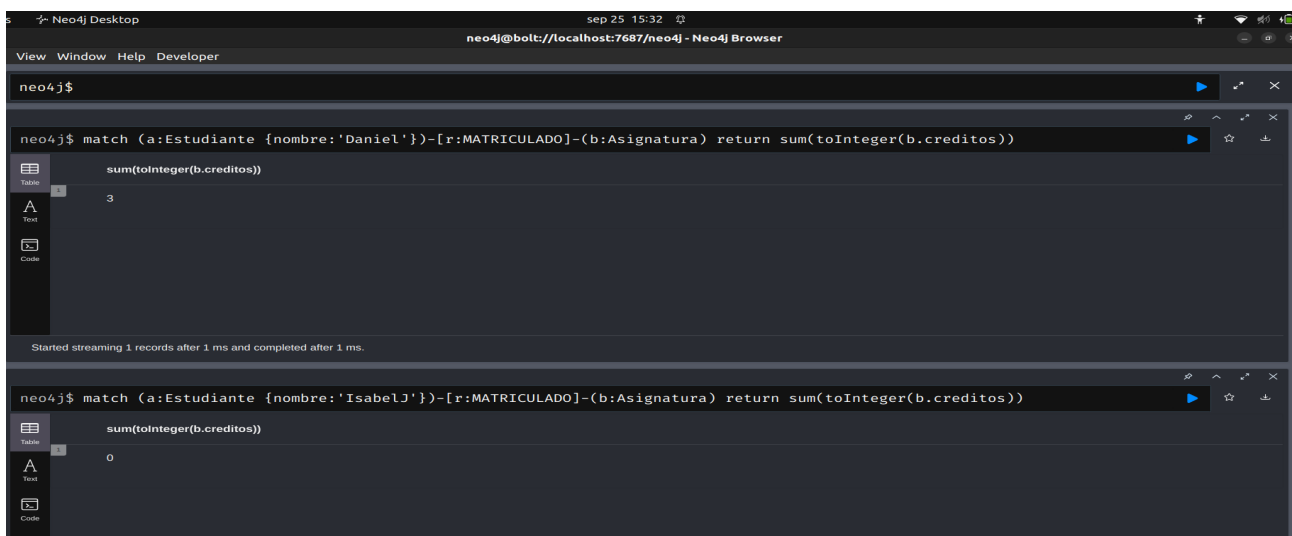


Figura 6: Numero de creditos matriculado por el estudiante Daniel

4. Nombres de asignaturas de primer curso

Se seleccionan todos los nodos asignatura filtrando por los que tengan valor de propiedad curso igual a 1.

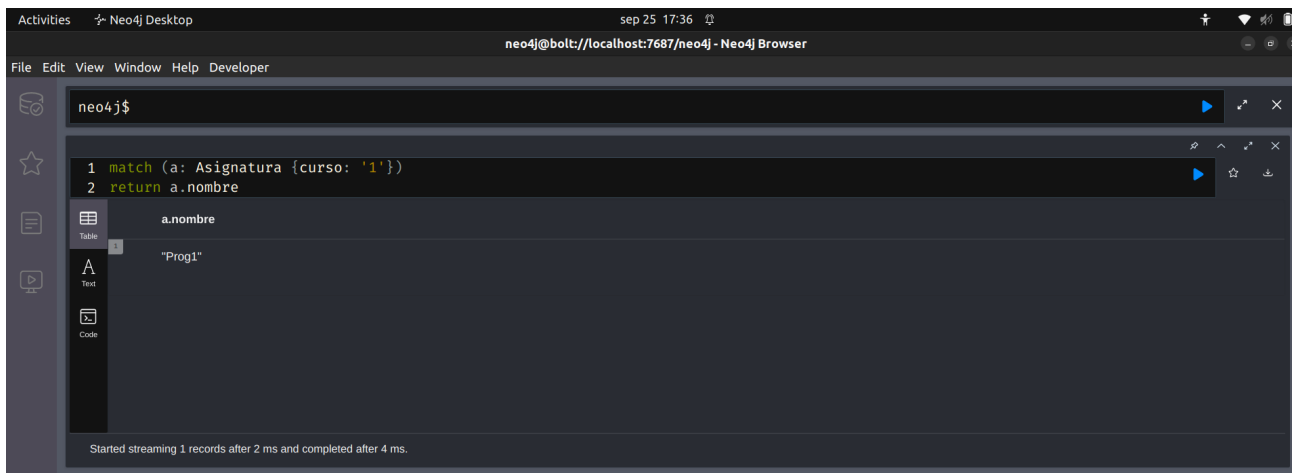


Figura 7: Asignaturas de primer curso

5. Nombres de asignaturas ordenados alfabéticamente

Se seleccionan los nodos Asignatura y se devuelve la propiedad nombre ordenando los valores.

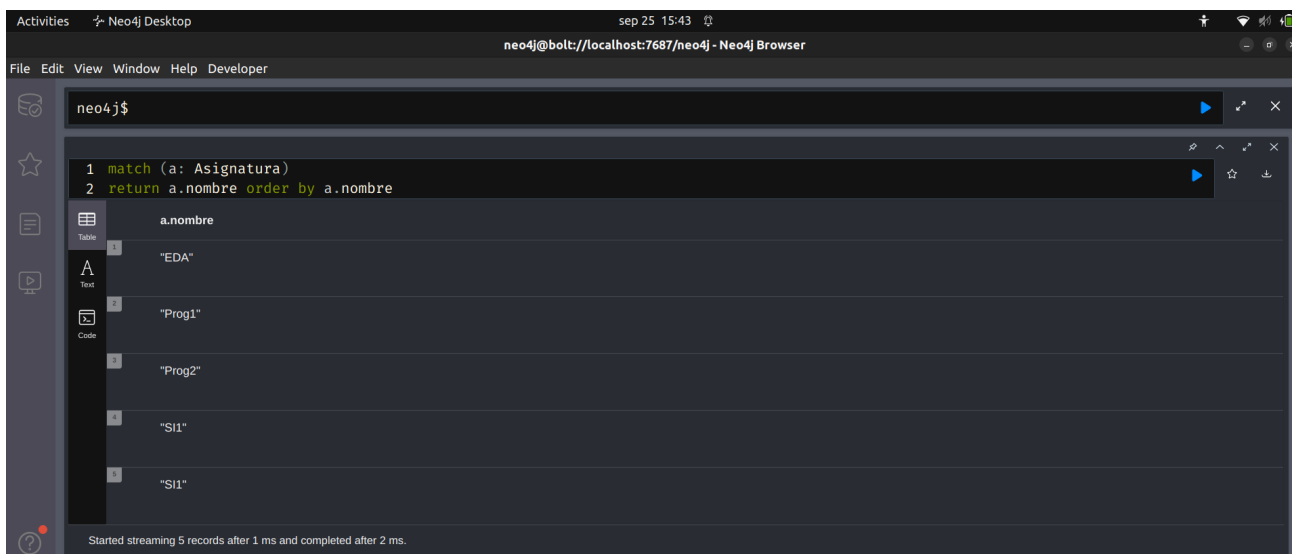


Figura 8: Nombre de asignaturas ordenadas alfabéticamente

6. Nombres de asignaturas de tercer o cuarto curso

Se seleccionan los nodos Asignatura, se filtran por valor de propiedad curso igual a 3 o 4 y se devuelve la propiedad nombre ordenando los valores.

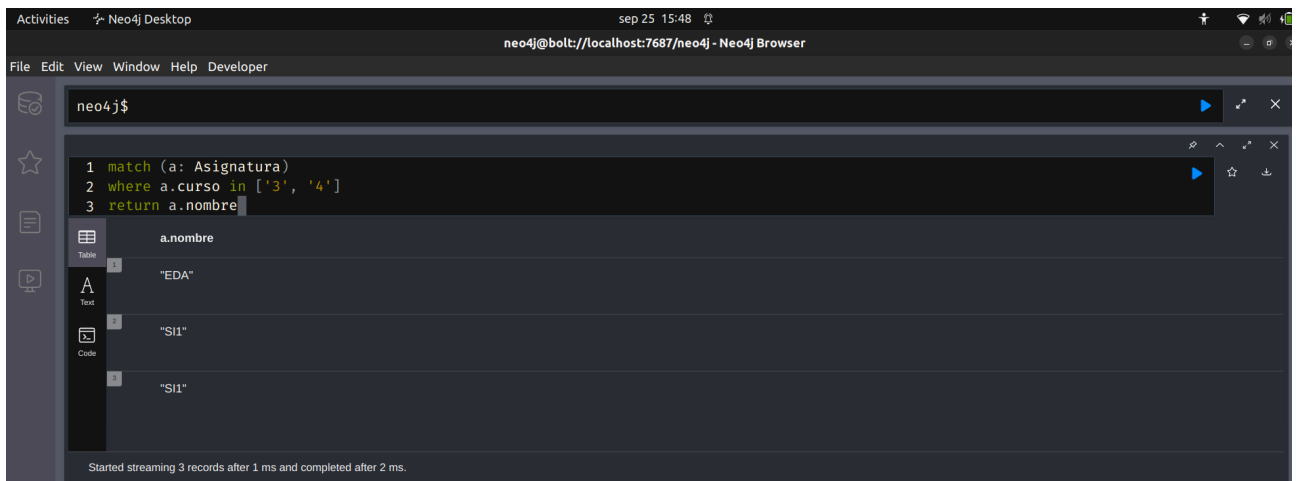


Figura 9: Nombres de asignatura de tercer y cuarto año

7. Eliminar la propiedad semestre de una asignatura

Se seleccionan los nodos Asignatura y se utiliza la instrucción delete sobre la propiedad semestre.

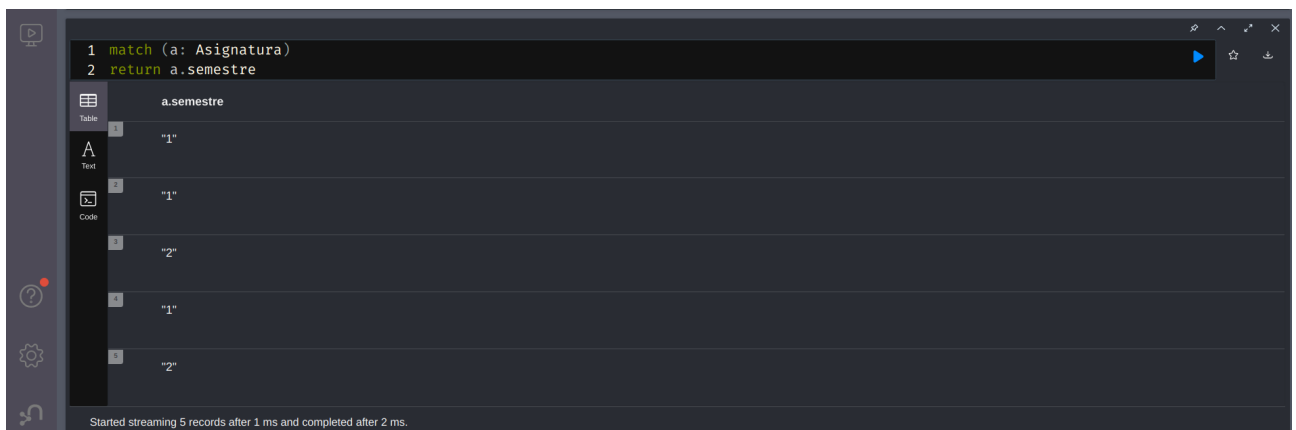


Figura 10: Comprobación de los valores de la propiedad semestre

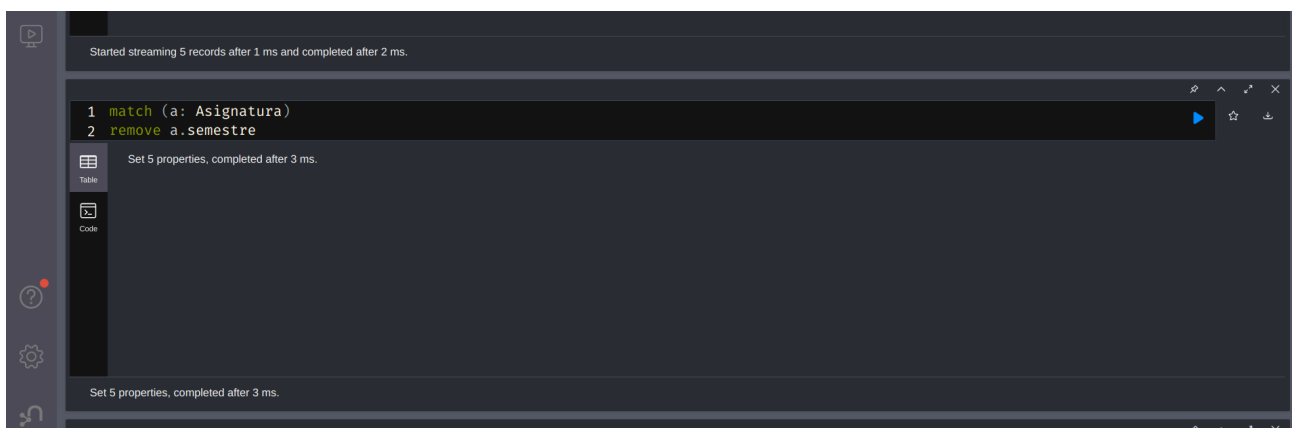


Figura 11: Instrucción delete para eliminar la propiedad semestre

Tras ejecutar la instrucción delete se observa que el resultado de la consulta es null.

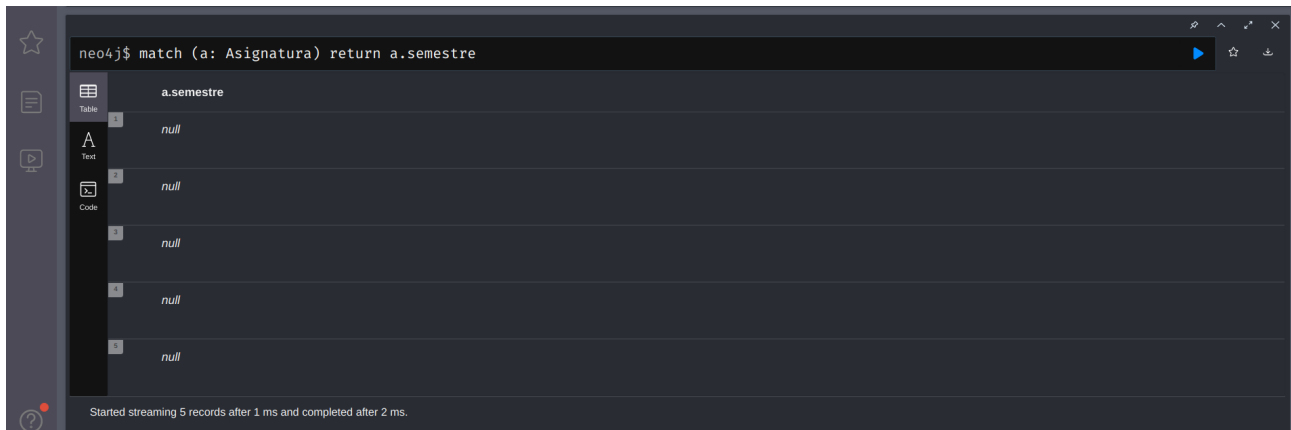


Figura 12: COmprobación del resultado al eliminar la propiedad semestre de Asignatura

8. Cambiar el nombre de una de las asignaturas

Se utiliza la instrucción SET para cambiar el valor de una propiedad. En este caso se busca la Asignatura Prog2 y se cambia por Calculo. Se comprueba que el cambio se haya producido.

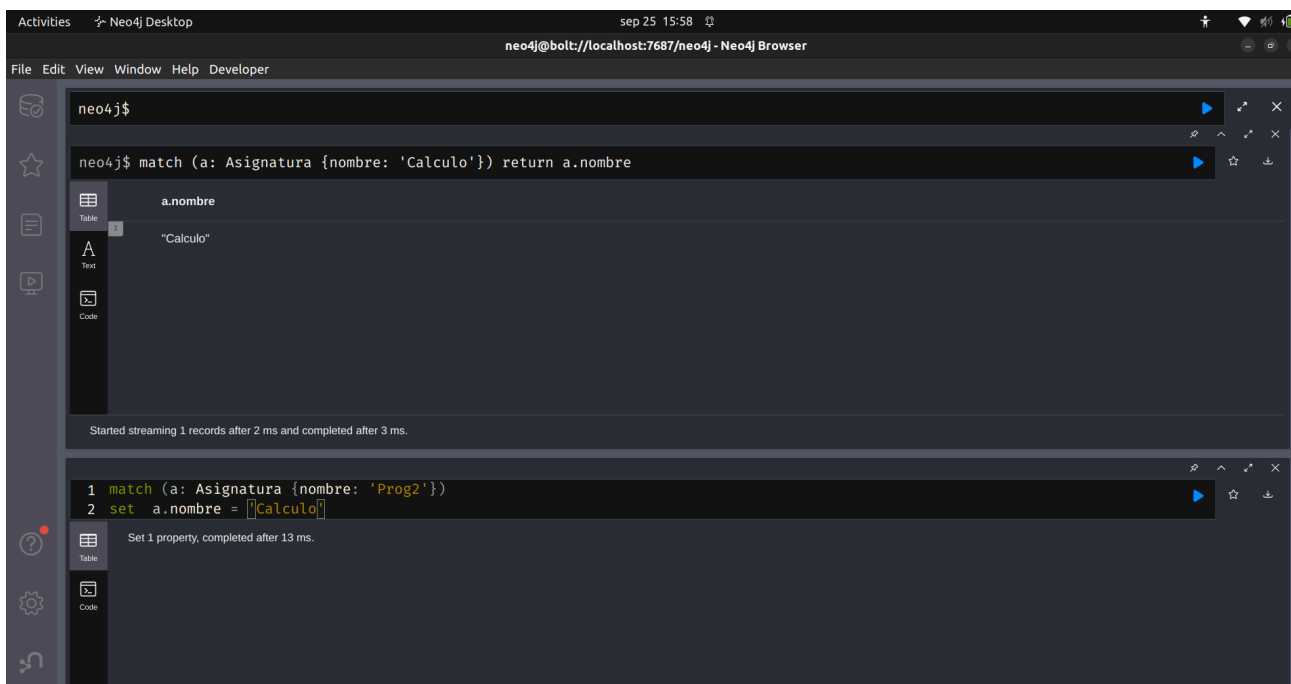


Figura 13: Cambio del nombre de una asignatura

9. Número de asignaturas en las que está matriculado la estudiante Elisa

Se seleccionan los nodos Estudiante que por la relación MATRICULADO esten conectados a los nodos Asignatura. Finalmente se agrega el resultado mediante Count para obtener el resultado.



Figura 14: Recuento de asignaturas matriculadas por Elisa

10. Nombres de los amigos del estudiante Pedro

Se seleccionan los nodos Estudiante filtrando por la propiedad nombre igual pedro, y mediante la relación AMIGO_DE, se buscan los nodos Estudiante conectados con Pedro. Se devuelve únicamente la propiedad nombre de Estudiante.



Figura 15: Amigos de Pedro

11. Nombres de los amigos y de los amigos de los amigos del estudiante Pedro

Se seleccionan los nodos Estudiante filtrando por la propiedad nombre igual pedro, y mediante la relación AMIGO_DE, se buscan los nodos Estudiante conectados con Pedro. Se repite el proceso para cada nodo resultado del primer nivel de relación. Se muestra solo el valor de la propiedad nombre.



Figura 16: Amigos de amigos de Pedro

12. Nombres de asignaturas en las que está matriculado alguno de los amigos de la estudiante Elisa

Se seleccionan los nodos Estudiante filtrando por nombre igual a Elisa. Después se buscan los nodos Estudiante conectados con Elisa mediante la relación AMIGO_DE. Después, se buscan las asignaturas conectadas mediante la relación MATRICULADO con alguno de los nodos Estudiante que son amigos de Elisa. Se devuelve solo el valor de la propiedad nombre de las asignaturas.



Figura 17: Nombre de las asignaturas matriculados por amigos de Elisa

13. Nombres de estudiantes que están matriculados en alguna de las asignaturas que imparte la profesora Elisap

Se seleccionan los nodos estudiante que por la relación MATRICULADO están conectados a alguna asignaturas. Por otro lado, Se seleccionan las asignaturas que están conectadas por la relación IMPARTE a los nodos Profesores filtrados por la propiedad nombre igual a Elisap.

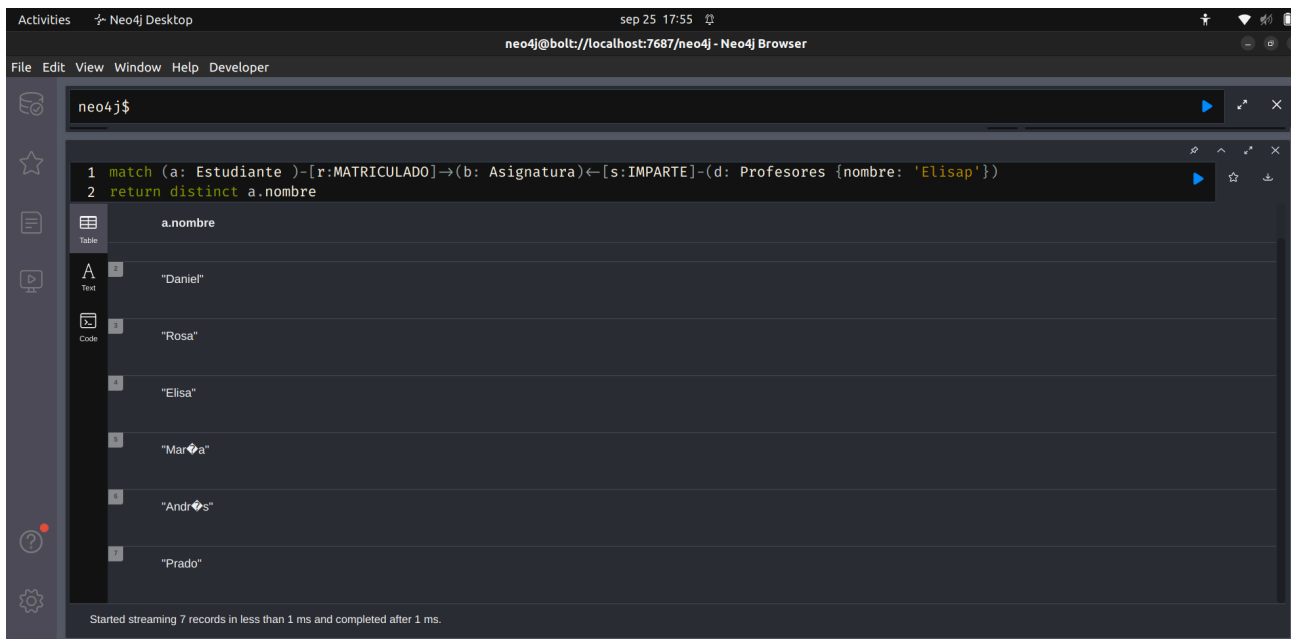


Figura 18: Estudiantes matriculados en asignaturas impartidas por Elisap

14. Nombres de estudiantes que están matriculados en alguna de las asignaturas que imparte alguno de los profesores que trabaja con la profesora Elisap.

A la consulta anterior, se añade que las asignaturas se filtren por las impartidas por nodos de profesores que, mediante la relación TRABAJA_CON estén conectados al nodo profesores filtrado por Elisap en la propiedad nombre.

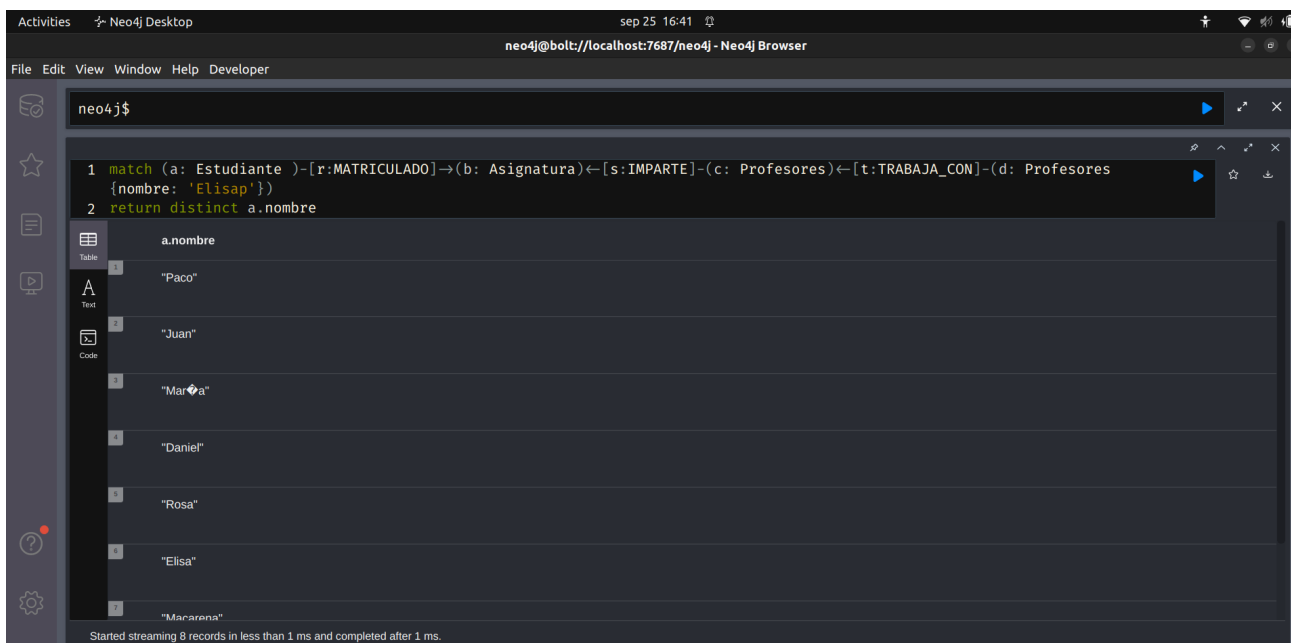


Figura 19: Estudiantes matriculados en asignaturas impartidas por profesores que trabajan con elisap

14. Propón otras dos consultas de pattern matching y resuélvelas con Neo4j.

1. Se propone buscar el total de créditos matriculados por asignatura. Se buscan todos los estudiantes que mediante la relación MATRICULADO estén conectados a alguna asignatura. Después se agrega el número de créditos de los alumnos mediante suma.

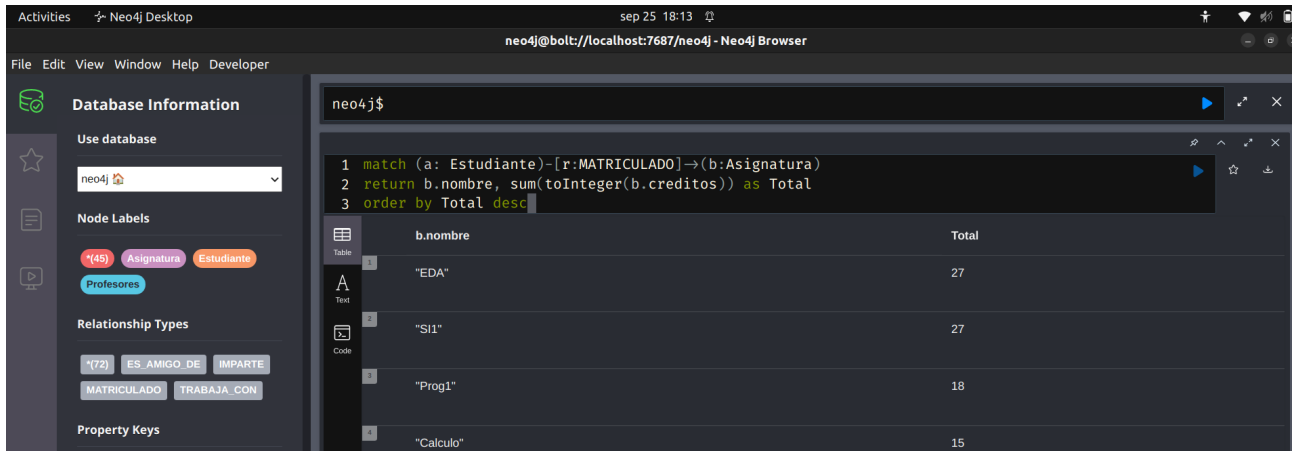


Figura 20: Total de créditos matriculados por asignatura

Se ve que las asignaturas con más créditos matriculados son EDA y SI1.

2. Se propone buscar a los amigos de Pedro que están matriculados en Calculo. Se buscan los nodos Estudiante filtrando por la propiedad nombre igual a Pedro que estén conectados mediante la relación ES_AMIGO_DE a otro nodo Estudiante que esté conectado mediante la relación MATRICULADO al nodo asignatura con valor en la propiedad nombre igual a calculo.



Se ve que la única amiga es Rosa.