Cypher and SQL, HW3

Farnoosh Koleini

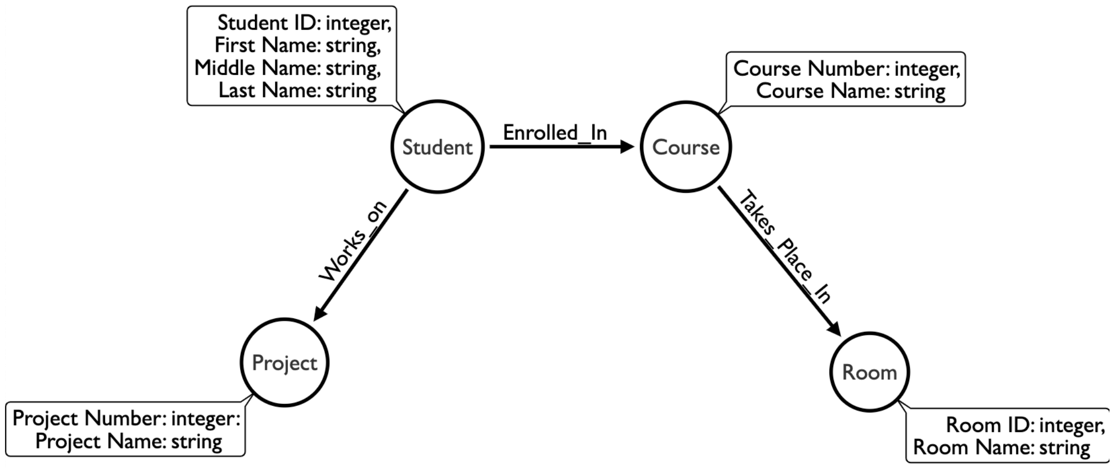
2023-02-19

1.Neo4j schema design and data load

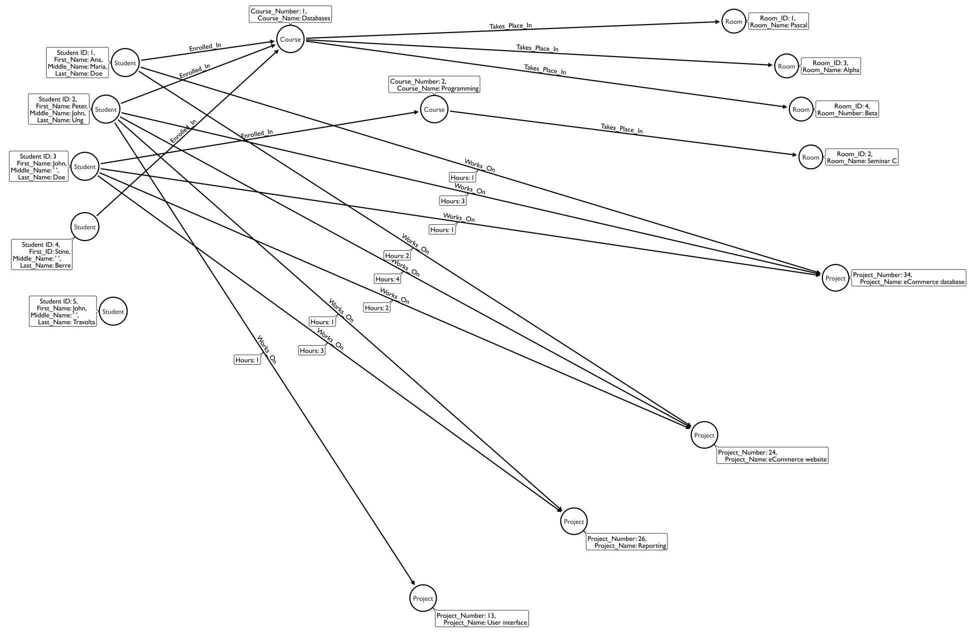
1.1 Schema design (20 pts)

Draw the database schema for the graph database, using http://www.apcjones.com/arrows (http://www.apcjones.com/arrows). Include a screenshot of it.

To create the schema for this dataset, first of all we can create a really general schema includes all main nodes and their relationship, then we can create the whole schema with all the details,nodes, and relationships. I uploaded the pdf file for the general schema called arrow schema (detail) in the zip file. Also, I uploaded the screenshot of the general schema, please find them.


Student ID: integer, First Name: string, Middle Name: string, Last Name: string
Student
Enrolled_In
Course
Course Number: integer, Course Name: string
Works_on
Takes_Place_In
Project
Project Number: integer: Project Name: string
Room
Room ID: integer, Room Name: string

2/19/23, 3:17 PM arrows-3.svg


Course_Number: 1, Course_Name: Databases
Course
Student ID: 1, First_Name: Ana, Middle_Name: Maria, Last_Name: Doe
Student
Student ID: 2, First_Name: Peter, Middle_Name: John, Last_Name: Ung
Student
Student ID: 3 First_Name: John, Middle_Name: ' ', Last_Name: Doe
Student
Student ID: 4, First_ID: Stine, Middle_Name: ' ', Last_Name: Berre
Student
Student ID: 5, First_Name: John, Middle_Name: ' ', Last_Name: Travolta
Student
Course_Number: 2, Course_Name: Programming
Course
Enrolled_In
Enrolled_In
Enrolled_In
Enrolled_In
Takes_Place_In
Takes_Place_In
Takes_Place_In
Takes_Place_In
Room
Room_ID: 1, Room_Name: Pascal
Room
Room_ID: 3, Room_Name: Alpha
Room
Room_ID: 4, Room_Number: Beta
Room
Room_ID: 2, Room_Name: Seminar C
Works_On
Hours: 1
Works_On
Hours: 3
Works_On
Hours: 1
Works_On
Hours: 2
Works_On
Hours: 4
Works_On
Hours: 2
Works_On
Hours: 1
Works_On
Hours: 3
Works_On
Hours: 1
Project
Project_Number: 34, Project_Name: eCommerce database
Project
Project_Number: 24, Project_Name: eCommerce website
Project
Project_Number: 26, Project_Name: Reporting
Project
Project_Number: 13, Project_Name: User interface

file:///Users/koleinif20/Downloads/arrows-3.svg 1/1

1.2 Data load (30 pts)

CREAT all the nodes

CREATE (n1:Student {Student_ID:1,First_Name:‘Ana’,Middle_Name:‘Maria’,Last_Name:‘Doe’}) CREATE (n2:Student{Student_ID:2,First_Name:‘Peter’,Middle_Name:‘John’,Last_Name:‘Ung’}) CREATE (n3:Student {Student_ID:3,First_Name:‘John’,Middle_Name:’‘,Last_Name:’Doe’}) CREATE (n4:Student {Student_ID:4,First_Name:‘Stine’,Middle_Name:’‘,Last_Name:’Berre’}) CREATE (n5:Student

{Student_ID:5,First_Name:‘John’,Middle_Name:’‘,Last_Name:’Travolta’}) CREATE (n6:Course {Course_Number:1,Course_Name:‘Databases’}) CREATE (n7:Course {Course_Number:2,Course_Name: ‘Programming’}) CREATE (n8:Project {Project_Number:34,Project_Name:‘eCommerce database’}) CREATE (n9:Project {Project_Number:24,Project_Name:‘eCommerce website’}) CREATE (n10:Project {Project_Number:26,Project_Name:‘Reporting’}) CREATE (n11:Project {Project_Number:13,Project_Name: ‘User interface’}) CREATE (n12:Room {Room_ID:1,Room_Name:‘Pascal’}) CREATE (n13:Room {Room_ID:3,Room_Name:‘Alpha’}) CREATE (n14:Room {Room_ID:4,Room_Name:‘Beta’}) CREATE (n15:Room {Room_ID:2,Room_Name:‘Seminar C’})

CREATE all the relationships

MATCH (a:Student{Student_ID: 1}), (b:Course {Course_Number: 1}) CREATE (a)-[:Enrolled_IN]->(b)

MATCH (c:Student{Student_ID: 2}), (d:Course {Course_Number: 1}) CREATE (c)-[:Enrolled_IN]->(d)

MATCH (e:Student{Student_ID: 4}), (f:Course {Course_Number: 1}) CREATE (e)-[:Enrolled_IN]->(f)

MATCH (f:Student{Student_ID: 3}), (g:Course {Course_Number: 2}) CREATE (f)-[:Enrolled_IN]->(g)

MATCH (h:Student{Student_ID: 1}), (i:Project{Project_Number: 34}) CREATE (h)-[: `Works_On` {Hours:‘1’}]->(i)

MATCH (j:Student{Student_ID: 2}), (k:Project{Project_Number: 34}) CREATE (j)-[: `Works_On` {Hours:‘3’}]->(k)

MATCH (l:Student{Student_ID: 3}), (m:Project{Project_Number: 34}) CREATE (l)-[: `Works_On` {Hours:‘1’}]->(m)

MATCH (n:Student{Student_ID: 1}), (o:Project{Project_Number: 24}) CREATE (n)-[: `Works_On` {Hours:‘2’}]->(o)

MATCH (p:Student{Student_ID: 2}), (q:Project{Project_Number: 24}) CREATE (p)-[: `Works_On` {Hours:‘4’}]->(q)

MATCH (r:Student{Student_ID: 3}), (t:Project{Project_Number: 24}) CREATE (r)-[: `Works_On` {Hours:‘2’}]->(t)

MATCH (s:Student{Student_ID: 2}), (w:Project{Project_Number: 26}) CREATE (s)-[: `Works_On` {Hours:‘1’}]->(w)

MATCH (u1:Student{Student_ID: 3}), (w1:Project{Project_Number: 26}) CREATE (u1)-[: `Works_On` {Hours:‘3’}]->(w1)

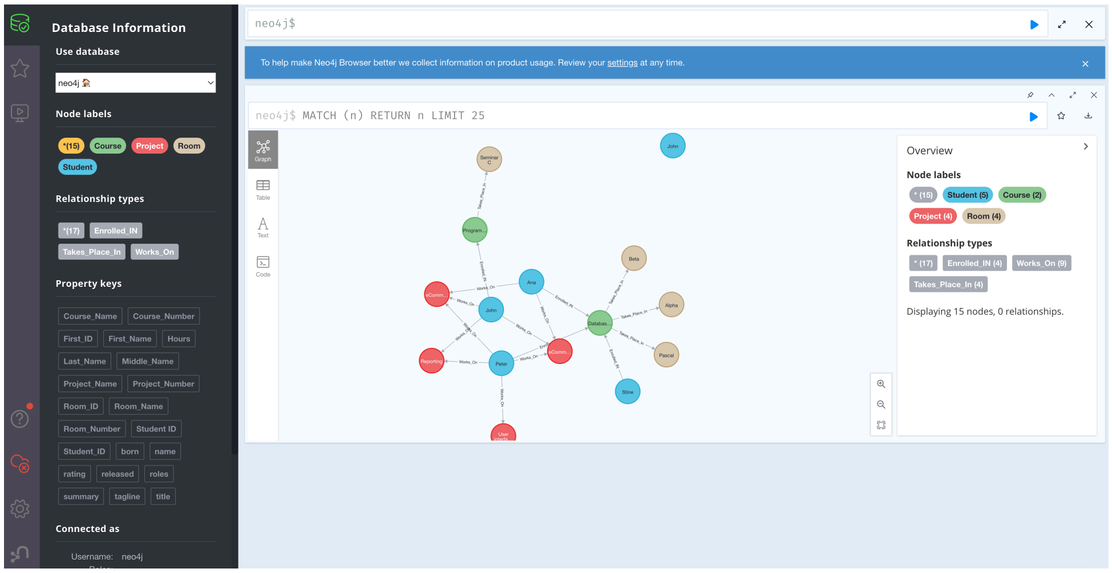
MATCH (v:Student{Student_ID: 2}), (x:Project{Project_Number: 13}) CREATE (v)-[: `Works_On` {Hours:‘3’}]->(x)

MATCH (y:Course{Course_Number: 1}), (z:Room{Room_ID: 1}) CREATE (y)-[: `Takes_Place_In`]->(z)

MATCH (y1:Course{Course_Number: 1}), (z1:Room{Room_ID: 3}) CREATE (y1)-[: `Takes_Place_In`]->(z1)

MATCH (y2:Course{Course_Number: 1}), (z2:Room{Room_ID: 4}) CREATE (y2)-[: `Takes_Place_In`]->(z2)

MATCH (y3:Course{Course_Number: 2}), (z3:Room{Room_ID: 2}) CREATE (y3)-[: `Takes_Place_In`]->(z3)


Database Information
Use database
neo4j
Node labels
*(15) Course Project Room Student
Relationship types
*(17) Enrolled_IN Takes_Place_In Works_On
Property keys
Course_Name Course_Number First_ID First_Name Hours Last_Name Middle_Name Project_Name Project_Number Room_ID Room_Name Room_Number Student ID Student_ID born name rating released roles summary tagline title
Connected as
Username: neo4j
neo4j$ MATCH (n) RETURN n LIMIT 25
To help make Neo4j Browser better we collect information on product usage. Review your settings at any time.
Overview
Node labels
* (15) Student (5) Course (2) Project (4) Room (4)
Relationship types
* (17) Enrolled_IN (4) Works_On (9) Takes_Place_In (4)
Displaying 15 nodes, 0 relationships.

Data analysis (60 pts: 5 pts each per SQL query, 5 pts each per Cypher query)

Write the SQL queries and the Cypher queries to answer the following questions.

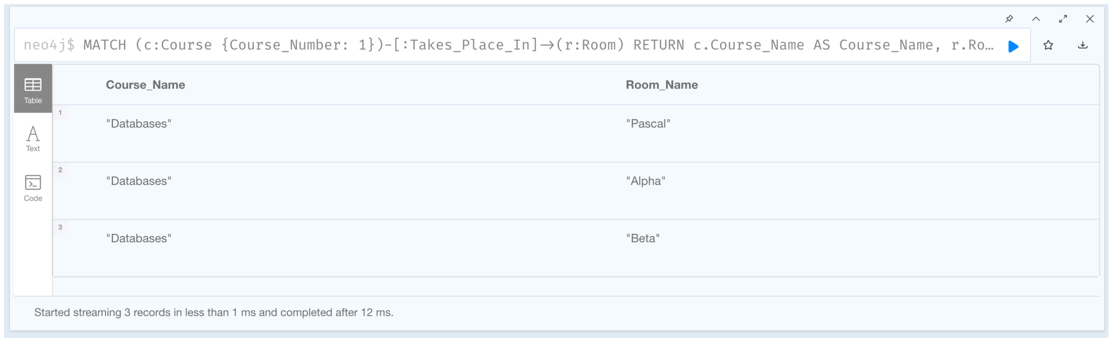
a. In which rooms does the course with Course number 1 takes place? Retrieve the course name and the names of the rooms in which the course takes place.

SQL query

SELECT c.Course_Name, r.Room_Name FROM Courses c INNER JOIN Rooms r ON c.Room_ID = r.Room_ID WHERE c.Course_Number = 1;

cypher query

MATCH (c:Course {Course_Number: 1})-[:Takes_Place_In]->(r:Room) RETURN c.Course_Name AS Course_Name, r.Room_Name AS Room_Name



```
neo4j$ MATCH (c:Course {Course_Number: 1})-[:Takes_Place_In]→(r:Room) RETURN c.Course_Name AS Course_Name, r.Ro…
```

	Course_Name	Room_Name
1	"Databases"	"Pascal"
2	"Databases"	"Alpha"
3	"Databases"	"Beta"

Started streaming 3 records in less than 1 ms and completed after 12 ms.

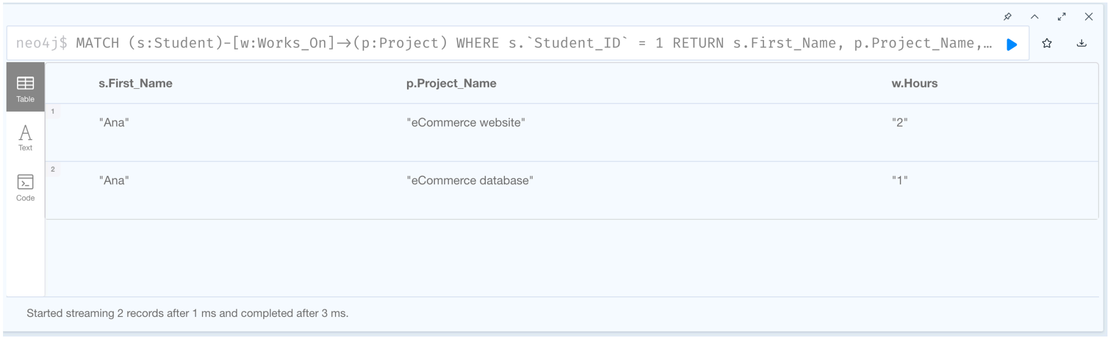
b. How many hours and in which projects does student with Student ID 1 works on? Retrieve the first name of the student, the project the student works on, and the corresponding num- ber of hours worked on the project.

SQL query

SELECT s.First_Name, p.Project_Name, w.Hours FROM Student s JOIN Works_On w ON s. `Student_ID` = w. `Student_ID` JOIN Project p ON w.Project_Number = p.Project_Number WHERE s. `Student_ID` = 1;

cypher query

MATCH (s:Student)-[w:Works_On]->(p:Project) WHERE s. `Student_ID` = 1 RETURN s.First_Name, p.Project_Name, w.Hours



```
neo4j$ MATCH (s:Student)-[w:Works_On]→(p:Project) WHERE s.`Student_ID` = 1 RETURN s.First_Name, p.Project_Name,…
```

	s.First_Name	p.Project_Name	w.Hours
1	"Ana"	"eCommerce website"	"2"
2	"Ana"	"eCommerce database"	"1"

Started streaming 2 records after 1 ms and completed after 3 ms.

c. Which students and how many hours do they work on the project with Project number 24? Retrieve the project name, the last name of the student, and the corresponding number of hours worked on the project.

SQL query

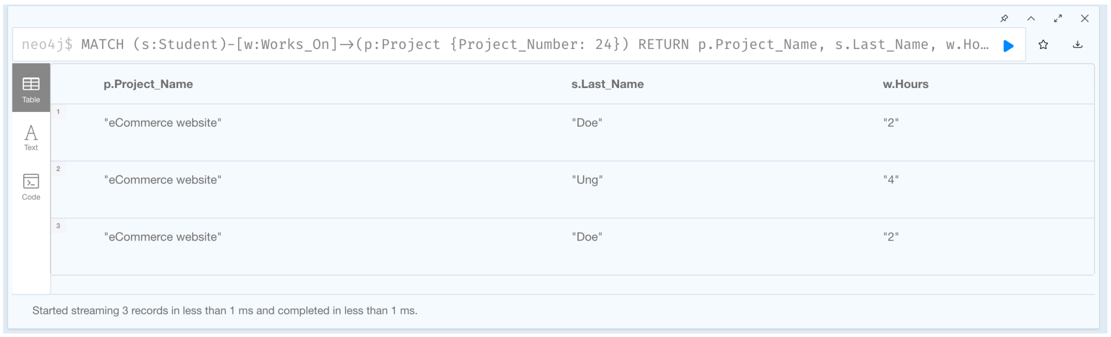
SELECT p.Project_Name, s.Last_Name, w.Hours FROM Project p JOIN Works_On w ON p.Project_Number = w.Project_Number JOIN Student s ON w.Student_ID = s.Student_ID WHERE p.Project_Number = 24;

cypher query

MATCH (s:Student)-[w:Works_On]->(p:Project {Project_Number: 24}) RETURN p.Project_Name, s.Last_Name, w.Hours

Or

MATCH (s:Student)-[w:Works_On]->(p:Project) WHERE p.Project_Number = 24 RETURN p.Project_Name, s.Last_Name, w.Hours



```
neo4j$ MATCH (s:Student)-[w:Works_On]→(p:Project {Project_Number: 24}) RETURN p.Project_Name, s.Last_Name, w.Ho…
```

	p.Project_Name	s.Last_Name	w.Hours
1	"eCommerce website"	"Doe"	"2"
2	"eCommerce website"	"Ung"	"4"
3	"eCommerce website"	"Doe"	"2"

Started streaming 3 records in less than 1 ms and completed in less than 1 ms.

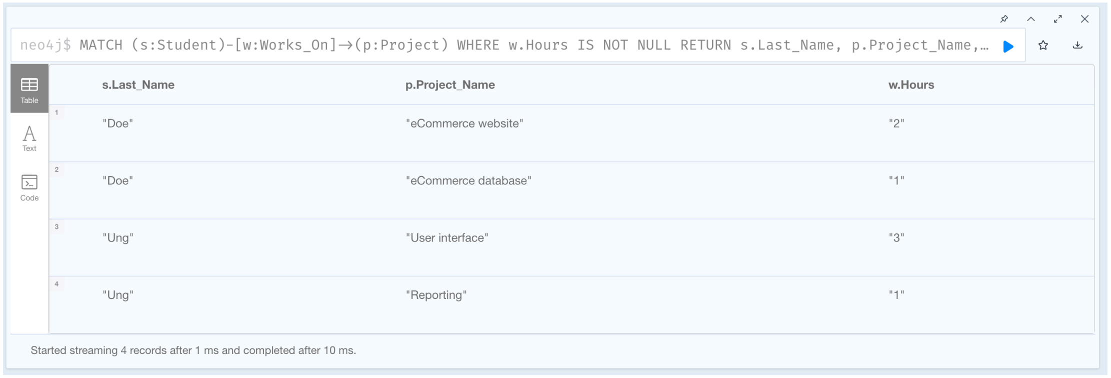
d. Which students work in which projects and how many hours? Retrieve the last name of the students, the name of the projects they work on, and the corresponding number of hours. Order the results by the last name of the students. Limit the results to four.

SQL query

SELECT s.last_name, p.project_name, w.hours FROM works_on w JOIN student s ON s.student_id = w.student_id JOIN project p ON p.project_number = w.project_number ORDER BY s.last_name LIMIT 4;

cypher query

MATCH (s:Student)-[w:Works_On]->(p:Project) WHERE w.Hours IS NOT NULL RETURN s.Last_Name, p.Project_Name, w.Hours LIMIT 4



```
neo4j$ MATCH (s:Student)-[w:Works_On]→(p:Project) WHERE w.Hours IS NOT NULL RETURN s.Last_Name, p.Project_Name,…
```

	s.Last_Name	p.Project_Name	w.Hours
1	"Doe"	"eCommerce website"	"2"
2	"Doe"	"eCommerce database"	"1"
3	"Ung"	"User interface"	"3"
4	"Ung"	"Reporting"	"1"

Started streaming 4 records after 1 ms and completed after 10 ms.

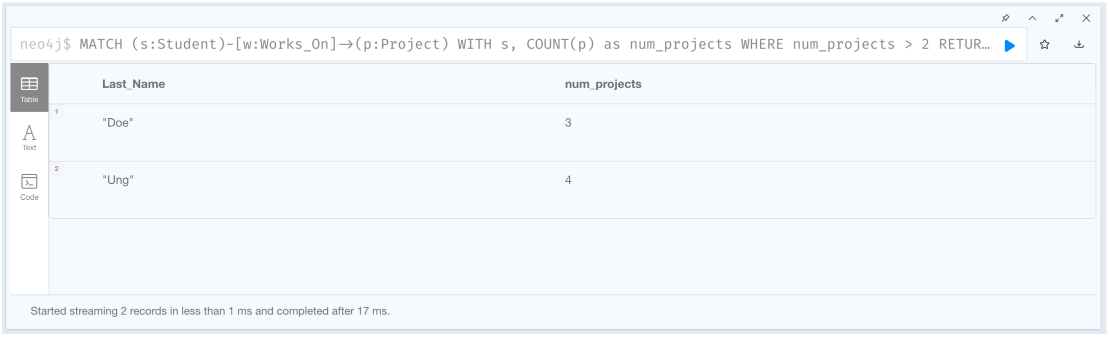
e) Whichstudentsworkonmorethantwoprojectsandonhowmanyprojectsexactly?Retrieve the last name of the students and the corresponding number of projects. Order the results by the number of projects.

SQL query

SELECT students.Last_Name, COUNT() *AS num_projects FROM works_on INNER JOIN students ON works_on.Student_ID = students.Student_ID GROUP BY works_on.Student_ID, students.Last_Name HAVING COUNT*() > 2 ORDER BY num_projects;

cypher query

MATCH (s:Student)-[w:Works_On]->(p:Project) WITH s, COUNT(p) as num_projects WHERE num_projects > 2 RETURN s.Last_Name as Last_Name, num_projects ORDER BY num_projects



```
neo4j$ MATCH (s:Student)-[w:Works_On]→(p:Project) WITH s, COUNT(p) as num_projects WHERE num_projects > 2 RETUR…
```

	Last_Name	num_projects
1	"Doe"	3
2	"Ung"	4

Started streaming 2 records in less than 1 ms and completed after 17 ms.

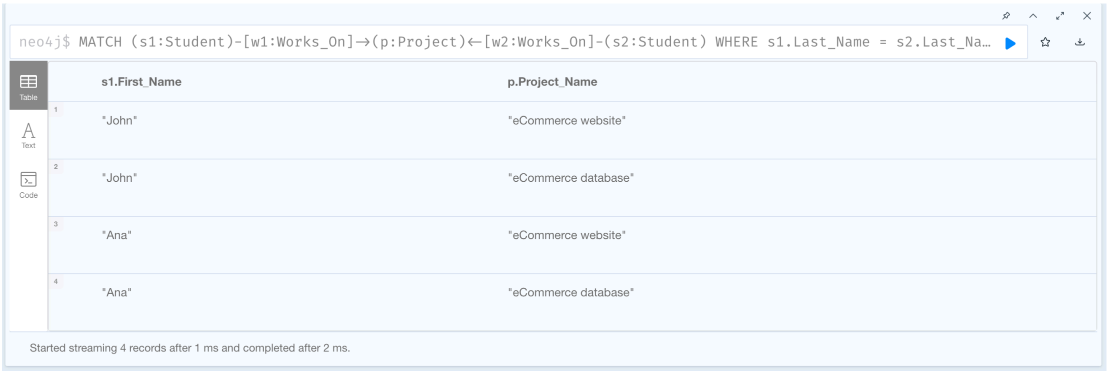
f. Which students have the same last name and work on the same projects? Retrieve the first name of the students and the name of projects they share.

SQL query

SELECT s1.first_name, s2.first_name, p.project_name FROM works_on w1 JOIN works_on w2 ON w1.project_number = w2.project_number AND w1.student_id < w2.student_id JOIN student s1 ON s1.student_id = w1.student_id JOIN student s2 ON s2.student_id = w2.student_id AND s1.last_name = s2.last_name JOIN project p ON p.project_number = w1.project_number ORDER BY s1.last_name, s2.last_name, p.project_name;

cypher query

MATCH (s1:Student)-[w1:Works_On]->(p:Project)<-[w2:Works_On]-(s2:Student) WHERE s1.Last_Name = s2.Last_Name AND w1.Hours = w2.Hours RETURN s1.First_Name, p.Project_Name



```
neo4j$ MATCH (s1:Student)-[w1:Works_On]→(p:Project)←[w2:Works_On]-(s2:Student) WHERE s1.Last_Name = s2.Last_Na…
```

	s1.First_Name	p.Project_Name
1	"John"	"eCommerce website"
2	"John"	"eCommerce database"
3	"Ana"	"eCommerce website"
4	"Ana"	"eCommerce database"

Started streaming 4 records after 1 ms and completed after 2 ms.