Project 2

I. Project Description

In your first project, you were asked to implement the University database in MySQL. Many of you have done an excellent job. Figure 1 shows an ER diagram.

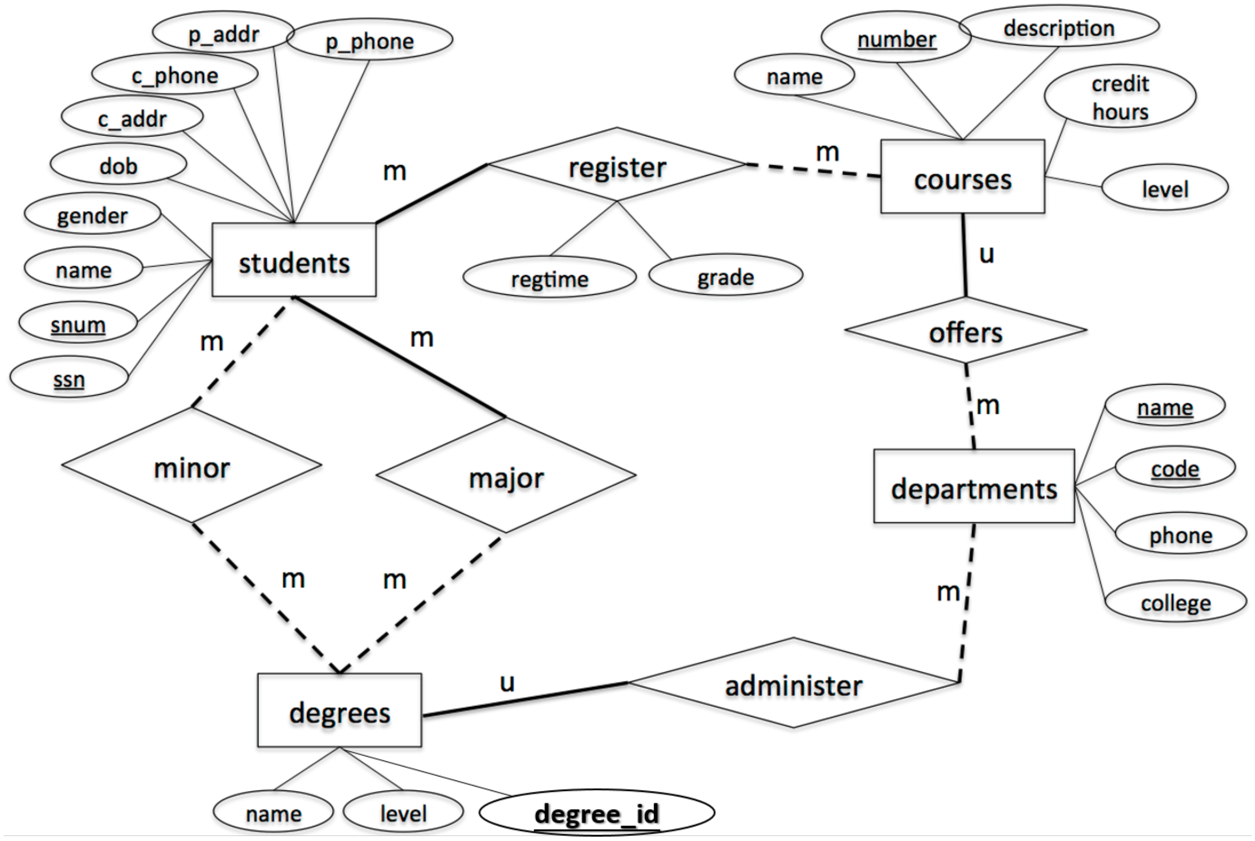

Figure 1. ER-diagram

This project is to implement the above design using a graphic model and query in Neo4J. The tasks include:

1. ImportData [Points: 0]

Create the graph database using the Data Importer. The csv files are provided which you can download. In order to enable Data Importer, three csv files are changed compared to Project 1, namely degrees.csv, major.csv, and minor.csv. These three csv files include a degree_id column, which is the identifier for degrees. You need to load those csv files into Neo4J to create nodes and edges reflecting the ER design:

The properties, labels for nodes and edges need to match with the ER-diagram exactly. For students nodes, use snum as the key; for courses, use number as the key; for departments, use code as the key; for degrees, use degree_id as the key

The direction of edges needs to follow:
students-register->courses
departments-administer->degrees
departments-offers->courses
students-major-> degrees
students-minor-> degrees

2. Query [Points: 82]

This script query the following information

1) (3pts) The campus addresses of the students whose name is "Amy"
2) (4pts) The major's names and levels of the students whose name is "Gail"
3) (4pts) The numbers and names of all courses offered by the department of Computer Science, order by course number
4) (4pts) The names of the students enrolled in Fall2022 semester.
5) (4pts) All degree names and levels administered by the department Computer Science, order by degree level
6) (4pts) The snums and names of all students who have a minor, order by student snum
7) (5pts) The names and snums of all non-undergraduate students ever enrolled in course "Database", order by snum. ("non-undergraduate students" means the major degrees of these students are MS or PhD levels)
8) (5pts) The names, snums and SSNs of the students whose name contains letter "n" or "N", order by snum
9) (4pts) The names, snums and SSNs of the students whose name is between "Amy" and "Nicole" (including Amy and Nicole), order by name
10) (5pts) The course number, name and the number of students ever registered for each course, order by course number (if a course has no student registered, the count should be 0)
11) (5pts) The count of female students who major or minor in Software Engineering degrees at any level
12) (5pts) The degree name, degree level, and number of students of the least popular majors (I.e., the major with the lowest number of students), order by degree name if there is a tie
13) (5pts) The degree name, degree level, and number of students of the least popular degrees (I.e., the degree program with the lowest number of students taking it as major or minor), order by degree name if there is a tie

14) (5pts) The numbers and names of courses and their corresponding average grades from students registered in the past semesters.
15) (5pts) The count of female students who major or minor in a degree managed by LAS departments
16) (5pts) The name and level of degrees that have more male students than female students (major or minor)
17) (5pts) The shortest paths between “Lisa” and “Computer Science” (ignore the directions of all edges). Note: “Lisa” is a student name
18) (5pts) All the adjacent nodes of “Lisa” (ignore the directions of all edges). Note: “Lisa” is a student name

3. ModifyRecords [Points: 18 + 6bonus]

This script modifies the following information

1) (4pts) Change the name of the student with ssn = 144673371 to Scott
2) (5pts) Change the major of the student with ssn = 144673371 to Computer Science, Master.
3) (4pts) Delete all registration records that were in “Summer2024”.
4) (5pts) Delete the “Applied Mathematics” department and any degrees “Applied Mathematics” department is administering. Delete all edges associated with the department and degrees, too.
5) (Bonus 6pts) If a group of courses have the same level and department_code, only keep the one with the smallest course number and delete the rest. If a course is deleted, all the edges associated with the courses should also be deleted.

Submission Instruction

1. ***Gradescope is set up for ‘Query’ tasks. Please copy your neo4j code for each query in one txt file and submit them in gradescope. The file names must be exact same as below for 18 Query files:***

   query_1.txt
   query_2.txt
   query_3.txt
   query_4.txt
   query_5.txt
   query_6.txt
   query_7.txt
   query_8.txt
   query_9.txt

query_10.txt
query_11.txt
query_12.txt
query_13.txt
query_14.txt
query_15.txt
query_16.txt
query_17.txt
query_18.txt

2. ***The ModifyRecords query will be graded manually by TA.***
   All the cypher code must be in one txt file and upload to gradescope.
   The file name should be "ModifyRecords.txt".