COMS4111 - Introduction to Databases
Homework 2

I started by enumerating the attributes for the 5 entities given in the Lecture 4 slides. While enumerating these attributes, the only independent entities out of the 5 are Department and Course. They are independent in sense that the attributes of these 2 tables don't refer to the attributes of the other tables.

Next, I defined the attributes of Student and Faculty. In addition to conventional and trivial attributes like firstname and lastname, I added 2 other attributes to each entity's schema namely "DeptID", "DeptName". These are 2 attributes that make them dependent on the table Department. Since, these 2 attributes refer the table Department, I added a foreign key constraint on these 2 columns. Thus, ("DeptID", "DeptName") in Student and Faculty schema refers to the ("DeptID", "DeptName") in the Department schema.

For a valid definition of a foreign key constraint, MySQL requires that the referred table declare the columns as unique.
If a foreign key constraint is added on the column(s) shared by 2 tables, it usually implies that the user queries shall often involve both the tables, perhaps, most often a join on the 2 tables. The execution of these queries can be optimized by having indexes on those columns in the tables. (MySQL does that by default on addition on a foreign constraint)

Thus for all the foreign key constraints I have added in my model, I have also added indexes on appropriate columns and declared the compound column as primary key which takes care of the uniqueness.

Following is a table that summarizes the foreign key constraints I have used in my model. Corresponding to a table in the model, it gives the list of attributes that are referenced from this table.

Table	Referenced Attributes
Student	StudentID, FirstName, LastName
Course	CourseID, FullName
Faculty	FacultyID, FacultyFName, FacultyLName
Department	ID, Name
Sections	CourseID, CourseName, SectionNumber

Next, I worked out what kind of relationships would exist between the entities already defined. I have modelled these relationships as separate schemas. The relationships that are relevant in this scenario are -

- Faculty ***heads*** the department
- Department ***offers*** Courses.
- Student ***enrolls in*** Courses.
- Student has ***completed*** some courses.
- The course X is a ***prerequisite for*** the course Y.
- Faculty ***instructs*** a course.

Following is a table, where I have mentioned the name of the schema that I have used in my model to represent the above relationships.

Relationship	Schema
heads	Headed_by
offers	Offered_by
enrolls	Has_enrolled_in
completed	Has_completed
instructs	Sections
prerequisite for	Prerequisite

The schema for the “Prerequisite” has 4 attributes. These imply that Course X represented as (CourseID, CourseName) has a prerequisite Course Y represented as (PrereqCourseID, PrereqCourseName). To validate both the courses, each of them individually is bound by a foreign key constraint to the “Course” table.

The “Sections” table serves 2 purposes: one is to enumerate the sections for a course and the other is to mention the instructor for each section. Its relationship with “Faculty” and “Course” imply the many-to-many relationship between the 2 entities i.e. a faculty can be an instructor for multiple courses and a course can have employ multiple instructors depending on the number of sections the course offers.

“Has_enrolled_in” and “Has_completed” represents the many-to-many relationship between “Student” and “Course” implying that a student can enrol (or has completed) multiple courses and a course can have multiple participating students.

“Headed_by” implies one-to-many relationship between “Faculty” and “Department” implying that a department can be headed by a single faculty, however, a faculty may head multiple departments.

“Offered_by” represents one-to-many relationship between “Department” and “Course” implying that a department may offer multiple courses but a course is always offered by a single department only.

The relationship between Department and Student (or Faculty) is one-to-many implying that a student/faculty can belong to a single department only but a department may have multiple members.

I have submitted DDL as a separate file named “DDL.sql”.

Following are the user stories -

- ➢ A student X in Computer Science department wants to get the list of the course CS department is offering.

  Query used:
  - SELECT CourseName FROM Offered_by WHERE DeptName = "Computer Science";

```
mysql> SELECT CourseName FROM Offered_by WHERE DeptName = "Computer Science";
+--------------------------------+
| CourseName                     |
+--------------------------------+
| Intro to DBMS                  |
| Principles of Network Security |
| Machine Learning               |
| Deep Learning                  |
| Analysis of Algorithms 1       |
| Analysis of Algorithms 2       |
| Advanced Machine Learning      |
| Bayesian Models                |
| Data Structures                |
| Graphics & Animation           |
+--------------------------------+
10 rows in set (0.00 sec)
```

➢ The student X is interested in enrolling for the course “Principles of Network Security”. However, he wants to find out if he has completed all the prerequisites.

Query used:

- SELECT CourseName, PrereqCourseName FROM Prerequisite WHERE CourseName = 'Principles of Network Security';

```
mysql> SELECT * FROM Course where FullName = 'Principles of Network Security';
+----------+--------------------------------+------------+
| CourseID | FullName                       | CourseCode |
+----------+--------------------------------+------------+
| 10       | Principles of Network Security | CS-0010    |
+----------+--------------------------------+------------+
1 row in set (0.00 sec)

mysql> SELECT CourseName, PrereqCourseName FROM Prerequisite WHERE CourseName = 'Principles of Network Security';
Empty set (0.00 sec)
```

Since the course has no prerequisites, the student X can enrol for this course.
(In the screenshot, the first query on table ‘Course’ is used to double-check if there is no typo in the spelling of the course name).

➢ “Khalil swift” is interested in enrolling for “Intro to DBMS”, so she pulls up the list of prerequisite(s) for the course. Although, she has not completed the prerequisite yet she has relevant industry experience in DBMS. So, she decides to talk to an instructor to discuss if she is an apt candidate for the course. She needs to find out all the instructor(s) for this course.

Query used:

- SELECT CourseName, PrereqCourseName FROM Prerequisite WHERE CourseName = 'Intro to DBMS';
- SELECT * FROM Has_completed WHERE StudentID = '187456';
- SELECT * FROM Sections WHERE CourseName='Intro to DBMS';

```
mysql> SELECT CourseName, PrereqCourseName FROM Prerequisite WHERE CourseName = 'Intro to DBMS';
+---------------+---------------------+
| CourseName    | PrereqCourseName    |
+---------------+---------------------+
| Intro to DBMS | Graphics & Animation |
+---------------+---------------------+
1 row in set (0.16 sec)

mysql> SELECT * FROM Has_completed WHERE StudentID = '187456';
+-----------+-------------+-------------+----------+-----------------------------------------+---------------+-------+
| StudentID | StudentFName | StudentLName | CourseID | CourseName                              | SectionNumber | Grade |
+-----------+-------------+-------------+----------+-----------------------------------------+---------------+-------+
| 187456    | Khalil      | Swift       | 31       | Digital Logic Design                    |             1 | B     |
| 187456    | Khalil      | Swift       | 61       | History of Black Magic                  |             2 | A     |
| 187456    | Khalil      | Swift       | 68       | Ethical Concerns associated with black magic |             2 | C     |
+-----------+-------------+-------------+----------+-----------------------------------------+---------------+-------+
3 rows in set (0.09 sec)
mysql> SELECT * FROM Sections WHERE CourseName='Intro to DBMS';
+-----------+---------------+---------------+-----------+------------------+------------------+
| CourseID  | CourseName    | SectionNumber | FacultyID | InstructorFName  | InstructorLName  |
+-----------+---------------+---------------+-----------+------------------+------------------+
| 1         | Intro to DBMS |             1 | 4773645   | Carmen           | Yundt            |
+-----------+---------------+---------------+-----------+------------------+------------------+
1 row in set (0.00 sec)
```

➢ A student Y is interested in enrolling for the online sessions of some of his tentative courses. However, Y is skeptical about the efficacy of online session. So, Y decides to pull up enrollment statistics on the sections. Y's aim to calculate the proportion of all the students who enrol for the online sessions of the course.

(A student can opt for online sessions by enrolling in Section 4 of any course, if the course offers the option for online sessions.)

Query used:

- SELECT SectionNumber, COUNT(*) FROM Has_enrolled_in GROUP BY SectionNumber;

```
mysql> SELECT SectionNumber, COUNT(*) FROM Has_enrolled_in GROUP BY SectionNumber;
+---------------+----------+
| SectionNumber | COUNT(*) |
+---------------+----------+
|             1 |       64 |
|             2 |       61 |
|             3 |       15 |
|             4 |       20 |
+---------------+----------+
4 rows in set (0.09 sec)
```

Therefore, Y calculates the proportion = $\frac{20}{64 + 61 + 15 + 20} = 0.125$ or 12.5%.

➢ Y concludes 12.5% is a significant percentage. So, Y decides to enrol for online sessions for some of his courses. Y decides to find out all the courses offered by his department that offer the students to enrol in online sessions. Assume, Y is a student in “Computer Science” department. He knows that the DeptID is ‘1’ for CS.

Query used:

- SELECT s.CourseName FROM Sections s JOIN Offered_by o ON s.CourseID = o.CourseID WHERE o.DeptID = '1' and s.SectionNumber = 1;

```
mysql> SELECT s.CourseName FROM Sections s JOIN Offered_by o ON s.CourseID = o.CourseID WHERE o.DeptID = '1' and s.SectionNumber = 1;
+-------------------------------+
| CourseName                    |
+-------------------------------+
| Intro to DBMS                 |
| Principles of Network Security |
| Machine Learning              |
| Data Structures               |
| Graphics & Animation          |
+-------------------------------+
5 rows in set (0.00 sec)
```

➢ University’s academic office has decided to increase their participation in the MOOCs. The aim is to start the “Section 4” for as many courses as possible. The office needs the list of all the courses that don’t offer Section 4 as of now, so that appropriate arrangements can be made to make, the online sessions for these courses, available on Video networks & MOOCs.

Query used:

- SELECT CourseName FROM Sections WHERE CourseName NOT IN (SELECT DISTINCT CourseName FROM Sections WHERE SectionNumber = 1);

```
mysql> SELECT CourseName FROM Sections WHERE CourseName NOT IN (SELECT DISTINCT CourseName FROM Sections WHERE SectionNumber = 1);
+-------------------------------------------+
| CourseName                                |
+-------------------------------------------+
| Analog Circuits                           |
| Wireless Communications                   |
| Communication Theory                      |
| Communication Networks                    |
| Analog & mixed signal design              |
| Building materials & construction         |
| Mechanics of Solids                       |
| Mechanics of Solids                       |
| Strength of materials                     |
| Strength of materials                     |
| Soil Mechanics                            |
| Soil Mechanics                            |
| Environmental Science                     |
| Design of steel Structures                |
| Design of steel Structures                |
| Deep Learning                             |
| Transportation Engg.                      |
| Transportation Engg.                      |
| Engineering Electronics 1                 |
| Basic Electronic Circuits                 |
| Basic Electronic Circuits                 |
| Semiconductor Processing                  |
| Lasers & Photonics                        |
| Bioelectrics                              |
| Analysis of Algorithms 1                  |
| Optical Communications                    |
| Number Theory in Cryptography             |
| Analysis of Algorithms 2                  |
| Fourier & Laplace Transforms              |
| University Physics                        |
| University Physics                        |
| Quantum Mechanics                         |
| Semiconductor Physics                     |
| Semiconductor Physics                     |
| Electricity, Magnetism, and Waves         |
| Mathematical Methods for Physics          |
| Advanced Machine Learning                 |
| Foundations of Modern Physics             |
| Foundations of Modern Physics             |
| Foundations of Modern Physics             |
| Ethical Concerns associated with black magic |
| Gray Hat and White hat practitioners      |
| Bayesian Models                           |
+-------------------------------------------+
43 rows in set (0.09 sec)
```

➢ In addition to Section 4, the office has decided to offer one more section for the five most popular courses. (Popularity is determined by number of enrollments for that course).

Query used:

- SELECT CourseName, COUNT(*) as NumEnrollments FROM Sections GROUP BY CourseName ORDER BY NumEnrollments DESC LIMIT 1,5;

```
mysql> SELECT CourseName, COUNT(*) as NumEnrollments FROM Sections GROUP BY CourseName ORDER BY NumEnrollments DESC LIMIT 1,5;
+------------------------------+----------------+
| CourseName                   | NumEnrollments |
+------------------------------+----------------+
| Electromagnetic Field Theory |              3 |
| Linear Electronic Circuits   |              3 |
| Foundations of Modern Physics |              3 |
| Microelectronics             |              3 |
| Soil Mechanics               |              2 |
+------------------------------+----------------+
5 rows in set (0.00 sec)
```

➢ Domains of different departments often overlap. Thus, students within a department often enrol in courses of the other departments to gather knowledge relevant to their fields. This often leads to the competition during the registration period. So, all the departments have decided to find out the most popular cross-department courses and start offering their own versions of these courses or assimilate the relevant content in the existing courses.

Query used:

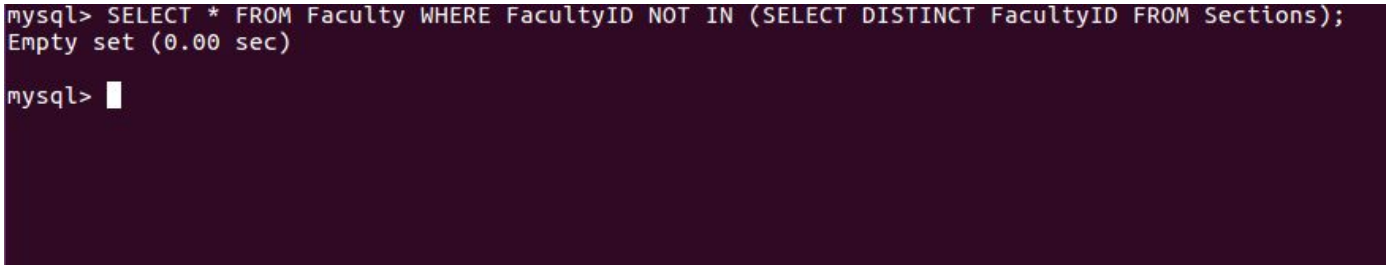
- SELECT o.CourseName FROM Has_enrolled_in e JOIN Student s JOIN Offered_by o ON e.StudentID = s.StudentID AND e.CourseID = o.CourseID and s.DeptId <> o.DeptID;

```
mysql> SELECT o.CourseName FROM Has_enrolled_in e JOIN Student s JOIN Offered_by o ON e.StudentID = s.StudentID AND e.CourseID = o.CourseID and s.DeptId <> o.DeptID;
+-----------------------------------------+
| CourseName                              |
+-----------------------------------------+
| Bioelectrics                            |
| Electricity, Magnetism, and Waves       |
| Intro to Quantum Optics                 |
| General Chemistry for Engineers         |
| Microelectronics                        |
| Lasers & Photonics                      |
| Solving NP-Hard problems with black magic |
| Graphics & Animation                    |
| Structural Analysis                     |
| Computational Black Magic               |
| Fourier & Laplace Transforms            |
| Analog Circuits                         |
| Ethical Concerns associated with black magic |
| Bayesian Models                         |
| Semiconductor Processing                |
| Practical Applications of black magic   |
| Linear Algebra                          |
| Fourier & Laplace Transforms            |
| University Physics                      |
| Signals & Systems                       |
| Solving NP-Hard problems with black magic |
| Optical Communications                  |
| Basic Electronic Circuits               |
| Signals & Systems                       |
| Abstract Algebra                        |
| Microelectronics                        |
| Embedded Hardware Design                |
| Black Magic lab 1                       |
| Principles of Network Security          |
| Design of concrete Structures           |
| General Chemistry for Engineers         |
| Practical Applications of black magic   |
| Bioelectrics                            |
| Chemistry for black magic               |
| Basic Electrical Science                |
| Principles of Network Security          |
| Engineering Electronics 1               |
| Data Structures                         |
| Design of concrete Structures           |
| Calculus 2                              |
| Wireless Communications                 |
+-----------------------------------------+
41 rows in set (0.00 sec)
```

➢ University's goal to help each student land an internship by this summer. However, not all the students seek industrial internships. On campus research opportunities are often limited due to lack of apt infrastructure. The Student committee has decided to seek the help of the faculty who are not instructors for any section and hence, can participate in this initiative by opening up new research positions in their respective department.

Query used:

- SELECT * FROM Faculty WHERE FacultyID NOT IN (SELECT DISTINCT FacultyID FROM Sections);



** END **

Submitted by -
Aayush Maini
am4810