CSC675 Database system

College Department Database Management System
Milestone 2
M2V1
November 9, 2022

HARAKALY Robert
SFSU ID: 922979730
rharakaly@sfsu.edu
San Francisco State University

M1V1	October 5, 2022	Completed first version
M1V2	November 9, 2022	Completed second version
M2 V1	November 9, 2022	Completed Milestone 2

Contents

1 Milestone 1

1.1 Project Description

The University has problems with students who contact professors to ask about prerequisite and about qualifications. The Database will allow students to show either their prerequisite or qualifications. The Database management system will be the gateway between students/professors/courses and extra curricular activities.
It will help students to provide an easier tool to prove qualifications for prerequisites and it will also help them facilitate communication between students/professors and administration. The database would provide students more visibility about extra curricular activities and employment which is under developed. Currently Universities do not provide a single unified location for these.

1.2 Use Cases

1. **Use Case:** Courses organization
   Actor: Manon, Administration at Sfsu
   Description: Recently, Miss Manon in the administration received many messages that some students can't go in a course. They have a course in a building downtown but then they have another course in another building in the main campus. They can't go in this course on time so they ask to change the Class section so they can finally go in this course.

   Solution: Intra is a platform for management of college/university. With this platform, Manon can see the timetable of each students (and also professors) and can see also in which facilities/building the course takes place. After this, she can finally remove the student from his former section and add him in the new section. The platform will also automatically tell to the students that are applying for a course that they can't apply in some section because it's in the same time or it's too far to be in time.

2. **Use Case:** Course
   Actor: Miss Barbara, Professor
   Description: Miss Barbara is a professor at Sfsu, she noticed that many of her students did not take notes of the course or they don't understand the course and need to review again the course. She also noticed that students do not their homework because they always forget the date when it is due or that they lost the paper.

   Solution: A platform where she can store information of her courses like the lesson, assignments or also the grades for her students. Students must be assigned in this course to see elements of this course and can also sent theirs assignments.

3. **Use Case:** Collaborative assignments
   Actor: Luca, Marketing Student at Sfsu
   Description: Luca is a Marketing Major and for his Master project to get his degree, he must think about a new marketing model for a company. He have a great idea to use new technologies to support his new marketing models. But he's not a Computer science student nor an engineer student, he needs some help from students with other Major. So he needs to contact them (or also contact the professor of the other field to ask him some questions) to succeed his project. But he doesn't know no one in other departments so he struggle to find a teammate and to communicate with him.

   Solution: The platform will have a communication features. So this way, Luca can find people in Computer Science department and can write them a message that he need some help. He can also see a professor from one specific course and ask him some questions to help him how to improve his project. If many people accept to help him, Luca have the possibility to create a chat group.

4. **Use Case:** Prerequisite courses
   Actor: Daniel, Exchange student at Sfsu
   Description: Daniel is an exchange student from Slovakia, he comes to Sfsu to study but the University wants him to have some prerequisites to have courses that he wants for his Minor. But as an exchange student, he don't have courses that are only available at

Sfsu. He must check the description of the course manually and suppose that the courses he had is equivalent to the course needed at Sfsu and he's not sure because some times, the description is not precise. He put his courses that he had and was hopping that he will be enrolled, but the professor who saw his courses, also didn't understand want he knows or not because the names off the courses are different and she don't have at all the descriptions of courses he had.

Solution: During registration of the courses, the platform will ask to Daniel is he's a new Sfsu student or not. If he is, the platform will ask him his grades from Sfsu. But Daniel is a "new" student so he can't have the Sfsu courses prerequisite. The platform will show him a MCQS of knowledge that he must know (all or the most important ones, depends on the importance of the knowledge) to be enrolled in the course he wants (The MCQS is based on knowledge learned in prerequisite courses). If Daniel checks everything, the professor will know that Daniel have the knowledge to be enrolled and if not, it's up to the professor to decide if Daniel will be enrolled or not. Then he platform will ask him to send his grades from his former school as a proof and also to describe what he learned in this course to be sure he has knowledge.

5. **Use Case:** The grader's work
   Actor: Ginaluca, Lecturer at Sfsu & Clement, its grader
   Description: Mister Gianluca is a lecturer at Sfsu. He has a grader called Clement who helps him with graduate students but he also helps students during assignments. But some students complain that the grades that Clement give them is not good, they think they deserve better. So Gianluca must verify the assignments but assignments in paper take many times to verify and Gianluca don't have much time for this.

   Solution: The assignments will be submitted in the platform. The grader can access in those assignments and give a grade, the professor also have accesses and can modify what the grader wrote but the grader can't modify the grade that the professor give to one student. So this way, it will be easier for the professor to check Clement's work and he saw that Clement did a great job, the other students will have the same grade.

6. **Use Case:** On-Campus Job
   Actor: Arjun, Student at sfsu
   Description: Arjun is a civil engineer student at Sfsu. He's from India so he needs money for his housing and for foods. He's looking at LinkedIn or other web sites to find a job in San Francisco. But he didn't knew that they are also paid jobs on-Campus. Fortunately, someone told him that there are jobs on Campus, he just need to put his resume and the HR will see if Arjun will be enrolled or not. Arjun is very lucky because someone told him that he can work on Campus and he was successfully enrolled in a job that one faculty member posted so now he can finally be in the payroll. But what if no one told him about this opportunity?

   Solution: The platform will have a window for job opportunities at University, this window even students and non-students have access but for the students, to give them more chances, they will always see all new jobs in the home page of the platform so they can be the first to see it and to apply.

7. **Use Case:** Researcher's publications
   Actor: Mister Hugo, Physics Researcher at Sfsu
   Description: Mister Hugo is a Physics Researcher at Sfsu and he noticed that no many students read his reports which is a shame because they can learn a lot from it.

   Solution: Hugo have a special page for his research in this platform so every students and professors can see his work.

8. **Use Case:** Need more players
   Actor: Thibault, coach of the sfsu soccer club
   Description: Thibault is the coach of the sfsu soccer club and he's worried because he needs more players to be more competitive this year. He asked to the head director of the sport University association if he can do something. The head president posted an announcement at sfsu's announcement board but one one noticed. Unfortunately, Thibault will have a complicated year.

   Solution: The University organization's announcement can be posted directly on the home page of the platform like the job announcements. So this way, everyone can notice this announcement and can come play Soccer and Thibault will be happy to have more players.

9. **Use Case:** Opportunity for students
   Actor: Jose, Lecturer at sfsu
   Description: Mister Jose is a Lecturer at sfsu, it means that he's not only teaching but he's also in a company. And he knows that many students are struggling with finding a job in the Bay Area. He also knows that students at sfsu are good and have many qualities that he can need for his company. He decided to give students an opportunity to learn even more than just having theories at University, so he wants to tell to his students that he can offer opportunities for a part-time job. But he feels bad because he's only offering for his students, others have not this opportunity.

   Solution: Faculty member as well as head president off association of administration has the right to add some announcement in the platform as an off-campus opportunity.

10. **Use Case:** Christmas Celebration
    Actor: Suzanne, president of Student Council
    Description: Suzanne is the president of the Student Council, she wants the best for students. Students had a complicated week because of the stress of final exams but now it's finished and she wants to organize a Christmas Party Celebration so students can now enjoy the end of the year.

    Solution: Suzanne can post an announcement in the platform but she can also sent a mail to everyone to announce the Party.

11. **Use Case:** Books
    Actor: Michael, Student at Sfsu
    Description: Michael is a student at Sfsu and he needs a book to do his homework. He

went to the Library to check books that they have but they are so many so he struggles to find the book he needs. He do not even know is the Library has this book.

Solution: The database can have store all books that they have and also where to find them in the library.

1.3 Database Requirements

1. Department

   1.1. Department is managed by one or many member of the Administration

   1.2. Department have many employees

   1.3. Department have many Facilities

2. Administration

   2.1. Administration admit Many students

   2.2. Administration can manage many departments

   2.3. Administration can manage many Class section

   2.4. Administration is a Employee

3. Course

   3.1. Course have many Students

   3.2. Course have many Professors

   3.3. Course can be assigned to a student by at least one member of the Faculty member

   3.4. Course can have one or many Class Section

   3.5. Course can have many Assignments

   3.6. Course have many Prerequisites

   3.7. Course have many Exams

   3.8. Course have one Description

   3.9. Course can have many Group Chat

   3.10. Course can be in many Fields

   3.11. Course have many lessons

4. Class Section

   4.1. One Class Section is assigned to one Course

   4.2. Class Section can have one Professor

   4.3. Class Section can have many students

   4.4. One Class Section is in one Facility

   4.5. Class Section can have many Homeworks

   4.6. Class Section can have many Exams

   4.7. Class Section can have many Group Chat

5. Learner

   5.1. Learner can have many Courses

   5.2. Learner can have one or many Field

   5.3. Learner have one Degree

   5.4. Learner can be a Grader

5.5. Learner can be a Exchange student
5.6. Learner can be in many class Section
5.7. Learner have many Homework
5.8. Learner can be in many associations
5.9. Learner can go in many events
5.10. Learner can have many activities
5.11. Learner can have many exams
5.12. Learner can be in many group chat
5.13. Learner can have many jobs
5.14. Learner can have many part time jobs

6. Student

6.1. Student is a Learner
6.2. Students can be in one or zero Student Council

7. Grader

7.1. Grader is a Learner
7.2. Grader is in one Payroll
7.3. Grader can give many grades to many students

8. Exchange student

8.1. Exchange student is a Learner
8.2. Exchange student must answer one or many MQCS for prerequisite

9. Employee

9.1. Employee can be a Faculty member
9.2. Employee can be a Administration member
9.3. Employee is in one payroll

10. Faculty member

10.1. Faculty member can be a Professor
10.2. Faculty member can be a Lecturer
10.3. Faculty member can be a Researcher
10.4. Faculty member is a Employee
10.5. Faculty member is in one payroll

11. Professor

11.1. Professor can have many Courses
11.2. Professor can have many students
11.3. Professor can be assigned in many class sections
11.4. Professor is a Faculty member

- 11.5. Professor can grade many students
- 11.6. Professor can give many homework
- 11.7. Professor can give many exams
- 11.8. Professor can be in many group chat

12. Researcher

    12.1. Researcher can write many paper research

    12.2. Researcher is a faculty member

13. Lecturer

    13.1. Lecturer is a Faculty member

    13.2. Lecturer can have many students

    13.3. Lecturer can have many courses

    13.4. Lecturer can be assigned in many class section

    13.5. Lecturer can have many jobs

    13.6. Lecturer can be in at least one payroll

    13.7. Lecturer can give many assignments

    13.8. Lecturer can give many exams

    13.9. Lecturer can graduate many students

    13.10. Lecturer can be in many group chat

14. Homework

    14.1. One Homework can be assigned to many Students

    14.2. One Homework can be assigned by many Professors

    14.3. One Homework can be assigned by many Lecturer

    14.4. One Homework can have zero or one grade

    14.5. One Homework can be graded by many Graders

15. Student Council

    15.1. Student Council can have many Students

    15.2. Student Council can create many Events

16. Association

    16.1. Association have many Learners

    16.2. Association can have many Activities

    16.3. Association can work with many Association

17. Library

    17.1. A Library is a Facility

    17.2. A Library have many Books

18. Book

    18.1. A Book can be in many Library
    18.2. A Book can be in possession off many Students

19. Activity

    19.1. There can be many students in one Activity
    19.2. Activity can be created by at least one Associations

20. Event

    20.1. There can be many students in one Event
    20.2. Event can be created by at least one Student Council

21. Prerequisites

    21.1. One prerequisite can be in many Courses
    21.2. Prerequisite is a Course

22. Assignment

    22.1. Assignment is given by a professor
    22.2. Assignment is done by many students
    22.3. Assignment is given in many courses

23. Exams

    23.1. Exam is a Assignment
    23.2. Exams are graded by many Graders

24. Group Chat

    24.1. One group chat can have many students
    24.2. One group chat can have many members of the Faculty member

25. HR

    25.1. HR can hire many employees
    25.2. HR is a employee

26. Payroll

    26.1. A payroll can have many employees

27. Paper research

    27.1. A paper research is written by one or many Researcher

28. Facility

    28.1. A Facility can belongs to many departments
    28.2. A Facility is a place where they are many courses

29. Amphitheater

    29.1. Amphitheater is a Facility

30. Laboratory

    30.1. Laboratory is a Facility

31. Classroom

    31.1. Classroom is a Facility

32. Degree

    32.1. A degree is enrolled by many students

33. Field

    33.1. A Field is enrolled by many students
    33.2. A Field is in many courses

34. Lesson

    34.1. A Lesson is in many courses
    34.2. A Lesson have then MCQS for international students

35. Part Time Job

    35.1. Part Time Job can be given to many Learners
    35.2. Part Time Job people are in Payroll
    35.3. Part Time Job is in a department

1.4 Detailed List of Main Entities, Attributes and Keys

1. Department (Strong)
    - 1.1. name: composite, alphanumeric
    - 1.2. department_id: key, numeric
    - 1.3. size: numeric
2. Administration (Weak)
    - 2.1. nb_person: numeric
    - 2.2. person_name: composite, alphanumeric
    - 2.3. size: numeric
3. Course (Strong)
    - 3.1. name: composite, alphanumeric
    - 3.2. course_id: key, numeric
    - 3.3. prerequisite: composite, alphanumeric
    - 3.4. field: composite, alphanumeric
4. Class Section (Strong)
    - 4.1. section_id: key, numeric
    - 4.2. Timetable: multivalue, timestamp
    - 4.3. class_size: numeric
    - 4.4. field: composite, alphanumeric
5. Learner (Strong)
    - 5.1. name: composite, alphanumeric
    - 5.2. id: key, numeric
    - 5.3. grade: numeric, multivalue
    - 5.4. GPA: numeric
    - 5.5. major: composite, alphanumeric
    - 5.6. minor: composite, alphanumeric
    - 5.7. degree: alphanumeric
6. Student (Weak)
    - 6.1. name: composite, alphanumeric
    - 6.2. id: key, numeric
    - 6.3. grade: numeric, multivalue
    - 6.4. GPA: numeric
    - 6.5. major: composite, alphanumeric
    - 6.6. minor: composite, alphanumeric
    - 6.7. degree: alphanumeric

7. Grader (Weak)
    7.1. name: composite, alphanumeric
    7.2. id: key, numeric
    7.3. grade: numeric, multivalue
    7.4. GPA: numeric
    7.5. major: composite, alphanumeric
    7.6. minor: composite, alphanumeric
    7.7. degree: alphanumeric
    7.8. course_to_grade: composite, alphanumeric
8. International Student (Weak)
    8.1. name: composite, alphanumeric
    8.2. id: key, numeric
    8.3. grade: numeric, multivalue
    8.4. GPA: numeric
    8.5. major: composite, alphanumeric
    8.6. minor: composite, alphanumeric
    8.7. degree: alphanumeric
9. Employee (Strong)
    9.1. name: composite, alphanumeric
    9.2. e_id: key, numeric
    9.3. wages: numeric
10. Faculty member (Weak)
    10.1. name: composite, alphanumeric
    10.2. f_id: key, numeric
    10.3. wages: numeric
11. Professor (Weak)
    11.1. name: composite, alphanumeric
    11.2. prof_id: key, numeric
    11.3. wages: numeric
12. Researcher (Weak)
    12.1. name: composite, alphanumeric
    12.2. r_id: key, numeric
    12.3. wages: numeric
13. Lecturer (Weak)

- 13.1. name: composite, alphanumeric
- 13.2. l_id: key, numeric
- 13.3. wages: numeric

14. Homework (Weak)
    - 14.1. number: composite, alphanumeric
    - 14.2. grade: numeric
    - 14.3. deadline: multivalue, timestamp
15. Assignments (Strong)
    - 15.1. number: composite, alphanumeric
    - 15.2. grade: numeric
    - 15.3. deadline: multivalue, timestamp
16. Student Council (Weak)
    - 16.1. name: composite, alphanumeric
    - 16.2. size: numeric
    - 16.3. budget: numeric
17. Association (Weak)
    - 17.1. name: composite, alphanumeric
    - 17.2. size: numeric
    - 17.3. budget: numeric
18. Book (Strong)
    - 18.1. title: composite, alphanumeric
    - 18.2. author: composite, alphanumeric
    - 18.3. price: numeric
19. Activity (Strong)
    - 19.1. name: composite, alphanumeric
    - 19.2. size: numeric
    - 19.3. date: multivalue, timestamp
    - 19.4. price: numeric
    - 19.5. type: alphanumeric
20. Event (Strong)
    - 20.1. name: composite, alphanumeric
    - 20.2. size: numeric
    - 20.3. date: multivalue, timestamp
    - 20.4. price: numeric

21. Exams (Weak)
    21.1. number: composite, alphanumeric
    21.2. grade: numeric
    21.3. date: multivalue, timestamp
22. Group Chat (Strong)
    22.1. name: composite, alphanumeric
    22.2. number_people: numeric
    22.3. chat_id: key, numeric
23. HR (Weak)
    23.1. name: composite, alphanumeric
    23.2. wages: numeric
    23.3. number_hr: numeric
24. Paper research (Weak)
    24.1. title: composite, alphanumeric
    24.2. author: composite, alphanumeric
    24.3. date_of_publication: multivalue, timestamp
25. Facility (Strong)
    25.1. name: composite, alphanumeric
    25.2. size_of_the_facility: numeric
    25.3. number_max_students: numeric
26. Amphitheater (Weak)
    26.1. name: composite, alphanumeric
    26.2. size: numeric
    26.3. number_max_students: numeric
27. Laboratory (Weak)
    27.1. name: composite, alphanumeric
    27.2. size: numeric
    27.3. number_max_students: numeric
28. Classroom (Weak)
    28.1. name: composite, alphanumeric
    28.2. size: numeric
    28.3. number_max_students: numeric
29. Library (Weak)

29.1. name: composite, alphanumeric
29.2. size: numeric
29.3. number_max_students: numeric

30. Field (Strong)

30.1. name: composite, alphanumeric
30.2. field_id: key, numeric
30.3.

31. Degree (Strong)

31.1. name: composite, alphanumeric
31.2. field_id: key, numeric
31.3. year_of_study: numeric

32. Lessons (Strong)

32.1. name: composite, alphanumeric
32.2. field: composite, alphanumeric
32.3. autor: composite, alphanumeric

33. Payroll (Strong)

33.1. list: multivalue, alphanumeric
33.2. wages_of_employees: numeric, multivalue

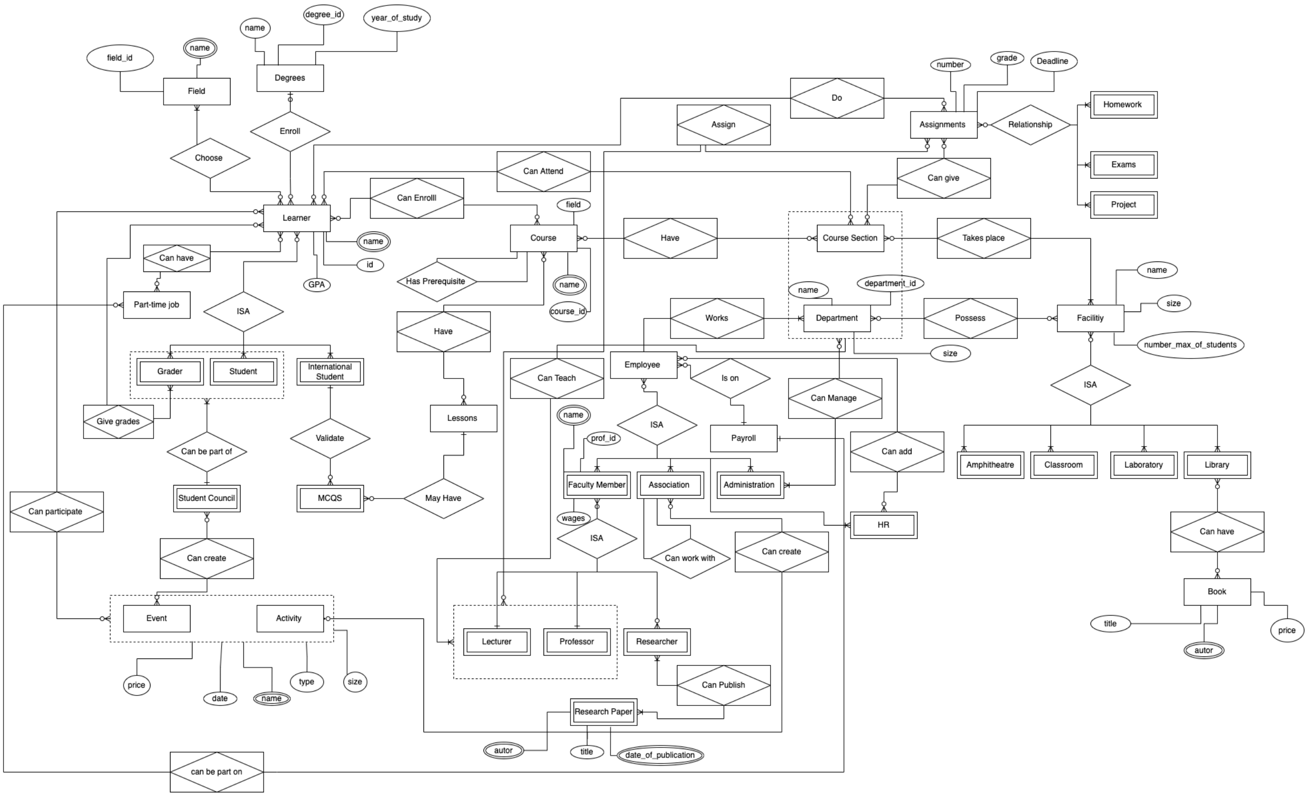
34. Prerequisite (Weak)

34.1. name: composite, alphanumeric
34.2. course_id: key, numeric
34.3. : field: composite, alphanumeric

35. Part-Time Job (Strong)

35.1. role: composite, alphanumeric
35.2. job_id: key, numeric
35.3. : field: composite, alphanumeric
35.4. department: composite, alphanumeric

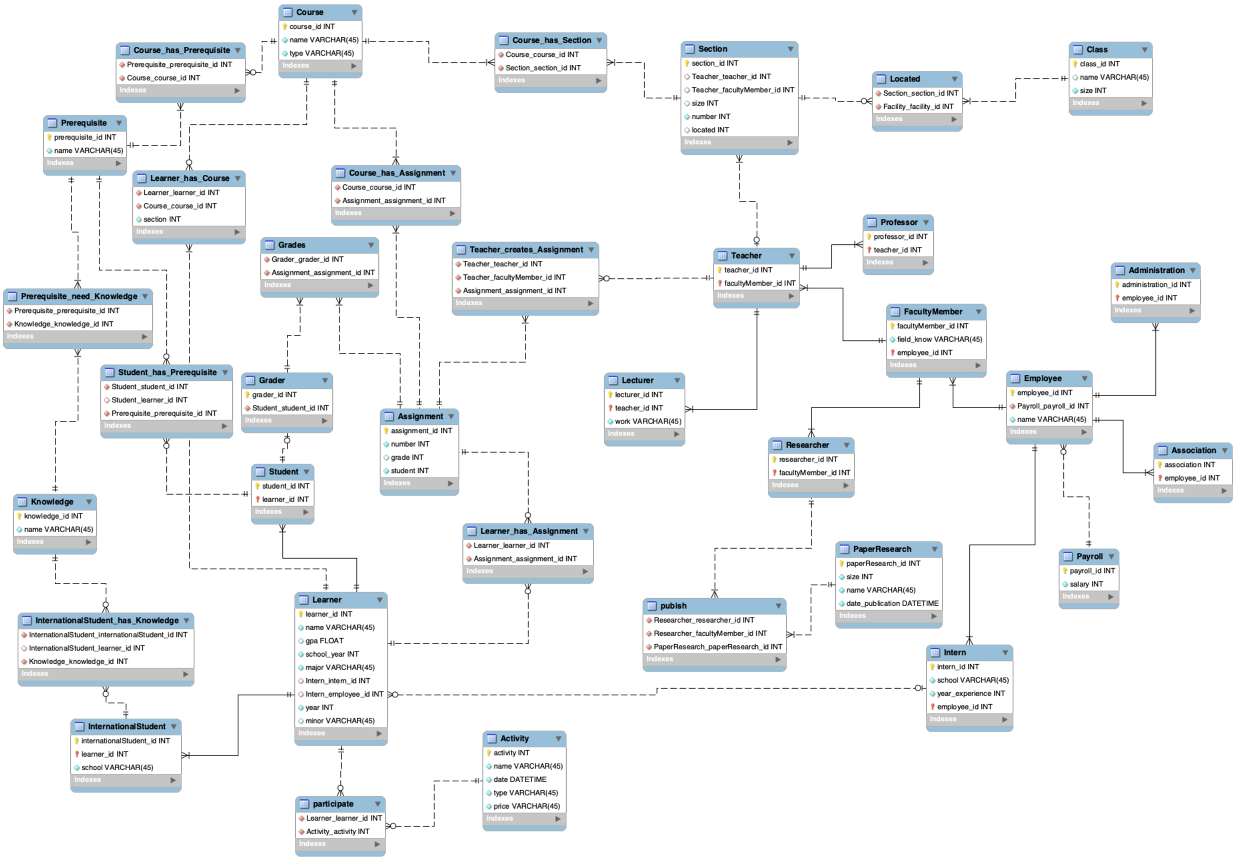
2 Entity Relationship Diagram (ERD)



3 Milestone 2

3.1 Checkpoint 1: Database Model

3.1.1 Database Model/EER



3.1.2 Forward Engineering

3.1.3 Inserting Data

3.2 Checkpoint 2: Database Testing

3.2.1 Business Requirements

Queries in file tests.sql:

1. Find all Students with their courses, its section and their professors
2. Find the number or students that each professors have
3. Find the average grade for each course that have each students

4. Find the overall average grade for each student

5. Show Students with all their assignment in all classes, if they didn?t had an assignment in a class, it shows 'null' and show also the professor

6. Find the average grade for each section

7. Find the average grade for each Professor's students

8. Show the Lecturer and the professor who gives the best grades (average)

9. Find the average grades for each major

10. Find the best student for each courses

11. Find how many activities do each students

12. Which activity did every students and how many times he did the same activity

13. Who graded each Assignment, which grade ? And who has been graded and for which course ?

14. Who graded the most Assignment

15. Find all courses with all their Prerequisites

16. Find all Knowledge that the student need to know to validate Prerequisite

17. How many Knowledge a student need to have to validate a Prerequisite

18. Show all International Students with all their knowledge

19. Show all International Students and all courses that they can take because they have the knowledge

20. Show How many students they are in each section and what is the max capacity of a section

21. Find all Researcher see all their publications

22. Count how many publication did each Researcher

23. Who published the most paper research ?

24. Which student have an Intern ? And in which position ?

Function, Procedure and Triggers in file inserts.sql:

1. Create a procedure where we can Put the International Students id and we'll get courses that he/she can take

2. Create a function to see if they are still place for a student to be in one specific course/section, if the result is 1, it means TRUE, 0 means FALSE

3. Create a Trigger that can insert data inside Learner_has_Assignment after inserting on Assignment

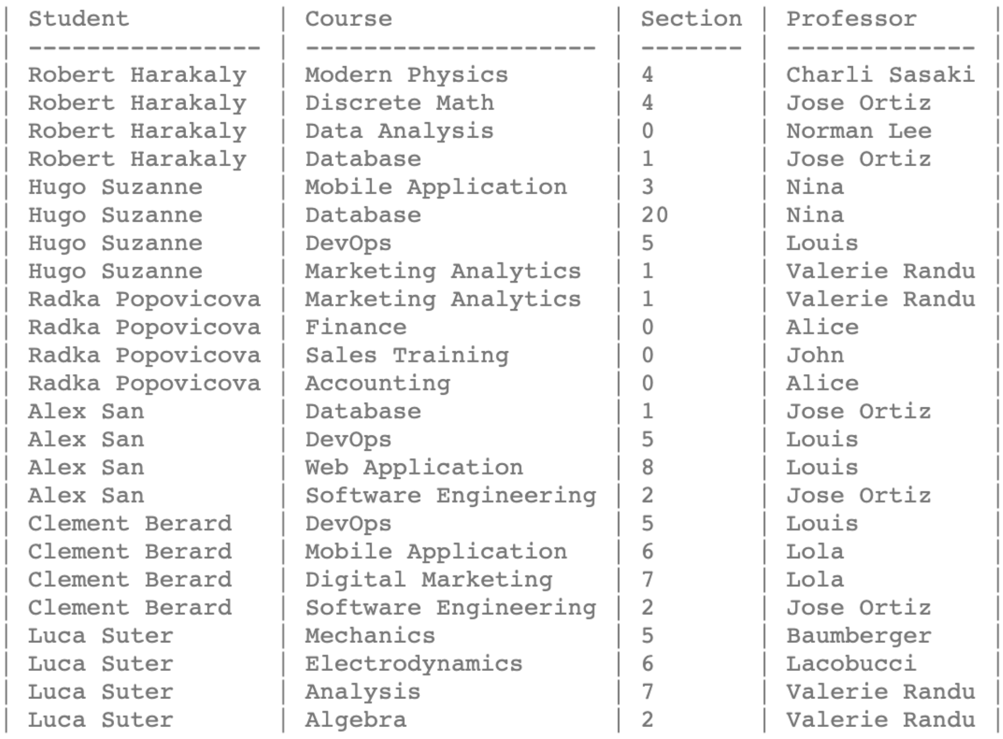
4. Create a Trigger that can insert data inside Prerequisite after inserting on Course

5. Create a Trigger that can update the students gpa each time the grade of his assignment is upgraded
6. Create a Trigger that see the size of a class, this will define the size of the section, so if the class changes, the size of the section also will change
7. Create a Function that returns a Boolean to know if the international student have the knowledge to take one specific course
8. Create a Function that will call the function number 2 and the function number 7 to know if the international student can take and be added in a specific course and section
9. Create a procedure for function 2
10. Create a procedure for function 8

3.2.2 Validating Business Requirements

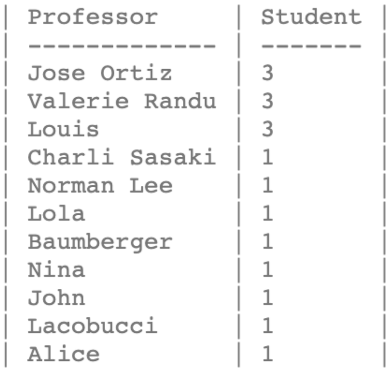
Queries in file tests.sql:

1. Find all Students with their courses, its section and their professors



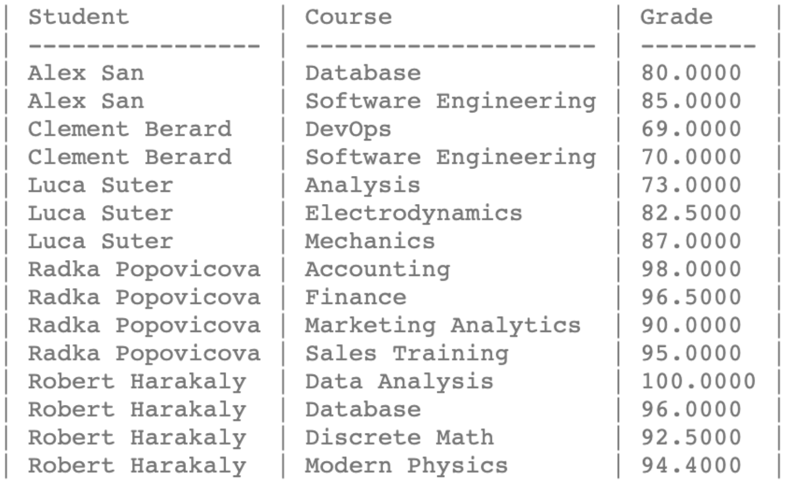
```
| Student           | Course                 | Section | Professor     |
| ----------------  | ---------------------  | ------- | ------------- |
| Robert Harakaly   | Modern Physics         | 4       | Charli Sasaki |
| Robert Harakaly   | Discrete Math          | 4       | Jose Ortiz    |
| Robert Harakaly   | Data Analysis          | 0       | Norman Lee    |
| Robert Harakaly   | Database               | 1       | Jose Ortiz    |
| Hugo Suzanne      | Mobile Application     | 3       | Nina          |
| Hugo Suzanne      | Database               | 20      | Nina          |
| Hugo Suzanne      | DevOps                 | 5       | Louis         |
| Hugo Suzanne      | Marketing Analytics    | 1       | Valerie Randu |
| Radka Popovicova  | Marketing Analytics    | 1       | Valerie Randu |
| Radka Popovicova  | Finance                | 0       | Alice         |
| Radka Popovicova  | Sales Training         | 0       | John          |
| Radka Popovicova  | Accounting             | 0       | Alice         |
| Alex San          | Database               | 1       | Jose Ortiz    |
| Alex San          | DevOps                 | 5       | Louis         |
| Alex San          | Web Application        | 8       | Louis         |
| Alex San          | Software Engineering   | 2       | Jose Ortiz    |
| Clement Berard    | DevOps                 | 5       | Louis         |
| Clement Berard    | Mobile Application     | 6       | Lola          |
| Clement Berard    | Digital Marketing      | 7       | Lola          |
| Clement Berard    | Software Engineering   | 2       | Jose Ortiz    |
| Luca Suter        | Mechanics              | 5       | Baumberger    |
| Luca Suter        | Electrodynamics        | 6       | Lacobucci     |
| Luca Suter        | Analysis               | 7       | Valerie Randu |
| Luca Suter        | Algebra                | 2       | Valerie Randu |
```

2. Find the number or students that each professors have



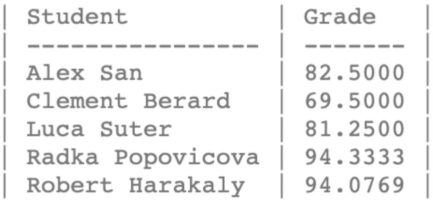
```
| Professor     | Student |
| ------------- | ------- |
| Jose Ortiz    | 3       |
| Valerie Randu | 3       |
| Louis         | 3       |
| Charli Sasaki | 1       |
| Norman Lee    | 1       |
| Lola          | 1       |
| Baumberger    | 1       |
| Nina          | 1       |
| John          | 1       |
| Lacobucci     | 1       |
| Alice         | 1       |
```

3. Find the average grade for each course that have each students



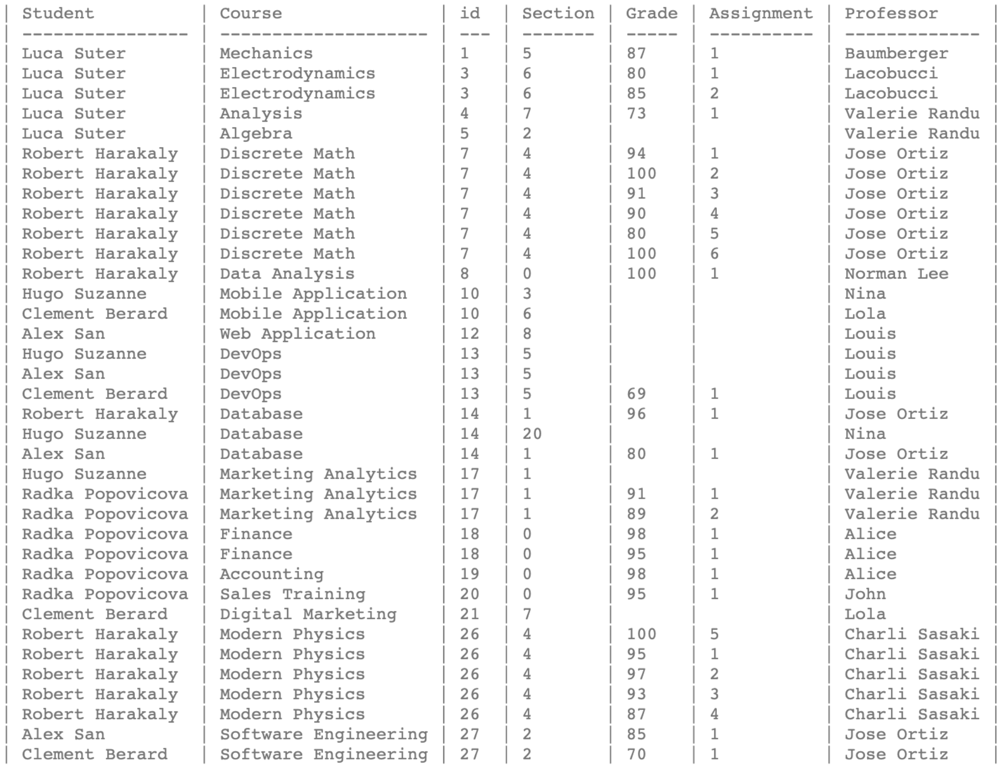
```
| Student          | Course               | Grade    |
| ---------------- | -------------------- | -------- |
| Alex San         | Database             | 80.0000  |
| Alex San         | Software Engineering | 85.0000  |
| Clement Berard   | DevOps               | 69.0000  |
| Clement Berard   | Software Engineering | 70.0000  |
| Luca Suter       | Analysis             | 73.0000  |
| Luca Suter       | Electrodynamics      | 82.5000  |
| Luca Suter       | Mechanics            | 87.0000  |
| Radka Popovicova | Accounting           | 98.0000  |
| Radka Popovicova | Finance              | 96.5000  |
| Radka Popovicova | Marketing Analytics  | 90.0000  |
| Radka Popovicova | Sales Training       | 95.0000  |
| Robert Harakaly  | Data Analysis        | 100.0000 |
| Robert Harakaly  | Database             | 96.0000  |
| Robert Harakaly  | Discrete Math        | 92.5000  |
| Robert Harakaly  | Modern Physics       | 94.4000  |
```

4. Find the overall average grade for each student



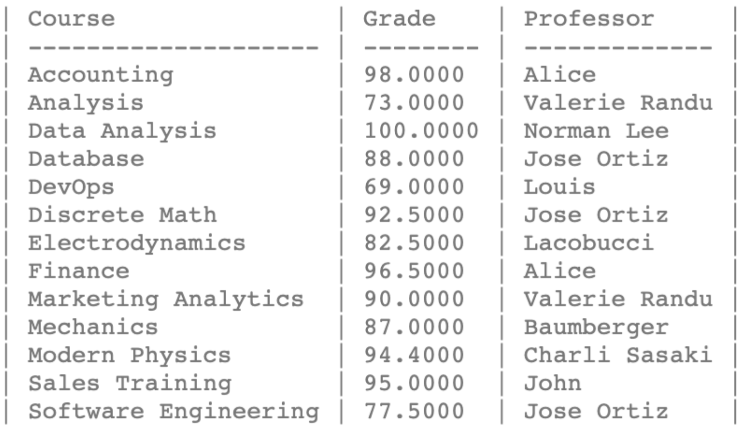
```
| Student          | Grade   |
| ---------------- | ------- |
| Alex San         | 82.5000 |
| Clement Berard   | 69.5000 |
| Luca Suter       | 81.2500 |
| Radka Popovicova | 94.3333 |
| Robert Harakaly  | 94.0769 |
```

5. Show Students with all their assignment in all classes, if they didn't had an assignment in a class, it shows 'null' and show also the professor



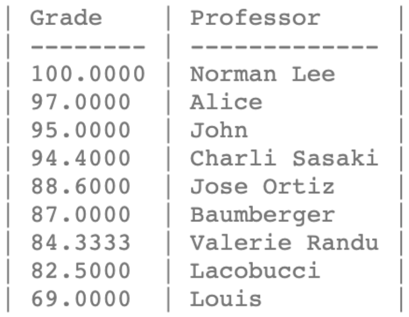
Student	Course	id	Section	Grade	Assignment	Professor
Luca Suter	Mechanics	1	5	87	1	Baumberger
Luca Suter	Electrodynamics	3	6	80	1	Lacobucci
Luca Suter	Electrodynamics	3	6	85	2	Lacobucci
Luca Suter	Analysis	4	7	73	1	Valerie Randu
Luca Suter	Algebra	5	2			Valerie Randu
Robert Harakaly	Discrete Math	7	4	94	1	Jose Ortiz
Robert Harakaly	Discrete Math	7	4	100	2	Jose Ortiz
Robert Harakaly	Discrete Math	7	4	91	3	Jose Ortiz
Robert Harakaly	Discrete Math	7	4	90	4	Jose Ortiz
Robert Harakaly	Discrete Math	7	4	80	5	Jose Ortiz
Robert Harakaly	Discrete Math	7	4	100	6	Jose Ortiz
Robert Harakaly	Data Analysis	8	0	100	1	Norman Lee
Hugo Suzanne	Mobile Application	10	3			Nina
Clement Berard	Mobile Application	10	6			Lola
Alex San	Web Application	12	8			Louis
Hugo Suzanne	DevOps	13	5			Louis
Alex San	DevOps	13	5			Louis
Clement Berard	DevOps	13	5	69	1	Louis
Robert Harakaly	Database	14	1	96	1	Jose Ortiz
Hugo Suzanne	Database	14	20			Nina
Alex San	Database	14	1	80	1	Jose Ortiz
Hugo Suzanne	Marketing Analytics	17	1			Valerie Randu
Radka Popovicova	Marketing Analytics	17	1	91	1	Valerie Randu
Radka Popovicova	Marketing Analytics	17	1	89	2	Valerie Randu
Radka Popovicova	Finance	18	0	98	1	Alice
Radka Popovicova	Finance	18	0	95	1	Alice
Radka Popovicova	Accounting	19	0	98	1	Alice
Radka Popovicova	Sales Training	20	0	95	1	John
Clement Berard	Digital Marketing	21	7			Lola
Robert Harakaly	Modern Physics	26	4	100	5	Charli Sasaki
Robert Harakaly	Modern Physics	26	4	95	1	Charli Sasaki
Robert Harakaly	Modern Physics	26	4	97	2	Charli Sasaki
Robert Harakaly	Modern Physics	26	4	93	3	Charli Sasaki
Robert Harakaly	Modern Physics	26	4	87	4	Charli Sasaki
Alex San	Software Engineering	27	2	85	1	Jose Ortiz
Clement Berard	Software Engineering	27	2	70	1	Jose Ortiz

6. Find the average grade for each section



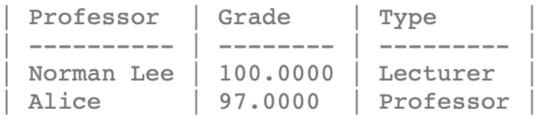
Course	Grade	Professor
Accounting	98.0000	Alice
Analysis	73.0000	Valerie Randu
Data Analysis	100.0000	Norman Lee
Database	88.0000	Jose Ortiz
DevOps	69.0000	Louis
Discrete Math	92.5000	Jose Ortiz
Electrodynamics	82.5000	Lacobucci
Finance	96.5000	Alice
Marketing Analytics	90.0000	Valerie Randu
Mechanics	87.0000	Baumberger
Modern Physics	94.4000	Charli Sasaki
Sales Training	95.0000	John
Software Engineering	77.5000	Jose Ortiz

7. Find the average grade for each Professor's students



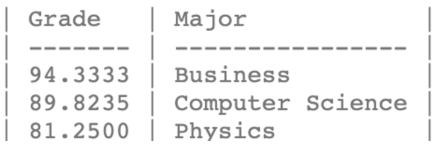
```
| Grade    | Professor     |
| -------- | ------------- |
| 100.0000 | Norman Lee    |
| 97.0000  | Alice         |
| 95.0000  | John          |
| 94.4000  | Charli Sasaki |
| 88.6000  | Jose Ortiz    |
| 87.0000  | Baumberger    |
| 84.3333  | Valerie Randu |
| 82.5000  | Lacobucci     |
| 69.0000  | Louis         |
```

8. Show the Lecturer and the professor who gives the best grades (average)



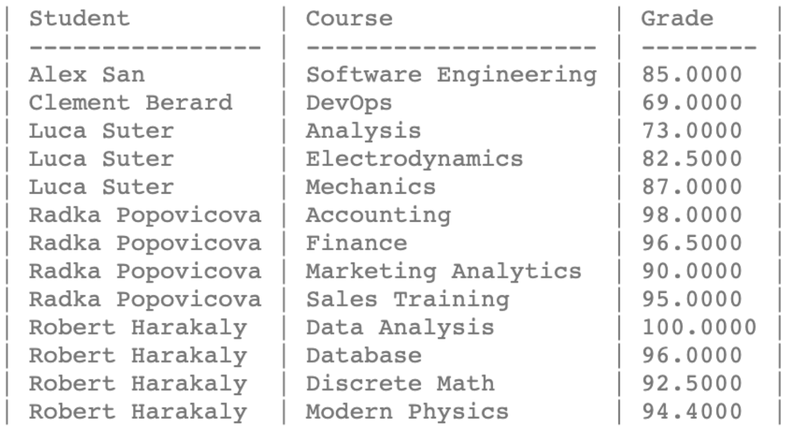
```
| Professor  | Grade    | Type      |
| ---------- | -------- | --------- |
| Norman Lee | 100.0000 | Lecturer  |
| Alice      | 97.0000  | Professor |
```

9. Find the average grades for each major



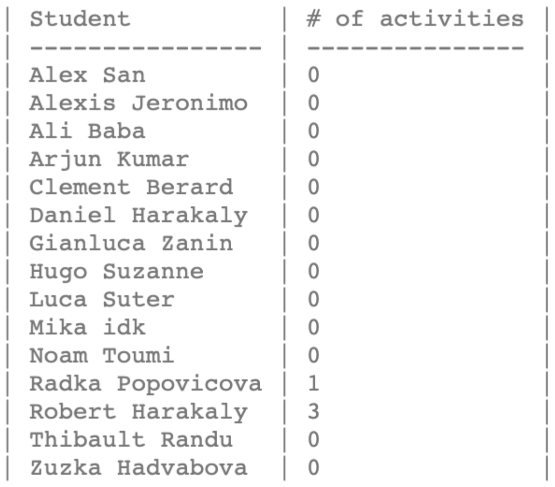
```
| Grade   | Major            |
| ------- | ---------------- |
| 94.3333 | Business         |
| 89.8235 | Computer Science |
| 81.2500 | Physics          |
```

10. Find the best student for each courses



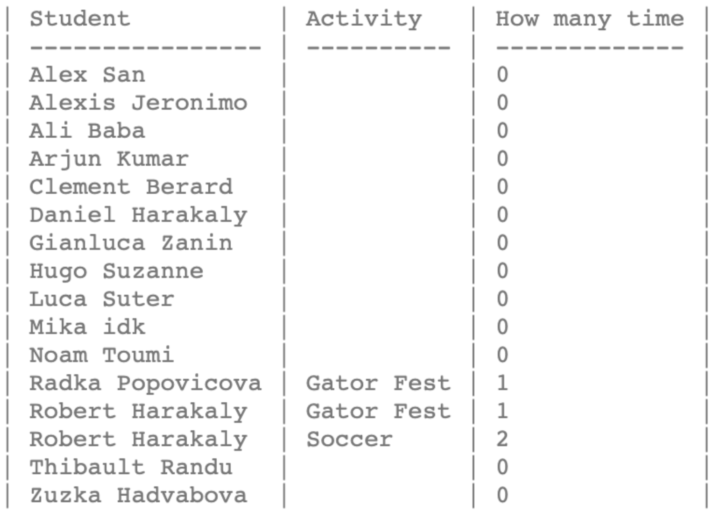
```
| Student           | Course               | Grade    |
| ----------------- | -------------------- | -------- |
| Alex San          | Software Engineering | 85.0000  |
| Clement Berard    | DevOps               | 69.0000  |
| Luca Suter        | Analysis             | 73.0000  |
| Luca Suter        | Electrodynamics      | 82.5000  |
| Luca Suter        | Mechanics            | 87.0000  |
| Radka Popovicova  | Accounting           | 98.0000  |
| Radka Popovicova  | Finance              | 96.5000  |
| Radka Popovicova  | Marketing Analytics  | 90.0000  |
| Radka Popovicova  | Sales Training       | 95.0000  |
| Robert Harakaly   | Data Analysis        | 100.0000 |
| Robert Harakaly   | Database             | 96.0000  |
| Robert Harakaly   | Discrete Math        | 92.5000  |
| Robert Harakaly   | Modern Physics       | 94.4000  |
```

11. Find how many activities do each students



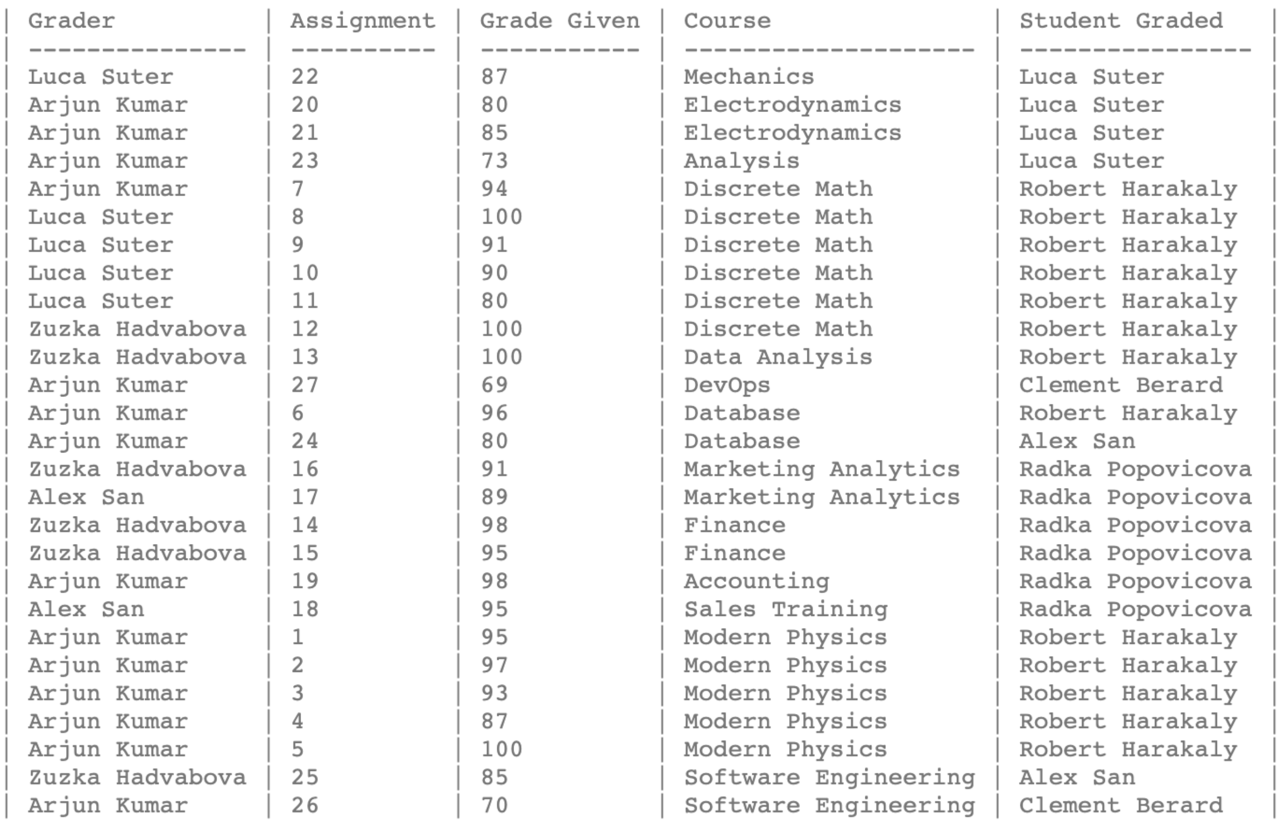
```
| Student          | # of activities |
| ---------------- | --------------- |
| Alex San         | 0               |
| Alexis Jeronimo  | 0               |
| Ali Baba         | 0               |
| Arjun Kumar      | 0               |
| Clement Berard   | 0               |
| Daniel Harakaly  | 0               |
| Gianluca Zanin   | 0               |
| Hugo Suzanne     | 0               |
| Luca Suter       | 0               |
| Mika idk         | 0               |
| Noam Toumi       | 0               |
| Radka Popovicova | 1               |
| Robert Harakaly  | 3               |
| Thibault Randu   | 0               |
| Zuzka Hadvabova  | 0               |
```

12. Which activity did every students and how many times he did the same activity



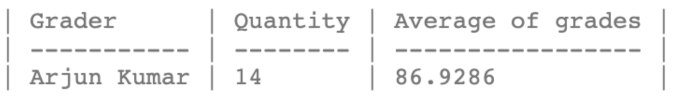
```
| Student          | Activity   | How many time |
| ---------------- | ---------- | ------------- |
| Alex San         |            | 0             |
| Alexis Jeronimo  |            | 0             |
| Ali Baba         |            | 0             |
| Arjun Kumar      |            | 0             |
| Clement Berard   |            | 0             |
| Daniel Harakaly  |            | 0             |
| Gianluca Zanin   |            | 0             |
| Hugo Suzanne     |            | 0             |
| Luca Suter       |            | 0             |
| Mika idk         |            | 0             |
| Noam Toumi       |            | 0             |
| Radka Popovicova | Gator Fest | 1             |
| Robert Harakaly  | Gator Fest | 1             |
| Robert Harakaly  | Soccer     | 2             |
| Thibault Randu   |            | 0             |
| Zuzka Hadvabova  |            | 0             |
```

13. Who graded each Assignment, which grade ? And who has been graded and for which course ?



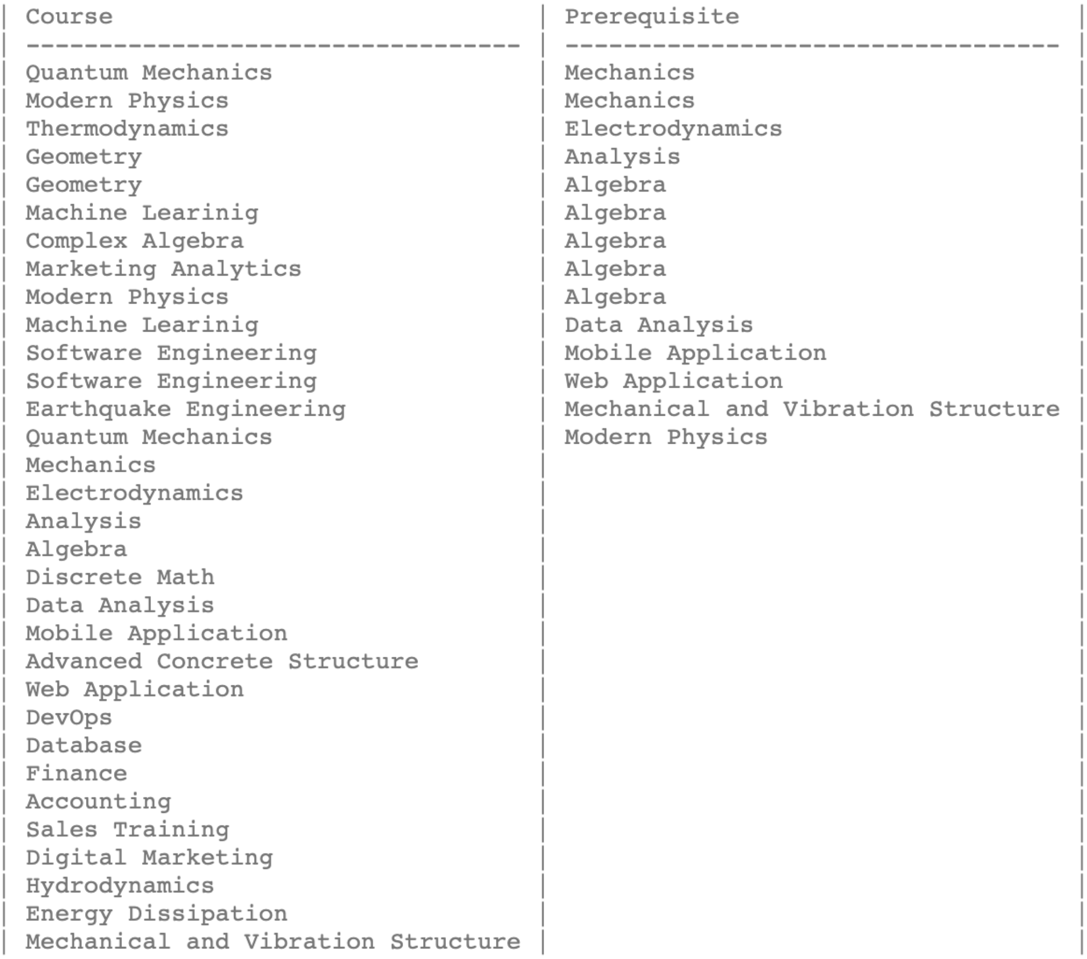
```
| Grader           | Assignment | Grade Given | Course                | Student Graded   |
| ---------------- | ---------- | ----------- | --------------------- | ---------------- |
| Luca Suter       | 22         | 87          | Mechanics             | Luca Suter       |
| Arjun Kumar      | 20         | 80          | Electrodynamics       | Luca Suter       |
| Arjun Kumar      | 21         | 85          | Electrodynamics       | Luca Suter       |
| Arjun Kumar      | 23         | 73          | Analysis              | Luca Suter       |
| Arjun Kumar      | 7          | 94          | Discrete Math         | Robert Harakaly  |
| Luca Suter       | 8          | 100         | Discrete Math         | Robert Harakaly  |
| Luca Suter       | 9          | 91          | Discrete Math         | Robert Harakaly  |
| Luca Suter       | 10         | 90          | Discrete Math         | Robert Harakaly  |
| Luca Suter       | 11         | 80          | Discrete Math         | Robert Harakaly  |
| Zuzka Hadvabova  | 12         | 100         | Discrete Math         | Robert Harakaly  |
| Zuzka Hadvabova  | 13         | 100         | Data Analysis         | Robert Harakaly  |
| Arjun Kumar      | 27         | 69          | DevOps                | Clement Berard   |
| Arjun Kumar      | 6          | 96          | Database              | Robert Harakaly  |
| Arjun Kumar      | 24         | 80          | Database              | Alex San         |
| Zuzka Hadvabova  | 16         | 91          | Marketing Analytics   | Radka Popovicova |
| Alex San         | 17         | 89          | Marketing Analytics   | Radka Popovicova |
| Zuzka Hadvabova  | 14         | 98          | Finance               | Radka Popovicova |
| Zuzka Hadvabova  | 15         | 95          | Finance               | Radka Popovicova |
| Arjun Kumar      | 19         | 98          | Accounting            | Radka Popovicova |
| Alex San         | 18         | 95          | Sales Training        | Radka Popovicova |
| Arjun Kumar      | 1          | 95          | Modern Physics        | Robert Harakaly  |
| Arjun Kumar      | 2          | 97          | Modern Physics        | Robert Harakaly  |
| Arjun Kumar      | 3          | 93          | Modern Physics        | Robert Harakaly  |
| Arjun Kumar      | 4          | 87          | Modern Physics        | Robert Harakaly  |
| Arjun Kumar      | 5          | 100         | Modern Physics        | Robert Harakaly  |
| Zuzka Hadvabova  | 25         | 85          | Software Engineering  | Alex San         |
| Arjun Kumar      | 26         | 70          | Software Engineering  | Clement Berard   |
```

14. Who graded the most Assignment



```
| Grader      | Quantity | Average of grades |
| ----------- | -------- | ----------------- |
| Arjun Kumar | 14       | 86.9286           |
```

15. Find all courses with all their Prerequisites



```
| Course                             | Prerequisite                       |
| ---------------------------------- | ---------------------------------- |
| Quantum Mechanics                  | Mechanics                          |
| Modern Physics                     | Mechanics                          |
| Thermodynamics                     | Electrodynamics                    |
| Geometry                           | Analysis                           |
| Geometry                           | Algebra                            |
| Machine Learinig                   | Algebra                            |
| Complex Algebra                    | Algebra                            |
| Marketing Analytics                | Algebra                            |
| Modern Physics                     | Algebra                            |
| Machine Learinig                   | Data Analysis                      |
| Software Engineering               | Mobile Application                 |
| Software Engineering               | Web Application                    |
| Earthquake Engineering             | Mechanical and Vibration Structure |
| Quantum Mechanics                  | Modern Physics                     |
| Mechanics                          |                                    |
| Electrodynamics                    |                                    |
| Analysis                           |                                    |
| Algebra                            |                                    |
| Discrete Math                      |                                    |
| Data Analysis                      |                                    |
| Mobile Application                 |                                    |
| Advanced Concrete Structure        |                                    |
| Web Application                    |                                    |
| DevOps                             |                                    |
| Database                           |                                    |
| Finance                            |                                    |
| Accounting                         |                                    |
| Sales Training                     |                                    |
| Digital Marketing                  |                                    |
| Hydrodynamics                      |                                    |
| Energy Dissipation                 |                                    |
| Mechanical and Vibration Structure |                                    |
```

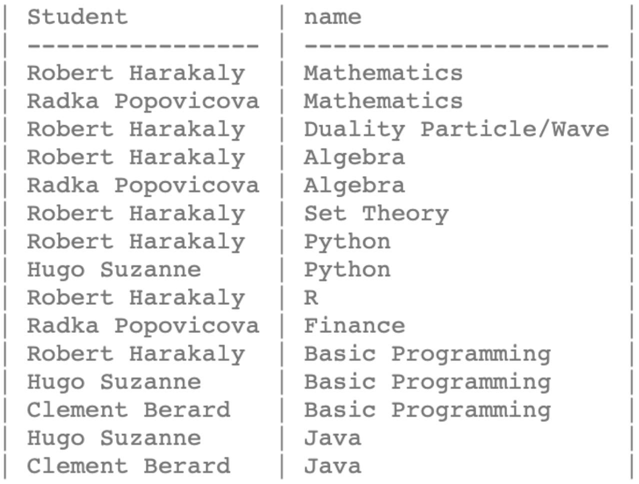
16. Find all Knowledge that the student need to know to validate Prerequisite

```
| Prerequisite                       | The International Student need to know |
| ---------------------------------- | -------------------------------------- |
| Machine Learinig                   | Mathematics                            |
| Machine Learinig                   | Algebra                                |
| Machine Learinig                   | Set Theory                             |
| Machine Learinig                   | Python                                 |
| Machine Learinig                   | R                                      |
| Machine Learinig                   | Basic Programming                      |
| Mobile Application                 | Basic Programming                      |
| Advanced Concrete Structure        | Basic Engineering                      |
| Web Application                    | Basic Programming                      |
| DevOps                             | Basic Programming                      |
| Database                           | Basic Programming                      |
| Complex Algebra                    | Mathematics                            |
| Complex Algebra                    | Advanced Mathematics                   |
| Quantum Mechanics                  | Mathematics                            |
| Quantum Mechanics                  | Advancced Physics                      |
| Quantum Mechanics                  | Duality Particle/Wave                  |
| Marketing Analytics                | Mathematics                            |
| Marketing Analytics                | Finance                                |
| Accounting                         | Finance                                |
| Sales Training                     | Finance                                |
| Digital Marketing                  | Finance                                |
| Digital Marketing                  | Python                                 |
| Hydrodynamics                      | Vibration                              |
| Energy Dissipation                 | Basic Engineering                      |
| Energy Dissipation                 | Basic Engineering                      |
| Mechanical and Vibration Structure | Basic Engineering                      |
| Software Engineering               | Basic Programming                      |
| Software Engineering               | Java                                   |
```

17. How many Knowledge a student need to have to validate a Prerequisite

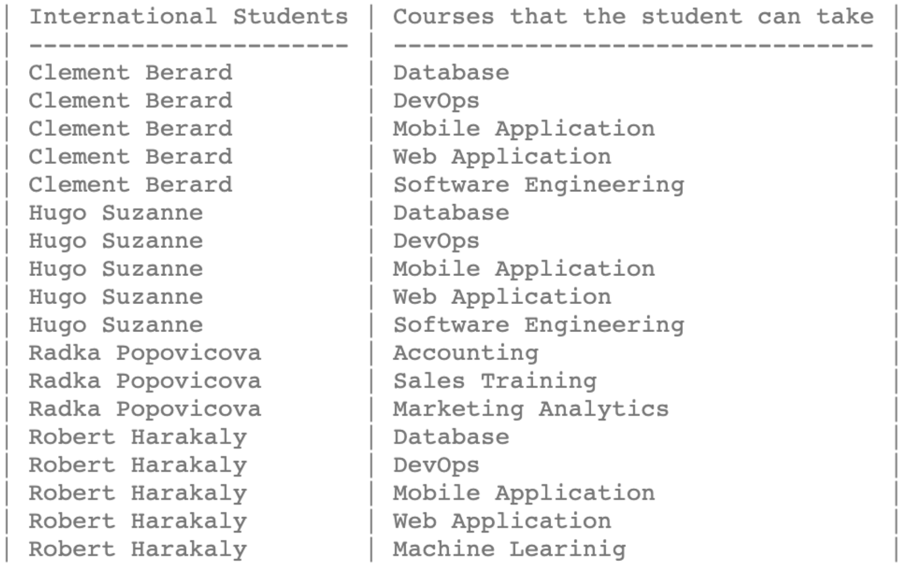
```
| Prerequisite                       | # Courses to know |
| ---------------------------------- | ----------------- |
| Accounting                         | 1                 |
| Advanced Concrete Structure        | 1                 |
| Complex Algebra                    | 2                 |
| Database                           | 1                 |
| DevOps                             | 1                 |
| Digital Marketing                  | 2                 |
| Energy Dissipation                 | 2                 |
| Hydrodynamics                      | 1                 |
| Machine Learinig                   | 6                 |
| Marketing Analytics                | 2                 |
| Mechanical and Vibration Structure | 1                 |
| Mobile Application                 | 1                 |
| Quantum Mechanics                  | 3                 |
| Sales Training                     | 1                 |
| Software Engineering               | 2                 |
| Web Application                    | 1                 |
```

18. Show all International Students with all their knowledge



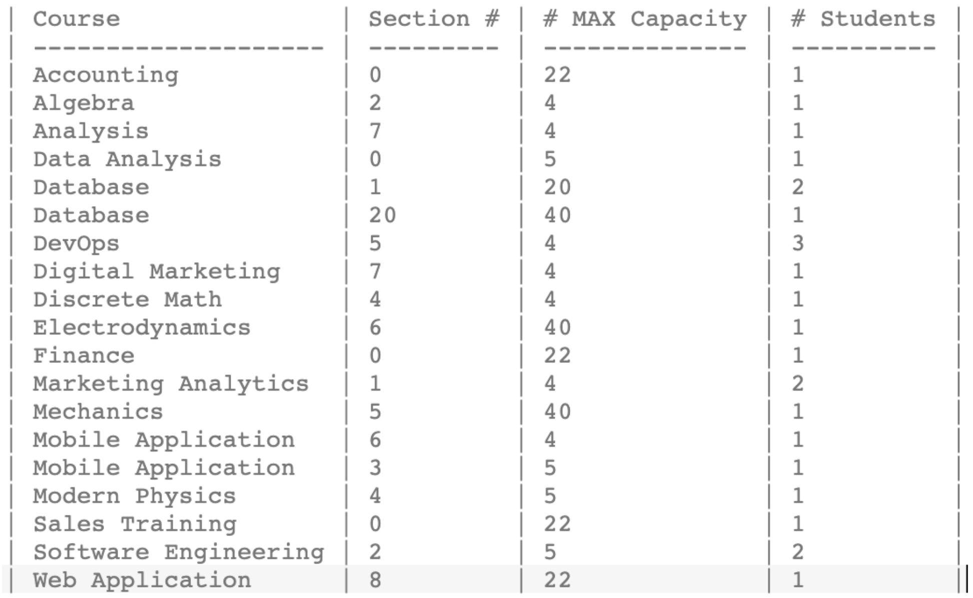
```
| Student          | name                  |
| ---------------- | --------------------- |
| Robert Harakaly  | Mathematics           |
| Radka Popovicova | Mathematics           |
| Robert Harakaly  | Duality Particle/Wave |
| Robert Harakaly  | Algebra               |
| Radka Popovicova | Algebra               |
| Robert Harakaly  | Set Theory            |
| Robert Harakaly  | Python                |
| Hugo Suzanne     | Python                |
| Robert Harakaly  | R                     |
| Radka Popovicova | Finance               |
| Robert Harakaly  | Basic Programming     |
| Hugo Suzanne     | Basic Programming     |
| Clement Berard   | Basic Programming     |
| Hugo Suzanne     | Java                  |
| Clement Berard   | Java                  |
```

19. Show all International Students and all courses that they can take because they have the knowledge



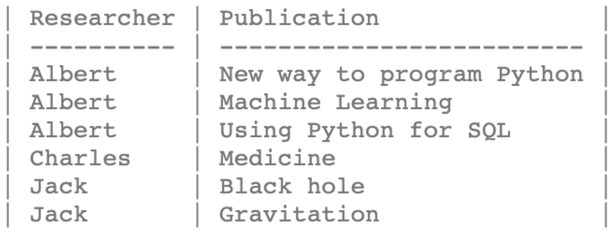
```
| International Students | Courses that the student can take |
| ---------------------- | --------------------------------- |
| Clement Berard         | Database                          |
| Clement Berard         | DevOps                            |
| Clement Berard         | Mobile Application                |
| Clement Berard         | Web Application                   |
| Clement Berard         | Software Engineering              |
| Hugo Suzanne           | Database                          |
| Hugo Suzanne           | DevOps                            |
| Hugo Suzanne           | Mobile Application                |
| Hugo Suzanne           | Web Application                   |
| Hugo Suzanne           | Software Engineering              |
| Radka Popovicova       | Accounting                        |
| Radka Popovicova       | Sales Training                    |
| Radka Popovicova       | Marketing Analytics               |
| Robert Harakaly        | Database                          |
| Robert Harakaly        | DevOps                            |
| Robert Harakaly        | Mobile Application                |
| Robert Harakaly        | Web Application                   |
| Robert Harakaly        | Machine Learinig                  |
```

20. Show How many students they are in each section and what is the max capacity of a section



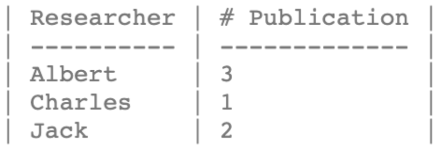
Course	Section #	# MAX Capacity	# Students
Accounting	0	22	1
Algebra	2	4	1
Analysis	7	4	1
Data Analysis	0	5	1
Database	1	20	2
Database	20	40	1
DevOps	5	4	3
Digital Marketing	7	4	1
Discrete Math	4	4	1
Electrodynamics	6	40	1
Finance	0	22	1
Marketing Analytics	1	4	2
Mechanics	5	40	1
Mobile Application	6	4	1
Mobile Application	3	5	1
Modern Physics	4	5	1
Sales Training	0	22	1
Software Engineering	2	5	2
Web Application	8	22	1

21. Find all Researcher see all their publications



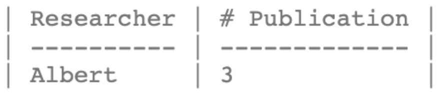
Researcher	Publication
Albert	New way to program Python
Albert	Machine Learning
Albert	Using Python for SQL
Charles	Medicine
Jack	Black hole
Jack	Gravitation

22. Count how many publication did each Researcher



Researcher	# Publication
Albert	3
Charles	1
Jack	2

23. Who published the most paper research ?



Researcher	# Publication
Albert	3

24. Which student have an Intern ? And in which position ?

```
| Student         | Position name      |
| --------------- | ------------------ |
| Zuzka Hadvabova | Nurse              |
| Ali Baba        | Shop               |
| Arjun Kumar     | Teacher Assistant  |
```

Function, Procedure and Triggers in file inserts.sql:

1. Create a procedure where we can Put the International Students id and we'll get courses that he/she can take

```
| International Students | Courses that the student can take |
| ---------------------- | --------------------------------- |
| Hugo Suzanne           | Database                          |
| Hugo Suzanne           | DevOps                            |
| Hugo Suzanne           | Mobile Application                |
| Hugo Suzanne           | Web Application                   |
| Hugo Suzanne           | Software Engineering              |
```

2. Create a function to see if they are still place for a student to be in one specific course/section, if the result is 1, it means TRUE, 0 means FALSE
   In this example, I created a query that call the function, we want to know iff the international student number 1 can take the course number 1 section number 5. The third column show us if this student can take this course.

```
| Student         | Course | Section | Boolean if there are some place to take this course |
| --------------- | ------ | ------- | --------------------------------------------------- |
| Robert Harakaly | DevOps | 5       | 1                                                   |
```

3. Create a Trigger that can insert data inside Learner_has_Assignment after inserting on Assignment

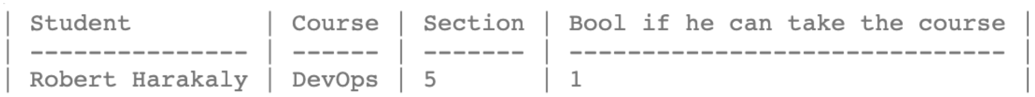
4. Create a Trigger that can insert data inside Prerequisite after inserting on Course

5. Create a Trigger that can update the students gpa each time the grade of his assignment is upgraded

6. Create a Trigger that see the size of a class, this will define the size of the section, so if the class changes, the size of the section also will change

7. Create a Function that returns a Boolean to know if the international student have the knowledge to take one specific course

8. Create a Function that will call the function number 2 and the function number 7 to know if the international student can take and be added in a specific course and section



Student	Course	Section	Bool if he can take the course
Robert Harakaly	DevOps	5	1