MCIS 5133\_030201S

Data Base Management Systems

Online Assignment Submission System

Instructor: AHMAD AL-SHAMI

Megha Baral

Ravindra Kotegar Sheruga

Sharmila Kurrar

November 12, 2020

**Academic Honesty Statement**

**Course Code:** MCIS 5133

**Project:** Online Assignment Submission System (OASS) Project

**Email ID:** [mbaral1096@muleriders.saumag.edu](mailto:mbaral1096@muleriders.saumag.edu),

[RKotegarSherugar2030@muleriders.saumag.edu](mailto:RKotegarSherugar2030@muleriders.saumag.edu),

[skurra2019@muleriders.saumag.edu](mailto:skurra2019@muleriders.saumag.edu).

**Declaration of Academic Honesty**

* Our team understand that this is a piece of individual coursework. Our team confirm that:
* The solutions are Our team own work. Our team have not collaborated with other students or anyone else in coming up with the solutions.
* Our team have not discussed any details of this solution to the coursework with other students or anyone else.
  + Discussions of general concepts related to the coursework are fine, but not any details relating to specific coursework questions. For example - studying the notes or slides of lectures relating to the module together is acceptable but talking about the homework questions is not.
* Our team have not shown any parts of my solution to other students or anyone else.
* Our team have not looked at the solutions written by any other students or by anyone else.
* Our team have appropriately cited any references (Internet, books, source code etc.) that Our team have used to produce my solutions or any parts of them.
* The penalties for plagiarism are serious — the MSc course Handbooks and the Department's "For Campus" site contain further details. Our team am aware that any violation of the above would constitute plagiarism and carry serious penalties.

**Date:** 11/12/2020

**Signature:**

Megha Baral

Ravindra Kotegar Sherugar

Sharmila Kurra

Table of Contents

[Chapter 1: Introduction to Business Case: 1](#_Toc55996928)

[Chapter 2: Screenshots of the Relational Model 2](#_Toc55996929)

[Chapter 3: Queries 3](#_Toc55996930)

[Chapter 4: Views 9](#_Toc55996931)

[Chapter 5: Query optimization 12](#_Toc55996932)

[5.1 Index: 14](#_Toc55996933)

[Chapter 6: Creativity 15](#_Toc55996934)

[Chapter 7: Conclusion 20](#_Toc55996935)

[Reference: 22](#_Toc55996936)

# Chapter 1: Introduction to Business Case

Online assignment submission system (OASS) is web-based technology that allows teachers and students to communicate with each other about the assignment and other academic activities more quickly and efficiently. Students can check all the modules, announcements about exam and assignment through this technology at the same time teacher can the check the assignments and exam and give points and feedback to the student.

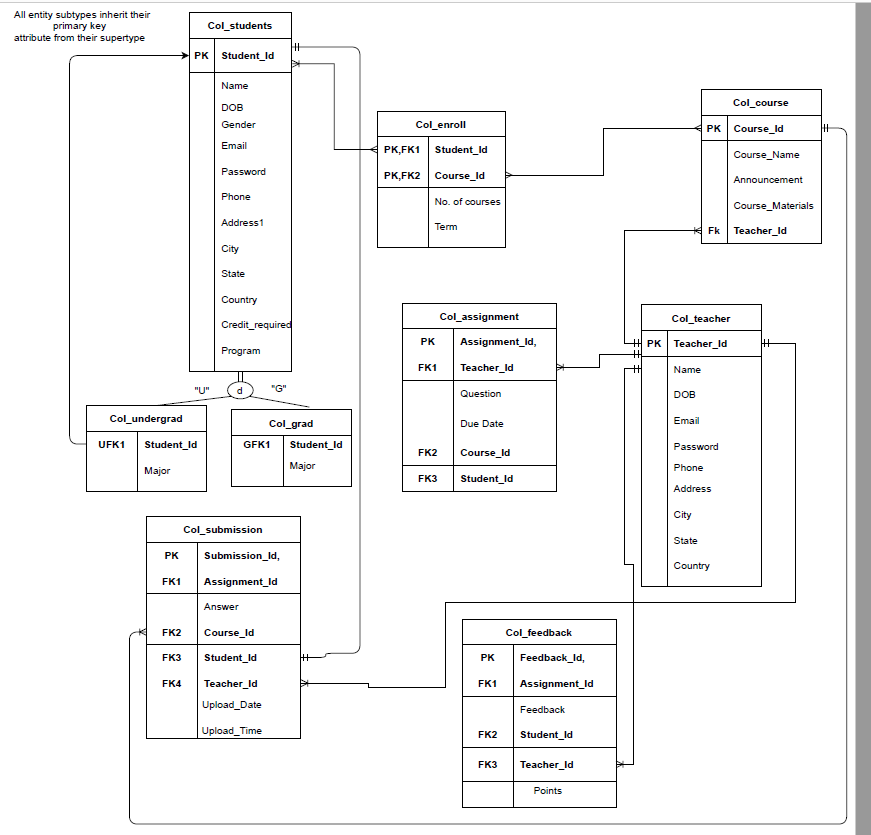
The main goal of the system is to minimize the work of lecturers by having specific time to submit tasks and grades. The project provides a simple and convenient forum for students and lecturers to share. Being an online program, it can be signed in with internet access from anywhere. Any student will obtain feedback from the faculty based on his assignment status. In order to view his test results, the student would have limited access to the faculty. Using this method, faculty will have more time to view the assignments and projects of students so that before deadline is reached, he can provide feedback to students. Students may adjust assignments and make and resubmit changes. This approach saves time during classes and working hours for lectures.

Assignment is another way of measuring the student's research and success level approved by many universities around the world during the semester test. But it is clearly boring and time consuming to manually send and collect assignments. The manual method wastes both the sender and the receiver's time. In order to solve the current problem in the assessment system, a web-based system is required to manipulate assignments.

The scope of OASS project are:

* It is a web-based online application.
* The project makes contact between lecturers and students simple and relaxed.
* The required suggestion and feedback can be given to students easily and efficiently.
* It offers the opportunity for the submitted assignment to be updated before the deadline.
* The introduction of the project simplifies the method of assessing universities.
* It will minimize the time for appointments and submission.

# Chapter 2: Screenshots of the Relational Model



# Chapter 3: Queries

The following queries are used to determine different scenarios by using conditions applied by more than one column and various operator and aliases, Aggregate functions, Joins and finally Group by and Having as well.

**1.Write a sql query to find the count of number of MALE and FEMALE STUDENTS**

SELECT Gender,

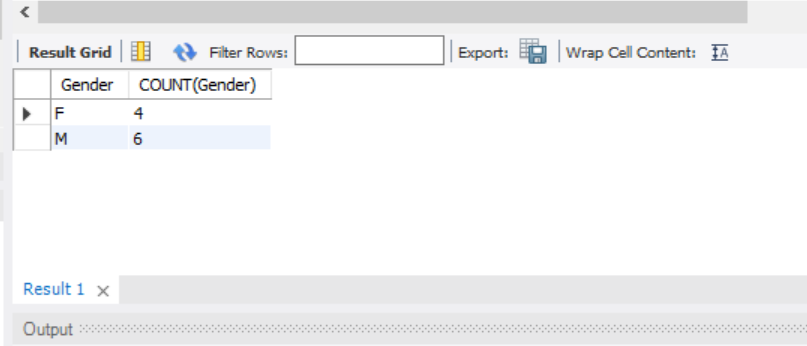
COUNT(Gender)

from Col\_students

group by Gender;

**Explanation:** This query is count number of male and female students.

**Result:**



**2. Write a sql query to find Course\_name and term where no.of courses is less than 5.**

SELECT Col\_course.Course\_Name, Num\_of\_Courses, Col\_enroll.Term

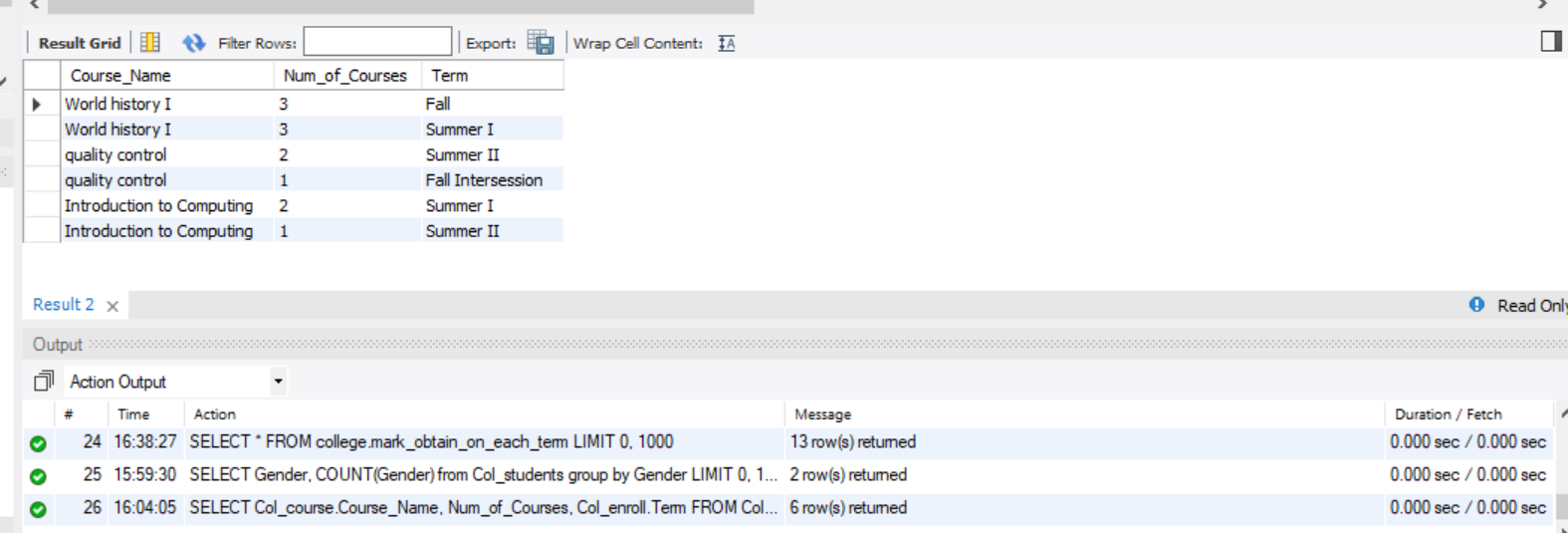
FROM Col\_course

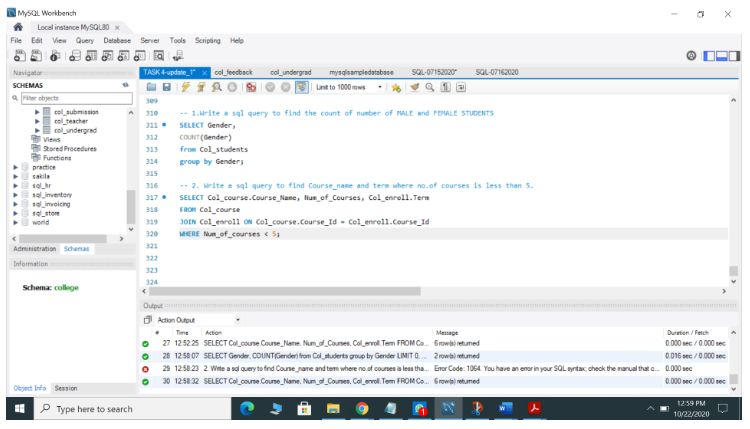
JOIN Col\_enroll ON Col\_course.Course\_Id = Col\_enroll.Course\_Id

WHERE Num\_of\_courses < 5;

**Explanation:** This query explain number of courses student are taking in each term with the name of courses.

**Result:**





**3. Write a query to update a record where teacher id is 001 and teacher id is 008**

UPDATE Col\_assignment

SET Question = 'Describe about dbms'

WHERE Teacher\_Id = 001 ;

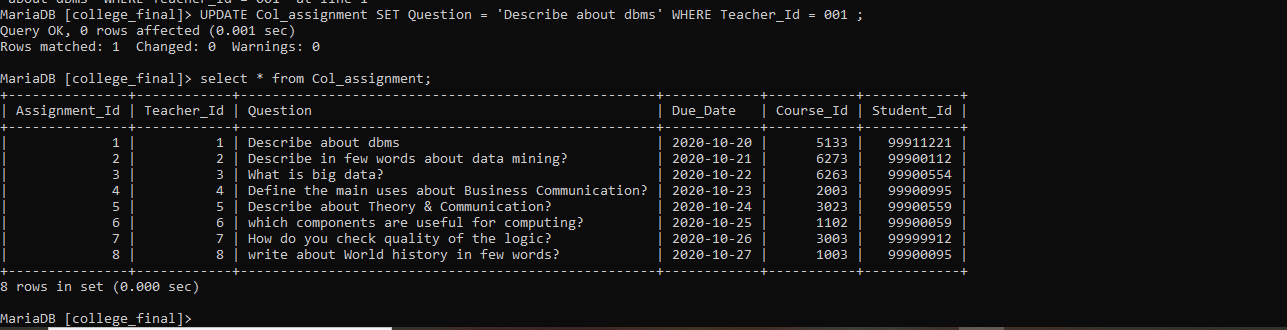
UPDATE Col\_teacher

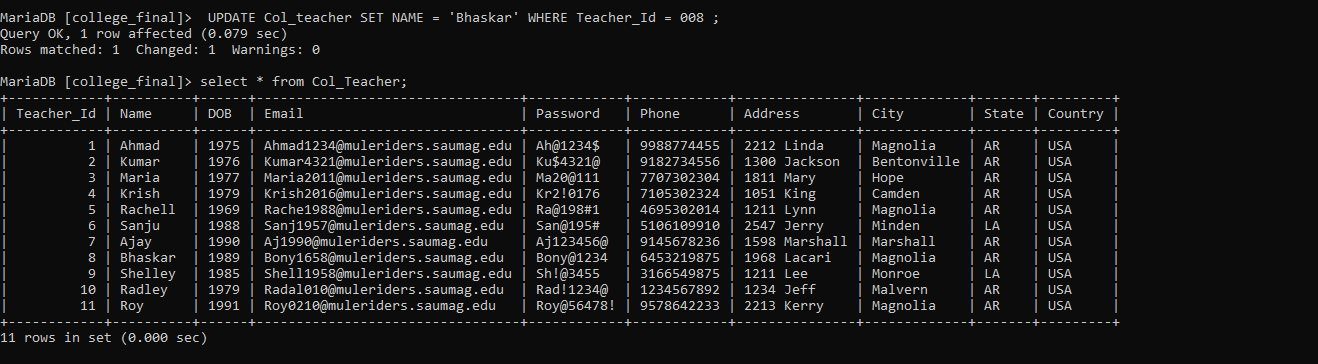
SET NAME = 'Bhaskar'

WHERE Teacher\_Id = 008 ;

**Explanation:** Query 3 returns output as updating the records .For example Question, Name values are updated in teacher table based on Teacher Id.

**Result:**





**4.Write a query to find teacher name and course name and assignment id where the results are sorted in ascending order of teacher name.**

SELECT Name,Col\_course.Course\_Name,Col\_assignment.Assignment\_Id

From Col\_teacher

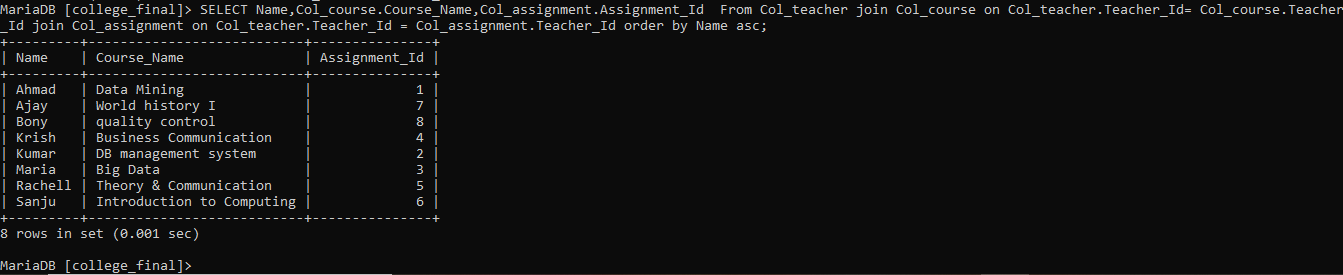
join Col\_course on Col\_teacher.Teacher\_Id= Col\_course.Teacher\_Id

join Col\_assignment on Col\_teacher.Teacher\_Id = Col\_assignment.Teacher\_Id

order by Name asc;

**Explanation:** Query 4 returns output in ascending order of Teacher name. Along with Teacher name ,Course name Assignment id values also present in the output. Applied join condition on teacher, course, assignment tables based on teacher id.

**Result:**



**5.Write a query to create alias name for question ,course name as course and announcement as Assign\_detail ,also results should sorted by descending order of assignment id**

SELECT Assignment\_Id,Col\_course.Course\_Name as 'Course',Question,Col\_course.Announcements as 'Assign\_detail'

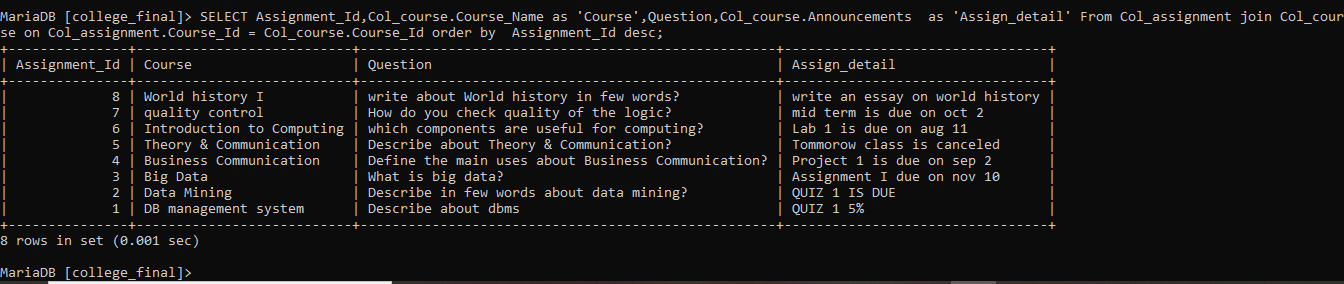
From Col\_assignment

join Col\_course on Col\_assignment.Course\_Id = Col\_course.Course\_Id

order by Assignment\_Id desc;

**Explanation:** Query 5 returns output in descending order of assignment id. Along with creating aliases for columns coursename, announcements by applying join condition between assignment and course table based on Course id.

**Result:**



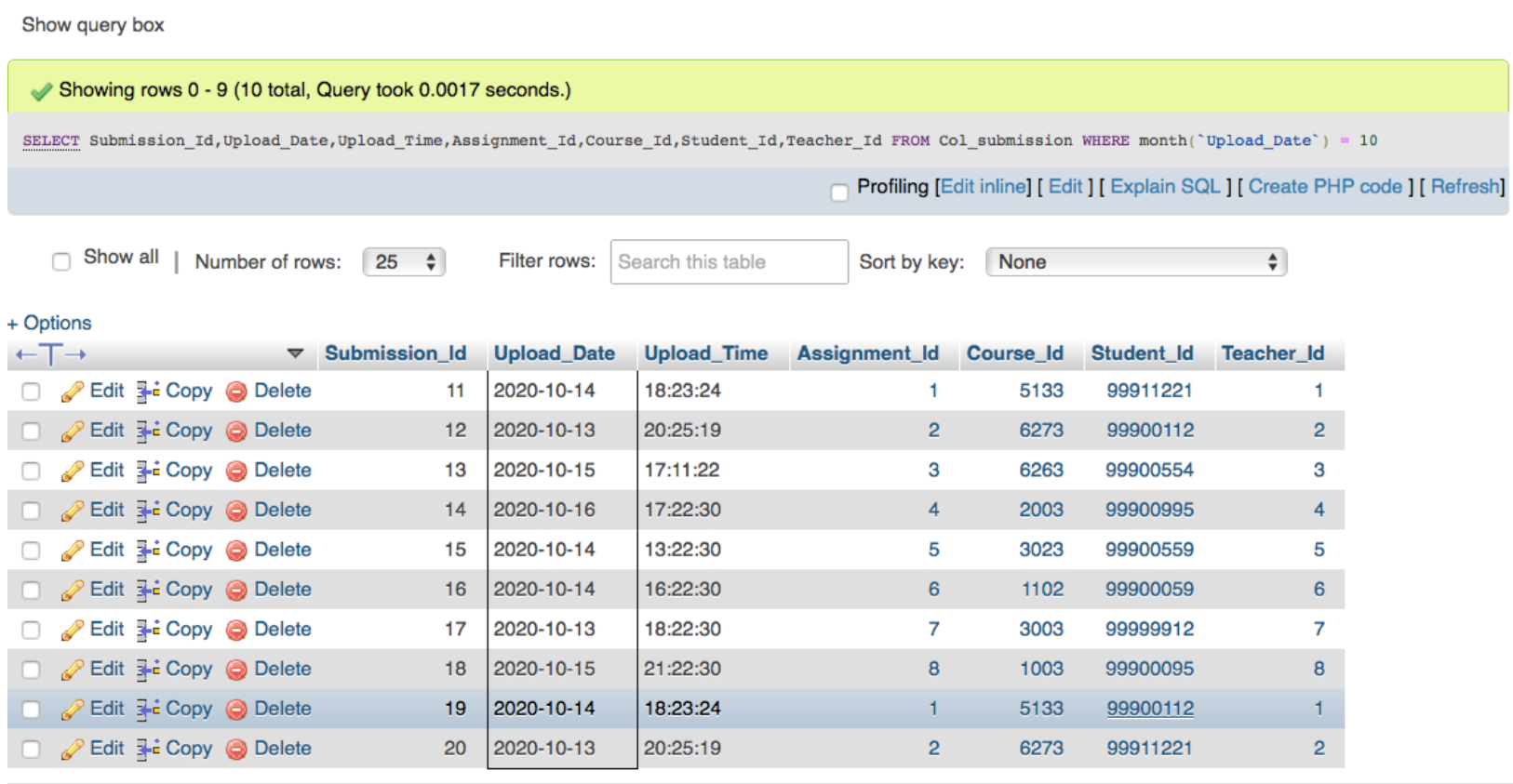
**6.Write a query to find assignment uploaded in this 10th month.**

SELECT Submission\_Id,Upload\_Date,Upload\_Time,Assignment\_Id,Course\_Id,Student\_Id,Teacher\_Id

FROM Col\_submission

WHERE month(`Upload\_Date`) = 10;

**Explanation:** Query 6 returns the output of submission table details in any specific month. For example I have used the month is equal to 10 to fetch the submission done in October.

**Result:** 

**7. Aggregate functions. (SUM, AVG, MIN, MAX, COUNT etc.)**

**--SUM of points received in Feedback for student in semester**

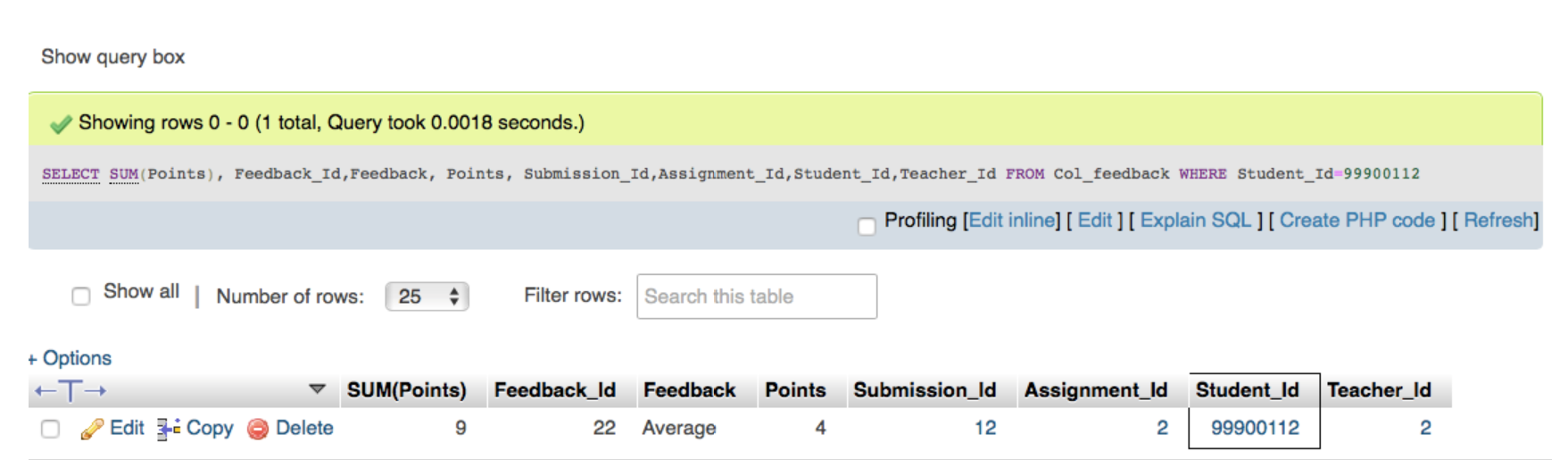
SELECT SUM(Points), Feedback\_Id,Feedback, Points, Submission\_Id,Assignment\_Id,Student\_Id,Teacher\_Id

FROM Col\_feedback

WHERE Student\_Id=99900112

**Explanation:**  The above query returns output for student sum of points along with feedback table components where the student id is equal to 99900112.

**Result:**



**--AVG Average results of total assignment submissions for a particular student in semester**

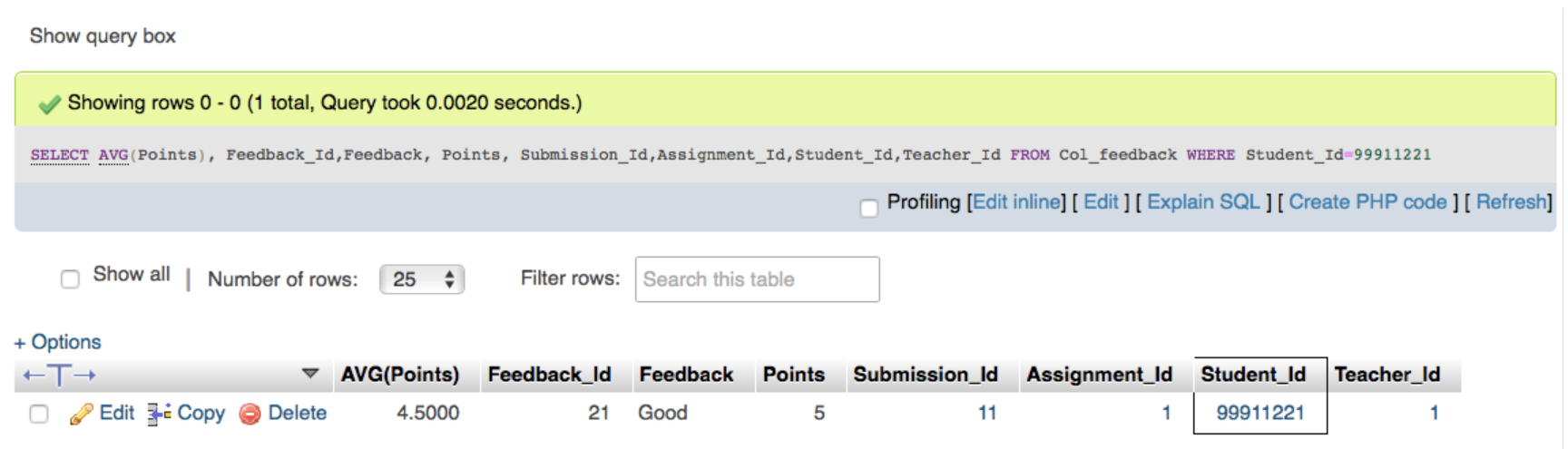
SELECT AVG(Points), Feedback\_Id,Feedback, Points, Submission\_Id,Assignment\_Id,Student\_Id,Teacher\_Id

FROM Col\_feedback

WHERE Student\_Id=99911221

**Explanation:**  The above query returns output for student Average of points along with feedback table components where the student id is equal to 99911221.

**Result:**



**--COUNT of total assignments submitted by student in semester**

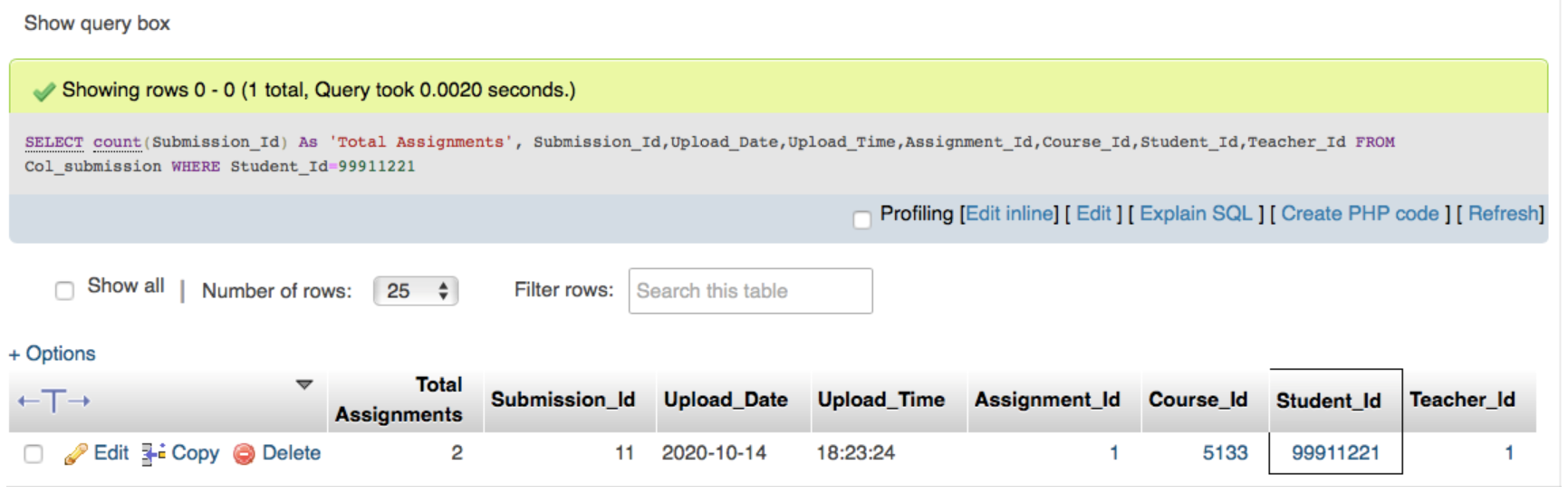
SELECT count(Submission\_Id) As 'Total Assignments', Submission\_Id,Upload\_Date,Upload\_Time,Assignment\_Id,Course\_Id,Student\_Id,Teacher\_Id

FROM Col\_submission

WHERE Student\_Id=99911221;

**Explanation: -**The above query returns output for student count of submission Id by creating alias as Total Assignments along with submission table components where the student id is equal to 99911221

**Result:**



# Chapter 4: Views

Views are nothing but Virtual tables which are used to display selected data but not to store any data within them.

The Following two queries are examples for views.

**View 1:**

CREATE VIEW MARK\_OBTAIN\_ON\_EACH\_TERM

AS

SELECT F.Points,S.Name,E.Term,CASE WHEN Points=0 then 'FAILS'

when Points>0 and Points<10 then 'Average'

when Points>10 and Points<20 then 'Good'

when Points>20 and Points<30 then 'Very Good'

else 'Excelent' end as 'MarkGroup'

from col\_feedback AS F

JOIN Col\_students AS S

ON F.Student\_Id=S.Student\_Id

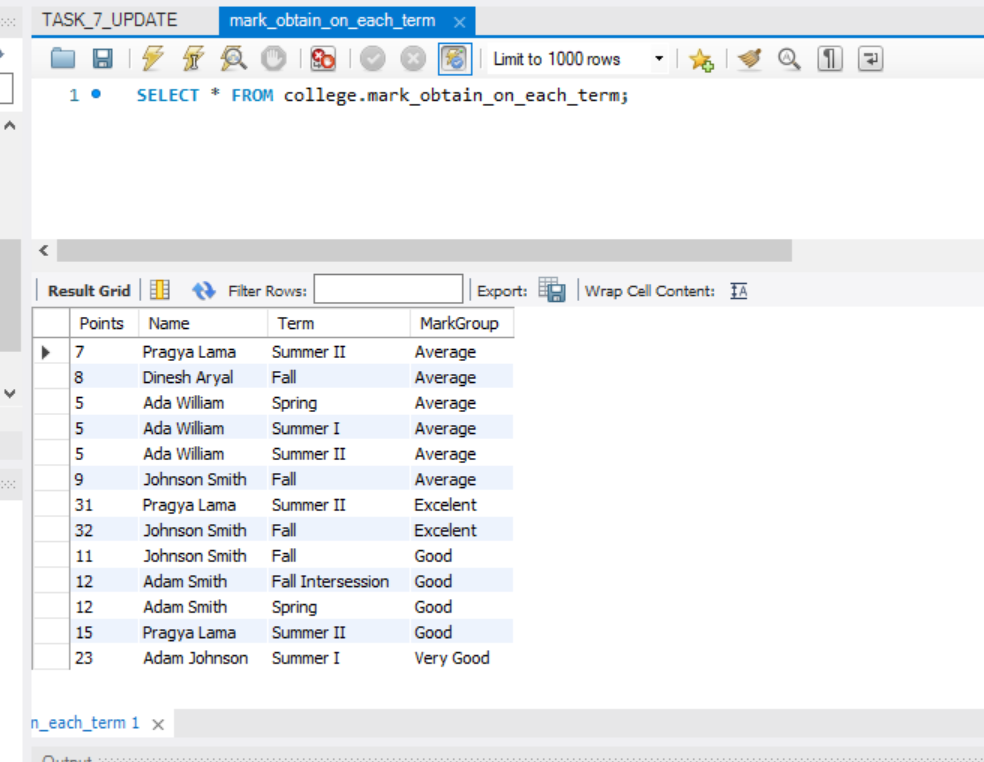
JOIN Col\_enroll as E

ON E.Student\_Id=F.Student\_Id

ORDER BY MarkGroup;

**Result:**





**Explanation:** This view 1 explains number of number of points students that obtain in each term grouped by Mark group which gives the feedback like average good and very good.

**View 2:**

CREATE VIEW CourseMarkwithPoint

AS

SELECT Col\_course.Course\_Name as 'Course',Question,Col\_course.Announcements as 'Assign\_detail',Col\_feedback.Points,Col\_students.Name

From Col\_assignment

join Col\_course

on Col\_assignment.Course\_Id = Col\_course.Course\_Id

join Col\_feedback

on Col\_feedback.Student\_Id=Col\_assignment.Student\_Id

join Col\_Students

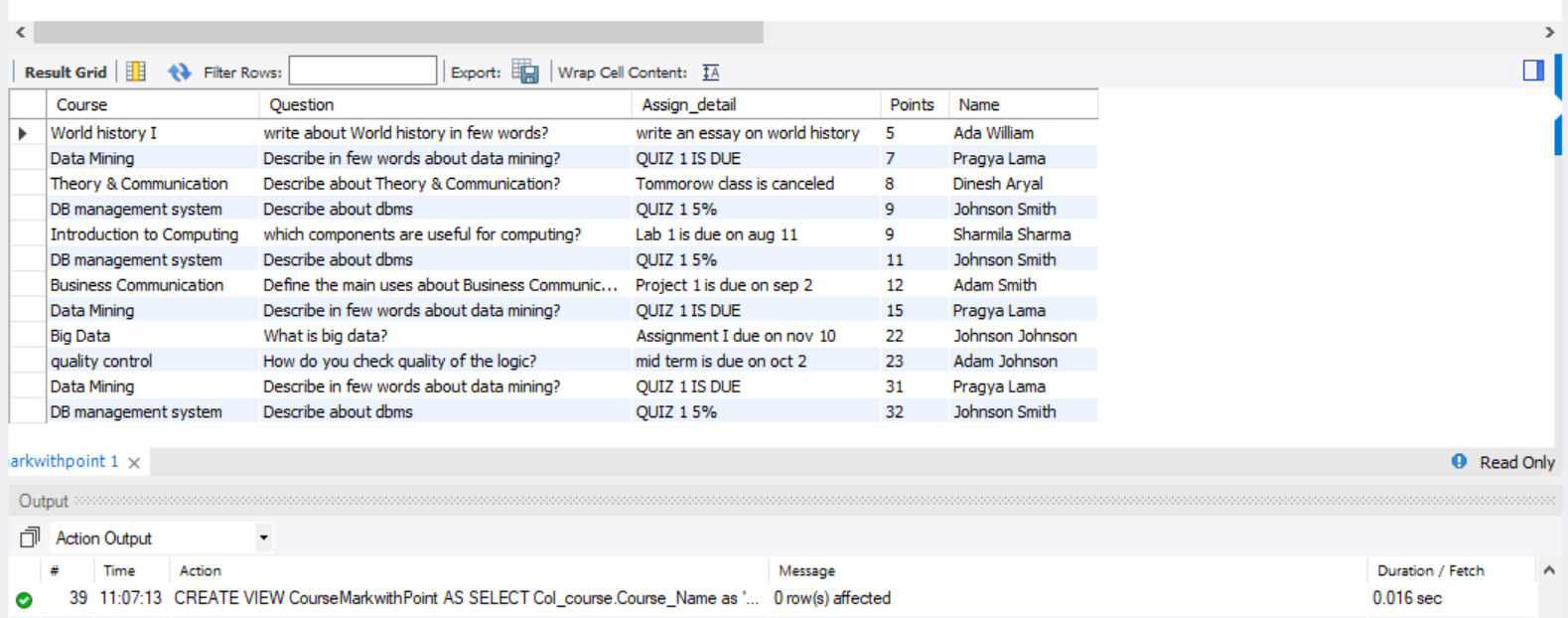
on Col\_assignment.Student\_Id=Col\_Students.Student\_Id

order by Points

limit 10;

**Result:**





**Explanation:**

This view 2 explains about the mark of each students obtained on each course with assignment details and questions.

**View 3:**

CREATE VIEW coursetakenbystudents

AS

SELECT Num\_of\_courses,col\_students.Name

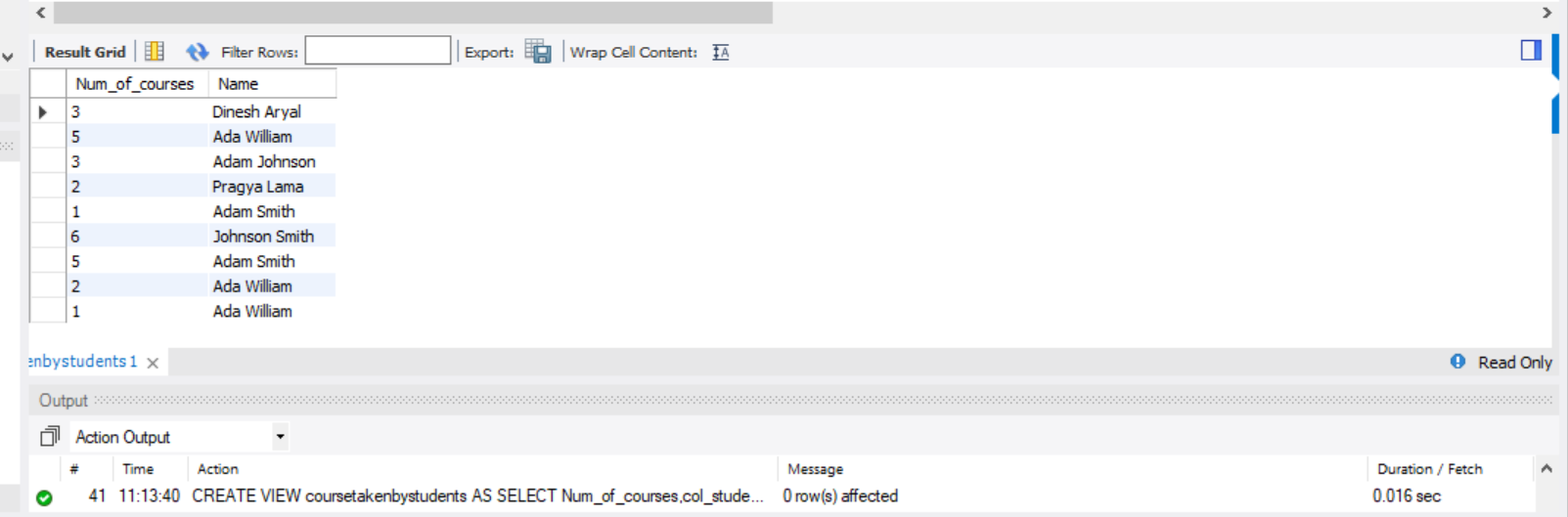
FROM col\_enroll

JOIN col\_students

ON col\_students.Student\_Id=col\_enroll.Student\_Id;

**Result:**





**Explanation:**

This view 3 explains the number of courses taken by each student.

# Chapter 5: Query optimization

An inefficient query can exhaust the resources of the production database and, if the query contains errors, cause slow performance or lack of service for other users. To minimize the effect on database efficiency, it is important that you optimize queries.

Our queries are optimized we just had to make few changes to make it more specific to the business. Some changes have described below with some screenshots.

**Task 7:**

SELECT Name,Student\_Id

FROM Col\_students

WHERE City='Magnolia';

SELECT Name

FROM Col\_students

WHERE Name LIKE '%Jo%';

**Explanation:** This query is very specific and only provide the information what is need instead of using select all we have only use select name which includes word ‘jo’ in their names. This makes our queries fast and easy to search.

ALTER VIEW coursetakenbystudents

AS

SELECT Num\_of\_courses,col\_students.Name

FROM col\_enroll

JOIN col\_students

ON col\_students.Student\_Id=col\_enroll.Student\_Id

ORDER BY Num\_of\_courses;

**Expalanation:**

Here while working on query optimization I found out this view would look much better if I sorted it by num\_of\_courses.

ALTER VIEW coursemarkwithpoint

AS

SELECT Col\_course.Course\_Name as 'Course',Question,Col\_course.Announcements as 'Assign\_detail',Col\_feedback.Points,Col\_students.Name

From Col\_assignment

join Col\_course

on Col\_assignment.Course\_Id = Col\_course.Course\_Id

join Col\_feedback

on Col\_feedback.Student\_Id=Col\_assignment.Student\_Id

join Col\_Students

on Col\_assignment.Student\_Id=Col\_Students.Student\_Id

order by Points

LIMIT 10;

ALter VIEW MARK\_OBTAIN\_ON\_EACH\_TERM

AS

SELECT F.Points,S.Name,E.Term,CASE WHEN Points=0 then 'FAILS'

when Points>0 and Points<10 then 'Average'

when Points>10 and Points<20 then 'Good'

when Points>20 and Points<30 then 'Very Good'

else 'Excelent' end as 'MarkGroup'

from col\_feedback AS F

JOIN Col\_students AS S

ON F.Student\_Id=S.Student\_Id

JOIN Col\_enroll as E

ON E.Student\_Id=F.Student\_Id

ORDER BY Points;

## 5.1 Index:

* The Concept of Index is basically sort data to make it easier to search.
* Indexing provides faster results.
* Indexes are very helpful in a manner that improving performance search results.
* Indexes can be defined more than one column in a Table.
* Examples for Indexes are:

CREATE INDEX STUDENTS\_NAME\_IDX

ON Col\_students (Name);

CREATE INDEX STUDENTS\_NAME\_IDX

ON Col\_course (Course\_Name);

# Chapter 6: Creativity

We have learned about Flask and MySQL connector using Anaconda Jupyter.

**Flask** is a micro [web framework](https://en.wikipedia.org/wiki/Web_framework) written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It is classified as a [microframework](https://en.wikipedia.org/wiki/Microframework) because it does not require particular tools or libraries.

Learning about this two python packages was very helpful because we were able to present out views in graphical ways.

* Connecting database with python Jupyter

We used this to do data analysis on the views which we created using college database.

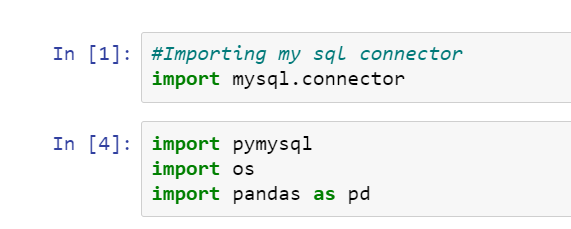
* Packages install:

conda install -c anaconda pymysql

conda install -c anaconda pymysql

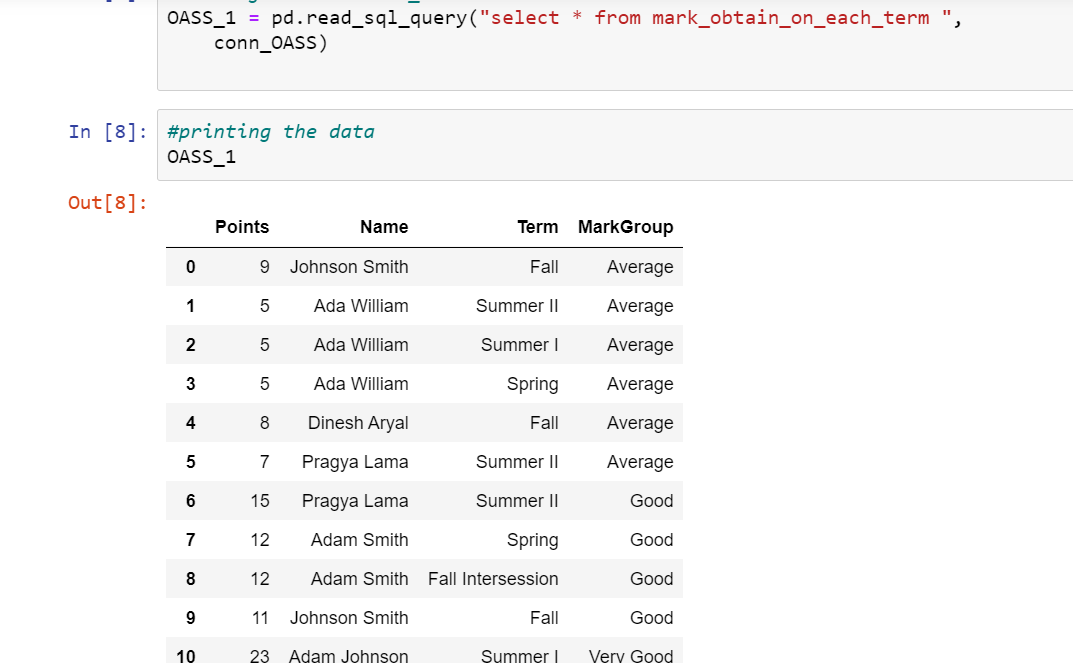
* Python code:

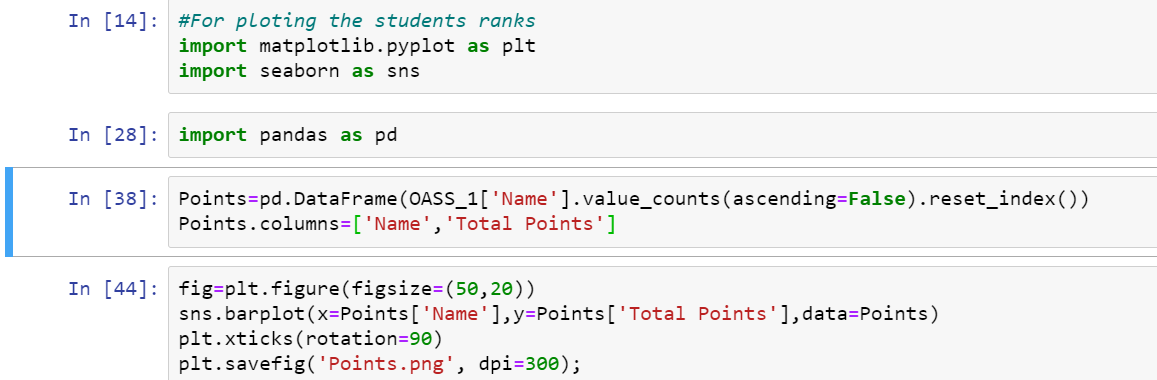
We did research and learn about the python code to connect with database.

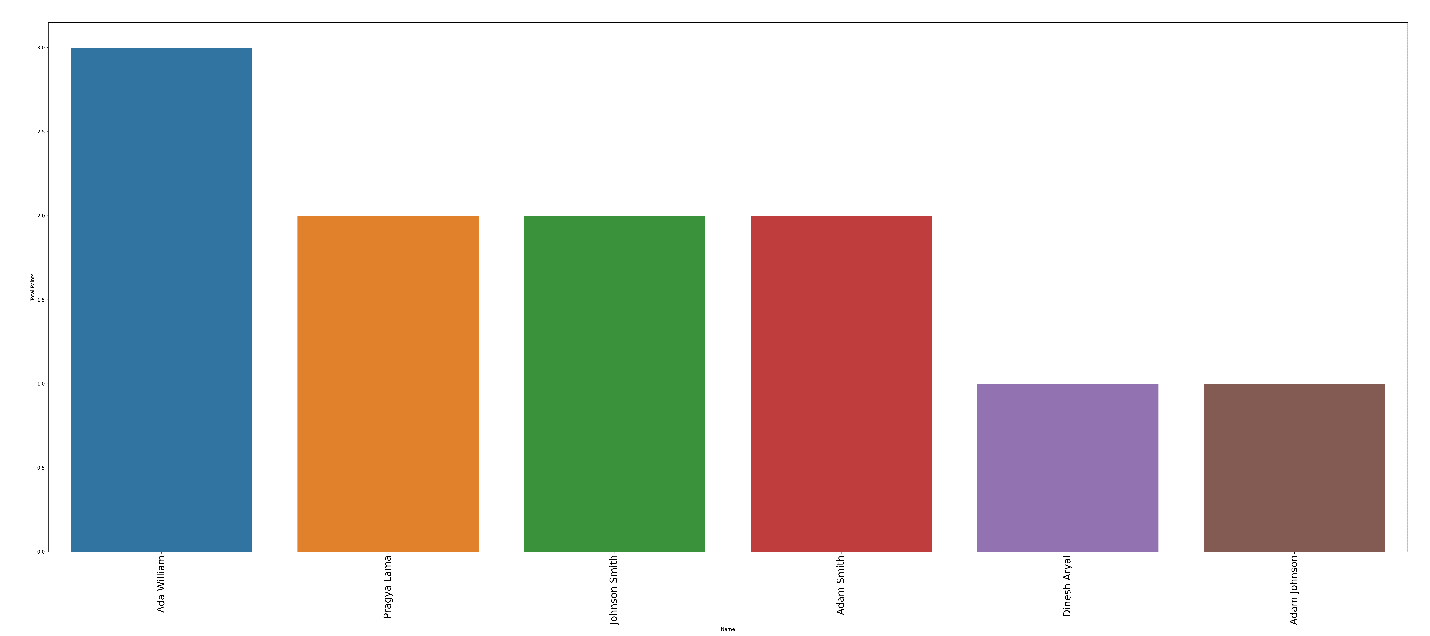




**View mark\_obtain\_on\_each\_term**

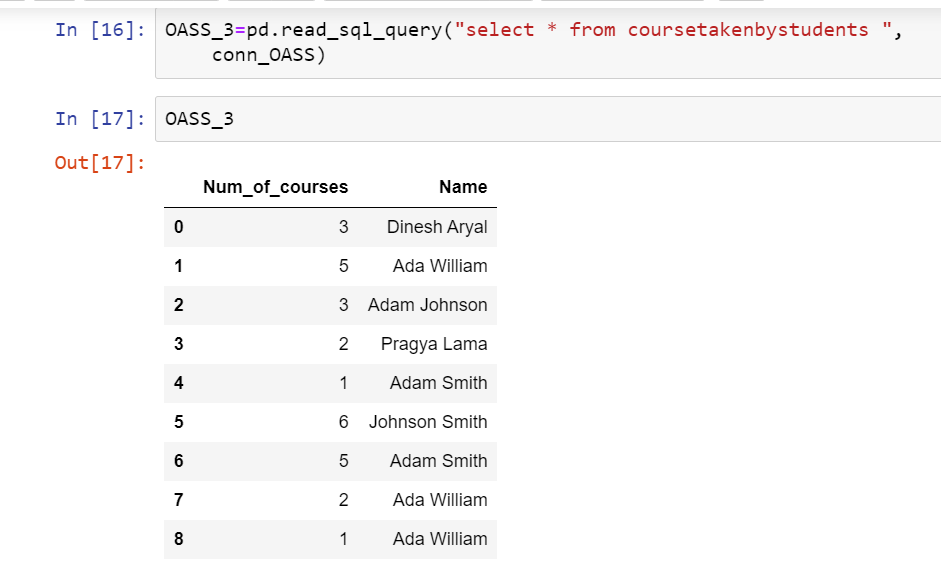


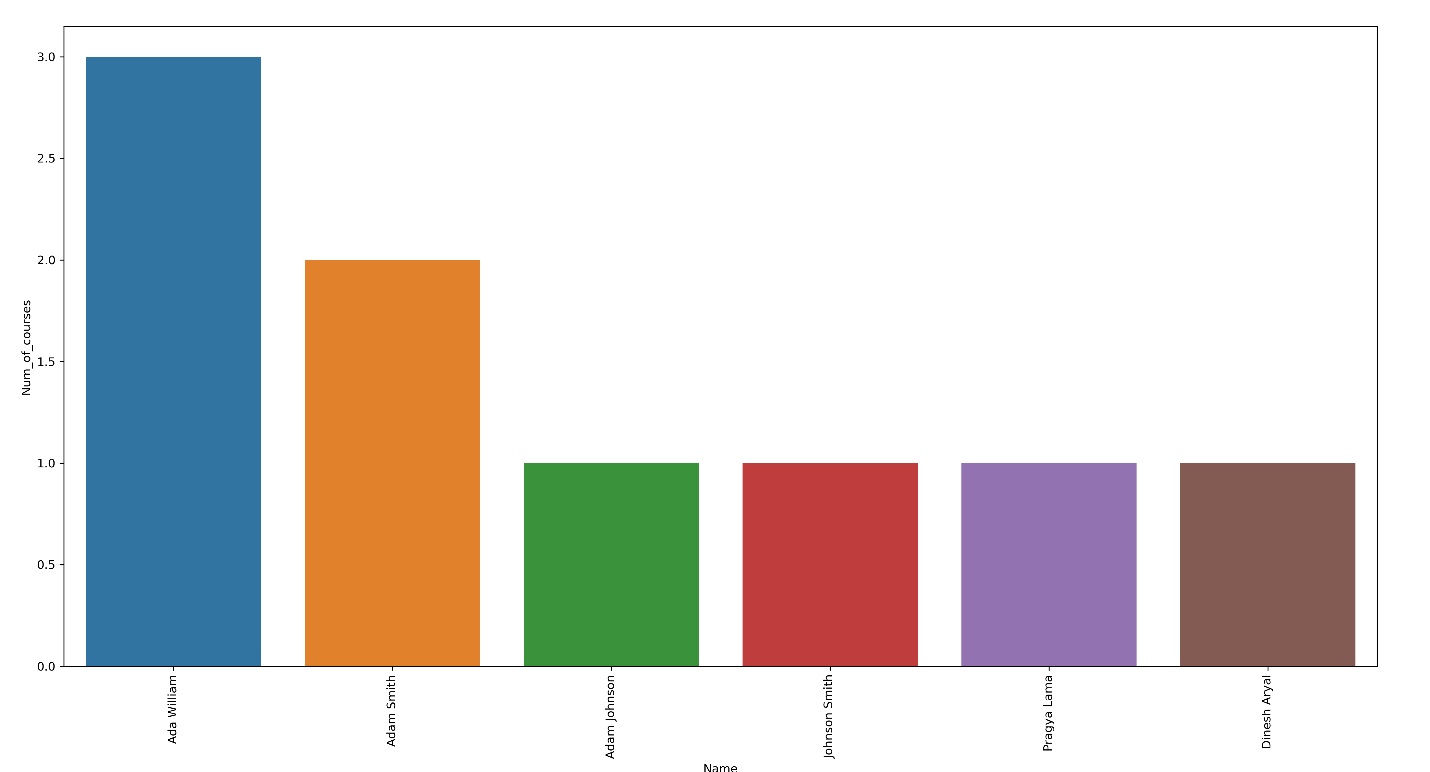




This figure shows that on average Student Ada William rank is highest than other students.

**Views coursetakenbystudents**





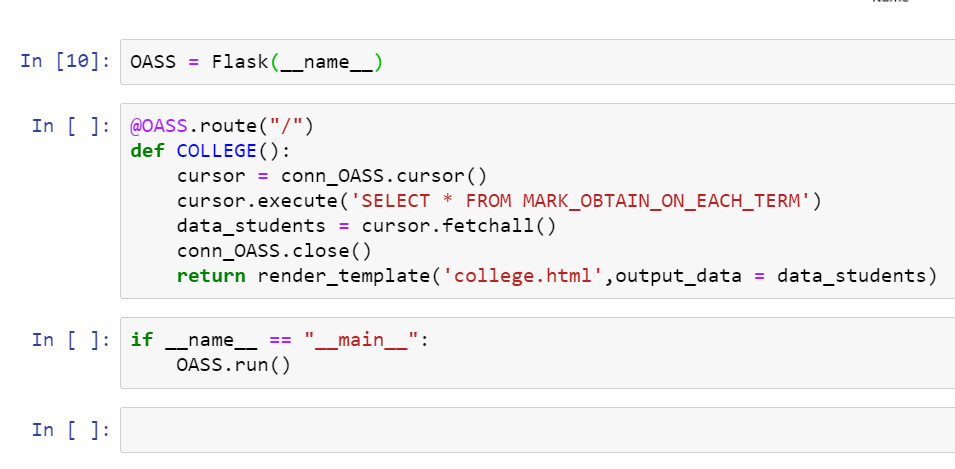
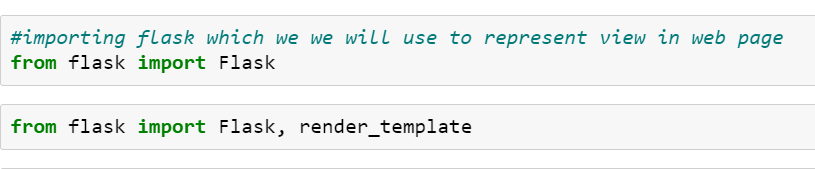
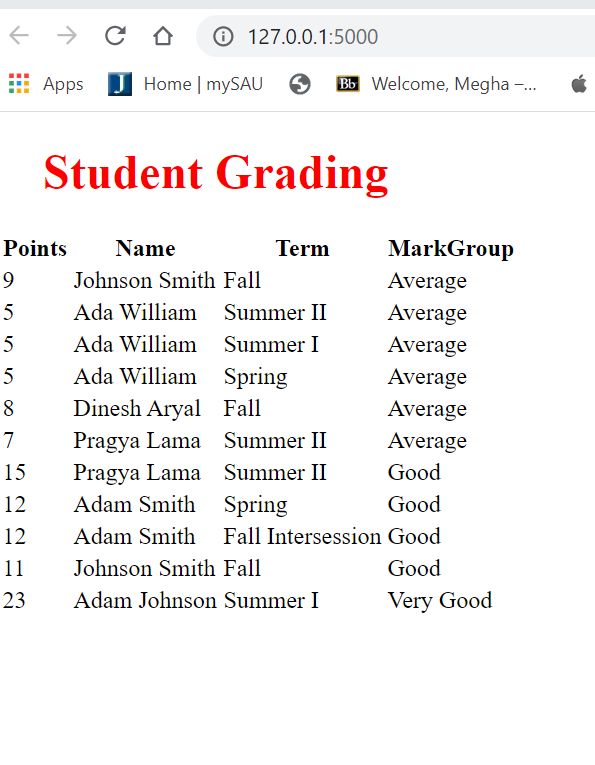
This figure shows that Ada William has taken maximum number of courses.

From these two views we can conclude that Num\_of\_courses are indirectly proportional to students points or grading. If students are smart and work hard, they can achieve good grades even if they have taken maximum num\_of\_courses.

* Connecting to web server with Flask.
* Installed package:

conda install -c anaconda flask

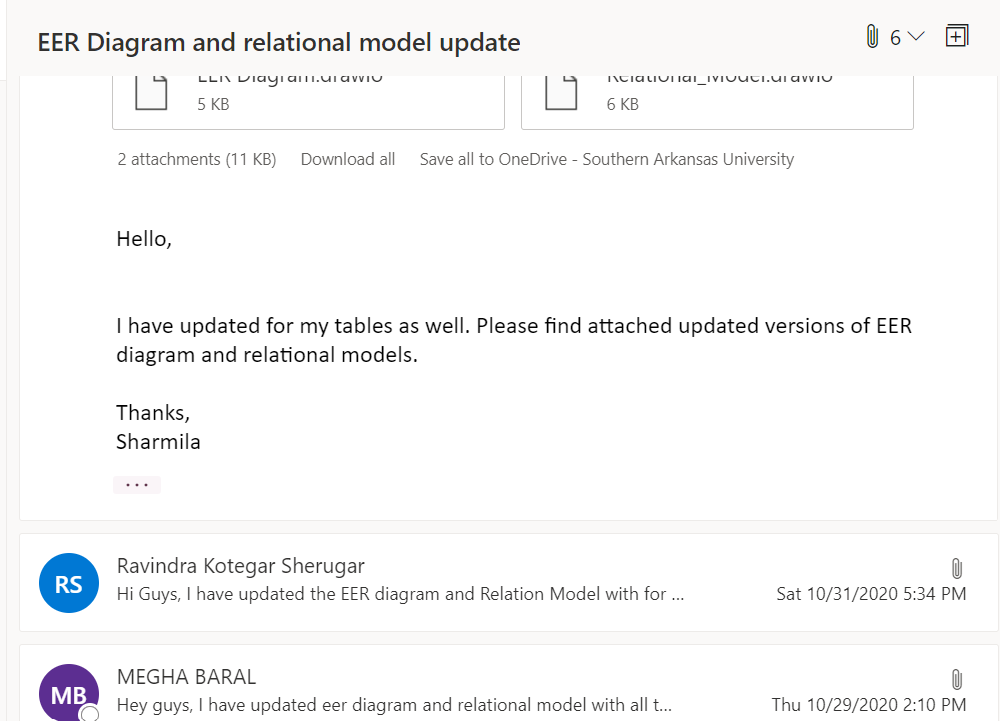
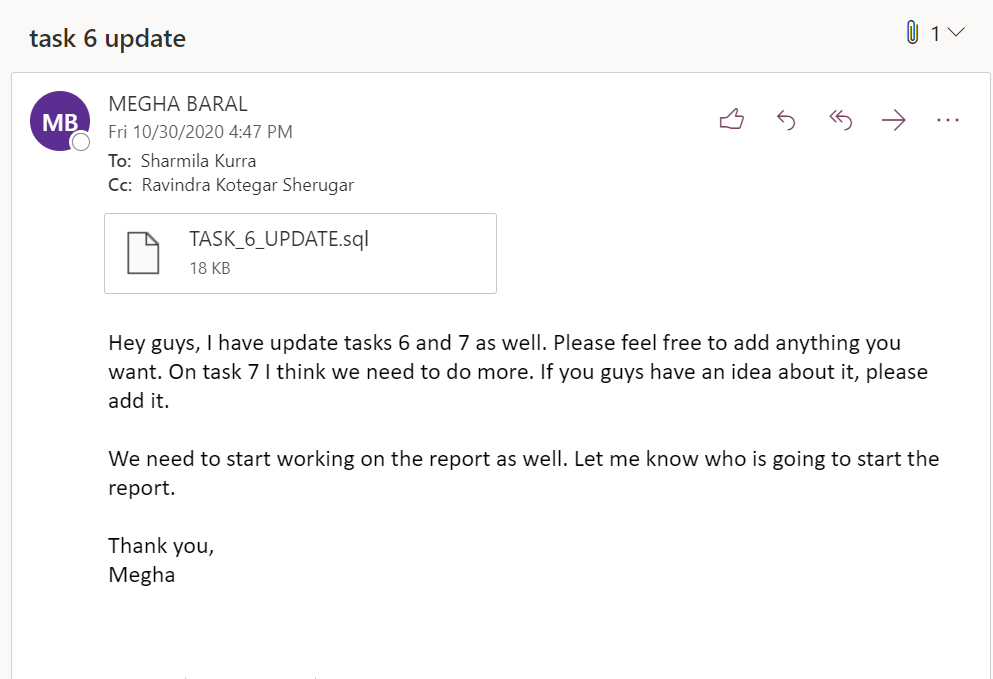
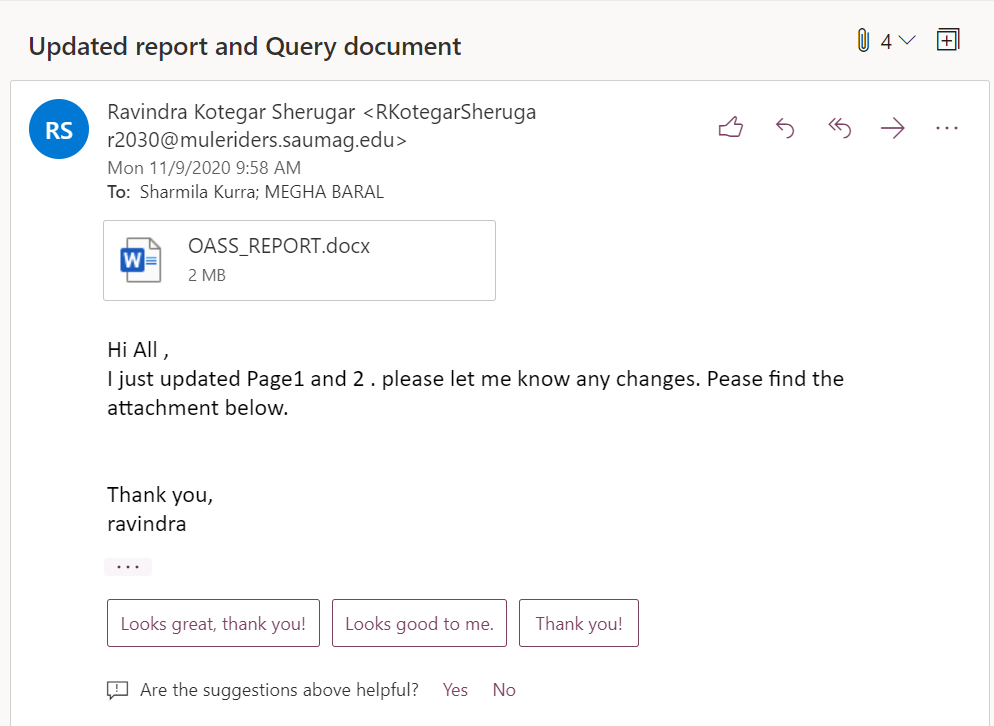
* Set up
* Create a folder “my\_poject” in desktop.
* Save python file in same folder.
* Create another folder “templates” inside “my\_project” folder
* Save html file in templates folder.
* Run it in localhost 127.0.0.1:5000
* Python code

# Chapter 7: Conclusion

In this project we have created college database. We have mentioned all the data based on our understanding and searching internet. Each members of our group have worked on two to three tables where they deal with both DML and DDL commands. This project helps to find us hypothesis with the help of queries and data we have collected is that Num\_of\_courses are indirectly proportional to students points or grading. If students are smart and work hard, they can achieve good grades even if they have taken maximum num\_of\_courses.

While working on this project we discuss through our emails, phone, and WhatsApp group which we have created specially for the project discussion. I have also included some of the screenshots of our group discussion through emails.



Issues we have face while doing this project was on creativity part. All three members of our group have less knowledge on html and CSS. But at the end we came up with idea of using Flask and python connector which are new and interesting for us to work on.

Hence, we all have tried to include all the knowledge we gained from database management system class so far. At last, we would like to thank our instructor Dr. Alshami for providing this opportunity.

# Reference:

* <https://creately.com/diagram/example/ispzfh3e/Online%20Assignment%20Submission%20System>
* Byrnes R & LO B (1996), A computer-aided assignment management system: improving the teaching-learning feedback cycle (Web Page), Southern Cross University, <http://www.opennet.new.au/cmluga/byrnesw2.htm>
* <https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html>
* <https://en.wikipedia.org/wiki/Flask_(web_framework)>
* <https://www.codegrepper.com/code-examples/python/save+image+from+jupyter+notebook>
* <https://www.sisense.com/blog/8-ways-fine-tune-sql-queries-production-databases/>