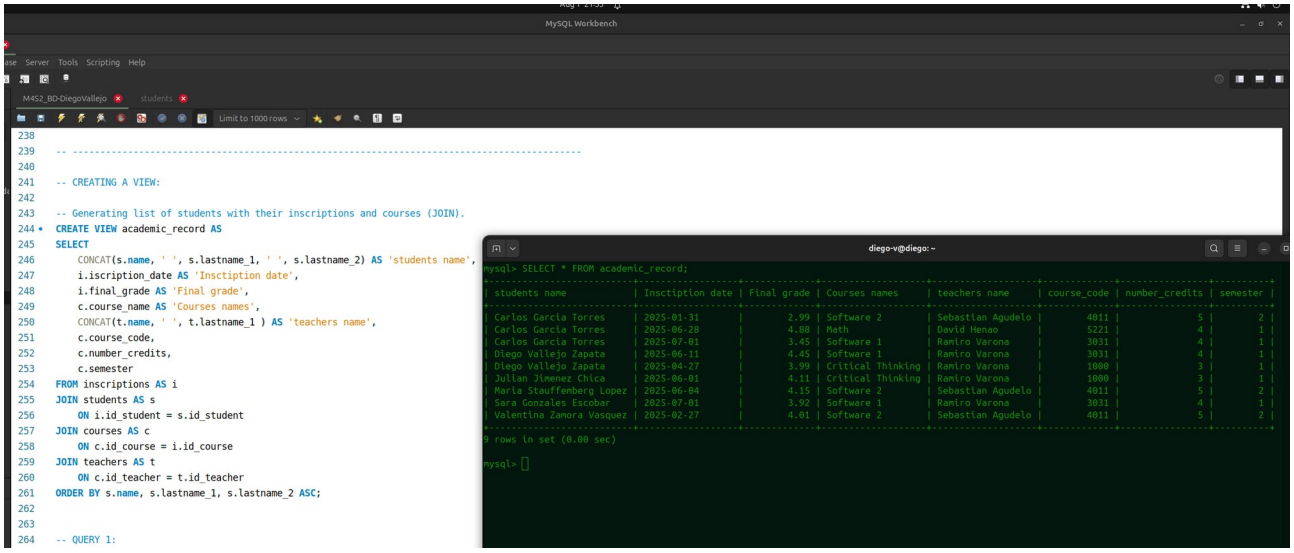
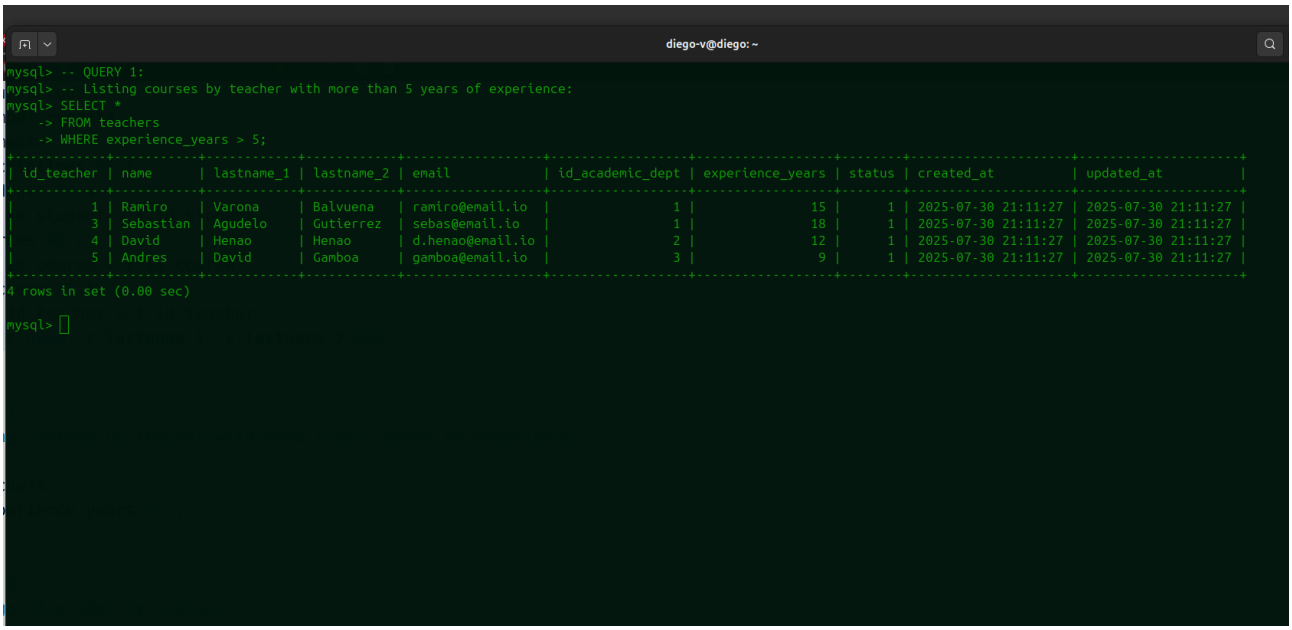
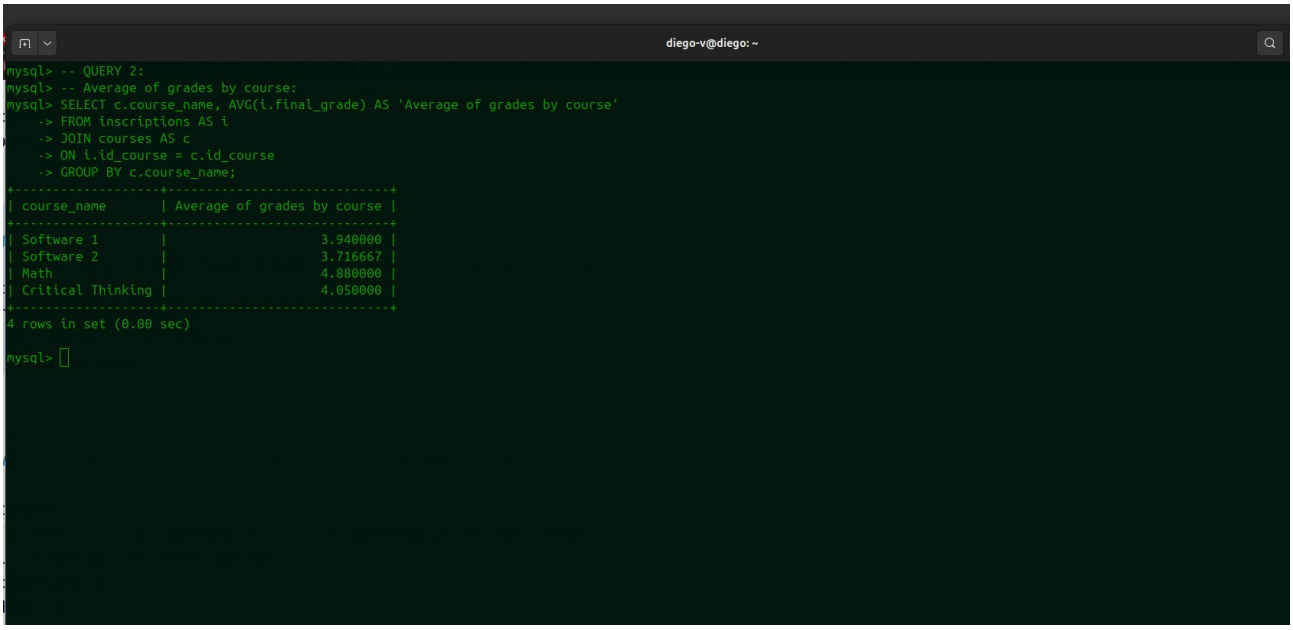
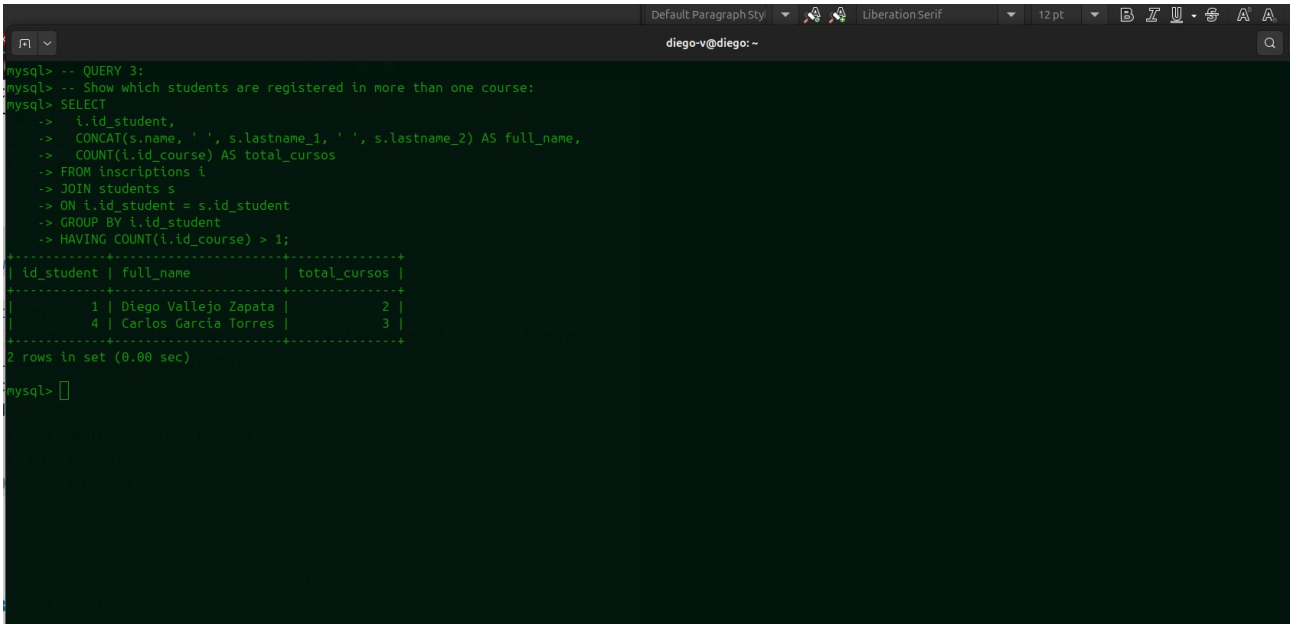
M4S2 Databases
By: Diego Vallejo Z.
Clan Hopper | Riwi.

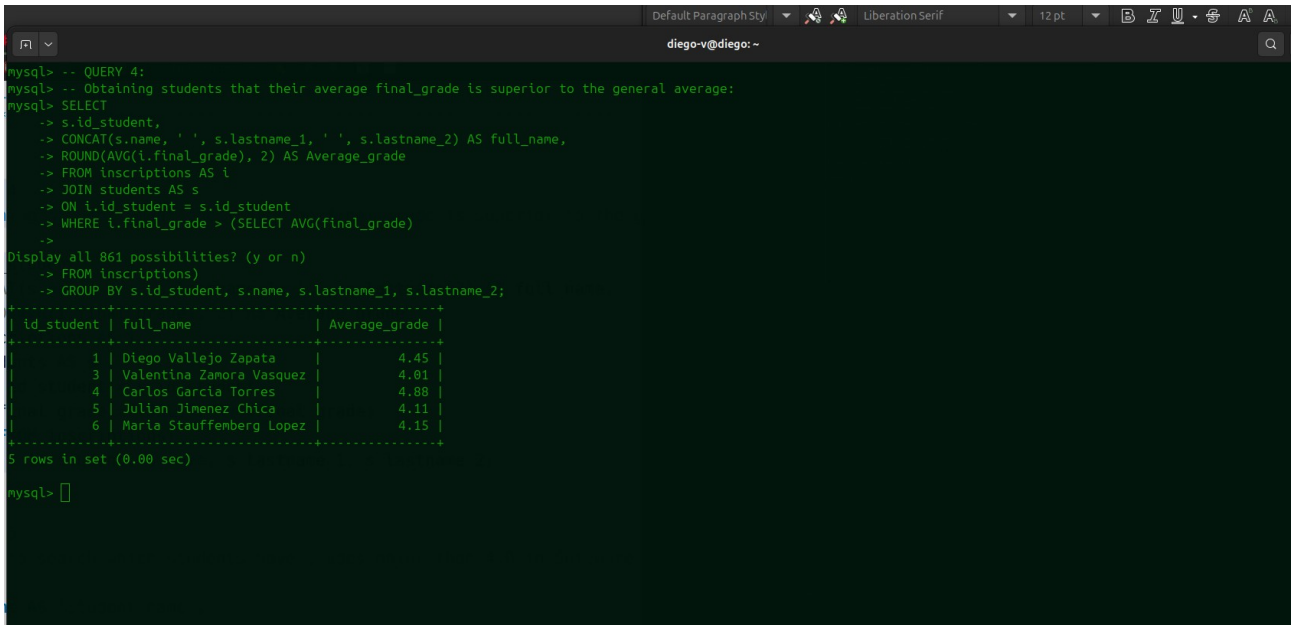
Executing a view:

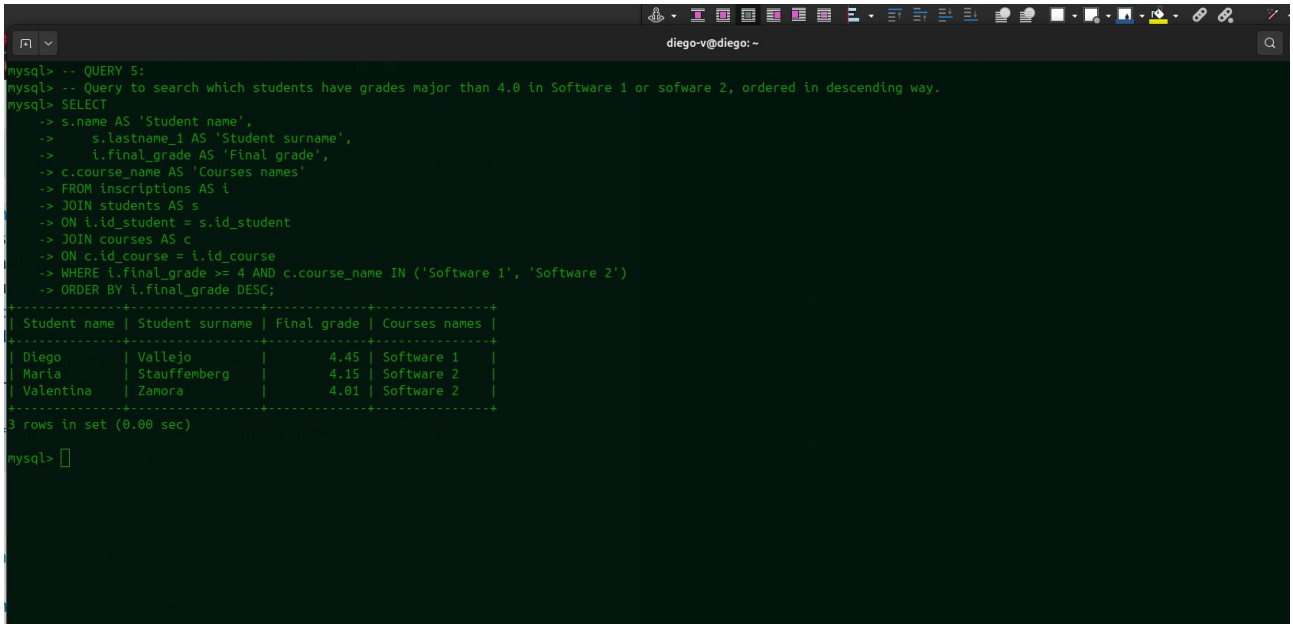


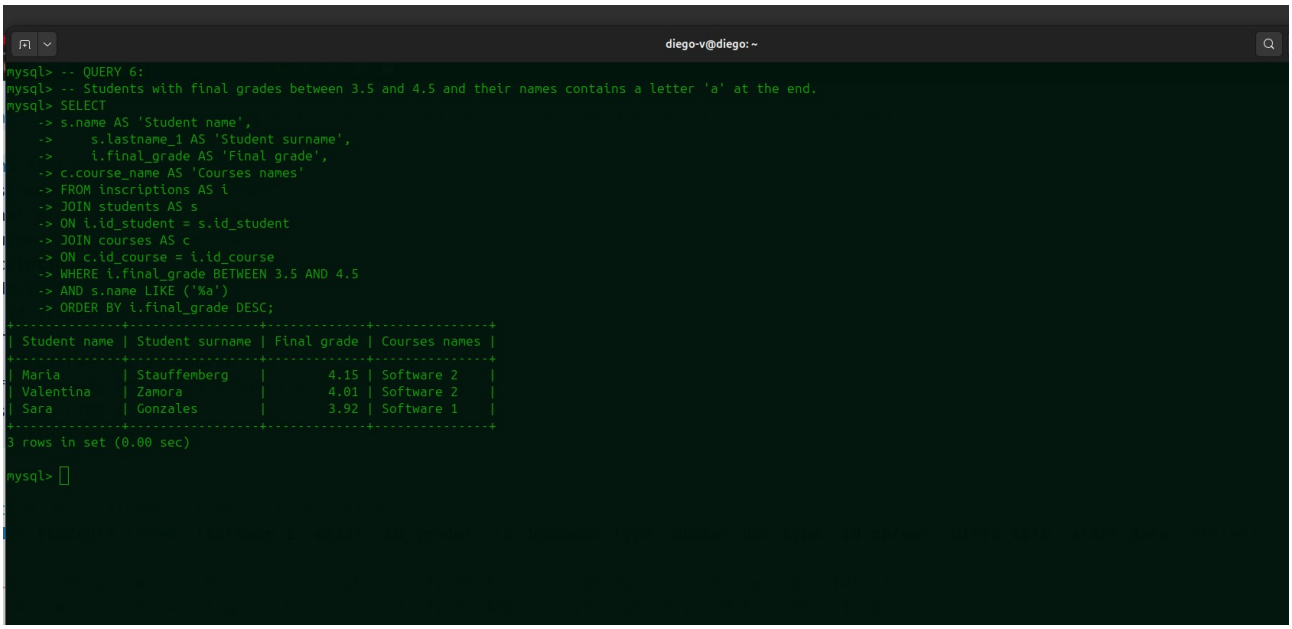


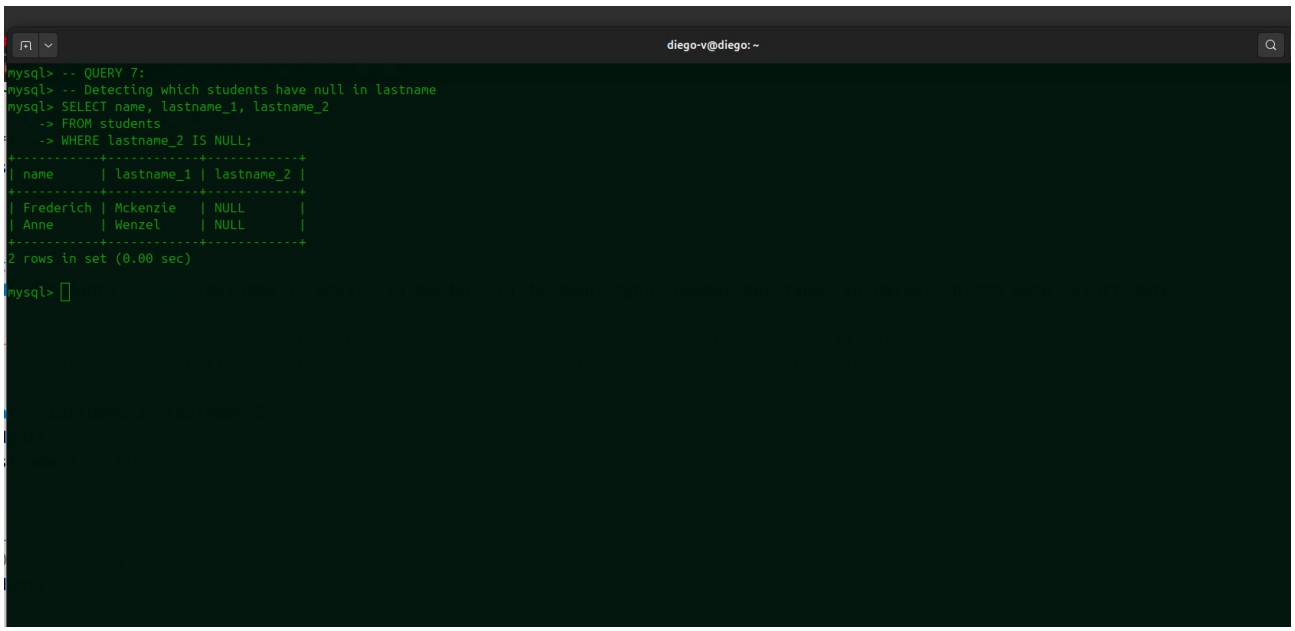

diego-v@diego: ~
mysql> -- QUERY 2:
mysql> -- Average of grades by course:
mysql> SELECT c.course_name, AVG(i.final_grade) AS 'Average of grades by course'
-> FROM inscriptions AS i
-> JOIN courses AS c
-> ON i.id_course = c.id_course
-> GROUP BY c.course_name;
course_name	Average of grades by course
Software 1	3.940000
Software 2	3.716667
Math	4.880000
Critical Thinking	4.050000
4 rows in set (0.00 sec)
mysql>

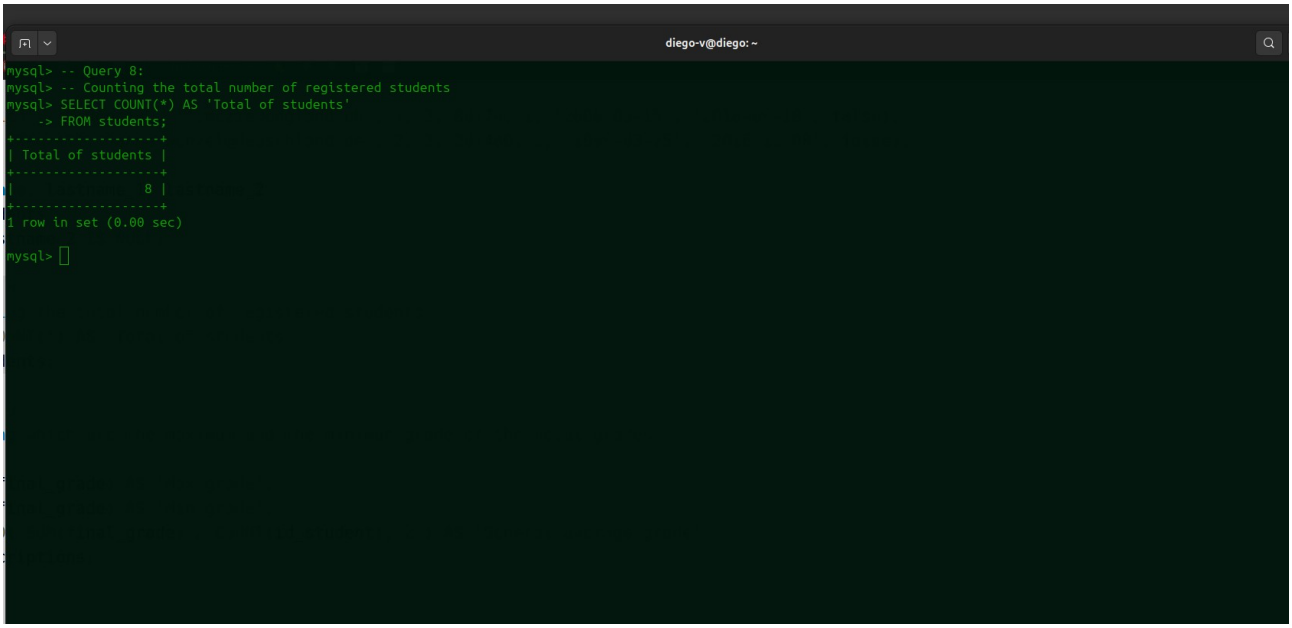

diego-v@diego: ~
mysql> -- QUERY 3:
mysql> -- Show which students are registered in more than one course:
mysql> SELECT
-> i.id_student,
-> CONCAT(s.name, ' ', s.lastname_1, ' ', s.lastname_2) AS full_name,
-> COUNT(i.id_course) AS total_cursos
-> FROM inscriptions i
-> JOIN students s
-> ON i.id_student = s.id_student
-> GROUP BY i.id_student
-> HAVING COUNT(i.id_course) > 1;
id_student	full_name	total_cursos
1	Diego Vallejo Zapata	2
4	Carlos Garcia Torres	3
2 rows in set (0.00 sec)
mysql>

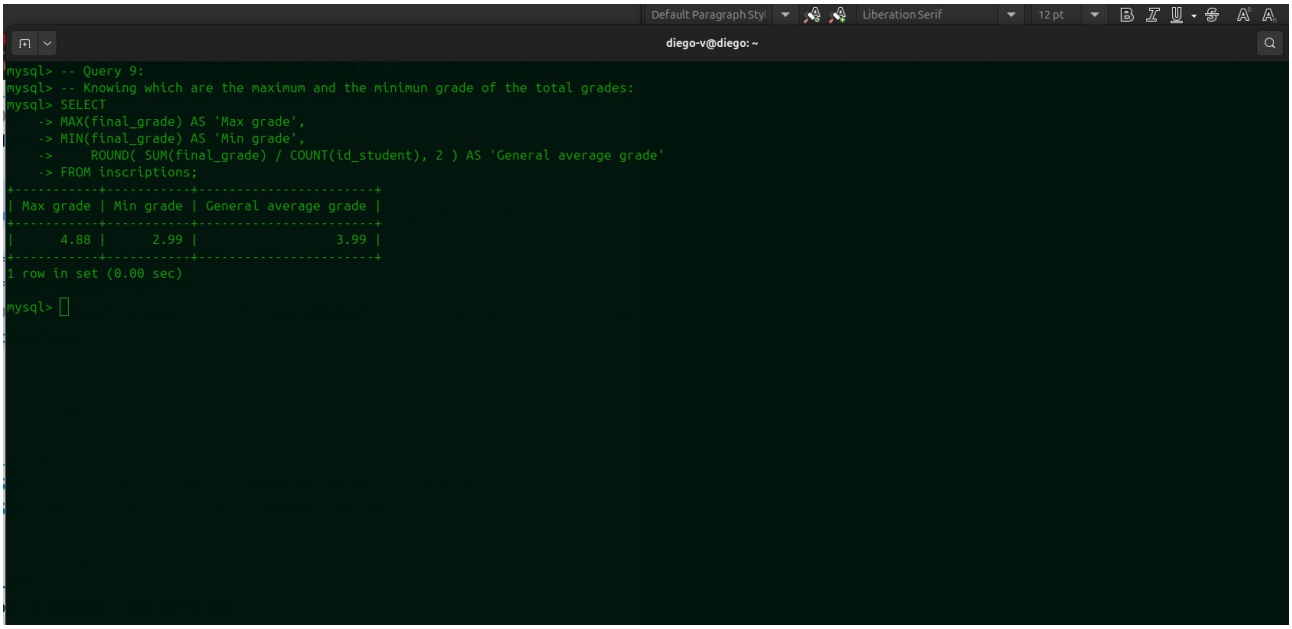

diego-v@diego: ~
mysql> -- QUERY 4:
mysql> -- Obtaining students that their average final_grade is superior to the general average:
mysql> SELECT
-> s.id_student,
-> CONCAT(s.name, ' ', s.lastname_1, ' ', s.lastname_2) AS full_name,
-> ROUND(AVG(i.final_grade), 2) AS Average_grade
-> FROM inscriptions AS i
-> JOIN students AS s
-> ON i.id_student = s.id_student
-> WHERE i.final_grade > (SELECT AVG(final_grade)
->
Display all 861 possibilities? (y or n)
-> FROM inscriptions)
-> GROUP BY s.id_student, s.name, s.lastname_1, s.lastname_2;
id_student	full_name	Average_grade
1	Diego Vallejo Zapata	4.45
3	Valentina Zamora Vasquez	4.01
4	Carlos Garcia Torres	4.88
5	Julian Jimenez Chica	4.11
6	Maria Stauffemberg Lopez	4.15
5 rows in set (0.00 sec)
mysql>


diego-v@diego: ~
mysql> -- QUERY 5:
mysql> -- Query to search which students have grades major than 4.0 in Software 1 or sofware 2, ordered in descending way.
mysql> SELECT
-> s.name AS 'Student name',
-> s.lastname_1 AS 'Student surname',
-> i.final_grade AS 'Final grade',
-> c.course_name AS 'Courses names'
-> FROM inscriptions AS i
-> JOIN students AS s
-> ON i.id_student = s.id_student
-> JOIN courses AS c
-> ON c.id_course = i.id_course
-> WHERE i.final_grade >= 4 AND c.course_name IN ('Software 1', 'Software 2')
-> ORDER BY i.final_grade DESC;
Student name	Student surname	Final grade	Courses names
Diego	Vallejo	4.45	Software 1
Maria	Stauffemberg	4.15	Software 2
Valentina	Zamora	4.01	Software 2
3 rows in set (0.00 sec)
mysql>


diego-v@diego: ~
mysql> -- QUERY 6:
mysql> -- Students with final grades between 3.5 and 4.5 and their names contains a letter 'a' at the end.
mysql> SELECT
-> s.name AS 'Student name',
-> s.lastname_1 AS 'Student surname',
-> i.final_grade AS 'Final grade',
-> c.course_name AS 'Courses names'
-> FROM inscriptions AS i
-> JOIN students AS s
-> ON i.id_student = s.id_student
-> JOIN courses AS c
-> ON c.id_course = i.id_course
-> WHERE i.final_grade BETWEEN 3.5 AND 4.5
-> AND s.name LIKE ('%a')
-> ORDER BY i.final_grade DESC;
Student name	Student surname	Final grade	Courses names
Maria	Stauffemberg	4.15	Software 2
Valentina	Zamora	4.01	Software 2
Sara	Gonzales	3.92	Software 1
3 rows in set (0.00 sec)
mysql>


diego-v@diego: ~
mysql> -- QUERY 7:
mysql> -- Detecting which students have null in lastname
mysql> SELECT name, lastname_1, lastname_2
-> FROM students
-> WHERE lastname_2 IS NULL;
name	lastname_1	lastname_2
Frederich	Mckenzie	NULL
Anne	Wenzel	NULL
2 rows in set (0.00 sec)
mysql>


diego-v@diego: ~
mysql> -- Query 8:
mysql> -- Counting the total number of registered students
mysql> SELECT COUNT(*) AS 'Total of students'
-> FROM students;
+------------------+
| Total of students |
+------------------+
| 8 |
+------------------+
1 row in set (0.00 sec)
mysql>


Default Paragraph Sty
Liberation Serif
12 pt
diego-v@diego: ~
mysql> -- Query 9:
mysql> -- Knowing which are the maximum and the minimun grade of the total grades:
mysql> SELECT
-> MAX(final_grade) AS 'Max grade',
-> MIN(final_grade) AS 'Min grade',
-> ROUND(SUM(final_grade) / COUNT(id_student), 2) AS 'General average grade'
-> FROM inscriptions;
+-----------+-----------+----------------------+
| Max grade | Min grade | General average grade |
+-----------+-----------+----------------------+
| 4.88 | 2.99 | 3.99 |
+-----------+-----------+----------------------+
1 row in set (0.00 sec)
mysql>