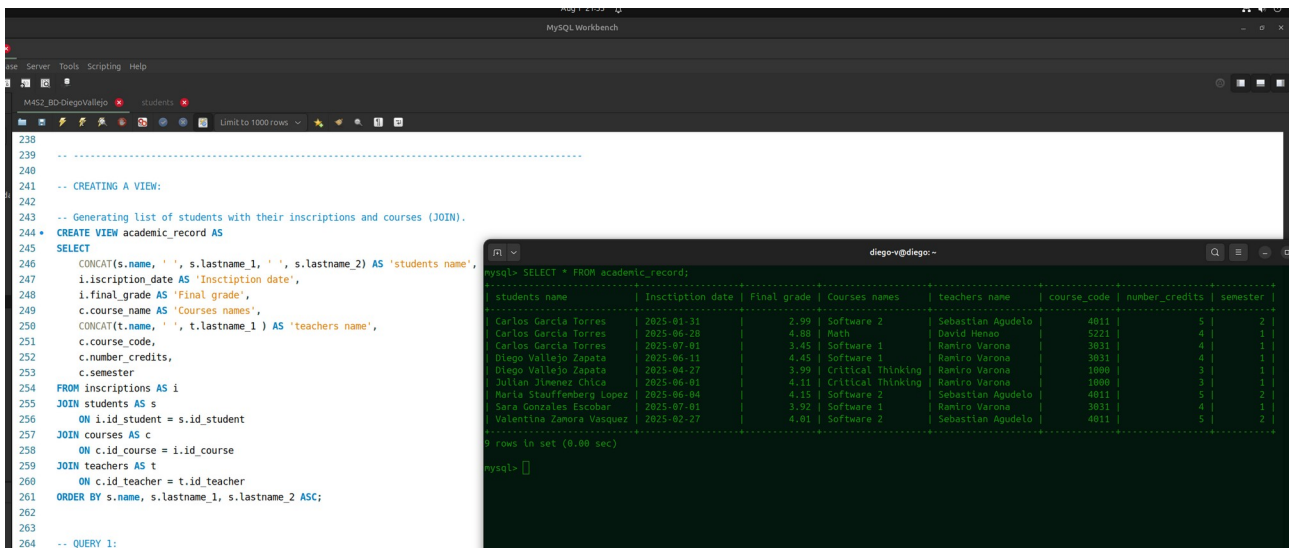


M4S2 Databases

By: Diego Vallejo Z.
Clan Hopper | Riwi.

Executing a view:



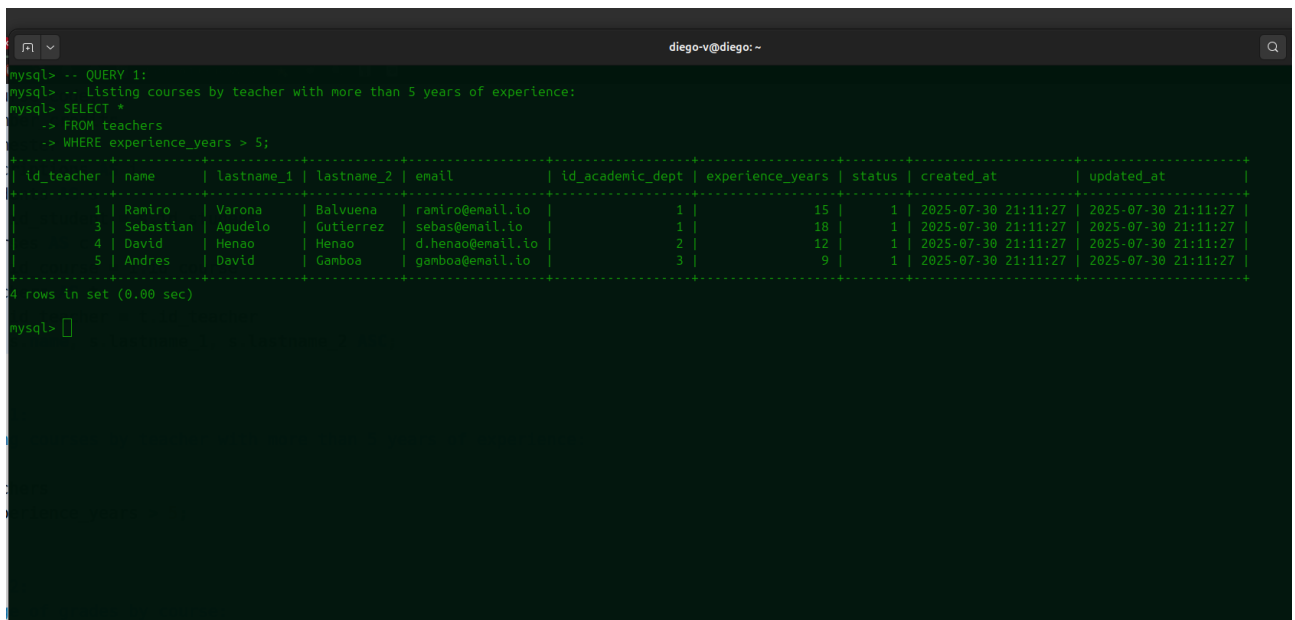
The screenshot shows the MySQL Workbench interface. The left pane displays the SQL editor with the following code:

```
238
239
240
241 -- CREATING A VIEW:
242
243 -- Generating list of students with their inscriptions and courses (JOIN).
244 CREATE VIEW academic_record AS
245 SELECT
246     CONCAT(s.name, ' ', s.lastname_1, ' ', s.lastname_2) AS 'students name',
247     i.inscription_date AS 'Inscription date',
248     i.final_grade AS 'Final grade',
249     c.course_name AS 'Courses names',
250     CONCAT(t.name, ' ', t.lastname_1) AS 'teachers name',
251     c.course_code,
252     c.number_credits,
253     c.semester
254 FROM inscriptions AS i
255 JOIN students AS s
256     ON i.id_student = s.id_student
257 JOIN courses AS c
258     ON c.id_course = i.id_course
259 JOIN teachers AS t
260     ON c.id_teacher = t.id_teacher
261 ORDER BY s.name, s.lastname_1, s.lastname_2 ASC;
262
263
264 -- QUERY 1:
```

The right pane shows the execution of the query, displaying the results of the 'academic_record' view. The results are as follows:

students name	Inscription date	Final grade	Courses names	teachers name	course_code	number_credits	semester
Carlos Garcia Torres	2025-01-31	2.99	Software 2	Sebastian Agudelo	4011	5	2
Carlos Garcia Torres	2025-06-28	4.08	Math	David Henao	5221	4	1
Carlos Garcia Torres	2025-07-01	3.45	Software 1	Ramiro Varona	3831	4	1
Diego Vallejo Zapata	2025-06-11	4.45	Software 1	Ramiro Varona	3831	4	1
Diego Vallejo Zapata	2025-04-27	3.99	Critical Thinking	Ramiro Varona	1000	3	1
Julian Jimenez Chica	2025-06-01	4.11	Critical Thinking	Ramiro Varona	1000	3	1
Marta Steuffenberg Lopez	2025-06-04	4.15	Software 2	Sebastian Agudelo	4011	5	2
Sara Gonzales Escobar	2025-07-01	3.92	Software 1	Ramiro Varona	3831	4	1
Valentina Zamora Vasquez	2025-02-27	4.01	Software 2	Sebastian Agudelo	4011	5	2

9 rows in set (0.00 sec)



The screenshot shows a terminal window with the following commands and output:

```
mysql> -- QUERY 1:
mysql> -- Listing courses by teacher with more than 5 years of experience:
mysql> SELECT *
-> FROM teachers
-> WHERE experience_years > 5;
```

id_teacher	name	lastname_1	lastname_2	email	id_academic_dept	experience_years	status	created_at	updated_at
1	Ramiro	Varona	Balvuela	ramiro@email.io	1	15	1	2025-07-30 21:11:27	2025-07-30 21:11:27
3	Sebastian	Agudelo	Gutierrez	sebas@email.io	1	18	1	2025-07-30 21:11:27	2025-07-30 21:11:27
4	David	Henao	Henao	d.henao@email.io	2	12	1	2025-07-30 21:11:27	2025-07-30 21:11:27
5	Andres	David	Gamboa	gamboa@email.io	3	9	1	2025-07-30 21:11:27	2025-07-30 21:11:27

4 rows in set (0.00 sec)

```
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>
```

```
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>
```

```
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(f.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 Stauffenberg Lopez | 4.15          |
+-----+
5 rows in set (0.00 sec)

mysql>
```

```
mysql> -- QUERY 5:
mysql> -- Query to search which students have grades major than 4.0 in Software 1 or software 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        | Stauffenberg    | 4.15        | Software 2     |
| Valentina    | Zamora          | 4.01        | Software 2     |
+-----+
3 rows in set (0.00 sec)

mysql>
```

```
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 |
+-----+
| Marla        | Stauffenberg    | 4.15        | Software 2     |
| Valentina    | Zamora          | 4.01        | Software 2     |
| Sara         | Gonzales        | 3.92        | Software 1     |
+-----+
3 rows in set (0.00 sec)

mysql>
```

```
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 |
+-----+
| Frederick | Mckenzie   | NULL       |
| Anne      | Wenzel     | NULL       |
+-----+
2 rows in set (0.00 sec)

mysql>
```

```
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>
```

```
mysql> -- Query 9:
mysql> -- Knowing which are the maximum and the minimum 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>
```