



Proyecto Final Introducción a las Bases de datos

Diseñar y desarrollar una base de datos en MySQL que refleje la relación 1:N entre tablas, aplicando claves primarias, foráneas, consultas básicas, joins, estadísticas y manipulación de datos (insertar, actualizar y eliminar).

Estructura del proyecto

Tendremos un sistema de gestión académica simple con estas entidades:

- Departamento
- Estudiante
- Profesor
- Curso
- Clase
- Inscripcion
- Calificacion

Insertar

Insertar 100 Registro en cada tabla

Consultas Basicas Y avanzada

1. Mostrar el nombre y apellido de todos los estudiantes ordenados alfabéticamente por apellido.

The screenshot shows a MySQL IDE interface. On the left, a tree view displays the database structure for 'gestion_academica', including tables like 'calificacion', 'clase', 'curso', 'departamento', 'estudiante', 'inscripcion', and 'profesor'. The main window shows a SQL query: `1. select nombre, apellido from estudiante order by apellido`. Below the query, a 'Result Grid' displays the results of the query, showing a list of students ordered by their last name. The results are as follows:

nombre	apellido
Dayana	Acevedo
Iván	Agular
Valeria	Alvarez
Rafael	Araya
Cecilia	Arias
Adrián	Barrientos
Alvaro	Benitez
Esteban	Bravo
Facundo	Bustamante
Bastian	Bustos
Andrea	Cabrera
Lorena	Caceres
Florencia	Campos
Tomás	Campos

On the right side of the IDE, there is a 'SQLAdditions' panel with a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or toggle automatic help.'



2. Mostrar todos los cursos que tienen más de 3 créditos.

The screenshot shows a database management tool interface. The SQL query entered is: `1 • select nombre, credits from curso where credits > 3`. The results are displayed in a table with two columns: `nombre` and `credits`. All listed courses have 4 credits.

nombre	credits
Programación en Python	4
Estructuras de Datos	4
Electrónica Digital	4
Cálculo Diferencial	4
Química Orgánica	4
Física Cuántica	4
Arquitectura Sostenible	4
Entrenamiento Deportivo	4
Anatomía Humana	4
Farmacología Clínica	4
Ingeniería Estructural	4
Mecánica de Materiales	4
Desarrollo de Aplicaciones	4
Arte Culinario	4

3. Mostrar el nombre del estudiante y el nombre del curso en el que está inscrito (INNER JOIN).

The screenshot shows a database management tool interface. The SQL query entered is: `1 • SELECT Estudiante.nombre AS estudiante, Curso.nombre AS curso FROM Inscripcion
2 INNER JOIN Estudiante ON Inscripcion.id_estudiante = Estudiante.id_estudiante
3 INNER JOIN Clase ON Inscripcion.id_clase = Clase.id_clase
4 INNER JOIN Curso ON Clase.id_curso = Curso.id_curso;`. The results are displayed in a table with two columns: `estudiante` and `curso`.

estudiante	curso
Ana	Programación en Python
Luis	Estructuras de Datos
María	Bases de Datos Avanzadas
Carlos	Electrónica Digital
Laura	Circuitos Eléctricos
Pedro	Cálculo Diferencial
Julia	Álgebra Lineal
David	Química Orgánica
Sofía	Física Cuántica
Miguel	Biología Molecular
Isabel	Gestión Empresarial
Diego	Microeconomía
Valeria	Psicología Comunitaria



4. Mostrar todos los estudiantes y, si están inscritos, el curso correspondiente (LEFT JOIN).

```
1 • SELECT Estudiante.nombre AS estudiante, Curso.nombre AS curso FROM Estudiante
2 LEFT JOIN Inscripcion ON Estudiante.id_estudiante = Inscripcion.id_estudiante
3 LEFT JOIN Clase ON Inscripcion.id_clase = Clase.id_clase
4 LEFT JOIN Curso ON Clase.id_curso = Curso.id_curso;
```

estudiante	curso
Ana	Programación en Python
Luis	Estructuras de Datos
María	Bases de Datos Avanzadas
Carlos	Electrónica Digital
Laura	Circuitos Eléctricos
Pedro	Cálculo Diferencial
Julia	Álgebra Lineal
David	Química Orgánica
Sofía	Física Cuántica
Miguel	Biología Molecular
Isabel	Gestión Empresarial
Diego	Microeconomía
Valeria	Principios de Contabilidad

5. Mostrar todos los cursos, incluyendo los que aún no tienen estudiantes inscritos (RIGHT JOIN).

```
2 FROM Inscripcion
3 RIGHT JOIN Clase ON Inscripcion.id_clase = Clase.id_clase
4 RIGHT JOIN Curso ON Clase.id_curso = Curso.id_curso
5 RIGHT JOIN Estudiante ON Inscripcion.id_estudiante = Estudiante.id_estudiante;
```

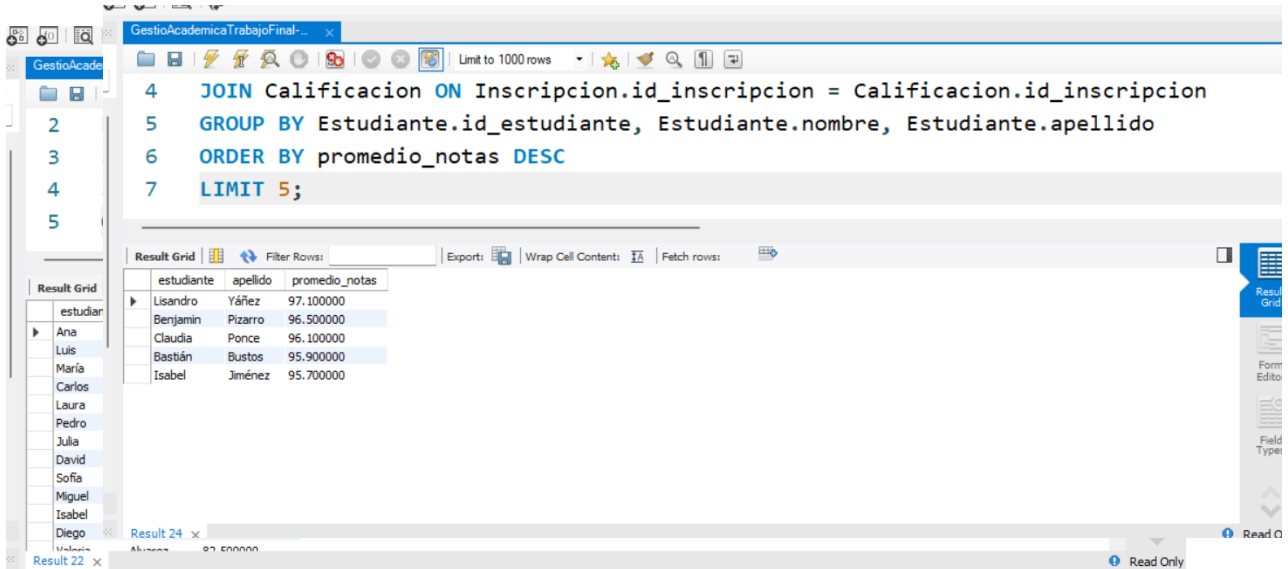
curso	estudiante
Seguridad Informática	Julietta
Desarrollo Web Full-Stack	Facundo
Creación de Videojuegos	Victoria
Análisis de Big Data	Cristian
Ciencia de Datos	Vanessa
Blockchain y Finanzas	Kevin
Redes y Protocolos	Mia
Sistemas Embebidos	Adrián
Automatización Industrial	Camilo
Agricultura Inteligente	Josefina
Mineralogía y Geología	Alan
Meteorología Aplicada	Martina
Psicología	Benjamin

6. Contar cuántos estudiantes hay por departamento.

```
1 • SELECT Departamento.nombre AS departamento, COUNT(Estudiante.id_estudiante) AS cantidad_estudiantes
2 LEFT JOIN Estudiante ON Departamento.id_departamento = Estudiante.id_departamento
3 GROUP BY Departamento.nombre;
```

departamento	cantidad_estudiantes
Ed. Secundaria en Matemática	1
Ed. Secundaria en Física	1
Ed. Secundaria en Química	1
Ed. Secundaria en Historia	1
Ed. Secundaria en Geografía	1
Ed. Secundaria en Lengua	1
Ed. Secundaria en Biología	1
Ed. Secundaria en Inglés	1
Estudios Internacionales	1
Negocios Internacionales	1
Ingeniería en Telecomunicaciones	1
Inteligencia de Negocios	0

7. Calcular el promedio de notas por estudiante.



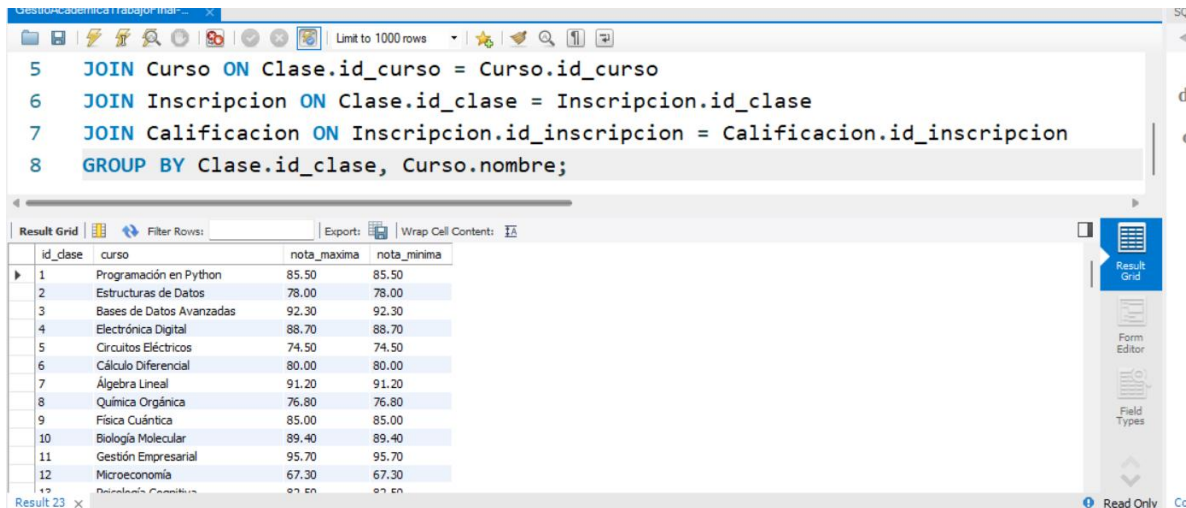
The screenshot shows a database management tool interface. The SQL query editor contains the following code:

```
4 JOIN Calificacion ON Incripcion.id_inscripcion = Calificacion.id_inscripcion
5 GROUP BY Estudiante.id_estudiante, Estudiante.nombre, Estudiante.apellido
6 ORDER BY promedio_notas DESC
7 LIMIT 5;
```

The Result Grid displays the following data:

estudiante	apellido	promedio_notas
Lisandro	Yáñez	97.100000
Benjamin	Pizarro	96.500000
Claudia	Ponce	96.100000
Bastían	Bustos	95.900000
Isabel	Jiménez	95.700000

8. Mostrar la nota máxima y mínima por clase.



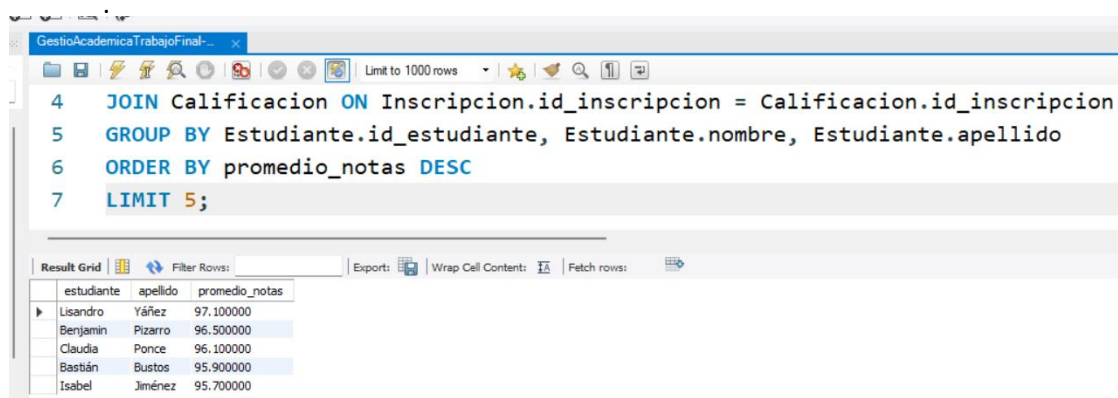
The screenshot shows a database management tool interface. The SQL query editor contains the following code:

```
5 JOIN Curso ON Clase.id_curso = Curso.id_curso
6 JOIN Incripcion ON Clase.id_clase = Incripcion.id_clase
7 JOIN Calificacion ON Incripcion.id_inscripcion = Calificacion.id_inscripcion
8 GROUP BY Clase.id_clase, Curso.nombre;
```

The Result Grid displays the following data:

id_clase	curso	nota_maxima	nota_minima
1	Programación en Python	85.50	85.50
2	Estructuras de Datos	78.00	78.00
3	Bases de Datos Avanzadas	92.30	92.30
4	Electrónica Digital	88.70	88.70
5	Circuitos Eléctricos	74.50	74.50
6	Cálculo Diferencial	80.00	80.00
7	Álgebra Lineal	91.20	91.20
8	Química Orgánica	76.80	76.80
9	Física Cuántica	85.00	85.00
10	Biología Molecular	89.40	89.40
11	Gestión Empresarial	95.70	95.70
12	Microeconomía	67.30	67.30

9. Mostrar los 5 estudiantes con el mayor promedio de notas.



The screenshot shows a database management tool interface. The SQL query editor contains the following code:

```
4 JOIN Calificacion ON Incripcion.id_inscripcion = Calificacion.id_inscripcion
5 GROUP BY Estudiante.id_estudiante, Estudiante.nombre, Estudiante.apellido
6 ORDER BY promedio_notas DESC
7 LIMIT 5;
```

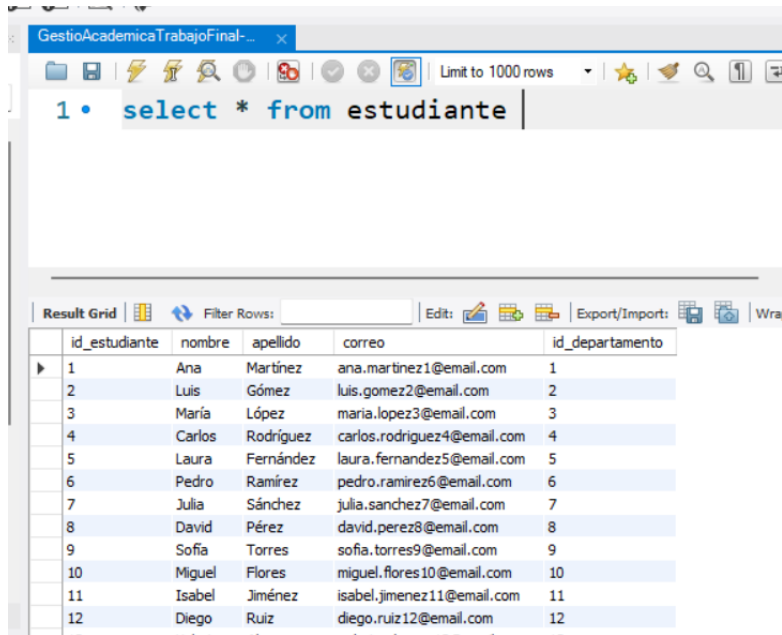
The Result Grid displays the following data:

estudiante	apellido	promedio_notas
Lisandro	Yáñez	97.100000
Benjamin	Pizarro	96.500000
Claudia	Ponce	96.100000
Bastían	Bustos	95.900000
Isabel	Jiménez	95.700000

Actualizar y borrar

10. Cambiar el correo electrónico de un estudiante

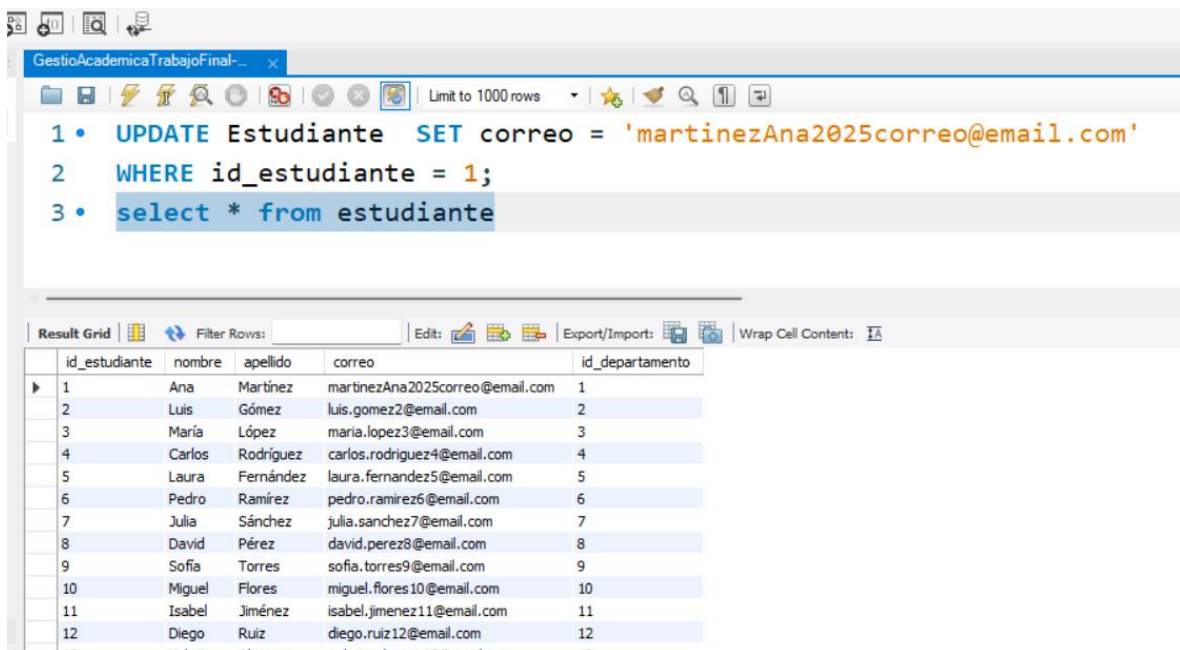
Modificaremos el estudiante con id : 1 que es ana martinez.



The screenshot shows a database management interface with a SQL query editor and a result grid. The query is: `1 • select * from estudiante`. The result grid displays 12 rows of student data.

id_estudiante	nombre	apellido	correo	id_departamento
1	Ana	Martínez	ana.martinez1@email.com	1
2	Luis	Gómez	luis.gomez2@email.com	2
3	María	López	maria.lopez3@email.com	3
4	Carlos	Rodríguez	carlos.rodriguez4@email.com	4
5	Laura	Fernández	laura.fernandez5@email.com	5
6	Pedro	Ramírez	pedro.ramirez6@email.com	6
7	Julia	Sánchez	julia.sanchez7@email.com	7
8	David	Pérez	david.perez8@email.com	8
9	Sofía	Torres	sofia.torres9@email.com	9
10	Miguel	Flores	miguel.flores10@email.com	10
11	Isabel	Jiménez	isabel.jimenez11@email.com	11
12	Diego	Ruiz	diego.ruiz12@email.com	12

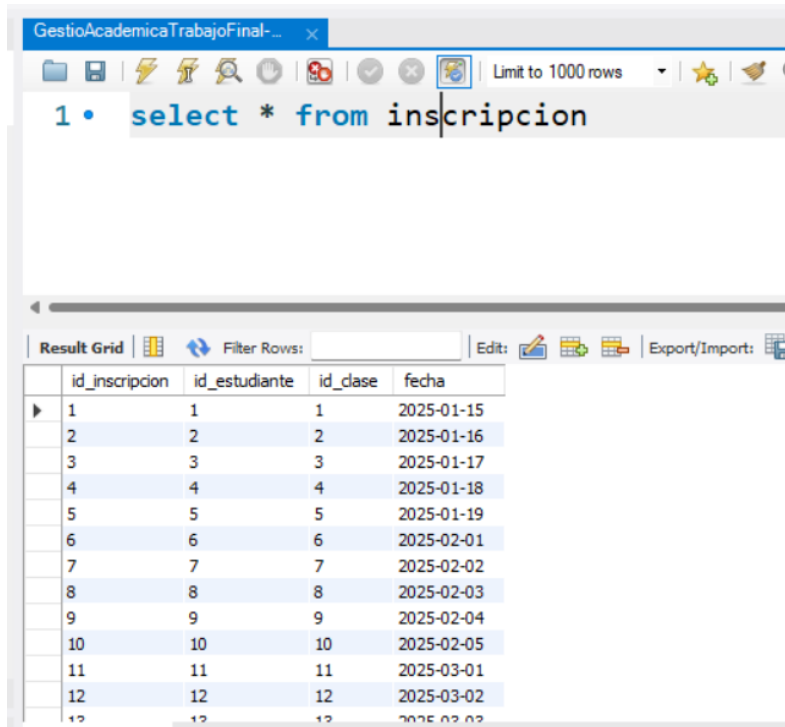
Como podemos ver ya se le modifico el correo al estudian con id: 1



The screenshot shows the same database management interface. The SQL query editor now contains an update query: `1 • UPDATE Estudiante SET correo = 'martinezAna2025correo@email.com'`, `2 WHERE id_estudiante = 1;`, and `3 • select * from estudiante`. The result grid shows the updated data for the first student.

id_estudiante	nombre	apellido	correo	id_departamento
1	Ana	Martínez	martinezAna2025correo@email.com	1
2	Luis	Gómez	luis.gomez2@email.com	2
3	María	López	maria.lopez3@email.com	3
4	Carlos	Rodríguez	carlos.rodriguez4@email.com	4
5	Laura	Fernández	laura.fernandez5@email.com	5
6	Pedro	Ramírez	pedro.ramirez6@email.com	6
7	Julia	Sánchez	julia.sanchez7@email.com	7
8	David	Pérez	david.perez8@email.com	8
9	Sofía	Torres	sofia.torres9@email.com	9
10	Miguel	Flores	miguel.flores10@email.com	10
11	Isabel	Jiménez	isabel.jimenez11@email.com	11
12	Diego	Ruiz	diego.ruiz12@email.com	12

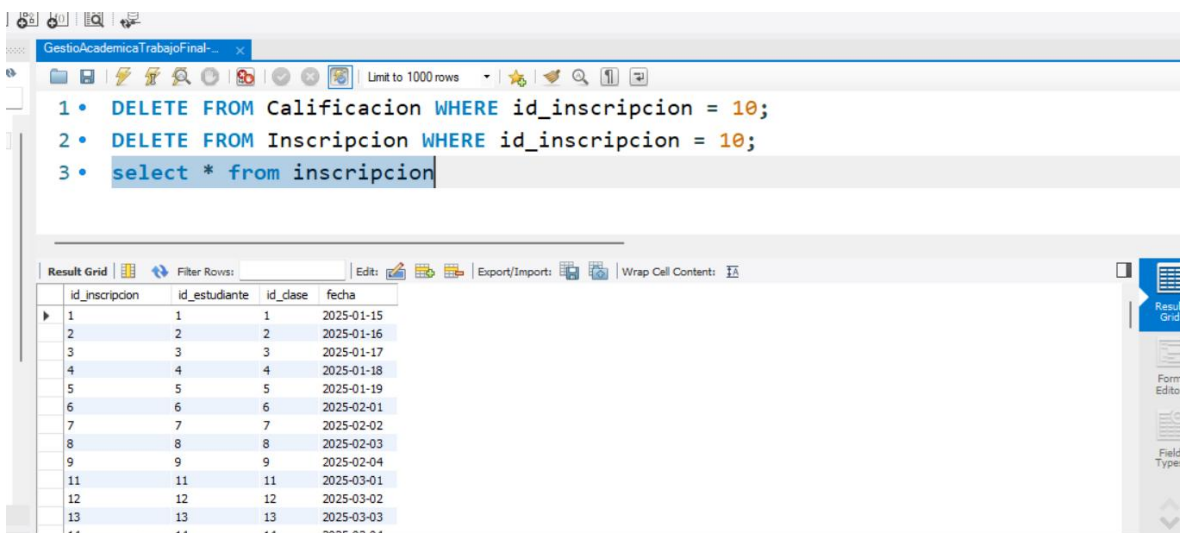
11. Eliminar una inscripción de un estudiante



The screenshot shows a database management tool window titled "GestioAcademicaTrabajoFinal-...". The SQL editor contains the query: `1 • select * from inscripcion`. Below the editor, the "Result Grid" displays the following data:

	id_inscripcion	id_estudiante	id_clase	fecha
1	1	1	1	2025-01-15
2	2	2	2	2025-01-16
3	3	3	3	2025-01-17
4	4	4	4	2025-01-18
5	5	5	5	2025-01-19
6	6	6	6	2025-02-01
7	7	7	7	2025-02-02
8	8	8	8	2025-02-03
9	9	9	9	2025-02-04
10	10	10	10	2025-02-05
11	11	11	11	2025-03-01
12	12	12	12	2025-03-02
13	13	13	13	2025-03-03

Vamos a eliminar la inscripcion del no. 10.



The screenshot shows the same database management tool window. The SQL editor now contains three queries: `1 • DELETE FROM Calificacion WHERE id_inscripcion = 10;`, `2 • DELETE FROM Inscripcion WHERE id_inscripcion = 10;`, and `3 • select * from inscripcion`. The "Result Grid" shows the data after the deletion of row 10:

	id_inscripcion	id_estudiante	id_clase	fecha
1	1	1	1	2025-01-15
2	2	2	2	2025-01-16
3	3	3	3	2025-01-17
4	4	4	4	2025-01-18
5	5	5	5	2025-01-19
6	6	6	6	2025-02-01
7	7	7	7	2025-02-02
8	8	8	8	2025-02-03
9	9	9	9	2025-02-04
11	11	11	11	2025-03-01
12	12	12	12	2025-03-02
13	13	13	13	2025-03-03
14	14	14	14	2025-03-04

Como podemos ver se elimino la no.10. Nota: para eliminar la inscripcion con id 10 tuvimos que eliminar la calificación con id 10 tambien ya que tiene llave foránea y para eliminar la inscripción había que eliminar la calificación primero

En el mismo repositorio de Bases de datos subir una carpeta que diga Trabajo final y agregar el link del repositorio al siguiente URL:

Fecha de entrega final: Lunes 5 de Mayo 2025