



Juan Luis Santos

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.



MySQL Workbench

BD_Proyecto final - Warning - x

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 2

Schemas

Filter objects

pos

sistemaescolar

universidaddb

Tables

calificacion

clase

curso

departamento

estudiante

inscripcion

profesor

Views

Stored Procedures

Functions

Administration Schemas

Information

Schema: universidaddb

1 SELECT Nombre, Apellido

2 FROM Estudiante

3 ORDER BY Apellido ASC;

4

Result Grid

Nombre	Apellido
Alma	Abascal
Rosalba	Abril
Leocadio	Aceto
Leonido	Acosta
Mar	Almazán
Encarnacion	Alsina
Maria Teresa	Amaya
Amaro	Aramburu
Nuria	Arnaz
Violeta	Arranz
Carina	Artigas
Evaristo	Barragán
Agapito	Battle
Teo	Bayón
Gertrudis	Bland
Anna	Blanes
Onofre	Bru
Reyna	Bustama...
Aroa	Cabañas
Edelmiro	Cabeza
Rico	Cabo
Isidora	Campoy

Object Info Session Estudiante 1 x

Read Only

Juan Luis Santos



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

MySQL Workbench

BD_Proyecto final - Warning - ...

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

pos

sistemaescolar

universidaddb

Tables

calificacion

clase

curso

departamento

estudiante

inscripcion

profesor

Views

Stored Procedures

Functions

Administration Schemas

Information

Schema: universidaddb

SQL File 2" curso

```
1 SELECT Nombre, Creditos
2 FROM Curso
3 WHERE Creditos > 3;
4
```

Limit to 1000 rows

Result Grid

Nombre	Creditos
Curso Minus 1	5
Curso Qui 3	5
Curso Qui 5	5
Curso Quiquam 7	5
Curso Distincto 9	4
Curso Expedita 10	4
Curso Est 14	5
Curso Magnam 16	5
Curso Molitia 18	4
Curso Maiores 21	5
Curso Aliquid 23	4
Curso Omnis 24	5
Curso Id 25	4
Curso Voluptas 26	4
Curso Reicendis 28	5
Curso Cumque 31	5
Curso Veniam 33	4
Curso A 36	5
Curso Dolore 37	5
Curso Vel 38	4
Curso Dolor 49	4
Curso Molestias 51	4

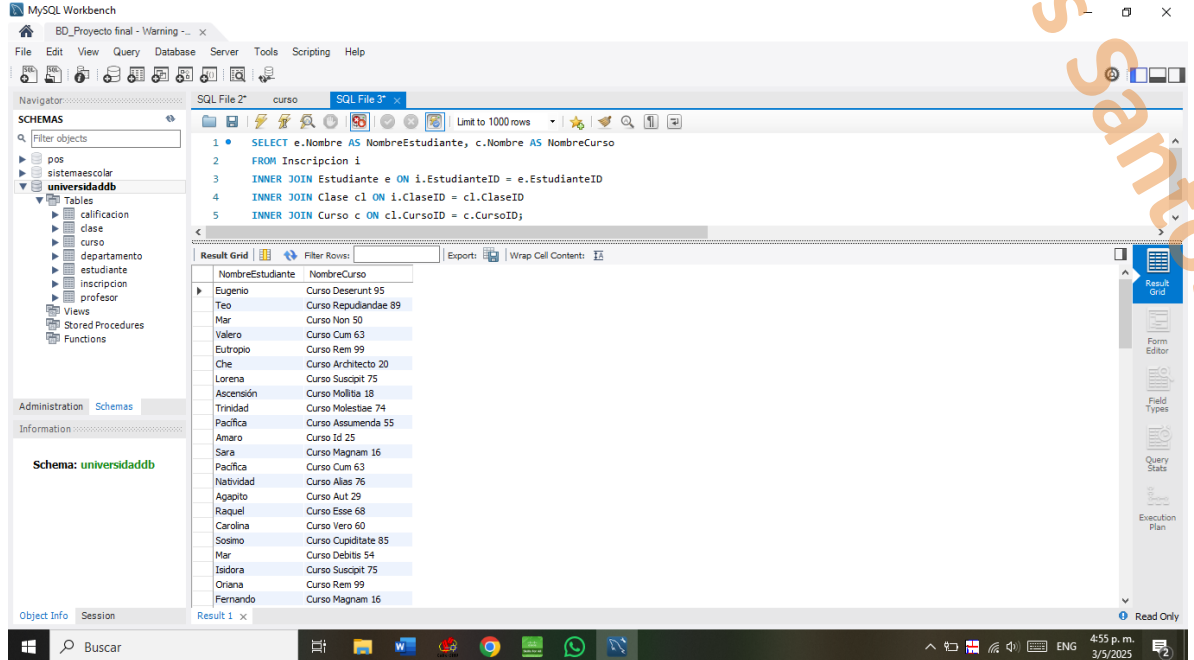
Object Info Session

Curso 2 x

Read Only

4:53 p.m. 3/5/2025

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



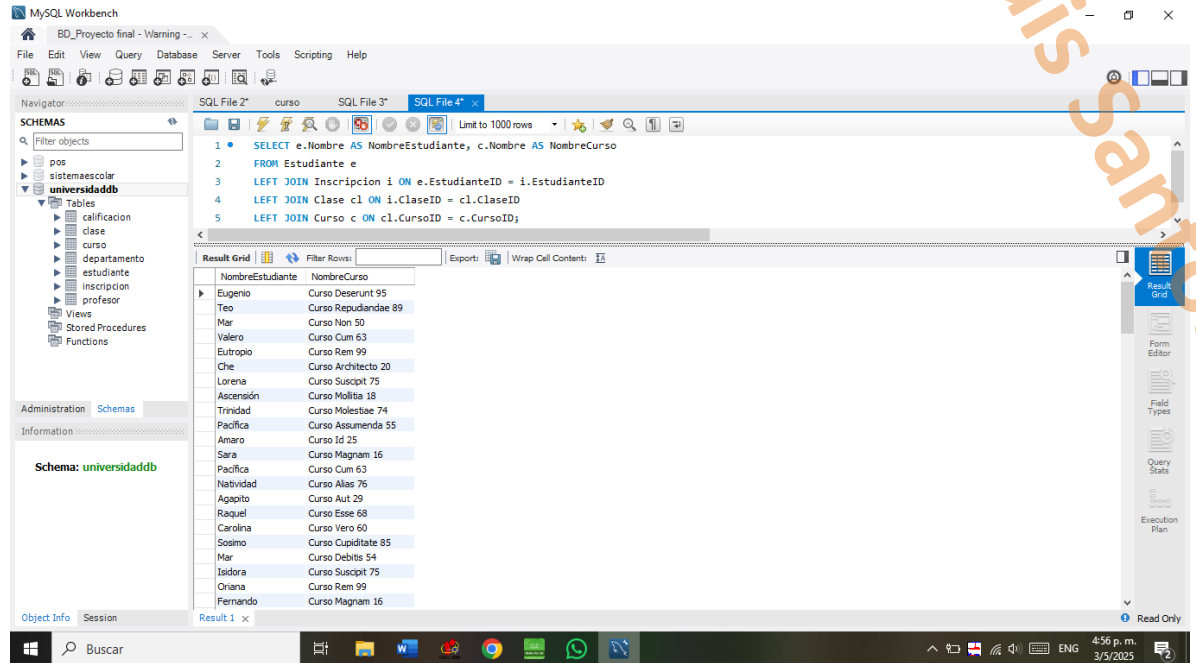
The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 SELECT e.Nombre AS NombreEstudiante, c.Nombre AS NombreCurso
2 FROM Inscripcion i
3 INNER JOIN Estudiante e ON i.EstudianteID = e.EstudianteID
4 INNER JOIN Clase c1 ON i.ClaseID = c1.ClaseID
5 INNER JOIN Curso c ON c1.CursoID = c.CursoID;
```

The query results are displayed in the Result Grid, showing a list of students and the courses they are enrolled in:

NombreEstudiante	NombreCurso
Eugenio	Curso Deserunt 95
Teo	Curso Repudandae 89
Mar	Curso Non 50
Valero	Curso Cum 63
Eutropio	Curso Rem 99
Che	Curso Architecto 20
Lorena	Curso Suscipit 75
Ascensio	Curso Molito 18
Trinidad	Curso Molestae 74
Pacifica	Curso Assumenda 55
Amaro	Curso Id 25
Sara	Curso Magnam 16
Pacifica	Curso Cum 63
Natividad	Curso Alias 76
Agapito	Curso Aut 29
Raquel	Curso Esse 68
Carolina	Curso Vero 60
Sosimo	Curso Cupiditate 85
Mar	Curso Debitis 54
Isidora	Curso Suscipit 75
Orlana	Curso Rem 99
Fernando	Curso Magnam 16

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



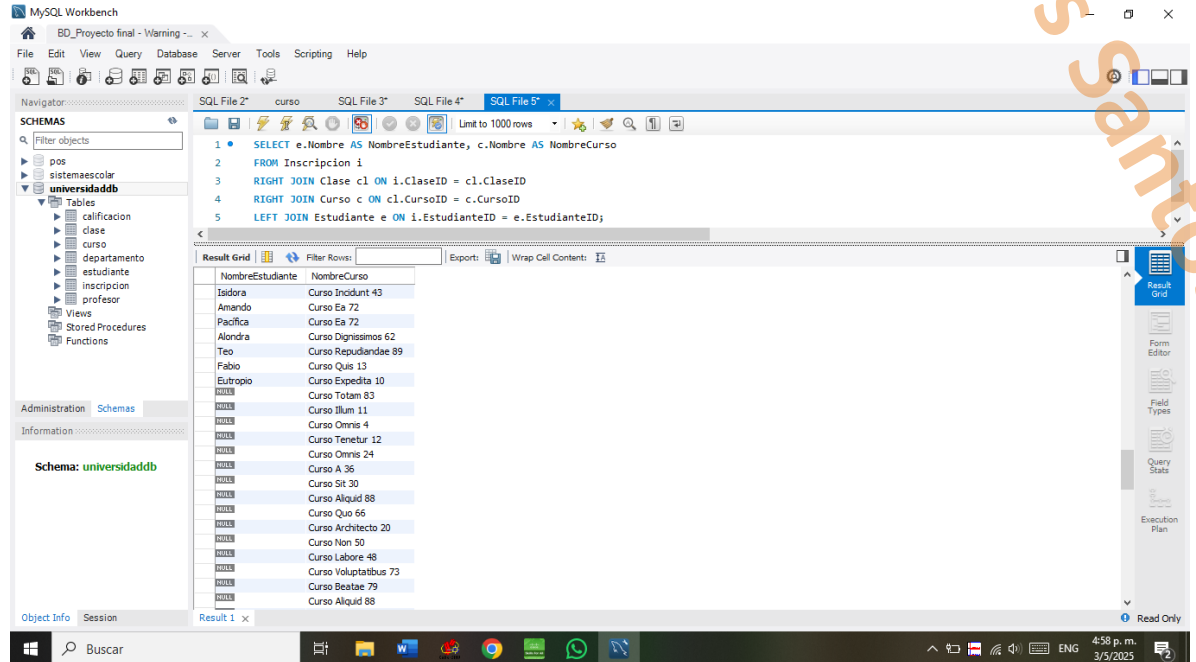
The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 SELECT e.Nombre AS NombreEstudiante, c.Nombre AS NombreCurso
2 FROM Estudiante e
3 LEFT JOIN Inscripcion i ON e.EstudianteID = i.EstudianteID
4 LEFT JOIN Clase c1 ON i.ClaseID = c1.ClaseID
5 LEFT JOIN Curso c ON c1.CursoID = c.CursoID;
```

The Results Grid displays the following data:

NombreEstudiante	NombreCurso
Eugenio	Curso Deserunt 95
Tico	Curso Repudiandae 89
Mar	Curso Non 50
Valero	Curso Cum 63
Eutropio	Curso Rem 99
Che	Curso Architecto 20
Lorena	Curso Suscipit 75
Ascensión	Curso Molitit 18
Trinidad	Curso Molestae 74
Pacifica	Curso Assumenda 55
Amaro	Curso Id 25
Sara	Curso Magnam 16
Pacifica	Curso Cum 63
Natividad	Curso Alias 76
Agapito	Curso Aut 29
Raquel	Curso Esse 68
Carolina	Curso Vero 60
Sosimo	Curso Cupiditate 85
Mar	Curso Debitis 54
Isidora	Curso Suscipit 75
Oriana	Curso Rem 99
Fernando	Curso Magnam 16

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



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 SELECT e.Nombre AS NombreEstudiante, c.Nombre AS NombreCurso
2 FROM Inscripcion i
3 RIGHT JOIN Clase c1 ON i.ClaseID = c1.ClaseID
4 RIGHT JOIN Curso c ON c1.CursoID = c.CursoID
5 LEFT JOIN Estudiante e ON i.EstudianteID = e.EstudianteID;
```

The Results Grid displays the following data:

NombreEstudiante	NombreCurso
Isidora	Curso Incidunt 43
Amando	Curso Ea 72
Pacifica	Curso Ea 72
Alondra	Curso Dignissimos 62
Teo	Curso Repudiandae 89
Fabio	Curso Quis 13
Eutropio	Curso Expedita 10
NULL	Curso Totam 83
NULL	Curso Illum 11
NULL	Curso Omnis 4
NULL	Curso Tenetur 12
NULL	Curso Omnis 24
NULL	Curso A 36
NULL	Curso Sit 30
NULL	Curso Aliquid 88
NULL	Curso Quo 66
NULL	Curso Architecto 20
NULL	Curso Non 50
NULL	Curso Labore 48
NULL	Curso Voluptatibus 73
NULL	Curso Beatae 79
NULL	Curso Aliquid 88



6. Contar cuántos estudiantes hay por departamento.

MySQL Workbench

BD_Proyecto final - Warning - ...

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

pos

sistemaescolar

universidaddb

Tables

calificacion

clase

curso

departamento

estudiante

inscripcion

profesor

Views

Stored Procedures

Functions

Administration Schemas

Information

Schema: universidaddb

SQL File 2* curso SQL File 3* SQL File 4* SQL File 5* SQL File 6*

1 SELECT d.Nombre AS NombreDepartamento, COUNT(e.EstudianteID) AS CantidadEstudiantes

2 FROM Estudiante e

3 INNER JOIN Departamento d ON e.DepartamentoID = d.DepartamentoID

4 GROUP BY d.Nombre;

5

Result Grid

NombreDepartamento	CantidadEstudiantes
Biología	14
Economía	15
Filosofía	21
Historia	13
Ingeniería	19
Matemáticas	18

Object Info Session Result 1 x

4:59 p.m. 3/5/2025

7. Calcular el promedio de notas por estudiante.

MySQL Workbench

BD_Proyecto final - Warning - ...

File Edit View Query Database Server Tools Scripting Help

Navigator: Filter objects

- pos
- sistemaescolar
- universidaddb
 - Tables
 - calificacion
 - clase
 - curso
 - departamento
 - estudiante
 - inscripcion
 - profesor
 - Views
 - Stored Procedures
 - Functions

Administration Schemas

Information

Schema: universidaddb

SQL File 2* curso SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7*

```
1 SELECT e.Nombre, e.Apellido, AVG(ca.Nota) AS PromedioNota
2 FROM Estudiante e
3 INNER JOIN Incripcion i ON e.EstudianteID = i.EstudianteID
4 INNER JOIN Calificacion ca ON i.IncripcionID = ca.IncripcionID
5 GROUP BY e.EstudianteID, e.Nombre, e.Apellido;
```

Result Grid

Nombre	Apellido	PromedioNota
Che	Tovar	71.710000
Santiago	España	79.900000
Nacho	Marito	88.270000
Anna	Bianes	76.770000
José	Pérez	76.670000
Lorena	Torreclla	68.250000
Reyna	Bustamante	74.360000
Violeta	Arranz	93.690000
Ascensión	Llopi	94.895000
Albano	Perera	86.343333
Consuelo	Pujol	79.810000
Amaro	Aramburu	72.440000
Leocadio	Acado	71.140000
Leonco	Acosta	85.945000
Ángel	Herranz	62.490000
Mateo	Vélez	78.220000
Eutropio	Mayo	87.330000
Amando	Lerma	79.395000
Sosimo	Luna	99.550000
Eligio	Oliver	76.015000
Adolfo	Rebollo	86.010000
Rico	Cabo	85.920000

Object Info Session Result 1 x

5:01 p.m. 3/5/2025

8. Mostrar la nota máxima y mínima por clase.

MySQL Workbench

BD_Proyecto final - Warning - ...

File Edit View Query Database Server Tools Scripting Help

Navigator: curso SQL File 2* SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8*

SCHEMAS

Filter objects

pos

sistemaescolar

universidaddb

Tables

calificacion

clase

curso

departamento

estudiante

inscripcion

profesor

Views

Stored Procedures

Functions

Administration Schemas

Information

Schema: universidaddb

Object Info Session

Result 1 x

SQL File 2* curso

```
1 SELECT c1.ClaseID, MAX(ca.Nota) AS NotaMaxima, MIN(ca.Nota) AS NotaMinima
2 FROM Clase c1
3 LEFT JOIN Inscripcion i ON c1.ClaseID = i.ClaseID
4 LEFT JOIN Calificacion ca ON i.InscripcionID = ca.InscripcionID
5 GROUP BY c1.ClaseID;
```

Result Grid

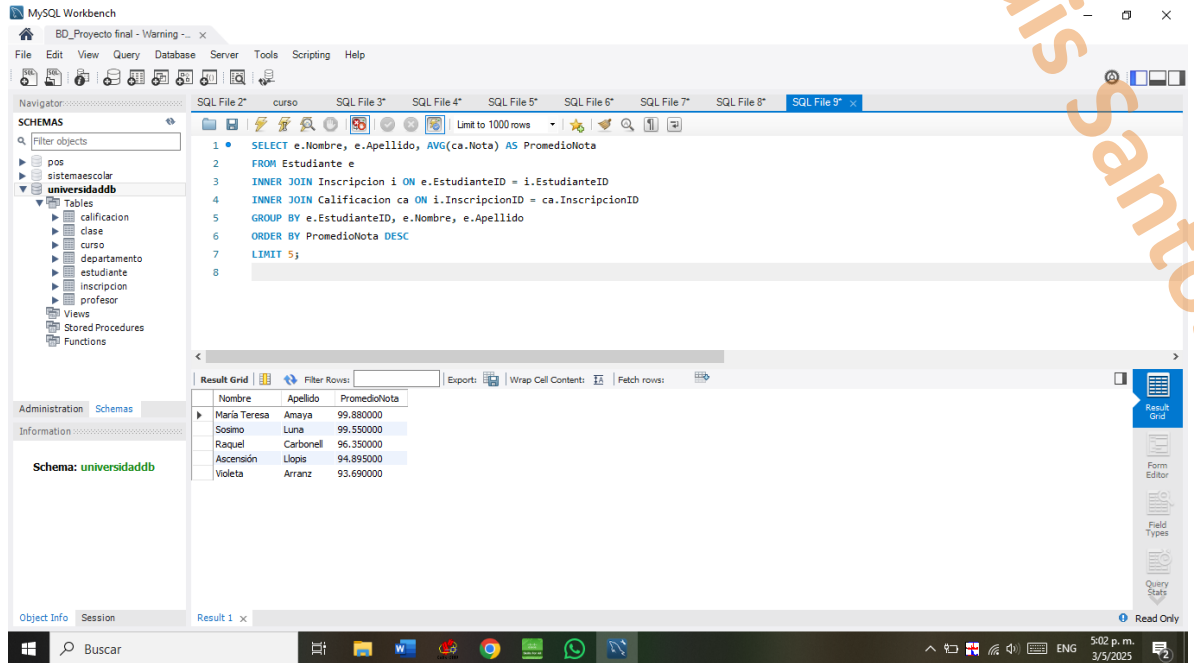
ClaseID	NotaMaxima	NotaMinima
1	NULL	NULL
2	NULL	NULL
3	95.39	70.30
4	82.57	82.57
5	NULL	NULL
6	NULL	NULL
7	82.72	82.72
8	NULL	NULL
9	73.19	63.06
10	67.96	67.96
11	80.39	61.38
12	74.60	74.60
13	NULL	NULL
14	NULL	NULL
15	NULL	NULL
16	94.78	94.78
17	76.77	76.77
18	89.49	85.92
19	63.67	63.67
20	84.61	64.20
21	NULL	NULL
22	71.71	71.71

Export: Wrap Cell Contents

Read Only

502 p.m. 3/5/2025

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



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 SELECT e.Nombre, e.Apellido, AVG(ca.Nota) AS PromedioNota
2 FROM Estudiante e
3 INNER JOIN Inscripcion i ON e.EstudianteID = i.EstudianteID
4 INNER JOIN Calificacion ca ON i.InscripcionID = ca.InscripcionID
5 GROUP BY e.EstudianteID, e.Nombre, e.Apellido
6 ORDER BY PromedioNota DESC
7 LIMIT 5;
```

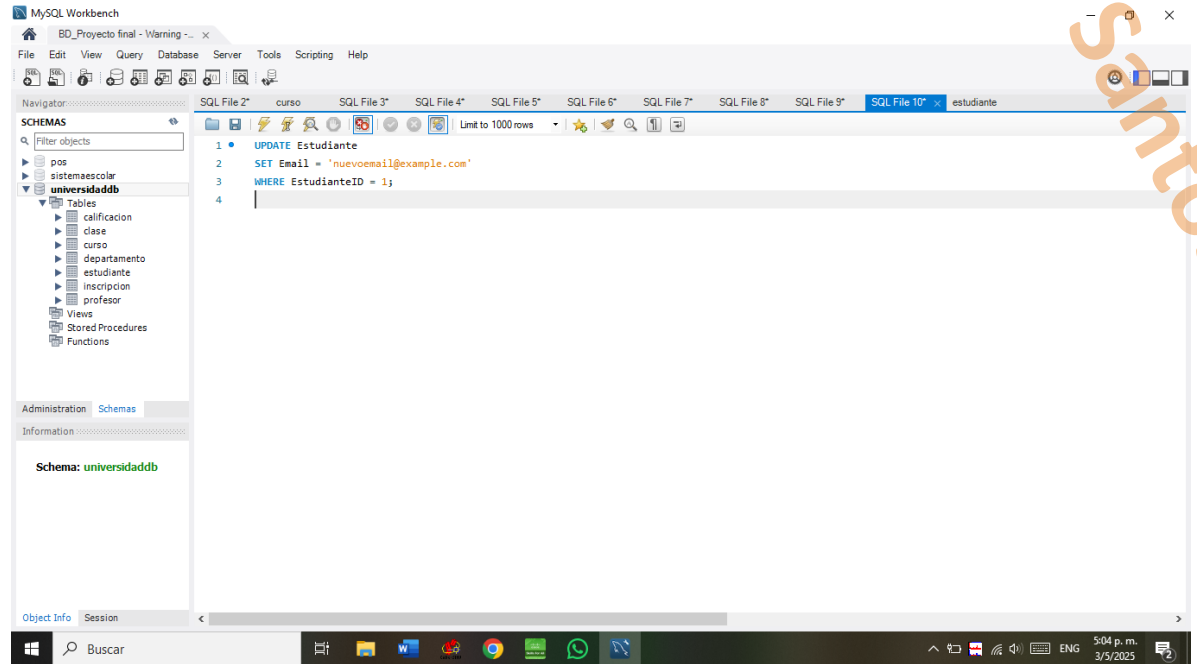
The Results window displays the following data:

Nombre	Apellido	PromedioNota
Maria Teresa	Anaya	99.880000
Sosimo	Luna	99.550000
Raquel	Carbonell	96.350000
Ascension	Llopis	94.895000
Violeta	Arranz	93.690000

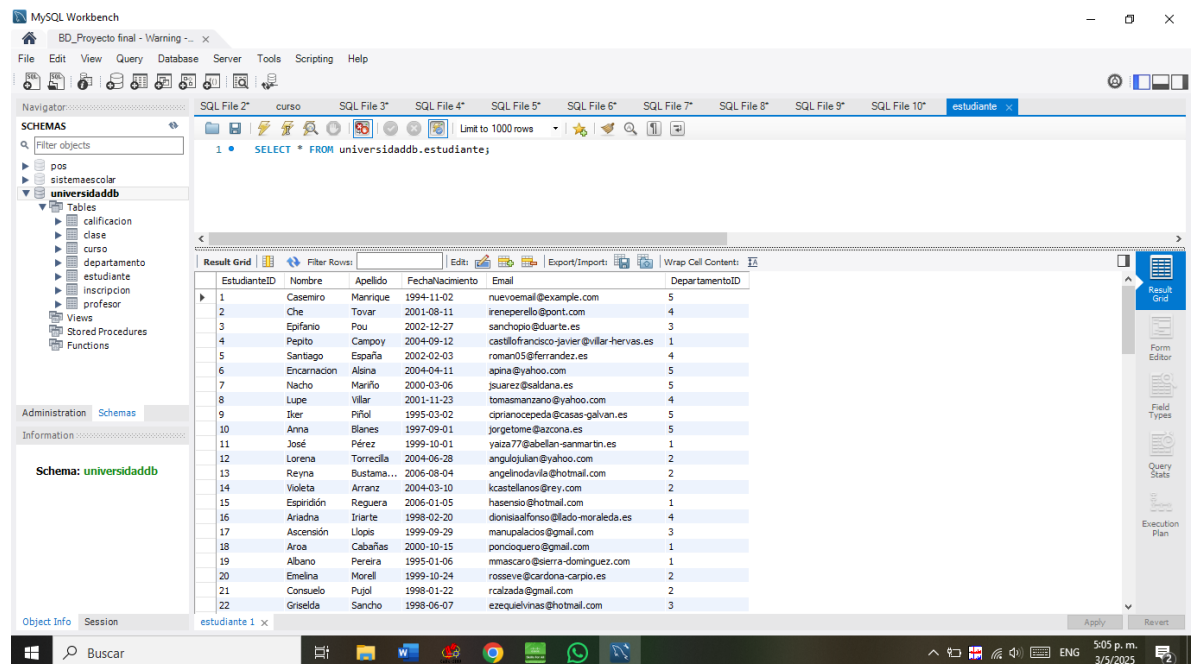
The interface also shows the database schema on the left, with the 'universidaddb' database selected. The bottom status bar indicates the system time as 5:02 p.m. on 3/5/2025.

Actualizar y borrar

10. Cambiar el correo electrónico de un estudiante

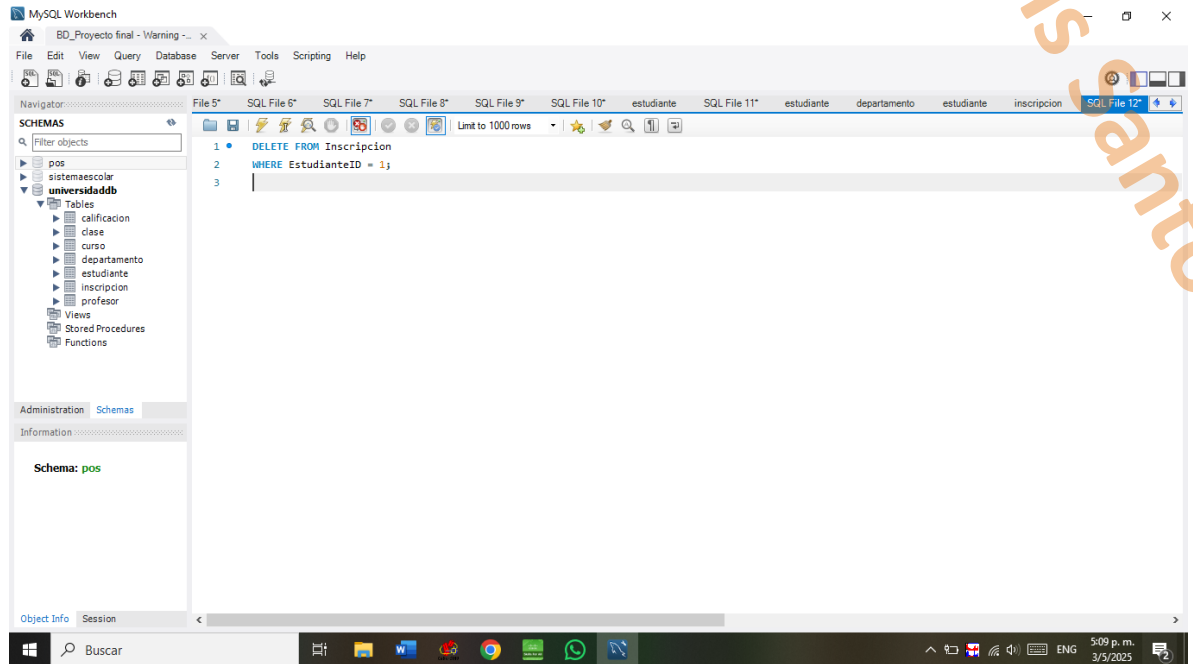


Se actualizo el ID 1

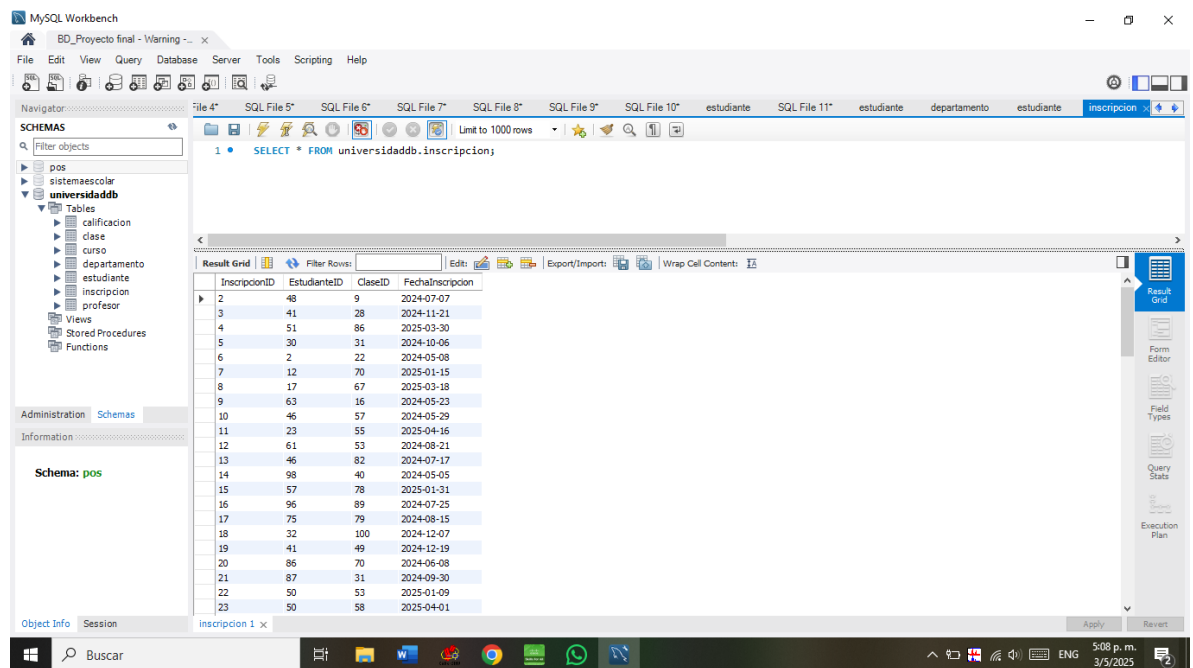




11. Eliminar una inscripción de un estudiante



Podemos ver como el ID 1 ya no esta en el listado





Como entregar:

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

https://docs.google.com/spreadsheets/d/1Aok9n7cke_G16fXLlgS_V_WXCdAkiCCPOu8bgR63zUo/edit?gid=0#gid=0

Fecha de entrega final: Lunes 5 de Mayo 2025

Juan Luis Santos