

EXAMEN FINAL DE UNIDAD 4

**ALUMNO: YASSER ABRAHAM IRACHETA
MEDRANO**

NO. CONTROL: 23490394

MATERIA: FUNDAMENTO DE BASE DE DATOS

**PROFESOR: JOSE RAMON BOGARIN
VALENZUELA**

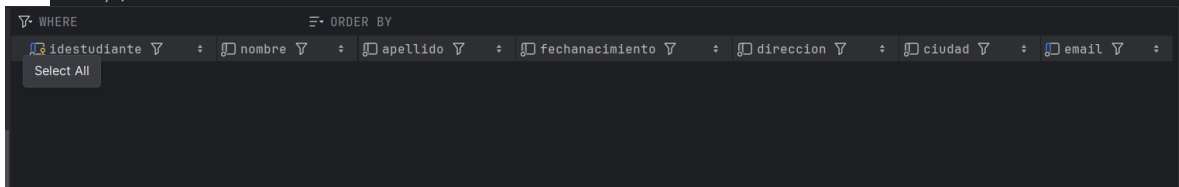
1. Entendiendo las Entidades: Identificar los elementos principales que necesita gestionar la universidad y la información relevante para cada uno.

- **Tabla: Estudiantes**

1. IDEstudiante (Clave Principal)
2. Nombre
3. Apellido
4. FechaNacimiento
5. Direccion
6. Ciudad

7. Email

```
2. create table Estudiantes
(
    IDEstudiante SERIAL primary key not null,
    Nombre VARCHAR(100)not null ,
    apellido VARCHAR(100)NOT NULL ,
    FechaNacimiento VARCHAR(100)NOT NULL,
    Direccion VARCHAR(100)NOT NULL,
    Ciudad VARCHAR (100)NOT NULL,
    Email VARCHAR(100) not null unique
);
```



- **Tabla: Cursos**

- **IDCurso (Clave Principal)**
- **NombreCurso**
- **Descripcion**
- **Creditos**
- **Semestre**
- **IDDepartamento (Clave Foránea, referencia a la tabla Departamentos)**

WHERE	ORDER BY
idcurso	nombrecurso
descripcion	creditos
semestre	iddepartamento

• Tabla: Inscripciones

- IDInscripcion (Clave Principal)
- IDEstudiante (Clave Foránea, referencia a la tabla Estudiantes)
- IDCurso (Clave Foránea, referencia a la tabla Cursos)
- FechaInscripcion

- Calificacion

idinscripciones	idestudiante	idcurso	fechainscripcion	calificacion
-----------------	--------------	---------	------------------	--------------

• Tabla: Profesores

- IDProfesor (Clave Principal)
- Nombre
- Apellido
- Titulo
- IDDepartamento (Clave Foránea, referencia a la tabla Departamentos)

WHERE	ORDER BY
idprofesor	nombre
apellido	titulo
iddepartamento	

• Tabla: Departamentos

- IDDepartamento (Clave Principal)
- NombreDepartamento

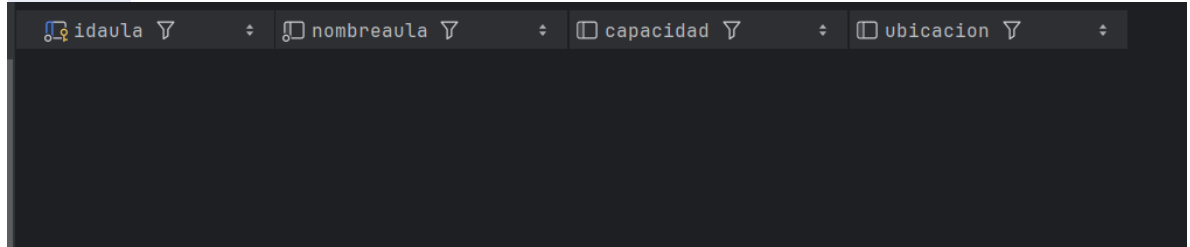
- Edificio

iddeparta...	nombredep...	edificio
--------------	--------------	----------

• Tabla: Aulas

- IDAula (Clave Principal)
- NombreAula
- Capacidad

- Ubicación

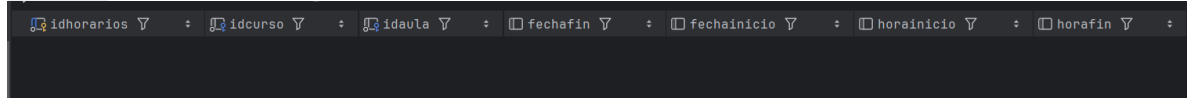


idaula	nombreaula	capacidad	ubicacion
--------	------------	-----------	-----------

- **Tabla: Horarios**

- IDHorario (Clave Principal)
- IDCurso (Clave Foránea, referencia a la tabla Cursos)
- IDAula (Clave Foránea, referencia a la tabla Aulas)
- FechaInicio
- FechaFin
- HoraInicio

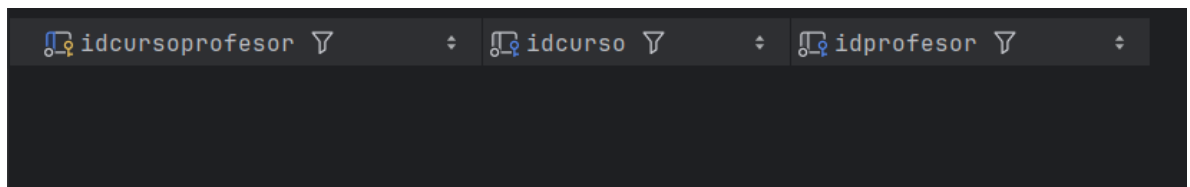
- HoraFin



idhorarios	idcurso	idaula	fechafin	fechainicio	horainicio	horafin
------------	---------	--------	----------	-------------	------------	---------

- **Tabla Intermedia: CursosProfesores** (Relación Muchos a Muchos entre Cursos y Profesores)

- IDCursoProfesor (Clave Principal)
- IDCurso (Clave Foránea, referencia a la tabla Cursos)
- IDProfesor (Clave Foránea, referencia a la tabla Profesores)

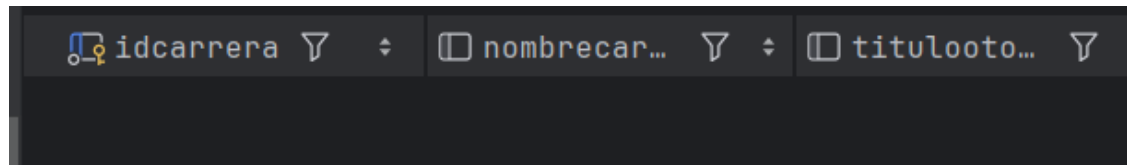


idcursoprofesor	idcurso	idprofesor
-----------------	---------	------------

- **Tabla: ProgramasEstudio**

- IDPrograma (Clave Principal)
- NombrePrograma
- DescripcionPrograma

- IDCarrera (Clave Principal)
- NombreCarrera
- TituloOtorgado



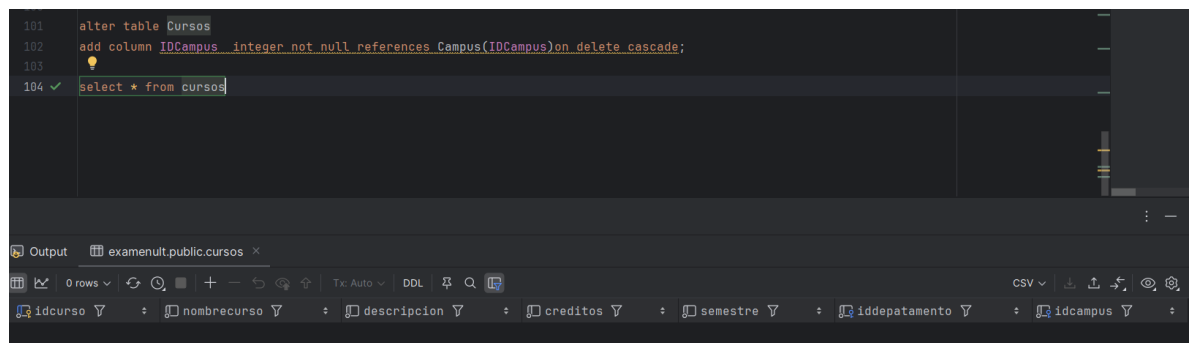
- Agregar una relación de muchos a muchos entre Estudiantes y Carreras

○ Modificar Tablas:

- En la tabla Estudiantes, agregar una clave foránea IDCarrera que haga referencia a la tabla Carreras.



- En la tabla Cursos, agregar una columna IDCampus como clave foránea, referenciando la tabla Campus.



- Modificar la tabla Profesores para incluir un campo Email

```

106 alter table Profesores
107 add column Email varchar(100) not null;
108
109 ✓ select * from Profesores;

```

Output examenult.public.profesores x

0 rows v

idprofesor	nombre	apellido	titulo	iddepartamento	email
------------	--------	----------	--------	----------------	-------

- **Eliminar Tablas/Campos**

- Eliminar la columna Ciudad de la tabla Estudiantes

```

110
111 alter table Estudiantes
112 drop column Ciudad_
113 ✓ select * from Estudiantes
114

```

Output examenult.public.estudiantes ×

0 rows ▾ 🔍 + - ↶ ↷ Tx: Auto ▾ DDL 🔍 📄

idestudiante ▾ : nombre ▾ : apellido ▾ : fechanacimiento ▾ : direccion ▾ : email ▾ : idcarrera ▾ : CSV ▾

Eliminar la tabla Aulas

DROP TABLE Aulas;

- **Insertando Datos:** Insertar datos de ejemplo en las tablas para representar la información de la universidad.

```
o -- Insertar en Departamentos
  INSERT INTO Departamento (NombreDepartamento, Edificio)
  VALUES
    ('Ingeniería de Sistemas', 'Edificio A'),
    ('Matemáticas', 'Edificio B'),
    ('Humanidades', 'Edificio C');
```

	iddepartamento	nombredepartamento	edificio
1	1	Ingeniería de Sistemas	Edificio A
2	2	Matemáticas	Edificio B
3	3	Humanidades	Edificio C

```
-- Insertar en Campus
INSERT INTO Campus (NombreCampus, DireccioCampus) VALUES
('Campus Central', 'Av. Universitaria 1000'),
('Campus Norte', 'Calle Ciencia 123');
```

	idcampus	nombrecampus	direcciocampus
1	1	Campus Central	Av. Universitaria 1000
2	2	Campus Norte	Calle Ciencia 123

```
-- Insertar en Carreras
INSERT INTO Carreras (NombreCarrera, TituloOtorgado) VALUES
('Ingeniería de Sistemas', 'Ingeniero de Sistemas'),
('Matemáticas Aplicadas', 'Licenciado en Matemáticas');
```

	idcarrera	nombrecarrera	titulootorgado
1	1	Ingeniería de Sistemas	Ingeniero de Sistemas
2	2	Matemáticas Aplicadas	Licenciado en Matemáticas

```
-- Insertar en Estudiante
INSERT INTO Estudiantes (Nombre, Apellido, FechaNacimiento,
Direccion, Email, IDCarrera) VALUES
('Juan', 'Pérez', '2000-05-15', 'Calle 123',
'juan.perez@example.com', 1),
('María', 'López', '1999-08-23', 'Av. Principal 456',
'maria.lopez@example.com', 2);
```

idestudiante	nombre	apellido	fechanacimiento	direccion	email	idcarrera
1	Juan	Pérez	2000-05-15	Calle 123	juan.perez@example.com	1
2	María	López	1999-08-23	Av. Principal 456	maria.lopez@example.com	2

```
-- Insertar en Profesor
INSERT INTO Profesores (Nombre, Apellido, Titulo,
IDDepartamento, Email) VALUES
('Carlos', 'Ramírez', 'Dr. en Computación', 1,
'carlos.ramirez@uni.edu'),
('Ana', 'Gómez', 'MSc. en Matemáticas', 2,
'ana.gomez@uni.edu');
```

idprofesor	nombre	apellido	titulo	iddepartamento	email
1	Carlos	Ramírez	Dr. en Computación	1	carlos.ramirez@uni.edu
2	Ana	Gómez	MSc. en Matemáticas	2	ana.gomez@uni.edu


```
-- Insertar en Curso
INSERT INTO Cursos (NombreCurso, Descripcion, Creditos,
Semestre, IDDepatamento, IDCampus) VALUES
('Estructuras de Datos', 'Curso avanzado de estructuras',
'4', '2025-1', 1, 1),
('Álgebra Lineal', 'Espacios vectoriales y matrices', '3',
'2025-1', 2, 1);
```

idcurso	nombrecurso	descripcion	creditos	semestre	iddepartamento	idcampus
1	1 Estructuras de Datos	Curso avanzado de estructuras	4	2025-1	1	
2	2 Álgebra Lineal	Espacios vectoriales y matrices	3	2025-1	2	

```
-- Insertar en Inscripcion
INSERT INTO Inscripciones (IDEstudiante, IDCurso,
Calificacion) VALUES
(1, 1, 85),
(2, 2, 90);
```

idinscripciones	idestudiante	idcurso	fechainscripcion	calificacion
1	1	1	1 2025-05-23	85
2	2	2	2 2025-05-23	90

```
INSERT INTO Aula (NombreAula, Capacidad, Ubicacion)
VALUES
('Aula 101', '40', 'Edificio A - Primer piso'),
('Aula 102', '35', 'Edificio A - Primer piso'),
('Laboratorio de Computo', '25', 'Edificio B - Segundo
piso'),
('Taller de Electrónica', '20', 'Edificio C - Planta
baja'),
('Aula Magna', '100', 'Edificio Principal - Auditorio'),
('Sala de Conferencias', '60', 'Edificio D - Tercer piso');
```

idaula	nombreaula	capacidad	ubicacion
1	1 Aula 101	40	Edificio A - Primer piso
2	2 Aula 102	35	Edificio A - Primer piso
3	3 Laboratorio de Computo	25	Edificio B - Segundo piso
4	4 Taller de Electrónica	20	Edificio C - Planta baja
5	5 Aula Magna	100	Edificio Principal - Audito...
6	6 Sala de Conferencias	60	Edificio D - Tercer piso

```
-- Insertar en Horario
INSERT INTO Horario(IDCurso, IDAula, Fechainicio, Fechafin,
Horainicio, Horafin) VALUES
(1, 2, '2025-03-01', '2025-06-30', '08:00', '10:00'),
(2, 3, '2025-03-01', '2025-06-30', '10:00', '12:00');
```

idhorarios	idcurso	idaula	fechainicio	fechafin	horainicio	horafin
1	1	1	1 2025-06-30	2025-03-01	08:00:00	10:00:00
2	2	2	2 2025-06-30	2025-03-01	10:00:00	12:00:00

```
-- Insertar en CursosProfesor
INSERT INTO CursosProfesores (IDCurso, IDProfesor) VALUES
(1, 1),
(2, 2);
```

	idcursoprofesor	idcurso	idprofesor
1	1	1	1
2	2	2	2

```
-- Insertar en ProgramasEstudios
INSERT INTO ProgramasEstudio (NombrePrograma,
Descripcionprograma) VALUES
('Ingeniería de Software', 'Formación en desarrollo y
análisis de software'),
('Matemáticas Puras', 'Estudios teóricos en matemática
pura');
```

	idprograma	nombreprograma	descripcionprograma
1	1	Ingeniería de Software	Formación en desarrollo y análisis de software
2	2	Matemáticas Puras	Estudios teóricos en matemática pura

```
-- Insertar en ProgramasCurso
INSERT INTO ProgramasCursos (IDPrograma, IDCurso) VALUES
(1, 1),
(2, 2);
```

	idprogramacurso	idprograma	idcurso
1	1	1	1
2	2	2	2

Actualizando Datos: Actualizar la información existente en las tablas para reflejar cambios o correcciones.

1. Actualizar dirección de un estudiante

182	✓	<code>select * from estudiante;</code>	
183			
184		<code>UPDATE Estudiante</code>	
185		<code>SET Direccion = 'Calle Actualizada 456'</code>	
186		<code>WHERE IDEstudiante = 1;</code>	
187			

Output examen4ultimo.public.estudiante						
	idestudiante	nombre	apellido	fechanacimiento	direccion	email
1	2	María	López	1999-08-23	Av. Principal 456	maria.lopez@example.com
2	1	Juan	Pérez	2000-05-15	Calle Actualizada 456	juan.perez@example.com

2. Actualizar el correo electrónico de un profesor

188		<code>UPDATE Profesor</code>	
189		<code>SET Email = 'carlos.ramirez.actualizado@uni.edu'</code>	
190		<code>WHERE IDProfesor = 1;</code>	
191			
192			

Output examen4ultimo.public.profesor						
	idprofesor	nombre	apellido	titulo	iddepartamento	email
1	2	Ana	Gómez	MSc. en Matemáticas		2 ana.gomez@uni.edu
2	1	Carlos	Ramírez	Dr. en Computación		1 carlos.ramirez.actualizado@uni.edu

3. Corregir un título otorgado por una carrera

192		<code>UPDATE Carreras</code>	
193		<code>SET TituloOtorgado = 'Ingeniero en Sistemas'</code>	
194		<code>WHERE IDCarrera = 1;</code>	
195			
196	✓	<code>select *from carreras</code>	
197			

Output examen4ultimo.public.carreras			
	idcarrera	nombrecarrera	tituloOtorgado
1	2	Matemáticas Aplicadas	Licenciado en Matemáticas
2	1	Ingeniería de Sistemas	Ingeniero en Sistemas

Eliminando Datos: Eliminar registros de las tablas que ya no sean relevantes.

Eliminar un aula que ya no existe

```
198 DELETE FROM Aula
199 WHERE NombreAula = 'Taller de Electrónica';
200
201 ✓ select * from aula*
```

Output examen4ultimo.public.aula x

	idaula	nombreaula	capacidad	ubicacion
1	1	Aula 101	40	Edificio A - Primer piso
2	2	Aula 102	35	Edificio A - Primer piso
3	3	Laboratorio de Computo	25	Edificio B - Segundo piso
4	5	Aula Magna	100	Edificio Principal - Auditorio
5	6	Sala de Conferencias	60	Edificio D - Tercer piso

Realizando Consultas (Búsquedas): Formular y ejecutar consultas para obtener información específica de la base de datos.

Selección Básica: Muestra todos los nombres y apellidos de los estudiantes.

202

203 ✓ `Select e.nombre, e.apellido`

204 `from estudiante e`

205

206

207

Output examen4ultimo.public.estudiante x

2 rows v

	nombre	apellido
1	María	López
2	Juan	Pérez

1. **Cláusula WHERE:** Encuentra todos los cursos que tienen 3 créditos.

205 ✓ `SELECT *`

206 `FROM Curso`

207 `WHERE Creditos = '3';`

208

209

210

211

Output examen4ultimo.public.curso x

1 row v

idcurso	nombrecurso	descripcion	creditos	semestre	iddepartamento	idcampus
2	Álgebra Lineal	Espacios vectoriales y matrices	3	2025-1	2	1

INNER JOIN: Obtén una lista que muestre el nombre del estudiante y el nombre del curso en el que está inscrito.

```
208
209 ✓ SELECT Estudiante.Nombre AS NombreEstudiante, Curso.NombreCurso AS NombreCurso
210 FROM Inscripcion
211 INNER JOIN Estudiante 1..n<->1: ON Inscripcion.IDEstudiante = Estudiante.IDEstudiante
212 INNER JOIN Curso 1..n<->1: ON Inscripcion.IDCurso = Curso.IDCurso;
213
214
215
216
217
```

Output Result 42 x

	nombreestudiante	nombrecurso
1	Juan	Estructuras de Datos
2	Maria	Álgebra Lineal

LEFT JOIN: Muestra todos los estudiantes y, si están inscritos en algún curso, el nombre del curso. Si un estudiante no está inscrito en ningún curso, el campo del nombre del curso debe mostrar un valor que lo indique (ej: NULL).

```
214 ✓ SELECT
215     Estudiante.Nombre AS NombreEstudiante,
216     Curso.NombreCurso AS NombreCurso
217 FROM
218     Estudiante
219 LEFT JOIN Inscripcion 1<->0..n: ON Estudiante.IDEstudiante = Inscripcion.IDEstudiante
220 LEFT JOIN Curso 1..n<->1: ON Inscripcion.IDCurso = Curso.IDCurso;
221
222
```

Output Result 44 x

	nombreestudiante	nombrecurso
1	Juan	Estructuras de Datos
2	Maria	Álgebra Lineal

RIGHT JOIN: Lista todos los cursos y, si tienen estudiantes inscritos, el nombre de los estudiantes. Muestra todos los cursos, incluso si no tienen estudiantes inscritos actualmente.

```
222 ✓ SELECT
223     Curso.NombreCurso AS NombreCurso,
224     Estudiante.Nombre AS NombreEstudiante
225 FROM
226     Inscripcion
227 RIGHT JOIN Curso 0..n<->1: ON Inscripcion.IDCurso = Curso.IDCurso
228 LEFT JOIN Estudiante 1..n<->1: ON Inscripcion.IDEstudiante = Estudiante.IDEstudiante;
229
230
231
232
233
234
```

Output Result 45 ×

2 rows ▾

	nombrecurso ▾	nombreestudiante ▾
1	Estructuras de Datos	Juan
2	Álgebra Lineal	María

GROUP BY y COUNT: Calcula cuántos estudiantes están inscritos en cada curso. Muestra el nombre del curso y la cantidad de estudiantes.

```
230 ✓ SELECT
231     Curso.NombreCurso,
232     COUNT(Inscripcion.IDEstudiante) AS CantidadEstudiantes
233 FROM
234     Curso
235 LEFT JOIN Inscripcion 1<->0..n: ON Curso.IDCurso = Inscripcion.IDCurso
236 GROUP BY
237     Curso.NombreCurso;
238
239
```

Output Result 46 ×

2 rows ▾

	nombrecurso ▾	cantidadestudiantes ▾
1	Álgebra Lineal	1
2	Estructuras de Datos	1

BETWEEN: Encuentra todos los estudiantes que nacieron entre el 1 de enero de 1995 y el 31 de diciembre de 1998.

```
239 ✓ SELECT *
240 FROM Estudiante
241 WHERE FechaNacimiento BETWEEN '1995-01-01' AND '1998-12-31';
242
243
244
245
246
247
```

Output examen4ultimo.public.estudiante x

0 rows

idestudiante	nombre	apellido	fechanacimiento	direccion	email	idcarrera
--------------	--------	----------	-----------------	-----------	-------	-----------

ORDER BY: Muestra todos los cursos ordenados alfabéticamente por su nombre.

```
243 ✓ SELECT *
244 FROM Curso
245 ORDER BY NombreCurso ASC;
246
247
248
249
```

Output examen4ultimo.public.curso x

2 rows

idcurso	nombrecurso	descripcion	creditos	semestre	iddepartamento	idcampus
1	2 Álgebra Lineal	Espacios vectoriales y matrices	3	2025-1	2	1
2	1 Estructuras de Datos	Curso avanzado de estructuras	4	2025-1	1	1

CTE: Crea una tabla de expresión común que liste el número de inscripciones por estudiante. Luego, consulta esta CTE para encontrar los 3 estudiantes con más inscripciones, mostrando el nombre del estudiante y el número de inscripciones.


```
247 WITH InscripcionesPorEstudiante AS (  
249     IDEstudiante,  
250     COUNT(*) AS TotalInscripciones  
251 FROM Inscripcion  
252 GROUP BY IDEstudiante  
253 )  
254  
255 SELECT  
256     E.Nombre,  
257     E.Apellido,  
258     I.TotalInscripciones  
259 FROM InscripcionesPorEstudiante I  
260 JOIN Estudiante E ON I.IDEstudiante = E.IDEstudiante  
261 ORDER BY I.TotalInscripciones DESC  
262 LIMIT 3;  
263
```

Output Result 49 ×

2 rows ▾

	nombre ▾	apellido ▾	totalinscripciones ▾
1	María	López	1
2	Juan	Pérez	1

Consulta Compleja 1: Para cada departamento, muestra el nombre del departamento y el nombre del curso con la mayor cantidad de estudiantes inscritos.

```
274 CursoMaximoPorDepartamento AS (  
275     CI.IDDepatamento,  
276     CI.NombreCurso,  
277     CI.TotalInscritos,  
278     ROW_NUMBER() OVER (PARTITION BY CI.IDDepatamento ORDER BY CI.TotalInscritos DESC)  
279 FROM ConteoInscripciones CI  
280 )  
281  
282  
283 SELECT  
284     D.NombreDepartamento,  
285     CMPD.NombreCurso,  
286     CMPD.TotalInscritos  
287 FROM CursoMaximoPorDepartamento CMPD  
288 JOIN Departamentos D ON CMPD.IDDepatamento = D.IDDepartamento  
289 WHERE CMPD.Rnk = 1;  
290
```

Output Result 50

	nombredepartamento	nombrecurso	totalinscritos
1	Ingeniería de Sistemas	Estructuras de Datos	1
2	Matemáticas	Álgebra Lineal	1

Consulta Compleja 2: Encuentra a los profesores que imparten más de dos cursos, mostrando su nombre, apellido y la cantidad de cursos que imparten.

```
291 ✓ SELECT P.Nombre,P.Apellido,  
292        COUNT(CP.IDCurso) AS CantidadCursos  
293 FROM Profesor P  
294 INNER JOIN CursosProfesor CP 1<->1..n: ON P.IDProfesor = CP.IDProfe  
295 GROUP BY P.IDProfesor, P.Nombre, P.Apellido  
296 HAVING COUNT(CP.IDCurso) > 2;  
297  
298  
299  
300  
301  
302  
303  
304
```

Output Result 51 ×

0 rows ▾

nombre ▾ apellido ▾ cantidadcursos ▾

Consulta Compleja 3: Lista los nombres de los programas de estudio y, para cada programa, el nombre del curso con el promedio de calificación más alto.

```
309 CursoTopPorPrograma AS (  
313     P.PromedioCalificacion,  
314     ROW_NUMBER() OVER (  
315         PARTITION BY P.IDPrograma ORDER BY P.PromedioCalificacion DESC  
316     ) AS Rnk  
317 FROM PromediosPorCurso P  
318 )  
319 SELECT  
320     PE.NombrePrograma,  
321     CTPP.NombreCurso,  
322     CTPP.PromedioCalificacion  
323 FROM CursoTopPorPrograma CTPP  
324 INNER JOIN ProgramasEstudios PE ON CTPP.IDPrograma = PE.IDPrograma  
325 WHERE CTPP.Rnk = 1;  
326  
327
```

Output Result 52 x

2 rows v

	nombreprograma ▼	nombrecurso ▼	promediocalificacion ▼
1	Ingeniería de Software	Estructuras de Datos	85
2	Matemáticas Puras	Álgebra Lineal	90