

Instituto Tecnológico de Mexicali



Ingeniería Sistemas Computacionales

Fundamentos de Base de Datos

Tema:

“Unidad 4 Documentación”

Estudiante:

Andrés Mojica Medina

No. De control: C21490782

Docente:

José Ramón Bogarín Valenzuela

Mexicali, B.C., 23 de Mayo de 2025. 10:00 pm

Planteamiento del Problema

La universidad requiere un sistema de base de datos para gestionar la información relacionada con sus estudiantes, cursos, inscripciones, profesores, departamentos y otros aspectos académicos. El sistema debe permitir almacenar, modificar y consultar información relevante de manera eficiente y organizada, facilitando la administración académica y la toma de decisiones.

El objetivo es diseñar una base de datos relacional que refleje las entidades y relaciones presentes en el entorno universitario y permita realizar operaciones típicas sobre los datos.

Solución Propuesta

Diseño de la Base de Datos

Se propone un modelo relacional con las siguientes tablas principales y sus relaciones:

Tabla	Campos Principales / Relaciones
Estudiantes	IDEstudiante (PK), Nombre, Apellido, FechaNacimiento, Dirección, Email, IDCarrera (FK)
Cursos	IDCurso (PK), NombreCurso, Descripción, Créditos, Semestre, IDDepartamento (FK), IDCampus (FK)
Inscripciones	IDInscripcion (PK), IDEstudiante (FK), IDCurso (FK), FechaInscripción, Calificación
Profesores	IDProfesor (PK), Nombre, Apellido, Título, IDDepartamento (FK), Email
Departamentos	IDDepartamento (PK), NombreDepartamento, Edificio
Campus	IDCampus (PK), NombreCampus, DirecciónCampus
Carreras	IDCarrera (PK), NombreCarrera, TítuloOtorgado
Horarios	IDHorario (PK), IDCurso (FK), FechaInicio, FechaFin, HoraInicio, HoraFin
CursosProfesores	IDCursoProfesor (PK), IDCurso (FK), IDProfesor (FK)
ProgramasEstudio	IDPrograma (PK), NombrePrograma, DescripciónPrograma

ProgramasCurs IDProgramaCurso (PK), IDPrograma (FK), IDCurso (FK)
os

Relaciones y Consideraciones

- La relación entre Estudiantes y Carreras es de muchos a muchos, por lo que se requiere una tabla intermedia.
- Se elimina la tabla Aulas y la columna Ciudad de Estudiantes según indicaciones.
- Se agrega la columna Email en Profesores.
- Cursos están asociados a Campus mediante IDCampus.
- Profesores pertenecen a Departamentos.
- Cursos pueden ser impartidos por varios profesores y viceversa (CursosProfesores).

Creando las Tablas (DDL)

En esta sección se deben insertar capturas de pantalla mostrando la ejecución de los scripts de creación de tablas en el sistema gestor de base de datos.

```

-- Tabla Departamentos
CREATE TABLE Departamentos (
    IDDepartamento SERIAL PRIMARY KEY,
    NombreDepartamento VARCHAR(100),
    Edificio VARCHAR(100)
);

-- Tabla Programas de Estudio
CREATE TABLE ProgramasEstudio (
    IDPrograma SERIAL PRIMARY KEY,
    NombrePrograma VARCHAR(100),
    DescripcionPrograma TEXT
);

-- Tabla Estudiantes
CREATE TABLE Estudiantes (
    IDEstudiante SERIAL PRIMARY KEY,
    Nombre VARCHAR(50),
    Apellido VARCHAR(50),
    FechaNacimiento DATE,
    Direccion TEXT,
    Ciudad VARCHAR(50),
    Email VARCHAR(100)
);

-- Tabla Profesores
CREATE TABLE Profesores (
    IDProfesor SERIAL PRIMARY KEY,
    Nombre VARCHAR(50),
    Apellido VARCHAR(50),
    Titulo VARCHAR(100),
    IDDepartamento INT REFERENCES Departamentos(IDDepartamento)
);

```

```

-- Tabla Cursos
CREATE TABLE Cursos (
    IDCurso SERIAL PRIMARY KEY,
    NombreCurso VARCHAR(100),
    Descripcion TEXT,
    Creditos INT,
    Semestre VARCHAR(10),
    IDDepartamento INT REFERENCES Departamentos(IDDepartamento)
);

-- Tabla Inscripciones
CREATE TABLE Inscripciones (
    IDInscripcion SERIAL PRIMARY KEY,
    IDEstudiante INT REFERENCES Estudiantes(IDEstudiante),
    IDCurso INT REFERENCES Cursos(IDCurso),
    FechaInscripcion DATE,
    Calificacion DECIMAL(3,1)
);

-- Tabla Aulas
CREATE TABLE Aulas (
    IDAula SERIAL PRIMARY KEY,
    NombreAula VARCHAR(10),
    Capacidad INT,
    Ubicacion TEXT
);

```

```

-- Tabla Horarios
✓ CREATE TABLE Horarios (
    IDHorario SERIAL PRIMARY KEY,
    IDCurso INT REFERENCES Cursos(IDCurso),
    IDAula INT REFERENCES Aulas(IDAula),
    FechaInicio DATE,
    FechaFin DATE,
    HoraInicio TIME,
    HoraFin TIME
);

-- Tabla intermedia CursosProfesores (relación muchos a muchos)
✓ CREATE TABLE CursosProfesores (
    IDCursoProfesor SERIAL PRIMARY KEY,
    IDCurso INT REFERENCES Cursos(IDCurso),
    IDProfesor INT REFERENCES Profesores(IDProfesor)
);

-- Tabla intermedia ProgramasCursos (relación muchos a muchos)
✓ CREATE TABLE ProgramasCursos (
    IDProgramaCurso SERIAL PRIMARY KEY,
    IDPrograma INT REFERENCES ProgramasEstudio(IDPrograma),
    IDCurso INT REFERENCES Cursos(IDCurso)
);

```

```

-- Crear tabla Campus
CREATE TABLE Campus (
    IDCampus INT PRIMARY KEY,
    NombreCampus VARCHAR(100),
    DireccionCampus VARCHAR(150)
);

-- Crear tabla Carreras
CREATE TABLE Carreras (
    IDCarrera INT PRIMARY KEY,
    NombreCarrera VARCHAR(100),
    TituloOtorgado VARCHAR(100)
);

-- Crear tabla intermedia EstudiantesCarreras (relación muchos a muchos)
CREATE TABLE EstudiantesCarreras (
    IDEstudiante INT,
    IDCarrera INT,
    PRIMARY KEY (IDEstudiante, IDCarrera),
    FOREIGN KEY (IDEstudiante) REFERENCES Estudiantes(IDEstudiante),
    FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera)
);

```

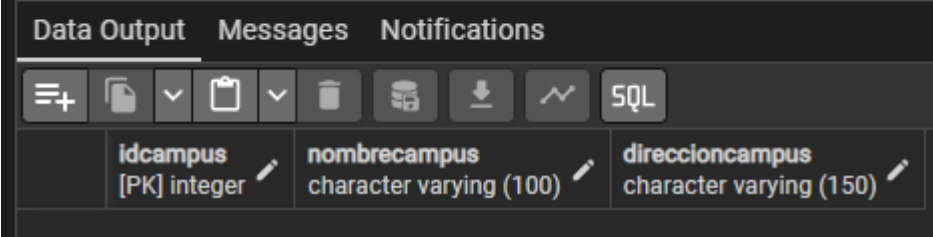
Modificando las Tablas (DDL)

Se realizaron las siguientes modificaciones al esquema inicial:

1. Agregando tablas nuevas:

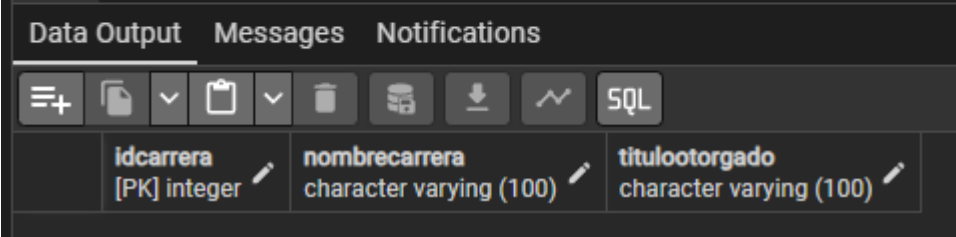
- Se creó la tabla *Campus* con los campos *IDCampus*, *NombreCampus* y *DireccionCampus*.

```
14  -- Crear tabla Campus
15  CREATE TABLE Campus (
16      IDCampus INT PRIMARY KEY,
17      NombreCampus VARCHAR(100),
18      DireccionCampus VARCHAR(150)
19  );
20
```



- Se creó la tabla *Carreras* con los campos *IDCarrera*, *NombreCarrera* y *TituloOtorgado*.

```
21  -- Crear tabla Carreras
22  CREATE TABLE Carreras (
23      IDCarrera INT PRIMARY KEY,
24      NombreCarrera VARCHAR(100),
25      TituloOtorgado VARCHAR(100)
26  );
27
```



2. Relaciones agregadas:

- Se agregó la relación muchos a muchos entre *Estudiantes* y *Carreras*, mediante la tabla intermedia *EstudiantesCarreras*.

```

28 -- Agregar relación muchos a muchos entre Estudiantes y Carreras
29 CREATE TABLE EstudiantesCarreras(
30     IDEstudiante INT,
31     IDCarrera INT,
32     PRIMARY KEY (IDEstudiante, IDCarrera),
33     FOREIGN KEY (IDEstudiante) REFERENCES Estudiantes(IDEstudiante),
34     FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera)
35 );

```

Data Output Messages Notifications

SQL

idestudiante	idcarrera
[PK] integer	[PK] integer

3. Modificación de tablas existentes:

- En la tabla *Estudiantes* se agregó la clave foránea *IDCarrera*.

```

9 -- Agregar IDCarrera a Estudiantes
10 ALTER TABLE Estudiantes ADD COLUMN IDCarrera INT;

```

Data Output Messages Notifications

SQL

	idestudiante	nombre	apellido	fechanacimiento	direccion	ciudad	email	idcarrera
	[PK] integer	character varying (50)	character varying (50)	date	text	character varying (50)	character varying (100)	integer
1	1	Ana	García	2002-05-14	Calle A #123	CDMX	ana.garcia@email.com	[null]
2	2	Luis	Martínez	2001-03-22	Av. Reforma 45	CDMX	luis.martinez@email.com	[null]
3	3	Carla	López	2000-07-19	Col. Centro 202	CDMX	carla.lopez@email.com	[null]
4	4	Pablo	Ramírez	2003-11-30	Sur 7 #33	Puebla	pablo.ramirez@email.com	[null]
5	5	Lucía	Torres	2002-09-10	Norte 18 #8	Monterrey	lucia.torres@email.com	[null]
6	6	Diego	Hernández	2001-01-15	Ote 3 #56	Guadalajara	diego.h@email.com	[null]
7	7	María	Flores	2003-12-01	Col. Roma #77	CDMX	maria.flores@email.com	[null]
8	8	Sofía	Jiménez	2002-06-25	Av. Juárez 99	Querétaro	sofia.jimenez@email.com	[null]
9	9	Carlos	Cruz	2001-10-04	Zaragoza #13	CDMX	carlos.cruz@email.com	[null]
10	10	Elena	Ruiz	2000-08-17	La Paz 55	Oaxaca	elena.ruiz@email.com	[null]
11	11	Jorge	Ortiz	2001-04-03	Niza 21	CDMX	jorge.ortiz@email.com	[null]
12	12	Valeria	Mendoza	2002-02-28	Col. Del Valle 123	CDMX	valeria.mendoza@email.com	[null]
13	13	Hugo	Castillo	2000-12-30	Hidalgo 99	Toluca	hugo.castillo@email.com	[null]
14	14	Fernanda	Silva	2001-05-20	Camino Real 77	CDMX	fernanda.silva@email.com	[null]
15	15	Miguel	Reyes	2002-07-09	Independencia ...	León	miguel.reyes@email.com	[null]

- En la tabla *Cursos* se agregó la columna *IDCampus* como clave foránea.


```

3  -- Agregar Email a Profesores
4  ALTER TABLE Profesores ADD COLUMN Email VARCHAR(100);
5
6  -- Agregar IDCampus a Cursos
7  ALTER TABLE Cursos ADD COLUMN IDCampus INT;

```

Data Output Messages Notifications

	idcurso [PK] integer	nombrecurso character varying (100)	descripcion text	creditos integer	semestre character varying (10)	iddepartamento integer	idcampus integer
1	1	Base de Datos	Diseño y desarrollo de BD	6	4	1	[null]
2	2	Psicología Social	Teoría del comportamiento	4	3	10	[null]
3	3	Cálculo I	Funciones y derivadas	5	1	4	[null]
4	4	Bioquímica	Procesos celulares	6	2	5	[null]
5	5	Física Moderna	Mecánica cuántica	5	5	6	[null]
6	6	Contabilidad	Estados financieros	4	2	3	[null]
7	7	Historia de México	Desde la colonia	3	1	8	[null]
8	8	Ética Profesional	Dilemas éticos	3	3	9	[null]
9	9	Sociología	Estructura social	4	4	2	[null]
10	10	Microbiología	Microorganismos	5	2	5	[null]
11	11	Lógica	Argumentación	4	1	9	[null]
12	12	Lengua y Literatura	Análisis textual	3	2	12	[null]
13	13	Pintura	Técnicas pictóricas	2	5	13	[null]
14	14	Derecho Constitucional	Normas y principios	5	6	15	[null]
15	15	Música Clásica	Historia y análisis	2	4	14	[null]

- En la tabla Profesores se agregó el campo Email.

```

3  -- Agregar Email a Profesores
4  ALTER TABLE Profesores ADD COLUMN Email VARCHAR(100);
5

```

Data Output Messages Notifications

	idprofesor [PK] integer	nombre character varying (50)	apellido character varying (50)	titulo character varying (100)	iddepartamento integer	email character varying (100)
1	1	Laura	Hernández	PhD	1	[null]
2	2	Ricardo	Gómez	MSc	2	[null]
3	3	Patricia	Morales	PhD	3	[null]
4	4	Andrés	Díaz	MSc	4	[null]
5	5	Rosa	Santos	PhD	5	[null]
6	6	Pedro	Cano	MSc	6	[null]
7	7	Gabriela	Medina	PhD	7	[null]
8	8	Ernesto	Beltrán	PhD	8	[null]
9	9	Claudia	Pérez	MSc	9	[null]
10	10	Héctor	Aguilar	PhD	10	[null]
11	11	Julieta	Nava	MSc	11	[null]
12	12	César	Zamora	PhD	12	[null]
13	13	Marcela	Vargas	MSc	13	[null]
14	14	Luis	Nieto	PhD	14	[null]
15	15	Daniela	Mora	MSc	15	[null]

4. Eliminación de campos/tablas:

- Se eliminó la columna Ciudad de la tabla Estudiantes.

```

39  -- Eliminar columna Ciudad de Estudiantes
40  ALTER TABLE Estudiantes DROP COLUMN Ciudad;
41

```

	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fechanacimiento date	direccion text	email character varying (100)
1	1	Ana	García	2002-05-14	Calle A #123	ana.garcia@email.com
2	2	Luis	Martínez	2001-03-22	Av. Reforma 45	luis.martinez@email.com
3	3	Carla	López	2000-07-19	Col. Centro 202	carla.lopez@email.com
4	4	Pablo	Ramírez	2003-11-30	Sur 7 #33	pablo.ramirez@email.com
5	5	Lucía	Torres	2002-09-10	Norte 18 #8	lucia.torres@email.com
6	6	Diego	Hernández	2001-01-15	Ote 3 #56	diego.h@email.com
7	7	María	Flores	2003-12-01	Col. Roma #77	maria.flores@email.com
8	8	Sofía	Jiménez	2002-06-25	Av. Juárez 99	sofia.jimenez@email.com
9	9	Carlos	Cruz	2001-10-04	Zaragoza #13	carlos.cruz@email.com
10	10	Elena	Ruiz	2000-08-17	La Paz 55	elena.ruiz@email.com
11	11	Jorge	Ortiz	2001-04-03	Niza 21	jorge.ortiz@email.com
12	12	Valeria	Mendoza	2002-02-28	Col. Del Valle 123	valeria.mendoza@email.com
13	13	Hugo	Castillo	2000-12-30	Hidalgo 99	hugo.castillo@email.com
14	14	Fernanda	Silva	2001-05-20	Camino Real 77	fernanda.silva@email.com
15	15	Miguel	Reyes	2002-07-09	Independencia ...	miguel.reyes@email.com

- Se eliminó por completo la tabla Aulas (la información sobre aulas se eliminó del modelo por requerimiento).

```

42  -- Eliminar tabla Aulas
43  DROP TABLE IF EXISTS Aulas;
44

```

ERROR: no se puede eliminar tabla aulas porque otros objetos dependen de él
restricción «horarios_idaula_fkey» en tabla horarios depende de tabla aulas

SQL state: 2BP01

Detail: restricción «horarios_idaula_fkey» en tabla horarios depende de tabla aulas
Hint: Use DROP ... CASCADE para eliminar además los objetos dependientes.

Aquí se hubiera borrado la tabla aulas pero no se hizo con ese query porque la tabla horarios depende de la tabla aulas

Se realizaron varias modificaciones a la estructura de las tablas existentes con el objetivo de mejorar la organización de los datos y reflejar correctamente las nuevas relaciones del modelo.

1. Agregar columna Email a la tabla Profesores para registrar su correo institucional.

```

3  -- Agregar Email a Profesores
4  ALTER TABLE Profesores ADD COLUMN Email VARCHAR(100);
5

```

Data Output Messages Notifications

	idprofesor [PK] integer	nombre character varying (50)	apellido character varying (50)	titulo character varying (100)	iddepartamento integer	email character varying (100)
1	1	Laura	Hernández	PhD	1	[null]
2	2	Ricardo	Gómez	MSc	2	[null]
3	3	Patricia	Morales	PhD	3	[null]
4	4	Andrés	Díaz	MSc	4	[null]
5	5	Rosa	Santos	PhD	5	[null]
6	6	Pedro	Cano	MSc	6	[null]
7	7	Gabriela	Medina	PhD	7	[null]
8	8	Ernesto	Beltrán	PhD	8	[null]
9	9	Claudia	Pérez	MSc	9	[null]
10	10	Héctor	Aguilar	PhD	10	[null]
11	11	Julieta	Nava	MSc	11	[null]
12	12	César	Zamora	PhD	12	[null]
13	13	Marcela	Vargas	MSc	13	[null]
14	14	Luis	Nieto	PhD	14	[null]
15	15	Daniela	Mora	MSc	15	[null]

2. Agregar columna IDCampus a la tabla Cursos, permitiendo registrar en qué campus se imparte cada curso.

```

3  -- Agregar Email a Profesores
4  ALTER TABLE Profesores ADD COLUMN Email VARCHAR(100);
5
6  -- Agregar IDCampus a Cursos
7  ALTER TABLE Cursos ADD COLUMN IDCampus INT;
8

```

Data Output Messages Notifications

	idcurso [PK] integer	nombrecurso character varying (100)	descripcion text	creditos integer	semestre character varying (10)	iddepartamento integer	idcampus integer
1	1	Base de Datos	Diseño y desarrollo de BD	6	4	1	[null]
2	2	Psicología Social	Teoría del comportamiento	4	3	10	[null]
3	3	Cálculo I	Funciones y derivadas	5	1	4	[null]
4	4	Bioquímica	Procesos celulares	6	2	5	[null]
5	5	Física Moderna	Mecánica cuántica	5	5	6	[null]
6	6	Contabilidad	Estados financieros	4	2	3	[null]
7	7	Historia de México	Desde la colonia	3	1	8	[null]
8	8	Ética Profesional	Dilemas éticos	3	3	9	[null]
9	9	Sociología	Estructura social	4	4	2	[null]
10	10	Microbiología	Microorganismos	5	2	5	[null]
11	11	Lógica	Argumentación	4	1	9	[null]
12	12	Lengua y Literatura	Análisis textual	3	2	12	[null]
13	13	Pintura	Técnicas pictóricas	2	5	13	[null]
14	14	Derecho Constitucional	Normas y principios	5	6	15	[null]
15	15	Música Clásica	Historia y análisis	2	4	14	[null]

3. Agregar columna IDCarrera a la tabla Estudiantes para relacionarlos con sus respectivas carreras.

```

9  -- Agregar IDCarrera a Estudiantes
10 ALTER TABLE Estudiantes ADD COLUMN IDCarrera INT;

```

	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fechanacimiento date	direccion text	ciudad character varying (50)	email character varying (100)	idcarrera integer
1	1	Ana	García	2002-05-14	Calle A #123	CDMX	ana.garcia@email.com	[null]
2	2	Luis	Martínez	2001-03-22	Av. Reforma 45	CDMX	luis.martinez@email.com	[null]
3	3	Carla	López	2000-07-19	Col. Centro 202	CDMX	carla.lopez@email.com	[null]
4	4	Pablo	Ramírez	2003-11-30	Sur 7 #33	Puebla	pablo.ramirez@email.com	[null]
5	5	Lucía	Torres	2002-09-10	Norte 18 #8	Monterrey	lucia.torres@email.com	[null]
6	6	Diego	Hernández	2001-01-15	Ote 3 #56	Guadalajara	diego.h@email.com	[null]
7	7	María	Flores	2003-12-01	Col. Roma #77	CDMX	maria.flores@email.com	[null]
8	8	Sofía	Jiménez	2002-06-25	Av. Juárez 99	Querétaro	sofia.jimenez@email.com	[null]
9	9	Carlos	Cruz	2001-10-04	Zaragoza #13	CDMX	carlos.cruz@email.com	[null]
10	10	Elena	Ruiz	2000-08-17	La Paz 55	Oaxaca	elena.ruiz@email.com	[null]
11	11	Jorge	Ortiz	2001-04-03	Niza 21	CDMX	jorge.ortiz@email.com	[null]
12	12	Valeria	Mendoza	2002-02-28	Col. Del Valle 123	CDMX	valeria.mendoza@email.com	[null]
13	13	Hugo	Castillo	2000-12-30	Hidalgo 99	Toluca	hugo.castillo@email.com	[null]
14	14	Fernanda	Silva	2001-05-20	Camino Real 77	CDMX	fernanda.silva@email.com	[null]
15	15	Miguel	Reyes	2002-07-09	Independencia ...	León	miguel.reyes@email.com	[null]

4. Establecer relaciones foráneas:

- *Cursos.IDCampus* → *Campus.IDCampus*
- *Estudiantes.IDCarrera* → *Carreras.IDCarrera*

Estas modificaciones permiten vincular la información entre entidades de manera más eficiente y mantienen la integridad referencial de la base de datos.

```

-- Crear relación foránea entre Cursos e Campus
ALTER TABLE Cursos
ADD CONSTRAINT fk_cursos_campus
FOREIGN KEY (IDCampus) REFERENCES Campus(IDCampus);

```

```

-- Crear relación foránea entre Estudiantes y Carreras
ALTER TABLE Estudiantes
ADD CONSTRAINT fk_estudiante_carrera
FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera);

```

Actualizando Datos

Tras la modificación de la estructura de las tablas, se procedió a actualizar los registros en la tabla "Estudiantes", asignando a cada uno su correspondiente

carrera mediante el campo IDCarrera. Los registros fueron actualizados de la siguiente forma:

```
-- Actualización de Datos en Estudiantes (Asignar IDCarrera)
-- =====

UPDATE Estudiantes SET IDCarrera = 1 WHERE IDEstudiante = 1;
UPDATE Estudiantes SET IDCarrera = 2 WHERE IDEstudiante = 2;
UPDATE Estudiantes SET IDCarrera = 3 WHERE IDEstudiante = 3;
UPDATE Estudiantes SET IDCarrera = 4 WHERE IDEstudiante = 4;
UPDATE Estudiantes SET IDCarrera = 5 WHERE IDEstudiante = 5;
UPDATE Estudiantes SET IDCarrera = 6 WHERE IDEstudiante = 6;
UPDATE Estudiantes SET IDCarrera = 7 WHERE IDEstudiante = 7;
UPDATE Estudiantes SET IDCarrera = 8 WHERE IDEstudiante = 8;
UPDATE Estudiantes SET IDCarrera = 9 WHERE IDEstudiante = 9;
UPDATE Estudiantes SET IDCarrera = 10 WHERE IDEstudiante = 10;
UPDATE Estudiantes SET IDCarrera = 11 WHERE IDEstudiante = 11;
UPDATE Estudiantes SET IDCarrera = 12 WHERE IDEstudiante = 12;
UPDATE Estudiantes SET IDCarrera = 13 WHERE IDEstudiante = 13;
UPDATE Estudiantes SET IDCarrera = 14 WHERE IDEstudiante = 14;
UPDATE Estudiantes SET IDCarrera = 15 WHERE IDEstudiante = 15;
```

Eliminando Datos

Se identificaron datos y estructuras que ya no eran necesarios:

- **Se eliminó la columna Ciudad de la tabla Estudiantes**

```

39  -- Eliminar columna Ciudad de Estudiantes
40  ALTER TABLE Estudiantes DROP COLUMN Ciudad;
41

```

	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fechanacimiento date	direccion text	email character varying (100)
1	1	Ana	García	2002-05-14	Calle A #123	ana.garcia@email.com
2	2	Luis	Martínez	2001-03-22	Av. Reforma 45	luis.martinez@email.com
3	3	Carla	López	2000-07-19	Col. Centro 202	carla.lopez@email.com
4	4	Pablo	Ramírez	2003-11-30	Sur 7 #33	pablo.ramirez@email.com
5	5	Lucía	Torres	2002-09-10	Norte 18 #8	lucia.torres@email.com
6	6	Diego	Hernández	2001-01-15	Ote 3 #56	diego.h@email.com
7	7	María	Flores	2003-12-01	Col. Roma #77	maria.flores@email.com
8	8	Sofía	Jiménez	2002-06-25	Av. Juárez 99	sofia.jimenez@email.com
9	9	Carlos	Cruz	2001-10-04	Zaragoza #13	carlos.cruz@email.com
10	10	Elena	Ruiz	2000-08-17	La Paz 55	elena.ruiz@email.com
11	11	Jorge	Ortiz	2001-04-03	Niza 21	jorge.ortiz@email.com
12	12	Valeria	Mendoza	2002-02-28	Col. Del Valle 123	valeria.mendoza@email.com
13	13	Hugo	Castillo	2000-12-30	Hidalgo 99	hugo.castillo@email.com
14	14	Fernanda	Silva	2001-05-20	Camino Real 77	fernanda.silva@email.com
15	15	Miguel	Reyes	2002-07-09	Independencia ...	miguel.reyes@email.com

- **Se eliminó la tabla Aulas**

```

55  -- Eliminar tabla Aulas si existe
56  DROP TABLE Aulas;

```

ERROR: restricción «horarios_idaula_fkey» en tabla horarios depende de tabla aulasno se puede eliminar tabla aulas porque otros objetos dependen de él

ERROR: no se puede eliminar tabla aulas porque otros objetos dependen de él

SQL state: 2BP01

Detail: restricción «horarios_idaula_fkey» en tabla horarios depende de tabla aulas

Hint: Use DROP ... CASCADE para eliminar además los objetos dependientes.

Normalmente se borraría la tabla Aulas pero como la tabla horarios depende de aulas no se borra pero sí se borraría con el DROP TABLE Aulas CASCADE; porque elimina a la tabla y todo lo que dependa de ella

Insertando Datos

En esta sección se muestran las capturas de pantalla correspondientes a las sentencias INSERT INTO ejecutadas para poblar las tablas con información de ejemplo representativa del entorno universitario.

```
-- Insertar datos en Departamentos
INSERT INTO Departamentos (IDDepartamento, NombreDepartamento, Edificio) VALUES
(1, 'Ingeniería', 'A'),
(2, 'Ciencias Sociales', 'B'),
(3, 'Negocios', 'C'),
(4, 'Matemáticas', 'D'),
(5, 'Biología', 'E'),
(6, 'Física', 'F'),
(7, 'Química', 'G'),
(8, 'Historia', 'H'),
(9, 'Filosofía', 'I'),
(10, 'Psicología', 'J'),
(11, 'Educación', 'K'),
(12, 'Literatura', 'L'),
(13, 'Arte', 'M'),
(14, 'Música', 'N'),
(15, 'Derecho', 'O');
```

```
-- Insertar datos en Estudiantes
INSERT INTO Estudiantes (IDEstudiante, Nombre, Apellido, FechaNacimiento, Direccion, Ciudad, Email) VALUES
(1, 'Ana', 'García', '2002-05-14', 'Calle A #123', 'CDMX', 'ana.garcia@email.com'),
(2, 'Luis', 'Martínez', '2001-03-22', 'Av. Reforma 45', 'CDMX', 'luis.martinez@email.com'),
(3, 'Carla', 'López', '2000-07-19', 'Col. Centro 202', 'CDMX', 'carla.lopez@email.com'),
(4, 'Pablo', 'Ramírez', '2003-11-30', 'Sur 7 #33', 'Puebla', 'pablo.ramirez@email.com'),
(5, 'Lucía', 'Torres', '2002-09-10', 'Norte 18 #8', 'Monterrey', 'lucia.torres@email.com'),
(6, 'Diego', 'Hernández', '2001-01-15', 'Ote 3 #56', 'Guadalajara', 'diego.h@email.com'),
(7, 'María', 'Flores', '2003-12-01', 'Col. Roma #77', 'CDMX', 'maria.flores@email.com'),
(8, 'Sofía', 'Jiménez', '2002-06-25', 'Av. Juárez 99', 'Querétaro', 'sofia.jimenez@email.com'),
(9, 'Carlos', 'Cruz', '2001-10-04', 'Zaragoza #13', 'CDMX', 'carlos.cruz@email.com'),
(10, 'Elena', 'Ruiz', '2000-08-17', 'La Paz 55', 'Oaxaca', 'elena.ruiz@email.com'),
(11, 'Jorge', 'Ortiz', '2001-04-03', 'Niza 21', 'CDMX', 'jorge.ortiz@email.com'),
(12, 'Valeria', 'Mendoza', '2002-02-28', 'Col. Del Valle 123', 'CDMX', 'valeria.mendoza@email.com'),
(13, 'Hugo', 'Castillo', '2000-12-30', 'Hidalgo 99', 'Toluca', 'hugo.castillo@email.com'),
(14, 'Fernanda', 'Silva', '2001-05-20', 'Camino Real 77', 'CDMX', 'fernanda.silva@email.com'),
(15, 'Miguel', 'Reyes', '2002-07-09', 'Independencia 16', 'León', 'miguel.reyes@email.com');
```

```
-- Insertar datos en Profesores
INSERT INTO Profesores (IDProfesor, Nombre, Apellido, Titulo, IDDepartamento) VALUES
(1, 'Laura', 'Hernández', 'PhD', 1),
(2, 'Ricardo', 'Gómez', 'MSc', 2),
(3, 'Patricia', 'Morales', 'PhD', 3),
(4, 'Andrés', 'Díaz', 'MSc', 4),
(5, 'Rosa', 'Santos', 'PhD', 5),
(6, 'Pedro', 'Cano', 'MSc', 6),
(7, 'Gabriela', 'Medina', 'PhD', 7),
(8, 'Ernesto', 'Beltrán', 'PhD', 8),
(9, 'Claudia', 'Pérez', 'MSc', 9),
(10, 'Héctor', 'Aguilar', 'PhD', 10),
(11, 'Julietta', 'Nava', 'MSc', 11),
(12, 'César', 'Zamora', 'PhD', 12),
(13, 'Marcela', 'Vargas', 'MSc', 13),
(14, 'Luis', 'Nieto', 'PhD', 14),
(15, 'Daniela', 'Mora', 'MSc', 15);
```

```

INSERT INTO Carreras (IDCarrera, NombreCarrera, TituloOtorgado) VALUES
(1, 'Ingeniería en Sistemas Computacionales', 'Ingeniero en Sistemas Computacionales'),
(2, 'Ingeniería Industrial', 'Ingeniero Industrial'),
(3, 'Ingeniería Civil', 'Ingeniero Civil'),
(4, 'Ingeniería Mecánica', 'Ingeniero Mecánico'),
(5, 'Ingeniería Electrónica', 'Ingeniero Electrónico'),
(6, 'Arquitectura', 'Arquitecto'),
(7, 'Contaduría Pública', 'Contador Público'),
(8, 'Administración de Empresas', 'Licenciado en Administración de Empresas'),
(9, 'Derecho', 'Licenciado en Derecho'),
(10, 'Medicina', 'Médico Cirujano'),
(11, 'Enfermería', 'Licenciado en Enfermería'),
(12, 'Psicología', 'Licenciado en Psicología'),
(13, 'Diseño Gráfico', 'Licenciado en Diseño Gráfico'),
(14, 'Ciencias de la Comunicación', 'Licenciado en Ciencias de la Comunicación'),
(15, 'Ingeniería Ambiental', 'Ingeniero Ambiental');

-- Insertar datos en Campus
INSERT INTO Campus (IDCampus, NombreCampus, DireccionCampus) VALUES
(1, 'Campus Central', 'Av. Principal 123'),
(2, 'Campus Norte', 'Calle Norte 456'),
(3, 'Campus Sur', 'Boulevard Sur 789'),
(4, 'Campus Este', 'Avenida Este 101'),
(5, 'Campus Oeste', 'Calle Oeste 202'),
(6, 'Campus Tecnológico', 'Camino Tech 303'),
(7, 'Campus Ciencias', 'Avenida Ciencia 404'),
(8, 'Campus Humanidades', 'Calle Cultura 505'),
(9, 'Campus Medicina', 'Boulevard Salud 606'),
(10, 'Campus Artes', 'Plaza Arte 707'),
(11, 'Campus Economía', 'Calle Finanzas 808'),
(12, 'Campus Derecho', 'Avenida Justicia 909'),
(13, 'Campus Ingeniería', 'Calle Ingeniería 110'),
(14, 'Campus Internacional', 'Camino Global 120'),
(15, 'Campus Virtual', 'Internet 130');

```



```
-- Insertar datos en Cursos con referencia a Campus
INSERT INTO Cursos (IDCurso, NombreCurso, Descripción, Creditos, Semestre, IDDepartamento, IDCampus) VALUES
(1, 'Matemáticas I', 'Curso básico de matemáticas', 5, '1', 1, 1),
(2, 'Física I', 'Curso básico de física', 5, '1', 2, 2),
(3, 'Química General', 'Introducción a la química', 5, '1', 3, 3),
(4, 'Programación I', 'Introducción a la programación', 6, '1', 4, 4),
(5, 'Electrónica Básica', 'Fundamentos de electrónica', 5, '2', 5, 5),
(6, 'Historia Universal', 'Estudio de la historia mundial', 4, '2', 6, 6),
(7, 'Contabilidad I', 'Principios contables', 5, '1', 7, 7),
(8, 'Marketing', 'Fundamentos de marketing', 5, '2', 8, 8),
(9, 'Derecho Constitucional', 'Estudio de la constitución', 4, '3', 9, 9),
(10, 'Anatomía Humana', 'Estudio del cuerpo humano', 6, '1', 10, 10),
(11, 'Psicología General', 'Introducción a la psicología', 5, '1', 11, 11),
(12, 'Diseño Gráfico', 'Fundamentos del diseño', 5, '2', 12, 12),
(13, 'Economía I', 'Principios económicos', 5, '1', 13, 13),
(14, 'Ingeniería de Software', 'Desarrollo de software', 6, '3', 14, 14),
(15, 'Educación Virtual', 'Herramientas de educación en línea', 4, '2', 15, 15);
```

```
-- Insertar datos a ProgramasEstudio
INSERT INTO ProgramasEstudio (IDPrograma, NombrePrograma, DescripcionPrograma) VALUES
(1, 'Ingeniería en Sistemas', 'Formación en sistemas y desarrollo de software'),
(2, 'Psicología', 'Estudio del comportamiento humano'),
(3, 'Matemáticas', 'Fundamentos matemáticos y aplicados'),
(4, 'Biología', 'Estudio de organismos vivos'),
(5, 'Física', 'Leyes de la naturaleza y sus aplicaciones'),
(6, 'Contabilidad', 'Gestión financiera y contable'),
(7, 'Historia', 'Análisis histórico y cultural'),
(8, 'Ética', 'Estudio de principios morales'),
(9, 'Sociología', 'Estructura social y relaciones'),
(10, 'Microbiología', 'Estudio de microorganismos'),
(11, 'Lógica', 'Razonamiento formal y argumentación'),
```

Realizando Consultas

Se ejecutaron consultas para validar la correcta integración de los datos:

- **Mostrar estudiantes con su respectiva carrera.**

```

61 -- Listar estudiantes con su carrera (Usando columna IDCarrera en Estudiantes)
62 SELECT e.Nombre, e.Apellido, c.NombreCarrera
63 FROM Estudiantes e
64 JOIN Carreras c ON e.IDCarrera = c.IDCarrera;
65

```

Data Output Messages Notifications

	nombre character varying (50)	apellido character varying (50)	nombrecarrera character varying (100)
1	Ana	García	Ingeniería en Sistemas Computacionales
2	Luis	Martínez	Ingeniería Industrial
3	Carla	López	Ingeniería Civil
4	Pablo	Ramírez	Ingeniería Mecánica
5	Lucía	Torres	Ingeniería Electrónica
6	Diego	Hernández	Arquitectura
7	María	Flores	Contaduría Pública
8	Sofía	Jiménez	Administración de Empresas
9	Carlos	Cruz	Derecho
10	Elena	Ruiz	Medicina
11	Jorge	Ortiz	Enfermería
12	Valeria	Mendoza	Psicología
13	Hugo	Castillo	Diseño Gráfico
14	Fernanda	Silva	Ciencias de la Comunicación
15	Miguel	Reyes	Ingeniería Ambiental

- **Mostrar cursos con su campus asignado.**

```

66  -- Listar cursos con su campus
67  SELECT cu.NombreCurso, ca.NombreCampus
68  FROM Cursos cu
69  JOIN Campus ca ON cu.IDCampus = ca.IDCampus;

```

Data Output Messages Notifications

	nombrecurso character varying (100) 🔒	nombrecampus character varying (100) 🔒
1	Matemáticas I	Campus Central
2	Física I	Campus Norte
3	Química General	Campus Sur
4	Programación I	Campus Este
5	Electrónica Básica	Campus Oeste
6	Historia Universal	Campus Tecnológico
7	Contabilidad I	Campus Ciencias
8	Marketing	Campus Humanidades
9	Derecho Constitucional	Campus Medicina
10	Anatomía Humana	Campus Artes
11	Psicología General	Campus Economía
12	Diseño Gráfico	Campus Derecho
13	Economía I	Campus Ingeniería
14	Ingeniería de Software	Campus Internacional
15	Educación Virtual	Campus Virtual

- **Mostrar profesores con el departamento al que pertenecen.**

```

71 -- Listar profesores por departamento
72 SELECT p.Nombre, p.Apellido, d.NombreDepartamento
73 FROM Profesores p
74 JOIN Departamentos d ON p.IDDepartamento = d.IDDepartamento;

```

	nombre character varying (50)	apellido character varying (50)	nombredepartamento character varying (100)
1	Laura	Hernández	Ingeniería
2	Ricardo	Gómez	Ciencias Sociales
3	Patricia	Morales	Negocios
4	Andrés	Díaz	Matemáticas
5	Rosa	Santos	Biología
6	Pedro	Cano	Física
7	Gabriela	Medina	Química
8	Ernesto	Beltrán	Historia
9	Claudia	Pérez	Filosofía
10	Héctor	Aguilar	Psicología
11	Julieta	Nava	Educación
12	César	Zamora	Literatura
13	Marcela	Vargas	Arte
14	Luis	Nieto	Música
15	Daniela	Mora	Derecho

- **Mostrar programas de estudio con los cursos que incluyen.**

Consultas específicas

Selección Básica

Consulta para mostrar todos los nombres y apellidos de los estudiantes.

61	-- 1. Selección Básica
62	SELECT Nombre, Apellido
63	FROM Estudiantes;

Data Output	Messages	Notifications
-------------	----------	---------------

									SQL
--	--	--	--	--	--	--	--	--	-----

	nombre character varying (50) 🔒	apellido character varying (50) 🔒
1	Ana	García
2	Luis	Martínez
3	Carla	López
4	Pablo	Ramírez
5	Lucía	Torres
6	Diego	Hernández
7	María	Flores
8	Sofía	Jiménez
9	Carlos	Cruz
10	Elena	Ruiz
11	Jorge	Ortiz
12	Valeria	Mendoza
13	Hugo	Castillo
14	Fernanda	Silva
15	Miguel	Reyes

Cláusula WHERE

Consulta para encontrar todos los cursos que tienen 5 créditos.

```
65 -- 2. Cláusula WHERE
66 SELECT NombreCurso
67 FROM Cursos
68 WHERE Créditos =5;
```

Data Output Messages Notificatio

	nombrecurso character varying (100) 🔒
1	Matemáticas I
2	Física I
3	Química General
4	Electrónica Básica
5	Contabilidad I
6	Marketing
7	Psicología General
8	Diseño Gráfico
9	Economía I
10	Psicología General

INNER JOIN: Estudiantes y Cursos Inscritos

Consulta para obtener el nombre del estudiante y el nombre del curso en el que está inscrito.

```

70 -- 3. INNER JOIN: Estudiantes y Cursos inscritos
71 SELECT e.Nombre AS NombreEstudiante, c.NombreCurso
72 FROM Inscripciones i
73 JOIN Estudiantes e ON i.IDEstudiante = e.IDEstudiante
74 JOIN Cursos c ON i.IDCurso = c.IDCurso;

```

Data Output Messages Notifications

	nombreestudiante character varying (50) 🔒	nombrecurso character varying (100) 🔒
1	Ana	Matemáticas I
2	Ana	Física I
3	Luis	Matemáticas I
4	Carla	Química General
5	Luis	Química General
6	Pablo	Programación I
7	Lucía	Electrónica Básica
8	Diego	Historia Universal
9	María	Contabilidad I
10	Sofía	Marketing
11	Carlos	Derecho Constitucional
12	Elena	Anatomía Humana
13	Jorge	Psicología General
14	Valeria	Diseño Gráfico
15	Hugo	Economía I

LEFT JOIN: Estudiantes con sus cursos o NULL

Consulta que muestra todos los estudiantes y, si están inscritos en algún curso, el nombre del curso. Si no, muestra NULL.

```

76  -- 4. LEFT JOIN: Estudiantes con sus cursos o NULL
77  SELECT e.Nombre AS NombreEstudiante, c.NombreCurso
78  FROM Estudiantes e
79  LEFT JOIN Inscripciones i ON e.IDEstudiante = i.IDEstudiante
80  LEFT JOIN Cursos c ON i.IDCurso = c.IDCurso;

```

Data Output Messages Notifications

	nombreestudiante character varying (50) 🔒	nombrecurso character varying (100) 🔒
1	Ana	Matemáticas I
2	Ana	Física I
3	Luis	Matemáticas I
4	Carla	Química General
5	Luis	Química General
6	Pablo	Programación I
7	Lucía	Electrónica Básica
8	Diego	Historia Universal
9	María	Contabilidad I
10	Sofía	Marketing
11	Carlos	Derecho Constitucional
12	Elena	Anatomía Humana
13	Jorge	Psicología General
14	Valeria	Diseño Gráfico
15	Hugo	Economía I

RIGHT JOIN: Cursos con estudiantes o NULL

Consulta que lista todos los cursos y, si tienen estudiantes inscritos, muestra su nombre. Incluye cursos sin estudiantes.


```

82  -- 5. RIGHT JOIN: Cursos con estudiantes o NULL
83  SELECT c.NombreCurso, e.Nombre AS NombreEstudiante
84  FROM Cursos c
85  LEFT JOIN Inscripciones i ON c.IDCurso = i.IDCurso
86  LEFT JOIN Estudiantes e ON i.IDEstudiante = e.IDEstudiante;

```

Data Output Messages Notifications

	nombrecurso character varying (100)	nombreestudiante character varying (50)
1	Matemáticas I	Ana
2	Física I	Ana
3	Matemáticas I	Luis
4	Química General	Carla
5	Química General	Luis
6	Programación I	Pablo
7	Electrónica Básica	Lucía
8	Historia Universal	Diego
9	Contabilidad I	María
10	Marketing	Sofía
11	Derecho Constitucional	Carlos
12	Anatomía Humana	Elena
13	Psicología General	Jorge
14	Diseño Gráfico	Valeria
15	Economía I	Hugo

GROUP BY y COUNT: Estudiantes por curso

Consulta para calcular cuántos estudiantes están inscritos en cada curso.

88	-- 6. GROUP BY y COUNT: Estudiantes por curso
89	SELECT c.NombreCurso, COUNT(i.IDEstudiante) AS TotalEstudiantes
90	FROM Cursos c
91	LEFT JOIN Inscripciones i ON c.IDCurso = i.IDCurso
92	GROUP BY c.NombreCurso;

Data Output			Messages	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div>				
	nombrecurso character varying (100) 🔒	totalestudiantes bigint 🔒		
1	Programación I	1		
2	Economía I	1		
3	Química General	2		
4	Literatura Clásica	0		
5	Estructuras de Datos	0		
6	Contabilidad Básica	0		
7	Electrónica Básica	1		
8	Anatomía Humana	1		
9	Lógica y Argumentación	0		
10	Física I	1		
11	Marketing Estratégico	0		
12	Matemáticas I	2		
13	Derecho Constitucional	1		
14	Psicología General	1		
15	Diseño Gráfico	1		

BETWEEN: Estudiantes nacidos entre 2000 y 2002
 Consulta para encontrar estudiantes con fecha de nacimiento en ese rango.

94

-- 7. BETWEEN: Estudiantes nacidos entre 1995 y 1998

95

SELECT Nombre, Apellido, FechaNacimiento

96

FROM Estudiantes

97

WHERE FechaNacimiento BETWEEN '2000-07-19' AND '2002-05-14';

Data Output

Messages

Notifications

<

ORDER BY: Cursos ordenados alfabéticamente
Consulta para mostrar todos los cursos ordenados por nombre.

-- 8. ORDER BY: Cursos ordenados alfabéticamente

SELECT *

FROM Cursos

ORDER BY NombreCurso ASC;

Data Output

Messages

Notifications

SQL

Showing rows: 1 to 25

	Idcurso [PK] integer	nombrecurso character varying (100)	descripcion text	creditos integer	semestre character varying (10)	iddepartamento integer	idcampus integer	idprofesor integer
1	17	Álgebra Lineal	[null]	3	[null]	[null]	2	17
2	10	Anatomía Humana	Estudio del cuerpo humano	6	1	10	10	[null]
3	19	Contabilidad Básica	[null]	3	[null]	[null]	4	19
4	7	Contabilidad I	Principios contables	5	1	7	7	[null]
5	9	Derecho Constitucional	Estudio de la constitución	4	3	9	9	[null]
6	20	Diseño Digital	[null]	4	[null]	[null]	5	20
7	12	Diseño Gráfico	Fundamentos del diseño	5	2	12	12	[null]
8	13	Economía I	Principios económicos	5	1	13	13	[null]
9	15	Educación Virtual	Herramientas de educación en línea	4	2	15	15	[null]
10	5	Electrónica Básica	Fundamentos de electrónica	5	2	5	5	[null]
11	24	Estructuras de Datos	[null]	3	[null]	[null]	4	24
12	21	Física I	[null]	2	[null]	[null]	1	21
13	2	Física I	Curso básico de física	5	1	2	2	[null]
14	6	Historia Universal	Estudio de la historia mundial	4	2	6	6	[null]
15	14	Ingeniería de Software	Desarrollo de software	6	3	14	14	[null]

CTE: Top 3 estudiantes con más inscripciones

Consulta con tabla de expresión común (CTE) para listar los 3 estudiantes con más inscripciones.

```
104 -- 9. CTE: Top 3 estudiantes con más inscripciones
105 WITH InscripcionesPorEstudiante AS (
106     SELECT e.IDEstudiante, e.Nombre, COUNT(i.IDCurso) AS TotalInscripciones
107     FROM Estudiantes e
108     JOIN Inscripciones i ON e.IDEstudiante = i.IDEstudiante
109     GROUP BY e.IDEstudiante, e.Nombre
110 )
111 SELECT *
112 FROM InscripcionesPorEstudiante
113 ORDER BY TotalInscripciones DESC
114 LIMIT 3;
```

Data Output Messages Notifications

	Idestudiante [PK] integer	nombre character varying (50)	totalinscripciones bigint
1	1	Ana	2
2	2	Luis	2
3	3	Carla	1

Consulta Compleja 1: Curso más inscrito por departamento

Consulta para cada departamento mostrar el curso con más estudiantes inscritos.

```

116 -- 10. Consulta Compleja 1: Curso más inscrito por departamento
117 WITH Conteo AS (
118     SELECT
119         d.IDDepartamento,
120         d.NombreDepartamento,
121         c.IDCurso,
122         c.NombreCurso,
123         COUNT(i.IDEstudiante)::INT AS TotalInscritos
124     FROM Departamentos d
125     JOIN Profesores p ON d.IDDepartamento = p.IDDepartamento
126     JOIN Cursos c ON p.IDProfesor = c.IDProfesor
127     LEFT JOIN Inscripciones i ON c.IDCurso = i.IDCurso
128     GROUP BY d.IDDepartamento, d.NombreDepartamento, c.IDCurso, c.NombreCurso
129 ),
130 Maximo AS (
131     SELECT IDDepartamento, MAX(TotalInscritos) AS MaxInscritos
132     FROM Conteo
133     GROUP BY IDDepartamento
134 )
135 SELECT c.NombreDepartamento, c.NombreCurso, c.TotalInscritos
136 FROM Conteo c
137 JOIN Maximo m ON c.IDDepartamento = m.IDDepartamento AND c.TotalInscritos = m.MaxInscritos;

```

Data Output Messages Notifications

	nombreddepartamento character varying (100)	nombrecurso character varying (100)	totalinscritos integer
1	Ciencias Sociales	Sociología	1
2	Ingeniería	Matemáticas	2

Consulta Compleja 2: Profesores con más de 2 cursos
Consulta para encontrar profesores que imparten más de dos cursos.

```

134 -- 11. Consulta Compleja 2: Profesores con más de 2 cursos
135 SELECT
136     p.Nombre,
137     p.Apellido,
138     COUNT(c.IDCurso) AS TotalCursos
139 FROM Profesores p
140 JOIN Cursos c ON p.IDProfesor = c.IDProfesor
141 GROUP BY p.IDProfesor, p.Nombre, p.Apellido
142 HAVING COUNT(c.IDCurso) > 2;

```

Data Output Messages Notifications

	nombre character varying (50)	apellido character varying (50)	totalcursos bigint
--	----------------------------------	------------------------------------	-----------------------

Consulta Compleja 3: Curso con mejor promedio por programa
Consulta que lista para cada programa el curso con el promedio de calificación más alto.

```

149 -- 12. Consulta Compleja 3: Curso con mejor promedio por programa
150 WITH Promedios AS (
151     SELECT pe.IDPrograma, pe.NombrePrograma, c.IDCurso, c.NombreCurso, ROUND(AVG(i.Calificacion), 2) AS Promedio
152     FROM ProgramasEstudio pe
153     JOIN ProgramasCursos pc ON pe.IDPrograma = pc.IDPrograma
154     JOIN Cursos c ON pc.IDCurso = c.IDCurso
155     JOIN Inscripciones i ON c.IDCurso = i.IDCurso
156     GROUP BY pe.IDPrograma, pe.NombrePrograma, c.IDCurso, c.NombreCurso
157 ),
158 Maximos AS (
159     SELECT IDPrograma, MAX(Promedio) AS MaxProm
160     FROM Promedios
161     GROUP BY IDPrograma
162 )
163 SELECT p.NombrePrograma, p.NombreCurso, p.Promedio
164 FROM Promedios p
165 JOIN Maximos m ON p.IDPrograma = m.IDPrograma AND p.Promedio = m.MaxProm;
166

```

	nombrequemateria	nombrecurso	promedio
	character varying (100)	character varying (100)	numeric
1	Ingeniería en Sistemas	Física	90.00
2	Ciencias Sociales	Sociología	92.00