



## INSTITUTO TECNOLOGICO DE MEXICALI

Alumno:

Marin Salazar Juan Sebastian 22490423

Materia:

Fundamentos de base de datos.

Profesor:

Jose Ramón Bogarin Valenzuela.

Tema:

Examen Final de Unidad 4

Fecha y Hora de Entrega:

23 de Mayo del 2025 a las 5:46 pm

Carrera:

ING. en Sistemas.

Profesor:

Jose Ramón Bogarin Valenzuela.

Mexicali, Baja California a 23 de Mayo del 2025.

## Problema Técnico: Gestionando la Información de una Universidad

Imagina que estás trabajando en el equipo de desarrollo de software para una universidad. La universidad necesita un sistema para gestionar la información de sus estudiantes, los cursos que ofrecen, las inscripciones de los estudiantes a los cursos, los profesores que imparten los cursos y los departamentos a los que pertenecen los profesores. Tu tarea es diseñar y trabajar con la base de datos que almacenará toda esta información.

### Objetivo General:

Diseñar una base de datos relacional y realizar diversas operaciones para gestionar la información de la universidad. Esto incluye la creación y modificación de la estructura de las tablas, la manipulación de los datos y la realización de consultas complejas para obtener información específica.

### Esquema de la Base de Datos:

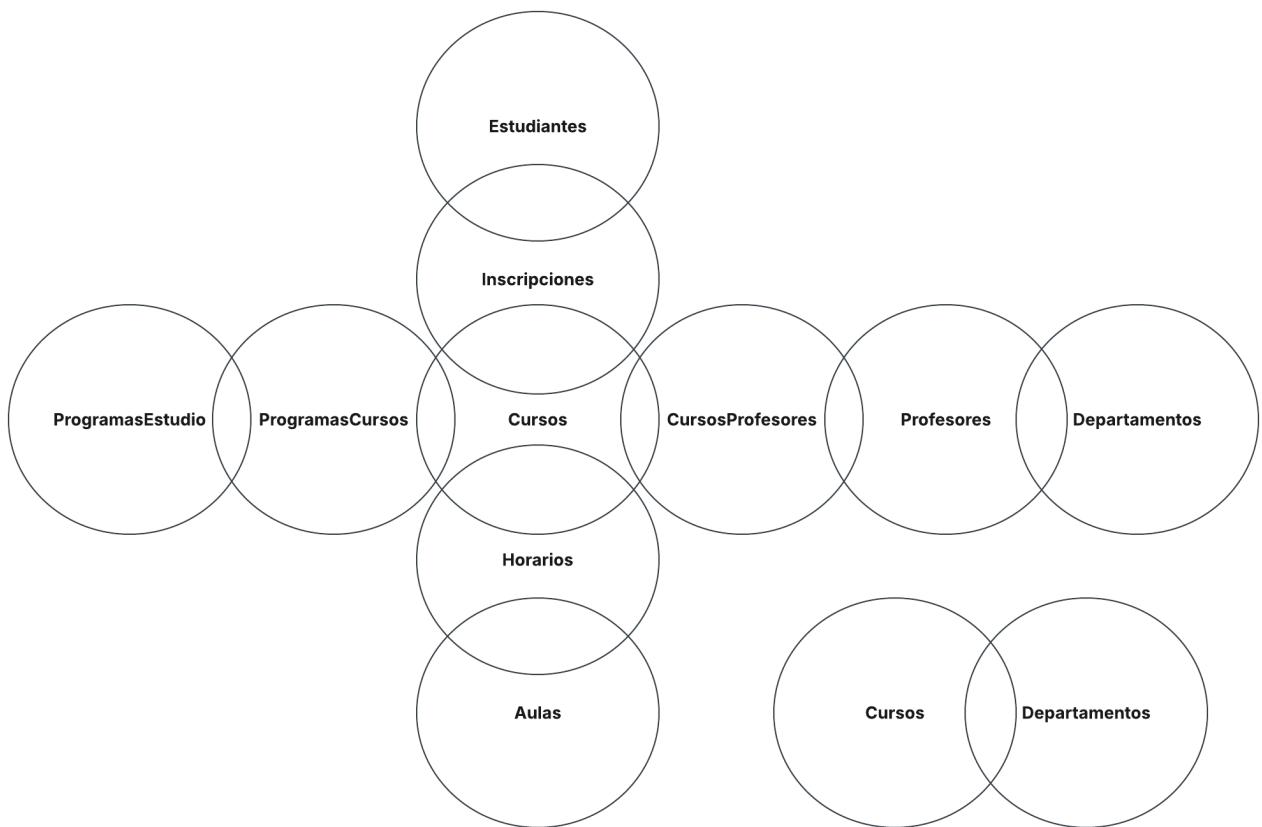
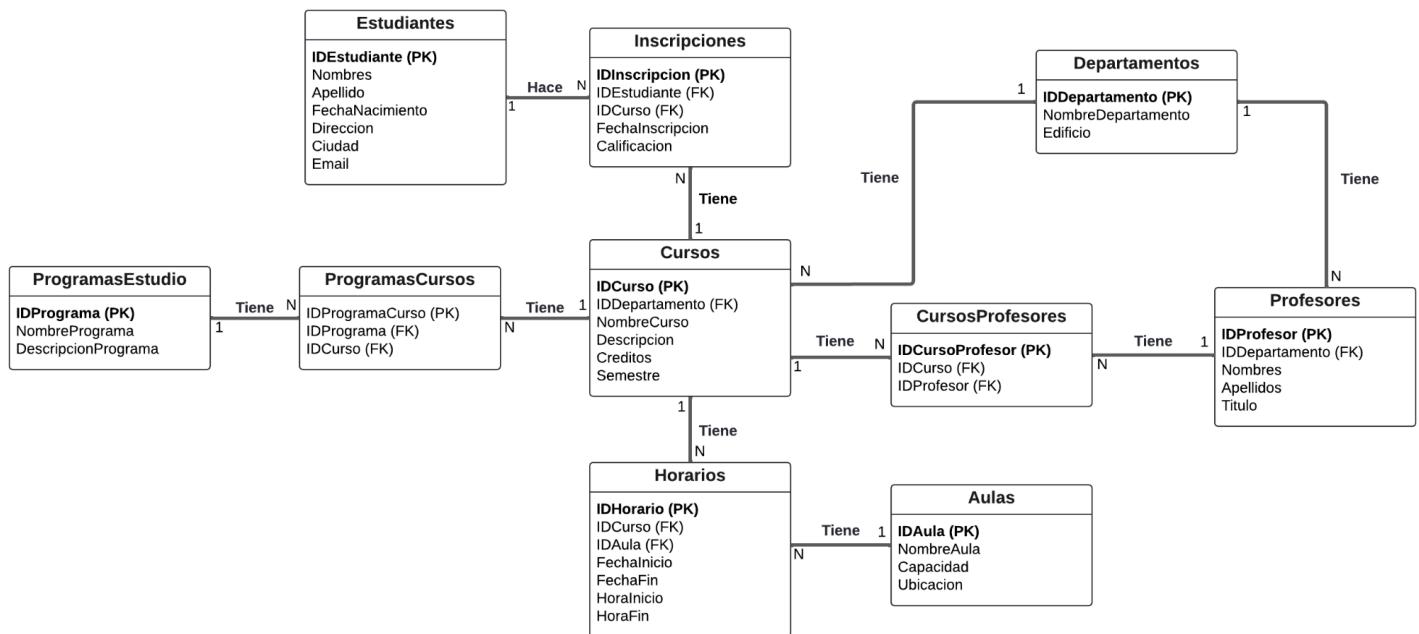
Aquí se presenta un esquema de la base de datos con 10 tablas y sus relaciones:

### Modelo de Datos Lógico (MDL):

- **Tabla: Estudiantes**
  - IDEstudiante (Clave Principal)
  - Nombre
  - Apellido
  - FechaNacimiento
  - Direccion
  - Ciudad
  - Email
- **Tabla: Cursos**
  - IDCurso (Clave Principal)
  - NombreCurso
  - Descripcion
  - Creditos
  - Semestre
  - IDDepartamento (Clave Foránea, referencia a la tabla Departamentos)
- **Tabla: Inscripciones**
  - IDInscripcion (Clave Principal)
  - IDEstudiante (Clave Foránea, referencia a la tabla Estudiantes)
  - IDCurso (Clave Foránea, referencia a la tabla Cursos)
  - Fechalincripcion
  - Calificacion

- **Tabla: Profesores**
  - IDProfesor (Clave Principal)
  - Nombre
  - Apellido
  - Titulo
  - >IDDepartamento (Clave Foránea, referencia a la tabla Departamentos)
- **Tabla: Departamentos**
  - IDDepartamento (Clave Principal)
  - NombreDepartamento
  - Edificio
- **Tabla: Aulas**
  - IDAula (Clave Principal)
  - 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
- **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)
- **Tabla: ProgramasEstudio**
  - IDPrograma (Clave Principal)
  - NombrePrograma
  - DescripcionPrograma
- **Tabla Intermedia: ProgramasCursos** (Relación Muchos a Muchos entre ProgramasEstudio y Cursos)
  - IDProgramaCurso (Clave Principal)
  - IDPrograma (Clave Foránea, referencia a la tabla ProgramasEstudio)
  - IDCurso (Clave Foránea, referencia a la tabla Cursos)

Antes de crear las tablas tendré presente un diagrama entidad relación y diagramas de venn para no complicarme en los querys solicitados.



### Pasos a Seguir:

Los estudiantes deben seguir estos pasos para resolver el problema:

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

#### Entidades Principales:

**Estudiantes:** Representa a los alumnos de la universidad.

Sus datos relevantes serían:

**IDEstudiante (PK)** - Que sería su id único

**Nombres** - Aquí pueden ser uno o varios nombres que tenga el alumno

**ApellidoPaterno** - Aquí es apellido paterno

**ApellidoMaterno** - Aquí el apellido materno (si no tiene se puede dejar vacío)

**FechaNacimiento** - La edad

**Direccion** - Donde vive el estudiante

**Ciudad** - De qué ciudad es

**Email** - El correo del alumno

**Inscripciones:** Relaciona Estudiantes con Cursos

Sus datos relevantes serían:

**IDInscripcion (PK)** - Identificador para cada inscripción

**IDEstudiante (FK)** - Estudiante que se inscribe

**IDCurso (FK)** - Curso al que se inscribe

**FechaInscripcion** - Cuando se realizó la inscripción

**Calificacion** - Calificación final del curso

**Cursos:** Representan las asignaturas o materias ofrecidas por la universidad.

Sus datos relevantes serían:

**IDCurso (PK)** - Identificador del curso

**IDDepartamento (FK)** - Identificador del departamento al que pertenece el curso

**NombreCurso** - Nombre del curso jaja

**Descripcion** - Descripción simple de la materia, para que sepas que show

**Creditos** - Créditos que dará el curso cuando se apruebe

**Semestre** - De que semestre es la materia

**Departamentos:** Divisiones académicas que agrupan cursos y profesores.

Sus datos relevantes serían:

**IDDepartamento (PK)** - Identificador único de departamento

**NombreDepartamento** - Nombre del área o departamento

**Edificio** - Donde se encuentra físicamente

**Profesores:** Representa a los profesores que imparten los cursos.

Sus datos relevantes serían:

**IDProfesor (PK)** - Identificador único de los profes

**IDDepartamento (FK)** - Id del departamento al que pertenece el profe

**Nombres** - Nombre o nombres del profe

**ApellidoPaterno** - Apellido paterno del profe

**ApellidoMaterno** - Apellido materno del profe (Si no tiene se puede dejar vacío)

**Titulo** - De que se graduó el profe

**CursosProfesores:** Siendo una tabla intermedia representa la relación muchos a muchos entre cursos y profesores.

Sus datos relevantes serían:

**IDCursoProfesor (PK)** - Identificador único de la tabla

**IDCurso (FK)** - Identificador del curso que será impartido

**IDProfesor (FK)** - Identificador del profe que impartirá el curso

**Aulas:** Representa los salones o laboratorios donde se impartirán los cursos.

Sus datos relevantes serían:

**IDAula (PK)** - Identificador único del aula

**NombreAula** - Nombre del aula

**Capacidad** - Número máximo de cuántos alumnos caben en ese aula

**Ubicacion** - Donde está físicamente

**Horarios:** Siendo una tabla intermedia relaciona cursos con aulas.

Sus datos relevantes serían:

**IDHorario (PK)** - Identificador único del horario

**IDCurso (FK)** - Id del curso que se le asignará aula

**IDAula (FK)** - Id del aula asignada al curso

**Fechalnicio** - Fecha en la que inicia el curso

**FechaFin** - Fecha en la que termina el curso

**Horalnicio** - Hora en la que inicia el curso

**HoraFin** - Hora en la que termina el curso

**ProgramasEstudio:** Plan de estudios en el que está el alumno.

Sus datos relevantes serían:

**IDPrograma (PK)** - Identificador único de programa de estudio

**NombrePrograma** - Nombre del programa de estudio

**DescripcionPrograma** - Descripción del programa de estudio

**ProgramasCursos:** Siendo una tabla intermedia de relación muchos a muchos entre programas y cursos.

Sus datos relevantes serían:

**IDProgramaCurso (PK)** - Identificador único de la tabla

**IDPrograma (FK)** - Id del programa al cual está asignado ese curso

**IDCurso (FK)** - Id del curso que está dentro de ese programa

- Diseñando las Tablas (MDL):** Organizar la información en tablas, definiendo campos, tipos de datos y claves principales, y establecer las relaciones entre las tablas.

Como voy a usar el Tec para basarme de ahí algunas decisiones tendré que explicar por que las tome él la estructuración de las tablas.

## Estudiantes

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDEstudiante	Serial	PK	No	Único	--
Nombre	Varchar(50)	--	No	--	--
Apellido	Varchar(100)	--	No	--	--
FechaNacimiento	Date	--	No	--	--
Direccion	Varchar(150)	--	No	--	--
Ciudad	Varchar(100)	--	No	--	--
Email	Varchar(150)	--	No	Único	--

IDEstudiante es serial porque me da flojera meter los ids de forma manual

Nombre es Varchar(50) porque estoy asumiendo que todos los alumnos pueden tener uno o dos nombres, raro sería que tengan tres en adelante

Apellido es más largos porque uno nunca sabe si su apellido es largo y para estar prevenido.

Ciudad es de 100 porque pueden ser extensas pero no tanto

Dirección es de 150 porque puede que sea muy extensa la dirección

Email es de 150 porque si pueden ser muy extensos y tiene que ser único el Email

## Inscripciones

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDInscripcion	Serial	PK	No	Único	--
IDEstudiante	Int	FK	No	--	--
IDCurso	Int	FK	No	--	--
FechalInscripcion	Date	--	No	--	Current Date
Calificacion	Smallint	--	Si	--	--

IDInscripcion es serial porque me da flojera hacerlo manual

IDEstudiante como el id de la tabla estudiante es serial ocupa un capo para cacharlo en esta tabla el cual puse int por mera comodidad y no serán muchos datos, se podría poner en cualquiera que sea numérico entero, tipo smallint, bigint, int y así

IDCurso lo mismo que con IDEstudiante

FechalInscripcion es la fecha que se inscribe el alumno y por defecto es la fecha de cuando se ingresa a la base de datos, pero puede ser agregado de forma manual

Calificacion es Smallint porque manejaré el sistema de 100 a 0, evitando decimales

## Cursos

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDCurso	Serial	PK	No	Único	--
IDDepartamento	Int	FK	No	--	--
NombreCurso	Varchar(150)	--	No	--	--
Descripcion	Text	--	Si	--	--
Creditos	Smallint	--	No	--	--
Semestre	Smallint	--	No	--	--

IDCurso lo puse serial para que no se tenga que ingresar el id de forma manual

IDDepartamento como el id de la tabla departamento es serial ocupa un capo para cacharlo en esta tabla el cual puse Int por comodidad y no serán muchos datos, se podría poner en cualquiera que sea numérico entero, tipo smallint, bigint, int y así

NombreCurso lo puse como Varchar(150) porque hay materias con nombres bien largos

Descripcion lo puse Text por que las descripciones de los cursos pueden ser muy largas

Creditos lo puse Smallint porque no dan tantos créditos por curso en el sistema que voy a usar

Semestre lo puse Smallint porque en teoria deberian ser pocos semestres

## Departamentos

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDDepartamento	Serial	PK	No	Único	--
NombreDepartamento	Varchar(150)	--	No	--	--
Edificio	Varchar(50)	--	No	--	--

IDDepartamento lo puse serial para que no se tenga que ingresar el id de forma manual

NombreDepartamento lo puse Varchar(150) porque pueden ser nombres largos

Edificio le puse Varchar(50) por si se terminan expandiendo los nombres de los edificios

## Profesores

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDProfesor	Serial	PK	No	--	--
IDDepartamento	Int	--	No	--	--
Nombre	Varchar(50)	--	No	--	--
Apellido	Varchar(100)	--	No	--	--
Titulo	Varchar(150)	--	No	--	--

IDProfesor lo puse serial para que no se tenga que ingresar el id de forma manual

IDDepartamento como el id de la tabla departamento es serial ocupa un capo para cacharlo en esta tabla el cual puse Int por comodidad y no serán muchos datos, se podría poner en cualquiera que sea numérico entero, tipo smallint, bigint, int y así

Nombre es Varchar(50) porque estoy asumiendo que todos los alumnos pueden tener uno o dos nombres, raro sería que tengan tres en adelante

Apellido Varchar(100) es más largos porque uno nunca sabe si su apellido es largo y para estar prevenido

Titulo es Varchar(150) por que hay títulos muy largos uno nunca sabe

## CursosProfesores

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDCursoProfesor	Serial	PK	No	Único	--
IDCurso	Int	FK	No	--	--
IDProfesor	Int	FK	No	--	--

IDCursoProfesor lo puse serial para que no se tenga que ingresar el id de forma manual

IDCurso como el id de la tabla Curso es serial ocupa un capo para cacharlo en esta tabla el cual puse Int por comodidad y no serán muchos datos, se podría poner en cualquiera que sea numérico entero, tipo smallint, bigint, int y así

IDProfesor como el id de la tabla Profesores es serial ocupa un capo para cacharlo en esta tabla el cual puse Int por comodidad y no serán muchos datos, se podría poner en cualquiera que sea numérico entero, tipo smallint, bigint, int y así

## Aulas

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDAula	Serial	PK	No	Único	--
NombreAula	Varchar(50)	--	No	--	--
Capacidad	Smallint	--	No	--	--
Ubicacion	Varchar(50)	--	No	--	--

IDAula lo puse serial para que no se tenga que ingresar el id de forma manual

NombreAula le puse Varchar(50) por si se termina extendiendo el nombre

Capacidad lo puse como Smallint porque como la capacidad no es tan grande cabe en este tipo de dato

Ubicacion le puse Varchar(50) por si se termina extendiendo la ubicación

## Horarios

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDHorario	Serial	PK	No	Único	--
IDCurso	Int	FK	No	--	--
IDAula	Int	FK	No	--	--
FechalInicio	Date	--	No	--	--
FechaFin	Date	--	No	--	--
HoralInicio	Time	--	No	Check	--
HoraFin	Time	--	No	Check	--

IDHorario lo puse serial para que no se tenga que ingresar el id de forma manual

IDCurso lo mismo que en las anteriores llaves foráneas

IDAula lo mismo que en las anteriores llaves foráneas

FechalInicio lo puse como date para capturar en formato

FechaFin lo puse como date para capturar en formato fecha

HoralInicio lo puse como time para capturar en formato hora

HoraFin lo puse como time para capturar en formato hora

## ProgramasEstudio

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDPrograma	Serial	PK	No	Único	--
NombrePrograma	Varchar(50)	--	No	--	--
DescripcionPrograma	Text	--	No	--	--

IDPrograma lo puse serial para que no se tenga que ingresar el id de forma manual

NombrePrograma lo puse como Varchar(50) para tener una longitud de caracteres buena para el nombre del programa

DescripcionPrograma lo puse como Text porque las descripciones son largas

## ProgramasCursos

Campo	Tipo de dato	PK / FK	Nulo	Restricción	Valor por defecto
IDProgramaCurso	Serial	PK	No	Único	--
IDPrograma	Int	FK	No	--	--
IDCurso	Int	FK	No	--	--

IDProgramaCurso lo puse serial para que no se tenga que ingresar el id de forma manual

IDPrograma lo mismo que en las anteriores llaves foráneas

IDCurso lo mismo que en las anteriores llaves foráneas

3. **Creando las Tablas (DDL):** Utilizar un sistema de gestión de bases de datos para crear las tablas definidas en el paso anterior.

--Tabla Estudiantes

CREATE TABLE Estudiantes(

```
IDEstudiante SERIAL PRIMARY KEY,
Nombre VARCHAR(50) NOT NULL,
Apellido VARCHAR(100) NOT NULL,
FechaNacimiento DATE NOT NULL,
Direccion VARCHAR(150) NOT NULL,
Ciudad VARCHAR(100) NOT NULL,
Email VARCHAR(150) NOT NULL UNIQUE
```

);

--Tabla Departamentos

CREATE TABLE Departamentos(

```
IDDepartamento SERIAL PRIMARY KEY,
NombreDepartamento VARCHAR(150) NOT NULL,
Edificio VARCHAR(50) NOT NULL
```

);

--Tabla Aulas

CREATE TABLE Aulas(

```
IDAula SERIAL PRIMARY KEY,
NombreAula VARCHAR(50) NOT NULL,
Capacidad SMALLINT NOT NULL,
Ubicacion VARCHAR(50) NOT NULL
```

);

--Tabla ProgramasEstudio

CREATE TABLE ProgramasEstudio(

```
IDPrograma SERIAL PRIMARY KEY,
NombrePrograma VARCHAR(50) NOT NULL,
DescripcionPrograma TEXT NOT NULL
```

);

--Tabla Cursos

```
CREATE TABLE Cursos(
    IDCurso SERIAL PRIMARY KEY,
    IDDepartamento INT NOT NULL,
    NombreCurso VARCHAR(150) NOT NULL,
    Descripcion TEXT,
    Creditos SMALLINT NOT NULL,
    Semestre SMALLINT NOT NULL,
    FOREIGN KEY (IDDepartamento) REFERENCES Departamentos(IDDepartamento)
ON DELETE CASCADE
);
```

--Tabla Inscripciones

```
CREATE TABLE Inscripciones(
    IDInscripcion SERIAL PRIMARY KEY,
    IDEstudiante INT NOT NULL,
    IDCurso INT NOT NULL,
    FechaInscripcion DATE DEFAULT CURRENT_DATE,
    Calificacion SMALLINT NOT NULL,
    FOREIGN KEY (IDEstudiante) REFERENCES Estudiantes(IDEstudiante) ON
DELETE CASCADE,
    FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCURSO) ON DELETE CASCADE
);
```

--Tabla Profesores

```
CREATE TABLE Profesores(
    IDProfesor SERIAL PRIMARY KEY,
    IDDepartamento INT NOT NULL,
    Nombre VARCHAR(50) NOT NULL,
    Apellido VARCHAR(100) NOT NULL,
    Titulo VARCHAR(150) NOT NULL,
    FOREIGN KEY (IDDepartamento) REFERENCES Departamentos(IDDepartamento)
ON DELETE CASCADE
);
```

--Tabla CursosProfesores

```
CREATE TABLE CursosProfesores(
    IDCursoProfesor SERIAL PRIMARY KEY,
    IDCurso INT NOT NULL,
    IDProfesor INT NOT NULL,
    FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCURSO) ON DELETE CASCADE,
    FOREIGN KEY (IDProfesor) REFERENCES Profesores(IDProfesor) ON DELETE
CASCADE
);
```

--Tabla Horarios

```
CREATE TABLE Horarios(
```

```
    IDHorario SERIAL PRIMARY KEY,  
    IDCurso INT NOT NULL,  
    IDAula INT NOT NULL,  
    FechaInicio DATE NOT NULL,  
    FechaFin DATE NOT NULL,  
    HoraInicio TIME NOT NULL,  
    HoraFin TIME NOT NULL,  
    FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCurso) ON DELETE CASCADE,  
    FOREIGN KEY (IDAula) REFERENCES Aulas(IDAula) ON DELETE CASCADE,  
    CHECK (FechaInicio <= FechaFin),  
    CHECK (HoraInicio < HoraFin)
```

```
);
```

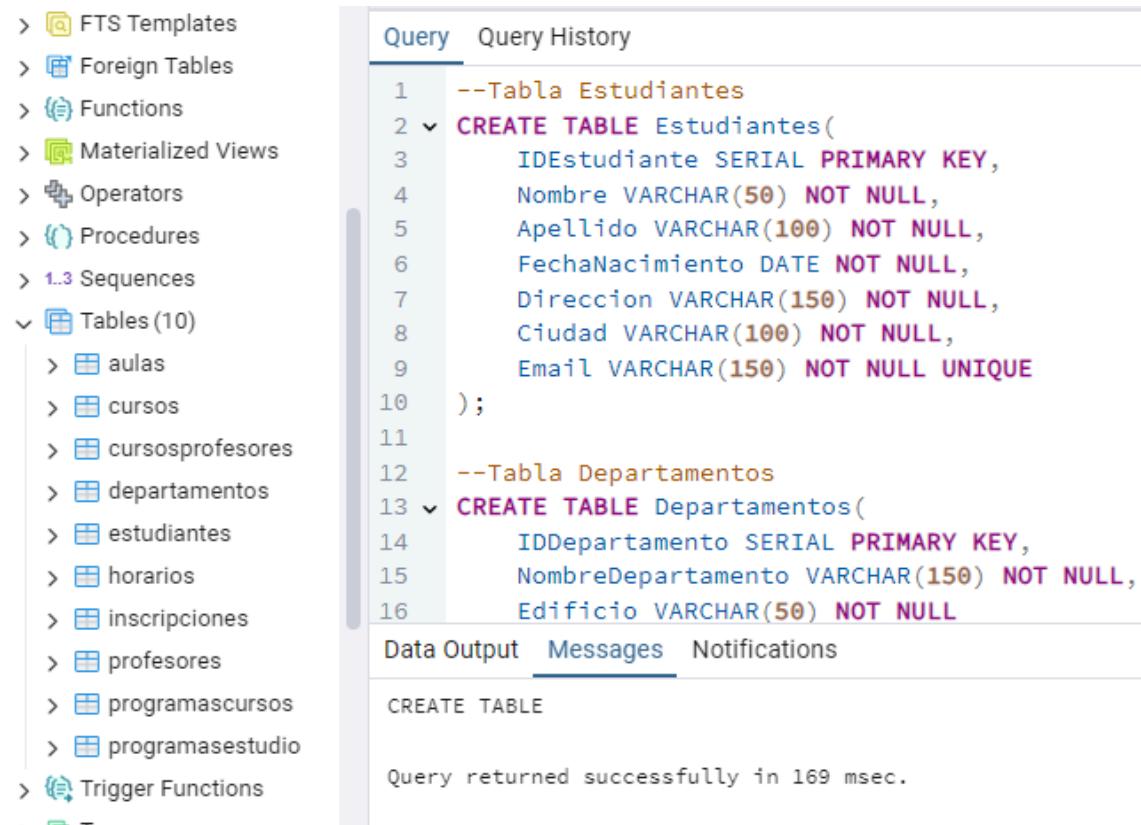
--Tabla ProgramasCursos

```
CREATE TABLE ProgramasCursos(
```

```
    IDProgramaCurso SERIAL PRIMARY KEY,  
    IDPrograma INT NOT NULL,  
    IDCurso INT NOT NULL,  
    FOREIGN KEY (IDPrograma) REFERENCES ProgramasEstudio(IDPrograma) ON  
DELETE CASCADE,  
    FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCurso) ON DELETE CASCADE
```

```
);
```

Resultado:



```

Query History
1 --Tabla Estudiantes
2 ▾ CREATE TABLE Estudiantes(
3     IDEstudiante SERIAL PRIMARY KEY,
4     Nombre VARCHAR(50) NOT NULL,
5     Apellido VARCHAR(100) NOT NULL,
6     FechaNacimiento DATE NOT NULL,
7     Direccion VARCHAR(150) NOT NULL,
8     Ciudad VARCHAR(100) NOT NULL,
9     Email VARCHAR(150) NOT NULL UNIQUE
10 );
11
12 --Tabla Departamentos
13 ▾ CREATE TABLE Departamentos(
14    >IDDepartamento SERIAL PRIMARY KEY,
15     NombreDepartamento VARCHAR(150) NOT NULL,
16     Edificio VARCHAR(50) NOT NULL

Data Output Messages Notifications
CREATE TABLE

Query returned successfully in 169 msec.

```

4. **Modificando las Tablas (DDL):** Realizar modificaciones a la estructura de las tablas, como agregar, modificar o eliminar campos, según sea necesario.

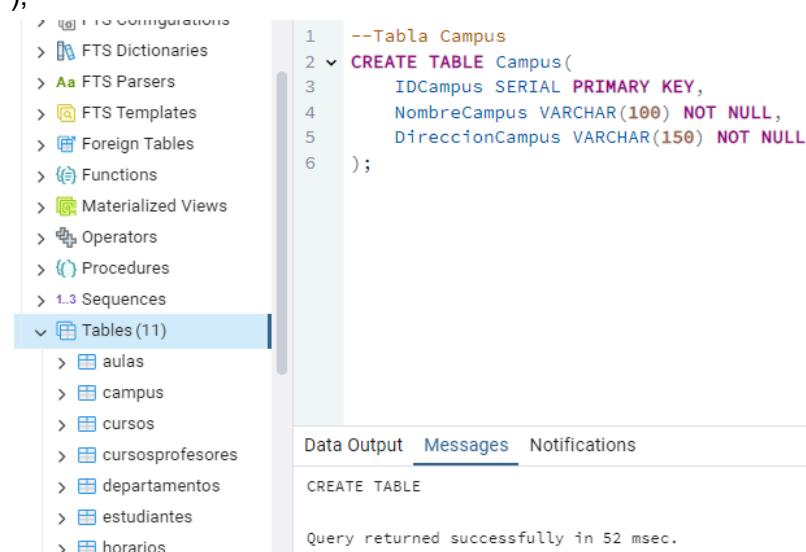
- **Agregar Tablas:**

- Tabla: Campus

- IDCampus (Clave Principal)
- NombreCampus
- DireccionCampus

--Tabla Campus

```
CREATE TABLE Campus(
    IDCampus SERIAL PRIMARY KEY,
    NombreCampus VARCHAR(100) NOT NULL,
    DireccionCampus VARCHAR(150) NOT NULL
);
```



```

--Tabla Campus
CREATE TABLE Campus(
    IDCampus SERIAL PRIMARY KEY,
    NombreCampus VARCHAR(100) NOT NULL,
    DireccionCampus VARCHAR(150) NOT NULL
);

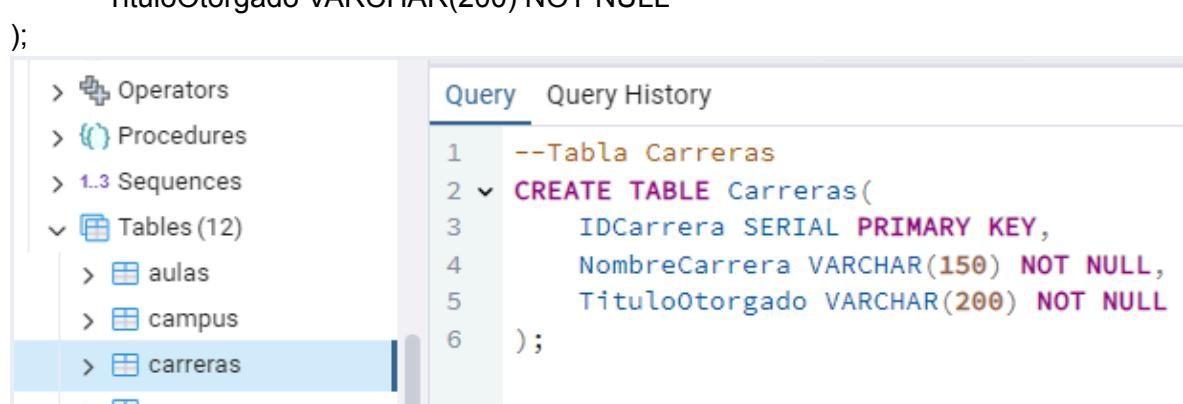
```

- Tabla: Carreras

- IDCarrera (Clave Principal)
- NombreCarrera
- TituloOtorgado

--Tabla Carreras

```
CREATE TABLE Carreras(
    IDCarrera SERIAL PRIMARY KEY,
    NombreCarrera VARCHAR(150) NOT NULL,
    TituloOtorgado VARCHAR(200) NOT NULL
);
```



```

--Tabla Carreras
CREATE TABLE Carreras(
    IDCarrera SERIAL PRIMARY KEY,
    NombreCarrera VARCHAR(150) NOT NULL,
    TituloOtorgado VARCHAR(200) NOT NULL
);

```

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

--Tabla EstudiantesCarreras

```
CREATE TABLE EstudiantesCarreras(
```

```
    IDEstudiantesCarreras SERIAL PRIMARY KEY,
    IDEstudiante INT NOT NULL,
    IDCampus INT NOT NULL,
    FOREIGN KEY (IDEstudiante) REFERENCES Estudiantes(IDEstudiante) ON
    DELETE CASCADE,
    FOREIGN KEY (IDCampus) REFERENCES Campus(IDCampus) ON DELETE
    CASCADE
```

```
);
```

```
--Tabla EstudiantesCarreras
CREATE TABLE EstudiantesCarreras(
    IDEstudiantesCarreras SERIAL PRIMARY KEY,
    IDEstudiante INT NOT NULL,
    IDCampus INT NOT NULL,
    FOREIGN KEY (IDEstudiante) REFERENCES Estudiantes(IDEstudiante) ON DELETE CASCADE,
    FOREIGN KEY (IDCampus) REFERENCES Campus(IDCampus) ON DELETE CASCADE
);
```

- Modificar Tablas:

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

--Para eliminar la tabla intermedia

```
DROP TABLE IF EXISTS EstudiantesCarreras;
```

```
DROP TABLE IF EXISTS EstudiantesCarreras;
```

```
ALTER TABLE Estudiantes
```

```
ADD COLUMN IDCarrera INT,
```

```
ADD CONSTRAINT fk_carrera
```

```
FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera);
```

```
ALTER TABLE Estudiantes
ADD COLUMN IDCarrera INT,
ADD CONSTRAINT fk_carrera
FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera);
```

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

ALTER TABLE Cursos

ADD COLUMN IDCampus INT,

ADD CONSTRAINT fk\_campus

FOREIGN KEY (IDCampus) REFERENCES Campus(IDCampus) ON DELETE CASCADE;

The screenshot shows a database interface with a sidebar containing navigation links like Materialized Views, Operators, Procedures, Sequences, and a list of Tables (12). The main area has tabs for Query, Data Output, Messages, and Notifications. The Query tab displays the SQL command:

```

1 < ALTER TABLE Cursos
2   ADD COLUMN IDCampus INT,
3   ADD CONSTRAINT fk_campus
4   FOREIGN KEY (IDCampus) REFERENCES Campus(IDCampus) ON DELETE CASCADE;
5

```

The Data Output tab shows the result of the query:

```

ALTER TABLE
Query returned successfully in 73 msec.

```

- Modificar la tabla Profesores para incluir un campo Email

ALTER TABLE Profesores

ADD COLUMN Email VARCHAR(150) NOT NULL UNIQUE;

The screenshot shows a database interface with a sidebar containing navigation links like Materialized Views, Operators, Procedures, Sequences, and a list of Tables (12). The main area has tabs for Query, Data Output, Messages, and Notifications. The Query tab displays the SQL command:

```

1 < ALTER TABLE Profesores
2   ADD COLUMN Email VARCHAR(150) NOT NULL UNIQUE;
3

```

The Data Output tab shows the result of the query:

```

ALTER TABLE
Query returned successfully in 58 msec.

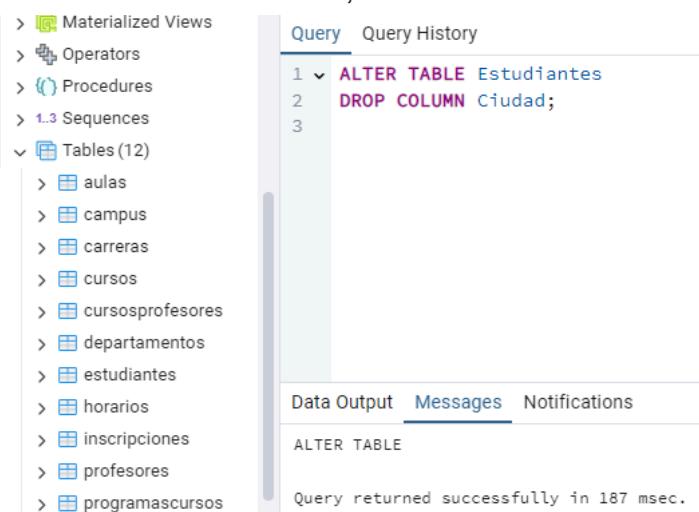
```

- **Eliminar Tablas/Campos**

- Eliminar la columna Ciudad de la tabla Estudiantes.

**ALTER TABLE Estudiantes**

**DROP COLUMN Ciudad;**

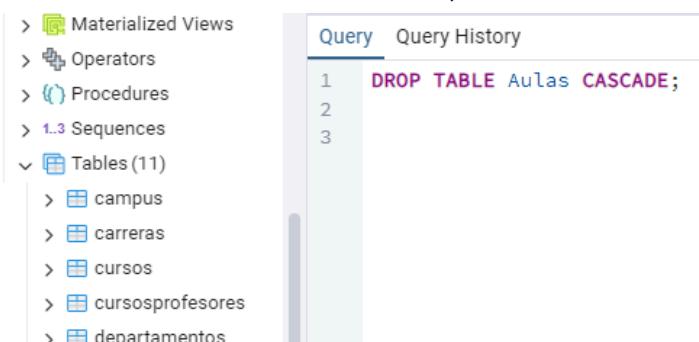


```
Query Query History
1 ▼ ALTER TABLE Estudiantes
2   DROP COLUMN Ciudad;
3

Data Output Messages Notifications
ALTER TABLE
Query returned successfully in 187 msec.
```

- Eliminar la tabla Aulas

**DROP TABLE Aulas CASCADE;**

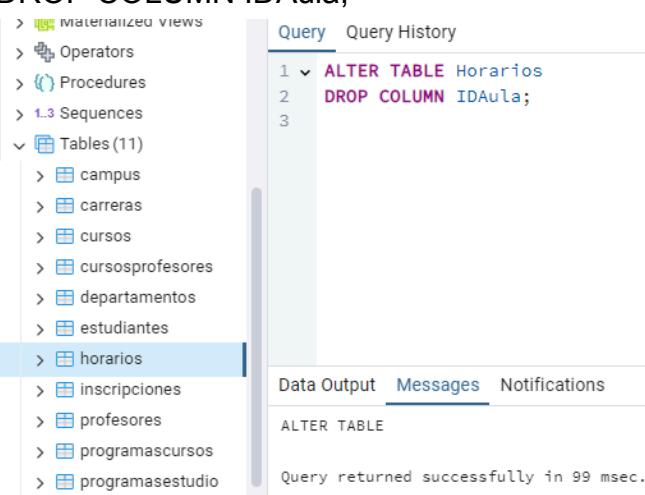


```
Query Query History
1 ▼ DROP TABLE Aulas CASCADE;
2
3

Data Output Messages Notifications
ALTER TABLE
Query returned successfully in 100 msec.
```

**ALTER TABLE Horarios**

**DROP COLUMN IDAula;**



```
Query Query History
1 ▼ ALTER TABLE Horarios
2   DROP COLUMN IDAula;
3

Data Output Messages Notifications
ALTER TABLE
Query returned successfully in 99 msec.
```

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

INSERT INTO Campus (NombreCampus, DireccionCampus)

VALUES

('Campus Central', 'Av. Principal 100, Ciudad A'),  
 ('Campus Norte', 'Calle Norte 50, Ciudad B'),  
 ('Campus Sur', 'Boulevard Sur 200, Ciudad C'),  
 ('Campus Tecnológico', 'Av. Innovación 300, Ciudad D'),  
 ('Campus Ciencias', 'Calle Ciencia 75, Ciudad E');

Query Query History

```
1 SELECT * FROM Campus;
```

Data Output Messages Notifications

	idcampus [PK] integer	nombrécampus character varying (100)	direccioncampus character varying (150)
1	1	Campus Central	Av. Principal 100, Ciudad A
2	2	Campus Norte	Calle Norte 50, Ciudad B
3	3	Campus Sur	Boulevard Sur 200, Ciudad C
4	4	Campus Tecnológico	Av. Innovación 300, Ciudad D
5	5	Campus Ciencias	Calle Ciencia 75, Ciudad E

INSERT INTO Carreras (NombreCarrera, TituloOtorgado)

VALUES

('Ingeniería en Sistemas Computacionales', 'Ingeniero en Sistemas'),  
 ('Ingeniería Civil', 'Ingeniero Civil'),  
 ('Ingeniería Mecánica', 'Ingeniero Mecánico'),  
 ('Ingeniería Electrónica', 'Ingeniero Electrónico'),  
 ('Ingeniería Industrial', 'Ingeniero Industrial'),  
 ('Ingeniería Química', 'Ingeniero Químico'),  
 ('Ingeniería Ambiental', 'Ingeniero Ambiental'),  
 ('Ingeniería en Telecomunicaciones', 'Ingeniero en Telecomunicaciones');

Query Query History

```
1 SELECT * FROM Carreras;
```

Data Output Messages Notifications

	idcarrera [PK] integer	nombrécarrera character varying (150)	titulootorgado character varying (200)
1	1	Ingeniería en Sistemas Computacionales	Ingeniero en Sistemas
2	2	Ingeniería Civil	Ingeniero Civil
3	3	Ingeniería Mecánica	Ingeniero Mecánico
4	4	Ingeniería Electrónica	Ingeniero Electrónico
5	5	Ingeniería Industrial	Ingeniero Industrial
6	6	Ingeniería Química	Ingeniero Químico
7	7	Ingeniería Ambiental	Ingeniero Ambiental
8	8	Ingeniería en Telecomunicaciones	Ingeniero en Telecomunicaciones

```
INSERT INTO Estudiantes (Nombre, Apellido, FechaNacimiento, Direccion, Email, IDCarrera)
VALUES
```

```
('Luis', 'García', '1999-04-12', 'Calle 123, Ciudad Alta', 'luis.garcia@email.com', 1),
('María', 'López', '2000-07-20', 'Avenida Central 456, Ciudad Baja', 'maria.lopez@email.com', 3),
('Carlos', 'Ramírez', '1998-11-05', 'Calle Falsa 789, Ciudad Costa', 'carlos.ramirez@email.com', 5),
('Ana', 'Martínez', '2001-02-28', 'Boulevard Norte 234, Ciudad Destino', 'ana.martinez@email.com', 2),
('Jorge', 'Sánchez', '1997-12-15', 'Calle Sur 345, Ciudad Enel', 'jorge.sanchez@email.com', 4),
('Laura', 'Fernández', '1999-09-09', 'Avenida Este 678, Ciudad Faro', 'laura.fernandez@email.com', 6),
('Pedro', 'Vargas', '2000-03-22', 'Calle Oeste 901, Ciudad Gala', 'pedro.vargas@email.com', 7),
('Sofía', 'Rojas', '1998-06-30', 'Calle Luna 123, Ciudad Hielo', 'sofia.rojas@email.com', 8);
```

Query History

```
1 SELECT * FROM Estudiantes;
```

Data Output Messages Notifications

Showing rows: 1 to 8							Page No:	1	of 1	<<	<<<	>>	>>>
	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (100)	fechanacimiento date	direccion character varying (150)	email character varying (150)	idcarrera integer						
1	1	Luis	García	1999-04-12	Calle 123, Ciudad Alta	luis.garcia@email.com	1						
2	2	Maria	López	2000-07-20	Avenida Central 456, Ciudad Baja	maria.lopez@email.com	3						
3	3	Carlos	Ramírez	1998-11-05	Calle Falsa 789, Ciudad Costa	carlos.ramirez@email.com	5						
4	4	Ana	Martínez	2001-02-28	Boulevard Norte 234, Ciudad Destino	ana.martinez@email.com	2						
5	5	Jorge	Sánchez	1997-12-15	Calle Sur 345, Ciudad Enel	jorge.sanchez@email.com	4						
6	6	Laura	Fernández	1999-09-09	Avenida Este 678, Ciudad Faro	laura.fernandez@email.com	6						
7	7	Pedro	Vargas	2000-03-22	Calle Oeste 901, Ciudad Gala	pedro.vargas@email.com	7						
8	8	Sofia	Rojas	1998-06-30	Calle Luna 123, Ciudad Hielo	sofia.rojas@email.com	8						

```
INSERT INTO Departamentos (NombreDepartamento, Edificio)
VALUES
```

```
('Departamento de Ingeniería en Sistemas', 'Edificio A'),
('Departamento de Ingeniería Civil', 'Edificio B'),
('Departamento de Ingeniería Mecánica', 'Edificio C'),
('Departamento de Ingeniería Electrónica', 'Edificio D'),
('Departamento de Ingeniería Industrial', 'Edificio E'),
('Departamento de Ingeniería Química', 'Edificio F'),
('Departamento de Ingeniería Ambiental', 'Edificio G'),
('Departamento de Ingeniería en Telecomunicaciones', 'Edificio H');
```

Query History

```
1 SELECT * FROM Departamentos;
```

Data Output Messages Notifications

Showing ro			
	iddepartamento [PK] integer	nombredepartamento character varying (150)	edificio character varying (50)
1	1	Departamento de Ingeniería en Sistemas	Edificio A
2	2	Departamento de Ingeniería Civil	Edificio B
3	3	Departamento de Ingeniería Mecánica	Edificio C
4	4	Departamento de Ingeniería Electrónica	Edificio D
5	5	Departamento de Ingeniería Industrial	Edificio E
6	6	Departamento de Ingeniería Química	Edificio F
7	7	Departamento de Ingeniería Ambiental	Edificio G
8	8	Departamento de Ingeniería en Telecomunicaciones	Edificio H

```
INSERT INTO Profesores (IDDepartamento, Nombre, Apellido, Titulo, Email)
VALUES
(1, 'José', 'Martínez', 'Doctor en Ciencias de la Computación', 'jose.martinez@uni.edu'),
(2, 'María', 'Gómez', 'Ingeniera Civil', 'maria.gomez@uni.edu'),
(3, 'Carlos', 'Pérez', 'Doctor en Ingeniería Mecánica', 'carlos.perez@uni.edu'),
(4, 'Ana', 'Rodríguez', 'Ingeniera Electrónica', 'ana.rodriguez@uni.edu'),
(5, 'Luis', 'Fernández', 'Maestro en Ingeniería Industrial', 'luis.fernandez@uni.edu'),
(6, 'Sofía', 'Vargas', 'Ingeniera Química', 'sofia.vargas@uni.edu'),
(7, 'Pedro', 'Luna', 'Doctor en Ciencias Ambientales', 'pedro.luna@uni.edu'),
(8, 'Laura', 'Ríos', 'Ingeniera en Telecomunicaciones', 'laura.rios@uni.edu');
```

Query    Query History

1    SELECT \* FROM Profesores;

Data Output    Messages    Notifications

	idprofesor [PK] integer	iddepartamento integer	nombre character varying (50)	apellido character varying (100)	titulo character varying (150)	email character varying (150)
1	1	1	José	Martínez	Doctor en Ciencias de la Computación	jose.martinez@uni.edu
2	2	2	María	Gómez	Ingeniera Civil	maria.gomez@uni.edu
3	3	3	Carlos	Pérez	Doctor en Ingeniería Mecánica	carlos.perez@uni.edu
4	4	4	Ana	Rodríguez	Ingeniera Electrónica	ana.rodriguez@uni.edu
5	5	5	Luis	Fernández	Maestro en Ingeniería Industrial	luis.fernandez@uni.edu
6	6	6	Sofía	Vargas	Ingeniera Química	sofia.vargas@uni.edu
7	7	7	Pedro	Luna	Doctor en Ciencias Ambientales	pedro.luna@uni.edu
8	8	8	Laura	Ríos	Ingeniera en Telecomunicaciones	laura.rios@uni.edu

```
INSERT INTO ProgramasEstudio (NombrePrograma, DescripcionPrograma)
VALUES
('Ingeniería en Sistemas Computacionales', 'Programa enfocado en el diseño y desarrollo de software y sistemas computacionales.'),
('Ingeniería Civil', 'Programa dedicado al diseño, construcción y mantenimiento de infraestructuras.'),
('Ingeniería Mecánica', 'Programa centrado en el diseño y fabricación de maquinaria y sistemas mecánicos.'),
('Ingeniería Electrónica', 'Programa orientado al diseño de circuitos electrónicos y sistemas de control.'),
('Ingeniería Industrial', 'Programa enfocado en la optimización de procesos productivos y gestión industrial.'),
('Ingeniería Química', 'Programa dedicado a la transformación de materias primas en productos químicos.'),
('Ingeniería Ambiental', 'Programa centrado en la protección y gestión del medio ambiente.'),
('Ingeniería en Telecomunicaciones', 'Programa orientado a la comunicación y transmisión de datos a distancia.');
```

Query    Query History

1    SELECT \* FROM ProgramasEstudio;

Data Output    Messages    Notifications

	idprograma [PK] integer	nombreprograma character varying (50)	descripcionprograma text
1	1	Ingeniería en Sistemas Computacionales	Programa enfocado en el diseño y desarrollo de software y sistemas computacionales.
2	2	Ingeniería Civil	Programa dedicado al diseño, construcción y mantenimiento de infraestructuras.
3	3	Ingeniería Mecánica	Programa centrado en el diseño y fabricación de maquinaria y sistemas mecánicos.
4	4	Ingeniería Electrónica	Programa orientado al diseño de circuitos electrónicos y sistemas de control.
5	5	Ingeniería Industrial	Programa enfocado en la optimización de procesos productivos y gestión industrial.
6	6	Ingeniería Química	Programa dedicado a la transformación de materias primas en productos químicos.
7	7	Ingeniería Ambiental	Programa centrado en la protección y gestión del medio ambiente.
8	8	Ingeniería en Telecomunicaciones	Programa orientado a la comunicación y transmisión de datos a distancia.

INSERT INTO Cursos (IDDepartamento, NombreCurso, Descripcion, Creditos, Semestre, IDCampus)  
 VALUES

- (1, 'Programación I', 'Introducción a la programación en lenguajes estructurados.', 3, 1, 1),
- (2, 'Estructuras de Concreto', 'Diseño y análisis de estructuras de concreto.', 4, 3, 2),
- (3, 'Mecánica de Fluidos', 'Principios básicos de la mecánica de fluidos.', 3, 2, 3),
- (4, 'Circuitos Eléctricos', 'Análisis y diseño de circuitos eléctricos básicos.', 3, 2, 4),
- (5, 'Gestión de Producción', 'Fundamentos de la gestión en procesos industriales.', 3, 4, 5),
- (6, 'Química Orgánica', 'Estudio de compuestos orgánicos y sus reacciones.', 4, 3, 1),
- (7, 'Ecología y Medio Ambiente', 'Principios ecológicos y conservación ambiental.', 3, 1, 2),
- (8, 'Telecomunicaciones Digitales', 'Sistemas y tecnologías de telecomunicaciones digitales.', 3, 5, 3);

Query    Query History

```
1  SELECT * FROM Cursos;
```

Data Output    Messages    Notifications

	idcurso [PK] integer	iddepartamento integer	nombrcurso character varying (150)	descripcion text	creditos smallint	semestre smallint	idcampus integer
1	1	1	Programación I	Introducción a la programación en lenguajes estructurados.	3	1	1
2	2	2	Estructuras de Concreto	Diseño y análisis de estructuras de concreto.	4	3	2
3	3	3	Mecánica de Fluidos	Principios básicos de la mecánica de fluidos.	3	2	3
4	4	4	Circuitos Eléctricos	Análisis y diseño de circuitos eléctricos básicos.	3	2	4
5	5	5	Gestión de Producción	Fundamentos de la gestión en procesos industriales.	3	4	5
6	6	6	Química Orgánica	Estudio de compuestos orgánicos y sus reacciones.	4	3	1
7	7	7	Ecología y Medio Ambiente	Principios ecológicos y conservación ambiental.	3	1	2
8	8	8	Telecomunicaciones Digitales	Sistemas y tecnologías de telecomunicaciones digitales.	3	5	3

INSERT INTO Horarios (IDCurso, FechaInicio, FechaFin, HoraInicio, HoraFin)  
 VALUES

- (1, '2025-08-01', '2025-12-15', '08:00:00', '10:00:00'),
- (2, '2025-08-01', '2025-12-15', '10:00:00', '12:00:00'),
- (3, '2025-08-02', '2025-12-16', '08:30:00', '10:30:00'),
- (4, '2025-08-02', '2025-12-16', '11:00:00', '13:00:00'),
- (5, '2025-08-03', '2025-12-17', '09:00:00', '11:00:00'),
- (6, '2025-08-03', '2025-12-17', '12:00:00', '14:00:00'),
- (7, '2025-08-04', '2025-12-18', '07:30:00', '09:30:00'),
- (8, '2025-08-04', '2025-12-18', '10:30:00', '12:30:00');

Query    Query History

```
1  SELECT * FROM Horarios;
```

Data Output    Messages    Notifications

	idhorario [PK] integer	idcurso integer	fechainicio date	fechafin date	horainicio time without time zone	horafin time without time zone
1	1	1	2025-08-01	2025-12-15	08:00:00	10:00:00
2	2	2	2025-08-01	2025-12-15	10:00:00	12:00:00
3	3	3	2025-08-02	2025-12-16	08:30:00	10:30:00
4	4	4	2025-08-02	2025-12-16	11:00:00	13:00:00
5	5	5	2025-08-03	2025-12-17	09:00:00	11:00:00
6	6	6	2025-08-03	2025-12-17	12:00:00	14:00:00
7	7	7	2025-08-04	2025-12-18	07:30:00	09:30:00
8	8	8	2025-08-04	2025-12-18	10:30:00	12:30:00

```
INSERT INTO ProgramasCursos (IDPrograma, IDCurso)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8);
```

Query    Query History

1    SELECT \* FROM ProgramasCursos;

Data Output    Messages    Notifications

	idprogramacurso [PK] integer	idprograma integer	idcurso integer
1		1	1
2		2	2
3		3	3
4		4	4
5		5	5
6		6	6
7		7	7
8		8	8

```
INSERT INTO CursosProfesores (IDCurso, IDProfesor)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8);
```

Query    Query History

1    SELECT \* FROM CursosProfesores;

Data Output    Messages    Notifications

	idcursoprofesor [PK] integer	idcurso integer	idprofesor integer
1		1	1
2		2	2
3		3	3
4		4	4
5		5	5
6		6	6
7		7	7
8		8	8



```
INSERT INTO Inscripciones (IDEstudiante, IDCurso, FechaInscripcion, Calificacion)
VALUES
(1, 1, '2025-01-15', 85),
(2, 2, '2025-01-16', 90),
(3, 3, '2025-01-17', 75),
(4, 4, '2025-01-18', 60),
(5, 5, '2025-01-19', 80),
(6, 6, '2025-01-20', 70),
(7, 7, '2025-01-21', 95),
(8, 8, '2025-01-22', 87);
```

Query    Query History

```
1  SELECT * FROM Inscripciones;
```

Data Output    Messages    Notifications

	idinscripcion [PK] integer	idestudiante integer	idcurso integer	fechainscripcion date	calificacion smallint
1	1	1	1	2025-01-15	85
2	2	2	2	2025-01-16	90
3	3	3	3	2025-01-17	75
4	4	4	4	2025-01-18	60
5	5	5	5	2025-01-19	80
6	6	6	6	2025-01-20	70
7	7	7	7	2025-01-21	95
8	8	8	8	2025-01-22	87

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

UPDATE Carreras

```
SET NombreCarrera = 'Ing. en Sistemas Computacionales'
```

```
WHERE NombreCarrera = 'Ingeniería en Sistemas Computacionales';
```

UPDATE Carreras

```
SET TituloOtorgado = 'Ingeniero Químico Industrial'
```

```
WHERE NombreCarrera = 'Ingeniería Química';
```

[Query](#) [Query History](#)

```
1 ✓ UPDATE Carreras
2   SET NombreCarrera = 'Ing. en Sistemas Computacionales'
3   WHERE NombreCarrera = 'Ingeniería en Sistemas Computacionales';
4
5 ✓ UPDATE Carreras
6   SET TituloOtorgado = 'Ingeniero Químico Industrial'
7   WHERE NombreCarrera = 'Ingeniería Química';
8
```

[Data Output](#) [Messages](#) [Notifications](#)

UPDATE 1

Query returned successfully in 88 msec.

[Query](#) [Query History](#)

```
1   SELECT * FROM Carreras;
```

[Data Output](#) [Messages](#) [Notifications](#)

	idcarrera [PK] integer	nombrercarrera character varying (150)	titulootorgado character varying (200)
1	2	Ingeniería Civil	Ingeniero Civil
2	3	Ingeniería Mecánica	Ingeniero Mecánico
3	4	Ingeniería Electrónica	Ingeniero Electrónico
4	5	Ingeniería Industrial	Ingeniero Industrial
5	7	Ingeniería Ambiental	Ingeniero Ambiental
6	8	Ingeniería en Telecomunicaciones	Ingeniero en Telecomunicaciones
7	1	Ing. en Sistemas Computacionales	Ingeniero en Sistemas
8	6	Ingeniería Química	Ingeniero Químico Industrial



UPDATE Estudiantes

SET Dirección = 'Calle Principal 321, Ciudad Alta',

Email = 'lgarcia@estudiantes.uni.edu'

WHERE Nombre = 'Luis' AND Apellido = 'García';

UPDATE Estudiantes

SET Email = 'mlopez@estudiantes.uni.edu'

WHERE Nombre = 'María' AND Apellido = 'López';

Query    Query History

---

```

1 UPDATE Estudiantes
2 SET Dirección = 'Calle Principal 321, Ciudad Alta',
3     Email = 'lgarcia@estudiantes.uni.edu'
4 WHERE Nombre = 'Luis' AND Apellido = 'García';
5
6 UPDATE Estudiantes
7 SET Email = 'mlopez@estudiantes.uni.edu'
8 WHERE Nombre = 'María' AND Apellido = 'López';
9

```

---

Data Output    Messages    Notifications

UPDATE 1

Query returned successfully in 68 msec.

Query    Query History

---

```

1 SELECT * FROM Estudiantes;

```

---

Data Output    Messages    Notifications

	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (100)	fechanacimiento date	direccion character varying (150)	email character varying (150)	idcarrera integer
1	3	Carlos	Ramírez	1998-11-05	Calle Falsa 789, Ciudad Costa	carlos.ramirez@email.com	5
2	4	Ana	Martínez	2001-02-28	Boulevard Norte 234, Ciudad Destino	ana.martinez@email.com	2
3	5	Jorge	Sánchez	1997-12-15	Calle Sur 345, Ciudad Enel	jorge.sanchez@email.com	4
4	6	Laura	Fernández	1999-09-09	Avenida Este 678, Ciudad Faro	laura.fernandez@email.com	6
5	7	Pedro	Vargas	2000-03-22	Calle Oeste 901, Ciudad Gala	pedro.vargas@email.com	7
6	8	Sofía	Rojas	1998-06-30	Calle Luna 123, Ciudad Hielo	sofia.rojas@email.com	8
7	1	Luis	García	1999-04-12	Calle Principal 321, Ciudad Alta	lgarcia@estudiantes.uni.edu	1
8	2	María	López	2000-07-20	Avenida Central 456, Ciudad Baja	mlopez@estudiantes.uni.e...	3

UPDATE Campus

SET NombreCampus = 'Campus Central Tecnológico'

WHERE NombreCampus = 'Campus Central';

UPDATE Campus

SET DireccionCampus = 'Av. Innovación 999, Ciudad A'

WHERE NombreCampus = 'Campus Central Tecnológico';

```
Query Query History
1 UPDATE Campus
2 SET NombreCampus = 'Campus Central Tecnológico'
3 WHERE NombreCampus = 'Campus Central';
4
5 UPDATE Campus
6 SET DireccionCampus = 'Av. Innovación 999, Ciudad A'
7 WHERE NombreCampus = 'Campus Central Tecnológico';
8

Data Output Messages Notifications
UPDATE 1

Query returned successfully in 95 msec.
```

```
Query Query History
1 SELECT * FROM Campus;

Data Output Messages Notifications
≡+ ↻ ↴ ↵ ↶ ↷ ↸ ↹ ↺ SQL



|   | idcampus [PK] integer | nombre campus character varying (100) | direccion campus character varying (150) |
|---|-----------------------|---------------------------------------|------------------------------------------|
| 1 | 2                     | Campus Norte                          | Calle Norte 50, Ciudad B                 |
| 2 | 3                     | Campus Sur                            | Boulevard Sur 200, Ciudad C              |
| 3 | 4                     | Campus Tecnológico                    | Av. Innovación 300, Ciudad...            |
| 4 | 5                     | Campus Ciencias                       | Calle Ciencia 75, Ciudad E               |
| 5 | 1                     | Campus Central Tecnológico            | Av. Innovación 999, Ciudad A             |


```

UPDATE Departamentos

SET NombreDepartamento = 'Depto. de Ingeniería en Sistemas'

WHERE NombreDepartamento = 'Departamento de Ingeniería en Sistemas';

Query Query History

```
1 UPDATE Departamentos
2 SET NombreDepartamento = 'Depto. de Ingeniería en Sistemas'
3 WHERE NombreDepartamento = 'Departamento de Ingeniería en Sistemas';
4
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 63 msec.

Query Query History

```
1 SELECT * FROM Departamentos;
```

Data Output Messages Notifications

	iddepartamento [PK] integer	nombredepartamento character varying (150)	edificio character varying (50)
1	2	Departamento de Ingeniería Civil	Edificio B
2	3	Departamento de Ingeniería Mecánica	Edificio C
3	4	Departamento de Ingeniería Electrónica	Edificio D
4	5	Departamento de Ingeniería Industrial	Edificio E
5	6	Departamento de Ingeniería Química	Edificio F
6	7	Departamento de Ingeniería Ambiental	Edificio G
7	8	Departamento de Ingeniería en Telecomunicaciones	Edificio H
8	1	Depto. de Ingeniería en Sistemas	Edificio A

UPDATE Profesores

```
SET Nombre = 'José Luis', Email = 'jose.luis@uni.edu'
WHERE Nombre = 'José' AND Apellido = 'Martínez';
```

UPDATE Profesores

```
SET Apellido = 'Gómez Ramírez'
WHERE Nombre = 'María' AND Apellido = 'Gómez';
```

[Query](#) [Query History](#)

```
1 UPDATE Profesores
2 SET Nombre = 'José Luis', Email = 'jose.luis@uni.edu'
3 WHERE Nombre = 'José' AND Apellido = 'Martínez';
4
5 UPDATE Profesores
6 SET Apellido = 'Gómez Ramírez'
7 WHERE Nombre = 'María' AND Apellido = 'Gómez';
8 |
```

[Data Output](#) [Messages](#) [Notifications](#)

UPDATE 1

Query returned successfully in 48 msec.

[Query](#) [Query History](#)

```
1 SELECT * FROM Profesores;
```

[Data Output](#) [Messages](#) [Notifications](#)

	idprofesor [PK] integer	iddepartamento integer	nombre character varying (50)	apellido character varying (100)	titulo character varying (150)	email character varying (150)
1	3	3	Carlos	Pérez	Doctor en Ingeniería Mecánica	carlos.perez@uni.edu
2	4	4	Ana	Rodríguez	Ingeniera Electrónica	ana.rodriguez@uni.edu
3	5	5	Luis	Fernández	Maestro en Ingeniería Industrial	luis.fernandez@uni.edu
4	6	6	Sofía	Vargas	Ingeniera Química	sofia.vargas@uni.edu
5	7	7	Pedro	Luna	Doctor en Ciencias Ambientales	pedro.luna@uni.edu
6	8	8	Laura	Ríos	Ingeniera en Telecomunicaciones	laura.rios@uni.edu
7	1	1	José Luis	Martínez	Doctor en Ciencias de la Computación	jose.luis@uni.edu
8	2	2	María	Gómez Ramírez	Ingeniera Civil	maria.gomez@uni.edu

UPDATE ProgramasEstudio

SET DescripcionPrograma = 'Programa con enfoque en software, redes y tecnologías emergentes.'

WHERE NombrePrograma = 'Ingeniería en Sistemas Computacionales';

```
Query Query History
1 UPDATE ProgramasEstudio
2 SET DescripcionPrograma = 'Programa con enfoque en software, redes y tecnologías emergentes.'
3 WHERE NombrePrograma = 'Ingeniería en Sistemas Computacionales';
4 |
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 73 msec.

```
Query Query History
1 SELECT * FROM ProgramasEstudio;
```

Data Output Messages Notifications

	idprograma [PK] integer	nombreprograma character varying (50)	descripcionprograma text
1	2	Ingeniería Civil	Programa dedicado al diseño, construcción y mantenimiento de infraestructuras.
2	3	Ingeniería Mecánica	Programa centrado en el diseño y fabricación de maquinaria y sistemas mecánicos.
3	4	Ingeniería Electrónica	Programa orientado al diseño de circuitos electrónicos y sistemas de control.
4	5	Ingeniería Industrial	Programa enfocado en la optimización de procesos productivos y gestión industrial.
5	6	Ingeniería Química	Programa dedicado a la transformación de materias primas en productos químicos.
6	7	Ingeniería Ambiental	Programa centrado en la protección y gestión del medio ambiente.
7	8	Ingeniería en Telecomunicaciones	Programa orientado a la comunicación y transmisión de datos a distancia.
8	1	Ingeniería en Sistemas Computacionales	Programa con enfoque en software, redes y tecnologías emergentes.

## UPDATE Cursos

```
SET NombreCurso = 'Programación Básica', Descripcion = 'Fundamentos de
programación con lenguaje C.'
WHERE NombreCurso = 'Programación I';
```

## UPDATE Cursos

```
SET Descripcion = 'Análisis estructural y diseño de elementos de concreto
reforzado.'
```

```
WHERE NombreCurso = 'Estructuras de Concreto';
```

Query    Query History

```
1 ✓ UPDATE CURSOS
2   SET NombreCurso = 'Programación Básica', Descripcion = 'Fundamentos de programación con lenguaje C.'
3   WHERE NombreCurso = 'Programación I';
4
5 ✓ UPDATE CURSOS
6   SET Descripcion = 'Análisis estructural y diseño de elementos de concreto reforzado.'
7   WHERE NombreCurso = 'Estructuras de Concreto';
8 |
```

Data Output    Messages    Notifications

UPDATE 1

Query returned successfully in 62 msec.

Query    Query History

```
1 SELECT * FROM Cursos;
```

Data Output    Messages    Notifications

	idcurso [PK] integer	iddepartamento integer	nombrcurso character varying (150)	descripcion text	creditos smallint	semestre smallint	idcampus integer
1	3	3	Mecánica de Fluidos	Principios básicos de la mecánica de fluidos.	3	2	3
2	4	4	Circuitos Eléctricos	Análisis y diseño de circuitos eléctricos básicos.	3	2	4
3	5	5	Gestión de Producción	Fundamentos de la gestión en procesos industriales.	3	4	5
4	6	6	Química Orgánica	Estudio de compuestos orgánicos y sus reacciones.	4	3	1
5	7	7	Ecología y Medio Ambiente	Principios ecológicos y conservación ambiental.	3	1	2
6	8	8	Telecomunicaciones Digitales	Sistemas y tecnologías de telecomunicaciones digitales.	3	5	3
7	1	1	Programación Básica	Fundamentos de programación con lenguaje C.	3	1	1
8	2	2	Estructuras de Concreto	Analisis estructural y diseño de elementos de concreto reforzado.	4	3	2

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

**DELETE FROM Cursos**

**WHERE NombreCurso = 'Gestión de Producción';**

```

query Query History
1 ▼ DELETE FROM Cursos
2   WHERE NombreCurso = 'Gestión de Producción';
3

Data Output Messages Notifications
DELETE 1

Query returned successfully in 116 msec.

```

**Data Output** **Messages** **Notifications**

	idcurso [PK] integer	iddepartamento integer	nombrcurso character varying (150)	descripcion text	creditos smallint	semestre smallint	idcampus integer
1	3	3	Mecánica de Fluidos	Principios básicos de la mecánica de fluidos.	3	2	3
2	4	4	Circuitos Eléctricos	Análisis y diseño de circuitos eléctricos básicos.	3	2	4
3	6	6	Química Orgánica	Estudio de compuestos orgánicos y sus reacciones.	4	3	1
4	7	7	Ecología y Medio Ambiente	Principios ecológicos y conservación ambiental.	3	1	2
5	8	8	Telecomunicaciones Digitales	Sistemas y tecnologías de telecomunicaciones digitales.	3	5	3
6	1	1	Programación Básica	Fundamentos de programación con lenguaje C.	3	1	1
7	2	2	Estructuras de Concreto	Análisis estructural y diseño de elementos de concreto reforzado.	4	3	2

**DELETE FROM Estudiantes**

**WHERE Email = 'jorge.sanchez@email.com';**

```

Query Query History
1 ▼ DELETE FROM Estudiantes
2   WHERE Email = 'jorge.sanchez@email.com';
3

Data Output Messages Notifications
DELETE 1

Query returned successfully in 120 msec.

```

**Data Output** **Messages** **Notifications**

	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (100)	fechanacimiento date	direccion character varying (150)	email character varying (150)	idcarrera integer
1	3	Carlos	Ramírez	1998-11-05	Calle Falsa 789, Ciudad Costa	carlos.ramirez@email.com	5
2	4	Ana	Martínez	2001-02-28	Boulevard Norte 234, Ciudad Destino	ana.martinez@email.com	2
3	6	Laura	Fernández	1999-09-09	Avenida Este 678, Ciudad Faro	laura.fernandez@email.com	6
4	7	Pedro	Vargas	2000-03-22	Calle Oeste 901, Ciudad Gala	pedro.vargas@email.com	7
5	8	Sofía	Rojas	1998-06-30	Calle Luna 123, Ciudad Hielo	sofia.rojas@email.com	8
6	1	Luis	García	1999-04-12	Calle Principal 321, Ciudad Alta	lgarcia@estudiantes.uni.edu	1
7	2	María	López	2000-07-20	Avenida Central 456, Ciudad Baja	mlopez@estudiantes.uni.e...	3

**DELETE FROM Profesores**  
**WHERE Email = 'maria.gomez@uni.edu';**

Query Query History

```
1 ✓ DELETE FROM Profesores
2 WHERE Email = 'maria.gomez@uni.edu';
3
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 59 msec.

Data Output Messages Notifications

Showing rows: 1 to 7  Page No: 1 of 1

	idprofesor [PK] integer	iddepartamento integer	nombre character varying (50)	apellido character varying (100)	titulo character varying (150)	email character varying (150)
1	3	3	Carlos	Pérez	Doctor en Ingeniería Mecánica	carlos.perez@uni.edu
2	4	4	Ana	Rodríguez	Ingeniera Electrónica	ana.rodriguez@uni.edu
3	5	5	Luis	Fernández	Maestro en Ingeniería Industrial	luis.fernandez@uni.edu
4	6	6	Sofía	Vargas	Ingeniera Química	sofia.vargas@uni.edu
5	7	7	Pedro	Luna	Doctor en Ciencias Ambientales	pedro.luna@uni.edu
6	8	8	Laura	Ríos	Ingeniera en Telecomunicaciones	laura.rios@uni.edu
7	1	1	José Luis	Martínez	Doctor en Ciencias de la Computación	jose.luis@uni.edu

**DELETE FROM Inscripciones**  
**WHERE IDEstudiante = 4 AND IDCurso = 4;**

Query Query History

```
1 ✓ DELETE FROM Inscripciones
2 WHERE IDEstudiante = 4 AND IDCurso = 4;
3
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 92 msec.

Data Output Messages Notifications

Showing rows: 1 to 6  Page No: 1 of 1

	idinscripcion [PK] integer	ideestudiante integer	idcurso integer	fechainscripcion date	calificacion smallint
1	1	1	1	2025-01-15	85
2	2	2	2	2025-01-16	90
3	3	3	3	2025-01-17	75
4	6	6	6	2025-01-20	70
5	7	7	7	2025-01-21	95
6	8	8	8	2025-01-22	87

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

### Consultas Específicas:

Los estudiantes deben formular consultas para responder a las siguientes solicitudes:

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

```
SELECT E.Nombre, E.Apellido
FROM Estudiantes E;
```

Query Query History

```
1 ▾ SELECT E.Nombre, E.Apellido
2   FROM Estudiantes E;
3
```

Data Output Messages Notifications

	nombre	apellido
character varying (50)	character varying (100)	
1	Carlos	Ramírez
2	Ana	Martínez
3	Laura	Fernández
4	Pedro	Vargas
5	Sofía	Rojas
6	Luis	García
7	Maria	López

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

```
SELECT C.NombreCurso
FROM Cursos C
WHERE C.Creditos = 3;
```

Query Query History

```
1 ▾ SELECT C.NombreCurso
2   FROM Cursos C
3   WHERE C.Creditos = 3;
```

Data Output Messages Notifica

	nombrcurso
character varying (150)	
1	Mecánica de Fluidos
2	Circuitos Eléctricos
3	Ecología y Medio Ambiente
4	Telecomunicaciones Digitales
5	Programación Básica

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

```
SELECT E.Nombre AS NombreEstudiante, C.NombreCurso AS NombreCurso
FROM Inscripciones I
INNER JOIN Estudiantes E ON I.IDEstudiante = E.IDEstudiante
INNER JOIN Cursos C ON I.IDCurso = C.IDCurso;
```

```
1 ✓ SELECT E.Nombre AS NombreEstudiante, C.NombreCurso AS NombreCurso
2   FROM Inscripciones I
3   INNER JOIN Estudiantes E ON I.IDEstudiante = E.IDEstudiante
4   INNER JOIN Cursos C ON I.IDCurso = C.IDCurso;
```

Data Output Messages Notifications

Showing

	nombreestudiante character varying (50)	nombrcurso character varying (150)
1	Luis	Programación Básica
2	Maria	Estructuras de Concreto
3	Carlos	Mecánica de Fluidos
4	Laura	Química Orgánica
5	Pedro	Ecología y Medio Ambiente
6	Sofía	Telecomunicaciones Digitales

4. 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).

```
SELECT E.Nombre AS NombreEstudiante,
E.Apellido AS ApellidoEstudiante,
C.NombreCurso AS NombreCurso
FROM Estudiantes E
LEFT JOIN Inscripciones I ON E.IDEstudiante = I.IDEstudiante
LEFT JOIN Cursos C ON I.IDCurso = C.IDCurso;
```

```
1 ✓ SELECT E.Nombre AS NombreEstudiante,
2   E.Apellido AS ApellidoEstudiante,
3   C.NombreCurso AS NombreCurso
4   FROM Estudiantes E
5   LEFT JOIN Inscripciones I ON E.IDEstudiante = I.IDEstudiante
6   LEFT JOIN Cursos C ON I.IDCurso = C.IDCurso;
7
```

Data Output Messages Notifications

Showing

	nombreestudiante character varying (50)	apellidoestudiante character varying (100)	nombrcurso character varying (150)
1	Luis	García	Programación Básica
2	Maria	López	Estructuras de Concreto
3	Carlos	Ramírez	Mecánica de Fluidos
4	Laura	Fernández	Química Orgánica
5	Pedro	Vargas	Ecología y Medio Ambiente
6	Sofía	Rojas	Telecomunicaciones Digitales
7	Ana	Martínez	[null]



5. **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.

```

SELECT C.NombreCurso,
E.Nombre AS NombreEstudiante,
E.Apellido AS ApellidoEstudiante
FROM Inscripciones I
RIGHT JOIN Cursos C ON I.IDCurso = C.IDCurso
LEFT JOIN Estudiantes E ON I.IDEstudiante = E.IDEstudiante;

```

```

1 ✓ SELECT C.NombreCurso,
2   E.Nombre AS NombreEstudiante,
3   E.Apellido AS ApellidoEstudiante
4   FROM Inscripciones I
5   RIGHT JOIN Cursos C ON I.IDCurso = C.IDCurso
6   LEFT JOIN Estudiantes E ON I.IDEstudiante = E.IDEstudiante;
7

```

Data Output Messages Notifications

	nombrecurso character varying (150)	nombreestudiante character varying (50)	apellidoestudiante character varying (100)
1	Programación Básica	Luis	García
2	Estructuras de Concreto	María	López
3	Mecánica de Fluidos	Carlos	Ramírez
4	Química Orgánica	Laura	Fernández
5	Ecología y Medio Ambiente	Pedro	Vargas
6	Telecomunicaciones Digitales	Sofía	Rojas
7	Circuitos Eléctricos	[null]	[null]

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

```

SELECT C.NombreCurso,
COUNT(I.IDEstudiante) AS CantidadEstudiantes
FROM Cursos C
LEFT JOIN Inscripciones I ON C.IDCurso = I.IDCurso
GROUP BY C.NombreCurso;

```

```

1 ✓ SELECT C.NombreCurso,
2   COUNT(I.IDEstudiante) AS CantidadEstudiantes
3   FROM Cursos C
4   LEFT JOIN Inscripciones I ON C.IDCurso = I.IDCurso
5   GROUP BY C.NombreCurso;
6

```

Data Output Messages Notifications

	nombrecurso character varying (150)	cantidadestudiantes bigint
1	Telecomunicaciones Digitales	1
2	Estructuras de Concreto	1
3	Mecánica de Fluidos	1
4	Circuitos Eléctricos	0
5	Programación Básica	1
6	Ecología y Medio Ambiente	1
7	Química Orgánica	1

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

```
SELECT *
FROM Estudiantes E
WHERE E.FechaNacimiento BETWEEN '1995-01-01' AND '1998-12-31';
```

```
1 ▾ SELECT *
2   FROM Estudiantes E
3 WHERE E.FechaNacimiento BETWEEN '1995-01-01' AND '1998-12-31';
```

Data Output Messages Notifications

	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (100)	fechanacimiento date	direccion character varying (150)	email character varying (150)	idcarrera integer
1	3	Carlos	Ramírez	1998-11-05	Calle Falsa 789, Ciudad Costa	carlos.ramirez@email.com	5
2	8	Sofía	Rojas	1998-06-30	Calle Luna 123, Ciudad Hielo	sofia.rojas@email.com	8

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

```
SELECT *
FROM Cursos C
ORDER BY C.NombreCurso ASC;
```

```
1 ▾ SELECT *
2   FROM Cursos C
3 ORDER BY C.NombreCurso ASC;
```

Data Output Messages Notifications

	idcurso [PK] integer	iddepartamento integer	nombrcurso character varying (150)	descripcion text	creditos smallint	semestre smallint	idcampus integer
1	4	4	Circuitos Eléctricos	Análisis y diseño de circuitos eléctricos básicos.	3	2	4
2	7	7	Ecología y Medio Ambiente	Principios ecológicos y conservación ambiental.	3	1	2
3	2	2	Estructuras de Concreto	Ánálisis estructural y diseño de elementos de concreto reforzado.	4	3	2
4	3	3	Mecánica de Fluidos	Principios básicos de la mecánica de fluidos.	3	2	3
5	1	1	Programación Básica	Fundamentos de programación con lenguaje C.	3	1	1
6	6	6	Química Orgánica	Estudio de compuestos orgánicos y sus reacciones.	4	3	1
7	8	8	Telecomunicaciones Digitales	Sistemas y tecnologías de telecomunicaciones digitales.	3	5	3



9. **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.

```
WITH InscripcionesPorEstudiante AS (
    SELECT E.IDEstudiante, E.Nombre, E.Apellido,
    COUNT(I.IDInscripcion) AS TotalInscripciones
    FROM Estudiantes E
    LEFT JOIN Inscripciones I ON E.IDEstudiante = I.IDEstudiante
    GROUP BY E.IDEstudiante, E.Nombre, E.Apellido
)
SELECT Nombre, Apellido, TotalInscripciones
FROM InscripcionesPorEstudiante
ORDER BY TotalInscripciones DESC
LIMIT 3;
```

```
1 v WITH InscripcionesPorEstudiante AS (
2     SELECT E.IDEstudiante, E.Nombre, E.Apellido,
3     COUNT(I.IDInscripcion) AS TotalInscripciones
4     FROM Estudiantes E
5     LEFT JOIN Inscripciones I ON E.IDEstudiante = I.IDEstudiante
6     GROUP BY E.IDEstudiante, E.Nombre, E.Apellido
7 )
8     SELECT Nombre, Apellido, TotalInscripciones
9     FROM InscripcionesPorEstudiante
10    ORDER BY TotalInscripciones DESC
11    LIMIT 3;
```

Data Output Messages Notifications

	nombre	apellido	totalinscripciones
1	Laura	Fernández	1
2	Maria	López	1
3	Pedro	Vargas	1

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

```

WITH InscripcionesPorCurso AS (
  SELECT D.Nombredepartamento AS NombreDepartamento,C.Nombrecurso AS
NombreCurso,
  COUNT(I.IDInscripcion) AS TotalInscripciones,
  D.IDDepartamento
  FROM Cursos C
  JOIN Departamentos D ON C.IDDepartamento = D.IDDepartamento
  LEFT JOIN Inscripciones I ON C.IDCurso = I.IDCurso
  GROUP BY D.Nombredepartamento, C.Nombrecurso, D.IDDepartamento,
C.IDCurso
),
CursoConMasEstudiantesPorDepartamento AS (
  SELECT NombreDepartamento, NombreCurso, TotalInscripciones,
  ROW_NUMBER() OVER (PARTITION BY IDDepartamento ORDER BY
TotalInscripciones DESC) AS Posicion
  FROM InscripcionesPorCurso
)
SELECT NombreDepartamento, NombreCurso, TotalInscripciones
FROM CursoConMasEstudiantesPorDepartamento
WHERE Posicion = 1;
  
```

```

1 WITH InscripcionesPorCurso AS (
2   SELECT D.Nombredepartamento AS NombreDepartamento,C.Nombrecurso AS NombreCurso,
3   COUNT(I.IDInscripcion) AS TotalInscripciones,
4   D.IDDepartamento
5   FROM Cursos C
6   JOIN Departamentos D ON C.IDDepartamento = D.IDDepartamento
7   LEFT JOIN Inscripciones I ON C.IDCurso = I.IDCurso
8   GROUP BY D.Nombredepartamento, C.Nombrecurso, D.IDDepartamento, C.IDCurso
9 ),
10 CursoConMasEstudiantesPorDepartamento AS (
11   SELECT NombreDepartamento, NombreCurso, TotalInscripciones
  
```

Data Output Messages Notifications

Showing rows: 1 to 7			
	nombredepartamento	nombrecurso	totalinscripciones
1	Dept. de Ingeniería en Sistemas	Programación Básica	1
2	Departamento de Ingeniería Civil	Estructuras de Concreto	1
3	Departamento de Ingeniería Mecánica	Mecánica de Fluidos	1
4	Departamento de Ingeniería Electrónica	Circuitos Eléctricos	0
5	Departamento de Ingeniería Química	Química Orgánica	1
6	Departamento de Ingeniería Ambiental	Ecología y Medio Ambiente	1
7	Departamento de Ingeniería en Telecomunicaciones	Telecomunicaciones Digitales	1

--Nota: Solo tengo un curso por departamento y un estudiante por curso menos en Circuitos Electricos



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

--Dado a que no tengo profesores que den 2 o más cursos tendre que insertar en la tabla profesores para mostrar que mi query sirve

```
INSERT INTO CursosProfesores (IDCurso, IDProfesor)
VALUES
(2, 1),
(3, 1);
```

Query    Query History

```
8 ✓ INSERT INTO CursosProfesores (IDCurso, IDProfesor)
9 VALUES
10 (2, 1),
11 (3, 1);|
```

Data Output    Messages    Notifications

```
INSERT 0 2

Query returned successfully in 60 msec.
```

```
SELECT P.Nombre, P.Apellido,
COUNT(CursosP.IDCurso) AS CantidadCursos
FROM Profesores P
INNER JOIN CursosProfesores CursosP ON P.IDProfesor = CursosP.IDProfesor
GROUP BY P.IDProfesor, P.Nombre, P.Apellido
HAVING COUNT(CursosP.IDCurso) > 2;
```

```
1 ✓ SELECT P.Nombre, P.Apellido,
2 COUNT(CursosP.IDCurso) AS CantidadCursos
3 FROM Profesores P
4 INNER JOIN CursosProfesores CursosP ON P.IDProfesor = CursosP.IDProfesor
5 GROUP BY P.IDProfesor, P.Nombre, P.Apellido
6 HAVING COUNT(CursosP.IDCurso) > 2;
```

Data Output    Messages    Notifications

Showing rows: 1 t

	nombre character varying (50)	apellido character varying (100)	cantidadcursos bigint
1	José Luis	Martínez	3

**12. 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.

WITH Promedios AS (

```

SELECT PE.IDPrograma, PE.Nombreprograma, C.IDCurso,
C.Nombrecurso, AVG(I.Calificacion) AS Promedio
FROM Programasestudio PE
INNER JOIN Programascursos PC ON PE.IDPrograma = PC.IDPrograma
INNER JOIN Cursos C ON pc.IDCurso = C.IDCurso
INNER JOIN Inscripciones I ON c.IDCurso = I.IDCurso
GROUP BY PE.IDPrograma, PE.Nombreprograma, C.IDCurso, C.Nombrecurso
),
```

Maximos AS (

```

SELECT IDPrograma,
MAX(Promedio) AS MaxPromedio
FROM Promedios
GROUP BY IDPrograma
)
```

```

SELECT P.Nombreprograma, P.Nombrecurso, P.Promedio
FROM Promedios P
```

```

JOIN Maximos M ON P.IDPrograma = M.IDPrograma AND P.Promedio =
M.MaxPromedio
```

```

ORDER BY P.Nombreprograma;
```

```

1 WITH Promedios AS (
2     SELECT PE.IDPrograma, PE.Nombreprograma, C.IDCurso,
3         C.Nombrecurso, AVG(I.Calificacion) AS Promedio
4     FROM Programasestudio PE
5     INNER JOIN Programascursos PC ON PE.IDPrograma = PC.IDPrograma
6     INNER JOIN Cursos C ON pc.IDCurso = C.IDCurso
7     INNER JOIN Inscripciones I ON c.IDCurso = I.IDCurso
8     GROUP BY PE.IDPrograma, PE.Nombreprograma, C.IDCurso, C.Nombrecurso
9 ),
10 Maximos AS (
11     SELECT IDPrograma,
12         MAX(Promedio) AS MaxPromedio
13     FROM Promedios
```

Data Output Messages Notifications

	nombreprograma character varying (50)	lock	nombrecurso character varying (150)	lock	promedio numeric	lock
1	Ingeniería Ambiental		Ecología y Medio Ambiente		95.000000000000000000	
2	Ingeniería Civil		Estructuras de Concreto		90.000000000000000000	
3	Ingeniería en Sistemas Computacionales		Programación Básica		85.000000000000000000	
4	Ingeniería en Telecomunicaciones		Telecomunicaciones Digitales		87.000000000000000000	
5	Ingeniería Mecánica		Mecánica de Fluidos		75.000000000000000000	
6	Ingeniería Química		Química Orgánica		70.000000000000000000	