



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Nombre del Estudiante:
Emanuel Padilla Valencia

Número de Control:
23490363

Nombre de la Materia:
Fundamentos de Bases de Datos

Nombre del Profesor:
José Ramón Bogarin Valenzuela

Examen Final de Unidad 4

Mexicali Baja California al 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):

- **Estudiantes**
- -IDEstudiante (Clave Principal)
- -Nombre ○ Apellido
- -FechaNacimiento
- -Direccion
- -Ciudad
- -Email

```
CREATE TABLE Estudiantes (  
    IDEstudiante SERIAL PRIMARY KEY,  
  
    Nombre VARCHAR(50) NOT NULL,  
    Apellido VARCHAR(50) NOT NULL,  
    FechaNacimiento DATE NOT NULL,  
    Direccion VARCHAR(50),  
    Ciudad VARCHAR(50),  
    Email VARCHAR(50) UNIQUE  
);
```

- **-Tabla: Cursos**
- -IDCurso (Clave Principal)
- -NombreCurso ○ Descripcion
- -Creditos
- -Semestre

- -IDDepartamento (Clave Foránea, referencia a la tabla Departamentos)

```
CREATE TABLE Cursos (
    IDCurso SERIAL PRIMARY KEY,
    IDDepartamento INT NOT NULL,
    NombreCurso VARCHAR(30) NOT NULL,
    Descripcion VARCHAR (75),
    Creditos INTEGER,
    Semestre INT NOT NULL,
    FOREIGN KEY (IDDepartamento) REFERENCES Departamentos (IDDepartamento)
);
```

- **Inscripciones**

- - IDInscripcion (Clave Principal)
- - IDEstudiante (Clave Foránea)
- - IDCurso (Clave Foránea)
- - FechaInscripcion
- - Calificacion

```
CREATE TABLE Incripciones (
    IDInscripcion SERIAL PRIMARY KEY,
    IDEstudiante INT NOT NULL,
    IDCurso INT NOT NULL,
    FechaInscripcion DATE NOT NULL,
    Calificacion INT NOT NULL,
    FOREIGN KEY (IDEstudiante) REFERENCES Estudiantes(IDEstudiante),
    FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCurso)
);
```

- **Profesores**

- - IDProfesor (Clave Principal)
- - Nombre
- - Apellido
- - Titulo
- - IDDepartamento (Clave Foránea)

```
CREATE TABLE Profesores (
    IDProfesor SERIAL PRIMARY KEY,
    IDDepartamento INT NOT NULL,
    Nombre_Profesor VARCHAR(25) NOT NULL,
    Apellido_Profesor VARCHAR(30) NOT NULL,
    Titulo VARCHAR(50),
    FOREIGN KEY (IDDepartamento) REFERENCES Departamentos (IDDepartamento)
);
```

- **Departamentos**

- - IDDepartamento (Clave Principal)
- - NombreDepartamento
- - Edificio

```
CREATE TABLE Departamentos (
    IDDepartamento SERIAL PRIMARY KEY,
    NombreDepartamento VARCHAR(30) NOT NULL,
    Edificio VARCHAR(30)
);
```

- **Aulas**

- - IDAula (Clave Principal)
- - NombreAula
- - Capacidad
- - Ubicacion

```
CREATE TABLE Aulas (
    IDAula SERIAL PRIMARY KEY,
    NombreAula VARCHAR(30) NOT NULL,
    Capacidad INT,
    Ubicacion VARCHAR(30)
);
```

- **Horarios**

- - IDHorario (Clave Principal)
- - IDCurso (Clave Foránea)
- - IDAula (Clave Foránea)
- - FechaInicio
- - FechaFin
- - HoraInicio
- - HoraFin

```
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),
    FOREIGN KEY (IDAula) REFERENCES Aulas(IDAula)
);
```

- **CursosProfesores** (Tabla Intermedia)

- - IDCursoProfesor (Clave Principal)
- - IDCurso (Clave Foránea)
- - IDProfesor (Clave Foránea)

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

- **ProgramasEstudio**
- - IDPrograma (Clave Principal)
- - NombrePrograma
- - DescripcionPrograma

```
CREATE TABLE ProgramasEstudio (
    IDPrograma SERIAL PRIMARY KEY,
    NombrePrograma VARCHAR(50) NOT NULL,
    DescripcionPrograma VARCHAR (75)
);
```

- **ProgramasCursos** (Tabla Intermedia)
- - IDProgramaCurso (Clave Principal)
- - IDPrograma (Clave Foránea)
- - IDCurso (Clave Foránea)

```
CREATE TABLE ProgramasCursos (
    IDProgramaCurso SERIAL PRIMARY KEY,
    IDPrograma INT NOT NULL,
    IDCurso INT NOT NULL,
    FOREIGN KEY (IDPrograma) REFERENCES ProgramasEstudio(IDPrograma),
    FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCurso)
);
```

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

Agregar Tablas:

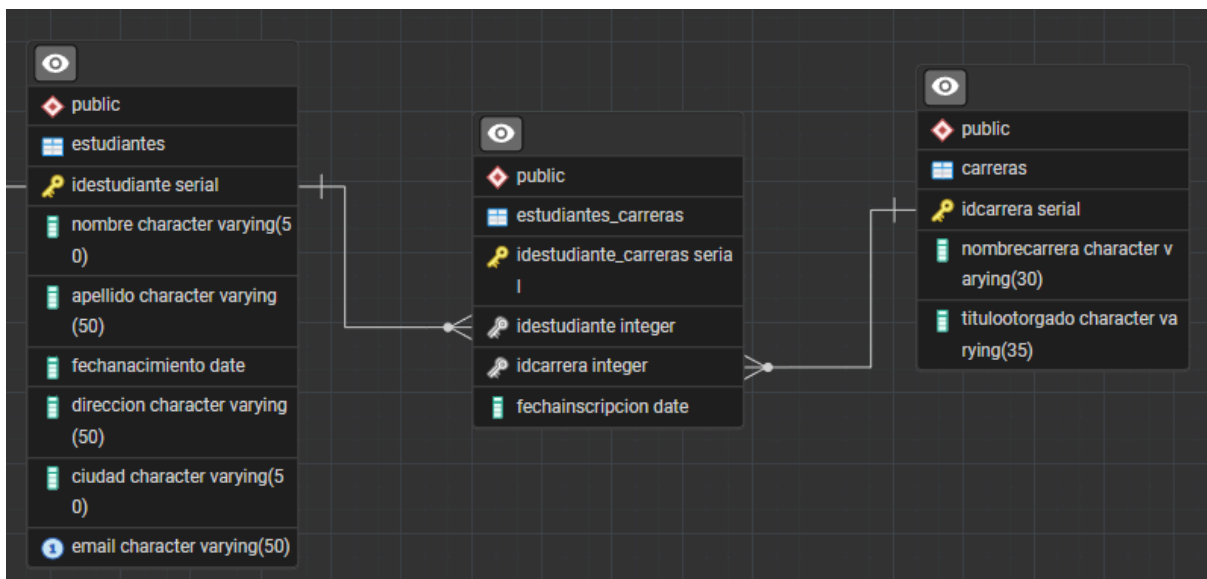
- **Tabla: Campus**
- IDCampus (Clave Principal)
- NombreCampus
- DireccionCampus
- **Tabla: Carreras**
- IDCarrera (Clave Principal)
- NombreCarrera
- TituloOtorgado

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

```
CREATE TABLE Campus (
    IDCampus SERIAL PRIMARY KEY,
    NombreCampus VARCHAR(50) NOT NULL,
    DireccionCampus VARCHAR(80)
);

CREATE TABLE Carreras (
    IDCarrera SERIAL PRIMARY KEY,
    NombreCarrera VARCHAR(30) NOT NULL,
    TituloOtorgado VARCHAR(35)
);

CREATE TABLE Estudiantes_Carreras (
    IDEstudiante_Carreras SERIAL PRIMARY KEY,
    IDEstudiante INT NOT NULL,
    IDCarrera INT NOT NULL,
    FechaInscripcion DATE DEFAULT CURRENT_DATE,
    FOREIGN KEY (IDEstudiante) REFERENCES Estudiantes(IDEstudiante),
    FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera)
);
```



Modificar tabla Estudiantes:

- Agregar campo: IDCarrera (Clave Foránea, referencia a tabla Carreras)

Eliminamos la relación muchos a muchos:

```
DROP TABLE Estudiantes_Carreras;
```

```
ALTER TABLE Estudiantes
ADD COLUMN IDCarrera INT;
```

```
ALTER TABLE Estudiantes
ADD CONSTRAINT FK_estudiante_carrera
FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera)
ON DELETE SET NULL;
```

idestudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fechanacimiento date	direccion character varying (50)
ciudad character varying (50)	email character varying (50)	idcarrera integer		

2. Modificar tabla Cursos:

- Agregar campo: IDCampus (Clave Foránea, referencia a tabla Campus)

```
ALTER TABLE Cursos
ADD COLUMN IDCampus INT,
ADD CONSTRAINT fk_curso_campus
FOREIGN KEY (IDCampus) REFERENCES Campus(IDCampus)
ON DELETE SET NULL;
```

idcurso [PK] integer	iddepartamento integer	nombrecurso character varying (30)	descripcion character varying (75)	creditos integer	semestre integer	idcampus integer
-------------------------	---------------------------	---------------------------------------	---------------------------------------	---------------------	---------------------	---------------------

3. Modificar tabla Profesores:

- Agregar campo: Email (Tipo de dato: texto/cadena)

```
ALTER TABLE Profesores
ADD COLUMN Email VARCHAR(50) UNIQUE;
```

idprofesor [PK] integer	iddepartamento integer	nombre_profesor character varying (25)	apellido_profesor character varying (30)	titulo character varying (50)	email character varying (50)
----------------------------	---------------------------	---	---	----------------------------------	---------------------------------

Eliminar Tablas/Campos

Eliminar columna de la tabla Estudiantes:

- Eliminar columna: Ciudad

```
ALTER TABLE Estudiantes
DROP COLUMN Ciudad;
```

Eliminar tabla:

- Eliminar tabla: Aulas

```
DROP TABLE Aulas;
```

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

Departamentos:

```
INSERT INTO Departamentos (NombreDepartamento, Edificio) VALUES
('Ingeniería', 'Edificio A'),
('Ciencias Sociales', 'Edificio B'),
('Administración', 'Edificio C'),
('Matemáticas', 'Edificio D'),
('Química', 'Edificio E'),
('Física', 'Edificio F'),
('Biología', 'Edificio G'),
('Derecho', 'Edificio H'),
('Artes', 'Edificio I'),
('Historia', 'Edificio J'),
('Idiomas', 'Edificio K'),
('Economía', 'Edificio L'),
('Filosofía', 'Edificio M'),
('Educación', 'Edificio N');
```

iddepartamento [PK] integer	nombredepartamento character varying (30)	edificio character varying (30)
1	Ingeniería	Edificio A
2	Ciencias Sociales	Edificio B
3	Administración	Edificio C
4	Matemáticas	Edificio D
5	Química	Edificio E
6	Física	Edificio F
7	Biología	Edificio G
8	Derecho	Edificio H
9	Artes	Edificio I
10	Historia	Edificio J
11	Idiomas	Edificio K
12	Economía	Edificio L
13	Filosofía	Edificio M
14	Educación	Edificio N

Campus

```
INSERT INTO Campus (NombreCampus, DireccionCampus) VALUES
('Campus Central', 'Av. Universidad 123'),
('Campus Norte', 'Calle Técnica 456'),
('Campus Sur', 'Calle Sur 200'),
('Campus Este', 'Av. Oriente 300'),
('Campus Oeste', 'Bulevar Poniente 400'),
('Campus Virtual', 'https://virtual.universidad.edu'),
('Campus Internacional', 'Av. Global 500'),
('Campus Experimental', 'Zona Rural 1'),
('Campus Central II', 'Calle Reforma 110'),
('Campus Científico', 'Parque Científico 2'),
('Campus Empresarial', 'Av. Negocios 10'),
('Campus Salud', 'Av. Médica 100');
```

idcampus [PK] integer	nombrecampus character varying (50)	direccioncampus character varying (80)
1	Campus Central	Av. Universidad 123
2	Campus Norte	Calle Técnica 456
3	Campus Sur	Calle Sur 200
4	Campus Este	Av. Oriente 300
5	Campus Oeste	Bulevar Poniente 400
6	Campus Virtual	https://virtual.universidad.edu
7	Campus Internacional	Av. Global 500
8	Campus Experimental	Zona Rural 1
9	Campus Central II	Calle Reforma 110
10	Campus Científico	Parque Científico 2
11	Campus Empresarial	Av. Negocios 10
12	Campus Salud	Av. Médica 100

Carreras:

```
INSERT INTO Carreras (NombreCarrera, TituloOtorgado) VALUES
('Ingeniería en Sistemas', 'Ingeniero en Sistemas'),
('Psicología', 'Licenciado en Psicología'),
('Administración de Empresas', 'Licenciado en Administración'),
('Ingeniería Industrial', 'Ingeniero Industrial'),
('Química', 'Licenciado en Química'),
('Física', 'Licenciado en Física'),
('Biología', 'Licenciado en Biología'),
('Derecho', 'Licenciado en Derecho'),
('Artes Plásticas', 'Licenciado en Artes'),
('Historia', 'Licenciado en Historia'),
('Lenguas Modernas', 'Licenciado en Lenguas'),
('Economía', 'Licenciado en Economía'),
('Pedagogía', 'Licenciado en Educación');
```

idcarrera [PK] integer	nombrecarrera character varying (30)	titulootorgado character varying (35)
1	Ingeniería en Sistemas	Ingeniero en Sistemas
2	Psicología	Licenciado en Psicología
3	Administración de Empresas	Licenciado en Administración
4	Ingeniería Industrial	Ingeniero Industrial
5	Química	Licenciado en Química
6	Física	Licenciado en Física
7	Biología	Licenciado en Biología
8	Derecho	Licenciado en Derecho
9	Artes Plásticas	Licenciado en Artes
10	Historia	Licenciado en Historia
11	Lenguas Modernas	Licenciado en Lenguas
12	Economía	Licenciado en Economía
13	Pedagogía	Licenciado en Educación

ProgramasEstudio:

```
INSERT INTO ProgramasEstudio (NombrePrograma, DescripcionPrograma) VALUES
('Programa de Ingeniería', 'Plan 2024 Ingeniería'),
('Programa de Psicología', 'Plan 2023 Psicología'),
('Programa de Química', 'Plan de estudios de Química'),
('Programa de Física', 'Plan de estudios de Física'),
('Programa de Biología', 'Plan de estudios de Biología'),
('Programa de Derecho', 'Plan de estudios de Derecho'),
('Programa de Artes', 'Plan de estudios de Artes'),
('Programa de Historia', 'Plan de estudios de Historia'),
('Programa de Lenguas', 'Plan de estudios de Lenguas'),
('Programa de Economía', 'Plan de estudios de Economía'),
('Programa de Filosofía', 'Plan de estudios de Filosofía'),
('Programa de Educación', 'Plan de estudios de Pedagogía');
```

idprograma [PK] integer	nombreprograma character varying (50)	descripcionprograma character varying (75)
1	Programa de Ingeniería	Plan 2024 Ingeniería
2	Programa de Psicología	Plan 2023 Psicología
3	Programa de Química	Plan de estudios de Química
4	Programa de Física	Plan de estudios de Física
5	Programa de Biología	Plan de estudios de Biología
6	Programa de Derecho	Plan de estudios de Derecho
7	Programa de Artes	Plan de estudios de Artes
8	Programa de Historia	Plan de estudios de Historia
9	Programa de Lenguas	Plan de estudios de Lenguas
10	Programa de Economía	Plan de estudios de Economía
11	Programa de Filosofía	Plan de estudios de Filosofía
12	Programa de Educación	Plan de estudios de Pedagogía

Profesores:

```
INSERT INTO Profesores (IDDepartamento, Nombre_Profesor, Apellido_Profesor, Titulo, Email) VALUES
(1, 'Carlos', 'Ramírez', 'Dr. en Ciencias Computacionales', 'cramirez@uni.edu'),
(2, 'Ana', 'López', 'Mtra. en Psicología', 'alopez@uni.edu'),
(3, 'Jorge', 'Gómez', 'Lic. en Administración', 'jgomez@uni.edu'),
(4, 'Laura', 'Ruiz', 'Dra. en Matemáticas', 'lruiz@uni.edu'),
(5, 'Pablo', 'Mendoza', 'Mtro. en Física', 'pmendoza@uni.edu'),
(6, 'Diana', 'Silva', 'Mtra. en Química', 'dsilva@uni.edu'),
(7, 'Raúl', 'Fernández', 'Lic. en Derecho', 'rfernandez@uni.edu'),
(8, 'Claudia', 'Herrera', 'Mtra. en Historia del Arte', 'cherrera@uni.edu'),
(9, 'Sergio', 'Cano', 'Lic. en Historia', 'scano@uni.edu'),
(10, 'Beatriz', 'Nava', 'Mtra. en Lingüística', 'bnava@uni.edu'),
(11, 'Tomás', 'Reyes', 'Mtro. en Economía', 'treyes@uni.edu'),
(12, 'Patricia', 'Torres', 'Lic. en Filosofía', 'ptorres@uni.edu'),
(1, 'Javier', 'Morales', 'Dr. en Computación', 'jmorales@uni.edu');
```

idprofesor [PK] integer	iddepartamento integer	nombre_profesor character varying (25)	apellido_profesor character varying (30)	titulo character varying (50)	email character varying (50)
1	1	Carlos	Ramírez	Dr. en Ciencias Computacionales	cramirez@uni.edu
2	2	Ana	López	Mtra. en Psicología	alopez@uni.edu
3	3	Jorge	Gómez	Lic. en Administración	jgomez@uni.edu
4	4	Laura	Ruiz	Dra. en Matemáticas	lruiz@uni.edu
5	5	Pablo	Mendoza	Mtro. en Física	pmendoza@uni.edu
6	6	Diana	Silva	Mtra. en Química	dsilva@uni.edu
7	7	Raúl	Fernández	Lic. en Derecho	rfernandez@uni.edu
8	8	Claudia	Herrera	Mtra. en Historia del Arte	cherrera@uni.edu
9	9	Sergio	Cano	Lic. en Historia	scano@uni.edu
10	10	Beatriz	Nava	Mtra. en Lingüística	bnava@uni.edu
11	11	Tomás	Reyes	Mtro. en Economía	treyes@uni.edu
12	12	Patricia	Torres	Lic. en Filosofía	ptorres@uni.edu
13	1	Javier	Morales	Dr. en Computación	jmorales@uni.edu

Estudiantes:

```
INSERT INTO Estudiantes (Nombre, Apellido, FechaNacimiento, Direccion, Email, IDCarrera) VALUES
('Luis', 'Hernández', '2000-05-10', 'Av. Siempre Viva 742', 'lhernandez@correo.com', 1),
('María', 'Pérez', '2001-08-15', 'Calle Luna 23', 'mperez@correo.com', 2),
('Sofía', 'Martínez', '2002-03-22', 'Calle Sol 19', 'smartinez@correo.com', 3),
('Jorge', 'Luna', '2000-01-20', 'Calle Río Bravo 10', 'jluna@correo.com', 4),
('Carmen', 'Salas', '2001-04-11', 'Av. Reforma 80', 'csalas@correo.com', 5),
('Marco', 'Delgado', '1999-12-05', 'Calle Norte 45', 'mdelgado@correo.com', 6),
('Andrea', 'Mendoza', '2002-07-19', 'Col. Centro 101', 'amendoza@correo.com', 7),
('Luis', 'Reyes', '2001-09-09', 'Av. Patria 77', 'lreyes@correo.com', 8),
('Rosa', 'Vargas', '2000-10-12', 'Calle Roble 23', 'rvargas@correo.com', 9),
('Gabriel', 'Moreno', '2002-03-18', 'Calle Cedros 56', 'gmoreno@correo.com', 10),
('Isabel', 'Cruz', '2000-06-25', 'Av. Chapultepec 34', 'icruz@correo.com', 1),
('Daniel', 'Suárez', '1999-08-14', 'Calle Norte 999', 'dsuarez@correo.com', 2),
('Natalia', 'Paredes', '2002-11-30', 'Zona Sur 23', 'nparedes@correo.com', 3);
```

idestudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fechanacimiento date	direccion character varying (50)	email character varying (50)	idcarrera integer
1	Luis	Hernández	2000-05-10	Av. Siempre Viva 742	lhernandez@correo.com	1
2	María	Pérez	2001-08-15	Calle Luna 23	mperez@correo.com	2
3	Sofía	Martínez	2002-03-22	Calle Sol 19	smartinez@correo.com	3
4	Jorge	Luna	2000-01-20	Calle Río Bravo 10	jluna@correo.com	4
5	Carmen	Salas	2001-04-11	Av. Reforma 80	csalas@correo.com	5
6	Marco	Delgado	1999-12-05	Calle Norte 45	mdelgado@correo.com	6
7	Andrea	Mendoza	2002-07-19	Col. Centro 101	amendoza@correo.com	7
8	Luis	Reyes	2001-09-09	Av. Patria 77	lreyes@correo.com	8
9	Rosa	Vargas	2000-10-12	Calle Roble 23	rvargas@correo.com	9
10	Gabriel	Moreno	2002-03-18	Calle Cedros 56	gmoreno@correo.com	10
11	Isabel	Cruz	2000-06-25	Av. Chapultepec 34	icruz@correo.com	1
12	Daniel	Suárez	1999-08-14	Calle Norte 999	dsuarez@correo.com	2
13	Natalia	Paredes	2002-11-30	Zona Sur 23	nparedes@correo.com	3

Cursos:

```
INSERT INTO Cursos (IDDepartamento, NombreCurso, Descripcion, Creditos, Semestre, IDCampus) VALUES
(1, 'Estructura de Datos', 'Curso sobre estructuras fundamentales', 6, 3, 1),
(2, 'Psicología General', 'Introducción a la Psicología', 5, 1, 2),
(3, 'Contabilidad I', 'Conceptos básicos de contabilidad', 4, 1, 1),
(1, 'Álgebra Lineal', 'Estudio de matrices y vectores', 6, 1, 1),
(2, 'Óptica', 'Fundamentos de la luz y visión', 5, 2, 2),
(3, 'Química Orgánica', 'Composición de compuestos orgánicos', 6, 2, 3),
(4, 'Derecho Penal', 'Introducción al derecho penal', 4, 3, 4),
(5, 'Dibujo Artístico', 'Técnicas básicas de dibujo', 3, 1, 5),
(6, 'México Colonial', 'Historia de México virreinal', 4, 4, 6),
(7, 'Inglés Avanzado', 'Perfeccionamiento del idioma inglés', 4, 5, 7),
(8, 'Macroeconomía', 'Estudio de la economía global', 5, 3, 8),
(9, 'Lógica Formal', 'Introducción a la lógica y razonamiento', 4, 1, 9),
(10, 'Didáctica General', 'Métodos de enseñanza', 5, 2, 10);
```

idcurso [PK] integer	iddepartamento integer	nombrecurso character varying (30)	descripcion character varying (75)	creditos integer	semestre integer	idcampus integer
1	1	Estructura de Datos	Curso sobre estructuras fundamental...	6	3	1
2	2	Psicología General	Introducción a la Psicología	5	1	2
3	3	Contabilidad I	Conceptos básicos de contabilidad	4	1	1
4	1	Álgebra Lineal	Estudio de matrices y vectores	6	1	1
5	2	Óptica	Fundamentos de la luz y visión	5	2	2
6	3	Química Orgánica	Composición de compuestos orgánic...	6	2	3
7	4	Derecho Penal	Introducción al derecho penal	4	3	4
8	5	Dibujo Artístico	Técnicas básicas de dibujo	3	1	5
9	6	México Colonial	Historia de México virreinal	4	4	6
10	7	Inglés Avanzado	Perfeccionamiento del idioma inglés	4	5	7
11	8	Macroeconomía	Estudio de la economía global	5	3	8
12	9	Lógica Formal	Introducción a la lógica y razonamiento	4	1	9
13	10	Didáctica General	Métodos de enseñanza	5	2	10

Horarios:

```
INSERT INTO Horarios (IDCurso, IDAula, FechaInicio, FechaFin, HoraInicio, HoraFin) VALUES
(1, 1, '2025-08-01', '2025-12-15', '08:00', '10:00'),
(2, 2, '2025-08-01', '2025-12-15', '10:15', '12:15'),
(3, 1, '2025-08-01', '2025-12-15', '08:00', '09:30'),
(4, 2, '2025-08-01', '2025-12-15', '10:00', '11:30'),
(5, 3, '2025-08-01', '2025-12-15', '12:00', '13:30'),
(6, 4, '2025-08-01', '2025-12-15', '14:00', '15:30'),
(7, 5, '2025-08-01', '2025-12-15', '16:00', '17:30'),
(8, 6, '2025-08-01', '2025-12-15', '09:00', '10:30'),
(9, 7, '2025-08-01', '2025-12-15', '11:00', '12:30'),
(10, 8, '2025-08-01', '2025-12-15', '13:00', '14:30'),
(11, 9, '2025-08-01', '2025-12-15', '15:00', '16:30'),
(12, 10, '2025-08-01', '2025-12-15', '17:00', '18:30');
```

idhorario [PK] integer	idcurso integer	idaula 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	10:15:00	12:15:00
23	3	1	2025-08-01	2025-12-15	08:00:00	09:30:00
24	4	2	2025-08-01	2025-12-15	10:00:00	11:30:00
25	5	3	2025-08-01	2025-12-15	12:00:00	13:30:00
26	6	4	2025-08-01	2025-12-15	14:00:00	15:30:00
27	7	5	2025-08-01	2025-12-15	16:00:00	17:30:00
28	8	6	2025-08-01	2025-12-15	09:00:00	10:30:00
29	9	7	2025-08-01	2025-12-15	11:00:00	12:30:00
30	10	8	2025-08-01	2025-12-15	13:00:00	14:30:00
31	11	9	2025-08-01	2025-12-15	15:00:00	16:30:00
32	12	10	2025-08-01	2025-12-15	17:00:00	18:30:00

CursosProfesores:

```
INSERT INTO CursosProfesores (IDCurso, IDProfesor) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10),
(11, 11),
(12, 12),
(13, 13);
```

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
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13

ProgramasCursos:

```
INSERT INTO ProgramasCursos (IDPrograma, IDCurso) VALUES
(1, 1),
(2, 2),
(1, 3),
(2, 4),
(3, 5),
(4, 6),
(5, 7),
(6, 8),
(7, 9),
(8, 10),
(9, 11),
(10, 12),
(11, 13);
```

idprogramacurso [PK] integer	idprograma integer	idcurso integer
1	1	1
2	2	2
3	1	3
4	2	4
5	3	5
6	4	6
7	5	7
8	6	8
9	7	9
10	8	10
11	9	11
12	10	12
13	11	13

Inscripciones

```
INSERT INTO Inscripciones (IDEstudiante, IDCurso, FechaInscripcion, Calificacion) VALUES
(1, 1, '2025-08-05', 90),
(2, 2, '2025-08-05', 85),
(3, 3, '2025-08-05', 88),
(4, 4, '2025-08-15', 91),
(5, 5, '2025-08-15', 86),
(6, 6, '2025-08-15', 89),
(7, 7, '2025-08-15', 84),
(8, 8, '2025-08-15', 90),
(9, 9, '2025-08-15', 87),
(10, 10, '2025-08-15', 93),
(11, 11, '2025-08-15', 85),
(12, 12, '2025-08-15', 88),
(13, 1, '2025-08-15', 92);
```

idinscripcion [PK] integer	idestudiante integer	idcurso integer	fechainscripcion date	calificacion integer
1	1	1	2025-08-05	90
2	2	2	2025-08-05	85
3	3	3	2025-08-05	88
4	4	4	2025-08-15	91
5	5	5	2025-08-15	86
6	6	6	2025-08-15	89
7	7	7	2025-08-15	84
8	8	8	2025-08-15	90
9	9	9	2025-08-15	87
10	10	10	2025-08-15	93
11	11	11	2025-08-15	85
12	12	12	2025-08-15	88
13	13	1	2025-08-15	92

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

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

8. 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:

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

```
347 SELECT Nombre, Apellido
348 FROM Estudiantes;
349
```

	nombre character varying (50)	apellido character varying (50)
1	Luis	Hernández
2	María	Pérez
3	Sofía	Martínez
4	Jorge	Luna
5	Carmen	Salas
6	Marco	Delgado
7	Andrea	Mendoza
8	Luis	Reyes
9	Rosa	Vargas
10	Gabriel	Moreno
11	Isabel	Cruz
12	Daniel	Suárez
13	Natalia	Paredes

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

```
SELECT *
FROM Cursos
WHERE Creditos = 3;
```

idcurso PK integer	iddepartamento integer	nombrecurso character varying (30)	descripcion character varying (75)	creditos integer	semestre integer	idcampus integer
8	5	Dibujo Artístico	Técnicas básicas de dibujo	3	1	5

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

```
SELECT
    e.Nombre AS NombreEstudiante,
    e.Apellido AS ApellidoEstudiante,
    c.NombreCurso
FROM
    Inscripciones i
INNER JOIN Estudiantes e ON i.IDEstudiante = e.IDEstudiante
INNER JOIN Cursos c ON i.IDCurso = c.IDCurso;
```

Output Mensajes Notificaciones

nombrestudiante character varying (50)	apellidostudiante character varying (50)	nombrecurso character varying (30)
Luis	Hernández	Estructura de Datos
María	Pérez	Psicología General
Sofía	Martínez	Contabilidad I
Jorge	Luna	Álgebra Lineal
Carmen	Salas	Óptica
Marco	Delgado	Química Orgánica
Andrea	Mendoza	Derecho Penal
Luis	Reyes	Dibujo Artístico
Rosa	Vargas	México Colonial
Gabriel	Moreno	Inglés Avanzado
Isabel	Cruz	Macroeconomía
Daniel	Suárez	Lógica Formal
Natalia	Paredes	Estructura de Datos

en el que está inscrito

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
FROM Estudiantes e
LEFT JOIN Inscripciones i ON e.IDEstudiante = i.IDEstudiante
LEFT JOIN Cursos c ON i.IDCurso = c.IDCurso;
```

nombreestudiante character varying (50) 🔒	apellidoestudiante character varying (50) 🔒	nombrecurso character varying (30) 🔒
Luis	Hernández	Estructura de Datos
María	Pérez	Psicología General
Sofía	Martínez	Contabilidad I
Jorge	Luna	Álgebra Lineal
Carmen	Salas	Óptica
Marco	Delgado	Química Orgánica
Andrea	Mendoza	Derecho Penal
Luis	Reyes	Dibujo Artístico
Rosa	Vargas	México Colonial
Gabriel	Moreno	Inglés Avanzado
Isabel	Cruz	Macroeconomía
Daniel	Suárez	Lógica Formal
Natalia	Paredes	Estructura de Datos

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 Estudiantes e
RIGHT JOIN Inscripciones i ON e.IDEstudiante = i.IDEstudiante
RIGHT JOIN Cursos c ON i.IDCurso = c.IDCurso;
```

nombrecurso character varying (30) 🔒	nombreestudiante character varying (50) 🔒	apellidoestudiante character varying (50) 🔒
Estructura de Datos	Luis	Hernández
Psicología General	María	Pérez
Contabilidad I	Sofía	Martínez
Álgebra Lineal	Jorge	Luna
Óptica	Carmen	Salas
Química Orgánica	Marco	Delgado
Derecho Penal	Andrea	Mendoza
Dibujo Artístico	Luis	Reyes
México Colonial	Rosa	Vargas
Inglés Avanzado	Gabriel	Moreno
Macroeconomía	Isabel	Cruz
Lógica Formal	Daniel	Suárez
Estructura de Datos	Natalia	Paredes
Didáctica General	[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;
```

nombrecurso character varying (30)	cantidadestudiantes bigint
Estructura de Datos	2
Derecho Penal	1
Inglés Avanzado	1
Álgebra Lineal	1
Psicología General	1
Contabilidad I	1
Macroeconomía	1
México Colonial	1
Dibujo Artístico	1
Lógica Formal	1
Óptica	1
Didáctica General	0
Química Orgánica	1

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

```
SELECT Nombre, Apellido, FechaNacimiento
FROM Estudiantes
WHERE FechaNacimiento BETWEEN '1995-01-01' AND '1998-12-31';
```

En mi BD no hay ningún estudiante de edad tan avanzada.

fechanacimiento date
2000-05-10
2001-08-15
2002-03-22
2000-01-20
2001-04-11
1999-12-05
2002-07-19
2001-09-09
2000-10-12
2002-03-18
2000-06-25
1999-08-14
2002-11-30

Pero sí serviría

```
SELECT Nombre, Apellido, FechaNacimiento
FROM Estudiantes
WHERE FechaNacimiento BETWEEN '2000-01-01' AND '2001-12-31';
```

nombre character varying (50)	apellido character varying (50)	fechanacimiento date
Luis	Hernández	2000-05-10
María	Pérez	2001-08-15
Jorge	Luna	2000-01-20
Carmen	Salas	2001-04-11
Luis	Reyes	2001-09-09
Rosa	Vargas	2000-10-12
Isabel	Cruz	2000-06-25

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

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

idcurso [PK] integer	iddepartamento integer	nombrecurso character varying (30)	descripcion character varying (75)	creditos integer	semestre integer	idcampus integer
4	1	Álgebra Lineal	Estudio de matrices y vectores	6	1	1
3	3	Contabilidad I	Conceptos básicos de contabilidad	4	1	1
7	4	Derecho Penal	Introducción al derecho penal	4	3	4
8	5	Dibujo Artístico	Técnicas básicas de dibujo	3	1	5
13	10	Didáctica General	Métodos de enseñanza	5	2	10
1	1	Estructura de Datos	Curso sobre estructuras fundamental...	6	3	1
10	7	Inglés Avanzado	Perfeccionamiento del idioma inglés	4	5	7
12	9	Lógica Formal	Introducción a la lógica y razonamiento	4	1	9
11	8	Macroeconomía	Estudio de la economía global	5	3	8
9	6	México Colonial	Historia de México virreinal	4	4	6
5	2	Óptica	Fundamentos de la luz y visión	5	2	2
2	2	Psicología General	Introducción a la Psicología	5	1	2
6	3	Química Orgánica	Composición de compuestos orgánic...	6	2	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
    Estudiantes.IDEstudiante,
    Estudiantes.Nombre,
    Estudiantes.Apellido,
    COUNT(Inscripciones.IDCurso) AS TotalInscripciones
  FROM Estudiantes
  LEFT JOIN Inscripciones ON Estudiantes.IDEstudiante = Inscripciones.IDEstudiante
  GROUP BY Estudiantes.IDEstudiante, Estudiantes.Nombre, Estudiantes.Apellido
)
SELECT
  Nombre,
  Apellido,
  TotalInscripciones
FROM InscripcionesPorEstudiante
ORDER BY TotalInscripciones DESC
LIMIT 3;

```

nombre character varying (50) 🔒	apellido character varying (50) 🔒	totalinscripciones bigint 🔒
Rosa	Vargas	1
Sofía	Martínez	1
Isabel	Cruz	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 ConteoInscripciones AS (
    SELECT
        d.NombreDepartamento,
        c.IDCurso,
        c.NombreCurso,
        COUNT(i.IDEstudiante) AS TotalInscritos
    FROM Cursos c
    JOIN Departamentos d ON c.IDDepartamento = d.IDDepartamento
    LEFT JOIN Incripciones i ON c.IDCurso = i.IDCurso
    GROUP BY d.NombreDepartamento, c.IDCurso, c.NombreCurso
),
MaximosPorDepartamento AS (
    SELECT
        NombreDepartamento,
        MAX(TotalInscritos) AS MaxInscritos
    FROM ConteoInscripciones
    GROUP BY NombreDepartamento
)
SELECT
    ci.NombreDepartamento,
    ci.NombreCurso,
    ci.TotalInscritos
FROM ConteoInscripciones ci
JOIN MaximosPorDepartamento mpd
    ON ci.NombreDepartamento = mpd.NombreDepartamento
    AND ci.TotalInscritos = mpd.MaxInscritos;

```

nombredepartamento character varying (30) 🔒	nombrecurso character varying (30) 🔒	totalinscritos bigint 🔒
Química	Dibujo Artístico	1
Biología	Inglés Avanzado	1
Administración	Química Orgánica	1
Historia	Didáctica General	0
Ingeniería	Estructura de Datos	2
Matemáticas	Derecho Penal	1
Ciencias Sociales	Óptica	1
Física	México Colonial	1
Administración	Contabilidad I	1
Ciencias Sociales	Psicología General	1
Artes	Lógica Formal	1
Derecho	Macroeconomía	1

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.


```
SELECT p.Nombre_Profesor,
       p.Apellido_Profesor,
       COUNT(cp.IDCurso) AS Cantidad_Cursos
FROM Profesores p
JOIN CursosProfesores cp ON p.IDProfesor = cp.IDProfesor
GROUP BY p.IDProfesor, p.Nombre_Profesor, p.Apellido_Profesor
HAVING COUNT(cp.IDCurso) > 2;
```

nombre_profesor	apellido_profesor	cantidad_cursos
character varying (25)	character varying (30)	bigint

Ninguno de los profesores de mi Base de Datos tiene más de un curso

```
SELECT p.Nombre_Profesor,
       p.Apellido_Profesor,
       COUNT(cp.IDCurso) AS Cantidad_Cursos
FROM Profesores p
JOIN CursosProfesores cp ON p.IDProfesor = cp.IDProfesor
GROUP BY p.IDProfesor, p.Nombre_Profesor, p.Apellido_Profesor
HAVING COUNT(cp.IDCurso) > 0;
```

nombre_profesor	apellido_profesor	cantidad_cursos
character varying (25)	character varying (30)	bigint
Tomás	Reyes	1
Sergio	Cano	1
Jorge	Gómez	1
Pablo	Mendoza	1
Laura	Ruiz	1
Beatriz	Nava	1
Diana	Silva	1
Javier	Morales	1
Ana	López	1
Raúl	Fernández	1
Patricia	Torres	1
Carlos	Ramírez	1
Claudia	Herrera	1

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 PromediosCursos AS (
  SELECT  c.IDCurso,
          c.NombreCurso,
          AVG(i.Calificacion) AS PromedioCalificacion
  FROM    Cursos c
  JOIN    Inscripciones i ON c.IDCurso = i.IDCurso
  GROUP BY c.IDCurso, c.NombreCurso
),
CursosConPrograma AS (
  SELECT  p.IDPrograma,
          p.NombrePrograma,
          pc.IDCurso,
          c.NombreCurso,
          c.PromedioCalificacion
  FROM    ProgramasEstudio p
  JOIN    ProgramasCursos pc ON p.IDPrograma = pc.IDPrograma
  JOIN    PromediosCursos c ON pc.IDCurso = c.IDCurso
),
MaxPromediosPorPrograma AS (
  SELECT  IDPrograma,
          MAX(PromedioCalificacion) AS MaxPromedio
  FROM    CursosConPrograma
  GROUP BY IDPrograma
)
SELECT  cp.NombrePrograma AS "Programa de Estudio",
        cp.NombreCurso AS "Curso con mejor promedio",
        cp.PromedioCalificacion AS "Promedio de calificación"
FROM    CursosConPrograma cp
JOIN    MaxPromediosPorPrograma m ON cp.IDPrograma = m.IDPrograma
        AND cp.PromedioCalificacion = m.MaxPromedio
ORDER BY cp.NombrePrograma;

```

Programa de Estudio 	Curso con mejor promedio 	Promedio de calificación 
Programa de Artes	México Colonial	87.0000000000000000
Programa de Biología	Derecho Penal	84.0000000000000000
Programa de Derecho	Dibujo Artístico	90.0000000000000000
Programa de Economía	Lógica Formal	88.0000000000000000
Programa de Física	Química Orgánica	89.0000000000000000
Programa de Historia	Inglés Avanzado	93.0000000000000000
Programa de Ingeniería	Estructura de Datos	91.0000000000000000
Programa de Lenguas	Macroeconomía	85.0000000000000000
Programa de Psicología	Álgebra Lineal	91.0000000000000000
Programa de Química	Óptica	86.0000000000000000