



Instituto Tecnológico de Mexicali

Fundamentos de base de datos

Luis Alonso Guevara Quiñonez - 23490377

-Docente-

Jose Ramon Bogarin Valenzuela

Examen Final de Unidad 4

Fecha limite entrega **23/05/2025 23:59**

Ingeniería en sistemas.

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

```
1 ✓ CREATE TABLE Estudiantes (
2     IDEstudiante SERIAL PRIMARY KEY,
3     Nombre VARCHAR(50) NOT NULL,
4     Apellido VARCHAR(50) NOT NULL,
5     FechaNacimiento DATE NOT NULL,
6     Direccion TEXT NOT NULL,
7     Ciudad VARCHAR(100) NOT NULL,
8     Email VARCHAR(100) UNIQUE NOT NULL
9 );
10
```

Se crea inicialmente la tabla usando CREATE TABLE para después ingresa los datos solicitados y así consecutivamente con las siguientes.

Tabla: Cursos

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

```
11 ✓ CREATE TABLE Cursos (
12     IDCurso SERIAL PRIMARY KEY,
13     NombreCurso VARCHAR(100) NOT NULL,
14     Descripcion TEXT,
15     Creditos INT CHECK (Creditos > 0),
16     Semestre INT NOT NULL CHECK (Semestre > 0),
17     IDDepartamento INT REFERENCES Departamentos(IDDepartamento)
18 );
```

Tabla: Inscripciones

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

```
20 ✓ CREATE TABLE Inscripciones (
21     IDInscripcion SERIAL PRIMARY KEY,
22     IDEstudiante INT NOT NULL REFERENCES Estudiantes(IDEstudiante),
23     IDCurso INT NOT NULL REFERENCES Cursos(IDCurso),
24     FechaInscripcion DATE NOT NULL,
25     Calificacion INT NOT NULL CHECK (Calificacion >= 0 AND Calificacion <= 100)
26 );
```

Tabla: Profesores

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

```
28 ✓ CREATE TABLE Profesores (
29     IDProfesor SERIAL PRIMARY KEY,
30     Nombre VARCHAR(50) NOT NULL,
31     Apellido VARCHAR(50) NOT NULL,
32     Titulo VARCHAR(100),
33     IDDepartamento INT REFERENCES Departamentos(IDDepartamento)
34 );
```

Tabla: Departamentos

- IDDepartamento (Clave Principal)
- NombreDepartamento
- Edificio

```
36 ✓ CREATE TABLE Departamentos (
37     IDDepartamento SERIAL PRIMARY KEY,
38     NombreDepartamento VARCHAR(70) NOT NULL,
39     Edificio VARCHAR(50)
40 );
```

- Tabla: Aulas
- IDAula (Clave Principal)
- NombreAula
- Capacidad
- Ubicacion

```
42 ✓ CREATE TABLE Aulas (
43     IDAula SERIAL PRIMARY KEY,
44     NombreAula VARCHAR(50) NOT NULL,
45     Capacidad INT NOT NULL CHECK (Capacidad > 0),
46     Ubicacion VARCHAR(100) NOT NULL
47 );
```

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

```
49 ✓ CREATE TABLE Horarios (
50     IDHorario SERIAL PRIMARY KEY,
51     IDCurso INT NOT NULL REFERENCES Cursos(IDCurso),
52     IDAula INT NOT NULL REFERENCES Aulas(IDAula),
53     FechaInicio DATE NOT NULL,
54     FechaFin DATE NOT NULL,
55     HoraInicio TIME NOT NULL,
56     HoraFin TIME NOT NULL
57 );
```

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)

```
59 v CREATE TABLE CursosProfesores (
60     IDCursoProfesor SERIAL PRIMARY KEY,
61     IDCurso INT NOT NULL REFERENCES Cursos(IDCurso),
62     IDProfesor INT NOT NULL REFERENCES Profesores(IDProfesor)
63 );
```

- Tabla: ProgramasEstudio

- IDPrograma (Clave Principal)
- NombrePrograma
- DescripcionPrograma

```
65 v CREATE TABLE ProgramasEstudio (
66     IDPrograma SERIAL PRIMARY KEY,
67     NombrePrograma VARCHAR(100) NOT NULL,
68     DescripcionPrograma TEXT
69 );
```

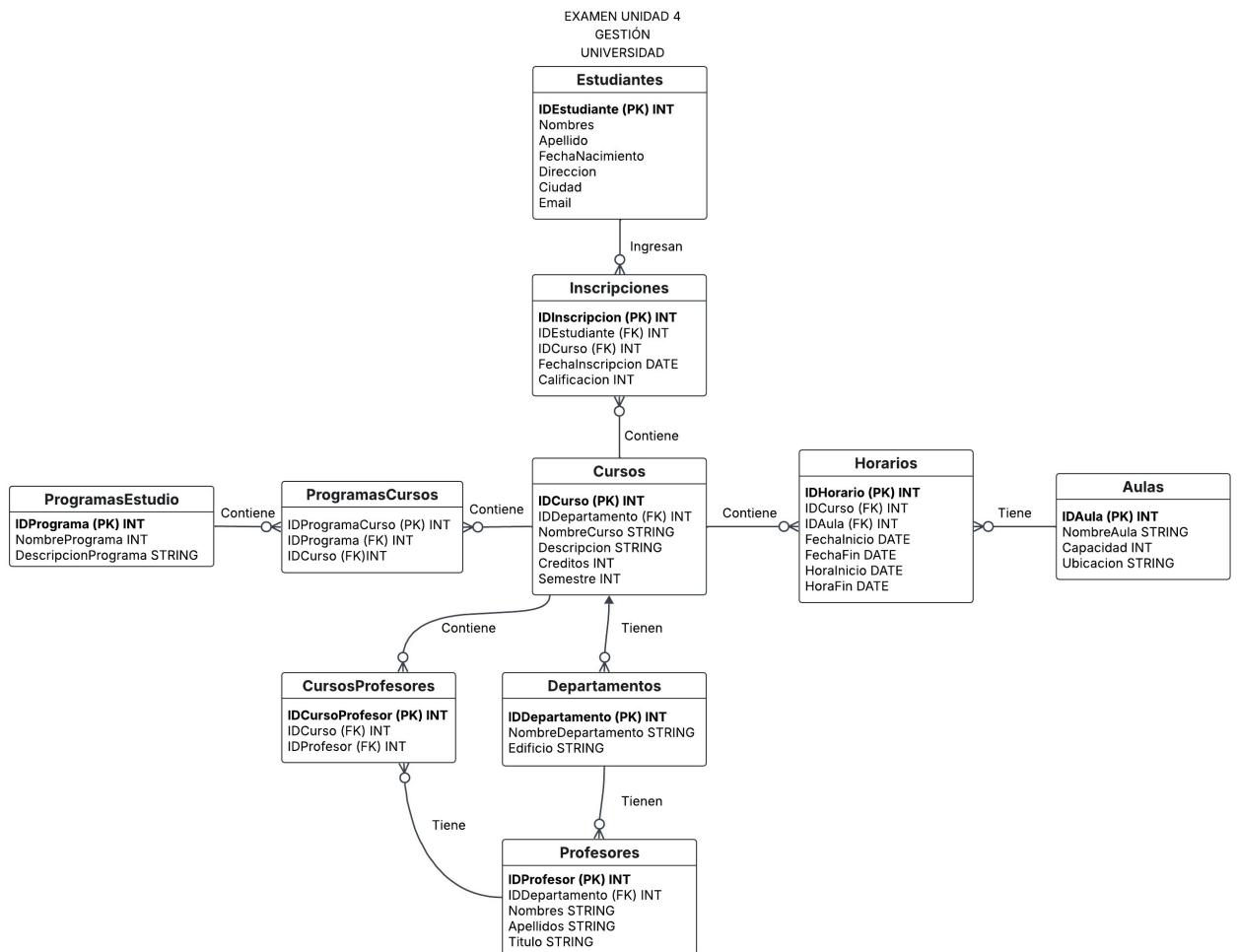
Tabla Intermedia: ProgramasCursos (Relación Muchos a Muchos entre ProgramasEstudio y Cursos)

- IDProgramaCurso (Clave Principal)
- IDPrograma (Clave Foránea, referencia a la tabla ProgramasEstudio)
- IDCurso (Clave Foránea, referencia a la tabla Cursos)

```
71 v CREATE TABLE ProgramasCursos (
72     IDProgramaCurso SERIAL PRIMARY KEY,
73     IDPrograma INT NOT NULL REFERENCES ProgramasEstudio(IDPrograma),
74     IDCurso INT NOT NULL REFERENCES Cursos(IDCurso)
75 );
```

- Entendiendo las Entidades:** Identificar los elementos principales que necesita gestionar la universidad y la información relevante para cada uno.
- Diseñando las Tablas (MDL):** Organizar la información en tablas, definiendo campos, tipos de datos y claves principales, y establecer las relaciones entre las tablas.
- Creando las Tablas (DDL):** Utilizar un sistema de gestión de bases de datos para crear las tablas definidas en el paso anterior.
- Modificando las Tablas (DDL):** Realizar modificaciones a la estructura de las tablas, como agregar, modificar o eliminar campos, según sea necesario.

Entendiendo las entidades y diseñando las tablas



Agregar Tablas:

Tabla: Campus

- IDCampus (Clave Principal)
- NombreCampus
- DireccionCampus

```
77 v CREATE TABLE Campus (
78     IDCampus SERIAL PRIMARY KEY,
79     NombreCampus VARCHAR(100) NOT NULL,
80     DireccionCampus TEXT NOT NULL
81 );
```

Tenemos IDCampus como clave primaria, nombreCampus como caracter (max 100) y no puede ser nulo.

Tabla: Carreras

- IDCarrera (Clave Principal)
- NombreCarrera
- TituloOtorgado

```
83 v CREATE TABLE Carreras (
84     IDCarrera SERIAL PRIMARY KEY,
85     NombreCarrera VARCHAR(100) NOT NULL,
86     TituloOtorgado VARCHAR(100) NOT NULL
87 );
```

Clave primaria IDCarrera entero autoincrementado, nombreCarrera como caracter (max 100) y no puede ser nulo.

Agregar una relación de muchos a muchos entre Estudiantes y Carreras

```
89 v CREATE TABLE EstudiantesCarreras (
90     IDEstudianteCarrera SERIAL PRIMARY KEY,
91     IDEstudiante INT NOT NULL REFERENCES Estudiantes(IDEstudiante),
92     IDCarrera INT NOT NULL REFERENCES Carreras(IDCarrera)
93 );
```

Clave primaria IDEstudianteCarrera entero autoincrementado y se hace referencia a las tablas Estudiantes y Carreras(FK).

Modificar Tablas:

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

```
98 v ALTER TABLE Estudiantes
99   ADD COLUMN IDCarrera INT,
100  ADD CONSTRAINT FkEstudiantesCarrera FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera);
101 ;
```

Alteramos la tabla estudiantes para despues añadir una columna nueva llamada IDCarrera por que será nuestra clave foranea y hara referencia a carrera) y eso se hara con ADD CONSTRAINT colocandole de nombre fkEstudiantesCarrera para poder identificarla

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

```
103 v ALTER TABLE Cursos
104   ADD COLUMN IDCampus INT,
105  ADD CONSTRAINT FkCursosCampus FOREIGN KEY (IDCampus) REFERENCES Campus(IDCampus);
```

- Modificar la tabla Profesores para incluir un campo Email

```
108 v ALTER TABLE Profesores
109   ADD COLUMN Email VARCHAR(150) UNIQUE NOT NULL;
110 ;
```

Alteramos la tabla profesores y añadimos la nueva columna Email de tipo unico y no puede ser nulo.

- Eliminar Tablas/Campos

- Eliminar la columna Ciudad de la tabla Estudiantes.

```
112 v ALTER TABLE Estudiantes
113   DROP COLUMN Ciudad;
```

- Eliminar la tabla Aulas

```
116   DROP TABLE Horarios;
117   DROP TABLE Aulas;
```

Elimine la tabla horario debido a que aulas dependia de ella para poder ser eliminada, e igual no la veia tan necesaria.

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

```

122  INSERT INTO Estudiantes (Nombre, Apellido, FechaNacimiento, Direccion, Email, IDCarrera) VALUES
123  ('Damian', 'Lugo', '2005-07-26', 'Calle 1', 'damianlugo1@gmail.com',1),
124  ('Luis', 'Alonso', '2004-01-02', 'Calle 2', 'luisalonso@gmail.com',2),
125  ('Donovan', 'Hernandez', '1998-06-20', 'Calle 3', 'donovanhernandez@gmail.com',3),
126  ('Emanuel', 'Padilla', '2003-09-08', 'Calle 4', 'emanuelpadilla@gmail.com',4),
127  ('Clara', 'Martinez', '2005-12-05', 'Calle 5', 'claramartinez@gmail.com',5),
128  ('Erick', 'Hidekio', '2005-11-02', 'Calle 6', 'erickhidekio@gmail.com',5);
129
130
131  ✓ INSERT INTO Cursos (NombreCurso, Descripcion, Creditos, Semestre, IDDepartamento) VALUES
132  ('Bases de Datos', 'Diseño y uso de bases de datos', 3, 4, 1),
133  ('Calculo 1', 'Introducción al calculo', 4, 1, 2),
134  ('Psicología General', 'Conceptos básicos de la psicología', 3, 2, 3),
135  ('Administración 1', 'Fundamentos de administración', 3, 1, 4),
136  ('Diseño Digital', 'Herramientas de diseño gráfico digital', 4, 3, 5);
137

```

Agregamos datos para la tabla estudiantes y cursos, pero antes de añadir cursos primero tenemos que insertar la tabla departamento para que nos permita insertar debido a la id.

```

139  ✓ INSERT INTO Inscripciones (IDEstudiante, IDCurso, FechaInscripcion, Calificacion) VALUES
140  (13, 6, '2025-05-19', 90),
141  (14, 7, '2025-05-20', 85),
142  (15, 8, '2025-05-21', 92),
143  (16, 9, '2025-05-22', 88),
144  (17, 10, '2025-05-23', 95);
145
146
147  ✓ INSERT INTO Profesores (Nombre, Apellido, Titulo, IDDepartamento, Email) VALUES
148  ('Carlos', 'Ramirez', 'Ing. en Ciencias Computacionales', 1,'carlosramirez@gmail.com'),
149  ('Elena', 'Torres', 'Mtra. en Matematicas', 2,'elenatorres@gmail.com'),
150  ('Jorge', 'Hernandez', 'Dr. en Psicología', 3,'jorgehernandez@gmail.com'),
151  ('Marta', 'Rios', 'Mtra. en Administracion', 4,'martarios@gmail.com'),
152  ('Tomas', 'Perez', 'Lic. en Diseño Grafico', 5,'tomasperez@gmail.com');
153
154
155  ✓ INSERT INTO Departamentos (NombreDepartamento, Edificio) VALUES
156  ('Ingenieria de Sistemas', 'Edificio A'),
157  ('Ciencias Basicas', 'Edificio B'),
158  ('Humanidades', 'Edificio C'),
159  ('Administración', 'Edificio D'),
160  ('Diseño Gráfico', 'Edificio E');
161
162
163  ✓ INSERT INTO CursosProfesores (IDCurso, IDProfesor) VALUES
164  (6, 1),
165  (7, 2),
166  (8, 3),
167  (9, 4),
168  (10, 5);
169
170
171  ✓ INSERT INTO ProgramasEstudio (NombrePrograma, DescripcionPrograma) VALUES
172  ('Programa de Sistemas', 'Programa enfocado en ingenieria y tecnologias de la informacion'),
173  ('Programa de Ciencias', 'Programa que integra diversas ciencias basicas.'),
174  ('Programa de Psicología', 'Formacion en el campo psicologico.'),
175  ('Programa de Administración', 'Carrera profesional en gestion y administracion.'),
176  ('Programa de Diseño', 'Carrera en diseño visual y digital.');
177
178  ✓ INSERT INTO ProgramasCursos (IDPrograma, IDCurso) VALUES
179  (1, 6),
180  (2, 7),
181  (3, 8),
182  (4, 9),
183  (5, 10);

```

Algunas id estan por encima del 1 no supe la razón del por que no empezaban del 1 hacia arriba pero igual lo adapte a los insert.

```
187 ✓ INSERT INTO Carreras (NombreCarrera, TituloOtorgado) VALUES
188 ('Ingeniería en Sistemas', 'Ingeniero en Sistemas'),
189 ('Matemáticas Aplicadas', 'Licenciado en Matemáticas'),
190 ('Psicología', 'Licenciado en Psicología'),
191 ('Administración de Empresas', 'Licenciado en Administración'),
192 ('Diseño Gráfico', 'Licenciado en Diseño');
193
194 ✓ INSERT INTO EstudiantesCarreras (IDEstudiante, IDCarrera) VALUES
195 (13, 1),
196 (14, 2),
197 (15, 1),
198 (16, 3),
199 (17, 4);
```

Ya por ultimo agregamos esos datos.

2. **Actualizando Datos:** Actualizar la información existente en las tablas para reflejar cambios o correcciones.
3. **Eliminando Datos:** Eliminar registros de las tablas que ya no sean relevantes.

```
201 SELECT * FROM Inscripciones;
202 ✓ UPDATE Estudiantes
203 SET Email = 'damianlugo@gmail.com'
204 WHERE IDEstudiante = 1;
205
206 ✓ UPDATE Cursos
207 SET NombreCurso = 'Calculo diferencial'
208 WHERE NombreCurso = 'Calculo 1';
209
210 ✓ DELETE FROM Estudiantes
211 WHERE IDEstudiante = 6;
```

Al ingresar datos en la tabla estudiantes el email de damian lugo estaba como damianlugo1@gmail.com y fue cambiado a damianlugo@gmail.com, actualice tambien el nombre del curso a calculo diferencial y elimine al estudiante 6, que realmente no hacia tanta falta debido a que en casi todas las tablas solo inserte 5 datos.

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

```
3 --Obtener nombres completos de los estudiantes inscritos en el curso "Bases de Datos"
4 SELECT e.Nombre, e.Apellido
5 FROM Estudiantes e
6 INNER JOIN Inscripciones i ON e.IDEstudiante = i.IDEstudiante
7 INNER JOIN Cursos c ON i.IDCurso = c.IDCurso
8 WHERE c.NombreCurso = 'Bases de Datos';
9
10 -- Mostrar los cursos y el nombre del departamento al que pertenecen
11 SELECT c.NombreCurso, d.NombreDepartamento
12 FROM Cursos c
13 INNER JOIN Departamentos d ON c>IDDepartamento = d.IDDepartamento;
```

Agregue esos 2 querys solicitando lo que esta en verde, donde el primero une la tabla de estudiantes con inscripciones y cursos para poder mostrar al estudiante que esta inscrito en el curso Base de Datos.

Y con el otro, uni la tabla cursos con departamento para poder mostrar a que departamento pertenecen los cursos.

	nombre character varying (50) 	apellido character varying (50) 
1	Damian	Lugo

Esto muestra el primer query.

	nombrcurso character varying (100) 	nombreddepartamento character varying (70) 
1	Bases de Datos	Ingenieria de Sistemas
2	Psicología General	Humanidades
3	Administración 1	Administración
4	Diseño Digital	Diseño Gráfico
5	Calculo diferencial	Ciencias Basicas

Y esto muestra el segundo query.

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

```
18 ✓ SELECT Nombre, Apellido  
19   FROM Estudiantes;
```

	nombre character varying (50) 	apellido character varying (50) 
1	Damian	Lugo
2	Luis	Alonso
3	Donovan	Hernandez
4	Emanuel	Padilla
5	Clara	Martinez
6	Erick	Hidekio

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

```
21 ✓ SELECT NombreCurso  
22   FROM Cursos  
23   WHERE Creditos = 3;
```

	nombrcurso character varying (100) 
1	Bases de Datos
2	Psicología General
3	Administración 1

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

```
26 ✓ SELECT e.Nombre, c.NombreCurso  
27   FROM Inscripciones i  
28   INNER JOIN Estudiantes e ON i.IDEstudiante = e.IDEstudiante  
29   INNER JOIN Cursos c ON i.IDCurso = c.IDCurso;
```

	nombre character varying (50) 	nombrcurso character varying (100) 
1	Damian	Bases de Datos
2	Luis	Calculo diferencial
3	Donovan	Psicología General
4	Emanuel	Administración 1
5	Clara	Diseño Digital

Uni la tabla inscripciones con estudiantes y cursos y cree los alias nombre y nombre cursos donde nombre tiene un alias de la tabla estudiantes y nombre curso un alias de la tabla curso y asi usando los inner join para poder mostrar los datos que estan representados en la imagen.

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).

```
34 ▾ SELECT e.Nombre, c.NombreCurso
35   FROM Estudiantes e
36   LEFT JOIN Inscripciones i ON e.IDEstudiante = i.IDEstudiante
37   LEFT JOIN Cursos c ON i.IDCurso = c.IDCurso;
```

	nombre character varying (50) 	nombrecurso character varying (100) 
1	Damian	Bases de Datos
2	Luis	Calculo diferencial
3	Donovan	Psicología General
4	Emanuel	Administración 1
5	Clara	Diseño Digital
6	Erick	[null]

Erick se muestra como “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.

```
41 ▾ SELECT c.NombreCurso, e.Nombre
42   FROM Cursos c
43   RIGHT JOIN Inscripciones i ON c.IDCurso = i.IDCurso
44   RIGHT JOIN Estudiantes e ON i.IDEstudiante = e.IDEstudiante;
```

	nombrecurso character varying (100) 	nombre character varying (50) 
1	Bases de Datos	Damian
2	Calculo diferencial	Luis
3	Psicología General	Donovan
4	Administración 1	Emanuel
5	Diseño Digital	Clara
6	[null]	Erick

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

```
48 ✓ SELECT c.NombreCurso, COUNT(i.IDEstudiante) AS CantidadEstudiantes
49   FROM Cursos c
50   LEFT JOIN Inscripciones i ON c.IDCurso = i.IDCurso
51   GROUP BY c.NombreCurso;
```

	nombrcurso character varying (100)	cantidadestudiantes bigint
1	Calculo diferencial	1
2	Psicología General	1
3	Administración 1	1
4	Bases de Datos	1
5	Diseño Digital	1

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

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

	nombre character varying (50)	apellido character varying (50)	fechanacimiento date
1	Donovan	Hernandez	1998-06-20

Mi compañero donovan ya esta bien ruco

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

```
59 ✓ SELECT NombreCurso
60   FROM Cursos
61   ORDER BY NombreCurso ASC;
```

	nombrcurso character varying (100)
1	Administración 1
2	Bases de Datos
3	Calculo diferencial
4	Diseño Digital
5	Psicología General

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.

```

66 v WITH InscripcionesPorEstudiante AS (
67   SELECT e.IDEstudiante, e.Nombre, COUNT(i.IDInscripcion) AS TotalInscripciones
68   FROM Estudiantes e
69   LEFT JOIN Inscripciones i ON e.IDEstudiante = i.IDEstudiante
70   GROUP BY e.IDEstudiante, e.Nombre
71 )
72   SELECT Nombre, TotalInscripciones
73   FROM InscripcionesPorEstudiante
74   ORDER BY TotalInscripciones DESC
75   LIMIT 3;

```

	nombre character varying (50)	totalinscripciones bigint
1	Donovan	1
2	Clara	1
3	Luis	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.

```

79 v WITH ConteoInscripciones AS (
80   SELECT c.IDCurso, c.NombreCurso, d.NombreDepartamento, COUNT(i.IDEstudiante) AS TotalEstudiantes
81   FROM Cursos c
82   INNER JOIN Departamentos d ON c.IDDepartamento = d.IDDepartamento
83   LEFT JOIN Inscripciones i ON c.IDCurso = i.IDCurso
84   GROUP BY c.IDCurso, c.NombreCurso, d.NombreDepartamento
85 ),
86 Maximos AS (
87   SELECT NombreDepartamento, MAX(TotalEstudiantes) AS MaxEstudiantes
88   FROM ConteoInscripciones
89   GROUP BY NombreDepartamento
90 )
91   SELECT ci.NombreDepartamento, ci.NombreCurso, ci.TotalEstudiantes
92   FROM ConteoInscripciones ci
93   INNER JOIN Maximos m ON ci.NombreDepartamento = m.NombreDepartamento AND ci.TotalEstudiantes = m.MaxEstudiantes;

```

	nombredepartamento character varying (70)	nombrcurso character varying (100)	totalestudiantes bigint
1	Humanidades	Psicología General	1
2	Diseño Gráfico	Diseño Digital	1
3	Ingeniería de Sistemas	Bases de Datos	1
4	Administración	Administración 1	1
5	Ciencias Basicas	Calculo diferencial	1

Solo hay 1 estudiante por cada curso.

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.

```

97  SELECT p.Nombre, p.Apellido, c.CantidadCursos
98  FROM Profesores p
99  INNER JOIN (
100     SELECT IDProfesor, COUNT(*) AS CantidadCursos
101     FROM CursosProfesores
102     GROUP BY IDProfesor
103 ) c ON p.IDProfesor = c.IDProfesor
104 WHERE c.CantidadCursos > 2;

```

	nombre character varying (50)	apellido character varying (50)	cantidadcursos bigint

No muestra nada debido a que no tengo un profesor que imparta mas de 1 curso.

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.

```

108  WITH Promedios AS (
109      SELECT pe.IDPrograma, pe.NombrePrograma, c.IDCurso, c.NombreCurso, AVG(i.Calificacion) AS Promedio
110      FROM ProgramasEstudio pe
111      INNER JOIN ProgramasCursos pc ON pe.IDPrograma = pc.IDPrograma
112      INNER JOIN Cursos c ON pc.IDCurso = c.IDCurso
113      LEFT JOIN Inscripciones i ON c.IDCurso = i.IDCurso
114      GROUP BY pe.IDPrograma, pe.NombrePrograma, c.IDCurso, c.NombreCurso
115  ),
116  Maximos AS (
117      SELECT IDPrograma, MAX(Promedio) AS MaxPromedio
118      FROM Promedios
119      GROUP BY IDPrograma
120  )
121  SELECT p.NombrePrograma, p.NombreCurso, p.Promedio
122  FROM Promedios p
123  INNER JOIN Maximos m ON p.IDPrograma = m.IDPrograma AND p.Promedio = m.MaxPromedio;

```

	nombreprograma character varying (100)	nombrecurso character varying (100)	promedio numeric
1	Programa de Ciencias	Calculo diferencial	85.000000000000000000
2	Programa de Administración	Administración 1	88.000000000000000000
3	Programa de Psicología	Psicología General	92.000000000000000000
4	Programa de Sistemas	Bases de Datos	90.000000000000000000
5	Programa de Diseño	Diseño Digital	95.000000000000000000