



# **Instituto Tecnológico de Mexicali**

**Ingeniería Sistemas Computacionales**

**Fundamentos De Base De Datos**

**Actividad:**

**"Examen Final de Unidad 4"**

**Alumno:**

**Delgado Rodriguez Jonhatan Eduardo**

**No. Control:**

**C21490757**

**Profesor:**

**José Ramón Bogarin Valenzuela**

**Mexicali, B.C. a 23 de Mayo de 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.

# Creación de Tablas

Query Query History

```
1  Create Table Estudiante (
2  Id_Estudiante Serial primary key,
3  Nombre Varchar(50) not null,
4  Apellido Varchar(50) not null,
5  Fecha_De_Nacimiento DATE not null,
6  Direccion Varchar(100) not null,
7  Ciudad Varchar(50) not null,
8  Email Varchar(50) unique not null
9  );
10 Create Table Departamento (
11 Id_Departamento Serial primary key,
12 NombreDepartamento Varchar(50) not null,
13 Edificio VarChar(30) not null
14 );
15
16 Create Table Cursos (
17 Id_Cursos Serial primary key,
18 NombreCurso Varchar(50) not null,
19 Descripcion Varchar(80) not null,
20 Creditos int not null,
21 Semestre int not null,
22 Id_Departamento int References Departamento(Id_Departamento) on delete cascade
23 );
24
25 Create Table Inscripciones (
26 Id_Inscripcion Serial primary key ,
27 Id_Estudiante int References Estudiante(Id_Estudiante) on delete cascade,
28 Id_Cursos int References Cursos(Id_Cursos) on delete cascade,
29 FechaInscripcion Date not null,
30 Calificacion Decimal(10,2) not null );
31
```

Query Query History

```
32 Create Table Profesores(
33 Id_Profesor Serial Primary Key,
34 NombreProfesor VarChar(50) not null,
35 Apellido Varchar(50)not null,
36 Titulo VarChar(50) not null,
37 Id_Departamento int References Departamento(Id_Departamento) on delete cascade
38 );
39 Create Table Aulas(
40 Id_Aulas Serial primary key ,
41 NombreAula Varchar(50)not null,
42 Capacidad int not null,
43 Ubicacion Varchar(50) not null
44 );
45 Create Table Horarios(
46 Id_Horario Serial primary key ,
47 Id_Cursos int References Cursos(Id_Cursos) on delete cascade,
48 Id_Aulas int References Aulas(Id_Aulas) on delete cascade,
49 FechaInicio DATE NOT NULL,
50 FechaFin DATE NOT NULL,
51 HoraInicio TIME NOT NULL,
52 HoraFin TIME NOT NULL
53 );
54
55 Create Table CursosProfesores(
56 Id_CursoProfesor Serial Primary key,
57 Id_Cursos int References Cursos(Id_Curso s) on delete cascade,
58 Id_Profesor int References Profesores(Id_Profesor) on delete cascade
59 );
```

```
Create table ProgramasEstudio(
Id_Programa Serial Primary key,
NombrePrograma Varchar(50)not null,
DescripcionPrograma Varchar(80) not null
);
```

```
Create table ProgramasCursos(
Id_ProgramaCurso Serial Primary key,
Id_Programa int References ProgramasEstudio(Id_Programa) on delete cascade,
Id_Cursos int References Cursos(Id_Cursos) on delete cascade
);
```

## Agregar Tablas

Query Query History

```
1  Create Table Campus(  
2    Id_Campus Serial Primary key,  
3    NombreCampus Varchar(80) not null,  
4    DireccionCampus Varchar(100) not null  
5  );  
6  Create Table Carreras(  
7    Id_Carrera Serial Primary key,  
8    NombreCarrera Varchar(50) not null,  
9    Titulo_Otorgado Varchar(50) not null  
10 );  
11 Create Table CarrerasEstudiante(  
12 Id_CarrerasEstudiantes Serial Primary key,  
13 Id_Carrera int References Carreras(Id_Carrera) on delete cascade,  
14 Id_Estudiante int References Estudiante(Id_Estudiante) on delete cascade  
15 );
```

Tables (13)

- > aulas
- > campus
- > carreras
- > carrerasestudiante
- > cursos
- > cursosprofesores
- > departamento
- > estudiante
- > horarios
- > inscripciones
- > profesores
- > programascursos
- > programasestudio

## Modificar Tablas

Query Query History

```
1  -- En la tabla Estudiantes,agregar una clave foránea ID_Carrera que haga referencia a la tabla Carreras--  
2  alter table Estudiante  
3  add column Id_Carrera int References Carreras(Id_Carrera) on delete cascade;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 85 msec.

estudiante

Columns (8)

- id\_estudiante
- nombre
- apellido
- fecha\_de\_nacimiento
- direccion
- ciudad
- email
- id\_carrera

```
5  --En la tabla Cursos, agregar una columna IDCampus como clave foránea, referenciando la tabla Campus.--  
6  alter table Cursos  
7  add column Id_Campus int References Campus(Id_Campus) on delete cascade;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 174 msec.

cursos

Columns (7)

- id\_cursos
- nombrecurso
- descripcion
- creditos
- semestre
- id\_departamento
- id\_campus

# Modificar Tablas

```
9 -- Modificar la tabla Profesores para incluir un campo Email--
10 v alter Table Profesores
11 add column email VARCHAR(80) UNIQUE NOT NULL;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 63 msec.

✓ profesores

Columns (6)

id\_profesor

nombreprofesor

apellido

titulo

id\_departamento

email

```
--
13 --Eliminar la columna Ciudad de la tabla Estudiantes--
14 v ALTER TABLE Estudiante
15 drop column Ciudad;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 68 msec.

✓ estudiante

Columns (7)

id\_estudiante

nombre

apellido

fecha\_de\_nacimiento

direccion

email

id\_carrera

```
16 --Eliminar la tabla Aulas--
17 drop table Aulas cascade
```

Data Output Messages Notifications

NOTICE: eliminando además restricción «horarios\_id\_aulas\_fkey» en tabla horarios

DROP TABLE

Query returned successfully in 85 msec.

✓ Tables (12)

> campus

> carreras

> carrerasestudiante

> cursos

> cursosprofesores

> departamento

> estudiante

> horarios

> inscripciones

> profesores

> programascursos

> programasestudio

# Insert de Datos

Query Query History

```
1  --INSERT DE DATOS EN TABLAS--
2  -- Tabla: departamento--
3  ✓ INSERT INTO departamento (id_departamento, nombredepartamento, edificio) VALUES
4  (1, 'Ingeniería en Sistemas', 'Edificio A1'),
5  (2, 'Ingeniería Industrial', 'Edificio B1'),
6  (3, 'Ciencias Básicas', 'Edificio C1'),
7  (4, 'Administración', 'Edificio D1'),
8  (5, 'Electromecánica', 'Edificio E1');
9
10 -- Tabla: campus--
11 ✓ INSERT INTO campus (id_campus, nombrecampus, direccioncampus) VALUES
12 (1, 'Instituto Tecnológico de Mexicali', 'Av. Instituto Politécnico Nacional S/N, Col. Plutarco Elías Calles');
13
14 -- Tabla: carreras--
15 ✓ INSERT INTO carreras (id_carrera, nombrecarrera, titulo_otorgado) VALUES
16 (1, 'Ingeniería en Sistemas Computacionales', 'Ingeniero en Sistemas Computacionales'),
17 (2, 'Ingeniería Industrial', 'Ingeniero Industrial'),
18 (3, 'Licenciatura en Administración', 'Licenciado en Administración'),
19 (4, 'Ingeniería Electromecánica', 'Ingeniero Electromecánico'),
20 (5, 'Contador Público', 'Licenciado en Contaduría');
21
22 -- Tabla: estudiante--
23 ✓ INSERT INTO estudiante (id_estudiante, nombre, apellido, fecha_de_nacimiento, direccion, email, id_carrera) VALUES
24 (1, 'Alejandro', 'Gómez', '2000-01-15', 'Col. Pueblo Nuevo #123', 'alejandro.gomez@itmexicali.edu.mx', 1),
25 (2, 'María', 'Luna', '1999-12-05', 'Col. Independencia #456', 'maria.luna@itmexicali.edu.mx', 2),
26 (3, 'Luis', 'Hernández', '2001-05-21', 'Col. Robledo #789', 'luis.hernandez@itmexicali.edu.mx', 3),
27 (4, 'Carmen', 'Salazar', '2000-07-03', 'Col. Libertad #321', 'carmen.salazar@itmexicali.edu.mx', 4),
28 (5, 'Jorge', 'Ruiz', '2000-03-30', 'Col. Jardines del Lago #654', 'jorge.ruiz@itmexicali.edu.mx', 5);
29
```

Query Query History

```
30
31 ✓ INSERT INTO profesores (id_profesor, nombreprofesor, apellido, titulo, id_departamento, email) VALUES
32 (1, 'Víctor', 'Medina', 'Doctor en Ciencias de la Computación', 1, 'victor.medina@itmexicali.edu.mx'),
33 (2, 'Silvia', 'Ramírez', 'Maestra en Ingeniería Industrial', 2, 'silvia.ramirez@itmexicali.edu.mx'),
34 (3, 'Arturo', 'Meza', 'Licenciado en Matemáticas', 3, 'arturo.meza@itmexicali.edu.mx'),
35 (4, 'Rosa', 'Hernández', 'Maestra en Administración', 4, 'rosa.hernandez@itmexicali.edu.mx'),
36 (5, 'Luis', 'Escobar', 'Doctor en Ingeniería Eléctrica', 5, 'luis.escobar@itmexicali.edu.mx');
37
38 -- Tabla: programasestudio--
39 ✓ INSERT INTO programasestudio (id_programa, nombreprograma, descripcionprograma) VALUES
40 (1, 'Plan 2015 ISC', 'Programa educativo de Ingeniería en Sistemas 2015'),
41 (2, 'Plan 2018 II', 'Programa educativo de Ingeniería Industrial 2018'),
42 (3, 'Plan 2016 ADM', 'Programa educativo de Administración 2016'),
43 (4, 'Plan 2017 ELEC', 'Programa educativo de Electromecánica 2017'),
44 (5, 'Plan 2019 CP', 'Programa educativo de Contaduría Pública 2019');
45
46 -- Tabla: cursos--
47 ✓ INSERT INTO cursos (id_cursos, nombrecurso, descripcion, creditos, semestre, id_departamento, id_campus) VALUES
48 (1, 'Estructura de Datos', 'Manejo de estructuras lineales y no lineales', 5, 4, 1, 1),
49 (2, 'Estadística', 'Probabilidad y estadística aplicada', 4, 3, 2, 1),
50 (3, 'Contabilidad Financiera', 'Fundamentos de contabilidad básica', 4, 2, 4, 1),
51 (4, 'Circuitos Eléctricos', 'Análisis de circuitos eléctricos de CD y CA', 5, 3, 5, 1),
52 (5, 'Álgebra Lineal', 'Espacios vectoriales, matrices, determinantes', 4, 1, 3, 1);
53
```

# Insert de Datos

Query Query History

```
54 -- Tabla: programascursos--
55 ▾ INSERT INTO programascursos (id_programacurso, id_programa, id_cursos) VALUES
56 (1, 1, 1),
57 (2, 2, 2),
58 (3, 3, 3),
59 (4, 4, 4),
60 (5, 1, 5);
61
62 -- Tabla: cursosprofesores--
63 ▾ INSERT INTO cursosprofesores (id_cursoProfesor, id_cursos, id_profesor) VALUES
64 (1, 1, 1),
65 (2, 2, 2),
66 (3, 3, 4),
67 (4, 4, 5),
68 (5, 5, 3);
69
70 -- Tabla: horarios (sin aula)--
71 ▾ INSERT INTO horarios (id_horario, id_cursos, id_aulas, fechainicio, fechafin, horainicio, horafin) VALUES
72 (1, 1, NULL, '2025-01-22', '2025-06-10', '08:00', '10:00'),
73 (2, 2, NULL, '2025-01-22', '2025-06-10', '10:00', '12:00'),
74 (3, 3, NULL, '2025-01-22', '2025-06-10', '13:00', '15:00'),
75 (4, 4, NULL, '2025-01-22', '2025-06-10', '15:00', '17:00'),
76 (5, 5, NULL, '2025-01-22', '2025-06-10', '17:00', '19:00');
77
78 -- Tabla: inscripciones--
79 ▾ INSERT INTO Inscripciones (id_inscripcion, id_estudiante, id_cursos, FechaInscripcion, calificacion) VALUES
80 (1, 1, 1, '2025-01-10', 95.0),
81 (2, 2, 2, '2025-01-11', 89.5),
82 (3, 3, 3, '2025-01-12', 92.0),
83 (4, 4, 4, '2025-01-13', 85.0),
84 (5, 5, 5, '2025-01-14', 90.5);
85
86 -- Tabla: carrerasestudiante--
87 ▾ INSERT INTO carrerasestudiante (id_carrerasestudiantes, id_carrera, id_estudiante) VALUES
88 (1, 1, 1),
89 (2, 2, 2),
90 (3, 3, 3),
91 (4, 4, 4),
92 (5, 5, 5);
```

Data Output Messages Notifications

INSERT 0 5

Query returned successfully in 51 msec.

## Actualizando Datos

```
2 --CAMBIAR LA DIRECCION DE UN ESTUDIANTE--
3 UPDATE estudiante
4 SET direccion = 'CALLE NOVENA #9'
5 WHERE id_estudiante = 4;
6 SELECT E.Id_Estudiante,E.nombre as Nombre_Estudiante,E.direccion as direccion FROM ESTUDIANTE E
```

Data Output Messages Notifications

	id_estudiante [PK] integer	nombre_estudiante character varying (50)	direccion character varying (100)
1	1	Alejandro	Col. Pueblo Nuevo #123
2	2	Maria	Col. Independencia #456
3	3	Luis	Col. Robledo #789
4	5	Jorge	Col. Jardines del Lago #654
5	4	Carmen	CALLE NOVENA #9

```
12 --Cambiar hora de inicio y hora de fin de un curso--
13 UPDATE horarios
14 SET horainicio = '09:00', horafin = '11:00'
15 WHERE id_cursos = 2;
16 select h.horainicio, h.horafin, h.id_cursos from horarios h;
```

Data Output Messages Notifications

	horainicio time without time zone	horafin time without time zone	id_cursos integer
1	08:00:00	10:00:00	1
2	13:00:00	15:00:00	3
3	15:00:00	17:00:00	4
4	17:00:00	19:00:00	5
5	09:00:00	11:00:00	2

## Eliminación de Datos

```
17 --Borramos la columna id_aulas ya que esta completamente en null y no se esta usando--
18 ALTER TABLE horarios
19 DROP COLUMN id_aulas
20 SELECT * FROM horarios
```

Data Output Messages Notifications

	id_horario [PK] integer	id_cursos integer	fechainicio date	fechafin date	horainicio time without time zone	horafin time without time zone
1	1	1	2025-01-22	2025-06-10	08:00:00	10:00:00
2	3	3	2025-01-22	2025-06-10	13:00:00	15:00:00
3	4	4	2025-01-22	2025-06-10	15:00:00	17:00:00
4	5	5	2025-01-22	2025-06-10	17:00:00	19:00:00
5	2	2	2025-01-22	2025-06-10	09:00:00	11:00:00



# Realización de Consultas

Query Query History

Execute script F5

```
1 --CONSULTAS--
2 --Consulta que muestere el nombre de un estudiante, a que carrera pertenece y que curso lleva--
3 SELECT e.nombre AS nombre_estudiante, c.nombrecarrera AS carrera, cu.nombrecurso AS curso
4 FROM estudiante e
5 INNER JOIN carreras c ON e.id_estudiante = c.id_estudiante
6 INNER JOIN carreras c ON ce.id_carrera = c.id_carrera
7 INNER JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
8 INNER JOIN cursos cu ON i.id_cursos = cu.id_cursos;
9 --Consulta que muestra el nombre de un est--
10 SELECT Concat(e.nombre,' ',e.apellido) as nombre_estudiante, cu.nombrecurso AS curso,
11 CONCAT(p.nombreprofesor, ' ', p.apellido) AS Profesor,i.calificacion FROM estudiante e
```

Data Output Messages Notifications

Showing rows: 1 to 5

	nombre_estudiante character varying (50)	carrera character varying (50)	curso character varying (50)
1	Alejandro	Ingeniería en Sistemas Computacionales	Estructura de Datos
2	Maria	Ingeniería Industrial	Estadística
3	Luis	Licenciatura en Administración	Contabilidad Financiera
4	Carmen	Ingeniería Electromecánica	Circuitos Eléctricos
5	Jorge	Contador Público	Álgebra Lineal

```
9 --Consulta que muestra el nombre completo de un alumno,el curso al que pertenece,el profesor que lo imparte
10 -- y la calificacion que tiene--
11 SELECT Concat(e.nombre,' ',e.apellido) as nombre_estudiante, cu.nombrecurso AS curso,
12 CONCAT(p.nombreprofesor, ' ', p.apellido) AS Profesor,i.calificacion FROM estudiante e
13 INNER JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
14 INNER JOIN cursos cu ON i.id_cursos = cu.id_cursos
15 INNER JOIN cursosprofesores cp ON cu.id_cursos = cp.id_cursos
16 INNER JOIN profesores p ON cp.id_profesor = p.id_profesor
17 WHERE e.id_estudiante = 3;
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1

	nombre_estudiante text	curso character varying (50)	profesor text	calificacion numeric (10,2)
1	LuisHernández	Contabilidad Financiera	Rosa Hernández	92.00

```
17 --Consulta que muestre el nombre de un estudiante,el curso al que pertenece y la fecha en la que se inscribio--
18 SELECT e.nombre AS nombre_estudiante,c.nombrecurso AS curso, i.fechainscripcion
19 FROM estudiante e
20 INNER JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
21 INNER JOIN cursos c ON i.id_cursos = c.id_cursos
22 WHERE e.id_estudiante = 1;
23
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1

	nombre_estudiante character varying (50)	curso character varying (50)	fechainscripcion date
1	Alejandro	Estructura de Datos	2025-01-10

# Consultas Especificas

Query Query History

```
1 -- 1. Selección Básica: Muestra todos los nombres y apellidos de los estudiantes--
2 Select E.Nombre,E.apellido from Estudiante E
```

Data Output Messages Notifications

Showing rows:		
	nombre character varying (50)	apellido character varying (50)
1	Alejandro	Gómez
2	María	Luna
3	Luis	Hernández
4	Jorge	Ruiz
5	Carmen	Salazar

```
3 -- 2. Cláusula WHERE: Encuentra todos los cursos que tienen 3 créditos.--
4 --insert de 2 cursos con 3 creditos porque en mis primeras inserciones no tenia ninguno--
5 INSERT INTO cursos (id_cursos, nombrecurso, descripcion, creditos, semestre) VALUES
6 (6, 'Bases de Datos', 'Introducción a los sistemas de bases de datos relacionales.', 3, 5),
7 (7, 'Redes de Computadoras', 'Fundamentos de redes, modelos OSI y TCP/IP.', 3, 6);
8 SELECT nombrecurso FROM cursos
9 WHERE creditos = 3;
```

Data Output Messages Notifications

Showing rows: 1 to 2	
	nombrecurso character varying (50)
1	Bases de Datos
2	Redes de Computadoras

```
10 --3. INNER JOIN: Obtén una lista que muestre el nombre del estudiante y el nombre del curso en el que está inscrito.--
11 SELECT e.nombre AS nombre_estudiante, c.nombrecurso AS curso FROM estudiante e
12 INNER JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
13 INNER JOIN cursos c ON i.id_cursos = c.id_cursos;
```

Data Output Messages Notifications

Showing rows: 1 to 5 Page No: 1 of 1		
	nombre_estudiante character varying (50)	curso character varying (50)
1	Alejandro	Estructura de Datos
2	María	Estadística
3	Luis	Contabilidad Financiera
4	Carmen	Circuitos Eléctricos
5	Jorge	Álgebra Lineal

```
15 --LEFT JOIN: Muestra todos los estudiantes y, si están inscritos en algún curso, el nombre del curso.--
16 --Si un estudiante no está inscrito en ningún curso, el campo del nombre del curso--
17 --debe mostrar un valor que lo indique (ej: NULL).--
18 INSERT INTO estudiante (id_estudiante, nombre, apellido, fecha_de_nacimiento, direccion,email)VALUES
19 (99, 'Carlos', 'Ruiz', '2000-05-10', 'Col. Hidalgo #500','CarloRuiz@itmexicali.edu.mx');
20 --insert de alumno no inscrito en ningun curso para la prueba de la consulta--
21 SELECT e.nombre AS nombre_estudiante, c.nombrecurso AS Nombre_Curso FROM estudiante e
22 LEFT JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
23 LEFT JOIN cursos c ON i.id_cursos = c.id_cursos;
```

Data Output Messages Notifications

Showing rows: 1 to 6 Page No:		
	nombre_estudiante character varying (50)	curso character varying (50)
1	Alejandro	Estructura de Datos
2	María	Estadística
3	Luis	Contabilidad Financiera
4	Carmen	Circuitos Eléctricos
5	Jorge	Álgebra Lineal
6	Carlos	[null]

## Consultas Especificas

```
25 --RIGHT JOIN: Lista todos los cursos y, si tienen estudiantes inscritos, el nombre de los estudiantes.
26 --Muestra todos los cursos, incluso si no tienen estudiantes inscritos actualmente--
27 ✓ SELECT e.nombre AS nombre_estudiante, c.nombrecurso AS curso FROM estudiante e
28 RIGHT JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
29 RIGHT JOIN cursos c ON i.id_cursos = c.id_cursos;
```

Data Output Messages Notifications

Showing rows: 1 to 7 Page No

	nombre_estudiante character varying (50)	curso character varying (50)
1	Alejandro	Estructura de Datos
2	Maria	Estadística
3	Luis	Contabilidad Financiera
4	Carmen	Circuitos Eléctricos
5	Jorge	Álgebra Lineal
6	[null]	Bases de Datos
7	[null]	Redes de Computador...

```
31 --6. GROUP BY y COUNT: Calcula cuántos estudiantes están inscritos en cada curso--
32 --Muestra el nombre del curso y la cantidad de estudiantes--
33 ✓ SELECT c.nombrecurso AS Nombre_Curso, COUNT(i.id_estudiante) AS Cantidad_De_Estudiantes FROM Cursos c
34 LEFT JOIN Inscripciones i ON c.id_cursos = i.id_cursos
35 GROUP BY c.nombrecurso
```

Data Output Messages Notifications

Showing rows: 1 to 7 Page No

	nombre_curso character varying (50)	cantidad_de_estudiantes bigint
1	Redes de Computador...	0
2	Circuitos Eléctricos	1
3	Estructura de Datos	1
4	Contabilidad Financiera	1
5	Álgebra Lineal	1
6	Estadística	1
7	Bases de Datos	0

```
37 --7.-BETWEEN: Encuentra todos los estudiantes que nacieron entre el 1 de enero de 1995 y el 31 de diciembre de 1998.--
38 -- Insert de dos alumnos que nacieron en esa el rango de fecha especificado --
39 INSERT INTO estudiante (id_estudiante,nombre, apellido, fecha_de_nacimiento, direccion, email, id_carrera)VALUES
40 ('7','Dario', 'Ramírez', '1997-11-02', 'Calle Luna 456', 'carlos.ramirez@email.com', 2),
41 ('8','Laura', 'González', '1996-05-14', 'Av. Siempre Viva 123', 'laura.gonzalez@email.com', 1);
42 ✓ SELECT * FROM estudiante
43 WHERE fecha_de_nacimiento BETWEEN '1995-01-01' AND '1998-12-31';
44
```

Data Output Messages Notifications

Showing rows: 1 to 2 Page No: 1 of 1

	id_estudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fecha_de_nacimiento date	direccion character varying (100)	email character varying (50)	id_carrera integer
1	7	Dario	Ramírez	1997-11-02	Calle Luna 456	carlos.ramirez@email.com	2
2	8	Laura	González	1996-05-14	Av. Siempre Viva 123	laura.gonzalez@email.com	1

# Consultas Especificas

Query Query History

```
1 --8. ORDER BY: Muestra todos los cursos ordenados alfabéticamente por su nombre--
2 SELECT *FROM cursos
3 ORDER BY nombrecurso ASC;
4
```

Data Output Messages Notifications

Showing rows: 1 to 7 Page No: 1								
	id_cursos [PK] integer	nombrecurso character varying (50)	descripcion character varying (80)	creditos integer	semestre integer	id_departamento integer	id_campus integer	
1	5	Álgebra Lineal	Espacios vectoriales, matrices, determinantes	4	1	3	1	
2	6	Bases de Datos	Introducción a los sistemas de bases de datos relacionales.	3	5	[null]	[null]	
3	4	Circuitos Eléctricos	Análisis de circuitos eléctricos de CD y CA	5	3	5	1	
4	3	Contabilidad Financiera	Fundamentos de contabilidad básica	4	2	4	1	
5	2	Estadística	Probabilidad y estadística aplicada	4	3	2	1	
6	1	Estructura de Datos	Manejo de estructuras lineales y no lineales	5	4	1	1	
7	7	Redes de Computador...	Fundamentos de redes, modelos OSI y TCP/IP.	3	6	[null]	[null]	

```
88 --9.-CTE: Crea una tabla de expresión común que liste el número de inscripciones por estudiante luego--
89 --consulta esta CTE para encontrar los 3 estudiantes con más inscripciones mostrando el nombre del estudiante--
90 --y el número de inscripciones.--
91 WITH inscripciones_por_estudiante AS (
92 SELECT e.id_estudiante, e.nombre, e.apellido, COUNT(i.id_inscripcion) AS total_inscripciones FROM estudiante e
93 INNER JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
94 GROUP BY e.id_estudiante, e.nombre, e.apellido)
95 SELECT nombre as Nombre_Alumno, apellido as Apellido_Estudiante, total_inscripciones
96 FROM inscripciones_por_estudiante
97 ORDER BY total_inscripciones DESC
98 LIMIT 3;
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No: 1			
	nombre_alumno character varying (50)	apellido_estudiante character varying (50)	total_inscripciones bigint
1	Luis	Hernández	1
2	Jorge	Ruiz	1
3	Carmen	Salazar	1

```
15 --Consulta Compleja 1: Para cada departamento, muestra el nombre del departamento
16 --y el nombre del curso con la mayor cantidad de estudiantes inscritos.--
17 SELECT d.nombredepartamento,c.nombrecurso, COUNT(i.id_estudiante) AS total_inscritos FROM departamento d
18 INNER JOIN cursos c ON c.id_departamento = d.id_departamento
19 LEFT JOIN inscripciones i ON i.id_cursos = c.id_cursos
20 GROUP BY d.id_departamento, d.nombredepartamento, c.id_cursos, c.nombrecurso
21 HAVING COUNT(i.id_estudiante) = (
22 SELECT MAX(contador)
23 FROM ( SELECT COUNT(i2.id_estudiante) AS contador FROM cursos c2
24 LEFT JOIN inscripciones i2 ON i2.id_cursos = c2.id_cursos WHERE c2.id_departamento = d.id_departamento
25 GROUP BY c2.id_cursos) AS subconsulta
26 ) ORDER BY d.nombredepartamento;
27
```

Data Output Messages Notifications

Showing rows: 1 to 5 Page No:			
	nombredepartamento character varying (50)	nombrecurso character varying (50)	total_inscritos bigint
1	Administración	Contabilidad Financiera	1
2	Ciencias Básicas	Álgebra Lineal	1
3	Electromecánica	Circuitos Eléctricos	1
4	Ingeniería en Sistemas	Estructura de Datos	1
5	Ingeniería Industrial	Estadística	1

## Consultas Especificas

```
28 --. Consulta Compleja 2: Encuentra a los profesores que imparten más de dos
29 --cursos, mostrando su nombre, apellido y la cantidad de cursos que imparten.--
30 --INSERT PARA QUE FUNCIONE LA CONSULTA--
31 INSERT INTO cursosprofesores (id_cursoprofesor,id_profesor, id_cursos) VALUES
32 (6,1, 2),
33 (7,1, 3),
34 (8,2,5),
35 (9,2, 1),
36 (10,3, 4),
37 SELECT p.nombreprofesor,p.apellido,COUNT(cp.id_cursos) AS cantidad_cursos FROM profesores p
38 JOIN cursosprofesores cp ON p.id_profesor = cp.id_profesor
39 GROUP BY p.id_profesor, p.nombreprofesor, p.apellido
40 HAVING COUNT(cp.id_cursos) > 2
41 ORDER BY cantidad_cursos DESC;
```

Data Output Messages Notifications

	nombreprofesor character varying (50)	apellido character varying (50)	cantidad_cursos bigint
1	Silvia	Ramírez	3
2	Victor	Medina	3
3	Arturo	Meza	2

```
42 --Consulta Compleja 3: Lista los nombres de los programas de estudio y, para cada programa--
43 --el nombre del curso con el promedio de calificación más alto.--
44 SELECT p.nombreprograma,c.nombrecurso,ROUND(AVG(i.calificacion), 2) AS promedio FROM programasestudio p
45 INNER JOIN programascursos pc ON p.id_programa = pc.id_programa
46 INNER JOIN cursos c ON pc.id_cursos = c.id_cursos
47 JOIN inscripciones i ON c.id_cursos = i.id_cursos
48 GROUP BY p.id_programa, p.nombreprograma, c.id_cursos, c.nombrecurso
49 HAVING AVG(i.calificacion) = (SELECT MAX(prom) FROM (SELECT AVG(i2.calificacion) AS prom FROM programascursos pc2
50 INNER JOIN cursos c2 ON pc2.id_cursos = c2.id_cursos
51 INNER JOIN inscripciones i2 ON c2.id_cursos = i2.id_cursos
52 WHERE pc2.id_programa = p.id_programa
53 GROUP BY c2.id_cursos
54 ) AS sub
55 )
56 ORDER BY p.nombreprograma;
```

Data Output Messages Notifications

	nombreprograma character varying (50)	nombrecurso character varying (50)	promedio numeric
1	Plan 2015 ISC	Estructura de Datos	95.00
2	Plan 2016 ADM	Contabilidad Financiera	92.00
3	Plan 2017 ELEC	Circuitos Eléctricos	85.00
4	Plan 2018 II	Estadística	89.50