

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto tecnológico de Mexicali



Ingeniería en sistemas computacionales

Fundamentos de Base de datos

Examen Final de Unidad 4

JOSE RAMON BOGARIN VALENZUELA

Torres Enríquez Braulio Adán

22490366

Mexicali Baja California a 23 de mayo de 2025

Hora de entrega: 17:50

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.

Entidades Principales

- **Estudiantes**
- **Profesores**
- **Cursos**
- **Departamentos Académicos**
- **Aulas**
- **Programas de Estudio**

Atributos

Estudiantes: ID, Nombre, Apellido, Fecha Nacimiento, Dirección, Ciudad, Email.

Profesores: ID, Nombre, Apellido, Título.

Cursos: ID, Nombre Curso, Descripción, Créditos, Semestre.

Departamentos académicos: ID, Nombre Departamento, Edificio.

Aulas: ID, Nombre Aula, Capacidad, Ubicación.

Programas de estudio: ID, Nombre Programa, Descripción Programa.

Estudiantes

Campo	Tipo	Claves PK/FK
IDEstudainte	SERIAL	PK
Nombre	VARCHAR	
Apellido	VARCHAR	
Titulo	VARCHAR	
Direccion	VARCHAR	
Ciudad	VARCHAR	
Email	VARCHAR	

Profesores

Campo	Tipo	Claves PK/FK
IDProfesor	SERIAL	PK
Nombre	VARCHAR	
Apellido	VARCHAR	
Titulo	VARCHAR	
IDDepartamento	INT	FK

Cursos

Campo	Tipo	Claves PK/FK
IDCurso	SERIAL	PK
Nombre Curso	VARCHAR	
Descripción	TEXT	
Creditos	SMALLINT	
Semestre	VARCHAR	
IDdepartamento	INT	FK

Departamentos académicos

Campo	Tipo	Claves PK/FK
IDDepartamento	SERIAL	PK
Nombre Departamento	VARCHAR	
Edificio	VARCHAR	

Aulas

Campo	Tipo	Claves PK/FK
IDAula	SERIAL	PK
Nombre Aula	VARCHAR	
Capacidad	INT	
Ubicación	VARCHAR	

Programas de estudio

Campo	Tipo	Claves PK/FK
IDPrograma	SERIAL	PK
Nombre Programa	VARCHAR	
Descripción Programa	TEXT	

Horarios

Campo	Tipo	Claves PK/FK
IDHorario	SERIAL	PK
IDCurso	INT	FK
IDAula	INT	FK
FechaInicio	Date	
FechaFin	Date	
HoraInicio	TIME	
HoraFin	TIME	

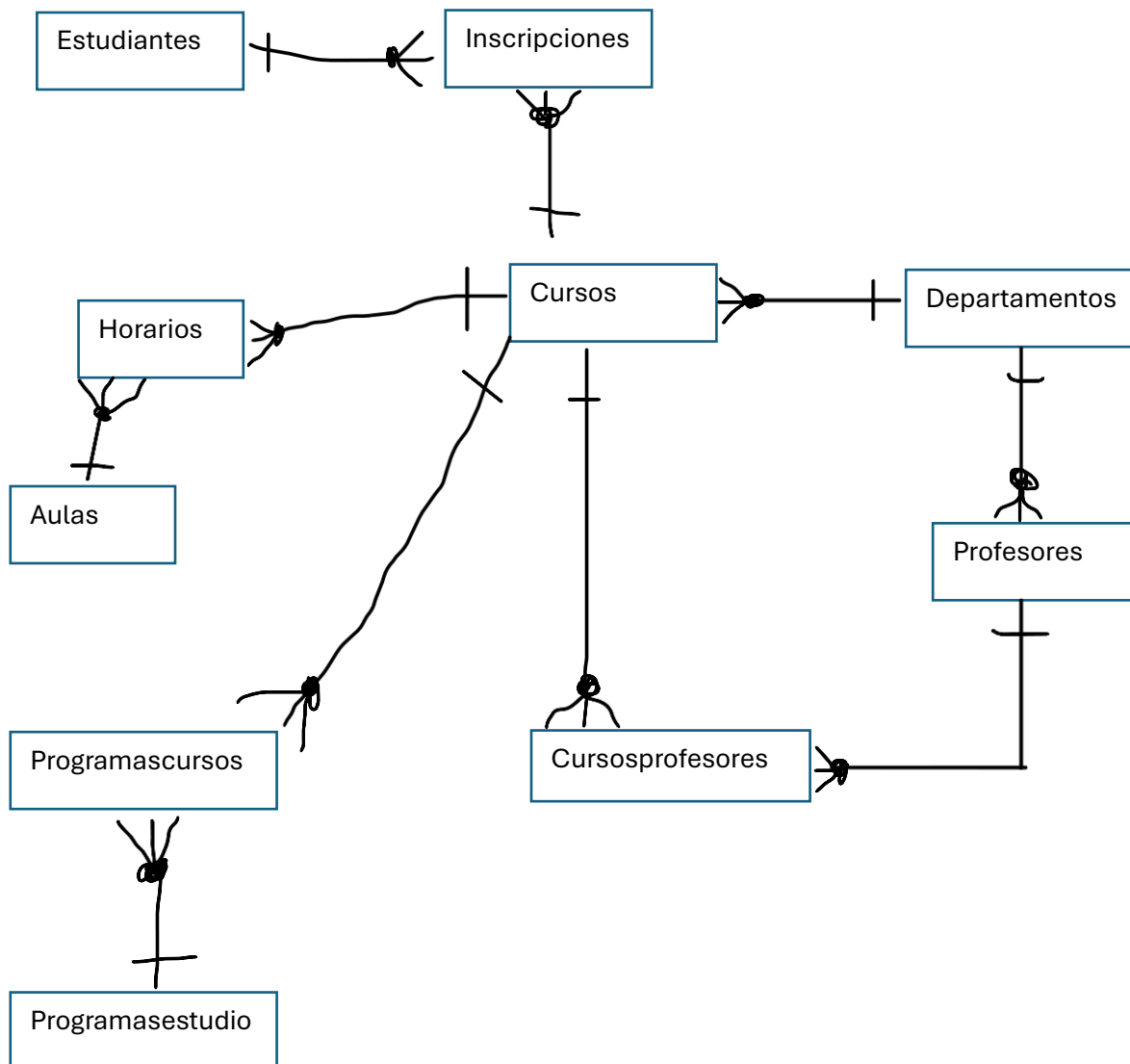
Cursos Profesores

Campo	Tipo	Claves PK/FK
IDCursosProfesor	SERIAL	PK
IDCurso	INT	FK
IDProfesor	INT	FK

Inscripciones

Campo	Tipo	Claves PK/FK
IDInscripcion	SERIAL	PK
IDEstudiante	INT	FK
IDCurso	INT	FK
FechaInscripcion	DATE	
Califiacion	DECIMAL	

Relación entre las entidades



Creación de tablas en PostgreSQL

```
1  -- 1. Tabla Departamentos (base para otras tablas)
2  ✓ CREATE TABLE Departamentos (
3      IDDepartamento SERIAL PRIMARY KEY,
4      NombreDepartamento VARCHAR(100) NOT NULL,
5      Edificio VARCHAR(50)
6  );
7
8  -- 2. Tabla ProgramasEstudio (independiente)
9  ✓ CREATE TABLE ProgramasEstudio (
10     IDPrograma SERIAL PRIMARY KEY,
11     NombrePrograma VARCHAR(100) UNIQUE NOT NULL,
12     DescripcionPrograma TEXT
13 );
14
15 -- 3. Tabla Aulas (independiente)
16 ✓ CREATE TABLE Aulas (
17     IDAula SERIAL PRIMARY KEY,
18     NombreAula VARCHAR(50) NOT NULL,
19     Capacidad INT NOT NULL,
20     Ubicacion VARCHAR(100) NOT NULL
21 );
22
23 -- 4. Tabla Profesores (depende de Departamentos)
24 ✓ CREATE TABLE Profesores (
25     IDProfesor SERIAL PRIMARY KEY,
26     Nombre VARCHAR(50) NOT NULL,
27     Apellido VARCHAR(50) NOT NULL,
28     Titulo VARCHAR(50),
29     IDDepartamento INT NOT NULL,
30     FOREIGN KEY (IDDepartamento) REFERENCES Departamentos(IDDepartamento)
31 );
32
```

```

33 -- 5. Tabla Cursos (depende de Departamentos)
34 ✓ CREATE TABLE Cursos (
35     IDCurso SERIAL PRIMARY KEY,
36     NombreCurso VARCHAR(100) NOT NULL,
37     Descripcion TEXT,
38     Creditos SMALLINT NOT NULL,
39     Semestre VARCHAR(20) NOT NULL,
40     IDDepartamento INT NOT NULL,
41     FOREIGN KEY (IDDepartamento) REFERENCES Departamentos(IDDepartamento)
42 );
43
44 -- 6. Tabla Estudiantes (independiente)
45 ✓ CREATE TABLE Estudiantes (
46     IDEstudiante SERIAL PRIMARY KEY,
47     Nombre VARCHAR(50) NOT NULL,
48     Apellido VARCHAR(50) NOT NULL,
49     FechaNacimiento DATE,
50     Direccion VARCHAR(100),
51     Ciudad VARCHAR(50),
52     Email VARCHAR(100) UNIQUE
53 );
54
55 -- 7. Tabla ProgramasCursos (relación muchos-a-muchos)
56 ✓ CREATE TABLE ProgramasCursos (
57     IDProgramaCurso SERIAL PRIMARY KEY,
58     IDPrograma INT UNIQUE NOT NULL,
59     IDCurso INT UNIQUE NOT NULL,
60     FOREIGN KEY (IDPrograma) REFERENCES ProgramasEstudio(IDPrograma) ON DELETE CASCADE,
61     FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCurso) ON DELETE CASCADE
62 );
63

```

```

64 -- 8. Tabla Horarios (depende de Cursos y Aulas)
65 ✓ CREATE TABLE Horarios (
66     IDHorario SERIAL PRIMARY KEY,
67     IDCurso INT NOT NULL,
68     IDAula INT NOT NULL,
69     FechaInicio DATE NOT NULL,
70     FechaFin DATE NOT NULL,
71     HoraInicio TIME NOT NULL,
72     HoraFin TIME NOT NULL,
73     FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCurso) ON DELETE CASCADE,
74     FOREIGN KEY (IDAula) REFERENCES Aulas(IDAula) ON DELETE CASCADE,
75     CHECK (FechaInicio <= FechaFin),
76     CHECK (HoraInicio < HoraFin)
77 );
78
79 -- 9. Tabla CursosProfesores (relación muchos-a-muchos)
80 ✓ CREATE TABLE CursosProfesores (
81     IDCursoProfesor SERIAL PRIMARY KEY,
82     IDCurso INT UNIQUE NOT NULL,
83     IDProfesor INT UNIQUE NOT NULL,
84     FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCurso) ON DELETE CASCADE,
85     FOREIGN KEY (IDProfesor) REFERENCES Profesores(IDProfesor) ON DELETE CASCADE
86 );
87
88 -- 10. Tabla Inscripciones (depende de Estudiantes y Cursos)
89 ✓ CREATE TABLE Inscripciones (
90     IDInscripcion SERIAL PRIMARY KEY,
91     IDEstudiante INT UNIQUE NOT NULL,
92     IDCurso INT UNIQUE NOT NULL,
93     FechaInscripcion DATE NOT NULL DEFAULT CURRENT_DATE,
94     Calificacion DECIMAL(3,1) CHECK (Calificacion BETWEEN 0 AND 10 OR Calificacion IS NULL),
95     FOREIGN KEY (IDEstudiante) REFERENCES Estudiantes(IDEstudiante) ON DELETE CASCADE,
96     FOREIGN KEY (IDCurso) REFERENCES Cursos(IDCurso) ON DELETE CASCADE
97 );

```


Agregar tablas Campus y Carreras

```
99  -- Tabla Campus
100  CREATE TABLE Campus (
101      IDCampus SERIAL PRIMARY KEY,
102      NombreCampus VARCHAR(100) NOT NULL,
103      DireccionCampus VARCHAR(200) NOT NULL
104  );
105
106  -- Tabla Carreras
107  CREATE TABLE Carreras (
108      IDCarrera SERIAL PRIMARY KEY,
109      NombreCarrera VARCHAR(100) UNIQUE NOT NULL,
110      TituloOtorgado VARCHAR(100) NOT NULL
111  );
-----
```

Agregar una relación de muchos a muchos entre Estudiantes y Carreras

```
112  -- MUCHOS A MUCHOS ESTUDIANTES CARRERAS
113  CREATE TABLE EstudiantesCarreras (
114      IEstudianteCarrera SERIAL UNIQUE PRIMARY KEY,
115      IEstudiante INT NOT NULL,
116      IDCarrera INT NOT NULL,
117      FOREIGN KEY (IEstudiante) REFERENCES Estudiantes(IEstudiante),
118      FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera)
119  );
```

Modificar Tablas

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

```
122 -- modificamos la tabla Estudiantes
123 ALTER TABLE Estudiantes ADD COLUMN IDCarrera INT;
124
125 -- Añadimos la clave foránea
126 ALTER TABLE Estudiantes ADD CONSTRAINT fk_estudiante_carrera
127 FOREIGN KEY (IDCarrera) REFERENCES Carreras(IDCarrera);
```

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

```
129 -- 2Agregar columna IDCampus a la tabla Cursos
130 ALTER TABLE Cursos ADD COLUMN IDCampus INT;
131
132 -- Añadir la clave foránea
133 ALTER TABLE Cursos ADD CONSTRAINT fk_curso_campus
134 FOREIGN KEY (IDCampus) REFERENCES Campus(IDCampus);
135
```

Modificar la tabla Profesores para incluir un campo Email

```
136 -- 3Agregar la columna Email
137 ALTER TABLE Profesores
138 ADD COLUMN Email VARCHAR(100) UNIQUE;
139
```

Eliminar Tablas/Campos

Eliminar la columna Ciudad de la tabla Estudiantes.

```
L40  --ELIMINAR  
L41  ALTER TABLE Estudiantes DROP COLUMN Ciudad;
```

Eliminar la tabla Aulas

```
L42  --Eliminar tabla Aulas|  
L43  DROP TABLE Aulas CASCADE;
```

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

```

145 -- Insertar datos
146 v INSERT INTO Campus (NombreCampus, DireccionCampus) VALUES
147 ('Campus Central', 'Av. Universidad 1000'),
148 ('Campus Norte', 'Calle Academia 200'),
149 ('Campus Sur', 'Boulevard Estudiantil 300');
150
151
152 v INSERT INTO Departamentos (NombreDepartamento, Edificio) VALUES
153 ('Ingeniería', 'Edificio A'),
154 ('Ciencias', 'Edificio B'),
155 ('Humanidades', 'Edificio C');
156
157 v INSERT INTO Carreras (NombreCarrera, TituloOtorgado) VALUES
158 ('Ingeniería de Software', 'Ingeniero'),
159 ('Lic. en Matemáticas', 'Licenciado'),
160 ('Lic. en Literatura', 'Licenciado');
161
162 v INSERT INTO Profesores (Nombre, Apellido, Titulo, IDDepartamento, Email) VALUES
163 ('Ana', 'García', 'PhD', 1, 'ana.garcia@uni.edu'),
164 ('Carlos', 'López', 'MSc', 2, 'carlos.lopez@uni.edu'),
165 ('María', 'Rodríguez', 'PhD', 3, 'maria.rodriguez@uni.edu');
166
167
168 v INSERT INTO Estudiantes (Nombre, Apellido, FechaNacimiento, Email) VALUES
169 ('Juan', 'Pérez', '2000-05-15', 'juan.perez@est.uni.edu'),
170 ('Leura', 'Gómez', '2001-03-22', 'leura.gomez@est.uni.edu'),
171 ('Pedro', 'Sánchez', '1999-11-10', 'pedro.sanchez@est.uni.edu');
172
173 v INSERT INTO ProgramasEstudio (NombrePrograma, DescripcionPrograma) VALUES
174 ('Programa de Ingeniería', 'Enfocado en desarrollo de software y sistemas'),
175 ('Programa de Ciencias Exactas', 'Abarca matemáticas y física avanzada'),
176 ('Programa de Artes Liberales', 'Estudios en humanidades y literatura');
177
178 v INSERT INTO Cursos (NombreCurso, Descripcion, Creditos, Semestre, IDDepartamento, IDCampus) VALUES
179 ('Programación I', 'Fundamentos de programación en Python', 4, '2023-1', 1, 1),
180 ('Cálculo I', 'Cálculo diferencial e integral', 5, '2023-1', 2, 2),
181 ('Literatura Universal', 'Análisis de obras clásicas', 3, '2023-2', 3, 3);
182
183 v INSERT INTO ProgramasCursos (IDPrograma, IDCurso) VALUES
184 (1, 1), -- Programa de Ingeniería -> Programación I
185 (2, 2), -- Programa de Ciencias Exactas -> Cálculo I
186 (3, 3); -- Programa de Artes Liberales -> Literatura Universal
187
188 v INSERT INTO Horarios (IDCurso, IDAula, FechaInicio, FechaFin, HoraInicio, HoraFin) VALUES
189 (1, 1, '2023-01-10', '2023-05-20', '08:00:00', '10:00:00'), -- Programación I
190 (2, 2, '2023-01-10', '2023-05-20', '14:00:00', '16:00:00'), -- Cálculo I
191 (3, 3, '2023-06-05', '2023-11-15', '10:00:00', '12:00:00'); -- Literatura Universal
192
193 v INSERT INTO CursosProfesores (IDCurso, IDProfesor) VALUES
194 (1, 1), -- Programación I -> Ana García
195 (2, 2), -- Cálculo I -> Carlos López
196 (3, 3); -- Literatura Universal -> María Rodríguez
197
198 v INSERT INTO Inscripciones (IDEstudiante, IDCurso, FechaInscripcion, Calificacion) VALUES
199 (1, 1, '2023-01-05', 9.5), -- Juan Pérez en Programación I
200 (2, 2, '2023-01-05', 8.0), -- Leura Gómez en Cálculo I
201 (3, 3, '2023-06-01', NULL); -- Pedro Sánchez en Literatura Universal (sin calificación aún)
202
203 v INSERT INTO EstudiantesCarreras (IDEstudiante, IDCarrera) VALUES
204 (1, 1), -- Juan Pérez -> Ingeniería de Software
205 (2, 2), -- Leura Gómez -> Lic. en Matemáticas
206 (3, 3); -- Pedro Sánchez -> Lic. en Literatura

```

Comprobación

```
208 --COMPROBACION
209 SELECT * FROM Campus;
210
```

Data Output Messages Notifications

	idcampus [PK] integer	nombrecampus character varying (100)	direccioncampus character varying (200)
1	1	Campus Central	Av. Universidad 1000
2	2	Campus Norte	Calle Academia 200
3	3	Campus Sur	Boulevard Estudiantil 300

```
211 SELECT * FROM Departamentos;
212
```

Data Output Messages Notifications

	iddepartamento [PK] integer	nombredepartamento character varying (100)	edificio character varying (50)
1	1	Ingeniería	Edificio A
2	2	Ciencias	Edificio B
3	3	Humanidades	Edificio C

213 **SELECT * FROM Carreras;**

214

Data Output Messages Notifications

	idcarrera [PK] integer	nombrecarrera character varying (100)	titulootorgado character varying (100)
1	1	Ingeniería de Software	Ingeniero
2	2	Lic. en Matemáticas	Licenciado
3	3	Lic. en Literatura	Licenciado

215 **SELECT * FROM Profesores;**

216

Data Output Messages Notifications

	idprofesor [PK] integer	nombre character varying (50)	apellido character varying (50)	titulo character varying (50)	iddepartamento integer	email character varying (100)
1	1	Ana	García	PhD	1	ana.garcia@uni.edu
2	2	Carlos	López	MSc	2	carlos.lopez@uni.edu
3	3	María	Rodríguez	PhD	3	maria.rodriguez@uni.edu

217 **SELECT * FROM Estudiantes;**

218

Data Output Messages Notifications

	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fechanacimiento date	direccion character varying (100)	email character varying (100)	idcarrera integer
1	1	Juan	Pérez	2000-05-15	[null]	juan.perez@est.uni.edu	[null]
2	2	Laura	Gómez	2001-03-22	[null]	laura.gomez@est.uni.edu	[null]
3	3	Pedro	Sánchez	1999-11-10	[null]	pedro.sanchez@est.uni.edu	[null]

219 **SELECT * FROM ProgramasEstudio;**

220

Data Output Messages Notifications

	idprograma [PK] integer	nombreprograma character varying (100)	descripcionprograma text
1	1	Programa de Ingeniería	Enfocado en desarrollo de software y sistemas
2	2	Programa de Ciencias Exactas	Abarca matemáticas y física avanzada
3	3	Programa de Artes Liberales	Estudios en humanidades y literatura

221 **SELECT * FROM Cursos;**

222

Data Output Messages Notifications

	idcurso [PK] integer	nombrecurso character varying (100)	descripcion text	creditos smallint	semestre character varying (20)	iddepartamento integer	idcampus integer
1	1	Programación I	Fundamentos de programación en Python	4	2023-1	1	1
2	2	Cálculo I	Cálculo diferencial e integral	5	2023-1	2	2
3	3	Literatura Universal	Análisis de obras clásicas	3	2023-2	3	3

223 **SELECT * FROM ProgramasCursos;**

224

Data Output Messages Notifications

	idprogramacurso [PK] integer	idprograma integer	idcurso integer
1	1	1	1
2	2	2	2
3	3	3	3

225 **SELECT * FROM Horarios;**

226

Data Output Messages Notifications

	idhorario [PK] integer	idcurso integer	idaula integer	fechainicio date	fechafin date	horainicio time without time zone	horafin time without time zone
1	1	1	1	2023-01-10	2023-05-20	08:00:00	10:00:00
2	2	2	2	2023-01-10	2023-05-20	14:00:00	16:00:00
3	3	3	3	2023-06-05	2023-11-15	10:00:00	12:00:00

227 **SELECT** * **FROM** CursosProfesores;

228

Data Output Messages Notifications

	idcursoprofesor [PK] integer	idcurso integer	idprofesor integer
1	1	1	1
2	2	2	2
3	3	3	3

229 **SELECT** * **FROM** Inscripciones;

230

Data Output Messages Notifications

	idinscripcion [PK] integer	idestudiante integer	idcurso integer	fechainscripcion date	calificacion numeric (3,1)
1	1	1	1	2023-01-05	9.5
2	2	2	2	2023-01-05	8.0
3	3	3	3	2023-06-01	[null]

231 **SELECT** * **FROM** EstudiantesCarreras;

232

Data Output Messages Notifications

	idestudiantecarrera [PK] integer	idestudiante integer	idcarrera integer
1	1	1	1
2	2	2	2
3	3	3	3

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

```
233 --ACTUALIZAR DATOS
234 --CAMPUS
235 v UPDATE Campus SET
236     NombreCampus = 'Campus Principal',
237     DireccionCampus = 'Av. Central 1500'
238 WHERE IDCampus = 1;
239
240 v UPDATE Campus SET
241     NombreCampus = 'Campus Zona Norte',
242     DireccionCampus = 'Calle del Conocimiento 250'
243 WHERE IDCampus = 2;
244
245 v UPDATE Campus SET
246     NombreCampus = 'Campus Zona Sur',
247     DireccionCampus = 'Boulevard Universitario 350'
248 WHERE IDCampus = 3;
249
250 --Departamentos
251 v UPDATE Departamentos SET
252     NombreDepartamento = 'Ingeniería y Tecnología',
253     Edificio = 'Edificio Tecnológico'
254 WHERE IDDepartamento = 1;
255
256 v UPDATE Departamentos SET
257     NombreDepartamento = 'Ciencias Exactas',
258     Edificio = 'Edificio Científico'
259 WHERE IDDepartamento = 2;
260
261 v UPDATE Departamentos SET
262     NombreDepartamento = 'Humanidades y Artes',
263     Edificio = 'Edificio Cultural'
264 WHERE IDDepartamento = 3;
265
266 --Carreras
267 v UPDATE Carreras SET
268     NombreCarrera = 'Ingeniería en Desarrollo de Software',
269     TituloOtorgado = 'Ingeniero en Software'
270 WHERE IDCarrera = 1;
271
272 v UPDATE Carreras SET
273     NombreCarrera = 'Matemáticas Aplicadas',
274     TituloOtorgado = 'Licenciado en Matemáticas'
275 WHERE IDCarrera = 2;
276
277 v UPDATE Carreras SET
278     NombreCarrera = 'Literatura Comparada',
279     TituloOtorgado = 'Licenciado en Literatura'
280 WHERE IDCarrera = 3;
```

```

282 --Profesores
283 v UPDATE Profesores SET
284     Nombre = 'Ana María',
285     Email = 'anamaria.garcia@uni.edu'
286 WHERE IDProfesor = 1;
287
288 v UPDATE Profesores SET
289     Titulo = 'PhD',
290     Email = 'carlos.lopez.m@uni.edu'
291 WHERE IDProfesor = 2;
292
293 v UPDATE Profesores SET
294     Apellido = 'Rodríguez-Gómez',
295     Email = 'mrodriguez@uni.edu'
296 WHERE IDProfesor = 3;
297
298 --Estudiantes
299 v UPDATE Estudiantes SET
300     FechaNacimiento = '2000-05-20',
301     Email = 'juan.perez.2020@est.uni.edu',
302     Direccion = 'Av. del Estudiante 456' -- Actualización válida
303 WHERE IDEstudiante = 1;
304
305 v UPDATE Estudiantes SET
306     Direccion = 'Calle Estudiante 123' -- Solo dirección (sin Ciudad)
307 WHERE IDEstudiante = 2;
308
309 v UPDATE Estudiantes SET
310     Nombre = 'Pedro Antonio',
311     Email = 'pedro.sanchez.2019@est.uni.edu',
312     Direccion = 'Boulevard Universitario 789' -- Actualización válida
313 WHERE IDEstudiante = 3;
314
315
316 --ProgramasEstudio
317 v UPDATE ProgramasEstudio SET
318     DescripcionPrograma = 'Programa integral de ingenierías con enfoque en tecnologías modernas'
319 WHERE IDPrograma = 1;
320
321 v UPDATE ProgramasEstudio SET
322     NombrePrograma = 'Programa de Ciencias Básicas y Aplicadas'
323 WHERE IDPrograma = 2;
324
325 v UPDATE ProgramasEstudio SET
326     DescripcionPrograma = 'Estudios avanzados en humanidades, arte y cultura contemporánea'
327 WHERE IDPrograma = 3;
328

```

```

329 --Cursos
330 v UPDATE Cursos SET
331     Creditos = 5,
332     Semestre = '2024-1'
333 WHERE IDCurso = 1;
334
335 v UPDATE Cursos SET
336     NombreCurso = 'Cálculo Diferencial',
337     Descripcion = 'Fundamentos del cálculo diferencial con aplicaciones'
338 WHERE IDCurso = 2;
339
340 v UPDATE Cursos SET
341     Semestre = '2024-2',
342     IDCampus = 1
343 WHERE IDCurso = 3;
344
345 --Horarios
346 v UPDATE Horarios SET
347     HoraInicio = '08:30:00',
348     HoraFin = '10:30:00'
349 WHERE IDHorario = 1;
350
351 v UPDATE Horarios SET
352     FechaInicio = '2024-01-15',
353     FechaFin = '2024-05-25'
354 WHERE IDHorario = 2;
355
356 v UPDATE Horarios SET
357     IDAula = 5
358 WHERE IDHorario = 3;
359
360 --Inscripciones
361 v UPDATE Inscripciones SET
362     Calificacion = 9.8
363 WHERE IDInscripcion = 1;
364
365 v UPDATE Inscripciones SET
366     FechaInscripcion = '2023-01-10'
367 WHERE IDInscripcion = 2;
368
369 v UPDATE Inscripciones SET
370     Calificacion = 7.5
371 WHERE IDInscripcion = 3;

```

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

```
373 --ELIMINAR DATOS
374 -- 1. Eliminar inscripción
375 DELETE FROM Inscripciones WHERE IDInscripcion = 1;
376
377 -- 2. Eliminar relación estudiante-carrera
378 DELETE FROM EstudiantesCarreras WHERE IDEstudianteCarrera = 1;
379
380 -- 3. Eliminar relación curso-profesor
381 DELETE FROM CursosProfesores WHERE IDCursoProfesor = 1;
382
383 -- 4. Eliminar horario
384 DELETE FROM Horarios WHERE IDHorario = 1;
385
386 -- 5. Eliminar relación programa-curso
387 DELETE FROM ProgramasCursos WHERE IDProgramaCurso = 1;
388
389 -- 6. Eliminar curso
390 DELETE FROM Cursos WHERE IDCurso = 3;
391
392 -- 7. Eliminar estudiante
393 DELETE FROM Estudiantes WHERE IDEstudiante = 1;
394
395 -- 8. Eliminar profesor
396 DELETE FROM Profesores WHERE IDProfesor = 3;
397
398 -- 9. Eliminar programa de estudio
399 DELETE FROM ProgramasEstudio WHERE IDPrograma = 3;
400
401 -- 10. Eliminar carrera
402 DELETE FROM Carreras WHERE IDCarrera = 1;
403
404 -- 11. Eliminar departamento
405 DELETE FROM Departamentos WHERE IDDepartamento = 3;
406
407 -- 12. Eliminar campus
408 DELETE FROM Campus WHERE IDCampus = 3;
```

Comprobación

```
411 --COMPROBACION 2
412 SELECT * FROM Inscripciones;
413
```

Data Output Messages Notifications

	idinscripcion [PK] integer	idestudiante integer	idcurso integer	fechainscripcion date	calificacion numeric (3,1)
1	2	2	2	2023-01-10	8.0

```
414 SELECT * FROM EstudiantesCarreras;
415
```

Data Output Messages Notifications

	idestudiantecarrera [PK] integer	idestudiante integer	idcarrera integer
1	2	2	2
2	3	3	3

```
416 SELECT * FROM CursosProfesores;
417
```

Data Output Messages Notifications

	idcursoprofesor [PK] integer	idcurso integer	idprofesor integer
1	2	2	2

```
418 SELECT * FROM Horarios;
419
```

Data Output Messages Notifications

	idhorario [PK] integer	idcurso integer	idaula integer	fechainicio date	fechafin date	horainicio time without time zone	horafin time without time zone
1	2	2	2	2024-01-15	2024-05-...	14:00:00	16:00:00

```
420 SELECT * FROM ProgramasCursos;
421
```

Data Output Messages Notifications

	idprogramacurso [PK] integer	idprograma integer	idcurso integer
1	2	2	2

```
422 SELECT * FROM Cursos;
423
```

Data Output Messages Notifications

	idcurso [PK] integer	nombrecurso character varying (100)	descripcion text	creditos smallint	semestre character varying (20)	iddepartamento integer	idcampus integer
1	1	Programación I	Fundamentos de programación en Python	5	2024-1	1	1
2	2	Cálculo Diferencial	Fundamentos del cálculo diferencial con aplicacion...	5	2023-1	2	2

```
424 SELECT * FROM Estudiantes;
425
```

Data Output Messages Notifications

	idestudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fechanacimiento date	direccion character varying (100)	email character varying (100)	idcarrera integer
1	2	Laura	Gómez	2001-03-22	Calle Estudiante 123	laura.gomez@est.uni.edu	[null]
2	3	Pedro Antonio	Sánchez	1999-11-10	Boulevard Universitario 7...	pedro.sanchez.2019@est.uni.e...	[null]

```
426 SELECT * FROM Profesores;
427
```

Data Output Messages Notifications

	idprofesor [PK] integer	nombre character varying (50)	apellido character varying (50)	titulo character varying (50)	iddepartamento integer	email character varying (100)
1	1	Ana María	García	PhD	1	anamaria.garcia@uni.edu
2	2	Carlos	López	PhD	2	carlos.lopez.m@uni.edu

```
428 SELECT * FROM ProgramasEstudio;
429
```

Data Output Messages Notifications

	idprograma [PK] integer	nombreprograma character varying (100)	descripcionprograma text
1	1	Programa de Ingeniería	Programa integral de ingenierías con enfoque en tecnologías modern...
2	2	Programa de Ciencias Básicas y Aplicad...	Abarca matemáticas y física avanzada

430 **SELECT * FROM Carreras;**
431

Data Output Messages Notifications

	idcarrera [PK] integer	nombrecarrera character varying (100)	titulootorgado character varying (100)
1	2	Matemáticas Aplicadas	Licenciado en Matemátic...
2	3	Literatura Comparada	Licenciado en Literatura

432 **SELECT * FROM Departamentos;**
433

Data Output Messages Notifications

	iddepartamento [PK] integer	nombredepartamento character varying (100)	edificio character varying (50)
1	1	Ingeniería y Tecnología	Edificio Tecnológico
2	2	Ciencias Exactas	Edificio Científico

434 **SELECT * FROM Campus;**
435
436

Data Output Messages Notifications

	idcampus [PK] integer	nombrecampus character varying (100)	direccioncampus character varying (200)
1	1	Campus Principal	Av. Central 1500
2	2	Campus Zona Norte	Calle del Conocimiento 2...

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.

```
439 --Consulta 1
440 SELECT Nombre, Apellido
441 FROM Estudiantes;
```

Data Output Messages Notifications

	nombre character varying (50)	apellido character varying (50)
1	Laura	Gómez
2	Pedro Antonio	Sánchez

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

```
443 --CONSULTA 2
444 SELECT * FROM Cursos
445 WHERE Creditos = 3;
```

Data Output Messages Notifications

	idcurso [PK] integer	nombrecurso character varying (100)	descripcion text	creditos smallint	semestre character varying (20)	iddepartamento integer	idcampus integer
--	-------------------------	--	---------------------	----------------------	------------------------------------	---------------------------	---------------------

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

```
447 --Consulta 3
448 SELECT
449     E.Nombre AS NombreEstudiante,
450     E.Apellido AS ApellidoEstudiante,
451     C.NombreCurso AS CursoInscrito
452 FROM
453     Estudiantes E
454 INNER JOIN
455     Inscripciones I ON E.IDEstudiante = I.IDEstudiante
456 INNER JOIN
457     Cursos C ON I.IDCurso = C.IDCurso;
```

Data Output Messages Notifications

	nombrestudiante character varying (50)	apellidostudiante character varying (50)	cursoinscrito character varying (100)
1	Laura	Gómez	Cálculo Diferencial

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).

```

459 --Consulta 4
460 SELECT
461     E.Nombre AS NombreEstudiante,
462     E.Apellido AS ApellidoEstudiante,
463     COALESCE(C.NombreCurso, 'No inscrito') AS Curso
464 FROM
465     Estudiantes E
466 LEFT JOIN
467     Inscripciones I ON E.IDEstudiante = I.IDEstudiante
468 LEFT JOIN
469     Cursos C ON I.IDCurso = C.IDCurso;
470

```

Data Output Messages Notifications

	nombreestudiante character varying (50)	apellidoestudiante character varying (50)	curso character varying
1	Laura	Gómez	Cálculo Diferencial
2	Pedro Antonio	Sánchez	No inscrito

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.

```

471 --Consulta 5
472 SELECT
473     C.NombreCurso AS Curso,
474     COALESCE(E.Nombre || ' ' || E.Apellido, 'Sin estudiantes inscritos') AS Estudiante
475 FROM
476     Estudiantes E
477 RIGHT JOIN
478     Inscripciones I ON E.IDEstudiante = I.IDEstudiante
479 RIGHT JOIN
480     Cursos C ON I.IDCurso = C.IDCurso
481 ORDER BY
482     C.NombreCurso;

```

Data Output Messages Notifications

	curso character varying (100)	estudiante text
1	Cálculo Diferencial	Laura Gómez
2	Programación I	Sin estudiantes inscrit...

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

```
484 --Consulta 6
485 SELECT
486     C.NombreCurso AS Curso,
487     COUNT(I.IDEstudiante) AS Cantidad_Estudiantes
488 FROM
489     Cursos C
490 LEFT JOIN
491     Inscripciones I ON C.IDCurso = I.IDCurso
492 GROUP BY
493     C.NombreCurso
494 ORDER BY
495     Cantidad_Estudiantes DESC;
```

Data Output Messages Notifications

	curso character varying (100)	cantidad_estudiantes bigint
1	Cálculo Diferencial	1
2	Programación I	0

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

```
498 SELECT
499     IDEstudiante,
500     Nombre,
501     Apellido,
502     FechaNacimiento,
503     EXTRACT(YEAR FROM FechaNacimiento) AS AñoNacimiento
504 FROM
505     Estudiantes
506 WHERE
507     FechaNacimiento BETWEEN '1995-01-01' AND '1998-12-31'
508 ORDER BY
509     FechaNacimiento;
```

Data Output Messages Notifications

idestudiante [PK] integer	nombre character varying (50)	apellido character varying (50)	fechanacimiento date	añonacimiento numeric
------------------------------	----------------------------------	------------------------------------	-------------------------	--------------------------

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

```
511  --Consulta 8
512  SELECT
513      IDCurso,
514      NombreCurso AS Curso,
515      Creditos,
516      Semestre
517  FROM
518      Cursos
519  ORDER BY
520      NombreCurso ASC;
```

Data Output Messages Notifications

	idcurso [PK] integer	curso character varying (100)	creditos smallint	semestre character varying (20)
1	2	Cálculo Diferencial	5	2023-1
2	1	Programación I	5	2024-1

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.

```
522 --Consulta 9
523 WITH InscripcionesPorEstudiante AS (
524     SELECT
525         E.IDEstudiante,
526         E.Nombre || ' ' || E.Apellido AS NombreCompleto,
527         COUNT(I.IDInscripcion) AS TotalInscripciones
528     FROM
529         Estudiantes E
530     LEFT JOIN
531         Inscripciones I ON E.IDEstudiante = I.IDEstudiante
532     GROUP BY
533         E.IDEstudiante, E.Nombre, E.Apellido
534 )
535 SELECT
536     NombreCompleto,
537     TotalInscripciones
538 FROM
539     InscripcionesPorEstudiante
540 ORDER BY
541     TotalInscripciones DESC
542 LIMIT 3;
```

Data Output Messages Notifications

	nombrecompleto text	totalinscripciones bigint
1	Laura Gómez	1
2	Pedro Antonio Sánch...	0

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

```

544 --Consulta 10
545 WITH CursosPorDepartamento AS (
546     SELECT
547         D.IDDepartamento,
548         D.NombreDepartamento,
549         C.IDCurso,
550         C.NombreCurso,
551         COUNT(I.IDInscripcion) AS TotalInscritos,
552         RANK() OVER (PARTITION BY D.IDDepartamento ORDER BY COUNT(I.IDInscripcion) DESC) AS Ranking
553     FROM
554         Departamentos D
555     INNER JOIN
556         Cursos C ON D.IDDepartamento = C.IDDepartamento
557     LEFT JOIN
558         Inscripciones I ON C.IDCurso = I.IDCurso
559     GROUP BY
560         D.IDDepartamento, D.NombreDepartamento, C.IDCurso, C.NombreCurso
561 )
562 SELECT
563     NombreDepartamento AS Departamento,
564     NombreCurso AS Curso_Mas_Popular,
565     TotalInscritos AS Estudiantes_Inscritos
566 FROM
567     CursosPorDepartamento
568 WHERE
569     Ranking = 1
570 ORDER BY
571     TotalInscritos DESC;

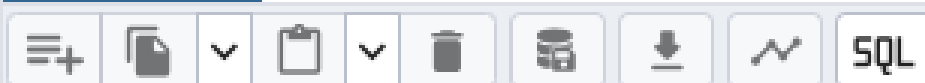
```

Data Output Messages Notifications			
	departamento character varying (100)	curso_mas_popular character varying (100)	estudiantes_inscritos bigint
1	Ciencias Exactas	Cálculo Diferencial	1
2	Ingeniería y Tecnología	Programación I	0

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.

```
571 --Consultas Complejas
572 --Consulta 11
573 SELECT
574     P.Nombre,
575     P.Apellido,
576     COUNT(CP.IDCurso) AS Cantidad_Cursos
577 FROM
578     Profesores P
579 INNER JOIN
580     CursosProfesores CP ON P.IDProfesor = CP.IDProfesor
581 GROUP BY
582     P.IDProfesor, P.Nombre, P.Apellido
583 HAVING
584     COUNT(CP.IDCurso) > 2
585 ORDER BY
586     Cantidad_Cursos DESC;
```

Data Output Messages Notifications



nombre
character varying (50)

apellido
character varying (50)

cantidad_cursos
bigint

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.

```

588 --Consulta 12
589 WITH PromediosPorCurso AS (
590     SELECT
591         PC.IDPrograma,
592         C.IDCurso,
593         C.NombreCurso,
594         ROUND(AVG(I.Calificacion), 2) AS PromedioCalificacion,
595         RANK() OVER (PARTITION BY PC.IDPrograma ORDER BY AVG(I.Calificacion) DESC) AS Ranking
596     FROM
597         ProgramasCursos PC
598     INNER JOIN
599         Cursos C ON PC.IDCurso = C.IDCurso
600     LEFT JOIN
601         Inscripciones I ON C.IDCurso = I.IDCurso
602     GROUP BY
603         PC.IDPrograma, C.IDCurso, C.NombreCurso
604     HAVING
605         AVG(I.Calificacion) IS NOT NULL
606 )
607 SELECT
608     PE.NombrePrograma AS ProgramaEstudio,
609     PPC.NombreCurso AS CursoMejorCalificado,
610     PPC.PromedioCalificacion
611 FROM
612     PromediosPorCurso PPC
613 INNER JOIN
614     ProgramasEstudio PE ON PPC.IDPrograma = PE.IDPrograma
615 WHERE
616     PPC.Ranking = 1
617 ORDER BY
618     PPC.PromedioCalificacion DESC;

```

Data Output			Messages	Notifications
	programaestudio character varying (100)	cursomejorcalificado character varying (100)	promediocalificacion numeric	
1	Programa de Ciencias Básicas y Aplicad...	Cálculo Diferencial	8.00	