



**TECNOLÓGICO
NACIONAL DE MÉXICO**

DONOVAN ALEJANDRO HERNANDEZ HERNANDEZ

NO. CONTRoL 23490557

FUNDAMENTOS DE BASE DE DATOS

ING. JOSE RAMON BOGARIN VALENZUELA

EXAMEN FINAL UNIDAD 4

Gestionando información de una Universidad/Instituto.

Planteamiento del problema: 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. La tarea es diseñar y trabajar con la base de datos que almacenará toda esta información.

Desarrollo y solución a la base de datos.

Esquema de Base de Datos.

Creación de Tablas

SQL - Historial de Consultas

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

```
1 CREATE TABLE Departamentos (  
2   IDDepartamento SERIAL PRIMARY KEY,  
3   NombreDepartamento VARCHAR(100) NOT NULL,  
4   Edificio VARCHAR(100)  
5 );
```

```
CREATE TABLE Cursos (  
    IDCurso SERIAL PRIMARY KEY,  
    NombreCurso VARCHAR(100) NOT NULL,  
    Descripcion TEXT,  
    Creditos INTEGER,  
    Semestre VARCHAR(20),  
    IDDepartamento INTEGER REFERENCES Departamentos(IDDepartamento) ON DELETE SET NULL  
);
```

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

```
CREATE TABLE Inscripciones (  
    IDInscripcion SERIAL PRIMARY KEY,  
    IDEstudiante INTEGER REFERENCES Estudiantes(IDEstudiante) ON DELETE CASCADE,  
    IDCurso INTEGER REFERENCES Cursos(IDCurso) ON DELETE CASCADE,  
    FechaInscripcion DATE,  
    Calificacion NUMERIC(4,2)  
);
```

```
CREATE TABLE Aulas (  
    IDAula SERIAL PRIMARY KEY,  
    NombreAula VARCHAR(50) NOT NULL,  
    Capacidad INTEGER,  
    Ubicacion VARCHAR(100)  
);
```

```
CREATE TABLE Horarios (  
    IDHorario SERIAL PRIMARY KEY,  
    IDCurso INTEGER REFERENCES Cursos(IDCurso) ON DELETE CASCADE,  
    IDAula INTEGER REFERENCES Aulas(IDAula) ON DELETE SET NULL,  
    FechaInicio DATE,  
    FechaFin DATE,  
    HoraInicio TIME,  
    HoraFin TIME  
);
```

Tablas de Relacion

```
CREATE TABLE CursosProfesores (  
    IDCursoProfesor SERIAL PRIMARY KEY,  
    IDCurso INTEGER REFERENCES Cursos(IDCurso) ON DELETE CASCADE,  
    IDProfesor INTEGER REFERENCES Profesores(IDProfesor) ON DELETE CASCADE  
);
```

```
CREATE TABLE ProgramasEstudio (  
    IDPrograma SERIAL PRIMARY KEY,  
    NombrePrograma VARCHAR(100) NOT NULL,  
    DescripcionPrograma TEXT  
);
```

```
CREATE TABLE ProgramasCursos (  
    IDProgramaCurso SERIAL PRIMARY KEY,  
    IDPrograma INTEGER REFERENCES ProgramasEstudio(IDPrograma) ON DELETE CASCADE,  
    IDCurso INTEGER REFERENCES Cursos(IDCurso) ON DELETE CASCADE  
);
```

Creacion de Tablas extra: Campus y Carreras

```
CREATE TABLE Campus (  
    IDCampus SERIAL PRIMARY KEY,  
    NombreCampus VARCHAR(100) NOT NULL,  
    DireccionCampus VARCHAR(200)  
);
```

```
CREATE TABLE Carreras (  
    IDCarrera SERIAL PRIMARY KEY,  
    NombreCarrera VARCHAR(100) NOT NULL,  
    TituloOtorgado VARCHAR(100)  
);
```

```
CREATE TABLE EstudiantesCarreras (  
    IDEstudiante INTEGER REFERENCES Estudiantes(IDEstudiante) ON DELETE CASCADE,  
    IDCarrera INTEGER REFERENCES Carreras(IDCarrera) ON DELETE CASCADE,  
    PRIMARY KEY (IDEstudiante, IDCarrera)  
);
```

Modificacion de Tablas

Agregando IDCarrera referenciando a Carrera.

```
ALTER TABLE Estudiantes  
ADD COLUMN IDCarrera INTEGER REFERENCES Carreras(IDCarrera);
```

```
ALTER TABLE Cursos  
ADD COLUMN IDCampus INTEGER REFERENCES Campus(IDCampus);
```

Agregando ID de campus para Campus.

```
ALTER TABLE Profesores  
ADD COLUMN Email VARCHAR(100) UNIQUE;
```

Agregando sección de Email que es único.

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

Eliminando Ciudad de la tabla de Ciudad.

```
ALTER TABLE Horarios  
DROP CONSTRAINT IF EXISTS horarios_idaula_fkey;
```

Eliminando clave foránea a aula en Horarios, para posteriormente eliminar la tabla Aulas.

```
ALTER TABLE Horarios  
DROP COLUMN IF EXISTS IDAula;
```

Eliminamos también IDAula.

```
DROP TABLE AULAS;
```

Finalmente eliminamos Aulas.

Inserción Datos, Actualización, Eliminación y Consultas de Búsqueda.

Datos Campus

```
INSERT INTO Campus (NombreCampus, DireccionCampus)
VALUES ('Campus Mexicali', 'Calle Instituto Tecnológico, Mexicali'),
       ('Campus Tijuana', 'Calle Río Nuevo, Tijuana');
```

Carreras

```
INSERT INTO Carreras (NombreCarrera, TituloOtorgado)
VALUES ('Ingeniería en Sistemas', 'Titulado en Ingeniería de Sistemas'),
       ('Ingeniería en Mecatrónica', 'Titulado en Mecatrónica'),
       ('Ingeniería Civil', 'Titulado en Ing. Civil');
```

Estudiantes

```
INSERT INTO Estudiantes (Nombre, Apellido, FechaNacimiento, Direccion, Email, IDCarrera)
VALUES
('Donovan', 'Hernandez', '2002-09-19', 'Cerritos, Mexicali', 'donovan.hernandez@gmail.com', 1),
('Damian', 'Lugo', '2005-06-22', 'Av. Real 456, Mexicali', 'damian@gmail.com', 2),
('Alonso', 'Guevara', '2004-11-10', 'Calle Libertad 789, Mexicali', 'alonso@gmail.com', 3),
('Juanito', 'Alcachofa', '1995-02-01', 'Calle Jardín 101, Tijuana', 'juanito@gmail.com', 2),
('Emanuel', 'Padilla', '1999-07-30', 'Calle Sol 202, Mexicali', 'emanuel@gmail.com', 1),
('Miguel', 'Hidalgo', '1995-05-25', 'Calle Historia 303, Tijuana', 'miguel@gmail.com', 3);
```

Profesores

```
INSERT INTO Profesores (Nombre, Apellido, Titulo, IDDepartamento)
VALUES
('Jose', 'Bogarin', 'Doctor en Matemáticas', 1),
('Ana', 'Lepe', 'Licenciada en Ciencias de la Computación', 3),
('Luis', 'Alvarado', 'Doctor en Filosofía', 2),
('Daniel', 'Larusso', 'Licenciada en Literatura', 4);
```


Cursos Profesores – Tabla de Relacion

```
INSERT INTO CursosProfesores (IDCurso, IDProfesor)
VALUES
(1, 1),
(3, 1),
(2, 2),
(4, 2);
```

Departamento

```
INSERT INTO Departamentos (NombreDepartamento, Edificio)
VALUES
('Matemáticas', 'Edificio A'),
('Filosofía', 'Edificio B'),
('Ciencias de la Computación', 'Edificio C'),
('Humanidades', 'Edificio D');
```

Cursos

```
1  INSERT INTO Cursos (IDCurso, NombreCurso, Descripcion, Creditos, Semestre, IDDepartame
2  VALUES
3  (1, 'Ecuaciones', 'Curso de Ecuaciones Diferenciales', 3, 'Primero', 1),
4  (2, 'Ética', 'Curso de Ética Profesional', 3, 'Segundo', 2),
5  (3, 'Programación', 'Curso de Programación en C', 3, 'Tercero', 3),
6  (4, 'Lectura', 'Curso de Lectura Crítica y Comprensión', 3, 'Cuarto', 4);
```

Data Output Mensajes Notificaciones

INSERT 0 4

Consulta retornó exitosamente en 175 msec.

Inscripciones

```
INSERT INTO Inscripciones (IDEstudiante, IDCurso, FechaInscripcion, Calificacion)
VALUES
(1, 1, '2023-08-01', 9.5),
(1, 2, '2023-08-01', 8.0),
(2, 3, '2023-08-01', 7.5),
(4, 1, '2023-08-01', 8.5),
(5, 2, '2023-08-01', 9.0),
(6, 3, '2023-08-01', 6.5);
```

Programa Estudio

```
Consulta Historial de Consultas
1 INSERT INTO ProgramasEstudio (NombrePrograma, DescripcionPrograma)
2 VALUES
3 ('Ingeniería de Sistemas', 'Programa de estudios orientado a la tecnología y la informática'),
4 ('Ingeniería Mecatrónica', 'Programa de estudios interdisciplinario entre la ingeniería mecánica y electrónica'),
5 ('Ingeniería Civil', 'Programa de estudios enfocado en la construcción y el diseño de infraestructuras'),
6 ('Ciencias Sociales', 'Programa de estudios centrado en la filosofía, sociología y psicología');
```

Programa Cursos

```
INSERT INTO ProgramasCursos (IDPrograma, IDCurso)
VALUES
(1, 1),
(1, 3),
(2, 1),
(2, 4),
(3, 2),
(3, 4),
(4, 2),
(4, 4);
```

Consultas

Actualización correo

```
UPDATE Estudiantes
SET Email = 'donovan,hdez@gmail.com'
WHERE IDEstudiente = 1;
```

Cambio de carrera

```
UPDATE Estudiantes  
SET IDCarrera = 1  
WHERE IDEstudiente = 4;
```

Eliminar estudiante

```
DELETE FROM Estudiantes  
WHERE IDEstudiente = 3;
```

Consulta Búsqueda (Post-Modificaciones)

```
1  SELECT E.Nombre, E.Apellido, C.NombreCarrera
2  FROM Estudiantes E
3  JOIN Carreras C ON E.IDCarrera = C.IDCarrera;
```

Data Output Mensajes Notificaciones



	nombre character varying (50) 🔒	apellido character varying (50) 🔒	nombrecarrera character varying (100) 🔒
1	Damian	Lugo	Ingenieria en Mecatronica
2	Emanuel	Padilla	Ingeniería en Sistemas
3	Miguel	Hidalgo	Ingenieria Civil
4	Donovan	Hernandez	Ingeniería en Sistemas
5	Juanito	Alcachofa	Ingeniería en Sistemas

Consultas Especoificas

Mostrar nombre de estudiantes y sus apellidos.

Consulta

Historial de Consultas

1

2

SELECT

Nombre,

Apellido

FROM

Estudiantes;

Data Output

Mensajes

Notificaciones

+

▼

▼

SQL

	<div>nombre</div> <div>character varying (50)</div>	<div>apellido</div> <div>character varying (50)</div>
1	Damian	Lugo
2	Emanuel	Padilla
3	Miguel	Hidalgo
4	Donovan	Hernandez
5	Juanito	Alcachofa

Cursos con 3 creditos (todos).

Consulta

Historial de Consultas

1

2

3

SELECT

NombreCurso

FROM

Cursos

WHERE

Creditos

=

3

;

Data Output

Mensajes

Notificaciones

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	<div>nombrecurso</div> <div>character varying (100)</div> <div>🔒</div>
1	Ecuaciones
2	Ética
3	Programación
4	Lectura
5	Ecuaciones
6	Ética
7	Programación
8	Lectura

INNER JOIN Lista estudiantes y curso inscrito

Consulta

Historial de Consultas

1

2

3

4

SELECT

C.NombreCurso,

E.Nombre

AS

Estudiante,

E.Apellido

FROM

Cursos

C

RIGHT JOIN

Inscripciones

I

ON

C.IDCurso

=

I.IDCurso

RIGHT JOIN

Estudiantes

E

ON

I.IDEstudiante

=

E.IDEstudiante;

Data Output

Mensajes

Notificaciones

≡+

📄

▼

📋

▼

🗑️

🗑️

📥

📥

📈

SQL

	<div>nombrecurso</div> <div>character varying (100)</div>	<div>estudiante</div> <div>character varying (50)</div>	<div>apellido</div> <div>character varying (50)</div>
1	Ecuaciones	Donovan	Hernandez
2	Ética	Donovan	Hernandez
3	Programación	Damian	Lugo
4	Ecuaciones	Juanito	Alcachofa
5	Ética	Emanuel	Padilla
6	Programación	Miguel	Hidalgo

(También todos los estudiantes están inscritos a un curso.)

Todos los cursos tienen un estudiante.

Consulta

Historial de Consultas

1

2

3

4

SELECT

C.NombreCurso,

COUNT(I.IDInscripcion)

AS

NumeroDeEstudiantes

FROM

Cursos C

LEFT JOIN

Inscripciones I

ON

C.IDCurso = I.IDCurso

GROUP BY

C.IDCurso;

Data Output

Mensajes

Notificaciones

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	nombrecurso character varying (100) 🔒	numero de estudiantes bigint 🔒
1	Programación	2
2	Ecuaciones	0
3	Lectura	0
4	Ética	0
5	Ética	2
6	Programación	0
7	Ecuaciones	2
8	Lectura	0

Group By con Count. Muestra lista y calcula cuantos alumnos hay en cada curso.

Between para encontrar alumnos entre 1995 y 1998.

Consulta

Historial de Consultas

1

2

3

SELECT

Nombre,

Apellido

FROM

Estudiantes

WHERE

FechaNacimiento

BETWEEN

'1995-01-01'

AND

'1998-12-31';

Data Output

Mensajes

Notificaciones

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	<div>nombre</div> <div>character varying (50)</div> <div>🔒</div>	<div>apellido</div> <div>character varying (50)</div> <div>🔒</div>
1	Miguel	Hidalgo
2	Juanito	Alcachofa

Order By ordena por orden alfabetico los cursos y los muestra.

Consulta

Historial de Consultas

1

2

3

SELECT

NombreCurso

FROM

Cursos

ORDER BY

NombreCurso

ASC;

Data Output

Mensajes

Notificaciones

≡

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

nombrecurso

character varying (100)

🔒

1

Ecuaciones

2

Ecuaciones

3

Ética

4

Ética

5

Lectura

6

Lectura

7

Programación

8

Programación

Tabla de expresión común, lista numero de inscripciones y muestra quien tuvo mas en algún curso.

Consulta

Historial de Consultas

1

2

3

4

5

6

7

8

9

10

WITH

InscripcionesPorEstudiante

AS

(

SELECT

E.IDEstudiante,

E.Nombre,

E.Apellido,

COUNT(I.IDInscripcion)

FROM

Estudiantes

E

LEFT JOIN

Inscripciones

I

ON

E.IDEstudiante

=

I.IDEstudiante

GROUP BY

E.IDEstudiante

)

SELECT

Nombre,

Apellido,

NumeroDeInscripciones

FROM

InscripcionesPorEstudiante

ORDER BY

NumeroDeInscripciones

DESC

LIMIT

3;

Data Output

Mensajes

Notificaciones

≡

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	nombre character varying (50) 🔒	apellido character varying (50) 🔒	numerosdeinscripciones bigint 🔒
1	Donovan	Hernandez	2
2	Juanito	Alcachofa	1
3	Miguel	Hidalgo	1

Consulta compleja 1: Mostrar nombre de departamentos, y mayor cantidad de alumnos inscritos.

```

1  WITH CursosConInscripciones AS (
2      SELECT C.IDCurso, C.NombreCurso, C.IDDepartamento, COUNT(I.IDInscripcion) AS NumeroDeInscripciones
3      FROM Cursos C
4      LEFT JOIN Incripciones I ON C.IDCurso = I.IDCurso
5      GROUP BY C.IDCurso
6  )
7  SELECT D.NombreDepartamento, C.NombreCurso
8  FROM CursosConInscripciones C
9  JOIN Departamentos D ON C.IDDepartamento = D.IDDepartamento
10 WHERE C.NumeroDeInscripciones = (
11     SELECT MAX(NumeroDeInscripciones)
12     FROM CursosConInscripciones
13     WHERE IDDepartamento = C.IDDepartamento
14 );

```

Data Output Mensajes Notificaciones

	nombredepartamento character varying (100)	nombrecurso character varying (100)
1	Ciencias de la Computación	Programación
2	Humanidades	Lectura
3	Filosofía	Ética
4	Matemáticas	Ecuaciones
5	Humanidades	Lectura

Consulta Compleja 2: Mostrando profesores que imparten mas cursos

Consulta

Historial de Consultas

1

2

3

4

5

SELECT

P.Nombre,

P.Apellido,

COUNT(CP.IDCurso)

AS

NumeroDeCursos

FROM

Profesores

P

JOIN

CursosProfesores

CP

ON

P.IDProfesor

=

CP.IDProfesor

GROUP

BY

P.IDProfesor

HAVING

COUNT(CP.IDCurso)

>

2;

Data Output

Mensajes

Notificaciones

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

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

Consulta compleja 3: Programas de estudio y cual fue el mas alto.

Consulta

Historial de Consultas

1

2

3

4

5

6

7

8

9

10

11

12

13

14

WITH PromedioCalificacion AS (
SELECT IC.IDCurso, AVG(IC.Calificacion) AS PromedioCalificacion
FROM Inscripciones IC
GROUP BY IC.IDCurso
)
SELECT P.NombrePrograma, C.NombreCurso
FROM PromedioCalificacion PC
JOIN Cursos C ON PC.IDCurso = C.IDCurso
JOIN ProgramasEstudio P ON C.IDDepartamento = P.IDPrograma
WHERE PC.PromedioCalificacion = (
SELECT MAX(PromedioCalificacion)
FROM PromedioCalificacion
WHERE IDCurso = PC.IDCurso
);

Data Output

Mensajes

Notificaciones

≡+

	<div>nombreprograma</div> <div>character varying (100)</div> <div>🔒</div>	<div>nombrecurso</div> <div>character varying (100)</div> <div>🔒</div>
1	Ingeniería Civil	Programación
2	Ingeniería Mecatrónica	Ética
3	Ingeniería de Sistemas	Ecuaciones