

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.

<u>Planteamiento del problema</u>: 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

```
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
);
```

```
CREATE TABLE Departamentos (
IDDepartamento SERIAL PRIMARY KEY,
NombreDepartamento VARCHAR(100) NOT NULL,
Edificio VARCHAR(100)
);
```

```
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
);
```

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)
);
```

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, Actualizacion, Eliminando y Consultas de Busqueda.

Datos Campus

Carreras

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

```
INSERT INTO Cursos (IDCurso, NombreCurso, Descripcion, Creditos, Semestre, IDDepartame
VALUES
(1, 'Ecuaciones', 'Curso de Ecuaciones Diferenciales', 3, 'Primero', 1),
(2, 'Ética', 'Curso de Ética Profesional', 3, 'Segundo', 2),
(3, 'Programación', 'Curso de Programación en C', 3, 'Tercero', 3),
(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 VINSERT 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 IDEstudiante = 1;
```

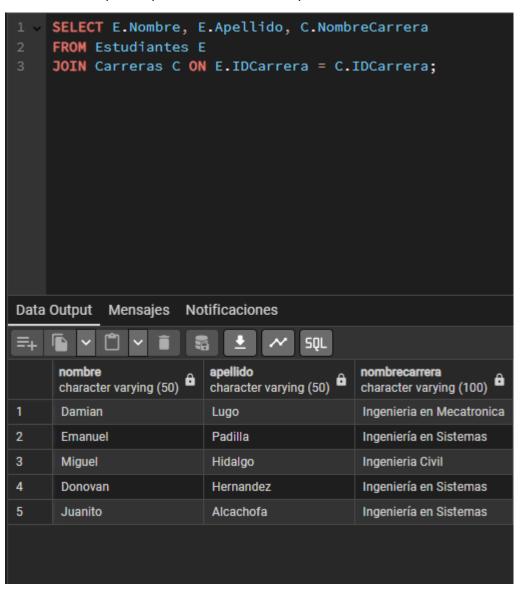
Cambio de carrera

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

Eliminar estudiante

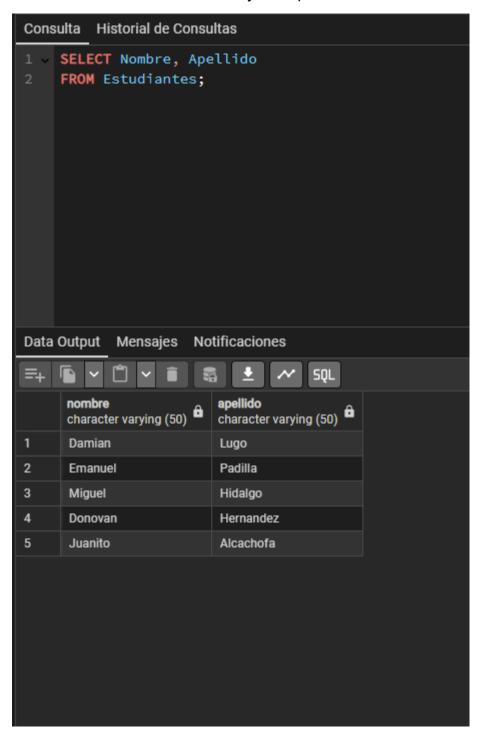
```
DELETE FROM Estudiantes
WHERE IDEstudiante = 3;
```

Consulta Busqueda (Post-Modificaciones)

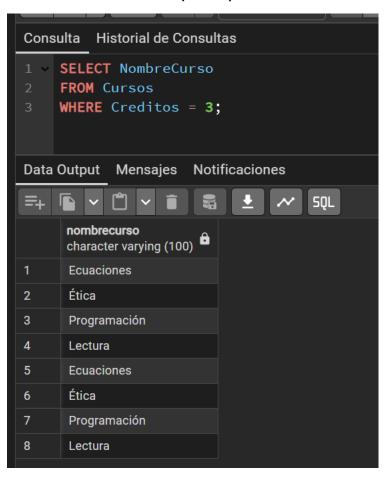


Consulyas Especoificas

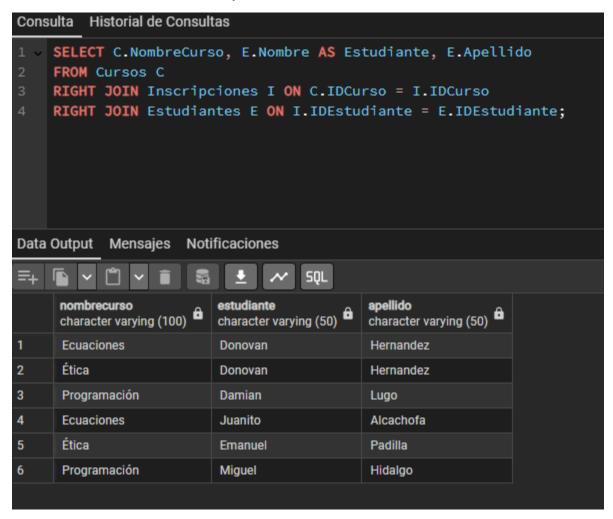
Muestrar nombre de estudiantes y sus apellidos.



Cursos con 3 creditos (todos).

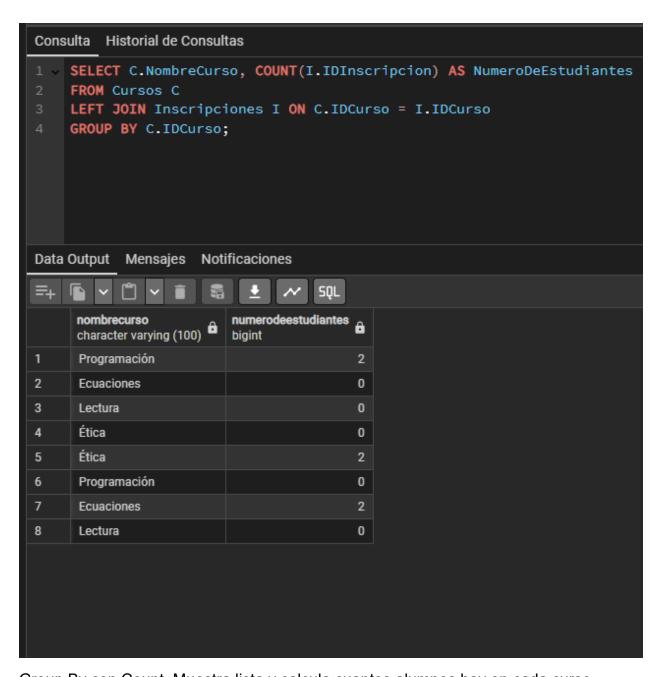


INNER JOIN Lista estudiantes y curso inscrito



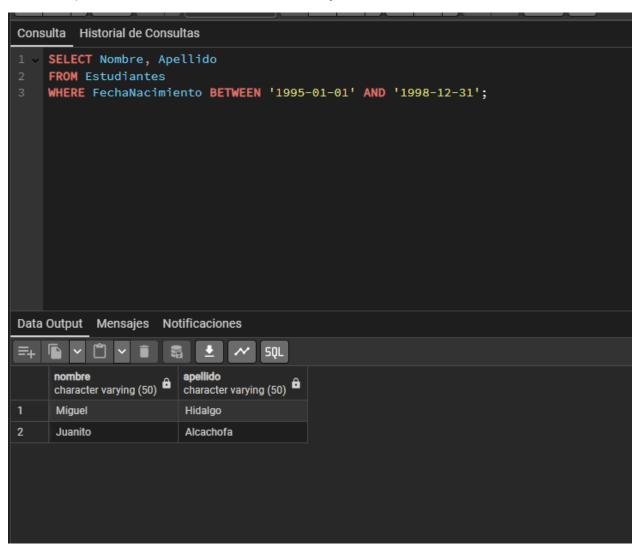
(También todos los estudiantes están inscritos a un curso.)

Todos los cursos tienen un estudiante.



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

Between para encontrar alumnos entre 1995 y 1998.



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

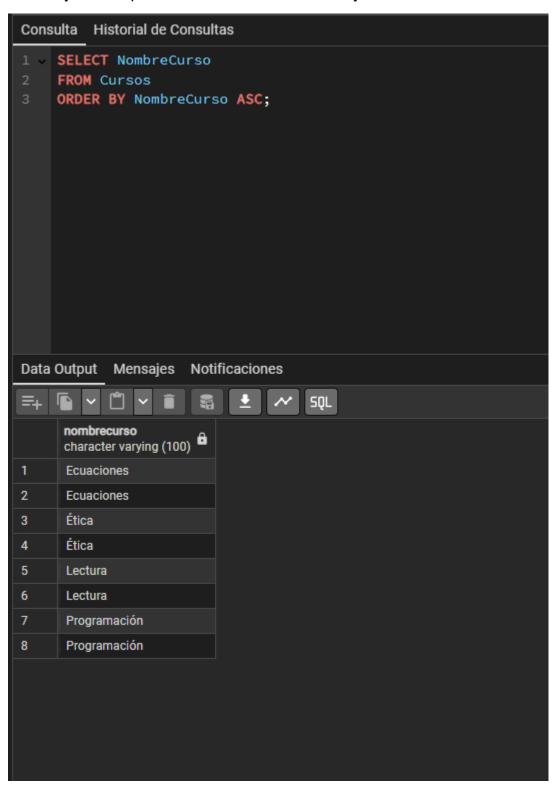
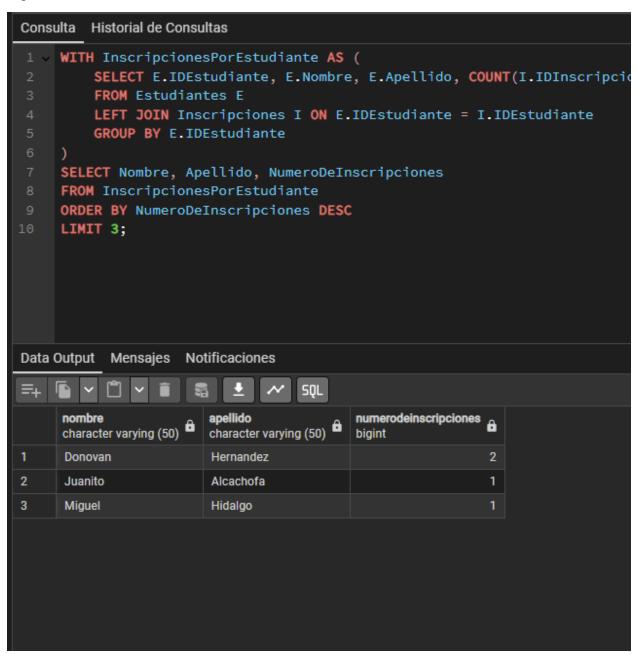
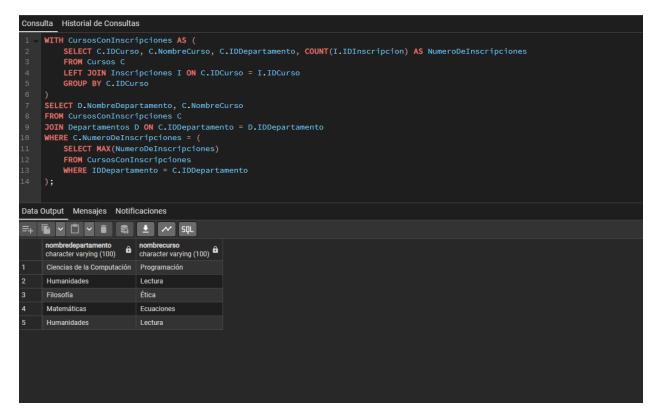


Tabla de expresión común, lista numero de inscripciones y muestra quien tuvo mas en algún curso.



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



Consulta Compleja 2: Mostrando profesores que imparten mas cursos

```
Consulta Historial de Consultas
    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
                                           numerodecursos
                        apellido
     character varying (50) character varying (50)
```

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

