



Instituto Tecnológico de Mexicali

Fundamento de base de datos

Unidad 3

Tarea 3

Luis Alonso Guevara Quiñonez - 23490377

-Docente-

Jose Ramon Bogarin Valenzuela

Ingeniería en sistemas.

```
CREATE TABLE Students (  
    student_id SERIAL PRIMARY KEY,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL  
);  
  
CREATE TABLE Courses (  
    course_id SERIAL PRIMARY KEY,  
    title varchar(100) NOT NULL,  
    credits INT NOT NULL CHECK (credits > 0)  
)  
  
CREATE TABLE Enrollments (  
    enrollment_id SERIAL PRIMARY KEY,  
    student_id INT REFERENCES Students(student_id) NOT NULL,  
    course_id INT REFERENCES Courses(course_id) NOT NULL,  
    enrolled_on DATE DEFAULT CURRENT_DATE,  
    UNIQUE (student_id, course_id)  
);
```

--INSERCIÓN DE DATOS

```
INSERT INTO Students (first_name, last_name, email)  
VALUES  
('Ana', 'Gómez', 'ana.gomez@example.com'),  
('Luis', 'Martínez', 'luis.martinez@example.com'),  
('Carla', 'Rojas', 'carla.rojas@example.com'),  
('Pedro', 'Fernández', 'pedro.fernandez@example.com'),  
('Lucía', 'Ramírez', 'lucia.ramirez@example.com');
```

```
INSERT INTO Courses (title, credits)  
VALUES
```

('Matemáticas I', 4),
('Historia Universal', 3),
('Programación en Java', 5),
('Física General', 4),
('Inglés Intermedio', 2);

```
INSERT INTO Enrollments (student_id, course_id, enrolled_on)
VALUES
(1, 1, '2024-09-01'),
(1, 3, '2024-09-02'),
(2, 2, '2024-09-03'),
(3, 3, '2024-09-04'),
(4, 4, '2024-09-05');
```

--Agregar columna birth_date (fecha) a Students.

```
ALTER TABLE Students
ADD birth_date DATE;
```

--Cambiar el tipo de credits en Courses de entero estándar a entero pequeño.

```
ALTER TABLE Courses
ALTER COLUMN credits TYPE SMALLINT;
```

-- Renombrar la tabla Enrollments a Registrations

```
ALTER TABLE Enrollments
RENAME TO Registrations;
```

-- Eliminar la columna birth_date de Students

```
ALTER TABLE Students
DROP COLUMN birth_date;
```

-- Eliminar la tabla Registrations

DROP TABLE Registrations;

--3. Consultas complejas a resolver (sin mostrar la sintaxis completa)

--Consulta con INNER JOIN

--Quiero obtener el nombre completo del alumno, el título del curso y la fecha de inscripción,

--pero solo para aquellos cursos que tengan 4 o más créditos.

SELECT s.first_name, s.last_name, c.title, e.enrolled_on

FROM Students s

INNER JOIN Enrollments e ON s.student_id = e.student_id

INNER JOIN Courses c ON e.course_id = c.course_id

WHERE c.credits >= 4;

--Debes usar explícitamente la cláusula INNER JOIN entre las tablas correspondientes.

--Consulta con CTE (Common Table Expression)

--Primero, dentro de un CTE, calcula cuántas inscripciones (enrollments) tiene cada estudiante.

--A continuación, selecciona el nombre completo de aquellos estudiantes cuya cantidad de inscripciones sea mayor a uno, ordenándolos de mayor a menor por su total.

--Debes definir y usar la sintaxis de WITH ... AS (...) para el CTE.

WITH enrollment_counts AS (

SELECT student_id,

COUNT(*) AS total_enrollments

FROM Enrollments

GROUP BY student_id

)

SELECT s.first_name, s.last_name, ec.total_enrollments

FROM enrollment_counts ec

JOIN Students s ON ec.student_id = s.student_id

WHERE ec.total_enrollments > 1

ORDER BY ec.total_enrollments DESC;

1. Componentes del esquema inicial (sin SQL, solo listado de lo que debe contener cada tabla)

Tabla Students

Campos

student_id (PK, entero autoincremental)

first_name (texto, no nulo)

last_name (texto, no nulo)

email (texto, único, no nulo)

```
1  ✓ CREATE TABLE Students (  
2    student_id SERIAL PRIMARY KEY,  
3    first_name VARCHAR(50) NOT NULL,  
4    last_name VARCHAR(50) NOT NULL,  
5    email VARCHAR(100) UNIQUE NOT NULL  
6  );
```

Tabla Courses

Campos

course_id (PK, entero autoincremental)

title (texto, no nulo)

credits (entero, no nulo, > 0)

```
7  ✓ CREATE TABLE Courses (  
8    course_id SERIAL PRIMARY KEY,  
9    title varchar(100) NOT NULL,  
10   credits INT NOT NULL CHECK (credits > 0)  
11  );
```

Tabla Enrollments

Campos

enrollment_id (PK, entero autoincremental)

student_id (FK → Students.student_id, no nulo)

course_id (FK → Courses.course_id, no nulo)

enrolled_on (fecha, valor por defecto = fecha actual)

Restricciones adicionales

Par único (student_id, course_id)

```
12 CREATE TABLE Enrollments (  
13   enrollment_id SERIAL PRIMARY KEY,  
14   student_id INT REFERENCES Students(student_id) NOT NULL,  
15   course_id INT REFERENCES Courses(course_id) NOT NULL,  
16   enrolled_on DATE DEFAULT CURRENT_DATE,  
17   UNIQUE (student_id, course_id)  
18 );
```

2. Modificaciones al esquema con DDL (solo descripción de qué hacer, no SQL)

Agregar columna birth_date (fecha) a Students.

```
48 ALTER TABLE Students  
49 ADD birth_date DATE;
```

Cambiar el tipo de credits en Courses de entero estándar a entero pequeño.

```
52 ALTER TABLE Courses  
53 ALTER COLUMN credits TYPE SMALLINT;
```

Renombrar la tabla Enrollments a Registrations.

```
56 ALTER TABLE Enrollments  
57 RENAME TO Registrations;
```

Eliminar la columna birth_date de Students.

```
60 ALTER TABLE Students  
61 DROP COLUMN birth_date;
```

Eliminar la tabla Registrations.

```
64 DROP TABLE Registrations;
```

3. Consultas complejas a resolver (sin mostrar la sintaxis completa)

(Para poder aplicar las consultas tuve que volver a hacer la tabla enrollments ya que habia sido nombrada y eliminada)

Consulta con INNER JOIN

Quiero obtener el nombre completo del alumno, el título del curso y la fecha de inscripción, pero solo para aquellos cursos que tengan 4 o más créditos.

Debes usar explícitamente la cláusula INNER JOIN entre las tablas correspondientes.

```
70 SELECT s.first_name, s.last_name, c.title, e.enrolled_on
71 FROM Students s
72 INNER JOIN Enrollments e ON s.student_id = e.student_id
73 INNER JOIN Courses c ON e.course_id = c.course_id
74 WHERE c.credits >= 4;
```

	first_name character varying (50)	last_name character varying (50)	title character varying (100)	enrolled_on date
1	Ana	Gómez	Matemáticas I	2024-09-01
2	Ana	Gómez	Programación en Java	2024-09-02
3	Carla	Rojas	Programación en Java	2024-09-04
4	Pedro	Fernández	Física General	2024-09-05

Consulta con CTE (Common Table Expression)

Primero, dentro de un CTE, calcula cuántas inscripciones (enrollments) tiene cada estudiante.

A continuación, selecciona el nombre completo de aquellos estudiantes cuya cantidad de inscripciones sea mayor a uno, ordenándolos de mayor a menor por su total.

Debes definir y usar la sintaxis de WITH ... AS (...) para el CTE.

```
81 WITH enrollment_counts AS (
82   SELECT student_id,
83   COUNT(*) AS total_enrollments
84   FROM Enrollments
85   GROUP BY student_id
86 )
87 SELECT s.first_name, s.last_name, ec.total_enrollments
88 FROM enrollment_counts ec
89 JOIN Students s ON ec.student_id = s.student_id
90 WHERE ec.total_enrollments > 1
91 ORDER BY ec.total_enrollments DESC;
```

	first_name character varying (50) 🔒	last_name character varying (50) 🔒	total_enrollments bigint 🔒
1	Ana	Gómez	2