



Instituto Tecnológico de Mexicali

Fundamento de base de datos

Unidad 3

Tarea 4

Luis Alonso Guevara Quiñonez - 23490377

-Docente-

Jose Ramon Bogarin Valenzuela

Ingeniería en sistemas.

--Gestión de empleados y asignación de proyectos

--Eres responsable de crear y mantener la base de datos de una empresa de desarrollo de software. El objetivo es registrar información de empleados, proyectos y asignaciones de empleados a proyectos.

--Tu tarea consiste en completar 5 partes para practicar DDL, DML y consultas avanzadas en PostgreSQL.

--1. Diseño del esquema inicial (solo descripción, no SQL)

--Tabla Employees

--employee_id: clave primaria, entero autoincremental

--first_name: texto, no nulo

--last_name: texto, no nulo

--email: texto, único y no nulo

--position: texto (ej. "Backend Developer", "QA", etc.)

```
CREATE TABLE Employees (  
  employee_id SERIAL PRIMARY KEY,  
  first_name varchar(50) NOT NULL,  
  last_name varchar(50) NOT NULL,  
  email varchar(100) UNIQUE NOT NULL,  
  positionn varchar(50)  
)
```

--Tabla Projects

--project_id: clave primaria, entero autoincremental

--project_name: texto, no nulo

--start_date: fecha

--end_date: fecha (puede ser nula)

```
CREATE TABLE Projects (  
  project_id SERIAL PRIMARY KEY,  
  project_name varchar(100) NOT NULL,  
  start_date DATE,  
  end_date DATE  
);
```

--Tabla Assignments
--assignment_id: clave primaria, entero autoincremental
--employee_id: clave foránea a Employees
--project_id: clave foránea a Projects
--assigned_date: fecha, por defecto la fecha actual

```
CREATE TABLE Assignments (  
assignment_id SERIAL PRIMARY KEY,  
employee_id INT REFERENCES Employees(employee_id),  
project_id INT REFERENCES Projects(project_id),  
assigned_date DATE DEFAULT CURRENT_DATE  
);
```

--2. Modificaciones al esquema (DDL, descritas sin SQL)

--Realiza las siguientes alteraciones al esquema inicial:

--Agrega una columna salary (número decimal) a la tabla Employees.

```
ALTER TABLE Employees  
ADD COLUMN salary DECIMAL;
```

--Cambia el tipo de la columna position a un campo de máximo 100 caracteres.

```
ALTER TABLE Employees  
ALTER COLUMN positionn TYPE VARCHAR(100);
```

--Renombra la tabla Assignments a TeamAssignments.

```
ALTER TABLE Assignments  
RENAME TO TeamAssignments;
```

-- Elimina la columna salary de la tabla Employees.

```
ALTER TABLE Employees  
DROP COLUMN salary;
```

-- Elimina por completo la tabla TeamAssignments.

```
DROP TABLE TeamAssignments;
```

```
SELECT * FROM Employees;
```

```
SELECT * FROM Projects;
```

```
SELECT * FROM Assignments;
```

```
SELECT employee_id, first_name, last_name FROM Employees;
```

```
SELECT project_id, project_name FROM Projects;
```

--3. CRUD con DML (descrito sin SQL)

--Create

--Agrega al menos 3 empleados, incluyendo nombre, apellido, email y posición.

```
INSERT INTO Employees (first_name, last_name, email, positionn) VALUES
```

```
('Donovan', 'Hernandez', 'donovanhernandez@gmail.com', 'QA'),
```

```
('Damian', 'Lugo', 'damianlugo@gmail.com', 'Backend Developer'),
```

```
('Kevin', 'Guevara', 'kevinzahir@gmail.com', 'FrontEnd'),
```

```
('Luis', 'Quiñonez', 'luisalonso@gmail.com', 'AI'),
```

```
('Daniela', 'Guevara', 'danielaguevara@gmail.com', 'FullStack');
```

--Agrega al menos 2 proyectos, con fecha de inicio y fin.

```
INSERT INTO Projects (project_name, start_date, end_date) VALUES
```

```
('Sistema de inventario', '2024-01-10', '2024-07-15'),
```

```
('Plataforma de comercio', '2024-02-01', NULL),
```

```
('App de reserva', '2023-11-20', '2024-04-30'),
```

```
('Red social', '2024-03-01', NULL),
```

```
('Gestor de tareas', '2023-12-05', '2024-06-01');
```

--Asigna 2 empleados al mismo proyecto y un tercero a un proyecto diferente.

```
INSERT INTO Assignments (employee_id, project_id, assigned_date)
```

```
VALUES
```

```
(6, 1, '2024-04-01'),
```

```
(7, 1, '2024-04-01'),
```

(8, 2, '2024-04-01'),

(9, 3, '2024-04-01'),

(10, 4, '2024-04-01');

--Read

--Muestra todos los empleados asignados a un proyecto específico.

SELECT e.first_name, e.last_name

FROM Employees e

JOIN Assignments a ON e.employee_id = a.employee_id

WHERE a.project_id = 1;

--Lista los empleados que tienen correos que terminan en @company.com.

SELECT * FROM Employees

WHERE email LIKE '%@company.com';

--Cambia el email de un empleado específico.

UPDATE Employees

SET email = 'kevinguevara@company.com'

WHERE first_name = 'Kevin' AND last_name = 'Guevara';

--Actualiza el nombre de un proyecto.

UPDATE Projects

SET project_name = 'Sistema de Inventario 2.0'

WHERE project_id = 1;

--Elimina una asignación específica (por employee_id y project_id).

DELETE FROM Assignments

WHERE employee_id = 6 AND project_id = 1;

--Elimina todos los empleados que no estén asignados a ningún proyecto.

```
DELETE FROM Employees
WHERE employee_id NOT IN (
    SELECT DISTINCT employee_id FROM Assignments
);
```

--4. Consultas complejas a desarrollar

--A. Consulta usando INNER JOIN

--Mostrar el nombre completo del empleado, el nombre del proyecto y la fecha en la que fue asignado. Solo deben aparecer los empleados asignados a proyectos que empezaron después de enero 2023.

--Usa INNER JOIN entre las tablas correspondientes.

```
SELECT
    e.first_name, e.last_name, p.project_name, a.assigned_date
FROM Employees e
INNER JOIN Assignments a ON e.employee_id = a.employee_id
INNER JOIN Projects p ON p.project_id = a.project_id
WHERE p.start_date > '2023-01-01';
```

--B. Consulta usando CTE

--En una subconsulta con CTE, cuenta cuántos proyectos tiene asignado cada empleado.

--Luego muestra el nombre completo y la cantidad de proyectos solo para empleados con más de un proyecto asignado.

--Ordena por cantidad de asignaciones descendente.

--Usa WITH para crear el CTE.

```
WITH project_counts AS (
    SELECT employee_id,
    COUNT(project_id) AS project_count
FROM Assignments
GROUP BY employee_id
)
SELECT e.first_name, e.last_name, pc.project_count
```

```
FROM project_counts pc
```

```
JOIN Employees e ON e.employee_id = pc.employee_id
```

```
WHERE pc.project_count > 1
```

```
ORDER BY pc.project_count DESC;
```

--NO TENGO NINGUN EMPLEADO CON MAS DE UN PROYECTO PROFE Y YA ES BN TARDE
PA MOVERLE YA CASI SON LAS 12

Gestión de empleados y asignación de proyectos

Eres responsable de crear y mantener la base de datos de una empresa de desarrollo de software. El objetivo es registrar información de empleados, proyectos y asignaciones de empleados a proyectos. Tu tarea consiste en completar 5 partes para practicar DDL, DML y consultas avanzadas en PostgreSQL.

1. Diseño del esquema inicial (solo descripción, no SQL)

Tabla Employees

employee_id: clave primaria, entero autoincremental

first_name: texto, no nulo

last_name: texto, no nulo

email: texto, único y no nulo

position: texto (ej. "Backend Developer", "QA", etc.)

```
12  CREATE TABLE Employees (  
13     employee_id SERIAL PRIMARY KEY,  
14     first_name varchar(50) NOT NULL,  
15     last_name varchar(50) NOT NULL,  
16     email varchar(100) UNIQUE NOT NULL,  
17     positionn varchar(50)  
18 )
```

```

19  --Tabla Projects
20  --project_id: clave primaria, entero autoincremental
21  --project_name: texto, no nulo
22  --start_date: fecha
23  --end_date: fecha (puede ser nula)
24  CREATE TABLE Projects (
25  project_id SERIAL PRIMARY KEY,
26  project_name varchar(100) NOT NULL,
27  start_date DATE,
28  end_date DATE
29  );

```

```

30  --Tabla Assignments
31  --assignment_id: clave primaria, entero autoincremental
32  --employee_id: clave foránea a Employees
33  --project_id: clave foránea a Projects
34  --assigned_date: fecha, por defecto la fecha actual
35
36  CREATE TABLE Assignments (
37  assignment_id SERIAL PRIMARY KEY,
38  employee_id INT REFERENCES Employees(employee_id),
39  project_id INT REFERENCES Projects(project_id),
40  assigned_date DATE DEFAULT CURRENT_DATE
41  );

```

2. Modificaciones al esquema (DDL, descritas sin SQL)

Realiza las siguientes alteraciones al esquema inicial:

Agrega una columna salary (número decimal) a la tabla Employees

```

45  ALTER TABLE Employees
46  ADD COLUMN salary DECIMAL;

```

Cambia el tipo de la columna position a un campo de máximo 100 caracteres.

```

49  ALTER TABLE Employees
50  ALTER COLUMN positionn TYPE VARCHAR(100);

```

Renombra la tabla Assignments a TeamAssignments.

```

53  ALTER TABLE Assignments
54  RENAME TO TeamAssignments;

```


Elimina la columna salary de la tabla Employees.

```
57 ALTER TABLE Employees
58 DROP COLUMN salary;
```

Elimina por completo la tabla TeamAssignments.

```
61 DROP TABLE TeamAssignments;
```

```
68 --3. CRUD con DML (descrito sin SQL)
69 --Create
70 --Agrega al menos 3 empleados, incluyendo nombre, apellido, email y posición.
71 INSERT INTO Employees (first_name, last_name, email, positionn) VALUES
72 ('Donovan', 'Hernandez', 'donovanhernandez@gmail.com', 'QA'),|
73 ('Damian', 'Lugo', 'damianlugo@gmail.com', 'Backend Developer'),
74 ('Kevin', 'Guevara', 'kevinzahir@gmail.com', 'FrontEnd'),
75 ('Luis', 'Quiñonez', 'luisalonso@gmail.com', 'AI'),
76 ('Daniela', 'Guevara', 'danielaguevara@gmail.com', 'FullStack');
77
78 --Agrega al menos 2 proyectos, con fecha de inicio y fin.
79 INSERT INTO Projects (project_name, start_date, end_date) VALUES
80 ('Sistema de inventario', '2024-01-10', '2024-07-15'),
81 ('Plataforma de comercio', '2024-02-01', NULL),
82 ('App de reserva', '2023-11-20', '2024-04-30'),
83 ('Red social', '2024-03-01', NULL),
84 ('Gestor de tareas', '2023-12-05', '2024-06-01');
85
86 --Asigna 2 empleados al mismo proyecto y un tercero a un proyecto diferente.
87 INSERT INTO Assignments (employee_id, project_id, assigned_date)
88 VALUES
89 (6, 1, '2024-04-01'), (7, 1, '2024-04-01'),
90 (8, 2, '2024-04-01'),
91 (9, 3, '2024-04-01'),
92 (10, 4, '2024-04-01');
```

Muestra todos los empleados asignados a un proyecto específico.

```
95 SELECT e.first_name, e.last_name
96 FROM Employees e
97 JOIN Assignments a ON e.employee_id = a.employee_id
98 WHERE a.project_id = 1;
```

| | first_name character varying (50) 🔒 | last_name character varying (50) 🔒 |
|---|--|---------------------------------------|
| 1 | Donovan | Hernandez |
| 2 | Damian | Lugo |

Lista los empleados que tienen correos que terminan en @company.com.

```
101  SELECT * FROM Employees
102  WHERE email LIKE '%@company.com';
```

| | employee_id [PK] integer | first_name character varying (50) | last_name character varying (50) | email character varying (100) | positionn character varying (100) |
|---|-----------------------------|--------------------------------------|-------------------------------------|----------------------------------|--------------------------------------|
| 1 | 8 | Kevin | Guevara | kevinguevara@company.com | FrontEnd |

Cambia el email de un empleado específico.

```
105  UPDATE Employees
106  SET email = 'kevinguevara@company.com'
107  WHERE first_name = 'Kevin' AND last_name = 'Guevara';
```

Actualiza el nombre de un proyecto.

```
110  UPDATE Projects
111  SET project_name = 'Sistema de Inventario 2.0'
112  WHERE project_id = 1;
```

Elimina una asignación específica (por employee_id y project_id).

```
115  DELETE FROM Assignments
116  WHERE employee_id = 6 AND project_id = 1;
```

Elimina todos los empleados que no estén asignados a ningún proyecto.

```
119  DELETE FROM Employees
120  WHERE employee_id NOT IN (
121      SELECT DISTINCT employee_id FROM Assignments
122  );
```

4. Consultas complejas a desarrollar

A. Consulta usando INNER JOIN

Mostrar el nombre completo del empleado, el nombre del proyecto y la fecha en la que fue asignado. Solo deben aparecer los empleados asignados a proyectos que empezaron después de enero 2023.

Usa INNER JOIN entre las tablas correspondientes.

```
128 SELECT
129     e.first_name, e.last_name, p.project_name, a.assigned_date
130 FROM Employees e
131 INNER JOIN Assignments a ON e.employee_id = a.employee_id
132 INNER JOIN Projects p ON p.project_id = a.project_id
133 WHERE p.start_date > '2023-01-01';
```

| | first_name character varying (50) | last_name character varying (50) | project_name character varying (100) | assigned_date date |
|---|--------------------------------------|-------------------------------------|---|-----------------------|
| 1 | Donovan | Hernandez | Sistema de Inventario | 2024-04-01 |
| 2 | Damian | Lugo | Sistema de Inventario | 2024-04-01 |
| 3 | Kevin | Guevara | Plataforma de comercio | 2024-04-01 |
| 4 | Luis | Quiñonez | App de reserva | 2024-04-01 |
| 5 | Daniela | Guevara | Red social | 2024-04-01 |

B. Consulta usando CTE

En una subconsulta con CTE, cuenta cuántos proyectos tiene asignado cada empleado.

Luego muestra el nombre completo y la cantidad de proyectos solo para empleados con más de un proyecto asignado.

Ordena por cantidad de asignaciones descendente.

Usa WITH para crear el CTE.

```
140 WITH project_counts AS (
141     SELECT employee_id,
142            COUNT(project_id) AS project_count
143 FROM Assignments
144 GROUP BY employee_id
145 )
146 SELECT e.first_name, e.last_name, pc.project_count
147 FROM project_counts pc
148 JOIN Employees e ON e.employee_id = pc.employee_id
149 WHERE pc.project_count > 1
150 ORDER BY pc.project_count DESC;
151 --NO TENGO NINGUN EMPLEADO CON MAS DE UN PROYECTO PROFE Y YA ES BN TARDE PA MOVERLE YA CASI SON LAS 12
```

| | first_name character varying (50) | last_name character varying (50) | project_count bigint |
|--|--------------------------------------|-------------------------------------|-------------------------|
|--|--------------------------------------|-------------------------------------|-------------------------|

Especifique por que no tenia ningun dato