

Instituto Tecnológico de Mexicali



Ingeniería Sistemas Computacionales

Fundamentos de Base de Datos

Tema:

“U3_Tarea#4”

Estudiante:

Mac Callum Merino Fatima Berenice

No. De control:

C21490774

Docente:

José Ramón Bogarín Valenzuela

Mexicali, B.C., 30 de abril de 2025.

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.)

Tabla Projects

project_id: clave primaria, entero autoincremental

project_name: texto, no nulo

start_date: fecha

end_date: fecha (puede ser nula)

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

Restricción única: combinación (employee_id, project_id)

```

----- Tablas -----

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,
  position VARCHAR(100)
);

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

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,
  UNIQUE (employee_id, project_id)
);

```

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.

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

Renombra la tabla Assignments a TeamAssignments.

Elimina la columna salary de la tabla Employees.

Elimina por completo la tabla TeamAssignments.

```

----- Modificacion de tablas -----

ALTER TABLE Employees
ADD COLUMN salary NUMERIC(10,2);

ALTER TABLE Assignments
RENAME TO TeamAssignments;

ALTER TABLE Employees
DROP COLUMN salary;

DROP TABLE TeamAssignments;

```

3. CRUD con DML (descrito sin SQL)

Create

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

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

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

```
----- CRUD con DML (Registros) -----  
----- Create -----  
  
INSERT INTO Employees (first_name, last_name, email, position) VALUES  
( 'Ana', 'Paredes', 'ana@company.com', 'Backend Developer'),  
( 'Luis', 'García', 'luis@company.com', 'QA'),  
( 'María', 'López', 'maria@gmail.com', 'Frontend Developer');  
  
INSERT INTO Projects (project_name, start_date, end_date) VALUES  
( 'Sistema de Ventas', '2024-05-01', '2024-08-30'),  
( 'App Móvil', '2024-06-15', NULL);  
  
INSERT INTO Assignments (employee_id, project_id) VALUES  
(1, 1),  
(2, 1),  
(3, 2);
```

Read

Muestra todos los empleados asignados a un proyecto específico.

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

```
----- Read -----  
  
-- Empleados asignados al proyecto 'Sistema de Ventas'  
SELECT e.first_name, e.last_name  
FROM Employees e  
JOIN Assignments a ON e.employee_id = a.employee_id  
JOIN Projects p ON p.project_id = a.project_id  
WHERE p.project_name = 'Sistema de Ventas';  
  
-- Empleados con correo @company.com  
SELECT * FROM Employees  
WHERE email LIKE '%@company.com';
```

Update

Cambia el email de un empleado específico.

Actualiza el nombre de un proyecto.

```

----- Update -----

-- Cambiar email de Luis García
UPDATE Employees
SET email = 'luis.garcia@company.com'
WHERE first_name = 'Luis' AND last_name = 'García';

-- Cambiar nombre del proyecto
UPDATE Projects
SET project_name = 'Sistema de Gestión de Ventas'
WHERE project_name = 'Sistema de Ventas';

```

Delete

Elimina una asignación específica (por employee_id y project_id).

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

```

----- Delete -----

-- Eliminar asignación específica
DELETE FROM Assignments
WHERE employee_id = 1 AND project_id = 1;

-- Eliminar empleados no asignados
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. Sólo deben aparecer los empleados asignados a proyectos que empezaron después de enero 2023.

Usa INNER JOIN entre las tablas correspondientes.

94

----- Consultas -----

95

96

-- A. INNER JOIN con condición de fecha

97

SELECT e.first_name || ' ' || e.last_name AS nombre_completo,

98

p.project_name,

99

a.assigned_date

100

FROM Employees e

101

JOIN Assignments a ON e.employee_id = a.employee_id

102

JOIN Projects p ON p.project_id = a.project_id

103

WHERE p.start_date > '2023-01-01';

104

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	nombre_completo	project_name	assigned_date
	text	character varying (100)	date
1	Luis García	Sistema de Gestión de Ventas	2025-04-30
2	María López	App Móvil	2025-04-30

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.

```
-- B. CTE con conteo de proyectos por empleado
WITH ProyectosPorEmpleado AS (
  SELECT employee_id, COUNT(*) AS total_proyectos
  FROM Assignments
  GROUP BY employee_id
)
SELECT e.first_name || ' ' || e.last_name AS nombre_completo,
       p.total_proyectos
FROM ProyectosPorEmpleado p
JOIN Employees e ON e.employee_id = p.employee_id
WHERE p.total_proyectos > 1
ORDER BY p.total_proyectos DESC;
```