INSTITUTO TECNOLOGICO DE MEXICALI

FUNDAMENTOS DE BASE DE DATOS

TAREA 4

MAESTRA: JOSE RAMON BOGARIN VALENZUELAJOSE RAMON BOGARIN VALENZUELA

NOMBRE:

- REBECA ELIZABETT MARTINEZ RENDON

NUMERO DE CONTROL:

23490390

BASE DE DATOS

```sql
-- Tabla: Employees
CREATE TABLE Employees (
    employee_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    position VARCHAR(50)
);

-- Tabla: Projects
CREATE TABLE Projects (
    project_id SERIAL PRIMARY KEY,
    project_name TEXT NOT NULL,
    start_date DATE,
    end_date DATE
);

-- Tabla: Assignments
CREATE TABLE Assignments (
    assignment_id SERIAL PRIMARY KEY,
    employee_id INTEGER NOT NULL REFERENCES Employees(employee_id) ON DELETE CASCADE,
    project_id INTEGER NOT NULL REFERENCES Projects(project_id) ON DELETE CASCADE,
    assigned_date DATE DEFAULT CURRENT_DATE,
    UNIQUE (employee_id, project_id)
);
INSERT INTO Employees (first_name, last_name, email, position) VALUES
('Ana', 'Martínez', 'ana.martinez@example.com', 'Backend Developer'),
('Luis', 'Gómez', 'luis.gomez@example.com', 'Frontend Developer'),
('Carla', 'Ramírez', 'carla.ramirez@example.com', 'QA Engineer'),
('Pedro', 'Lopez', 'pedro.lopez@example.com', 'DevOps Engineer'),
('Sofía', 'Torres', 'sofia.torres@example.com', 'Project Manager');

INSERT INTO Projects (project_name, start_date, end_date) VALUES
('Sistema de Inventario', '2024-01-15', '2024-06-30'),
('Plataforma de E-learning', '2024-03-01', NULL),
('App de Reservas', '2024-02-10', '2024-08-20'),
('Reestructuración Backend', '2024-04-05', NULL),
('Sistema de Tickets', '2024-01-01', '2024-05-15');


INSERT INTO Assignments (employee_id, project_id, assigned_date) VALUES
(1, 1, '2024-01-15'),
(2, 2, '2024-03-05'),
(3, 3, '2024-02-12'),
(4, 4, '2024-04-07'),
(5, 5, '2024-01-02');
```

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.

```
52  SELECT * FROM Employees
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (50)
1	1	Ana	Martínez	ana.martinez@example.co…	Backend Developer
2	2	Luis	Gómez	luis.gomez@example.com	Frontend Developer
3	3	Carla	Ramírez	carla.ramirez@example.com	QA Engineer
4	4	Pedro	Lopez	pedro.lopez@example.com	DevOps Engineer
5	5	Sofía	Torres	sofia.torres@example.com	Project Manager

```
48  --Agrega una columna salary (número decimal) a la tabla Employees.
49  ALTER TABLE Employees
50  ADD COLUMN salary NUMERIC(10, 2);
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (50)	salary numeric (10,2)
1	1	Ana	Martínez	ana.martinez@example.co…	Backend Developer	[null]
2	2	Luis	Gómez	luis.gomez@example.com	Frontend Developer	[null]
3	3	Carla	Ramírez	carla.ramirez@example.com	QA Engineer	[null]
4	4	Pedro	Lopez	pedro.lopez@example.com	DevOps Engineer	[null]
5	5	Sofía	Torres	sofia.torres@example.com	Project Manager	[null]

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

```
54  --Cambia el tipo de la columna position a un campo de máximo 100 caracteres.
55  ALTER TABLE Employees
56  ALTER COLUMN position TYPE VARCHAR(100);
57
```

- Renombra la tabla Assignments a TeamAssignments.

```
58  --Renombra la tabla Assignments a TeamAssignments.
59  SELECT * FROM Assignments
```

Data Output | Messages | Notifications

	assignment_id [PK] integer	employee_id integer	project_id	assigned_date date
1	1	1	1	2024-01-15
2	2	2	2	2024-03-05
3	3	3	3	2024-02-12
4	4	4	4	2024-04-07
5	5	5	5	2024-01-02

Graph Visualiser

```
58  --Renombra la tabla Assignments a TeamAssignments.
59   SELECT * FROM Assignments
60  ALTER TABLE Assignments
61  RENAME TO TeamAssignments;
62  SELECT * FROM TeamAssignments
```

- Elimina la columna salary de la tabla Employees.

```
65
66  SELECT * FROM Employees
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (100)	salary numeric (10,2)
1	1	Ana	Martínez	ana.martinez@example.co…	Backend Developer	[null]
2	2	Luis	Gómez	luis.gomez@example.com	Frontend Developer	[null]
3	3	Carla	Ramírez	carla.ramirez@example.com	QA Engineer	[null]
4	4	Pedro	Lopez	pedro.lopez@example.com	DevOps Engineer	[null]
5	5	Sofía	Torres	sofia.torres@example.com	Project Manager	[null]

Total rows: 5 | Query complete 00:00:00.186

```
63  --Elimina la columna salary de la tabla Employees.
64  SELECT * FROM Employees
65  ALTER TABLE Employees
66  DROP COLUMN salary;
67
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (100)
1	1	Ana	Martínez	ana.martinez@example.co…	Backend Developer
2	2	Luis	Gómez	luis.gomez@example.com	Frontend Developer
3	3	Carla	Ramírez	carla.ramirez@example.com	QA Engineer
4	4	Pedro	Lopez	pedro.lopez@example.com	DevOps Engineer
5	5	Sofía	Torres	sofia.torres@example.com	Project Manager

- Elimina por completo la tabla TeamAssignments.

```
68  --Elimina por completo la tabla TeamAssignments.
69  SELECT * FROM TeamAssignments
```

Data Output | Messages | Notifications

	assignment_id [PK] integer	employee_id integer	project_id integer	assigned_date date
1	1	1	1	2024-01-15
2	2	2	2	2024-03-05
3	3	3	3	2024-02-12
4	4	4	4	2024-04-07
5	5	5	5	2024-01-02

```
68  --Elimina por completo la tabla TeamAssignments.
69  SELECT * FROM TeamAssignments
70  DROP TABLE TeamAssignments;
```

Data Output | Messages | Notifications

```
DROP TABLE

Query returned successfully in 64 msec.
```

```
68  --Elimina por completo la tabla TeamAssignments.
69  SELECT * FROM TeamAssignments
70  DROP TABLE TeamAssignments;
```

Data Output | Messages | Notifications

```
ERROR:  no existe la relación «teamassignments»
LINE 1: SELECT * FROM TeamAssignments
                      ^

SQL state: 42P01
Character: 15
```

3. CRUD con DML (descrito sin SQL)

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

```sql
--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, position)
VALUES
  ('LaurA', 'Gómez', 'laura.gomez@example.com', 'Analista de Datos'),
  ('CarlosS', 'Pérez', 'carlos.perez@example.com', 'Desarrollador Backend'),
  ('AnNa', 'Ramírez', 'ana.ramirez@example.com', 'Diseñadora UX/UI');

  SELECT * FROM Employees
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (100)
1	1	Ana	Martínez	ana.martinez@example.co…	Backend Developer
2	2	Luis	Gómez	luis.gomez@example.com	Frontend Developer
3	3	Carla	Ramírez	carla.ramirez@example.com	QA Engineer
4	4	Pedro	Lopez	pedro.lopez@example.com	DevOps Engineer
5	5	Sofía	Torres	sofia.torres@example.com	Project Manager


```sql
--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, position)
VALUES
  ('LaurA', 'Gómez', 'laura.gomez@example.com', 'Analista de Datos'),
  ('CarlosS', 'Pérez', 'carlos.perez@example.com', 'Desarrollador Backend'),
  ('AnNa', 'Ramírez', 'ana.ramirez@example.com', 'Diseñadora UX/UI');

  SELECT * FROM Employees
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (100)
1	1	Ana	Martínez	ana.martinez@example.co…	Backend Developer
2	2	Luis	Gómez	luis.gomez@example.com	Frontend Developer
3	3	Carla	Ramírez	carla.ramirez@example.com	QA Engineer
4	4	Pedro	Lopez	pedro.lopez@example.com	DevOps Engineer
5	5	Sofía	Torres	sofia.torres@example.com	Project Manager
6	6	LaurA	Gómez	laura.gomez@example.com	Analista de Datos
7	7	CarlosS	Pérez	carlos.perez@example.com	Desarrollador Backend
8	8	AnNa	Ramírez	ana.ramirez@example.com	Diseñadora UX/UI

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

```
86
87  --Agrega al menos 2 proyectos, con fecha de inicio y fin.
88  SELECT * FROM Projects
89
90  INSERT INTO Projects (project_name, start_date, end_date) VALUES
91  ('SisteASa de InventaArio', '2024-03-15', '2024-067-30'),
92  ('ApLICA de ReservaAs', '2024-04-10', '2024-08-28');
93
```

Data Output | Messages | Notifications

	project_id [PK] integer	project_name text	start_date date	end_date date
1	1	Sistema de Inventario	2024-01-15	2024-06-30
2	2	Plataforma de E-learning	2024-03-01	[null]
3	3	App de Reservas	2024-02-10	2024-08-20
4	4	Reestructuración Backend	2024-04-05	[null]
5	5	Sistema de Tickets	2024-01-01	2024-05-15

```
87  --Agrega al menos 2 proyectos, con fecha de inicio y fin.
88  SELECT * FROM Projects
89
90  INSERT INTO Projects (project_name, start_date, end_date) VALUES
91  ('SisteASa de InventaArio', '2024-03-15', '2024-07-30'),
92  ('ApLICA de ReservaAs', '2024-04-10', '2024-08-28');
93
```

Data Output | Messages | Notifications

	project_id [PK] integer	project_name text	start_date date	end_date date
1	1	Sistema de Inventario	2024-01-15	2024-06-30
2	2	Plataforma de E-learning	2024-03-01	[null]
3	3	App de Reservas	2024-02-10	2024-08-20
4	4	Reestructuración Backend	2024-04-05	[null]
5	5	Sistema de Tickets	2024-01-01	2024-05-15
6	6	SisteASa de InventaArio	2024-03-15	2024-07-30
7	7	ApLICA de ReservaAs	2024-04-10	2024-08-28

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

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

SELECT employee_id, first_name, last_name FROM Employees;
SELECT project_id, project_name FROM Projects;

CREATE TABLE TeamAssignments (
    assignment_id SERIAL PRIMARY KEY,
    employee_id INTEGER NOT NULL REFERENCES Employees(employee_id) ON DELETE CASCADE,
    project_id INTEGER NOT NULL REFERENCES Projects(project_id) ON DELETE CASCADE,
    assigned_date DATE DEFAULT CURRENT_DATE,
    UNIQUE (employee_id, project_id)
);
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)
1	1	Ana	Martínez
2	2	Luis	Gómez
3	3	Carla	Ramírez
4	4	Pedro	Lopez
5	5	Sofía	Torres
6	6	LaurA	Gómez
7	7	CarlosS	Pérez
8	8	AnNa	Ramírez

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

SELECT employee_id, first_name, last_name FROM Employees;
SELECT project_id, project_name FROM Projects;

CREATE TABLE TeamAssignments (
    assignment_id SERIAL PRIMARY KEY,
    employee_id INTEGER NOT NULL REFERENCES Employees(employee_id) ON DELETE CASCADE,
    project_id INTEGER NOT NULL REFERENCES Projects(project_id) ON DELETE CASCADE,
    assigned_date DATE DEFAULT CURRENT_DATE,
    UNIQUE (employee_id, project_id)
);
```

Data Output | Messages | Notifications

	project_id [PK] integer	project_name text
1	1	Sistema de Inventario
2	2	Plataforma de E-learning
3	3	App de Reservas
4	4	Reestructuración Backend
5	5	Sistema de Tickets
6	6	SisteASa de InventaArio
7	7	ApLICA de ReservaAs

```
INSERT INTO TeamAssignments (employee_id, project_id, assigned_date) VALUES
(6, 6, CURRENT_DATE),
(7, 6, CURRENT_DATE),
(8, 7, CURRENT_DATE);
SELECT * FROM TeamAssignments;

SELECT
    e.first_name || ' ' || e.last_name AS empleado,
    p.project_name AS proyecto,
    t.assigned_date
FROM TeamAssignments t
JOIN Employees e ON t.employee_id = e.employee_id
JOIN Projects p ON t.project_id = p.project_id;
```

Data Output | Messages | Notifications

	empleado text	proyecto text	assigned_date date
1	LaurA Gómez	SisteASa de InventaArio	2025-04-30
2	CarlosS Pérez	SisteASa de InventaArio	2025-04-30
3	AnNa Ramír...	ApLICA de ReservaAs	2025-04-30

- Muestra todos los empleados asignados a un proyecto específico.

```
113  --Read
114  --Muestra todos los empleados asignados a un proyecto específico.
115
116  SELECT
117      e.first_name || ' ' || e.last_name AS empleado,
118      p.project_name AS proyecto,
119      t.assigned_date
120  FROM TeamAssignments t
121  INNER JOIN Employees e ON t.employee_id = e.employee_id
122  INNER JOIN Projects p ON t.project_id = p.project_id
123  WHERE p.project_name = 'SisteASa de InventaArio';
124
```

Data Output | Messages | Notifications

	empleado text	proyecto text	assigned_date date
1	LaurA Gómez	SisteASa de InventaArio	2025-04-30
2	CarlosS Pérez	SisteASa de InventaArio	2025-04-30

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

```
132
133  --Lista los empleados que tienen correos que terminan en @company.com.
134
135  SELECT
136      employee_id,
137      first_name || ' ' || last_name AS nombre_empleado,
138      email
139  FROM Employees
140  WHERE email LIKE '%@example.com';
```

Data Output | Messages | Notifications

	employee_id [PK] integer	nombre_empleado text	email character varying (100)
1	1	Ana Martínez	ana.martinez@example.co…
2	2	Luis Gómez	luis.gomez@example.com
3	3	Carla Ramírez	carla.ramirez@example.com
4	4	Pedro Lopez	pedro.lopez@example.com
5	5	Sofía Torres	sofia.torres@example.com
6	6	LaurA Gómez	laura.gomez@example.com
7	7	CarlosS Pérez	carlos.perez@example.com
8	8	AnNa Ramírez	ana.ramirez@example.com

- Cambia el email de un empleado específico.

```
145  UPDATE Employees
146  SET email = 'pedro.lopez12@example.com'
147  WHERE first_name = 'Pedro' AND last_name = 'Lopez';
148
```

Data Output | Messages | Notifications

```
UPDATE 1

Query returned successfully in 63 msec.
```

```
142  --Update
143  --Cambia el email de un empleado específico.
144  SELECT * FROM Employees
145  UPDATE Employees
146  SET email = 'pedro.lopez12@example.com'
147  WHERE first_name = 'Pedro' AND last_name = 'Lopez';
148
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (100)
1	1	Ana	Martínez	ana.martinez@example.com	Backend Developer
2	2	Luis	Gómez	luis.gomez@example.com	Frontend Developer
3	3	Carla	Ramírez	carla.ramirez@example.com	QA Engineer
4	5	Sofía	Torres	sofia.torres@example.com	Project Manager
5	6	LaurA	Gómez	laura.gomez@example.com	Analista de Datos
6	7	CarlosS	Pérez	carlos.perez@example.com	Desarrollador Backend
7	8	AnNa	Ramírez	ana.ramirez@example.com	Diseñadora UX/UI
8	4	Pedro	Lopez	pedro.lopez12@example.com	DevOps Engineer

- Actualiza el nombre de un proyecto.

```
148
149  --Actualiza el nombre de un proyecto.
150  SELECT * FROM Projects
```

Data Output | Messages | Notifications

	project_id [PK] integer	project_name text	start_date date	end_date date
1	1	Sistema de Inventario	2024-01-15	2024-06-30
2	2	Plataforma de E-learning	2024-03-01	[null]
3	3	App de Reservas	2024-02-10	2024-08-20
4	4	Reestructuración Backend	2024-04-05	[null]
5	5	Sistema de Tickets	2024-01-01	2024-05-15
6	6	SisteASa de InventaArio	2024-03-15	2024-07-30
7	7	ApLICA de ReservaAs	2024-04-10	2024-08-28

```
152
153  UPDATE Projects
154  SET project_name = 'Sistema de Inventario v2'
155  WHERE project_name = 'SisteASa de InventaArio';
156
```

Data Output | Messages | Notifications

```
UPDATE 1

Query returned successfully in 70 msec.
```

```
148
149  --Actualiza el nombre de un proyecto.
150  SELECT * FROM Projects
151  WHERE project_name LIKE '%Inventario%';
152
```

Data Output | Messages | Notifications

	project_id [PK] integer	project_name text	start_date date	end_date date
1	1	Sistema de Inventario	2024-01-15	2024-06-30

```
148
149  --Actualiza el nombre de un proyecto.
150  SELECT * FROM Projects
151  WHERE project_name LIKE '%Inventario%';
152
153  UPDATE Projects
154  SET project_name = 'Sistema de Inventario v2'
155  WHERE project_name = 'SisteASa de InventaArio';
156
```

Data Output | Messages | Notifications

	project_id [PK] integer	project_name text	start_date date	end_date date
1	1	Sistema de Inventario	2024-01-15	2024-06-30
2	6	Sistema de Inventario v2	2024-03-15	2024-07-30

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

```
157
158  --Delete
159  --Elimina una asignación específica (por employee_id y project_id).
160  SELECT * FROM TeamAssignments
161  WHERE employee_id = 6 AND project_id = 6;
162
```

Data Output | Messages | Notifications

	assignment_id [PK] integer	employee_id integer	project_id integer	assigned_date date
1	1	6	6	2025-04-30

```
158  --Delete
159  --Elimina una asignación específica (por employee_id y project_id).
160  SELECT * FROM TeamAssignments
161  WHERE employee_id = 6 AND project_id = 6;
162
163  DELETE FROM TeamAssignments
164  WHERE employee_id = 6 AND project_id = 6;
165
```

Data Output | Messages | Notifications

```
DELETE 1

Query returned successfully in 46 msec.
```

```
158  --Delete
159  --Elimina una asignación específica (por employee_id y project_id).
160  SELECT * FROM TeamAssignments
161  WHERE employee_id = 6 AND project_id = 6;
162
163  DELETE FROM TeamAssignments
164  WHERE employee_id = 6 AND project_id = 6;
165
```

Data Output | Messages | Notifications

	assignment_id [PK] integer	employee_id integer	project_id integer	assigned_date date

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

```
166  --Elimina todos los empleados que no estén asignados a ningún proyecto
167
168  SELECT * FROM Employees
169  WHERE employee_id NOT IN (
170      SELECT DISTINCT employee_id FROM TeamAssignments
171  );
172
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (100)
1	1	Ana	Martínez	ana.martinez@example.com	Backend Developer
2	2	Luis	Gómez	luis.gomez@example.com	Frontend Developer
3	3	Carla	Ramírez	carla.ramirez@example.com	QA Engineer
4	5	Sofía	Torres	sofia.torres@example.com	Project Manager
5	6	LaurA	Gómez	laura.gomez@example.com	Analista de Datos
6	4	Pedro	Lopez	pedro.lopez12@example.com	DevOps Engineer

```
166  --Elimina todos los empleados que no estén asignados a ningún proyecto
167
168  SELECT * FROM Employees
169  WHERE employee_id NOT IN (
170      SELECT DISTINCT employee_id FROM TeamAssignments
171  );
172  DELETE FROM Employees
173  WHERE employee_id NOT IN (
174      SELECT DISTINCT employee_id FROM TeamAssignments
175  );
```

Data Output | Messages | Notifications

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	position character varying (100)

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.

```sql
--4. Consultas complejas a desarrollar
--A. Consulta usando INNER JOIN
SELECT
    e.first_name || ' ' || e.last_name AS nombre_empleado,
    p.project_name AS nombre_proyecto,
    t.assigned_date AS fecha_asignacion
FROM TeamAssignments t
INNER JOIN Employees e ON t.employee_id = e.employee_id
INNER JOIN Projects p ON t.project_id = p.project_id
WHERE p.start_date > '2023-01-31';
```

Data Output | Messages | Notifications

	nombre_empleado text	nombre_proyecto text	fecha_asignacion date
1	CarlosS Pérez	Sistema de Inventario v2	2025-04-30
2	AnNa Ramírez	ApLICA de ReservaAs	2025-04-30


```sql
--4. Consultas complejas a desarrollar
--A. Consulta usando INNER JOIN
SELECT
    e.first_name || ' ' || e.last_name AS nombre_empleado,
    p.project_name AS nombre_proyecto,
    t.assigned_date AS fecha_asignacion
FROM TeamAssignments t
INNER JOIN Employees e ON t.employee_id = e.employee_id
INNER JOIN Projects p ON t.project_id = p.project_id
WHERE p.start_date > '2023-01-31';
```

Data Output | Messages | Notifications

	nombre_empleado text	nombre_proyecto text	fecha_asignacion date
1	CarlosS Pérez	Sistema de Inventario v2	2025-04-30
2	AnNa Ramírez	ApLICA de ReservaAs	2025-04-30

5. **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. Consulta usando CTE
--En una subconsulta con CTE, cuenta cuántos proyectos tiene asignado cada empleado.
WITH ProyectoPorEmpleado AS (
    SELECT
        employee_id,
        COUNT(*) AS cantidad_proyectos
    FROM TeamAssignments
    GROUP BY employee_id
)
SELECT
    e.first_name || ' ' || e.last_name AS nombre_empleado,
    ppe.cantidad_proyectos
FROM ProyectoPorEmpleado ppe
JOIN Employees e ON ppe.employee_id = e.employee_id
ORDER BY ppe.cantidad_proyectos DESC;
```

Data Output | Messages | Notifications

Show

	nombre_empleado text	cantidad_proyectos bigint
1	CarlosS Pérez	1
2	AnNa Ramírez	1