EJERCICIO - PostgreSQL

lunes, 15 de septiembre de 2025 12:21

Tema:	Ejercicio PostgreSQL
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	Quinto - A
Alumnos participantes:	Cholota Guaman Carlos Sebastian
Asignatura:	Sistemas de Bases de Datos Distribuidos
Docente:	Ing. Rubén Caiza, Mg.

INFORME DE TALLER
Objetivos

General:

Desarrollar habilidades en el manejo de subconsultas SQL mediante la creación y análisis de un esquema relacional en PostgreSQL, que permita extraer información compleja y realizar comparaciones entre entidades como empleados, departamentos y proyectos, fortaleciendo la capacidad de análisis de datos y optimización de consultas en bases de datos

Específicos:

- Dominar el uso de subconsultas en cláusulas SELECT, WHERE, FROM y HAVING.
- Aplicar funciones de agregación (AVG, MAX, etc.) dentro de subconsultas para obtener métricas comparativas.
- Utilizar subconsultas correlacionadas para comparar datos entre filas relacionadas (por ejemplo, salario promedio por departamento).
- Identificar patrones de uso de subconsultas escalar vs. de conjunto, y cuándo conviene cada una.

Instrucciones

- Crear las tablas
- Insertar datos de prueba
- Realizar subconsultas
- Comentar cada consulta
- Validar resultados
- Documentar el proyecto

Ejercicios Subconsultas

```
CREATE TABLE departamentos (
  id INT PRIMARY KEY,
  nombre VARCHAR(50)
);
```

Actividades por desarrollar

Se contempla la ejecución de actividades fundamentales como la creación del esquema relacional, la definición de claves primarias y foráneas, y la inserción de datos de prueba que permitan validar distintos escenarios. Estas acciones preparatorias garantizarán una base sólida para aplicar subconsultas de manera efectiva, facilitando el análisis comparativo entre empleados, departamentos y proyectos dentro del entorno de PostgreSQL.

Resultados obtenidos

Creación de tablas

```
CREATE TABLE empleados (
  id INT PRIMARY KEY,
  nombre VARCHAR(50),
  salario DECIMAL(10,2),
  departamento_id INT,
  FOREIGN KEY (departamento_id) REFERENCES departamentos(id)
);
```

```
CREATE TABLE empleados (
    id INT PRIMARY KEY,
    nombre VARCHAR(50),
    salario DECIMAL(10,2),
    departamento_id INT,
    FOREIGN KEY (departamento_id) REFERENCES departamentos(id)
```

```
);
```

```
CREATE TABLE proyectos (
  id INT PRIMARY KEY,
  nombre VARCHAR(50),
  departamento_id INT,
  FOREIGN KEY (departamento_id) REFERENCES departamentos(id)
);
```

```
CREATE TABLE proyectos (
    id INT PRIMARY KEY,
    nombre VARCHAR(50),
    departamento_id INT,
    FOREIGN KEY (departamento_id) REFERENCES departamentos(id)
);
```

Inserta algunos datos de prueba (mínimo 3 departamentos y 6 empleados) y realiza las siguientes consultas usando subconsultas:

```
INSERT INTO departamentos (id, nombre) VALUES
(1, 'TI'),
(2, 'MARKETING'),
(3, 'FINANZAS'),
(4, 'RRHH'),
(5, 'LOGÍSTICA');
```

```
INSERT INTO empleados (id, nombre, salario, departamento_id) VALUES
(1, 'ANA PÉREZ', 1200.00, 1),
(2, 'LUIS SOTO', 1100.00, 2),
(3, 'MARTA LÓPEZ', 1350.00, 1),
(4, 'CARLOS GÓMEZ', 1250.00, 4),
(5, 'JULIA VEGA', 1400.00, 3);
(6, 'DAVID MORALES', 1180.00, 1),
(7, 'SANDRA JIMÉNEZ', 1125.50, 2),
(8, 'FERNANDO CASTRO', 1450.75, 3),
(9, 'PATRICIA REYES', 1070.00, 5);
```

SUBCONSULTAS

Empleado con salario mayor al promedio:

Obtén todos los empleados cuyo salario es mayor al salario promedio de todos los empleados.

(Subconsulta en WHERE con función de agregación)

Query Query History

```
SELECT *
FROM empleados
WHERE salario > (
    SELECT AVG(salario)
    FROM empleados
);
```

Data Output Messages Notifications

	id [PK] integer	nombre character varying (50)	salario numeric (10,2)	departamento_id integer
1	3	MARTA LÓPEZ	1350.00	1
2	4	CARLOS GÓMEZ	1250.00	4
3	5	JULIA VEGA	1400.00	3
4	8	FERNANDO CASTRO	1450.75	3

Departamentos sin empleados:

Lista los nombres de los departamentos que no tienen empleados asignados.

(Subconsulta en NOT IN)

Query Query History

```
SELECT nombre
FROM departamentos
WHERE id NOT IN (
    SELECT departamento_id
    FROM empleados
    WHERE departamento_id IS NOT NULL
);
```

Data Output Messages Notifications

	nombre character varying (50)
1	DESARROLLO DE PRODUCTO
2	SERVICIO AL CLIENTE

Empleado con salario más alto:

Muestra el nombre y salario del empleado que gana más.

(Subconsulta escalar en WHERE o SELECT)

```
SELECT nombre, salario
FROM empleados
WHERE salario = (
    SELECT MAX(salario)
    FROM empleados
);
```

	nombre character varying (50)	salario numeric (10,2)
1	FERNANDO CASTRO	1450.75

Salario promedio por departamento:

Para cada empleado, muestra: nombre, salario y el salario promedio de su departamento.

(Subconsulta correlacionada que compare empleados.departamento_id)

```
SELECT
    e.nombre,
    e.salario,
    (
        SELECT AVG(e2.salario)
        FROM empleados e2
        WHERE e2.departamento_id = e.departamento_i
    ) AS promedio_departamento
FROM empleados e;
```

	nombre character varying (50)	salario numeric (10,2)	promedio_departamento numeric
1	ANA PÉREZ	1200.00	1243.3333333333333333
2	LUIS SOTO	1100.00	1112.7500000000000000
3	MARTA LÓPEZ	1350.00	1243.3333333333333333
4	CARLOS GÓMEZ	1250.00	1250.0000000000000000
5	JULIA VEGA	1400.00	1425.3750000000000000
6	DAVID MORALES	1180.00	1243.3333333333333333
7	SANDRA JIMÉNEZ	1125.50	1112.7500000000000000
8	FERNANDO CASTRO	1450.75	1425.3750000000000000
9	PATRICIA REYES	1070.00	1070.0000000000000000
10	ROBERTO SALAZAR	950.00	950.0000000000000000

Departamentos con promedio mayor al promedio general:

Lista los departamentos cuyo promedio de salario es mayor al promedio general de la empresa.

(Subconsulta en HAVING o FROM)

```
SELECT d.nombre AS departamento, AVG(e.salario) AS promedio_departamento
FROM empleados e
JOIN departamentos d ON e.departamento_id = d.id
GROUP BY d.nombre
HAVING AVG(e.salario) > (
    SELECT AVG(salario) FROM empleados
);
```

	departamento character varying (50)	promedio_departamento numeric
1	RRHH	1250.0000000000000000
2	TI	1243.3333333333333333
3	FINANZAS	1425.3750000000000000