INFORME DE EJERCICIOS DE SUBCONSULTAS EN POSTGRESQL

I. PORTADA

Tema:	Ejercicios de Subconsultas en PostgreSQL
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	Quinto “A”
Alumnos participantes:	Gómez Llerena Luis Fernando
Asignatura:	Sistemas de Base de Datos Distribuidas
Docente:	Ing. Jose Ruben Caiza Caizabuano, Mg.

II. INFORME DE EJERCICIOS DE SUBCONSULTAS EN POSTGRESQL

2.1 Objetivos

2.1.1 Objetivo General:

Dominar la implementación de consultas SQL avanzadas, específicamente subconsultas (subqueries), para resolver problemas de recuperación y filtrado de datos complejos en un esquema relacional de orquestas.

2.1.2 Objetivos Específicos:

- Utilizar subconsultas en la cláusula WHERE con operadores como IN para filtrar registros de una tabla (ej. orchestras) basándose en los resultados de otra consulta (ej. ciudades de conciertos en concerts).
- Aplicar subconsultas para realizar cálculos agregados (ej. AVG()) y utilizar el resultado como condición de filtro para los datos de la consulta principal.
- Implementar subconsultas correlacionadas o anidadas en la cláusula HAVING para filtrar grupos de datos (ej. orquestas) basándose en una comparación con un promedio calculado a partir de un subconjunto de datos agrupados.

2.2 Modalidad

Presencial

2.3 Tiempo de duración

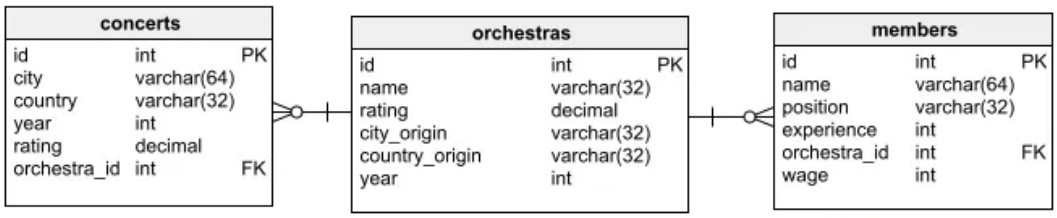
Presenciales: 3

No presenciales: 0

2.4 Instrucciones

Ejercicios de subconsultas SQL

Conjunto de datos: Orquestas Los siguientes ejercicios utilizan el conjunto de datos de orquestas que contiene tres tablas.



La orchestrastabla almacena todas las orquestas. Las columnas son id, name, rating, city_origin, country_origin, y yearen las que se fundó la orquesta.

La concertstabla contiene todos los conciertos de las orquestas. Las columnas son id, city, country, year, rating, y orchestra_id(hace referencia a la orchestrastabla).

La members tabla almacena los miembros de cada orquesta (es decir, los músicos que la tocan). Las columnas son id, name, position(es decir, el instrumento tocado), wage, experience, y orchestra_id(hace referencia a la orchestrastabla).

2.5 Listado de equipos, materiales y recursos

Listado de equipos y materiales generales empleados en la guía práctica:

- Computador con Windows/Linux/macOS.
- PostgreSQL

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:
☒Plataformas educativas
☒Simuladores y laboratorios virtuales
☐Aplicaciones educativas
☒Recursos audiovisuales
☐Gamificación
☐Inteligencia Artificial
Otros (Especifique): _____

2.6 Actividades desarrolladas

2.6.1 Creación de la Base de Datos

Primera se realiza la creación de la base de datos con el nombre deseado.

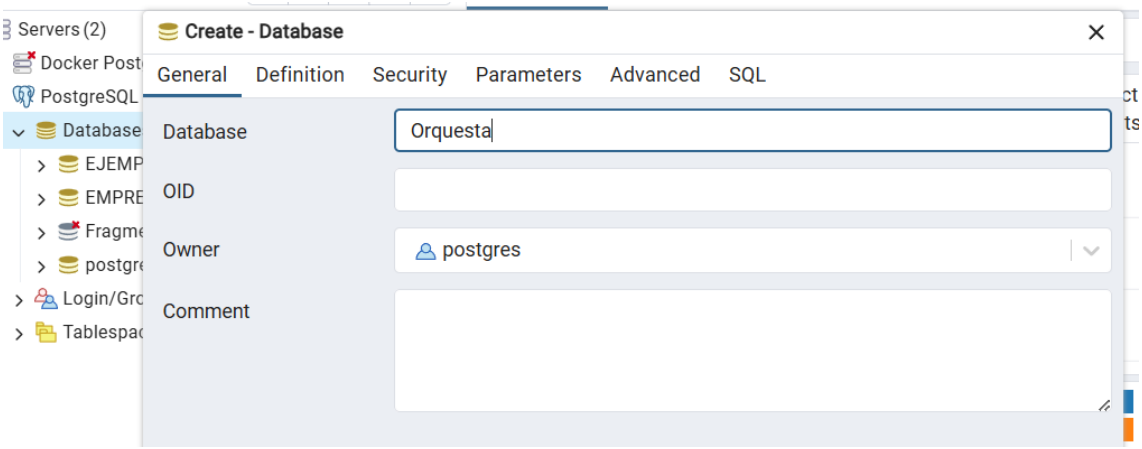


Ilustración 1 Creación de la Base de Datos

2.6.2 Creación de Tablas dentro de la Base Creada

Creación de la Tabla Orchestras

Ilustración 2 Creación de la Tabla Orchestras

Creación de la Tabla Concerts

Query | Query History

```
1  CREATE TABLE concerts ( id SERIAL PRIMARY KEY, city VARCHAR(50),
2  country VARCHAR(50),
3  year INT,
4  rating DECIMAL(3,2),
5  orchestra_id INT REFERENCES orchestras(id)
6  );
7
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 82 msec.
```

Ilustración 3 Creación de la Tabla Concerts

Creación de la Tabla Members

Query | Query History

```
1  CREATE TABLE members ( id SERIAL PRIMARY KEY, name VARCHAR(100),
2  position VARCHAR(50),
3  wage DECIMAL(10,2),
4  experience INT,
5  orchestra_id INT REFERENCES orchestras(id)
6  );
7
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 79 msec.
```

Ilustración 4 Creación de la Tabla Members

Resultado Final de la Ceación Correcta de las Tablas

Ilustración 5 Resultado Final de la Creación Correcta de las Tablas

2.6.3 Inserción de Datos en las Diferentes Tablas Creadas

Query | Query History

```
('Orquesta Ciudad de Mexico', 4.6, 'Ciudad de Mexico', 'Mexico', 1960),
('Filarmonica Boston', 4.4, 'Boston', 'USA', 1925), ('Orquesta Los Angel
('Filarmonica Montreal', 4.6, 'Montreal', 'Canada', 1965),
('Orquesta Sydney', 4.2, 'Sydney', 'Australia', 1955),
('Sinfonica Melbourne', 4.4, 'Melbourne', 'Australia', 1960), ('Filarmon
('Sinfonica Estocolmo', 4.6, 'Estocolmo', 'Suecia', 1925),
('Filarmonica Copenhague', 4.4, 'Copenhague', 'Dinamarca', 1910),
('Orquesta Helsinki', 4.3, 'Helsinki', 'Finlandia', 1935),
('Sinfonica Roma', 4.7, 'Roma', 'Italia', 1905),
('Filarmonica Milan', 4.5, 'Milan', 'Italia', 1915),
('Orquesta Venecia', 4.2, 'Venecia', 'Italia', 1920),
('Sinfonica Lisboa', 4.3, 'Lisboa', 'Portugal', 1940),
('Filarmonica Amsterdam', 4.6, 'Amsterdam', 'Paises Bajos', 1930), ('Orq
```

Data Output | Messages | Notifications

```
INSERT 0 30

Query returned successfully in 79 msec.
```

Ilustración 6 Inserción de Datos

```
Query  Query History
11  ('Filarmonica Boston', 4.4, 'Boston', 'USA', 1925),
12  ('Orquesta Los Angeles', 4.5, 'Los Angeles', 'USA', 1945), ('Sinfonica T
13  ('Filarmonica Montreal', 4.6, 'Montreal', 'Canada', 1965),
14  ('Orquesta Sydney', 4.2, 'Sydney', 'Australia', 1955),
15  ('Sinfonica Melbourne', 4.4, 'Melbourne', 'Australia', 1960), ('Filarmon
16  ('Sinfonica Estocolmo', 4.6, 'Estocolmo', 'Suecia', 1925),
17  ('Filarmonica Copenhague', 4.4, 'Copenhague', 'Dinamarca', 1910),
18  ('Orquesta Helsinki', 4.3, 'Helsinki', 'Finlandia', 1935),
19  ('Sinfonica Roma', 4.7, 'Roma', 'Italia', 1905),
20  ('Filarmonica Milan', 4.5, 'Milan', 'Italia', 1915),
21  ('Orquesta Venecia', 4.2, 'Venecia', 'Italia', 1920),
22  ('Sinfonica Lisboa', 4.3, 'Lisboa', 'Portugal', 1940),
23  ('Filarmonica Amsterdam', 4.6, 'Amsterdam', 'Paises Bajos', 1930), ('Orq
24

Data Output  Messages  Notifications

INSERT 0 30

Query returned successfully in 71 msec.
```

Ilustración 7 Inserción de Datos

```
Query  Query History
10  ('Orquesta Ciudad de Mexico', 4.6, 'Ciudad de Mexico', 'Mexico', 1960),
11  ('Filarmonica Boston', 4.4, 'Boston', 'USA', 1925), ('Orquesta Los Angel
12  ('Filarmonica Montreal', 4.6, 'Montreal', 'Canada', 1965),
13  ('Orquesta Sydney', 4.2, 'Sydney', 'Australia', 1955),
14  ('Sinfonica Melbourne', 4.4, 'Melbourne', 'Australia', 1960), ('Filarmon
15  ('Sinfonica Estocolmo', 4.6, 'Estocolmo', 'Suecia', 1925),
16  ('Filarmonica Copenhague', 4.4, 'Copenhague', 'Dinamarca', 1910),
17  ('Orquesta Helsinki', 4.3, 'Helsinki', 'Finlandia', 1935),
18  ('Sinfonica Roma', 4.7, 'Roma', 'Italia', 1905),
19  ('Filarmonica Milan', 4.5, 'Milan', 'Italia', 1915),
20  ('Orquesta Venecia', 4.2, 'Venecia', 'Italia', 1920),
21  ('Sinfonica Lisboa', 4.3, 'Lisboa', 'Portugal', 1940),
22  ('Filarmonica Amsterdam', 4.6, 'Amsterdam', 'Paises Bajos', 1930), ('Orq
23

Data Output  Messages  Notifications

INSERT 0 30

Query returned successfully in 71 msec.
```

Ilustración 8 Inserción de Datos

Verificación de la Correcta Inserción de Datos

Query | Query History | Scratch Pad

```
SELECT * FROM public.orchestras
ORDER BY id ASC LIMIT 100
```

Data Output | Messages | Notifications

Showing rows: 1 to 90 Page No: 1 of 1

	id [PK] integer	name character varying (100)	rating numeric (3,2)	city_origin character varying (50)	country_origin character varying (50)	year integer
1	1	Sinfonica Quito	4.50	Quito	Ecuador	1970
2	2	Filarmonica NY	4.80	New York	USA	1930
3	3	Orquesta Paris	4.20	Paris	Francia	1900
4	4	Sinfonica Madrid	4.60	Madrid	Espana	1980
5	5	Filarmonica Berlin	4.90	Berlin	Alemania	1920
6	6	Orquesta Viena	4.70	Viena	Austria	1890
7	7	Sinfonica Tokio	4.30	Tokio	Japon	1965
8	8	Filarmonica Londres	4.80	Londres	Reino Unido	1950
9	9	Orquesta Moscu	4.40	Moscu	Rusia	1935

Ilustración 9 Verificación de Datos

2.6.4 Ejercicios de Subconsultas a la Base Creada

Ejercicio 1.- Seleccionar orquestas con ciudad de origen donde se realizó un concierto en 2013

Ejercicio: Seleccione los nombres de todas las orquestas que tienen la misma ciudad de origen que cualquier ciudad en la que alguna orquesta actuó en 2013.

	name character varying (32)
1	Sinfonica Quito
2	Filarmonica NY
3	Orquesta Paris
4	Filarmonica Berlin
5	Sinfonica Tokio
6	Filarmonica Londres
7	Orquesta Moscu
8	Filarmonica Buenos Aires
9	Orquesta Ciudad de Mexico
10	Sinfonica Chicago
11	Orquesta Los Angeles
12	Filarmonica Montreal
13	Orquesta Sydney
14	Sinfonica Melbourne
15	Filarmonica Auckland
16	Sinfonica Estocolmo
17	Filarmonica Copenhague
18	Orquesta Helsinki

Ilustración 10 Resultado de la Primera Subconsulta

Explicación de la solución: Nuestro objetivo es seleccionar nombres de orquestas que cumplan una condición, por lo que comenzamos con SELECT name FROM orchestras. Luego, la condición se aplicará a la city_origincolumna, como se indica en las instrucciones. Queremos seleccionar sólo las orquestas cuya ciudad de origen pertenece al grupo de ciudades donde se realizaron conciertos en el año 2013. Para crear esta condición en la WHEREcláusula, utilizamos la subconsulta SQL. Creemos una (sub)consulta que seleccione todas las ciudades donde se realizaron conciertos en 2013: S ELECT city FROM concerts WHERE year = 2013. Devuelve una columna que contiene los nombres de las ciudades. Para garantizar que la ciudad de origen pertenezca a las ciudades devueltas por la subconsulta, utilizamos el INoperador .

Ejercicio 2: Seleccionar miembros que pertenezcan a orquestas de alto nivel

Ejercicio: Seleccione los nombres y posiciones (es decir, instrumento tocado) de todos los miembros de la orquesta que tengan más de 10 años de experiencia y que no pertenezcan a orquestas con una calificación inferior a 8.0.

Query | Query History

```
1  SELECT m.name, m.position FROM members m
2  JOIN orchestras o ON m.orchestra_id = o.id WHERE m.experience > 10
3  AND o.rating >= 8.0;
4
```

Data Output | Messages | Notifications

Showing rows: 1 to 7 | Page No:

	name character varying (100)	position character varying (50)
1	Pedro Solano	Trompeta
2	Elena Robles	Arpa
3	Javier Perez	Contrabajo
4	Marta Rios	Clarinete
5	Director Maestro	Direccion
6	Solista Estrella	Piano
7	Gerente Orquesta	Gestion

Ilustración 11 Resultado SubConsulta 2

Explicacion de la Solución: a consulta principal selecciona el nombre y la posición de la tabla members. Para conectar a los miembros con la calificación de su orquesta, usamos una cláusula JOIN para unir members (m) con orchestras (o) usando la llave foránea orchestra_id y la llave primaria id. Finalmente, la cláusula WHERE aplica los dos filtros solicitados.

Ejercicio 3: Seleccionar miembros que ganen más que los violinistas

Ejercicio: Mostrar el nombre y puesto de los miembros de la orquesta que ganan más que el salario promedio de todos los violinistas.

	name character varying (64)	position character varying (32)	wage integer
1	Anna Smith	Piano	2000
2	Marie Dubois	Cello	1500
3	Sophie Muller	Flauta	1800
4	Kenji Tanaka	Percusion	1600
5	Emily Brown	Clarinete	1400
6	David Wilson	Oboe	1700
7	Olga Ivanova	Viola	1500
8	Miguel Hernandez	Fagot	1400
9	Sarah Johnson	Saxofon	1600
10	Tom White	Percusion	1500
11	Linda Martinez	Piano	1800
12	Robert King	Trombon	1400
13	Isabella Rossi	Flauta	1500
14	Emma Taylor	Cello	1600
15	Noah Brown	Oboe	1500
16	Olivia Wilson	Clarinete	1400
17	Sophia Lee	Violin	1600

Ilustración 12 Resultado Subconsulta 3

Explicación de la Solución: Esta subconsulta utiliza la función de agregación AVG() en la columna wage y la filtra por aquellos registros donde el position (instrumento) sea 'violin', utilizando ILIKE para asegurar que la búsqueda sea insensible a mayúsculas y minúsculas. Esta subconsulta devuelve un único valor (el promedio salarial).

Ejercicio 4: Seleccione orquestas de alta calificación más nuevas que la orquesta de cámara

Ejercicio: Mostrar los nombres de las orquestas que se crearon después de la 'Orquesta de Cámara' y que tienen una calificación mayor a 7.5.

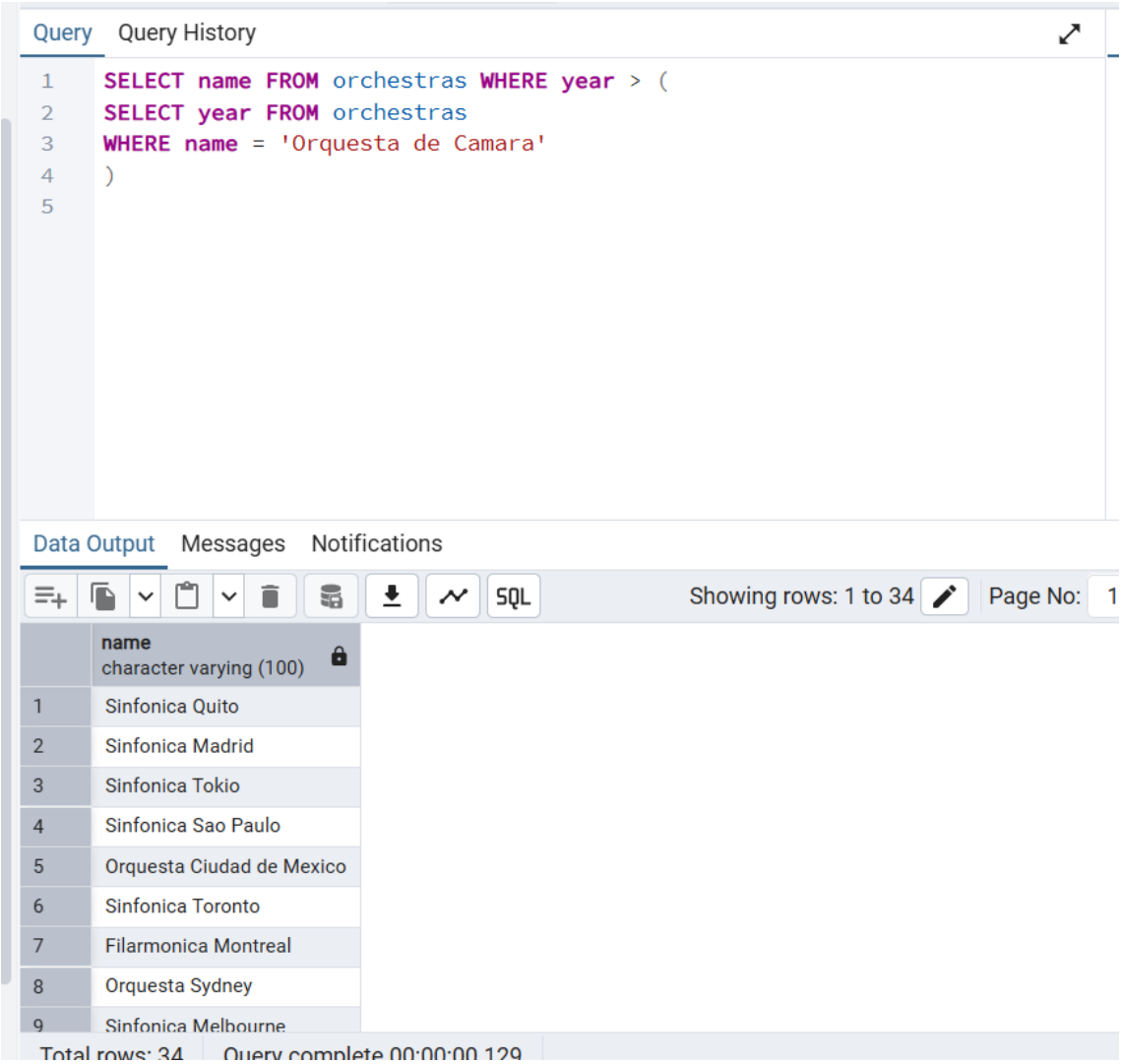


Ilustración 13 Resultado SubConsulta 4

Explicación de la solución: El requisito principal es comparar el año de fundación de todas las orquestas con el año específico en que se creó una orquesta en particular: 'Orquesta de Camara'. Para obtener este año de forma dinámica, utilizamos una subconsulta escalar (una que devuelve un solo valor): SQL SELECT year FROM orchestras WHERE name = 'Orquesta de Camara' [cite: 302, 303, 304] Esta subconsulta consulta la tabla orchestras y aísla el valor del year de la orquesta con ese nombre específico.

Ejercicio 5: Seleccionar intérpretes en grandes orquestas

Ejercicio: Muestre el nombre y el número de miembros de cada orquesta que tenga más miembros que el promedio de miembros de todas las orquestas en la tabla.

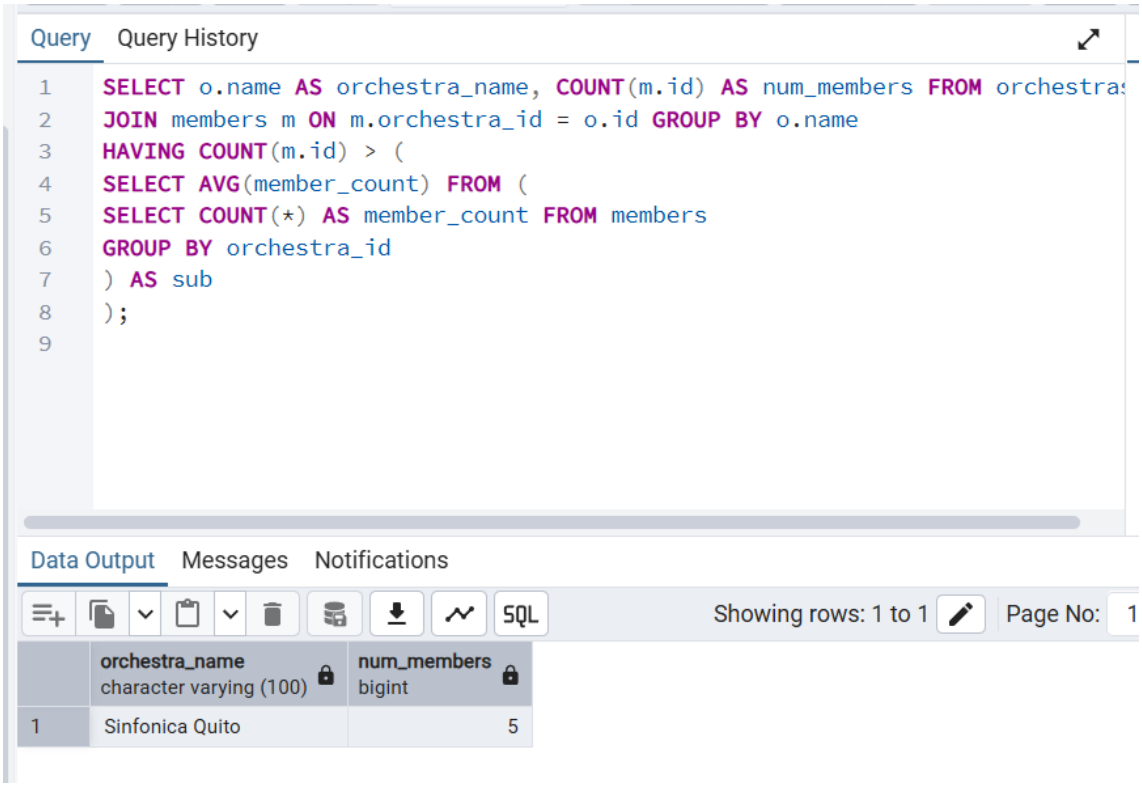


Ilustración 14 Resultado de Subconsulta 5

Explicación de la Solución: El objetivo es listar orquestas cuyo recuento de miembros supera el promedio de miembros por orquesta. La consulta principal usa JOIN y GROUP BY para contar los miembros de cada orquesta. El filtro se aplica en la cláusula HAVING, ya que opera sobre el resultado agrupado. Dentro del HAVING, se emplea una doble subconsulta anidada para calcular el promedio: la subconsulta interna calcula el recuento de miembros por orquesta, y la subconsulta externa calcula el promedio (AVG) de esos recuentos.

2.7 Resultados obtenidos

La realización de los ejercicios de subconsultas en PostgreSQL permitió la validación práctica de diversas técnicas de anidamiento de consultas. Específicamente, se comprobó la eficacia de las subconsultas escalares (Ejercicios 3 y 4) para obtener un único valor de referencia (el salario promedio de violinistas o el año de fundación de una orquesta específica) y usarlo directamente en la cláusula WHERE de la consulta principal. Así mismo, se demostró el uso del operador IN (Ejercicio 1) para filtrar la tabla principal basándose en un conjunto de resultados devuelto por la subconsulta. Finalmente, el Ejercicio 5 fue crucial para dominar las subconsultas anidadas de doble nivel y el uso de HAVING, logrando comparar el tamaño de cada grupo (orquesta) con la media calculada de todos los grupos, lo que confirma el entendimiento de las funciones de agregación y el filtrado avanzado. En conjunto, las pruebas evidenciaron el dominio en la construcción de consultas complejas y eficientes.

2.8 Habilidades blandas empleadas en la práctica

☐ Liderazgo
☐ Trabajo en equipo
☐ Comunicación asertiva
☐ La empatía
☒ Pensamiento crítico
☐ Flexibilidad

☒ La resolución de conflictos
☒ Adaptabilidad
☒ Responsabilidad

Durante el desarrollo de la práctica se aplicaron distintas habilidades blandas que facilitaron la correcta implementación y el análisis de las consultas SQL avanzadas.

El pensamiento crítico fue clave para analizar la estructura de las tres tablas y determinar la lógica anidada necesaria en cada ejercicio, especialmente al definir qué parte del problema debía resolver la subconsulta (como calcular un promedio o encontrar un año de referencia).

La resolución de conflictos fue necesaria al enfrentar y depurar errores sintácticos en la definición de las subconsultas y al ajustar los tipos de operadores (IN, > o AVG) para asegurar el retorno correcto de los datos.

La adaptabilidad se puso en práctica al ajustarse a la necesidad de construir subconsultas de doble anidamiento para el Ejercicio 5, demostrando flexibilidad ante la complejidad técnica.

La responsabilidad se reflejó en el cumplimiento de cada ejercicio y en la verificación de que los resultados de las consultas fueran lógicamente coherentes con los datos insertados, garantizando la calidad del trabajo.

2.9 Conclusiones

- Se dominó el uso del operador IN con subconsultas para filtrar registros de la tabla orchestras cuya ciudad de origen coincidía con el conjunto de ciudades devuelto por la subconsulta de concerts, validando el filtrado basado en la pertenencia a un grupo.

- Se implementaron subconsultas escalares con éxito, utilizando la función AVG() para obtener un valor de referencia (salario promedio de violinistas). Este valor se usó directamente como criterio de comparación (>) en el WHERE de la consulta principal para filtrar salarios individuales.

- Se logró la construcción de subconsultas anidadas de doble nivel para calcular el promedio de miembros por orquesta. Al aplicar este promedio en la cláusula HAVING, se demostró la capacidad de filtrar grupos (orquestas) basándose en una métrica agregada que requería dos pasos de cálculo.

2.10 Recomendaciones

- Optimización con JOIN: Para mejorar el rendimiento, se recomienda reemplazar las subconsultas con el operador IN por cláusulas JOIN siempre que sea posible. Esto generalmente resulta en planes de ejecución más rápidos en PostgreSQL.

- Uso de Subconsultas Escalares: Para obtener valores de referencia (promedios, máximos, años específicos), es vital utilizar la subconsulta escalar (AVG(), MAX(), SELECT single_column), ya que es el método más eficiente para devolver un único dato que se usa en la condición WHERE.

2.11 Anexos

Resultado Final de la Base de Datos Implementada dentro de PostgreSQL.