**INFORME DE EJERCICIOS DE SUBCONSULTAS EN POSTGRESQL**

# 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.

# INFORME DE EJERCICIOS DE SUBCONSULTAS EN POSTGRESQL

1. **PP**
2. **YY**

## Objetivos

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.

1. 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.

## Modalidad

Presencial

## Tiempo de duración

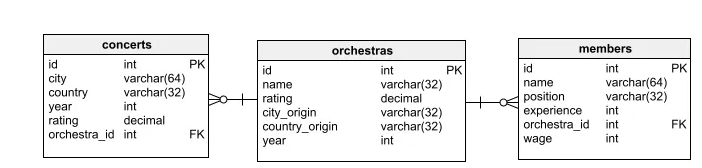
**Presenciales:** 3

**No presenciales:** 0

## 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).

## 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): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Actividades desarrolladas

### Creación de la Base de Datos

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

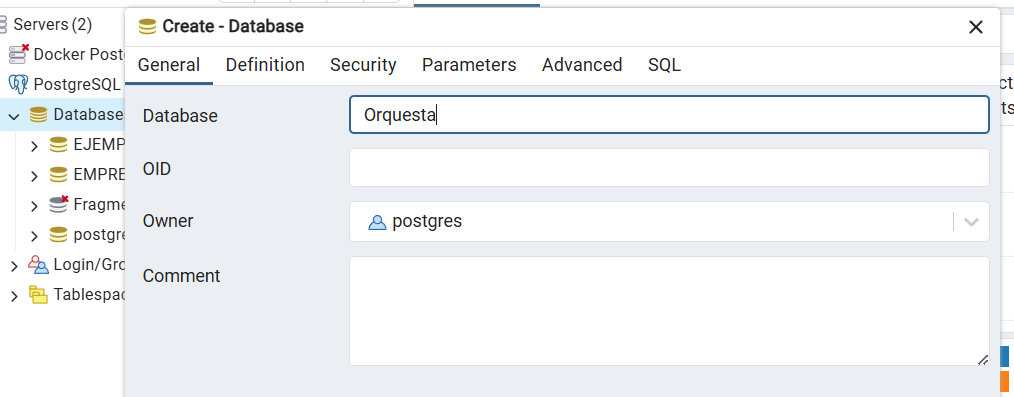
****

Ilustración 1 Creación de la Base de Datos

### 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**

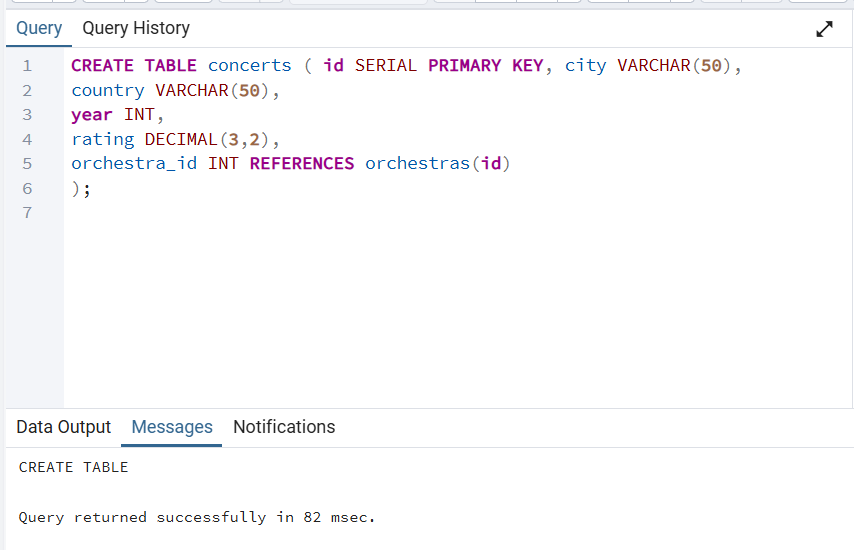
****

Ilustración 3 Creación de la Tabla Concerts

**Creación de la Tabla Members**

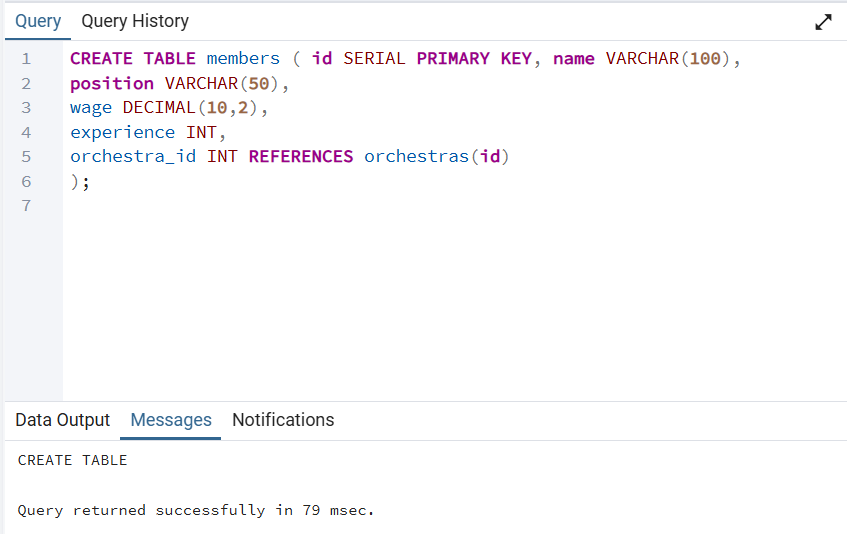
****

Ilustración 4 Creación de la Tabla Members

**Resultado Final de la Ceación Correcta de las Tablas**

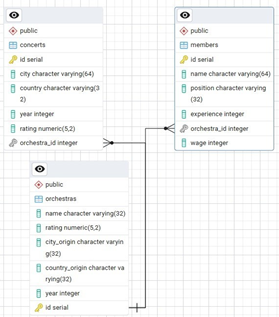


Ilustración 5 Resultado Final de la Creación Correcta de las Tablas

### Inserción de Datos en las Diferentes Tablas Creadas

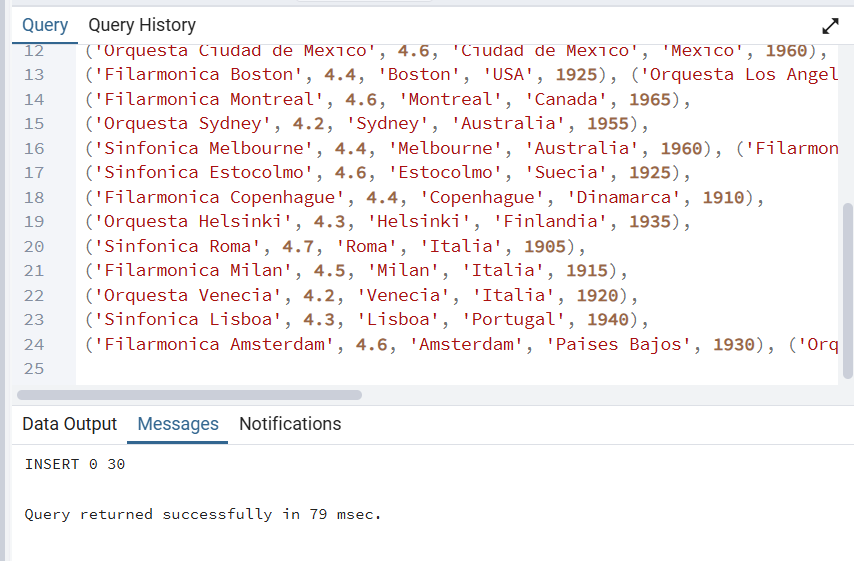
****

Ilustración 6 Inserción de Datos

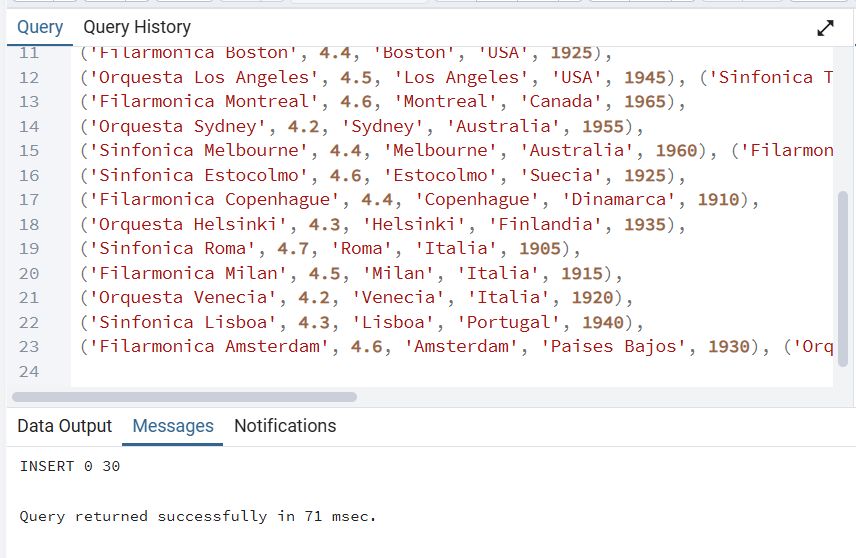
****

Ilustración 7 Inserción de Datos

****

Ilustración 8 Inserción de Datos

**Verificación de la Correcta Inserción de Datos**

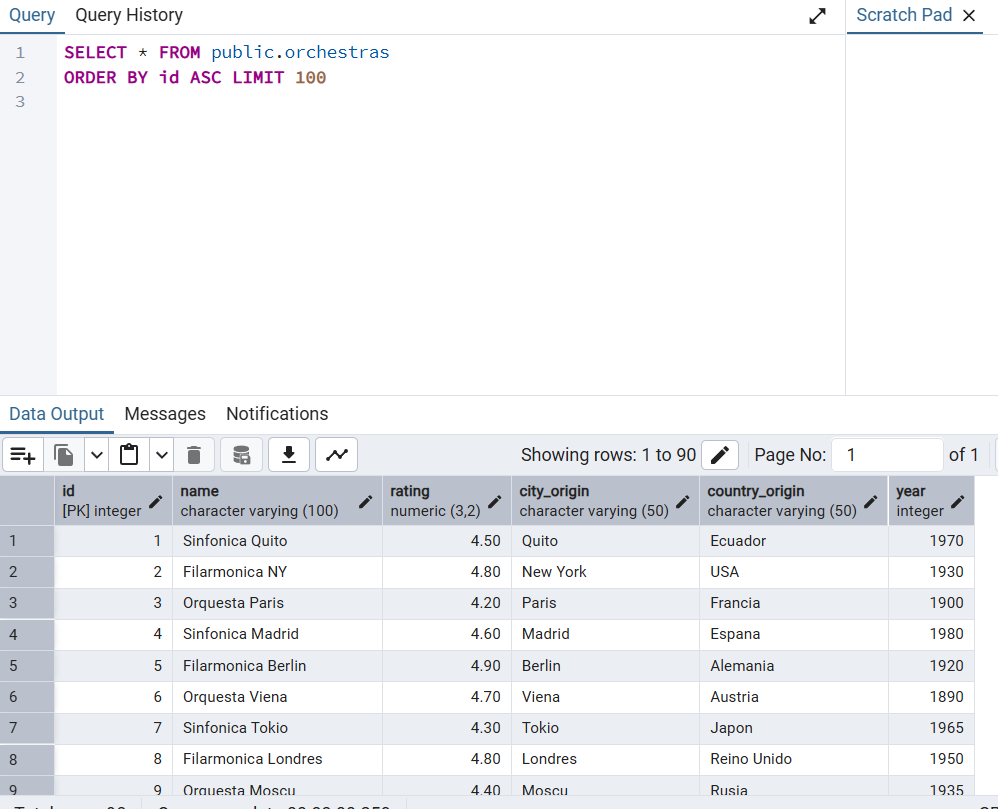
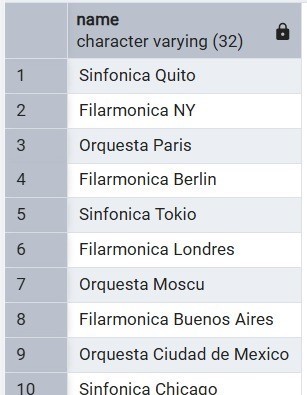
****

Ilustración 9 Verificación de Datos

### 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.



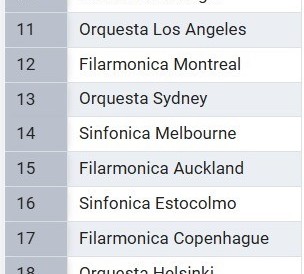


Ilustración 10 Resultado de la Primera Subconsulta

**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.

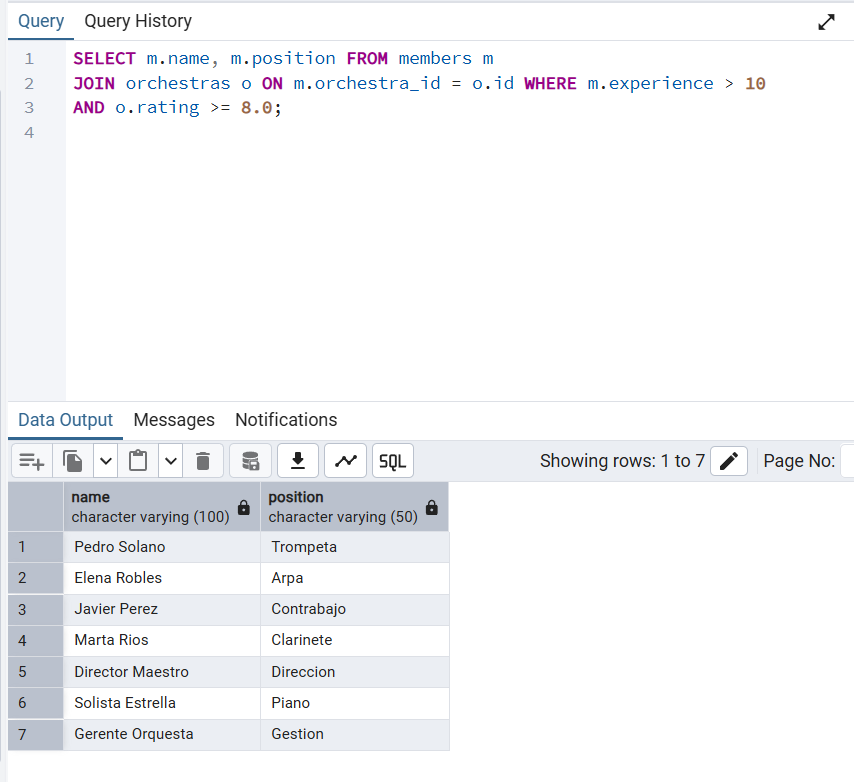
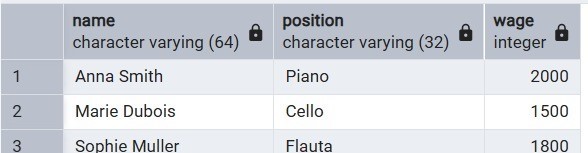


Ilustración 11 Resultado SubConsulta 2

**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.



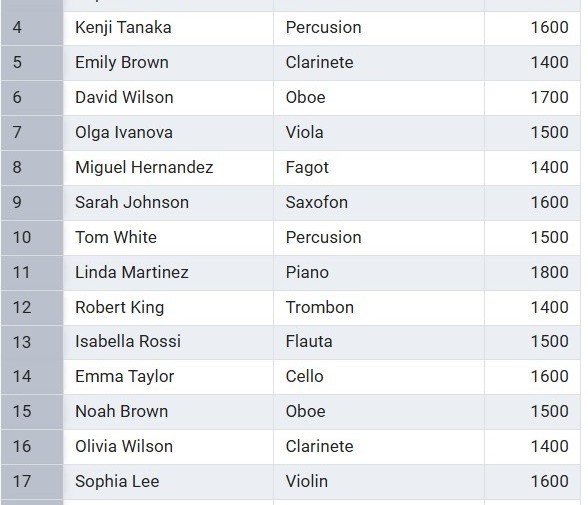


Ilustración 12 Resultado Subconsulta 3

**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.

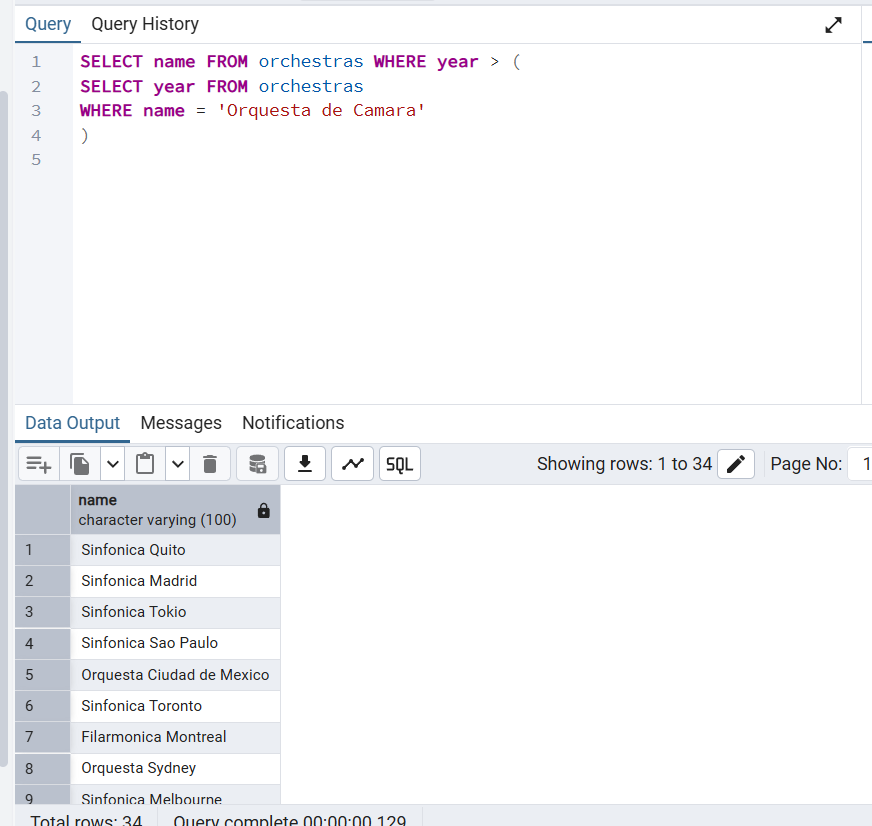


Ilustración 13 Resultado SubConsulta 4

**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.

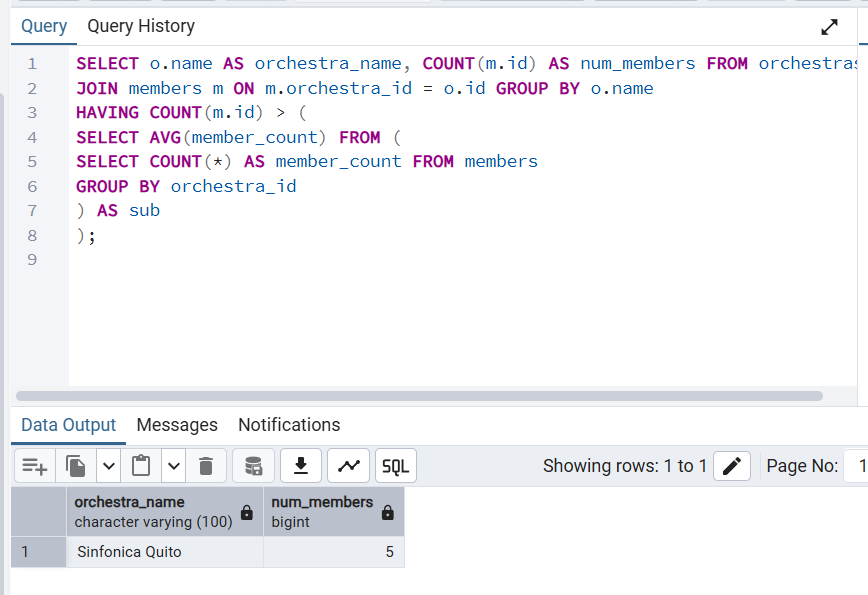


Ilustración 14 Resultado de Subconsulta 5

**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.

## 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.

## 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.

## 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.

## 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.

## Anexos

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

