*Introduccion a subconsultas*

Las subconsultas, también conocidas como "subqueries", nos permiten utilizar los resultados de una consulta dentro de otra consulta.

Veamos un ejemplo práctico.

Dada la siguiente tabla empleados



Se nos pide seleccionar a todas las personas que ganan sobre el promedio.

Este tipo de preguntas podemos contestarlas utilizando subconsultas.

La idea para contestar esto es la siguiente.

1. Calculamos el promedio SELECT avg(sueldo) FROM empleados
2. Seleccionamos todos los empleados cuyo sueldo es mayor a la consulta anterior. SELECT \* FROM empleados WHERE sueldo > (SELECT AVG(sueldo) FROM empleados)

Ejercicio

Utilizando los mismos datos de la tabla empleados, selecciona todos los registros cuyo sueldo sea menor o igual al promedio.

SELECT \* FROM empleados WHERE sueldo <= (SELECT AVG(sueldo) FROM empleados);

Tabla

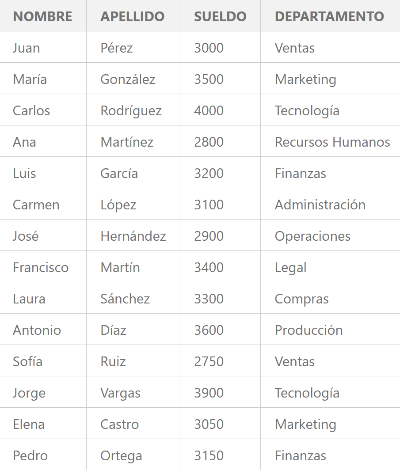
Descripción generada automáticamente

se selecciona primero todas las columnas de la tabla y la tabla misma para luego hace la sub consulta con WHERE y hacer la comparación que se pide.

*Subconsultas y where parte 1*

Dentro de las subconsultas, podemos utilizar las mismas cláusulas que hemos aprendido hasta ahora, como la cláusula WHERE. Esto significa que podemos aplicar la cláusula WHERE tanto dentro de la subconsulta como fuera de ella.

Ejercicio

Dada la siguiente **tabla empleados**

Selecciona toda la información de los registros que sean mayores al promedio del departamento de finanzas.

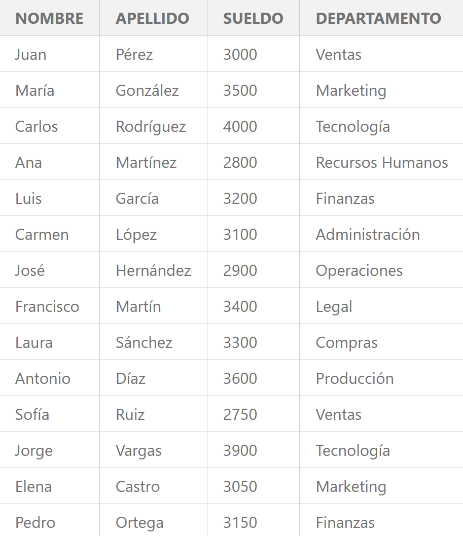
Tip:

* Se pide el promedio exclusivamente del departamento de finanzas por lo que no hay necesidad de agrupar los datos.
* Para este tipo de problema usualmente hay más de una solución.

SELECT \* from empleados WHERE sueldo > (SELECT avg(sueldo) from empleados where departamento = 'Finanzas')

Se selecciona todas las columnas de la tabla empleados, luego se hace la subconsulta donde se hace la validación del promedio de la columna sueldo de la tabla empleados y para finalizar se hace una validacion para que le promedio sea el del departamento de finanzas.

*Subconsultas y where parte 2*

Ejercicio

Utilizando los datos de la **tabla empleados**, selecciona todos los empleados cuyo sueldo sea mayor al empleado que tiene el mayor sueldo del departamento de finanzas.

select \* from empleados where sueldo > (select MAX(sueldo) from empleados where departamento = 'Finanzas')

se selecciona todas la columnas de la tabla empleados para luego agregar una comparación donde se incluye una subconsulta para encontrar el sueldo mas alto del departamento de Finanzas.

*Subconsultas y where parte 3*

Ejercicio

Se tiene la siguiente **tabla notas**:



Selecciona todos los registros superiores al promedio de nota.

SELECT \* FROM notas where notas > (SELECT AVG(notas) from notas)

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Se seleccionan todas las columnas de la tabla notas y luego se hace la comparación para mostrar todas las notas que sean mayores al promedio de la nota en general.

*Subconsultas con IN*

El operador IN es un operador muy útil en subconsultas. Para entenderlo, primero probaremos una consulta sencilla utilizandolo directamente sin subconsultas.



Queremos seleccionar todos los códigos de Argentina, Brasil, Chile o Colombia. Una forma de abordar el problema sería combinar todas las opciones con where y múltiples operadores or. Otra opción es utilizando el operador *IN* de la siguiente manera:

SELECT \*

FROM paises

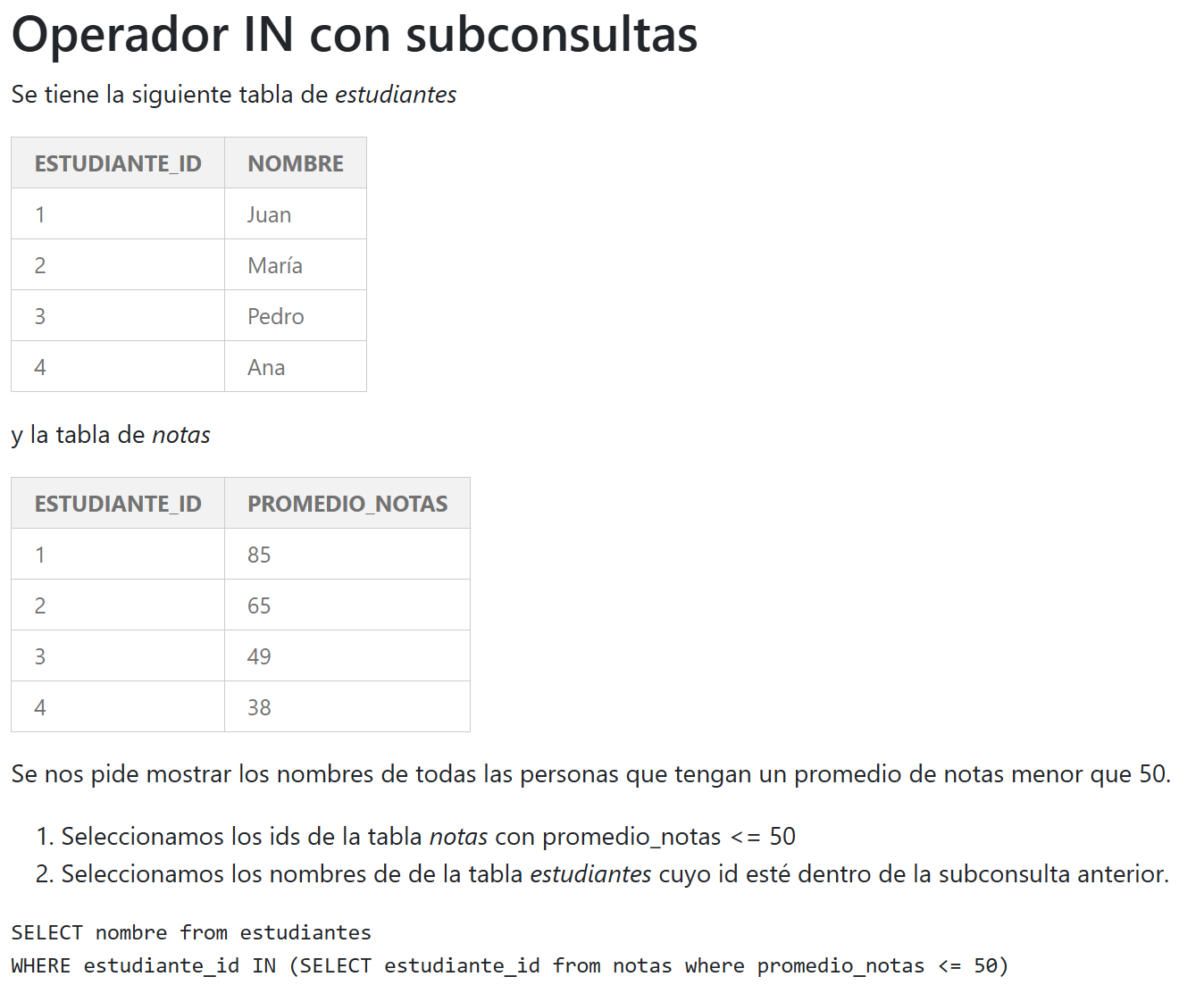
WHERE pais IN ('Argentina', 'Brasil', 'Chile', 'Colombia')

De la misma forma podemos hacer una consulta como la siguiente:

SELECT \*

FROM table

WHERE columna IN (SELECT \* from otra\_tabla)



Ejercicio

Se tiene una tabla *estudiantes* con un código y un nombre

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Muestra los nombres de todos los estudiantes que tengan un promedio de notas sobre 50

Tip 1: No necesitas agrupar ni promediar ni contar Tip 2: Hay más de una forma de resolver este ejercicio, no te adelantes a joins e intenta resolverlo utilizando subqueries

SELECT nombre from estudiantes

WHERE estudiante\_id IN (SELECT estudiante\_id from promedios where promedio\_notas > 50)

Se selecciona los nombres desde la tabla estudiantes, para luego hacer una validación con WHERE e IN con una subconsulta, se incluye la columna estudiante\_id de la tabla promedios y se hace la subconsulta para validar los promedios que estén sobre 50.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

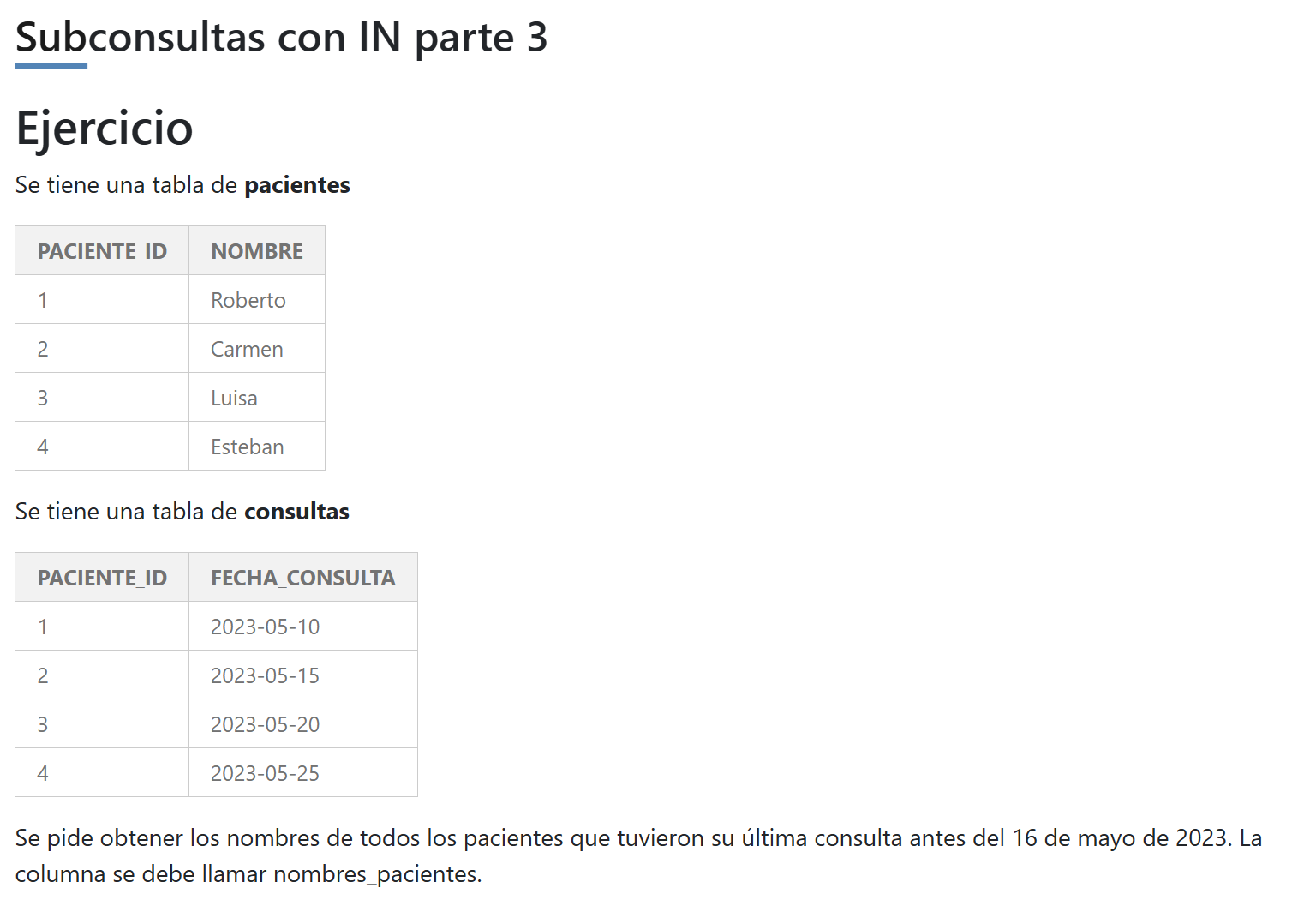
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

SELECT nombre as nombres\_seleccionados from libros

where libro\_id IN (SELECT libro\_id from valoraciones where valoracion\_promedio >4)

se selecciona la columna nombre de la tabla libros y se le asigna un alias luego se hace la validación con where seleccionando la columna libro\_id de la tabla valoraciones y se incluye el IN para hacer una subconsulta donde se selecciona las columnas libro\_id de la tabla valoraciones y además se hace la validación de que el promedio sea sobre 4.



select nombre as nombres\_pacientes from pacientes

where paciente\_id IN (select paciente\_id from consultas where fecha\_consulta < '2023-05-16')

*Subconsultas en el FROM*

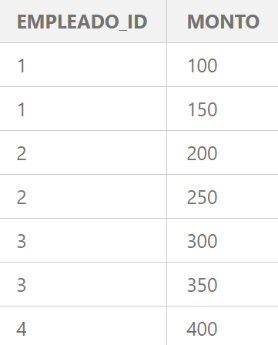
Las subconsultas, también conocidas como "subqueries", nos permiten utilizar los resultados de una consulta dentro de otra consulta. En los ejercicios anteriores utilizamos las subconsultas dentro de la claúsula *WHERE*, pero también es posible utilizarlas dentro de otras claúsulas. En este ejercicio abordaremos como utilizarla dentro de FROM

Una subconsulta en el FROM tiene la siguiente forma.

SELECT \* FROM (SELECT \* FROM tabla1)

En este caso no parece tan útil ya que simplemente estamos seleccionando lo mismo, pero veamos un caso donde si sería necesario.

Se tiene la tabla *ventas* que tiene el código de vendedor y el monto de cada venta realizada. Nos piden saber cuanto es el promedio total vendido.



Para esto primero necesitamos sumar los montos por vendedor y luego sobre *estos resultados* sacamos el promedio de las ventas.

SELECT AVG(total\_venta) as promedio\_ventas

FROM (SELECT empleado\_id, SUM(monto) as total\_venta

FROM ventas

GROUP BY empleado\_id )

¿Cómo llegamos a esto?

Si queremos saber los promedios, primero tenemos que saber los totales, para eso necesitamos sumar por empleado.

SELECT empleado\_id, SUM(monto) as total\_venta FROM ventas GROUP BY empleado\_id

El código anterior nos generará los siguientes resultados.

Tabla

Descripción generada automáticamente

Luego sacamos el promedio de los montos de esta nueva tabla.

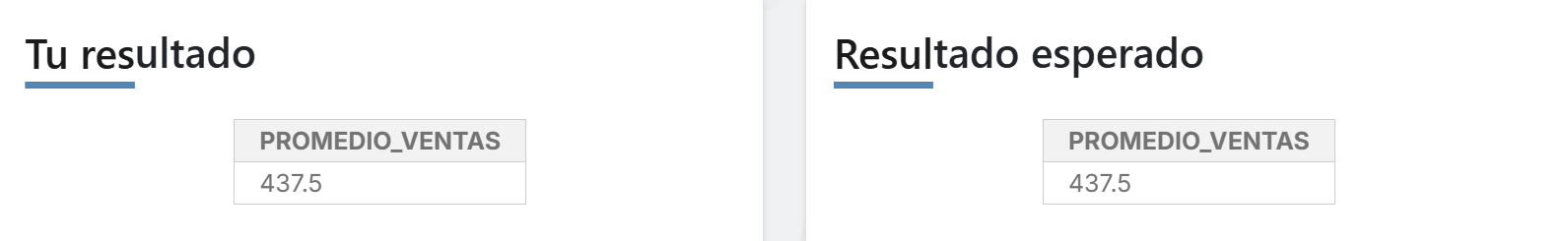
SELECT AVG(total\_venta) as promedio\_ventas FROM (SELECT empleado\_id, SUM(monto) as total\_venta FROM ventas GROUP BY empleado\_id )

Este tipo de ejercicio suele ser un poco mas complejo de pensar y escribir y requiere de cierta práctica dominar, por lo mismo el primer ejercicio consistirá en escribir el mismo query. Intenta hacerlo sin mirar la respuesta.

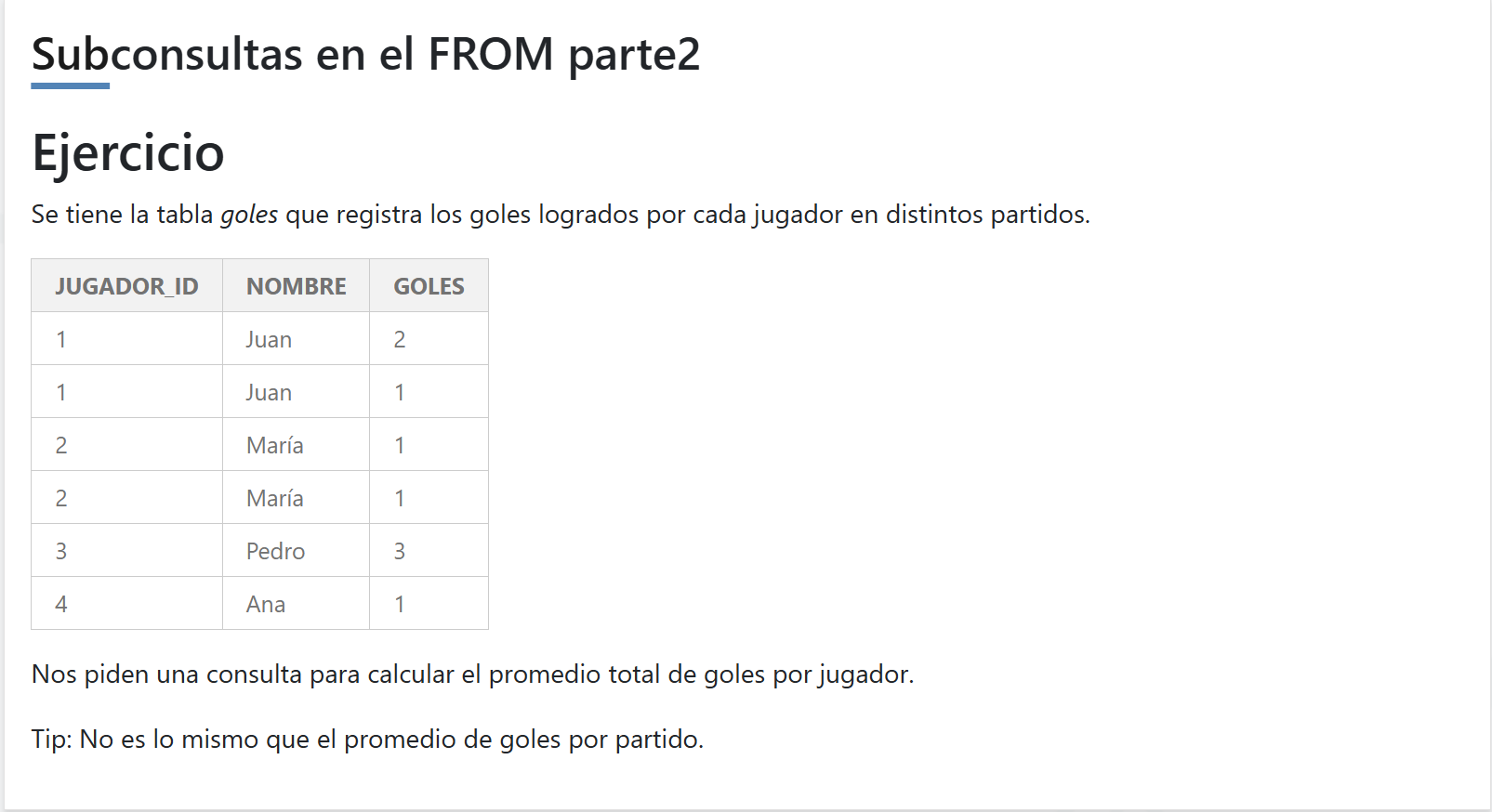
Ejercicio

Se tiene la tabla *ventas* que tiene el código de vendedor y el monto de la venta. Nos piden saber cuanto es el promedio total vendido. El resultado debe estar en la columna promedio\_ventas

SELECT AVG(total\_venta) AS promedio\_ventas FROM

(SELECT empleado\_id, SUM(monto) as total\_venta from ventas group by empleado\_id)

La solucion a traves de una subconsulta es primero hacer la suma de todos los empleados, seleccionando la columna del id del empleado y luego hacer la suma y colocarle un alias para después agrupar ese resultado y con eso se puede sacar el promedio desde el resultado de la suma, en la lógica primero se hace la subconsulta en donde va la operación de la suma, en este caso aparece al final del código y después se hace la operación del promedo con AVG. Esta subconsulta se hace con la clausula FROM.



SOLUCION

SELECT AVG(total\_goles) AS PROMEDIO\_GOLES FROM

(SELECT jugador\_id, SUM(goles) AS total\_goles FROM goles GROUP BY jugador\_id)

La logica para solucionar esta consulta es parecida a la anterior, primero se hace la lógica para sumar los goles y luego hacer el promedio, para esto se selecciona la columna jugador\_id y la de goles y en esta se usa la función SUM para sumar los goles pro cada jugador y darle un alias y luego para que se agrupen y luego se saca el promedio con AVG incluyendo en su operación el alias que se creo para sumar los goles.