

Explicación de solución punto 3

Por: Sebastian Carmona Estrada

Fecha: 06/02/2024

Prueba: PRUEBA DE SQL

“Preguntas: ➤ Seleccione nombre, apellido y salario de todos los empleados. ➤ Seleccione nombre, apellido y salario de todos los empleados que ganen más de 4 millones. ➤ Cuente los empleados por sexo. ➤ Seleccione los empleados que no han hecho solicitud de vacaciones. ➤ Seleccione los empleados que tengan más de una solicitud de vacaciones y muestre cuantas solicitudes tienen los que cumplen. ➤ Determine el salario promedio de los empleados. ➤ Determine la cantidad de días promedio solicitados de vacaciones por cada empleado. ➤ Seleccione el empleado que mayor cantidad de días de vacaciones ha solicitado, muestre el nombre, apellido y cantidad de días totales solicitados. ➤ Consulte la cantidad de días aprobados y rechazados por cada empleado, en caso de no tener solicitudes mostrar 0.”

Solución:

```
19 SELECT nombre, apellido, salario FROM EMPLEADO; /* Seleccionar todos los empleados*/
20
21 SELECT nombre, apellido, salario FROM EMPLEADO WHERE salario > 4000000; /* Seleccionar todos los empleados que ganen mas de 4 millones*/
22
23 SELECT sexo, COUNT(*) as cantidad_empleados FROM EMPLEADO GROUP by sexo; /* Contar los empleados por sexo*/
24
25 SELECT * from EMPLEADO where id NOT IN (SELECT id_emp FROM VACACIONES); /* Seleccionar empleados que no han solicitado vacaciones */
26
27 SELECT e.*, COUNT(v.ID) as numero_solicitudes /* Selecciona los empleados que han solicitado vacaciones más de una vez y cuenta las veces que han solicitado vacaciones*/
28 FROM EMPLEADO e
29 JOIN VACACIONES v on e.ID = v.id_emp
30 GROUP by e.id
31 HAVING COUNT(v.id) > 1;
32
33
34 SELECT AVG(salario) as salario_promedio from EMPLEADO; /* Calcula el valor promedio del salario entre los empleados*/
35
36 SELECT e.id, e.nombre, e.apellido, COALESCE(AVG(v.cantidad_dias), 0) AS dias_vacaciones_promedio /* Calcula el promedio de los dias de vacaciones solicitados por empleado*/
37 FROM EMPLEADO e
38 LEFT JOIN VACACIONES v ON e.id = v.id_emp
39 GROUP BY e.id, e.nombre, e.apellido;
40
41 SELECT e.nombre, e.apellido, SUM(v.cantidad_dias) AS dias_totales_solicitados /* Esta función encuentra el empleado que más días de vacaciones ha solicitado en total y muestr
42 FROM EMPLEADO e
43 JOIN VACACIONES v ON e.id = v.id_emp
44 GROUP BY e.nombre, e.apellido
45 ORDER BY dias_totales_solicitados DESC
46 LIMIT 1;
47
48 SELECT e.id, e.nombre, e.apellido, /* Muestra la cantidad de días de vacaciones aprobados y rechazados para cada empleado*/
49 COALESCE(SUM(CASE WHEN v.estado = 'A' THEN v.cantidad_dias ELSE 0 END), 0) AS dias_aprobados,
50 COALESCE(SUM(CASE WHEN v.estado = 'R' THEN v.cantidad_dias ELSE 0 END), 0) AS dias_rechazados
51 FROM EMPLEADO e
52 LEFT JOIN VACACIONES v ON e.id = v.id_emp
53 GROUP BY e.id, e.nombre, e.apellido;
54
```

Explicación:

1. **SELECT nombre, apellido, salario FROM EMPLEADO;**
 - a. Esta consulta devuelve los campos nombre, apellido y salario de la tabla EMPLEADO.
 - b. No hay ningún filtro aplicado, por lo que devuelve todos los empleados junto con su información básica.

2. **SELECT nombre, apellido, salario FROM EMPLEADO WHERE salario > 4000000;**
 - a. Aquí, se seleccionan los mismos campos que en la consulta anterior, pero con una condición adicional.
 - b. La condición salario > 4000000 filtra los resultados para incluir solo aquellos empleados cuyo salario sea mayor que 4 millones.
 - c. Esto permite identificar específicamente a los empleados que tienen un salario más alto que el umbral especificado.

3. **SELECT sexo, COUNT() as cantidad_empleados FROM EMPLEADO GROUP by sexo;**
 - a. Esta consulta agrupa los datos por el campo sexo y cuenta el número de empleados para cada categoría.
 - b. Utiliza la función de agregación COUNT() para contar el número de filas para cada valor único de sexo.
 - c. El resultado muestra cuántos empleados hay para cada género, proporcionando una visión de la distribución de género en la empresa.
 - d.

4. **SELECT from EMPLEADO where id NOT IN (SELECT id_emp FROM VACACIONES);**
 - a. Esta consulta selecciona todos los campos de la tabla EMPLEADO donde el ID del empleado no está presente en la tabla VACACIONES.
 - b. Utiliza una subconsulta para obtener los IDs de los empleados que han solicitado vacaciones (id_emp de la tabla VACACIONES).
 - c. Devuelve los empleados que no tienen registros de vacaciones en la tabla VACACIONES, lo que indica que aún no han solicitado vacaciones.

5. **SELECT e., COUNT(v.ID) as numero_solicitudes FROM EMPLEADO e JOIN VACACIONES v on e.ID = v.id_emp GROUP by e.id HAVING COUNT(v.id) > 1;**
 - a. Esta consulta selecciona todos los campos de la tabla EMPLEADO y cuenta el número de solicitudes de vacaciones para cada empleado.
 - b. Se une con la tabla VACACIONES utilizando la condición e.ID = v.id_emp.
 - c. Los resultados se agrupan por el ID del empleado (e.id) y se cuentan las solicitudes de vacaciones para cada uno.
 - d. La cláusula HAVING COUNT(v.id) > 1 filtra los resultados para incluir solo aquellos empleados que han solicitado vacaciones más de una vez.

6. SELECT AVG(salario) as salario_promedio from EMPLEADO;

- a. Esta consulta calcula el salario promedio de todos los empleados en la tabla EMPLEADO.
- b. Utiliza la función de agregación AVG() para calcular el promedio del salario.
- c. El resultado se etiqueta como salario_promedio utilizando la cláusula AS.
- d. Devuelve un solo valor que representa el salario promedio de todos los empleados.

7. SELECT e.id, e.nombre, e.apellido, COALESCE(AVG(v.cantidad_dias), 0) AS dias_vacaciones_promedio FROM EMPLEADO e LEFT JOIN VACACIONES v ON e.id = v.id_emp GROUP BY e.id, e.nombre, e.apellido;

- a. Esta consulta calcula el promedio de los días de vacaciones solicitados por empleado.
- b. Utiliza un LEFT JOIN para combinar la tabla EMPLEADO con la tabla VACACIONES en función del ID del empleado (e.id = v.id_emp).
- c. La función de agregación AVG() calcula el promedio de la cantidad de días de vacaciones (v.cantidad_dias) para cada empleado.
- d. Utiliza la función COALESCE() para manejar los casos en los que un empleado no tiene registros de vacaciones, devolviendo cero en lugar de NULL.
- e. Los resultados se agrupan por el ID del empleado, el nombre y el apellido.

8. SELECT e.nombre, e.apellido, SUM(v.cantidad_dias) AS dias_totales_solicitados FROM EMPLEADO e JOIN VACACIONES v ON e.id = v.id_emp GROUP BY e.nombre, e.apellido ORDER BY dias_totales_solicitados DESC LIMIT 1;

- a. Esta consulta encuentra al empleado que ha solicitado el mayor número total de días de vacaciones.
- b. Utiliza un JOIN para combinar la tabla EMPLEADO con la tabla VACACIONES en función del ID del empleado.
- c. Utiliza la función de agregación SUM() para sumar la cantidad total de días de vacaciones (v.cantidad_dias) solicitados por cada empleado.
- d. Los resultados se agrupan por el nombre y el apellido del empleado.
- e. Luego, se ordenan en orden descendente según el total de días solicitados y se limita el resultado a una sola fila utilizando LIMIT 1.

9. SELECT e.id, e.nombre, e.apellido, COALESCE(SUM(CASE WHEN v.estado = 'A' THEN v.cantidad_dias ELSE 0 END), 0) AS dias_aprobados, COALESCE(SUM(CASE WHEN v.estado = 'R' THEN v.cantidad_dias ELSE 0 END), 0) AS dias_rechazados FROM EMPLEADO e LEFT JOIN VACACIONES v ON e.id = v.id_emp GROUP BY e.id, e.nombre, e.apellido;

- a. Esta consulta muestra la cantidad de días de vacaciones aprobados y rechazados para cada empleado.
- b. Utiliza un LEFT JOIN para combinar la tabla EMPLEADO con la tabla VACACIONES en función del ID del empleado.
- c. Utiliza la cláusula CASE para sumar la cantidad de días de vacaciones aprobados y rechazados, dependiendo del estado ('A' para aprobado, 'R' para rechazado) en la tabla VACACIONES.
- d. La función SUM() suma la cantidad de días de vacaciones correspondientes a cada estado para cada empleado.
- e. Los resultados se agrupan por el ID del empleado, el nombre y el apellido.