**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:**

**Texto

Descripción generada automáticamente**

**Explicación:**

1. **SELECT nombre, apellido, salario FROM EMPLEADO;**
   1. Esta consulta devuelve los campos nombre, apellido y salario de la tabla EMPLEADO.
   2. 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;**
   1. Aquí, se seleccionan los mismos campos que en la consulta anterior, pero con una condición adicional.
   2. La condición salario > 4000000 filtra los resultados para incluir solo aquellos empleados cuyo salario sea mayor que 4 millones.
   3. 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;**
   1. Esta consulta agrupa los datos por el campo sexo y cuenta el número de empleados para cada categoría.
   2. Utiliza la función de agregación COUNT() para contar el número de filas para cada valor único de sexo.
   3. El resultado muestra cuántos empleados hay para cada género, proporcionando una visión de la distribución de género en la empresa.
4. **SELECT from EMPLEADO where id NOT IN (SELECT id\_emp FROM VACACIONES);**
   1. Esta consulta selecciona todos los campos de la tabla EMPLEADO donde el ID del empleado no está presente en la tabla VACACIONES.
   2. Utiliza una subconsulta para obtener los IDs de los empleados que han solicitado vacaciones (id\_emp de la tabla VACACIONES).
   3. 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;**
   1. Esta consulta selecciona todos los campos de la tabla EMPLEADO y cuenta el número de solicitudes de vacaciones para cada empleado.
   2. Se une con la tabla VACACIONES utilizando la condición e.ID = v.id\_emp.
   3. Los resultados se agrupan por el ID del empleado (e.id) y se cuentan las solicitudes de vacaciones para cada uno.
   4. 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;**
   1. Esta consulta calcula el salario promedio de todos los empleados en la tabla EMPLEADO.
   2. Utiliza la función de agregación AVG() para calcular el promedio del salario.
   3. El resultado se etiqueta como salario\_promedio utilizando la cláusula AS.
   4. 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;**
   1. Esta consulta calcula el promedio de los días de vacaciones solicitados por empleado.
   2. Utiliza un LEFT JOIN para combinar la tabla EMPLEADO con la tabla VACACIONES en función del ID del empleado (e.id = v.id\_emp).
   3. La función de agregación AVG() calcula el promedio de la cantidad de días de vacaciones (v.cantidad\_dias) para cada empleado.
   4. Utiliza la función COALESCE() para manejar los casos en los que un empleado no tiene registros de vacaciones, devolviendo cero en lugar de NULL.
   5. 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;**
   1. Esta consulta encuentra al empleado que ha solicitado el mayor número total de días de vacaciones.
   2. Utiliza un JOIN para combinar la tabla EMPLEADO con la tabla VACACIONES en función del ID del empleado.
   3. Utiliza la función de agregación SUM() para sumar la cantidad total de días de vacaciones (v.cantidad\_dias) solicitados por cada empleado.
   4. Los resultados se agrupan por el nombre y el apellido del empleado.
   5. 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;**
   1. Esta consulta muestra la cantidad de días de vacaciones aprobados y rechazados para cada empleado.
   2. Utiliza un LEFT JOIN para combinar la tabla EMPLEADO con la tabla VACACIONES en función del ID del empleado.
   3. 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.
   4. La función SUM() suma la cantidad de días de vacaciones correspondientes a cada estado para cada empleado.
   5. Los resultados se agrupan por el ID del empleado, el nombre y el apellido.