

Solución del Punto 3 – Prueba SQL

Para la solución de esta prueba se usará la herramienta de SQLITE ONLINE.

- 1) Seleccione nombre, apellido y salario de todos los empleados.

```
1 SELECT NOMBRE, APELLIDO, SALARIO
2 FROM EMPLEADO
```

NOMBRE	APELLIDO	SALARIO
JUAN	PELAEZ	3500000
ANDRES	GARCIA	5500000
LAURA	PEREZ	2500000
PEPE	MARTINEZ	3800000
MARGARITA	CORRALES	4500000

En esta consulta llamamos a los 3 campos que nos piden de la tabla EMPLEADO.

- 2) Seleccione nombre, apellido y salario de todos los empleados que ganen más de 4 millones.

```
1 SELECT NOMBRE, APELLIDO, SALARIO
2 FROM EMPLEADO
3 WHERE SALARIO > 4000000;
```

NOMBRE	APELLIDO	SALARIO
ANDRES	GARCIA	5500000
MARGARITA	CORRALES	4500000

Esta consulta es similar a la anterior, sin embargo, realizamos un filtro usando el WHERE para identificar los empleados con salarios mayores a 4 millones.

3) Cuento los empleados por sexo.

```
1 SELECT SEXO, COUNT(*) AS EMPLEADOS
2 FROM EMPLEADO
3 GROUP BY SEXO;
```

SEXO	EMPLEADOS
F	2
M	3

En esta consulta como debemos agrupar por sexo, seleccionamos el campo sexo y usamos la función COUNT(*) para que nos muestre la cantidad de registros en base a la agrupación.

4) Seleccione los empleados que no han hecho solicitud de vacaciones.

```
1 SELECT E.ID, CONCAT(E.NOMBRE, ' ', E.APELLIDO) AS NOMBRE
2 FROM EMPLEADO E
3 LEFT JOIN VACACIONES V ON E.ID = V.ID_EMP
4 WHERE V.ID_EMP IS NULL;
```

ID	NOMBRE
4	PEPE MARTINEZ
5	MARGARITA CORRALES

En esta consulta debemos realizar un LEFT JOIN, es decir, traernos todos los datos de la tabla de la izquierda y los datos coincidentes de la derecha y los que no coincidan serán valores nulos, como podemos darnos cuenta, los valores nulos corresponden a los empleados que no han hecho solicitud de vacaciones ya que el id de los empleados no se encontrará en la tabla VACACIONES. Procedemos a mostrar la columna de ID y una concatenación del NOMBRE y APELLIDO de la tabla EMPLEADO, realizamos el LEFT JOIN donde la relación se hará por la columna ID de la tabla EMPLEADO y la columna ID_EMP de la tabla VACACIONES y por último filtraremos los registros nulos de la columna ID_EMP.

- 5) Seleccione los empleados que tengan más de una solicitud de vacaciones y muestre cuantas solicitudes tienen los que cumplen.

```
1 SELECT E.ID, CONCAT(E.NOMBRE, ' ', E.APELLIDO) AS NOMBRE, COUNT(V.ID) AS NUMERO_SOLICITUDES
2 FROM EMPLEADO E
3 INNER JOIN VACACIONES V ON E.ID = V.ID_EMP
4 GROUP BY E.ID, NOMBRE
5 HAVING COUNT(V.ID) > 1;
```

ID	NOMBRE	NUMERO_SOLICITUDES
2	ANDRES GARCIA	3
3	LAURA PEREZ	2

Esta consulta tiene una estructura un poco similar a la anterior pero acá debemos contabilizar las solicitudes de vacaciones por empleado, donde igual a la consulta anterior llamamos a los campos ID y a la concatenación para obtener el NOMBRE de la tabla EMPLEADO, pero además usamos la función COUNT(V.ID) para contabilizar la cantidad de solicitudes de la tabla VACACIONES y la cual se verá condicionada por la clausula HAVING con la condición de filtrar solo a los empleados que tengan más de 1 solicitud de vacaciones. Se usa el INNER JOIN como unión porque necesitamos solo los registros que tengan coincidencia en ambas tablas.

- 6) Determine el salario promedio de los empleados.

```
1 SELECT AVG(SALARIO) AS SALARIO_PROMEDIO
2 FROM EMPLEADO;
```

SALARIO_PROMEDIO
3960000

En esta consulta procedemos a usar la función AVG() para calcular el promedio de los salarios de los empleados de la tabla EMPLEADO.

7) Determine la cantidad de días promedio solicitados de vacaciones por cada empleado.

```
1 SELECT V.ID_EMP, CONCAT(E.NOMBRE, ' ', E.APELLIDO) AS NOMBRE, AVG(V.CANTIDAD_DIAS) AS dias_promedio_vacaciones
2 FROM VACACIONES V
3 LEFT JOIN EMPLEADO E ON V.ID_EMP = E.ID
4 GROUP BY V.ID_EMP, NOMBRE;
```

ID_EMP	NOMBRE	dias_promedio_vacaciones
1	JUAN PELAEZ	14
2	ANDRES GARCIA	11.333333333333334
3	LAURA PEREZ	8

En esta consulta se realiza un LEFT JOIN de la tabla VACACIONES hacia la tabla EMPLEADO, por ende ya se omiten los empleados que no han solicitado vacaciones debido a la naturaleza del LEFT JOIN, seguido de esto llamamos a los campos que se mostraran donde es relevante recalcar el uso de la función AVG() para el calculo del promedio de los días de vacaciones.

8) Seleccione el empleado que mayor cantidad de días de vacaciones ha solicitado, muestre el nombre, apellido y cantidad de días totales solicitados.

```
1 SELECT E.NOMBRE, E.APELLIDO, SUM(V.CANTIDAD_DIAS) AS TOTAL_DIAS_SOLICITADOS
2 FROM EMPLEADO E
3 INNER JOIN VACACIONES V ON E.ID = V.ID_EMP
4 GROUP BY E.ID, E.NOMBRE, E.APELLIDO
5 ORDER BY TOTAL_DIAS_SOLICITADOS DESC
6 LIMIT 1;
```

NOMBRE	APELLIDO	TOTAL_DIAS_SOLICITADOS
ANDRES	GARCIA	34

En esta consulta realizamos un INNER JOIN entre ambas tablas para obtener solo los registros que coincidan entre ellos, seleccionamos los campos a visualizar, así como la función SUM() que nos permitirá sumar la cantidad de días solicitado por empleados, como necesitamos ver a la persona que más días ha solicitado usamos ORDER BY DESC para ordenar en orden descendente y también usamos LIMIT para mostrar el primer registro que luego de ordenar, será la persona que más días ha solicitado de vacaciones.

- 9) Consulte la cantidad de días aprobados y rechazados por cada empleado, en caso de no tener solicitudes mostrar 0.

```
1 SELECT E.ID, CONCAT(E.NOMBRE, ' ', E.APELLIDO) AS NOMBRE,
2        COALESCE(SUM(CASE WHEN V.ESTADO = 'A' THEN V.CANTIDAD_DIAS ELSE 0 END), 0) AS DIAS_APROBADOS,
3        COALESCE(SUM(CASE WHEN V.ESTADO = 'R' THEN V.CANTIDAD_DIAS ELSE 0 END), 0) AS DIAS_RECHAZADOS
4 FROM EMPLEADO E
5 LEFT JOIN VACACIONES V ON E.ID = V.ID_EMP
6 GROUP BY E.ID, NOMBRE;
```

ID	NOMBRE	DIAS_APROBADOS	DIAS_RECHAZADOS
1	JUAN PELAEZ	14	0
2	ANDRES GARCIA	20	14
3	LAURA PEREZ	16	0
4	PEPE MARTINEZ	0	0
5	MARGARITA CORRALES	0	0

En esta consulta se maneja muy similar a las anteriores en cuestión de la unión entre ambas tablas y sus llaves, de igual forma llamamos a los campos de ID, la concatenación de NOMBRE y APELLIDO pero ahora necesitamos crear unas columnas DIAS_APROBADOS y DIAS_RECHAZADOS. La columna DIAS_APROBADOS muestra la cantidad de días aprobados de vacaciones de los empleados bajo la condición que si el registro contiene ESTADO = A, entonces se agregará a la suma de días aprobados y si no cumple con esta condición, se le asignará un valor de 0. Bajo este mismo orden de ideas trabaja la columna DIAS_RECHAZADOS, donde si el registro contiene ESTADO = R, entonces se agregará a la suma de días rechazados y si no cumple con esta condición, se le asignará un valor de 0.