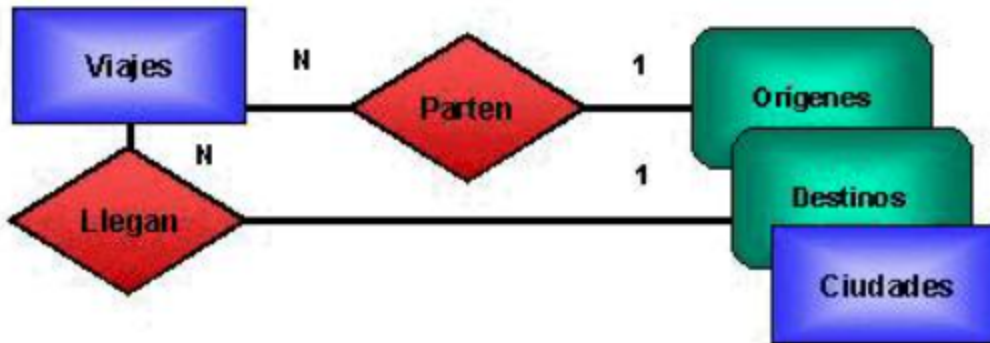


13. Lectura Consultas en SQL Usando Roles y Subconsultas

En SQL, es común enfrentar situaciones donde una misma tabla debe ser **referenciada múltiples veces en una consulta**, ya sea para **comparar subconjuntos** de registros o para que la tabla desempeñe **diferentes roles** dentro de la misma consulta.

Para manejar estas situaciones, se utilizan **alias** y **sinónimos** que permiten **referenciar la misma tabla**



Uso de Alias en Consultas SQL

Alias: Son nombres temporales asignados a tablas o columnas dentro de una consulta para cuando una tabla participa más de una vez en la consulta.

Ejemplo 1: Relaciones entre Ciudades

- Consideremos un modelo donde se registran viajes entre ciudades, con las siguientes tablas:
 - viajes**(idviaje, idorigen, iddestino, fecha)
 - ciudades**(idciudad, nombreciudad)

Para obtener el identificador del viaje, el nombre de la ciudad de origen, el nombre de la ciudad de destino y la fecha del viaje, se puede utilizar la siguiente **consulta con alias** :

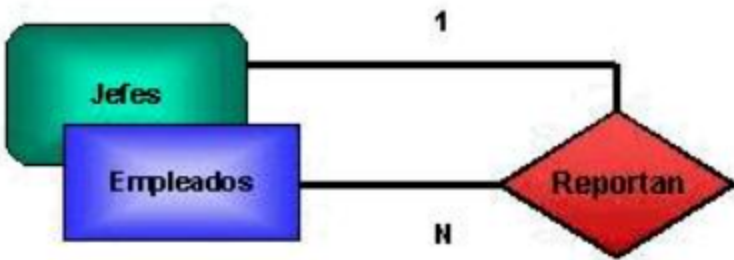
Alias_Ciudad.sql

```
1
2  SELECT idviaje, origen.nombreciudad AS ciudad_origen, destino.nombreciudad AS
   ciudad_destino, fecha FROM viajes JOIN ciudades AS origen ON viajes.idorigen =
   origen.idciudad JOIN ciudades AS destino ON viajes.iddestino = destino.idciudad;`
3
```

- En esta consulta, la tabla **ciudades** se utiliza dos veces con diferentes alias (**origen** y **destino**) para distinguir su rol en cada caso.

Ejemplo 2: Relaciones Jerárquicas entre Empleados

- Consideremos una tabla que registra empleados y sus jefes:
 - **empleados**(idempleado, nombre, idjefe)



Para obtener el **nombre del empleado y el nombre de su jefe**, se puede realizar la siguiente consulta utilizando alias:

Alias_Empleados.sql

```
1
2  SELECT e.nombre AS empleado, j.nombre AS jefe FROM empleados AS e LEFT JOIN empleados
3  AS j ON e.idjefe = j.idempleado;`
```

Aquí, la tabla **empleados** se referencia dos veces: una como **e** para los empleados y otra como **j** para los jefes.

Uso de Sinónimos en Consultas SQL

Sinónimos: Son **nombres alternativos** permanentes asignados a tablas o vistas en la base de datos, permitiendo referenciarlas con diferentes nombres **sin duplicar** su contenido.

Creación de Sinónimos:

Create_Synonym

```
1
2  CREATE SYNONYM origen FOR ciudades; CREATE SYNONYM destino FOR ciudades;`
3
```

Después de crear estos sinónimos, la **consulta del ejemplo 1** se puede **simplificar** de la siguiente manera:

Sinonimos_Consulta_1

```
1
2  SELECT idviaje, origen.nombreciudad AS ciudad_origen, destino.nombreciudad AS
3  ciudad_destino, fecha FROM viajes JOIN origen ON viajes.idorigen = origen.idciudad JOIN
  destino ON viajes.iddestino = destino.idciudad;`
```

Esto **elimina la necesidad de utilizar alias temporales** en cada consulta, **mejorando la legibilidad y mantenimiento** del código SQL.

Uso de Subconsultas en SQL

Subconsultas: Son **consultas anidadas dentro de otra** consulta principal y se utilizan para realizar **operaciones más complejas**, como **filtrar resultados** basados en conjuntos de datos derivados.

Ejemplo: Identificar Productos No Vendidos

Consideremos las siguientes tablas:

- **productos**(idproducto, descripcion, precio)
- **ventasdiarias**(idproducto, fecha, cantidad)

Para encontrar los productos que no se han vendido, se puede utilizar una subconsulta de la siguiente manera:

Productos_no_vendidos

```
1
2  SELECT idproducto, descripcion FROM productos WHERE idproducto NOT IN (SELECT
   idproducto FROM ventasdiarias);`
3
```

En esta consulta, la subconsulta (SELECT idproducto FROM ventasdiarias) obtiene la lista de productos vendidos, y la consulta principal selecciona aquellos productos cuyo idproducto no está en esa lista.

Consideraciones Importantes:

- Las subconsultas deben retornar un conjunto de datos **compatible con la cláusula en la que se utilizan**.

Dudas

¿Cómo se asigna un alias a una tabla en una consulta SQL?

¿Cuándo es útil utilizar una subconsulta?