
Cómo se usan los JOINS en el análisis de datos

1. Relacionar información distribuida en tablas

En análisis de datos, necesitas combinar datos de múltiples tablas para obtener información consolidada. Por ejemplo:

- Relacionar ventas con clientes (clientes en una tabla, ventas en otra).
- Unir datos de transacciones con productos o categorías.
- Combinar datos de empleados con departamentos.

2. Generar reportes

Al combinar datos de varias tablas, puedes:

- Crear reportes financieros (ventas por mes, cliente o región).
- Analizar comportamiento del cliente (qué productos compra más, con qué frecuencia).
- Ver métricas clave como ingresos, costos o márgenes de ganancia.

3. Preparar datos para modelos de machine learning

Los JOINS ayudan a consolidar diferentes fuentes de datos (clientes, productos, transacciones, etc.) en una tabla amplia que puede usarse para entrenar modelos predictivos.

Ejemplo práctico de uso de JOINS en análisis de datos

Escenario: Analizar el comportamiento de compra de los clientes

Tablas disponibles:

1. **Clientes**

ClienteID	Nombre	Edad	País
1	Ana	28	México
2	Luis	35	Chile
3	Marta	22	Perú

2. **Pedidos**

PedidoID	ClienteID	Producto	Precio	Fecha
101	1	Laptop	800	2025-01-01

102	2	Teléfono	500	2025-01-02
103	1	Tablet	300	2025-01-03
104	4	Audífonos	100	2025-01-04

Objetivo del análisis:

Queremos analizar cuánto gasta cada cliente y obtener una lista con los nombres de los clientes, sus productos comprados y los montos totales gastados.

Paso 1: Usar INNER JOIN para combinar clientes y pedidos

Unimos las tablas **Clientes** y **Pedidos** mediante ClienteID para obtener solo los clientes que han realizado pedidos.

Query:

```
SELECT c.nombre,
       p.producto,
       p.precio
FROM Clientes AS c
INNER JOIN Pedidos AS p
ON c.ClienteID = p.ClienteID;
```

Resultado:

Nombre	Producto	Precio
Ana	Laptop	800
Luis	Teléfono	500
Ana	Tablet	300

♦ **Interpretación:** Ana y Luis han realizado compras. Marta no aparece porque no ha hecho ningún pedido.

Paso 2: Usar agregación para calcular el gasto total por cliente

Podemos extender la consulta para sumar el gasto total de cada cliente.

Query:

```
SELECT c.nombre,  
       SUM(p.precio) AS gasto_total  
FROM Clientes AS c  
INNER JOIN Pedidos AS p  
       ON c.ClienteID = p.ClienteID  
GROUP BY c.nombre; -- Hacemos el uso de GROUP BY para no aplicar la función agregada a toda la tabla.
```

Resultado:

Nombre	Gasto_Total
Ana	1100
Luis	500

- ◆ **Interpretación:** Ana gastó \$1100 en total, mientras que Luis gastó \$500.
-

Paso 3: Usar LEFT JOIN para incluir clientes sin pedidos

Si queremos incluir a todos los clientes, incluso los que no han realizado pedidos, usamos un **LEFT JOIN**.

Query:

```
SELECT c.nombre,  
       SUM(p.precio) AS gasto_total  
FROM Clientes AS c  
LEFT JOIN Pedidos AS p  
       ON c.ClienteID = p.ClienteID  
GROUP BY c.nombre;
```

Resultado:

Nombre	Gasto_Total
Ana	1100

Luis	500
Marta	0

♦ **Interpretación:** Ahora vemos que Marta no ha gastado nada porque no tiene pedidos registrados.

Análisis final

- **JOINS ayudan a consolidar información** de diferentes tablas para obtener una vista clara del comportamiento del cliente.
- Puedes usar este enfoque para segmentar clientes, calcular métricas como LTV (Lifetime Value), o identificar patrones de compra.

Si necesitas otro ejemplo con datos más complejos o gráficos, ¡dímelo!