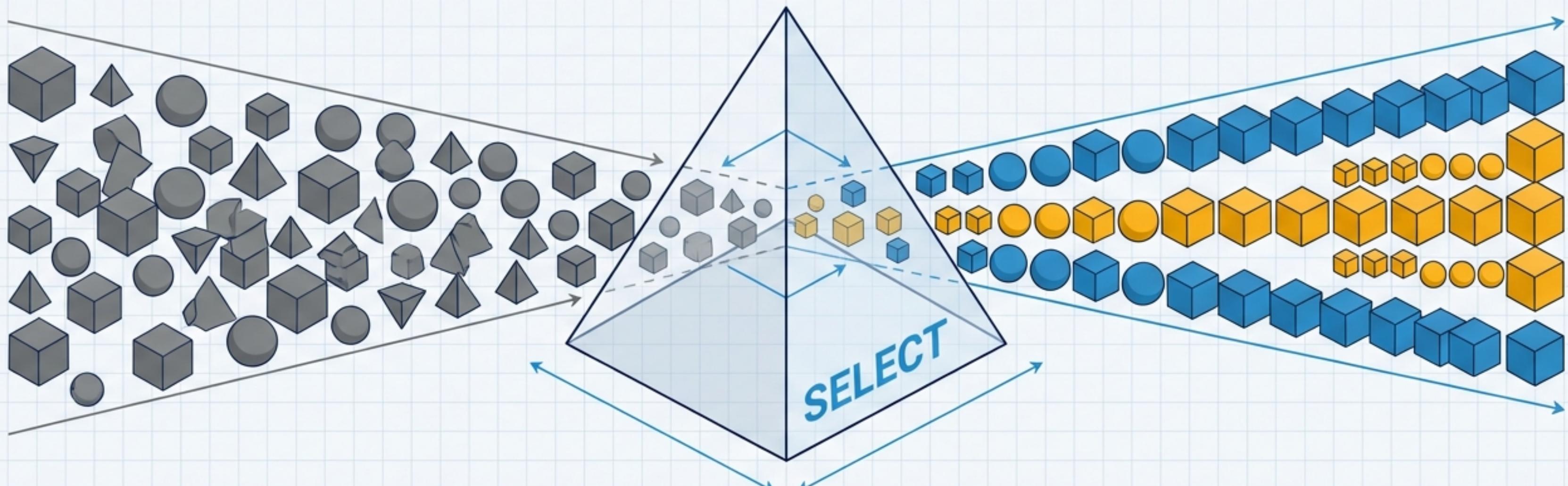


El Arte de Preguntar a los Datos: La Sentencia SELECT

Transformando información bruta en conocimiento estratégico.



El objetivo principal de una base de datos es almacenar información para acceder a ella cuando sea necesario.

La sentencia SELECT es la llave maestra: permite seleccionar datos de interés, filtrarlos y presentarlos en un orden específico.

Es la herramienta fundamental para la administración y explotación de cualquier SGBD.

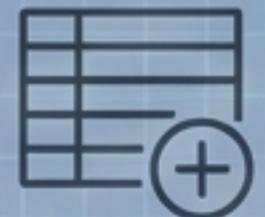
DML: El Lenguaje de Manipulación de Datos

SELECT



El Observador: Consultar información sin alterarla.

INSERT



Añadir nuevos datos.

UPDATE



Modificar datos existentes.

DELETE



Eliminar filas no deseadas.

DML (Data Manipulation Language) es el sublenguaje diseñado para gestionar la información en las tablas.

Nota: Las transacciones DML se ejecutan secuencialmente y ofrecen la posibilidad de revertirse si el resultado no es el esperado.

Anatomía de una Consulta: SELECT + FROM

**SELECT [ALL | DISTINCT] <campos>
FROM <tablas>;**

Qué quiero ver

De dónde lo saco

* (Asterisco)

Tráeme todo. Selecciona todas las columnas.

DISTINCT

Sin repetidos. Descarta valores duplicados.

ALL

Selecciona todas las filas (comportamiento por defecto).

Real World Example

SELECT IDEMPLEADO, NOMBRE, APELLIDO FROM EMPLEADOS;

Claridad ante todo: El uso de Alias

ANTES

FECHA_DE_MATRICULACION	CLI_ID_NUM_2023	IMP_TOT_EUR
2023-01-15	10001	25000.50
2023-01-16	10002	25000.50
2023-01-17	10003	25000.50
2023-01-19	10004	25000.50
2023-01-20	10005	25000.50
2023-01-21	10006	25000.50
2023-01-22	10007	25000.50
2023-01-23	10008	25000.50

AS

DESPUÉS

f_mat	Cliente	Total
2023-01-15	10001	25000.50
2023-01-16	10002	25000.50
2023-01-17	10003	25000.50
2023-01-22	10004	25000.50

A veces, los nombres de columnas o tablas son **extensos** o **poco descriptivos**. SQL permite asignar "**motes**" o **alias** para facilitar la lectura.

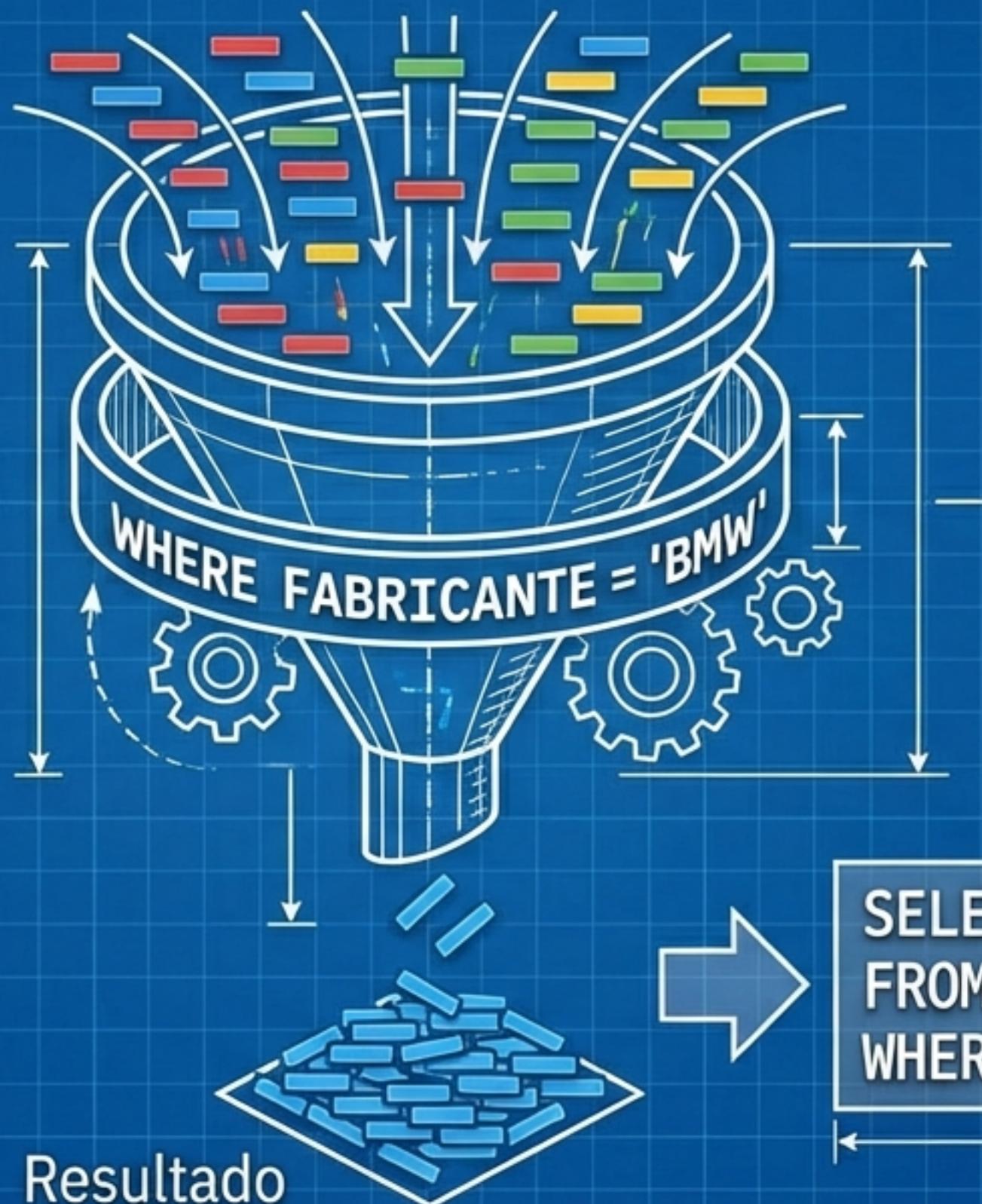
Sintaxis: Se coloca el alias justo después del campo original.

```
SELECT FECHA_DE_MATRICULACION "f_mat" FROM COCHES;
```

El Poder del Filtro: La Cláusula WHERE

SELECT por defecto trae **todo**. Para restringir los resultados a las filas que nos interesan, usamos WHERE.

Esta cláusula establece una **condición** específica que cada fila debe cumplir para ser incluida en el resultado.



```
SELECT MATRICULA, COLOR  
FROM COCHES  
WHERE FABRICANTE = 'BMW';
```

Operadores Lógicos: Herramientas de Precisión

Comparación

- = (Igual)
- > (Mayor que)
- < (Menor que)
- <> (Distinto)

Rango

BETWEEN ... AND

Ideal para fechas o intervalos numéricos.

Lógica

AND (Y)

Ambas condiciones ciertas.

OR (O)

Al menos una cierta.

```
SELECT * FROM ALMACEN WHERE FECHA BETWEEN '15/05/2019' AND '15/05/2020';
```

Esto permite preguntas complejas: 'Muéstrame productos de tal fecha Y tal precio'.

Poniendo Orden: La Cláusula ORDER BY

- Facilita la interpretación de los datos organizándolos según un criterio específico.
- Tipos de datos ordenables:
 - Carácter
 - Número
 - Fecha.

ASC (Ascendente)
A-Z, 0-9.
Es el valor por defecto.

DESC (Descendente)
Z-A, 9-0.



Matices del Orden: Índices y Nulos



Índices Numéricos

```
SELECT product_id, name  
FROM products  
ORDER BY 1;
```

En lugar de escribir el nombre de la columna, se puede usar su posición (1 = primera columna).

1	2	3
1		
2		
3		
4		
5		

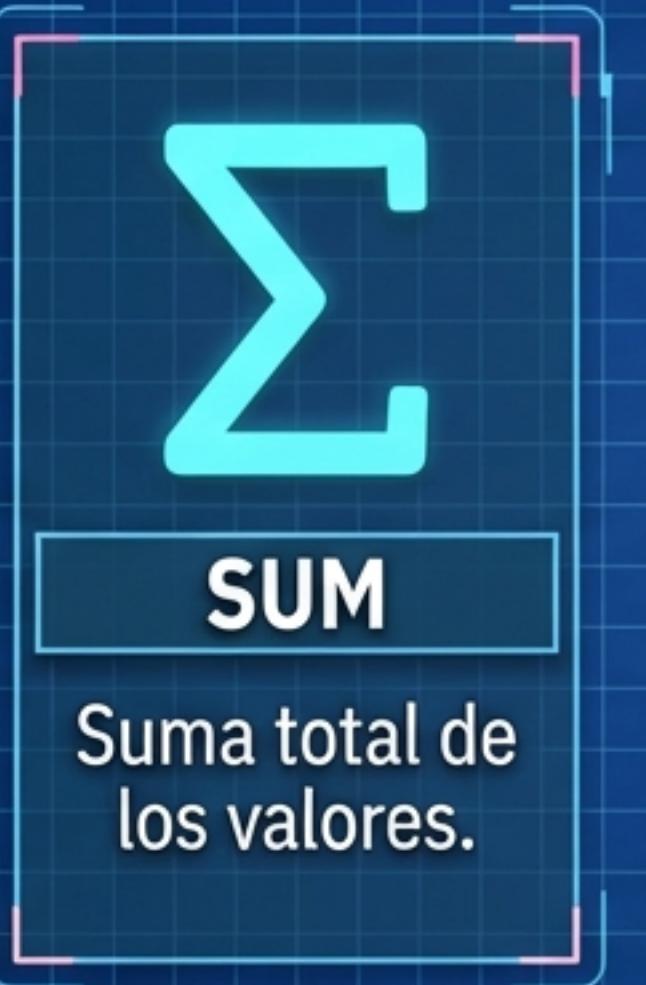
Gestionando el Vacío (NULLs)



`ORDER BY product_id NULLS FIRST;`
Los valores NULL no cumplen condiciones normales. Usa `NULLS FIRST` o `NULLS LAST` para controlar su posición.

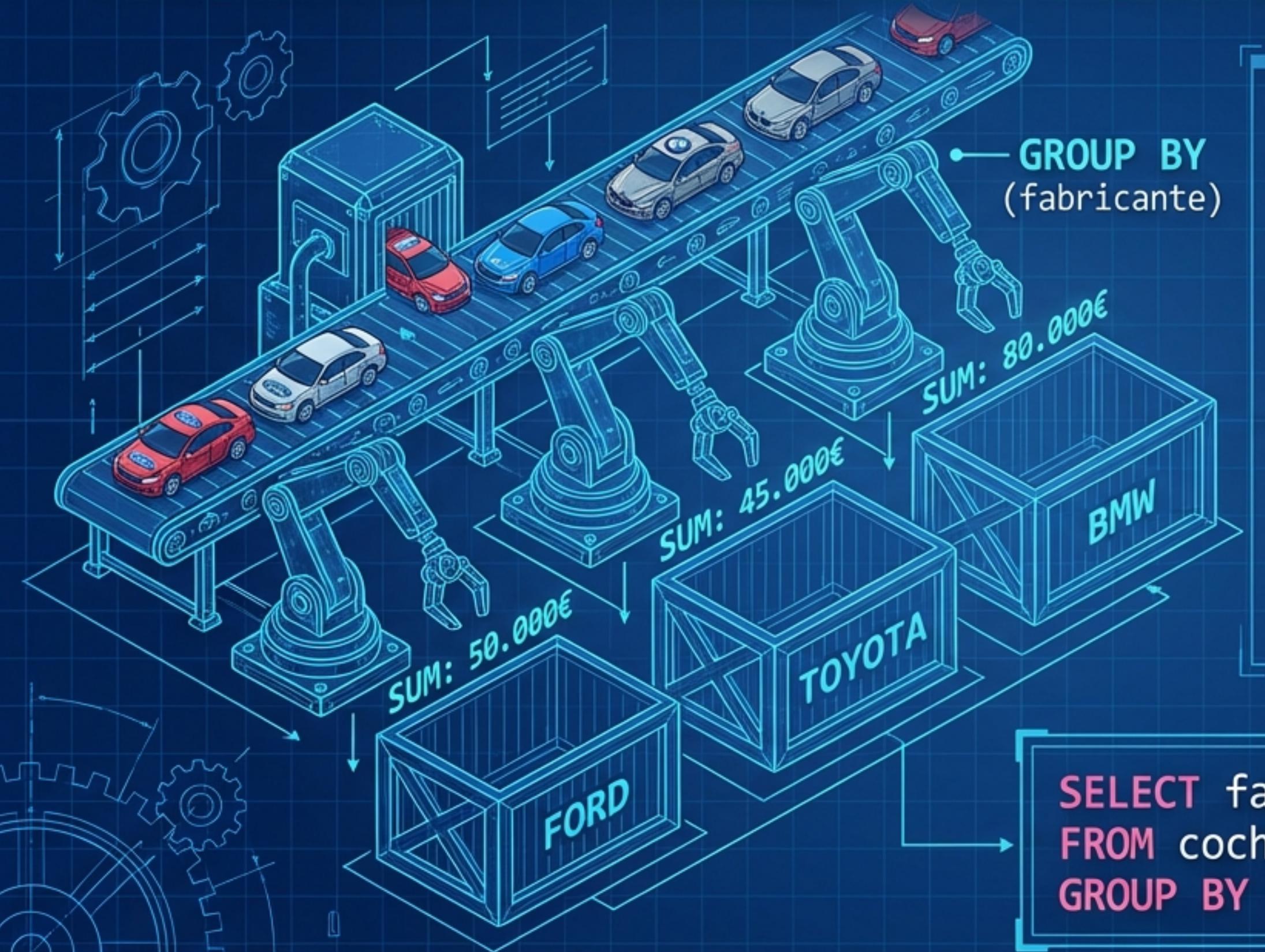
Matemáticas en tus Datos: Funciones de Agregado

Estas funciones permiten calcular resúmenes instantáneos, convirtiendo filas en cifras clave.



```
SELECT AVG(precio)  
FROM productos;
```

Agrupación Inteligente: La Cláusula **GROUP BY**

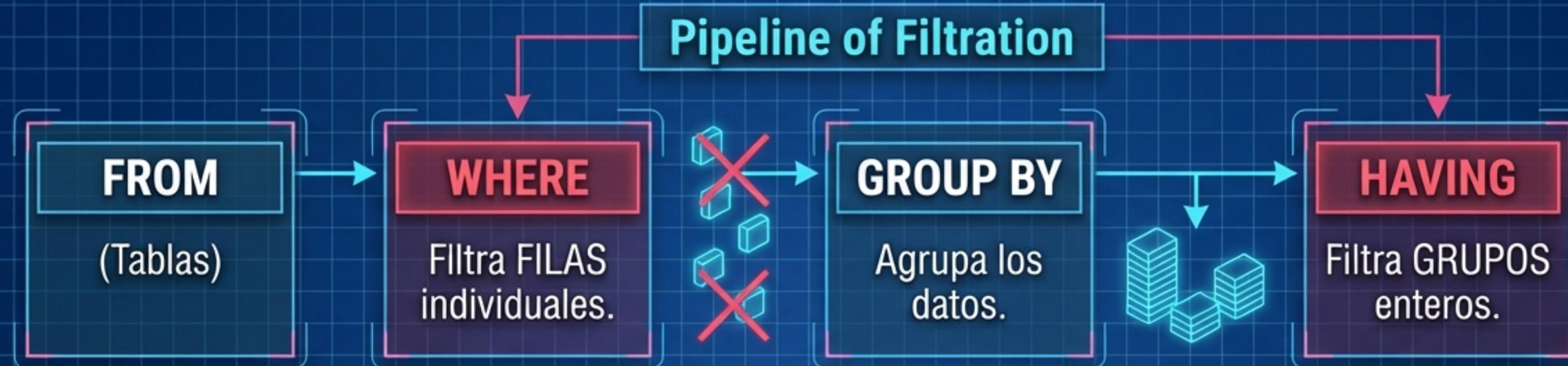


Permite crear **subtotales**. Agrupa filas que tienen los mismos valores en filas de resumen.

Regla de Oro: Se usa frecuentemente junto a funciones de agregado (SUM, AVG).

```
SELECT fabricante, SUM(precio)  
FROM coches  
GROUP BY fabricante;
```

Filtrando Grupos: La Cláusula HAVING



WHERE filtra antes de agrupar.
HAVING filtra después de agrupar (solo sobre agregados).

```
SELECT OrderID, SUM(Quantity)
FROM OrderDetails
GROUP BY OrderID HAVING SUM(Quantity) > 100;
```

Transformación: Cálculos y Concatenación

La base de datos guarda datos crudos; tú presentas información procesada.

Cálculos Aritméticos



```
SELECT precio * 1.21 AS 'Precio_IVA'  
FROM PRODUCTOS;
```

Operadores: + - * /

Concatenación de Texto (||)

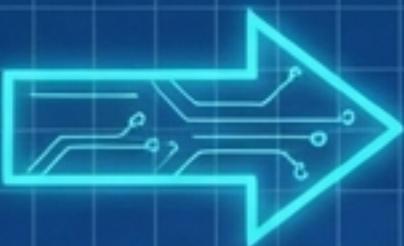


```
SELECT tipo || '-' || modelo AS 'Clave_Pieza'  
FROM PIEZAS;
```

Caso Práctico: Gestionando la Tienda de Informática

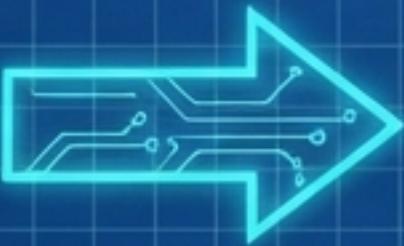
De la pregunta de negocio a la sentencia SQL.

¿Qué pedidos superan los 100€?



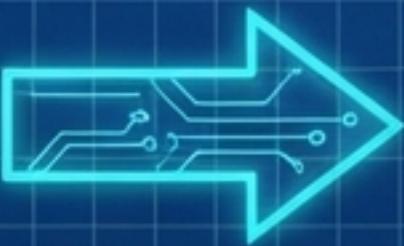
```
SELECT *  
FROM PEDIDOS  
WHERE importe > 100;
```

¿Cuál es el precio medio de productos 'medac'?



```
SELECT AVG(precio)  
FROM PRODUCTOS  
WHERE id_fab = 'medac';
```

¿Cuándo fue el pedido más antiguo?



```
SELECT MIN(fecha_pedido)  
FROM PEDIDOS;
```

La Hoja de Ruta Completa



Esta estructura es **universal**.

Estas herramientas **sirven para todos los perfiles**: desde Recursos Humanos calculando salarios medios, hasta **Gestores de Almacén** controlando el stock.