EMPEZAMOS LA EXPLOTACIÓN, usando LMD (Lenguaje de Manipulación de datos). Con esto podemos hacer inserciones, borrados, modificaciones y consultas.

Para las consultas no hacen falta las restricciones, ya que estas solo influyen a la hora de manipular los datos.

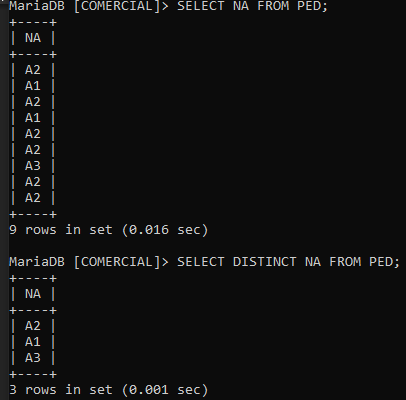
**COMANDOS PARA CONSULTAS BÁSICAS**

**FILTRAR POR COLUMNAS**

**SELECT [\*/COLUMA1/COLUMNA2..] FROM *Nombre\_Tabla/Tablas*** → Nos muestra una tabla con el contenido de una tabla dentro de una BD.

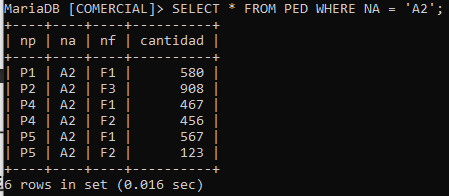
**SELECT [COLUM] AS *Nombre* FROM Tabla** → Nos mostrará los datos de una columna con el nombre indicado.

**SELECT [ALL|DISTINCT] …** → **ALL** es por defecto y nos mostrará todos los valores, aunque estén repetidos, mientras que con **DISTINCT**, solo nos los mostrará una vez.

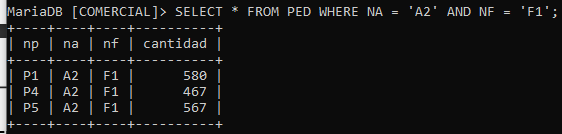


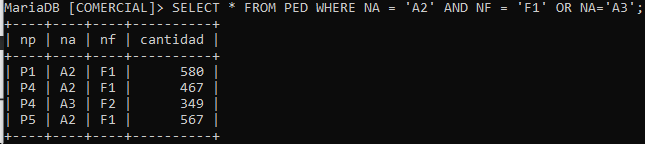
**FILTRAR POR FILAS**

**SELECT [COLUM] FROM *Tabla* WHERE *Condición*\_*lógica****→* Se muestran las filas que cumplan la condición dada.

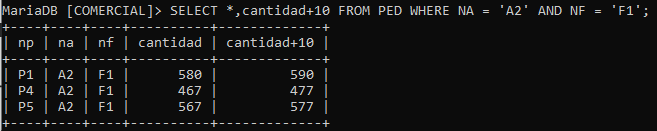
****

Se pueden usar operadores lógicos AND, OR, NOT. Además, si hay un AND y un OR, se realiza primero el AND y a continuación, el resultado se compara con el operador OR



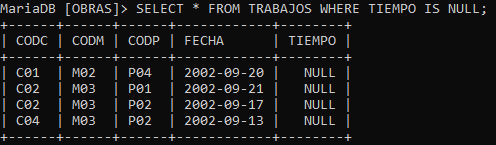


Se pueden usar operadores aritméticos.

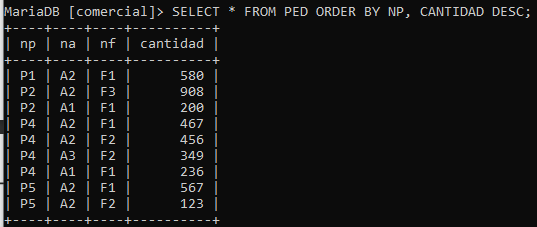


Si queremos sacar aquellos que tengan valores NULL, usamos IS NULL o IS NOT NULL en la condición. NO SE PUEDE USAR *COLUMNA* = ‘NULL’, esto buscará los valores que se llamen ‘NULL’.

NO ES LO MISMO QUE NO HAYA ALGO A QUE EXISTA EL VALOR NULL.



**SELECT [COLUM] FROM *Tabla* ORDER BY *Colum* [DESC|ASC]** → Nos muestra los valores de manera ascendente de manera predeterminada o con DESC de manera descendente. Podemos ordenar varias columnas poniendo detrás una coma.



**EJEMPLO DE EJERCICIO DE EXAMEN (SACAR EL ENUNCIADO)**

**SELECT** (SALARIO \* 10) + COMISION

**FROM** EMP

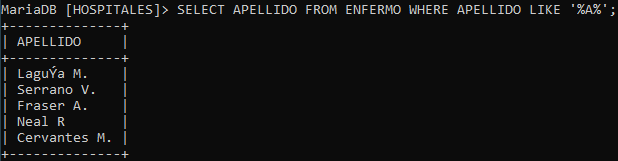
**WHERE** (SALARIO + COMISION) > 1000

Muestra de los empleados el salario multiplicado por 10 más la comisión de aquellos empleados cuyo salario + comisión sea mayor que 1000.

**COMPARACIÓN DE CADENAS**

Para la búsqueda de cadenas se usa % para buscar varios caracteres o \_ para uno solo (En Windows \* y ? respectivamente). Se tiene que usar la sentencia:

**WHERE *NOMBRE* LIKE ‘*[%/\_]CADENA[%/\_]*’;**

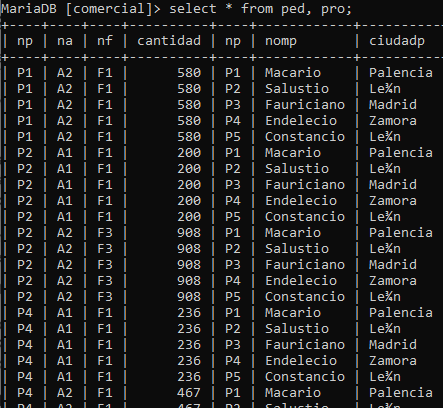


**COLUMNA BETWEEN X AND Y** es lo mismo que **COLUMNA >= X AND COLUMNA <= Y.**

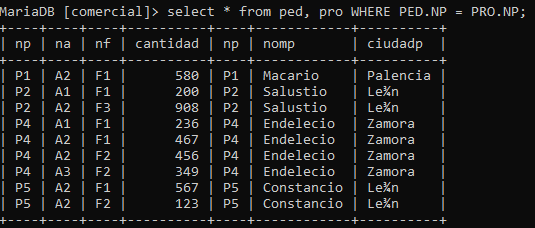
**COMBINACIÓN DE TABLAS**

Tablas que están relacionadas directamente o a través de otras. Ej: Un alumno tiene incluido solo el DNI del profesor. El objetivo es conseguir los demás datos de Profesor combinando ambas tablas.

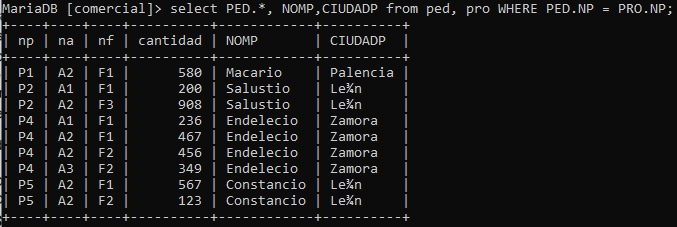
Producto cartesiano: Combinar todos con todos, multiplicar las columnas de las tablas:



Si quiero que salgan solo aquellas filas que tienen sentido, tengo que hacer que las columnas repetidas sean iguales.



No tiene sentido que se repitan las columnas, por lo que:



ENLAZAR LAS TABLAS **SIEMPRE.**

**UNIÓN – INTERSECCIÓN – MINUS**

**UNION**→ Combina los resultados de ambas tablas.

**INTERSECT**→ Intersección de ambas tablas.

**MINUS**→ Lo que no hay ni en una ni en otra. Quitarle a A lo que hay en B

**INNER JOIN / LEFT JOIN / RIGHT JOIN / FULL JOIN**

[**Página de ayuda**](https://programacionymas.com/blog/como-funciona-inner-left-right-full-join)

**INNER JOIN→** Se unen tablas con valores en común. Los valores que no están relacionados no se muestran

**LEFT JOIN→** Que todos los valores de la tabla de la izquierda salgan, aunque no estén relacionados

**RIGHT JOIN→** Lo mismo, pero con la tabla de la derecha

**FULL JOIN→** Unir todos los elementos de las tablas tengan o no una relación

**ESCRIBIR EN UN FICHERO**

Se usa INTO OUTFILE [*Ruta*] al final de las instrucciones.

SELECT \*

FROM PED

INTO OUTFILE ‘/TEST.DAT’;

**LIMIT**

Cláusula que limita la cantidad de filas que se mostrarán en una consulta. Siempre serán las primeras a no ser que le indiquemos a partir de cual fila queremos aplicar el límite empezando por 0. Se recomienda que los valores estén ordenados.

**AGRUPAMIENTO / GROUP BY**

Agrupar valores. Algunas funciones de agrupaciones son:

* **AVG(n)**
* **COUNT(\* / EXPRESION)**
* **MAX(EXPR)**
* **MIN(EXPR)**
* **SUM(EXPR)**
* **STDDEV(EXPR)**

Podemos condicionar el agrupamiento con la cláusula **HAVING**.

Si queremos sustituir un NULL por otro número, usamos IFNULL(*num*).