

Tutoría Base de Datos

07 / 11 / 23

Alternativa Xampp = [WampServer](#)

JOIN.....	2
ROUND.....	2
CONCAT.....	2
CONCAT_WS.....	3
DISTINCT.....	3
COUNT(elem).....	3
SUM (int).....	3
AVG (int).....	3
MAX (int), MIN(int).....	3
GROUP BY.....	4
Consultas y Subconsultas.....	4
PROCEDURE.....	4
x%.....	4

JOIN

Sirve para relacionar tablas. Para hacer un join necesitamos un elemento en común entre las dos tablas.

```
SELECT * FROM clientes
INNER JOIN
ventas
ON clientes.id = ventas.id_clientes;
```

Aparecen todos los datos de ambas tablas donde coinciden el id del cliente con el de ventas.

```
SELECT * FROM clientes
INNER JOIN
ventas
ON clientes.id = ventas.id_clientes
INNER JOIN
ventas
ON ventas.id_productos = producto.id;
```

Localiza los clientes con el id que corresponda con el id del cliente de la venta y los id de los productos en la tabla ventas con el id en la tabla productos.

ROUND

```
ROUND( int, int )
```

Redondea el primer número con los decimales del segundo.

Ej: `ROUND(35.3292 , 2) = 35.33`

CONCAT

```
CONCAT( elem, elem , ... )
```

Concatena los elementos que metamos.

```
SELECT CONCAT(nombre, " ", ciudad) as Nombre_y_Ciudad FROM clientes;
```

Concatena el nombre y la ciudad y los separa con el (" ").

CONCAT_WS

```
CONCAT_WS ( separador, column, ... )
```

```
SELECT CONCAT_WS(" ", nombre, ciudad) as Nombre_y_Ciudad FROM clientes;
```

Concatena los elementos establecidos con la separación que queramos con el String que le digamos. Separa automáticamente nombre y ciudad con (" "). Esto es útil cuando tengamos más de 2 elementos que concatenar.

DISTINCT

Muestra la lista de elementos que sean distintos a los establecidos

```
SELECT DISTINCT ventas.id_productos from ventas;
```

Muestra todos los id de los productos en la tabla ventas

COUNT(elem)

Cuenta el elemento elegido.

SUM (int)

Suma el valor número de los elementos de la columna elegida.

AVG (int)

Calcula el valor número medio de todos los valores que tienen los elementos de la columna.

MAX (int), MIN(int)

Coge el valor máximo / mínimo del elemento elegido.

GROUP BY

Sirve para agrupar elementos con cierta condición.

```
SELECT ventas.id_clientes, clientes.nombre
FROM clientes
INNER JOIN ventas
ON clientes.id = ventas.id_clientes
GROUP BY ventas.id_clientes
HAVING COUNT(*) >= 3;
```

Consultas y Subconsultas

Sirven para obtener una tabla con los elementos pedidos.

```
SELECT productos.nombre, precio FROM productos
WHERE precio > (SELECT MIN(precio) FROM productos);
```

Muestra los nombres y precios de los productos que tengan un precio mayor al precio mínimo de estos.

PROCEDURE

Funciona como un método. Te realiza lo que declares en su interior.

```
delimiter $$
CREATE PROCEDURE CalcEdad (IN an_nac INT, OUT edad INT)
begin
    DECLARE an_actual INT DEFAULT 2023;
    SET edad = an_actual - an_nac;
end $$
delimiter ;

SET @an = 1980;
CALL CalcEdad(@an, @suEdad);
SELECT @suEdad;
```

Calcula la edad actual mediante el año de nacimiento.

Esto se puede ampliar para tener varios IN y varios OUT.

x%

Te coge los elementos que empiezan con esta letra o conjunto de letras.

a% Los que empiezan por "a".

Mig% Los que empiezan por "Mig".