**NUEVOS CONCEPTOS:**

**CRUD:** Create, Read, Update, Delete

Base de datos ←→ Reglas de Negocio ←→ Interfaz de Usuario

SELECT con varias tablas

* Producto cartesiano
* INNER, JOIN. OUTER JOIN

**Producto cartesiano:**

| **A / B** | **1** | **2** | **3** |
| --- | --- | --- | --- |
| **4** | 1,4 | 2,4 | 3,4 |
| **5** | 1,5 | 2,5 | 3,5 |

**JOIN TABLAS:**

* **CROSS:** Se realiza el producto cartesiano. Cuando SELECT pongo varias tablas sin WHERE

select \* from municipio, personas where municipio.id\_municipio=personas.id\_municipios ;

* **INNER:** Solo registros que cumplen una condición. En SELECT pongo una condición…
* **OUTER:** Además de los registros que cumplen la condición, el resto de registros de una de las tablas puede ser LEFT y RIGHT

select \* from municipio LEFT JOIN personas on municipio.id\_municipio=personas.id\_municipio ;

select \* from municipio LEFT JOIN personas using (id\_municipio) ;

select \* from personas RIGHT JOIN municipio on municipio.id\_municipio=personas.id\_municipio ;

* **UNIÓN**:

select \* from A UNION select \* from B ;

select id\_persona,nombre from personas union select \* from municipio ;

* **UNION ALL:**

— Añadimos el valor (3) a B —

select \* from A union all select \* from B ;

**AGREGACIÓN:**

Funciones de agregación:

* **COUNT**: Cuenta el número de filas del dataset
* **COUNT (expr):** Cuenta el número de filas en el que la expr no es NULL
* **COUNT (distinct):** Cuenta el número de valores diferentes para la expr

select count(\*), sum(cantidad) from ticket\_detalle;

select count(\*), sum(cantidad), avg(cantidad) from ticket\_detalle;

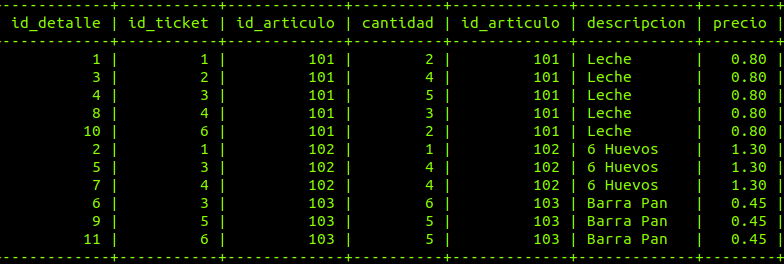
Con **AVG** podremos comprobar la cantidad media de la tabla

select max(cantidad), min(cantidad), avg(cantidad), count(cantidad) from ticket\_detalle;

Con **MAX** y **MIN** comprobaremos los valores medios de la tabla, en este caso de Cantidad

select \* from ticket\_detalle, articulo where ticket\_detalle.id\_articulo=articulo.id\_articulo ;

Esta consulta esta realiza con JOIN esto significa que, nos mostrará una tabla “nueva”, es decir, en este caso nos encontraremos con precio y descripción

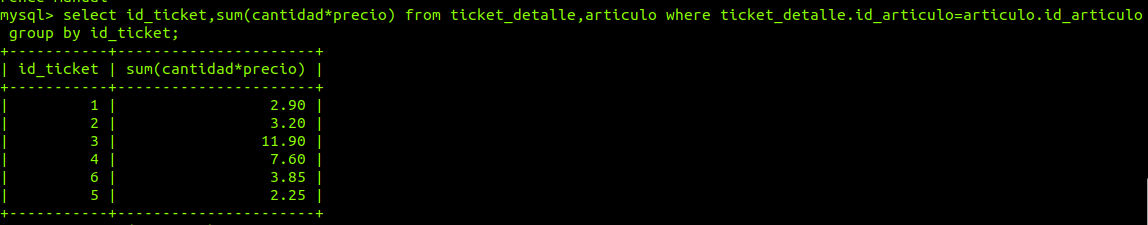


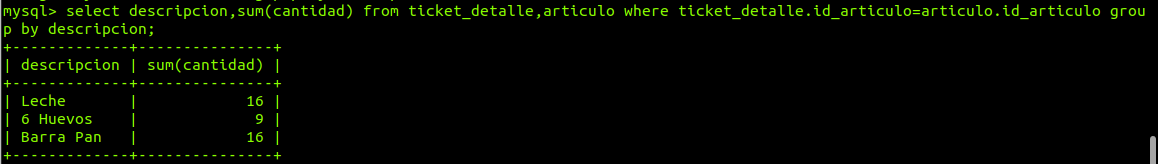
**Cada vez que tenga que relacionar dos tablas relacionadas entre si usamos el where, la corrección de columnas tienen que ser igual**

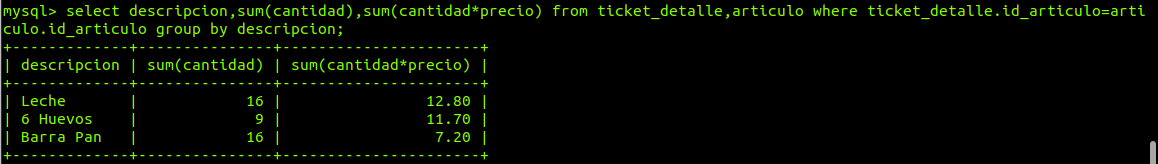
select \*,cantidad\*precio from ticket\_detalle,articulo where ticket\_detalle.id\_articulo=articulo.id\_articulo;

Si queremos comprobar la cantidad del precio total de los artículos usaremos el comando:

select sum(cantidad\*precio) from ticket\_detalle,articulo where ticket\_detalle.id\_articulo=articulo.id\_articulo







Consulta para saber el nombre del cliente y el precio total que se han gastado

select cliente.nombre,

ticket\_detalle.cantidad,

articulo.precio

from cliente,

ticket,

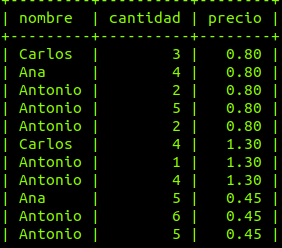
ticket\_detalle,

articulo

where cliente.id\_cliente = ticket.id\_cliente and

ticket.id\_ticket = ticket\_detalle.id\_ticket and

ticket\_detalle.id\_articulo = articulo.id\_articulo ;



SQL es un LENGUAJE de consultas.

Como todos los lenguajes, hay varias formas de decir lo mismo, en este caso, hay varias formas de realizar la misma consulta

**EJEMPLO:**

Select ….. Select …..

From a,b → From ( a join b using (xxxx) ) join c using (yyyy)

Where a.xxxx = b.xxxx AND

b.yyyy = c.yyyy