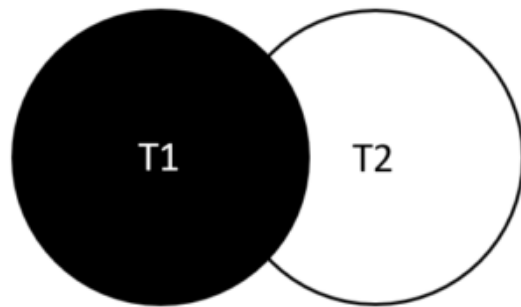


```
mysql -u root -p  
password: root
```

## OUTER JOINS

### LEFT JOIN



Por **cada fila** de la tabla  $t_1$  se genera una nueva tupla en la tabla resultado adjuntándole **cada una de las filas** de la tabla  $t_2$  en las que se cumpla el match.

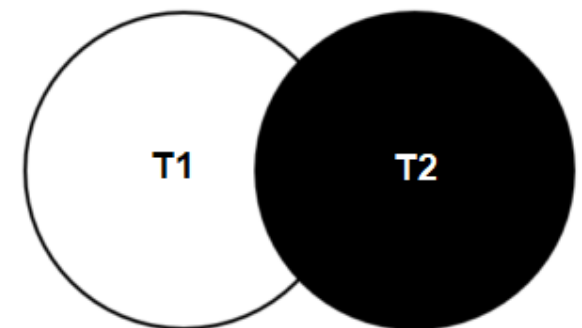
La aridad de la tabla resultado dependerá de la aridad de las tablas involucradas en el join y de la condición (**ON** o **USING**)

¿Cuál es **número** de **TUPLAS** esperado luego de un **LEFT JOIN**?

### RIGHT JOIN

Por **cada fila** de la tabla  $t_2$  se genera una nueva tupla en la tabla resultado adjuntándole **cada una de las filas** de la tabla  $t_1$  en las que se cumpla el match.

La aridad de la tabla resultado dependerá, al igual que para un **LEFT JOIN**, de la aridad de las tablas involucradas en el join y de la condición (**ON** o **USING**)



## SQL Query Execution Order

En SQL, el "orden de ejecución" es la jerarquía bajo la cual el lenguaje evalúa las cláusulas dentro de una consulta.

- Suele ser diferente al orden que está escrita la consulta.
- Es útil cuando se busca depurar y optimizar consultas.

**FROM:** señala la tabla en dónde se almacenan los datos requeridos.

**JOIN + ON:** obtiene los registros coincidentes basado en columna(s) compartida(s) cuando los datos requeridos viven en varias tablas.

**WHERE:** filtra los datos de origen solicitados a nivel de fila.  
No permite alias de columna o funciones de agregación a aplicar.

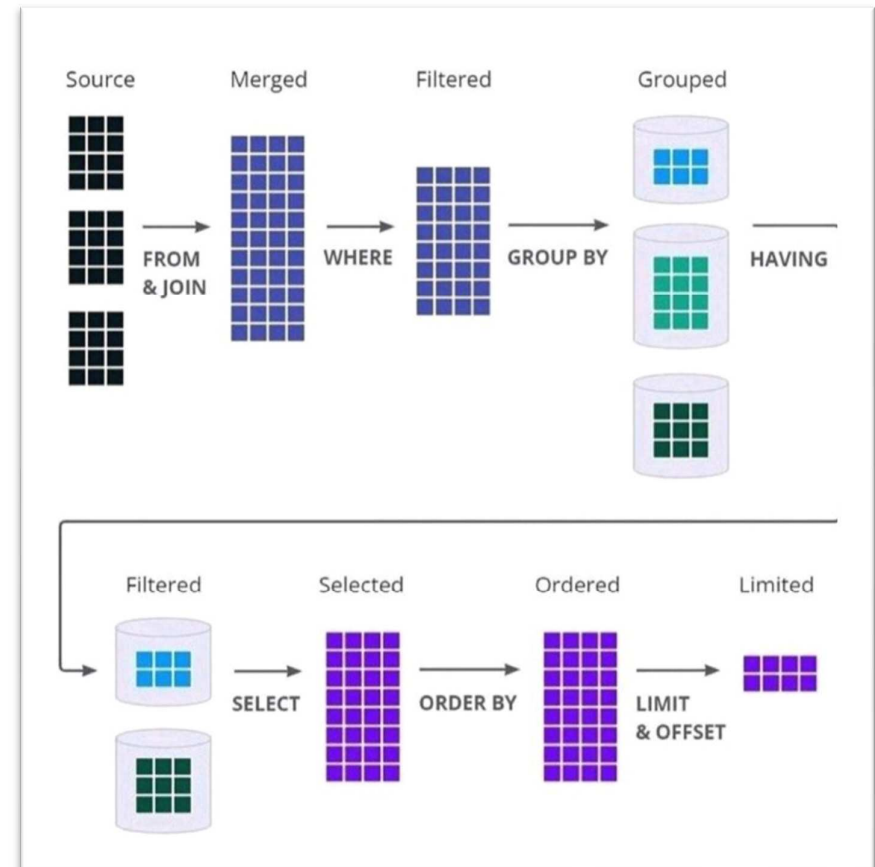
**GROUP BY:** agrupa filas con el mismo valor juntas. A menudo se usa con funciones agregadas. Permiten alias de columna.

**HAVING:** aplica condiciones y/o filtros después de que tiene lugar la agregación. No permite alias de columna.

**SELECT:** proyecta columnas específicas.

**ORDER BY:** ordena las filas resultantes después de filtrado y agregación. Admite alias de columna.

**LIMIT:** reduce el número de filas devueltas por la consulta.



Fuente: <https://data-science-infinity.teachable.com/>

## Ejercicio 1 – Base de datos ‘replica’

0. Mandar llamar al archivo ‘base\_replicas.sql’.

Contiene datos de clientes, productos, pedidos de ventas, artículos de línea de pedidos de ventas, de una tienda minorista de venta de modelos a escala de autos clásicos. Verificar que el número de registros en `orderdetails` sea:

```
+-----+  
| num_registros |  
+-----+  
|           2996 |  
+-----+  
1 row in set (0.01 sec)
```

1. Cada cliente puede haber realizado 0,1 o N órdenes, enlistar todos los nombres de los clientes y el detalle de las ordenes que realizaron.
2. Encontrar el número de órdenes realizadas por cada cliente y ordena del mayor número al menor.
3. ¿Qué sucede si se intenta resolver 2 con un **INNER JOIN**?
4. Usando un **RIGHTJOIN** recupera el nombre y status de los clientes que hayan realizado más de 4 órdenes.
5. ¿Quiénes son los clientes que no han realizado ningún pedido? Resguárdalos en una vista `clientes_obj`
6. Hallar en qué ciudades debe enfocar sus objetivos el área de ventas. *[Usar 5]*
7. Elimina la vista `clientes_obj`