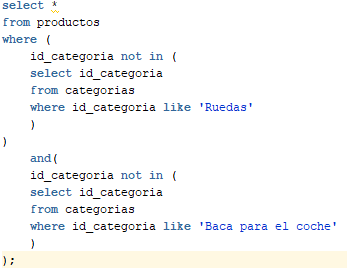
**Uso de WHERE y AND{**

Asi lo usamos de la manera mas facil



**}**

**Mayor que todos {**

**Selecciona los productos cuyo precio sea mayor o igual que el de todos los demás**

**Usamos ALL para comparar el precio de cada producto con todos**

select \*

from productos

where precio\_venta >= all (

select precio\_venta

from productos

);

**}**

**JOIN ON {**

Para mostrar resultados de distintas tablas en una consulta

Select a.x, b.x, c.x

from table1 a join table2 b on (a.xx = b.xx)

join table3 c on (b.xxx = c.xxx);

**}**

**Operadores de Conjunto {**

-El operador **UNION** combina los resultados de dos consultas en un conjunto de resultados único, eliminando las duplicaciones.

-El operador **INTERSECT** devuelve los registros comunes a ambas consultas.

-El operador **MINUS** devuelve los registros que aparecen en la primera consulta pero no en la segunda consulta.

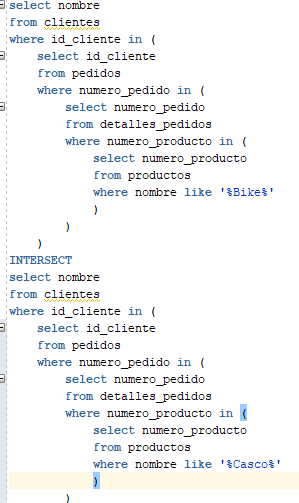
**INTERSECT**

**42.Intersección: Listar los clientes que han comprado tanto productos que contengan la cadena 'Bike' en el nombre como productos que contengan la cadena 'Casco' en el nombre.**

La razón por la que la consulta con INTERSECT devuelve menos resultados que las consultas individuales es porque **INTERSECT devuelve sólo los valores comunes en ambas consultas.**

En este caso, la primera consulta devuelve los clientes que han comprado productos que contienen la palabra "Bike", mientras que la segunda consulta devuelve los clientes que han comprado productos que contienen la palabra "Casco". Cuando se utiliza **INTERSECT**, la consulta devuelve sólo los clientes que han comprado tanto productos que contienen "Bike" como productos que contienen "Casco". Es decir, sólo se mostrarán los clientes que hayan comprado ambos tipos de productos.

Por lo tanto, si la consulta con "Bike" devuelve 27 nombres y la consulta con "Casco" devuelve 23 nombres, no es necesariamente el caso que la consulta INTERSECT deba devolver 27 nombres. En cambio, devolverá sólo aquellos nombres que aparecen en ambas consultas. Si hay clientes que han comprado productos que contienen "Bike" pero no "Casco" (o viceversa), no se mostrarán en la consulta INTERSECT.



**MINUS**

**43.Listar los clientes que han comprado algún producto que contenga la cadena 'Bike' en el nombre pero ningún producto que contenga la cadena 'Casco' en el nombre.**

La consulta selecciona los nombres de clientes que han realizado pedidos que contienen productos con la palabra "Bike" en su nombre, pero no han realizado pedidos que contengan productos con la palabra "Casco" en su nombre.

La cláusula "**MINUS**" se utiliza para **excluir los resultados que aparecen en la segunda consulta de los resultados de la primera consulta**. En este caso, la consulta devuelve los nombres de clientes que están en la primera consulta pero no en la segunda consulta. Es decir, devuelve los clientes que han comprado productos con la palabra "Bike" en su nombre pero no han comprado productos con la palabra "Casco" en su nombre

.

**UNION**

**44. Listar los clientes que han comprado algún producto que contenga la cadena 'Bike'o la cadena 'Casco' en el nombre.**

Esta consulta devuelve una lista de nombres de clientes que han realizado pedidos que incluyen productos que contienen la palabra "Bike" y/o la palabra "Casco".

La diferencia entre INTERSECT y UNION en SQL es que INTERSECT devuelve solo los resultados que aparecen en ambas consultas, mientras que **UNION devuelve todos los resultados únicos de ambas consultas**.

**}**

**Group By{**

Lo usamos por ejemplo cuando nos piden el precio mas alto de **cada categoria,** ya que aqui queremos agrupar por categoria para que en la consulta si hay 3 categorias muestre solo esas 3.

categoria 1 id prod 2 precio 4$

categoria 2 id prod 1 precio 5$

categoria 3 id prod 3 precio 6$

categoria 1 id prod 4 precio 2$

categoria 1 id prod 6 precio 4$

el resultado del group by:

categoria 1 precio 10$

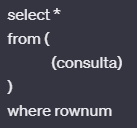
categoria 2 precio 5$

categoria 3 precio 6$

**Rownum**

Lo utilizamos para seleccionar un numero de tablas especificas en el resultado de la consulta.

El rownum se ejecuta antes que las cláusulas “*order by”* y “*where”* por lo que usaremos la siguiente sentencia



**}**