SQL PARTE 2

JOIN IMPLICITO -> tengo 2 tablas y les hago la intersección bajo una condición

SELECT \* FROM cliente, factura

Where clie\_codigo = fact\_cliente

Este join hace exactamente lo mismo

JOIN -> clausula que permite unir 2 o mas tablas para acotar el producto cartesiano que aparece en el FROM cuando agarro 2 tablas

SELECT \* FROM cliente JOIN factura ON clie\_codigo = fact\_cliente

Prefiero siempre usar este caso porque WHERE se usa mas para un filtrado especifico, una condición adicional

INNER JOIN -> es una interseccion, funciona igual que el JOIN pero otros motores lo escriben de esta forma

Para el group by, si quiero contar la cantidad de ocurrencias, debo agrupar por los campos no calculables

LEFT/RIGHT JOIN -> a una de las tablas me la trae siempre con todas las filas. Es decir, las que cumplen con el join y las que no.

LEFT JOIN -> en este caso, me agrega en el resultado todo lo que este a la izquierda del igual que no cumpla con la restricción además de lo que traería normalmente. En este caso, me trae todos los clientes con factura más los que no tienen factura

RIGHT JOIN -> le da prioridad a todo lo que esta en la derecha

CASE -> permite establecer condición multiple dentro de un select (es como un switch de C)

Hay 2 formas

Con valor de columna

**CASE** columna **WHEN** valor1 **THEN** exp1 **WHEN** valor2 **THEN** exp2 **WHEN** valor3 **THEN** exp3 **ELSE** opcion **END**

Con condición

**CASE WHEN** condicion1 **THEN** exp1 **WHEN** condicion2 **THEN** exp2 **WHEN** condicion 3 **THEN** exp3 **ELSE** opcion **END**

TOP -> delimita la cantidad de filas desde el principio. En este caso, me devuelve las n primeras filas de la tabla *table* ordenados de mayor a menor por columna 1

SELECT TOP n col1, col2 FROM table

ORDER BY col1

UNION -> une varios resultados de SELECT distintos. Solo puede llevar un ORDER BY al final. Tengo que elegir la misma cantidad de columnas en cada select y además, cada una debe ser del mismo tipo y cantidad, es decir:

**SELECT** col1, col2 **FROM** tabla1

**UNION**

**SELECT** col3, col4 **FROM** tabla2

Col1 y col3 deben ser del mismo tipo y tamaño

El union es disyuntivo -> NO REPITE FILAS

Si no quiero disyunción -> pongo UNION ALL

SUBCONSULTAS -> utilización recursiva de un select dentro de otro select. El que esta adentro se resuelve primero. Puede colocarse dentro de cualquier SELECT, FROM, WHERE, ORDER BY, GROUP BY y HAVING.

En la linea del select -> el select debe devolverme una fila y una columna

**SELECT** col1, (**SELECT** col5 **FROM** tabla2 **WHERE** col3 = tablas1.col1)

**FROM** tabla1

En el where y en el having-> el select me debe devolver cualquier expresión que pueda ser utilizada en una condición

**SELECT** col1, col2 **FROM** tabla1

**WHERE** col1 = (**SELECT** col3 **FROM** tabla2 **WHERE** col3 = tablas1.col1)

--aca el select devuelve un valor o columna/fila

**SELECT** col1, col2 **FROM** tabla1

**WHERE** col1 **IN** (**SELECT** col3 **FROM** tabla2) -- aca devuelve una columna

SELECT col1, col2 FROM tabla1

WHERE EXISTS (SELECT col3 FROM tabla2 WHERE col3 = col1)

--aca devuelve un conjunto de filas

En el group by -> me tiene que devolver un valor, es decir una fila y una columna

En el order by -> debe devolver una fila y una columna

En el from -> me tiene que devolver una tabla, es decir, una fila y una columna