* Describe el funcionamiento general de la sentencia **JOIN**.

R: El JOIN permite combinar filas de dos tablas que compartan una llave primaria con una llave secundaria entre ellas, lo que significa que existe una relación entre ambas tablas.

* ¿Cuáles son los tipos de **JOIN** y cuál es el funcionamiento de los mismos?

R: INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN y FULL JOIN.

* INNER JOIN – Une las filas de dos tablas, mostrando solo las filas que tengan una coincidencia en estas.
* LEFT JOIN – Une las filas de dos tablas, mostrando todas las filas de la tabla izquierda (Es la que se especifica primero en la cláusula FROM) tengan o no coincidencia con la tabla Derecha.
* RIGHT JOIN – Une las filas de dos tablas, mostrando todas las filas de la tabla derecha (Es la que se especifica después de la cláusula JOIN) tengan o no coincidencia con la tabla izquierda.
* CROSS JOIN – Une las filas de dos tablas, mostrando las filas de ambas considerando todas las posibles combinaciones entre ambas.
* FULL JOIN – Une las filas de dos tablas, mostrando las filas de ambas tablas sin considerar si existen o no coincidencias entre ambas.
* **¿**Cuál el funcionamiento general de los **TRIGGER** y qué propósito tienen?

R: un trigger es un tipo especial de procedimiento almacenado. Estos se ejecutan de forma automáticamente en respuesta a los eventos ocurridos en una tabla o vista.

Estos eventos suelen ser INSERT, UPDATE y DELETE, también pueden ser eventos como CREATE, ALTER y DROP.

* ¿Qué es y para qué sirve un **STORED PROCEDURE**?

R: Un SP (Procedimiento almacenado) es un bloque de código o instrucciones SQl que se almacena en una base de datos y que permite ser invocado como una unidad para realizar tareas específicas.

* Traer todos los productos que tengan una venta.

R: SELECT DISTINCT nombre FROM productos prd INNER JOIN ventas vts ON vts.idProducto = prd.idProducto

* Traer todos los productos que tengan ventas y la cantidad total de productos vendidos.

R: SELECT nombre, SUM(cantidad) AS totalVendido FROM productos prd INNER JOIN ventas vts ON vts.idProducto = prd.idProducto GROUP BY nombre

* Traer todos los productos (independientemente de si tienen ventas o no) y la suma total ($) vendida por producto.

R: SELECT prd.nombre, SUM(prd.precio \* vts.cantidad) AS valorVenta FROM productos prd LEFT JOIN ventas vts ON vts.idProducto = prd.idProducto GROUP BY nombre