1. CONOCIMIENTOS SQL
   1. Describe el funcionamiento general de la sentencia JOIN.

La sentencia JOIN nos permite el unir dos o mas tablas por una clave en común entre ellas, dando como resultado una tabla que combina todos los campos de dichas tablas (o los campos especificados previamente en el select).

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

Tenemos el INNER, para que esta sentencia nos arroje resultado la clave de búsqueda o de unión deberá existir en las tablas a unir mediante el JOIN.

Esta el LEFT, esta sentencia nos mostrara los todos resultados (tomando en cuenta filtros) de la tabla a la izquierda, y de la tabla a la derecha mostrara los valores si existe relación alguna con la clave de unión, si no existiese relación en la tabla de la derecha mostraría NULL por los campos seleccionados en dicha tabla.

Tenemos el RIGHT, siendo lo contrario de LEFT, mostrara inicialmente los valores de la tabla a la derecha, y en la tabla izquierda arrojara valores si existe la relación mediante la clave buscada, sino es asi muestra NULL en la tabla a la izquierda.

Y finalmente tenemos el OUTER, el cual al aplicar el JOIN entre dos tablas aplica internamente el LEFT y RIGHT, siendo asi que mostrara los valores de ambas tablas tengan o no relación la una con la otra, nuevamente al existir relación muestra los valores, de lo contrario mostrara null.

* 1. ¿Cuál el funcionamiento general de los TRIGGER y qué propósito tienen?

Los triggers son disparadores creados mediante código, el cual se ejecutara en el momento para el cual están programados, siendo que pueden ejecutarse antes de realizar alguna acción como INSERT, UPDATE, DELETE o después de ellas, llevando a cabo la acción que tengan especificadas internamente como ejemplo pudiera ser validar existencias entre otras, normalmente usados para llevar a cabo el llenado de una bitácora de acciones realizadas en la base de datos.

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

Un procedimiento almacenado es un bloque de código el cual nos permitirá ejecutar varias líneas para llevar a cabo una tarea en especifico, pudiendo este recibir o no parámetros de entrada, para usarlos dentro de las sentencias del procedimiento, de igual manera nos podrá retornara el resultado acerca de la ejecución del código, importante aquí podemos manejar las transacciones para asegurar el resultado.

* 1. Traer todos los productos que tengan una venta.

SELECT nombre FROM productos p INNER JOIN ventas v ON p.idProducto = v.idProducto WHERE cantidad != 0

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

Select nombre, SUM(cantidad) as CantVendida FROM productos p INNER JOIN ventas v ON p.idProducto = v.idProducto Where cantidad != 0 GROUP BY nombre

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

SELECT nombre, CASE WHEN CANTIDAD IS NULL THEN 0 ELSE SUM(cantidad\*precio) END As CantVendida FROM producto p LEFT JOIN ventas v ON p.idProducto = v.idProducto Where cantidad != 0 GROUP BY nombre

1. EJERCICIO PRÁCTICO: BD
   1. Crea un diagrama relacional de BD para el escenario descrito anteriormente.

