Guillermo Leonel Vásquez López

1.1

Esta función permite combinar datos de múltiples tablas con el funcionamiento de condiciones relacionadas, dependiendo el tipo de JOIN se realizarán las diferentes funciones como INNER, LEFT, RIGHT Y FULL.

1.2

INNER JOIN:

Entrega únicamente las filas que tienen coincidencias en ambas tablas basadas con una condición especifica.

```
SELECT * FROM tabla1 INNER JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

LEFT JOIN:

Muestra las filas de la tabla principal (Izquierda) y las filas que coinciden de la tabla secundaria (Derecha), pero si no hay coincidencias entregara un valor NULL.

```
SELECT * FROM tabla1 LEFT JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

RIGHT JOIN:

Como Left Join pero lo hace al contrario ya que se tiene que ejecutar con respecto a la tabla secundaria, ósea que entrega las filas de la izquierda que coincidan con la tabla de la derecha y las que no tienen coincidencia las rellena con un valor NULL.

```
SELECT * FROM tabla1 RIGHT JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

FULL JOIN:

Entrega las filas de ambas tablas, combinando las filas cuando hay coincidencias y agregando un valor NULL donde no tenga las coincidencias.

```
SELECT * FROM tabla1 FULL JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

CROSS JOIN:

Se entrega una combinación con respecto de las dos filas como es tal una combinación.

```
SELECT * FROM tabla1 CROSS JOIN tabla2;
```

SELF JOIN:

Esta función es capaz de unir esa tabla consigo misma, funcionalmente es una opción funcional para comparar una filas dentro de la tabla.

```
SELECT a.columna1, b.columna2 FROM tabla AS a, tabla AS b WHERE a.columnaX = b.columnaY;
```

1.3

Estas son una herramienta capaz de automatizar y controlar el comportamiento de la base de datos en respuesta a eventos específicos. Estos eventos nos dan funcionalidad y flexibilidad para el uso de la base de datos. Podemos delimitar restricciones adicionales, automatizar mantenimientos, validación de datos y autorizar auditorias con respectos cambios de datos.

1.4

Se trata de una colección de instrucciones de SQL predefinidas y almacenadas en la base de datos. Estas herramientas poderosas permiten encapsular lógicas complejas con la funcionalidad de mejorar el rendimiento, otorgar seguridad y una sencilla abstracción en cuanto a las ejecuciones de la base de datos.

1.5

Agregue una capsula GROUP BY y la condición HAVING para que se filtren los resultados.

```
SELECT productos.*
FROM productos
JOIN (
    SELECT idProducto, COUNT(*) AS total_ventas
    FROM ventas
    GROUP BY idProducto
    HAVING total_ventas = 1
) AS ventas_por_producto ON productos.idProductos = ventas_por_producto.idProducto;
```

1.6

Se utiliza una consulta JOIN entre las tablas productos y ventas para agrupar los resultados por productos para calcular la cantidad de productos vendidos.

```
SELECT productos.idProductos, productos.nombre, productos.precio,
       COUNT(ventas.idVenta) AS total_ventas
FROM productos
JOIN ventas ON productos.idProductos = ventas.idProducto
GROUP BY productos.idProductos, productos.nombre, productos.precio;
```

1.7

Se hace uso de la combinación de LEFT JOIN y SUM para que todos los productos estén incluidos incluso si no tienen ventas.

```
SELECT productos.idProductos, productos.nombre, productos.precio,
       COALESCE(SUM(ventas.cantidad * productos.precio), 0) AS total_vendido
FROM productos
LEFT JOIN ventas ON productos.idProductos = ventas.idProducto
GROUP BY productos.idProductos, productos.nombre, productos.precio;
```