

1. CONOCIMIENTOS SQL

1.1 Describe el funcionamiento general de la sentencia JOIN

La sentencia JOIN permite poder realizar consultas de datos uniendo los resultados de dos o más tablas teniendo algún campo en común. Es decir, permite la unión de datos de dos o más tablas en la consulta sólo si entre las tablas tienen algún campo en relación.

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

- ◆ INNER JOIN: Muestra el resultado de la consulta donde el campo en común coincide o existe un registro en ambas tablas. Es decir, sólo resultados existentes en ambas tablas relacionadas.
- ◆ LEFT JOIN: Muestra el resultado de la consulta tanto de la primer tabla como los que coinciden con la segunda tabla según el campo en común.
- ◆ RIGHT JOIN: Muestra el resultado de la consulta tanto de la segunda tabla como los que coinciden con la primer tabla según el campo en común.
- ◆ FULL JOIN: Muestra el resultado de la consulta de ambas tablas completamente, es decir, ya sea que haya coincidencia o no en el campo común.

1.3 ¿Cuál es el funcionamiento general de los TRIGGER y qué propósito tienen?

La sentencia TRIGGER permite crear un tipo de procedimiento o disparador que se ejecuta según una acción realizada, por ejemplo al insertar, actualizar, eliminar, entre otras. Es decir, el procedimiento se ejecuta una vez que alguna acción específica es ejecutada en alguna tabla. Pueden ser utilizados para diferentes propósitos como copias o replicas de datos, actualización de datos que tenga en común o en relación con otras tablas, evitar la perdida de datos al realizar una actualización o eliminación, entre otros.

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

La sentencia STORED PROCEDURE es un procedimiento almacenado, es decir, guarda sentencias de acciones que se deseen ejecutar, permitiendo ejecutar el procedimiento sin necesidad de escribir la acción completa nuevamente. Por ejemplo, puede guardar una consulta que utiliza varias sentencias JOIN que involucran mas de dos tablas o varios datos que sean necesarios, si es una consulta que es necesaria más de una vez puede ser guardada como un STORED PROCEDURE para que pueda ser ejecutada cuando se necesite. Puede considerarse como un tipo de método que puede ser llamado varias veces.

1.5 Traer todos los productos que tengan una venta

```
43      -- 1.5) Traer todos los productos que tengan una venta
44 •   SELECT PRO.NOMBRE, COUNT(VEN.CANTIDAD) AS TOTAL FROM PRODUCTOS AS PRO
45     JOIN VENTAS AS VEN ON PRO.ID_PRODUCTO = VEN.ID_PRODUCTO
46     GROUP BY PRO.NOMBRE HAVING COUNT(VEN.ID_VENTA) = 1;
47
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	NOMBRE	TOTAL		
▶	TECLADO	1		

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

```
48      -- 1.6) Traer todos los productos que tengan ventas y la cantidad total de productos vendidos
49 •   SELECT PRO.NOMBRE, SUM(VEN.CANTIDAD) AS TOTAL FROM PRODUCTOS AS PRO
50     JOIN VENTAS AS VEN ON PRO.ID_PRODUCTO = VEN.ID_PRODUCTO
51     GROUP BY PRO.NOMBRE;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	NOMBRE	TOTAL		
▶	MONITOR	9		
	LAPTOP	23		
	MICROFONO	19		
	PC	8		
	TECLADO	5		

1.7 Traer todos los productos y la suma total vendida por producto

FULL OUTER JOIN no es soportado por MySQL nativamente, por lo que se realiza una UNION entre LEFT JOIN y RIGHT JOIN

```
60      -- 1.7) Traer todos los productos y la suma total vendida por producto
61 •  SELECT PRO.NOMBRE, SUM(PRO.PRECIO * VEN.CANTIDAD) AS TOTAL
62   FROM PRODUCTOS AS PRO
63   LEFT JOIN VENTAS AS VEN ON PRO.ID_PRODUCTO = VEN.ID_PRODUCTO
64   GROUP BY PRO.NOMBRE
65
66 UNION
67
68 SELECT PRO.NOMBRE, SUM(PRO.PRECIO * VEN.CANTIDAD) AS TOTAL
69   FROM PRODUCTOS AS PRO
70   RIGHT JOIN VENTAS AS VEN ON PRO.ID_PRODUCTO = VEN.ID_PRODUCTO
71   GROUP BY PRO.NOMBRE;
72
```

Result Grid		
	NOMBRE	TOTAL
▶	LAPTOP	69000.00
	PC	32000.00
	MOUSE	NULL
	TECLADO	750.00
	MONITOR	18000.00
	MICROFONO	6650.00
	AUDIFONOS	NULL

2. EJERCICIO PRÁCTICO: BD

2.1 Crea un diagrama relacional de BD para el escenario descrito

