

EXAMEN SQL – Bases de Datos - UNAHUR

Nombre y Apellido: DNI:
.....

Modelo Relacional. (PK = Negrita | FK linea)

cliente (**id_cli**, nombre, correo, país)

orden (**id_orden**, cliente_id, fecha)

producto (**id_prod**, nombre, precio)

orden_Detalle (**id_orden_detalle**, orden_id, producto_id, cantidad)

Resolver las siguientes consultas:

1. **Listar un informe con el código y nombre de los productos, su cantidad y monto total vendido. Cuya cantidad total vendida sea superior al promedio de cantidad vendida entre todos los productos.**

```
SELECT Pid_prod, P.nombre, SUM(OD.cantidad) cantidad,  
       SUM(OD.cantidad*P.precio) Monto_vendido  
FROM productos P INNER JOIN orden_detalle OD ON  
       Pid_prod=OD.producto_id  
GROUP BY Pid_prod, P.nombre  
HAVING cantidad > (SELECT AVG(od.cantidad) FROM orden_detalle OD)
```

2. **Mostrar sólo los productos que superen los \$5.000 en ventas totales.**

```
SELECT p.id_prod, p.nombre, SUM(od.cantidad * p.precio) AS total_ventas  
FROM Orden_Detalle od JOIN productos p ON od.producto_id = p.id_prod  
GROUP BY p.id_prod, p.nombre  
HAVING SUM (od.cantidad * p.precio) > 5000;
```

3. **Mostrar el total gastado por cada cliente en formato: cliente_id, nombre, total_gastado. Ordenado de manera que el mayor total se liste primero**

```
SELECT C.id_cli, C.nombre, SUM(P.precio*cantidad) total  
FROM clientes C INNER JOIN ordenes O ON C.id_cli = O.cliente_id  
       JOIN orden_detalle OD ON OD.orden_id=O.id_ordenes  
       JOIN productos P ON Pid_prod=OD.producto_id  
GROUP BY C.id_cli, C.nombre  
ORDER BY total DESC
```

4. En la tabla orden_detalle, cambiar el nombre de la columna cantidad por cantidades, y luego llevar a cabo el ingreso de 3 registros en la tabla.

```
ALTER TABLE orden_detalle CHANGE cantidad cantidades INT;  
o ALTER TABLE orden_detalle RENAME COLUMN cantidades TO cantidad;
```

```
INSERT INTO orden_detalle (id_orden_detalle, producto_id, orden_id, cantidades)  
VALUES (101, 1, 3, 500), (102, 2, 5, 200), (103, 3, 2, 800);
```

5. Eliminar de la tabla producto, aquellos que contengan la combinación “ada” dentro de su nombre. Y luego eliminar la tabla Producto.

```
DELETE FROM productos  
WHERE nombre LIKE '%ada%';
```

```
DROP TABLE productos
```

6. Listar los nombres de aquellos clientes que no realizaron ninguna compra.

```
SELECT C.id_cli, C.NOMBRE  
FROM cliente C LEFT JOIN ordenes OR ON C.id_cli = OR. id_ordenes  
GROUP BY C.id_cli, C.NOMBRE  
HAVING COUNT(OR. id_ordenes) = 0;
```

```
-----OTRA FORMA POSIBLE -----  
SELECT c.nombre  
FROM clientes C LEFT JOIN ordenes O ON C.id_cli=O.cliente_id  
WHERE O.cliente_id IS null
```

7. Mostrar un detalle de los clientes que han realizado compras en el mes de enero del 2021 y no realizaron ninguna en el mes de marzo del 2021

```
SELECT *  
FROM CLIENTE c  
WHERE exists (SELECT OR.cliente_id FROM ORDEN OR  
              WHERE OR.Fecha BETWEEN '20210101' AND '20210131'  
              AND OR.cliente_id = c.id_cli)  
              AND NOT EXISTS (SELECT ORD.cliente_id FROM ORDEN ORD  
                              WHERE ORD.Fecha BETWEEN '20210301' AND '20210331'  
                              AND ORD.cliente_id = c.id_cli)
```

```
---OTRA FORMA ---
```

```
SELECT C.*  
FROM clientes C INNER JOIN ordenes O ON C.id_cli=O.cliente_id  
WHERE O.fecha BETWEEN '20210101' AND '20210131'  
      AND C.id_cli NOT IN (SELECT O.cliente_id FROM ordenes O  
                          WHERE O.fecha BETWEEN '20210301' AND '20210331')
```

8.- Generar el código para crear la tabla Orden_detalle, y luego actualizar un 35% aquellas cantidades que tiene 10 unidades.

```
CREATE TABLE orden_detalle (  
  
    id_orden_detalle INT PRIMARY KEY AUTO_INCREMENT,  
    orden_id INT NOT NULL,  
    producto_id INT NOT NULL,  
    cantidad INT NOT NULL,  
  
    FOREIGN KEY (orden_id) REFERENCES ordenes(id_orden),  
    FOREIGN KEY (producto_id) REFERENCES productos(id_prod)  
  
UPDATE orden_detalle  
SET cantidad = cantidad * 1.35  
WHERE cantidad = 10;
```