

Mario Rubio Avila BD Tarea 3.1

Información sobre la presentación:

Las sentencias recuadradas son las sentencias finales que solicitan el ejercicio o el resto son sentencias individuales usadas para ir conformando la sentencia final dada la complejidad de estas o información adicional usadas para el entendimiento.

Algunas sentencias, aunque no se solicita la ordenación se las ordena para mostrar más claro el resultado.

Si aparecen dos sentencias finales para un mismo enunciado es porque no a sido posible discernir si se solicitaba una o la otra

Las sentencias por encima de la 15 se entrega ya que se encontraban hechas cuando se indico que no se entregasen más de la 15.

1.-Número e importe de todos los pedidos realizados en los últimos 60 días

Con respecto a día de hoy

```
SELECT NUM, TOTAL FROM PEDIDOS WHERE FECHA >=(current_date -  
INTERVAL '60' DAY);
```

Con respecto a FECHA_PREVISTA

```
SELECT NUM, TOTAL FROM PEDIDOS WHERE FECHA >=(FECHA_PREVISTA -  
INTERVAL '60' DAY);
```

2.-Número e importe de los pedidos cuyo importe esté entre 100 y 200 €

```
SELECT TOTAL, NUM FROM PEDIDOS WHERE TOTAL>=100 AND  
TOTAL<=200;
```

3.-Código y nombre de los productos ordenados ascendentemente por precio y nombre.

* SI USAS ORDER BY SIN ESPECIFICAR ES ASCENDENTE SI DESEAS ALGO DESCENDENTE UTILIZAR desc.

** Ordenado por precio :*

```
SELECT CODIGO, NOMBRE FROM PRODUCTOS ORDER BY PRECIO ASC;
```

** Ordenado por nombre :*

```
SELECT CODIGO, NOMBRE FROM PRODUCTOS ORDER BY NOMBRE ASC;
```

** Ordenado por las dos cosas :*

```
SELECT CODIGO, NOMBRE FROM PRODUCTOS ORDER BY PRECIO ASC,  
NOMBRE ASC;
```

4.-Clientes cuyo segundo apellido sea Perez

* SELECT CODIGO,NOMBRE,APELLIDOS,EDAD FROM CLIENTES WHERE APELLIDOS

- * Al principio-> descripcion LIKE 'mente %' OR
- * En el medio-> descripcion LIKE '% mente %' OR
- * Al final-> descripcion LIKE '% mente';

```
SELECT * FROM CLIENTES WHERE APELLIDOS LIKE '% Perez';
```

5.-Número total de productos que vende la empresa (en la columna debe aparecer “Nº de productos”):

```
SELECT COUNT(*) "Nº de productos" FROM PRODUCTOS;
```

6.-Numero total de productos que no han sido pedidos

Han sido comprados :

```
SELECT COUNT(*) FROM PRODUCTOS WHERE CODIGO IN (SELECT PRODUCTO FROM LINEAS);
```

No han sido comprados :

```
SELECT COUNT(*) FROM PRODUCTOS WHERE NOT (CODIGO IN (SELECT
PRODUCTO FROM LINEAS));
```

7.-De cada pedido, mostrar su número, importe y datos del cliente

El importe es cogido del total de la table pedidos

```
SELECT NUM, TOTAL, CODIGO, NOMBRE, APELLIDOS, EDAD
FROM PEDIDOS,CLIENTES
WHERE PEDIDOS.CLIENTE = CLIENTES.CODIGO;
```

8.-Código, nombre del cliente y número total de pedidos que ha hecho cada cliente, ordenado de más a menos pedidos

** Parte del calculo :*

```
SELECT CLIENTE, COUNT(CLIENTE) AS NPEDIDOS FROM PEDIDOS GROUP BY CLIENTE ORDER BY
NPEDIDOS ASC;
```

** Parte de la unión :*

```
SELECT * FROM CLIENTES INNER JOIN PEDIDOS ON CLIENTES.CODIGO = PEDIDOS.CLIENTE;
```

** Muestra los pedidos de los clientes ordenados de mayor a menor*

```
SELECT CLIENTES.CODIGO , CLIENTES.NOMBRE, CLIENTES.APELLIDOS, COUNT(*) AS NPEDIDOS
FROM CLIENTES
INNER JOIN PEDIDOS ON CLIENTES.CODIGO = PEDIDOS.CLIENTE
GROUP BY CLIENTES.CODIGO, CLIENTES.NOMBRE , CLIENTES.APELLIDOS
ORDER BY NPEDIDOS DESC;
```

**Otra manera mediante correlación :*

```
SELECT PEDIDOS.CLIENTE, CLIENTES.NOMBRE, CLIENTES.APELLIDOS, COUNT(*) AS NPEDIDOS
FROM CLIENTES, PEDIDOS
WHERE CLIENTES.CODIGO = PEDIDOS.CLIENTE
GROUP BY PEDIDOS.CLIENTE, CLIENTES.NOMBRE, CLIENTES.APELLIDOS
ORDER BY NPEDIDOS DESC;
```

9.-Código, nombre del cliente y número total de pedidos que ha realizado cada cliente durante 2016

```
SELECT CLIENTES.CODIGO , CLIENTES.NOMBRE, COUNT(CLIENTE) AS NPEDIDOS
FROM CLIENTES
INNER JOIN PEDIDOS ON CLIENTES.CODIGO = PEDIDOS.CLIENTE
WHERE PEDIDOS.FECHA LIKE '%/16'
GROUP BY CLIENTES.CODIGO, CLIENTES.NOMBRE, CLIENTE
ORDER BY NPEDIDOS ASC;
```

10.-Código, nombre y número total de pedidos de los clientes que han realizado más de un pedido.

```
SELECT CLIENTES.CODIGO , CLIENTES.NOMBRE, COUNT(CLIENTE) AS NPEDIDOS
FROM CLIENTES
INNER JOIN PEDIDOS ON CLIENTES.CODIGO = PEDIDOS.CLIENTE
GROUP BY CLIENTES.CODIGO, CLIENTES.NOMBRE, CLIENTE
HAVING COUNT(CLIENTE) > 1
ORDER BY NPEDIDOS ASC;
```

11.-Para cada pedido mostrar su número, código del cliente y nº total de líneas que tiene .

```
SELECT PEDIDOS.NUM AS "Numero Pedido", PEDIDOS.CLIENTE AS "Numero Cliente",
COUNT(NUM_PEDIDO) AS NLINEAS
FROM PEDIDOS
INNER JOIN LINEAS ON PEDIDOS.NUM = NUM_PEDIDO
GROUP BY PEDIDOS.NUM , PEDIDOS.CLIENTE, NUM_PEDIDO
ORDER BY "Numero Pedido" ASC;
```

12.-Código de cliente, nombre de producto y cantidad total que ha pedido cada cliente de cada producto

Muestra el total de cada producto pedido o no pedido.

```
SELECT PRODUCTOS.NOMBRE AS NOMBREPRODUCTO, SUM(CANTIDAD) AS  
CANTIDADPRODUCTO FROM LINEAS
```

```
RIGHT JOIN PRODUCTOS ON LINEAS.PRODUCTO = CODIGO
```

```
GROUP BY PRODUCTOS.NOMBRE;
```

Muestra la suma de cada producto comprado por cada cliente

```
SELECT PEDIDOS.CLIENTE, LINEAS.NUM_PEDIDO AS "NUMERO PEDIDO", PRODUCTOS.NOMBRE  
AS NOMBREPRODUCTO, SUM(CANTIDAD) AS CANTIDADPRODUCTO FROM LINEAS
```

```
INNER JOIN PRODUCTOS ON LINEAS.PRODUCTO = CODIGO
```

```
INNER JOIN PEDIDOS ON PEDIDOS.NUM = LINEAS.NUM_PEDIDO
```

```
GROUP BY LINEAS.NUM_PEDIDO, PRODUCTOS.NOMBRE, PEDIDOS.CLIENTE
```

```
ORDER BY "Numero Pedido" ASC;
```

13.-Para cada cliente mostrar su código, nombre , numero e importe del cada uno de sus pedidos

Cogiendo el importe de la tabla de líneas mediante una agrupación

```
SELECT PEDIDOS.NUM AS "NUMERO PEDIDO", PEDIDOS.CLIENTE , CLIENTES.NOMBRE,  
CLIENTES.APELLIDOS , SUM(LINEAS.IMPORTE) AS IMPORTEPEDIDO
```

```
FROM LINEAS, PEDIDOS, CLIENTES
```

```
WHERE PEDIDOS.NUM = LINEAS.NUM_PEDIDO AND CLIENTES.CODIGO =  
PEDIDOS.CLIENTE
```

```
GROUP BY PEDIDOS.NUM, PEDIDOS.CLIENTE, CLIENTES.NOMBRE, CLIENTES.APELLIDOS
```

```
ORDER BY PEDIDOS.NUM ASC;
```

Cogiendo el importe de la tabla pedidos el total

```
SELECT CLIENTES.CODIGO AS "CODIGO CLIENTE ", CLIENTES.NOMBRE, CLIENTES.APELLIDOS,  
PEDIDOS.NUM AS "N PEDIDO", PEDIDOS.TOTAL AS "TOTAL DEL PEDIDO"
```

```
FROM CLIENTES
```

```
LEFT JOIN PEDIDOS ON CLIENTES.CODIGO = PEDIDOS.CLIENTE
```

```
GROUP BY CLIENTES.CODIGO , CLIENTES.NOMBRE, CLIENTES.APELLIDOS ,  
PEDIDOS.NUM, PEDIDOS.TOTAL
```

```
ORDER BY CLIENTES.CODIGO ASC;
```

14.-Para cada cliente menor de edad mostrar su código y nombre, el importe más alto, el más bajo de los pedidos que ha realizado

Hago una unión de la columna nombre y apellidos con el fin de practicar con la unión de campos.

```
SELECT CLIENTES.CODIGO, CLIENTES.NOMBRE || ' ' || CLIENTES.APELLIDOS AS "Nombre y Apellidos", P.TOTAL  
  
FROM PEDIDOS P  
  
INNER JOIN CLIENTES ON CLIENTES.CODIGO = P.CLIENTE  
  
WHERE (CLIENTES.EDAD <18) AND ((TOTAL IN (SELECT MAX(TOTAL) FROM PEDIDOS))  
OR (TOTAL IN (SELECT MIN(TOTAL) FROM PEDIDOS)));
```

15.-Mostrar el código del producto, el nº de veces que ha sido pedido y la cantidad total de unidades que se han pedido

```
SELECT PRODUCTOS.CODIGO, PRODUCTOS.NOMBRE, SUM(NVL(LINEAS.CANTIDAD,0)) AS SUMAPRODUCTOS FROM LINEAS  
  
RIGHT JOIN PRODUCTOS ON PRODUCTOS.CODIGO = LINEAS.PRODUCTO  
  
GROUP BY PRODUCTOS.CODIGO, PRODUCTOS.NOMBRE;
```

17.-Datos del producto más caro del pedido 1

Nos indica el producto mas caro y mas barato

```
SELECT * FROM PRODUCTOS WHERE PRODUCTOS.PRECIO IN (SELECT  
MAX(PRODUCTOS.PRECIO) FROM PRODUCTOS) OR PRODUCTOS.PRECIO IN (SELECT  
MIN(PRODUCTOS.PRECIO) FROM PRODUCTOS);
```

Nos indica el producto mas caro del pedido 1

```
SELECT * FROM PRODUCTOS WHERE PRODUCTOS.PRECIO IN (SELECT  
MAX(PRODUCTOS.PRECIO) FROM PRODUCTOS WHERE CODIGO IN (SELECT LINEAS.PRODUCTO  
FROM LINEAS WHERE LINEAS.NUM_PEDIDO = 1));
```

23.-Datos del pedido más caro y del más barato

Muestra quien es el mínimo y el máximo

```
SELECT MAX(TOTAL) AS "Pedido Mayor", MIN(TOTAL) AS "Pedido Menor" FROM PEDIDOS;
```

Dos posibilidades que hacen lo mismo

```
SELECT * FROM PEDIDOS WHERE TOTAL >= ALL (SELECT MAX(TOTAL) FROM PEDIDOS) OR  
TOTAL <= ALL (SELECT MIN(TOTAL) FROM PEDIDOS);
```

```
SELECT * FROM PEDIDOS WHERE TOTAL IN (SELECT MAX(TOTAL) FROM PEDIDOS) OR TOTAL IN  
(SELECT MIN(TOTAL) FROM PEDIDOS);
```

26.-(Con subconsultas) Datos de los clientes que han pedido el producto de nombre 'PANTALON'.

Con nombre exacto

```
SELECT * FROM CLIENTES WHERE CLIENTES.CODIGO IN (SELECT CLIENTE FROM PEDIDOS
WHERE PEDIDOS.NUM IN (SELECT NUM_PEDIDO FROM LINEAS WHERE PRODUCTO IN (SELECT
CODIGO FROM PRODUCTOS WHERE NOMBRE = 'PANTALÓN')));
```

Con nombre con pequeño cambio por la tilde

```
SELECT * FROM CLIENTES WHERE CLIENTES.CODIGO IN (SELECT CLIENTE FROM PEDIDOS
WHERE PEDIDOS.NUM IN (SELECT NUM_PEDIDO FROM LINEAS WHERE PRODUCTO IN (SELECT
CODIGO FROM PRODUCTOS WHERE NOMBRE LIKE 'PANTAL%N')));
```

27.-(Sin subconsultas) Datos de los clientes que han pedido el producto de nombre 'PANTALON'.

```
SELECT CLIENTES.* FROM CLIENTES
    INNER JOIN PEDIDOS ON (CLIENTES.CODIGO = PEDIDOS.CLIENTE)
    INNER JOIN LINEAS ON (PEDIDOS.NUM = LINEAS.NUM_PEDIDO)
    INNER JOIN PRODUCTOS ON (LINEAS.PRODUCTO = PRODUCTOS.CODIGO)
    WHERE PRODUCTOS.NOMBRE LIKE 'PANTAL%N';
```