

EXAMEN PASADO

--1. Obtener el nombre completo (nombre y apellido) del empleado que no tiene jefe.

```
SELECT * FROM EMPLEADOS;
```

```
SELECT NOMBRE || ' ' || APELLIDO1  
FROM EMPLEADOS  
WHERE CODIGOJEFE IS NULL;
```

EJERCICIO DE ANTONIO

-- Tratar la salida de una consulta como si fuera una variable que cambia con cada fila

```
SELECT E.ENAME, E.JOB, E.DEPTNO, (SELECT D.DNAME FROM DEPT D WHERE  
D.DEPTNO = E.DEPTNO) AS UNA_CONSULTA  
FROM EMP E;
```

-- ACTUALIZA EL LIMITE DE CREDITO A LO COMPRADO HASTA AHORA.

-- CALCULAR CUANDO ME DEBE

```
SELECT P.CODIGOCLIENTE, nvl(SUM(DP.CANTIDAD*DP.PRECIOUNIDAD),0)  
FROM PEDIDOS P, DETALLEPEDIDOS DP  
WHERE P.CODIGOPEDIDO = DP.CODIGOPEDIDO  
GROUP BY P.CODIGOCLIENTE;
```

--11..Muestra el código, nombre y gama de los productos que nunca se han pedido (detalle pedidos)

```
select p.codigoproducto, p.nombre, p.gama  
from productos p, GAMASPRODUCTOS g  
where p.gama=g.gama  
and P.CODIGOPRODUCTO NOT IN (select CODIGOPRODUCTO from detallepedidos);
```

-- EL EXISTS ES TRUE SI SE CUMPLE O FALSE SI NO SE CUMPLE

```
select p.codigoproducto, p.nombre, p.gama  
from productos p, GAMASPRODUCTOS g  
where p.gama=g.gama  
and NOT exists (select CODIGOPRODUCTO from detallepedidos where  
codigoproducto=p.CODIGOPRODUCTO);
```

-- CUANTO SE HA GASTADO CADA CLIENTE

```
SELECT C.CODIGOCLIENTE,c.nombrecliente, (SELECT  
NVL(SUM(DP.PRECIOUNIDAD*DP.CANTIDAD),0)  
FROM PEDIDOS P, DETALLEPEDIDOS DP  
WHERE DP.CODIGOPEDIDO = P.CODIGOPEDIDO  
AND P.CODIGOCLIENTE = C.CODIGOCLIENTE) AS GASTADO  
FROM CLIENTES C;
```

-- DEVUELVE UN LISTADO DE LAS DIFERENTE GAMAS DE PRODUCTO QUE HA COMPRADO CADA CLIENTE

```

SELECT DISTINCT C.NOMBRECLIENTE, PR.GAMA
FROM PRODUCTOS PR, DETALLEPEDIDOS DP, PEDIDOS P, CLIENTES C
WHERE PR.CODIGOPRODUCTO = DP.CODIGOPRODUCTO
AND DP.CODIGOPEDIDO = P.CODIGOPEDIDO
AND P.CODIGOCLIENTE = C.CODIGOCLIENTE;

```

```

--74 Devuelve las oficinas donde no trabajan ninguno de los empleados
-- que hayan sido los representantes de ventas de algún cliente
-- que haya realizado la compra de algún producto de la gama Frutales.
SELECT O.*

```

```

FROM OFICINAS O
WHERE O.CODIGOOFICINA NOT IN
      (SELECT DISTINCT E.CODIGOOFICINA
       FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP, PRODUCTOS
PR, EMPLEADOS E
       WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
       AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
       AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO
       --AND UPPER(E.PUESTO) = 'REPRESENTANTE VENTAS'
       AND UPPER(PR.GAMA) = 'FRUTALES'
       AND C.CODIGOEMPLOADOREPVENTAS = E.CODIGOEMPLEADO);

```

```

--LOS CLIENTES QUE NO HAN COMPRADO FRUTALES.

```

```

SELECT * FROM (SELECT C.NOMBRECLIENTE, SUM(PA.CANTIDAD)
FROM CLIENTES C, PAGOS PA
WHERE C.CODIGOCLIENTE = PA.CODIGOCLIENTE
GROUP BY C.NOMBRECLIENTE
ORDER BY 2 DESC
)

```

```

WHERE ROWNUM = 1;

```

```

-- Realiza una vista que muestre los datos de un empleado
-- (nombre, apellidos, ciudad de la oficina)
-- y lo mismo para su jefe (en la misma fila)
-- y del jefe de su jefe

```

```

SELECT E.NOMBRE, O.CIUDAD, J.NOMBRE, OJ.CIUDAD, JJ.NOMBRE, OJJ.CIUDAD
FROM EMPLEADOS E, OFICINAS O, EMPLEADOS J, OFICINAS OJ, EMPLEADOS JJ,
OFICINAS OJJ
WHERE E.CODIGOOFICINA = O.CODIGOOFICINA
AND J.CODIGOOFICINA = OJ.CODIGOOFICINA
AND E.CODIGOJEFE = J.CODIGOEMPLEADO
AND JJ.CODIGOOFICINA = OJJ.CODIGOOFICINA
AND J.CODIGOJEFE = JJ.CODIGOEMPLEADO;

```

```

--18. Mostrar el código de los pedidos donde se haya vendido el producto de la gama
'Aromáticas' mas caro.

```

```

SELECT MAX(DP.PRECIOUNIDAD)
FROM DETALLEPEDIDOS DP, PRODUCTOS PR
WHERE DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO
AND UPPER(PR.GAMA) = 'AROMÁTICAS';

```

```

select p.codigoproducto, p.nombre, p.gama
from productos p, GAMASPRODUCTOS g
where p.gama=g.gama
AND P.CODIGOPRODUCTO NOT IN (SELECT CODIGOPRODUCTO FROM
DETALLEPEDIDOS);

```

--2. Cuánto dinero se hubiera gastado cada cliente si hubiera vendido
--los productos al precio de venta registrado en productos.

```

SELECT * FROM PRODUCTOS;

```

```

SELECT DISTINCT C.CODIGOCLIENTE, (DP.CANTIDAD*PR.PRECIOVENTA) AS
PRECIO_INVENTADO
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP, PRODUCTOS PR
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO;

```

--3. Si la diferencia entre el precio de venta y el precio del proveedor
--es el margen. Que margen hubiera tenido cada pedido.

```

SELECT DP.CODIGOPEDIDO, DP.CANTIDAD*(PR.PRECIOVENTA -
PR.PRECIOPROVEEDOR) AS MARGEN
FROM PRODUCTOS PR, DETALLEPEDIDOS DP
WHERE DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO;

```

--4. Obtener el país donde han invertido más dinero en compras.

```

SELECT O.PAIS
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP, OFICINAS O, EMPLEADOS E
WHERE O.CODIGOOFICINA = E.CODIGOOFICINA
AND E.CODIGOEMPLEADO = C.CODIGOEMPLEADOREPVENTAS
AND C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
GROUP BY O.PAIS
HAVING SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) = (SELECT
MAX(SUM(DP.CANTIDAD*DP.PRECIOUNIDAD))
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP,
PRODUCTOS PR, OFICINAS O, EMPLEADOS E
WHERE O.CODIGOOFICINA = E.CODIGOOFICINA
AND E.CODIGOEMPLEADO =
C.CODIGOEMPLEADOREPVENTAS
AND C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO

```

```
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO  
GROUP BY O.PAIS);
```

--5. Obtener el cliente que ha hecho el pedido con el comentario más largo
--(número de letras del comentario).

```
SELECT C.NOMBRECLIENTE  
FROM CLIENTES C, PEDIDOS P  
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE  
AND LENGTH(P.COMENTARIOS) = ( SELECT MAX(LENGTH(P.COMENTARIOS))  
FROM CLIENTES C, PEDIDOS P  
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE);
```

--6. Teniendo presente que el dinero de los pedidos rechazados es devuelto
--al cliente, obtener cuanto se ha gastado cada cliente en 'Plantas aromáticas'.

```
SELECT C.NOMBRECLIENTE,SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS GASTADO  
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP, PRODUCTOS PR  
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE  
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO  
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO  
AND UPPER(ESTADO)='RECHAZADO'  
AND UPPER(PR.GAMA) = 'AROMÁTICAS'  
GROUP BY C.NOMBRECLIENTE;
```

--7. Obtener el número de clientes distintos que han comprado en cada gama.

```
SELECT DISTINCT C.NOMBRECLIENTE, PR.GAMA  
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP,PRODUCTOS PR,  
GAMASPRODUCTOS G  
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE  
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO  
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO  
AND PR.GAMA = G.GAMA;
```

--8. Obtener los empleados que están en oficinas españolas y que no tienen
--ningún cliente a su cargo.

```
SELECT E.NOMBRE || ' '|| E.APELLIDO1 || ' '|| E.APELLIDO2 AS EMPLEADO  
FROM OFICINAS O, EMPLEADOS E  
WHERE O.CODIGOOFICINA = E.CODIGOOFICINA  
AND UPPER(O.PAIS)='ESPAÑA'
```

AND E.CODIGOEMPLEADO NOT IN (SELECT CODIGOEMPLEADOREPVENTAS FROM CLIENTES);

--9. Obtener el nombre de los clientes y el nombre de los empleados donde
--los empleados trabajan en el mismo país que los clientes.

```
SELECT C.NOMBRECLIENTE AS CLIENTE, E.NOMBRE AS EMPLEADO
FROM OFICINAS O, EMPLEADOS E, CLIENTES C
WHERE O.CODIGOOFICINA = E.CODIGOOFICINA
AND E.CODIGOEMPLEADO = C.CODIGOEMPLEADOREPVENTAS
AND O.PAIS = C.PAIS;
```

--10. Obtener el país donde se ha gastado más dinero en comprar productos.

```
SELECT C.PAIS
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
GROUP BY C.PAIS
HAVING SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) = (SELECT
MAX(SUM(DP.CANTIDAD*DP.PRECIOUNIDAD))
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP,
PRODUCTOS PR
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO
GROUP BY C.PAIS);
```

--11. EMMANUEL tienes problemas con su jefe y quieres quejarse al jefe de su jefe, pero no sabe

--ni quien se, ni en qué oficina trabaja (ciudad). ¿Podrías ayudarlo, necesita saber el nombre

--completo, el email, y la ciudad donde está su oficina?

```
SELECT JEFEDJEFE.NOMBRE, JEFEDJEFE.APELLIDO1, JEFEDJEFE.APELLIDO2,
JEFEDJEFE.EMAIL, O.CIUDAD
FROM OFICINAS O, EMPLEADOS E, EMPLEADOS JEFE, EMPLEADOS JEFEDJEFE
WHERE O.CODIGOOFICINA = E.CODIGOOFICINA
AND E.CODIGOJEFE = JEFE.CODIGOEMPLEADO
AND JEFE.CODIGOJEFE = JEFEDJEFE.CODIGOEMPLEADO
AND UPPER(E.NOMBRE) = 'EMMANUEL';
```

--12. Si por cada letra del comentario se le resta un día a la fecha de pedido,

--obtener el cliente(nombre) del pedido que se pidió el primero, en base a la fecha de pedido.

```
SELECT C.NOMBRECLIENTE
FROM CLIENTES C, PEDIDOS P
WHERE (ROUND(SYSDATE - P.FECHAPEDIDO,0) - LENGTH(P.COMENTARIOS)) = (
SELECT MAX(ROUND(SYSDATE - P.FECHAPEDIDO,0) - LENGTH(P.COMENTARIOS))
      FROM CLIENTES C, PEDIDOS P
      WHERE
C.CODIGOCLIENTE=P.CODIGOCLIENTE);
```

--13. Si de lo gastado (pedidos) por los clientes, los directores de oficina tienen una comisión del

--20%, y los representantes de ventas un 15%, obtén cuanta comisión tiene cada uno de los --empleados de estos tipos, mostrando el nombre del empleado.

```
SELECT DISTINCT E.NOMBRE, E.PUESTO, DECODE(UPPER(PUESTO),
'REPRESENTANTE VENTAS', SUM((DP.PRECIOUNIDAD*DP.CANTIDAD)*0.15),
'DIRECTOR OFICINA',SUM(DP.PRECIOUNIDAD*DP.CANTIDAD)*0.2) AS COMISION
FROM OFICINAS O, EMPLEADOS E, CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP
WHERE O.CODIGOOFICINA = E.CODIGOOFICINA
AND E.CODIGOEMPLEADO = C.CODIGOEMPLEADOREPVENTAS
AND C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
GROUP BY E.NOMBRE, E.PUESTO;
```

--14. Obtener en una sola consulta el nombre, ciudad y país de los empleados y el nombre ciudad
--y país de su jefe.

```
SELECT E.NOMBRE, O.CIUDAD, O.PAIS, JEFE.NOMBRE AS NOMBRE_JEFE,
O.CIUDAD, O.PAIS
FROM EMPLEADOS E, EMPLEADOS JEFE, OFICINAS O
WHERE O.CODIGOOFICINA=E.CODIGOOFICINA
AND O.CODIGOOFICINA = JEFE.CODIGOOFICINA
AND E.CODIGOJEFE = JEFE.CODIGOEMPLEADO;
```

-- PEDIDOS QUE HAN LLEGADO ENTRE 3 Y 5 DIAS
SELECT *
FROM PEDIDOS P
WHERE P.FECHAPEDIDO + 3 >= P.FECHAENTREGA
AND P.FECHAPEDIDO + 5 <= P.FECHAENTREGA;

```
SELECT *
FROM PEDIDOS P
WHERE P.FECHAESPERADA >= P.FECHAENTREGA
```

-- EN EL CASO DE QUE TE DIGA QUE SI NO TIENE FECHA DE ENTREGA
-- REGISTRADA NO VAN A LLEGAR.

UNION

SELECT *

FROM PEDIDOS P

WHERE P.FECHAENTREGA IS NULL

;

--15. Borrar los productos que no se han vendido nunca. Hacer posteriormente Rollback

SELECT *

FROM PRODUCTOS

WHERE CODIGOPRODUCTO NOT IN (SELECT CODIGOPRODUCTO FROM
DETALLEPEDIDOS);

DELETE

FROM PRODUCTOS

WHERE CODIGOPRODUCTO NOT IN (SELECT CODIGOPRODUCTO FROM
DETALLEPEDIDOS);

1. Muestra la ciudad y el código postal de las oficinas de España.

select ciudad, codigopostal

from OFICINAS

where lower(pais) = 'españa';

2. Obtener el nombre y apellidos del jefe de la empresa.

select nombre, apellido1, apellido2

from empleados

where codigojefe is null;

3. Mostrar el nombre y cargo de los empleados que no sean directores de oficina.

select nombre, puesto

from empleados

where lower(puesto) <> 'director oficina';

4. Muestra el número de empleados que hay en la empresa.

select count(*) as "Num empleados"

from empleados;

5. Muestra el número de clientes norteamericanos.

select count(*) as "Num clientes"

from clientes

where upper(pais) = 'USA';

6. Número de clientes de cada país.

select pais, count(*) as "Num clientes"

from clientes

group by pais;

MULTITABLAS

7. Muestra el nombre del cliente y el nombre de su representante de ventas (si lo tiene).

```
select c.nombrecliente as "Nombre cliente", e.nombre as "Nombre representante"
from clientes c, empleados e
where c.codigoempleadoventas=e.CODIGOEMPLEADO;
```

8. Nombre de los clientes que hayan hecho un pago en 2007

```
select distinct c.nombrecliente
from clientes c, pagos p
where c.CODIGOCLIENTE=p.CODIGOCLIENTE
and p.FECHAPAGO like '%/2007';
```

9. Los posibles estados de un pedido.

```
select distinct lower(estado)
from pedidos
```

10. Muestra el número de pedido, el nombre del cliente, la fecha de entrega y la fecha requerida de los pedidos que no han sido entregados a tiempo.

```
select p.codigopedido, c.nombrecliente, p.fechaentrega, p.fechaesperada
from clientes c, pedidos p
where c.codigocliente = p.CODIGOCLIENTE
and p.FECHAESPERADA < p.fechaentrega;
```

11. Muestra el código, nombre y gama de los productos que nunca se han pedido (detalle pedidos).

```
select p.codigoproducto, p.nombre, p.gama
from productos p, GAMASPRODUCTOS g
where p.gama=g.gama
and not exists (select CODIGOPRODUCTO from detallepedidos where
codigoproducto=p.CODIGOPRODUCTO);
```

12. Muestra el nombre y apellidos de los empleados que trabajan en Barcelona.

```
select e.nombre, e.apellido1 || ' ' || e.apellido2
from empleados e, oficinas o
where e.codigooficina = o.codigooficina
and trim(lower(o.ciudad)) = 'barcelona';
```

13. Muestra el código y la cantidad de veces que se ha pedido un producto al menos una vez.

```
select p.codigoproducto, sum(dp.cantidad) as "cantidad pedida"
from productos p, detallepedidos dp
where p.codigoproducto = dp.codigoproducto
group by p.codigoproducto;
```

14. Muestra el nombre de los clientes de Miami que han realizado algún pedido.

```
select distinct c.nombrecliente
from clientes c, pedidos p
where c.codigocliente=p.codigocliente
```


and trim(lower(c.ciudad))='miami';

15. Mostrar el precio final de cada pedido.

```
select p.codigopedido, sum(dp.cantidad * dp.PRECIOUNIDAD) as total
from pedidos p, detallepedidos dp
where p.codigopedido = dp.codigopedido
group by p.codigopedido;
```

16. Mostrar lo que ha pagado cada cliente.

```
select nombrecliente, sum(p.cantidad) as total_pagado
from clientes c, pagos p
where c.codigocliente=p.codigocliente
group by c.nombrecliente;
```

17. Mostrar el numero de productos de cada gama.

```
select p.gama, count(*) as numero_productos
from productos p, gamasproductos gp
where p.gama = gp.gama
group by p.gama;
```

18. Mostrar el código de los pedidos donde se haya vendido el producto de la gama 'Aromáticas' mas caro.

```
select distinct pe.codigopedido
from pedidos pe, detallepedidos dp
where pe.codigopedido = dp.codigopedido
and dp.codigoproducto in (select codigoproducto
                        from productos
                        where precioventa = (select max(precioventa)
                                           from productos p, gamasproductos g
                                           where p.gama = g.gama
                                           and lower(g.gama) = 'aromáticas'))
```

19. Mostrar el código de los pedidos donde se hayan vendido mas de 6 productos.

```
select pe.codigopedido
from pedidos pe, detallepedidos dp
where pe.codigopedido = dp.codigopedido
group by pe.codigopedido
having count(*)>=6;
```

20. Mostrar el codigo de los pedidos donde el precio del pedido sea superior a la media de todos los pedidos.

```
select pe.codigopedido
from pedidos pe
where
(select sum(dp.cantidad * dp.PRECIOUNIDAD) as total
 from pedidos p, detallepedidos dp
 where p.codigopedido = dp.codigopedido and pe.codigopedido = p.codigopedido
 group by p.codigopedido)
```

>

```
(select avg(t.total)
from (select p.codigopedido, sum(dp.cantidad * dp.PRECIOUNIDAD) as total
      from pedidos p, detallepedidos dp
      where p.codigopedido = dp.codigopedido
      group by p.codigopedido) t);
```

21. Realiza una vista que muestre los datos de un empleado (nombre, apellidos, ciudad de la oficina) y lo mismo para su jefe (en la misma fila).

```
create or replace view empleados_jefes as
select e_subor.nombre as nombre_subor,
       e_subor.apellido1 || ' ' || e_subor.apellido2 as apellidos_subor,
       o_subor.ciudad as ciudad_subor,
       e_jefe.nombre as nombre_jefe,
       e_jefe.apellido1 || ' ' || e_jefe.apellido2 as apellidos_jefe,
       o_subor.ciudad as ciudad_jefe
from empleados e_subor, empleados e_jefe, oficinas o_subor, oficinas o_jefe
where e_subor.codigojefe = e_jefe.codigoempleado
and o_subor.codigooficina = e_subor.codigooficina
and e_jefe.codigooficina = o_jefe.CODIGOOFICINA;
```

```
select * from empleados_jefes;
```

22. Realiza una vista que muestre el código de pedido y su total en euros.

```
create or replace view pedidos_total as
select p.codigopedido, sum(dp.cantidad * dp.PRECIOUNIDAD) as total
from pedidos p, detallepedidos dp
where p.codigopedido = dp.codigopedido
group by p.codigopedido;
```

```
select * from pedidos_total;
```

23. Realiza una vista con la información del pedido (código, fechapedido, fechaesperada, fechaentrega, nombre cliente y total en euros) ordenado por total de forma descendente.

```
create or replace view info_pedido as
select p.codigopedido,
       p.fechapedido,
       p.fechaesperada,
       p.fechaentrega,
       c.nombrecliente,
       pt.total
from pedidos p, clientes c, PEDIDOS_TOTAL pt
where p.CODIGOCLIENTE = c.CODIGOCLIENTE
and pt.CODIGOPEDIDO = p.CODIGOPEDIDO
order by pt.total desc;
```

```
select * from info_pedido;
```

24. Devolverme la gama de productos mas vendida. Sin vistas

```
select t.gama, t.cantidad
from (select p.gama, sum(dp.cantidad) as cantidad
      from detallepedidos dp, productos p
      where p.codigoproducto = dp.codigoproducto
      group by p.gama
      order by cantidad desc) t
where rownum = 1;
```

25. Devolverme la gama de productos mas vendida. Usa vistas

```
create or replace view gamas_vendidas as
select p.gama, sum(dp.cantidad) as cantidad
from detallepedidos dp, productos p
where p.codigoproducto = dp.codigoproducto
group by p.gama;
```

```
select gv.gama, gv.cantidad
from gamas_vendidas gv
where gv.CANTIDAD = (select max(gv.cantidad)
                    from gamas_vendidas gv);
```

26. Muestra el pais(cliente) donde menos pedidos se hacen.

```
create or replace view pedidos_paises as
select c.pais, count(*) as num_pedidos
from clientes c, info_pedido ip
where c.nombrecliente = ip.nombrecliente
group by c.pais;
```

```
select pp.pais, pp.num_pedidos
from PEDIDOS_PAISES pp
where pp.num_pedidos = (select min(num_pedidos)
                      from PEDIDOS_PAISES);
```

Ejercicios hechos en clase

--LOS CLIENTES QUE NO HAN COMPRADO FRUTALES.

```
SELECT DISTINCT C.*
```

```
FROM CLIENTES C, PEDIDOS P,detallepedidos DP, productos PR
```

```
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
```

```
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
```

```
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO
```

```
AND UPPER(PR.GAMA) <> 'FRUTALES';
```

-- HACER CON EL ROWNUM

-- SABRIENDO QUE ES UNICO OBTENER EL CLIENTES QUE HA PAGADO MÁS

--1. LO QUE HA GASTADO CADA CLIENTE

```
SELECT C.NOMBRECLIENTE, SUM(PA.CANTIDAD)
FROM CLIENTES C, PAGOS PA
WHERE C.CODIGOCLIENTE = PA.CODIGOCLIENTE
GROUP BY C.NOMBRECLIENTE;
```

```
-- ORDENO POR LO GASTADO DESC
SELECT C.NOMBRECLIENTE, SUM(PA.CANTIDAD)
FROM CLIENTES C, PAGOS PA
WHERE C.CODIGOCLIENTE = PA.CODIGOCLIENTE
GROUP BY C.NOMBRECLIENTE
ORDER BY 2 DESC;
```

```
SELECT * FROM (SELECT C.NOMBRECLIENTE, SUM(PA.CANTIDAD)
                FROM CLIENTES C, PAGOS PA
                WHERE C.CODIGOCLIENTE = PA.CODIGOCLIENTE
                GROUP BY C.NOMBRECLIENTE
                ORDER BY 2 DESC
            )
WHERE ROWNUM = 1;
```

```
-- LA FORMA NORMAL
SELECT MAX(SUM(PA.CANTIDAD))
FROM PAGOS PA
GROUP BY PA.CODIGOCLIENTE;
```

```
SELECT C.NOMBRECLIENTE, SUM(PA.CANTIDAD)
FROM CLIENTES C, PAGOS PA
WHERE C.CODIGOCLIENTE = PA.CODIGOCLIENTE
GROUP BY C.NOMBRECLIENTE
HAVING SUM(PA.CANTIDAD) = (
    SELECT MAX(SUM(PA.CANTIDAD))
    FROM PAGOS PA
    GROUP BY PA.CODIGOCLIENTE
);
```

```
-- CUANTO SE HA GASTADO CADA CLIENTE
SELECT C.CODIGOCLIENTE, c.nombrecliente, (SELECT
    NVL(SUM(DP.PRECIOUNIDAD*DP.CANTIDAD),0)
        FROM PEDIDOS P, DETALLEPEDIDOS DP
        WHERE DP.CODIGOPEDIDO = P.CODIGOPEDIDO
        AND P.CODIGOCLIENTE = C.CODIGOCLIENTE) AS
    GASTADO
FROM CLIENTES C;
```

```
-- DEVUELVE UN LISTADO DE LAS DIFERENTE GAMAS DE PRODUCTO QUE HA
COMPRADO CADA CLIENTE
SELECT DISTINCT C.NOMBRECLIENTE, PR.GAMA
```

```
FROM PRODUCTOS PR, DETALLEPEDIDOS DP, PEDIDOS P, CLIENTES C
WHERE PR.CODIGOPRODUCTO = DP.CODIGOPRODUCTO
AND DP.CODIGOPEDIDO = P.CODIGOPEDIDO
AND P.CODIGOCLIENTE = C.CODIGOCLIENTE;
```

```
--61 Devuelve el nombre del producto del que se han vendido
-- más unidades.
-- (Tenga en cuenta que tendrá que calcular cuál es el
-- número total de unidades que se han vendido de
-- cada producto a partir de los datos de la tabla detalle_pedido)
```

```
SELECT DP.CODIGOPRODUCTO, SUM(DP.CANTIDAD)
FROM DETALLEPEDIDOS DP
GROUP BY DP.CODIGOPRODUCTO
HAVING SUM(DP.CANTIDAD) = (SELECT MAX(SUM(DP.CANTIDAD))
                           FROM DETALLEPEDIDOS DP
                           GROUP BY DP.CODIGOPRODUCTO);
```

```
SELECT * FROM (
    SELECT CODIGOPRODUCTO, SUM(CANTIDAD)
    FROM DETALLEPEDIDOS
    GROUP BY CODIGOPRODUCTO
    ORDER BY SUM(CANTIDAD) DESC)
WHERE ROWNUM = 1;
```

EJERCICIOS CON UPDATE

```
--- ACTUALIZAR EN EMPLEADOS LA COMISION AL 15% DEL SUELO A AQUELLOS
EMPLEADOS QUE DESCONOZCO SU COMISION
```

```
UPDATE EMP
SET COMM = SAL * 0.15
WHERE COMM IS NULL;
```

```
ROLLBACK;
```

```
-- ACTUALIZAR EL PRECIO DE VENTA DEL PRODUCTO
-- AL PRECIO MEDIO AL CUAL HE VENDIDO FR-67
```

```
SELECT
AVG(DP.PRECIOUNIDAD)--SUM(DP.CANTIDAD*DP.PRECIOUNIDAD)/SUM(DP.CANTIDA
D)
FROM DETALLEPEDIDOS DP
WHERE UPPER(DP.codigoproducto) = 'FR-67';
```

```
UPDATE PRODUCTOS PR
SET pr.precioventa = (SELECT
SUM(DP.CANTIDAD*DP.PRECIOUNIDAD)/SUM(DP.CANTIDAD)
```

```

FROM DETALLEPEDIDOS DP
WHERE UPPER(DP.codigoproducto) = PR.CODIGOPRODUCTO
)
WHERE PR.CODIGOPRODUCTO IN (SELECT DP.CODIGOPRODUCTO FROM
DETALLEPEDIDOS DP);

```

```

SELECT CODIGOPRODUCTO, NOMBRE, PRECIOVENTA
FROM PRODUCTOS
WHERE CODIGOPRODUCTO NOT IN (SELECT DP.CODIGOPRODUCTO FROM
DETALLEPEDIDOS DP);

```

```

SELECT DP.CODIGOPRODUCTO,AVG(DP.PRECIOUNIDAD) AS RAUL,
SUM(DP.CANTIDAD*DP.PRECIOUNIDAD)/SUM(DP.CANTIDAD) AS ANTONIO
FROM DETALLEPEDIDOS DP
GROUP BY DP.CODIGOPRODUCTO;

```

```

---- Actualizar el limite de crédito a lo que se ha gastado cada cliente.
SELECT NVL(SUM(DP.PRECIOUNIDAD*DP.CANTIDAD),0)
FROM PEDIDOS P, DETALLEPEDIDOS DP
WHERE DP.CODIGOPEDIDO = P.CODIGOPEDIDO
AND P.CODIGOCLIENTE = 2;-- CAMBIANDO EL 1 POR CADA CODIGO DE CLIENTE
TENDRIAMOS LA SOLUCION

```

```

UPDATE CLIENTES C
SET C.LIMITECREDITO = (SELECT NVL(SUM(DP.PRECIOUNIDAD*DP.CANTIDAD),0)
FROM PEDIDOS P, DETALLEPEDIDOS DP
WHERE DP.CODIGOPEDIDO = P.CODIGOPEDIDO
AND P.CODIGOCLIENTE = C.CODIGOCLIENTE);

```

```

/*
3-DEVUELVE UN LISTADO CON EL NOMBRE, APELLIDO Y EMAIL DE LOS
EMPLEADOS

```

```

CUYO JEFE TIENE UN CODIGO DE JEFE IGUAL A 7

```

```

*/
select e.nombre, e.apellido1 || ' ' || e.apellido2 as apellidos, e.email
from empleados e, empleados j
where e.codigoempleado= j.codigoempleado
and j.codigojefe=j.codigojefe
and j.codigojefe in (
select j.codigojefe
from empleados j
where j.codigojefe= 7
);

```

```

/*

```

4-devuelve el nombre del puesto, nombre, apellido, email del jefe de la empresa

*/

```
select nombre, apellido1, apellido2, email
from empleados
where codigojefe is null;
```

/*

5-devuelve un listado con el nombre, apellido y puesto de aquellos empleados que no sean representantes de ventas

*/

```
select nombre, apellido1,apellido2, puesto
from empleados
where upper(puesto) <> 'REPRESENTANTE VENTAS';
```

/*

6-devuelve un listado con el nombre de todos los clientes españoles

*/

```
select nombrecliente, pais
from clientes
where upper(pais)= 'SPAIN';
```

/*

7-devuelve un listado con los distintos estados por lo que puede pasar un pedido

*/

```
select DISTINCT estado
from pedidos;
```

/*

8. Devuelve un listado con el código de cliente de aquellos clientes que realizaron algún pago en 2008. Tenga en cuenta que deberá eliminar aquellos códigos de cliente que aparezcan repetidos. Resuelva la consulta:

*/

```
select DISTINCT c.codigocliente, p.fechapago
from pagos p, clientes c
where c.codigocliente= p.codigocliente
and p.fechapago in (
    select p.fechapago
    from pagos p
    where to_char(p.fechapago,'yyyy')='2008'
```

);

/*

9. Devuelve un listado con el código de pedido, código de cliente, fecha esperada y fecha de

entrega de los pedidos que no han sido entregados a tiempo.

*/

```
select codigopedido, codigocliente, fechaesperada, fechaentrega
from pedidos
where fechaentrega > fechaesperada;
```

/*

10.Devuelve un listado con el código de pedido, código de cliente, fecha esperada y fecha de entrega de los pedidos cuya fecha de entrega ha sido al menos dos días antes de la fecha esperada.

*/

```
select p.codigopedido, c.codigocliente, p.FECHAESPERADA,
p.FECHAENTREGA
from pedidos p, clientes c
where c.codigocliente= p.codigocliente
and p.FECHAENTREGA<=p.FECHAESPERADA - 2;
```

/*

11.Devuelve un listado de todos los pedidos que fueron rechazados en 2009.

*/

```
select *
from pedidos
where upper(estado)= 'RECHAZADO'
and to_char(fechapedido, 'yyyy')= '2009';
```

/*

12.Devuelve un listado de todos los pedidos que han sido entregados en el mes de enero de cualquier año.

*/

```
select *
from pedidos
where to_char(fechaentrega, 'mm')= '01';
```

/*

13.Devuelve un listado con todos los pagos que se realizaron en el año 2008 mediante Paypal. Ordene el resultado de mayor a menor.

*/

```
select fechapago, formapago
from pagos
where to_char(fechapago, 'yyyy')= '2008'
and upper(formapago)= 'PAYPAL'
order by fechapago desc;
```

/*

14.Devuelve un listado con todas las formas de pago que aparecen en la tabla pago. Tenga en cuenta que no deben aparecer formas de pago repetidas.

*/

```
select DISTINCT formapago
from pagos;
```

/*

15.Devuelve un listado con todos los productos que pertenecen a la gama Ornamentales y que tienen más de 100 unidades en stock. El listado deberá estar ordenado por su precio de venta, mostrando en primer lugar los de mayor precio.

*/

```
select precioventa, gama, cantidadenstock
from productos
where upper(gama)= 'ORNAMENTALES'
and cantidadenstock> 100
order by precioventa desc;
```

/*

16.Devuelve un listado con todos los clientes que sean de la ciudad de Madrid y cuyo representante de ventas tenga el código de empleado 11 o 30.

*/

```
select *
from clientes
where upper(ciudad)= 'MADRID'
and codigoempleadorepventas= 11
or codigoempleadorepventas= 30;
```

/*

17.Obtén un listado con el nombre de cada cliente y el nombre y apellido de su representante de ventas.

*/

```
select c.nombrecliente, e.nombre, e.apellido1 || ' ' || e.apellido2 as apellidos_repre_ventas
from clientes c, empleados e
where e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS;
```

/*

18.Muestra el nombre de los clientes que hayan realizado pagos junto con el nombre de sus representantes de ventas.

*/

```
select p.idtransaccion, c.nombrecliente, e.nombre
from clientes c, pagos p, empleados e
where e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS
```

```
and p.codigocliente= c.codigocliente;
```

```
/*
```

```
19.Muestra el nombre de los clientes que no hayan  
realizado pagos junto con el  
nombre de sus representantes de ventas.
```

```
*/
```

```
select c.nombrecliente,p.codigocliente, e.nombre as repre_ventas  
from clientes c, empleados e, pagos p  
where c.CODIGOEMPLEADOREPVENTAS= e.CODIGOEMPLEADO  
and c.codigocliente= p.codigocliente(+)  
and p.codigocliente is null;
```

```
/*
```

```
--20.Devuelve el nombre de los clientes que han hecho pagos  
y el nombre de sus representantes junto con la ciudad de  
la oficina a la que pertenece el representante.
```

```
*/
```

```
select count(*), c.nombrecliente ,e.nombre, o.ciudad  
from pagos p, clientes c, empleados e, oficinas o  
where c.codigocliente= p.codigocliente  
and e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS  
and o.codigooficina= e.codigooficina  
group by p.codigocliente, c.nombrecliente,  
e.nombre, o.ciudad;
```

```
/*
```

```
--22.Lista la dirección de las oficinas que tengan clientes  
en Fuenlabrada.
```

```
*/
```

```
select c.codigocliente, c.lineadireccion2  
from oficinas o, empleados e, clientes c  
where o.codigooficina= e.codigooficina  
and e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS  
and c.lineadireccion2 in ('FUENLABRADA');
```

```
/*
```

```
--23.Devuelve el nombre de los clientes y el nombre de sus  
representantes junto con la ciudad de la oficina a la que pertenece  
el representante.
```

```
*/
```

```
select c.nombrecliente, e.nombre, o.ciudad  
from oficinas o, empleados e, clientes c  
where o.codigooficina= e.codigooficina  
and e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS;
```

```
/*
```

--24.Devuelve un listado con el nombre de los empleados
junto con el nombre de sus jefes.

*/

```
select count(*), subor.nombre as empleado,jefe.nombre as jefe
from empleados subor, empleados jefe
where subor.codigoempleado= jefe.codigojefe
group by subor.codigoempleado, jefe.codigojefe, subor.nombre, jefe.nombre ;
```

/*

--25.Devuelve un listado que muestre el nombre de cada empleado,
el nombre de su jefe y el nombre del jefe de su jefe.

*/

```
select subor.nombre as empleado, jefe.nombre as jefe,
jefe_jefe.nombre as jefe_del_jefe
from empleados subor, empleados jefe, empleados jefe_jefe
where subor.codigojefe= jefe.codigoempleado
and jefe.codigojefe= jefe_jefe.codigoempleado;
```

/*

--26.Devuelve el nombre de los clientes a los que no se
les ha entregado a tiempo un pedido.

*/

```
select count(*), c.nombrecliente,p.fechaesperada, p.fechaentrega
from pedidos p, clientes c
where c.codigocliente= p.codigocliente
and p.fechaentrega>p.fechaesperada
group by c.codigocliente,c.nombrecliente,
p.fechaesperada, p.fechaentrega;
```

/*

--27.Devuelve un listado de las diferentes gamas de producto
que ha comprado cada cliente.

*/

```
select c.codigocliente, gp.gama
from clientes c, pedidos p,
detallepedidos dp, productos pro, gamasproductos gp
where c.codigocliente= p.codigocliente
and p.codigopedido= dp.codigopedido
and pro.codigoproducto= dp.codigoproducto
and gp.gama= pro.gama
group by c.codigocliente,gp.gama
order by c.codigocliente;
```

/*

--28.Devuelve un listado que muestre solamente los clientes
que no han realizado ningún pago.

*/

```

select c.nombrecliente, c.codigocliente, p.IDTRANSACCION
from clientes c, pagos p
where p.codigocliente(+) = c.codigocliente
and p.IDTRANSACCION is null
group by c.nombrecliente, c.codigocliente, p.IDTRANSACCION;

```

/*

--29.Devuelve un listado que muestre solamente los
clientes que no han realizado ningún pedido.

*/

```

select c.codigocliente, c.nombrecliente, p.codigopedido
from pedidos p, clientes c
where c.codigocliente = p.codigocliente(+)
and p.codigocliente is null
group by c.codigocliente, c.nombrecliente, p.codigopedido;

```

/*

--30.Devuelve un listado que muestre los clientes que no
han realizado ningún pago y los que no han realizado ningún pedido.

*/

```

select c.codigocliente, p.codigopedido, pa.IDTRANSACCION
from pedidos p, clientes c, pagos pa
where c.codigocliente = p.codigocliente(+)
and c.codigocliente = pa.codigocliente(+)
and p.codigocliente is null
and pa.codigocliente is null
;

```

/*

--31.Devuelve un listado que muestre solamente los empleados
que no tienen una oficina asociada.

*/

--FALTA RESOLVER

```

select * --e.codigoempleado, o.codigooficina
from empleados e, oficinas o
where e.codigooficina = o.codigooficina(+)
and e.codigooficina is null;

```

```

select e.codigoempleado, o.codigooficina
from empleados e
left outer join oficinas o
on e.codigooficina = o.codigooficina
where e.codigooficina is null;

```

/*

--32.Devuelve un listado que muestre solamente los empleados
que no tienen un cliente asociado.

```

*/
select e.codigoempleado, c.codigocliente
from empleados e, clientes c
where e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS(+)
and c.codigocliente is null;

```

```

/*
--33.Devuelve un listado que muestre solamente los empleados que
no tienen un cliente asociado junto con los datos de la oficina
donde trabajan.

```

```

*/
select e.codigoempleado, c.codigocliente, o.*
from empleados e, clientes c, oficinas o
where e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS(+)
and o.codigooficina= e.codigooficina
and c.codigocliente is null;

```

```

/*
--34.Devuelve un listado que muestre los empleados
que no tienen una oficina asociada y los que no
tienen un cliente asociado.

```

```

*/
--NO DEVUELVE DATOS
select e.codigoempleado, c.codigocliente,o.codigooficina
from empleados e, clientes c, oficinas o
where e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS(+)
and e.codigooficina= o.codigooficina(+)
and c.codigocliente is null
and o.codigooficina is null;

```

```

select *
from empleados e
where e.codigooficina not in (
        select o.codigooficina
        from oficinas o)
and e.codigoempleado not in (
        select c.CODIGOEMPLEADOREPVENTAS
        from clientes c);

```

```

select o.codigooficina
from oficinas o;

```

```

/*
--35.Devuelve un listado de los productos que nunca han aparecido
en un pedido.

```

```

*/
select p.codigoproducto, dp.codigopedido

```

```
from productos p, detallepedidos dp
where p.codigoproducto= dp.codigoproducto(+)
and dp.codigopedido is null;
```

/*

--36.Devuelve un listado de los productos que nunca han aparecido en un pedido. El resultado debe mostrar el nombre, la descripción y la imagen del producto.

*/

```
select p.codigoproducto, dp.codigopedido, p.nombre, p.descripcion,
nvl(gp.imagen, 'NO TIENE IMAGEN') as imagen
from productos p, detallepedidos dp, gamasproductos gp
where p.codigoproducto= dp.codigoproducto(+)
and gp.gama= p.gama
and dp.codigopedido is null;
```

/*

--37.Devuelve las oficinas donde no trabajan ninguno de los empleados que hayan sido los representantes de ventas de algún cliente que haya realizado la compra de algún producto de la gama Frutales.

*/

```
select *
from oficinas o, empleados e, clientes c
where o.codigooficina(+)= e.codigooficina
and c.CODIGOEMPLEADOREPVENTAS= e.codigoempleado;
```

```
select * from calendario;
```

/*

--39.Devuelve un listado con los datos de los empleados que no tienen clientes asociados y el nombre de su jefe asociado.

*/

```
select subor.codigoempleado, subor.nombre, subor.codigojefe, jefe.nombre
from empleados subor, clientes c, empleados jefe
where subor.CODIGOEMPLEADO= c.CODIGOEMPLEADOREPVENTAS(+)
and subor.codigojefe= jefe.codigoempleado
and c.codigocliente is null;
```

/*

--40.¿Cuántos empleados hay en la compañía?

*/

```
select count(*)
from empleados;
```

/*

--41.¿Cuántos clientes tiene cada país?

*/

```
select count(pais)
```

```
from clientes;
```

```
/*
```

```
42.¿Cuál fue el pago medio en 2009?
```

```
*/
```

```
select avg(cantidad) as pago_medio  
from pagos  
where to_char(fechapago, 'yyyy')='2009';
```

```
select *
```

```
from pagos;
```

```
/*
```

```
43.¿Cuántos pedidos hay en cada estado? Ordena el resultado de forma  
descendente por el número de pedidos.
```

```
*/
```

```
select estado, count(codigopedido) as total_pedidos  
from pedidos  
group by estado  
order by count(codigopedido) desc;
```

```
/*
```

```
44.Calcula el precio de venta del producto más caro y más barato en una  
misma consulta.
```

```
*/
```

```
select max(precioventa) as caro, min(precioventa) as barato  
from productos;
```

```
/*45.Calcula el número de clientes que tiene la empresa.*/
```

```
select count(*)  
from clientes;
```

```
/*
```

```
46.¿Cuántos clientes existen con domicilio en la ciudad de Madrid?
```

```
*/
```

```
select COUNT(ciudad)  
from clientes  
where upper(ciudad)= 'MADRID';
```

```
/*
```

```
47.¿Calcula cuántos clientes tiene cada una de las ciudades  
que empiezan por M?
```

```
*/
```

```
select COUNT(CIUDAD)  
from clientes  
where upper(ciudad) like 'M%';
```

```
/*
```

```
48.Devuelve el nombre de los representantes de ventas
```

y el número de clientes al que atiende cada uno.

*/

```
select e.nombre, c.codigocliente--count(c.CODIGOCLIENTE)
from empleados e, clientes c
where e.codigoempleado= c.CODIGOEMPLEADOREPVENTAS;
--group by e.nombre;
```

/*49. Calcula el número de clientes que no tiene asignado representante de ventas.*/

```
select count(*)
from clientes c
where c.CODIGOEMPLEADOREPVENTAS is null;
```

/*

50. Calcula la fecha del primer y último pago realizado por cada uno de los clientes. El listado deberá mostrar el nombre y los apellidos de cada cliente.

*/

```
select p.fechapago, c.nombrecliente
from clientes c, pagos p
where p.codigocliente= c.codigocliente
and p.fechapago in (
    (select min(p.fechapago)
    from pagos p),

    (select max(p.fechapago)
    from pagos p)
);
```

/*

51. Calcula el número de productos diferentes que hay en cada uno de los pedidos.

*/

```
select *
from detallepedidos;
```

/*

52. Calcula la suma de la cantidad total de todos los productos que aparecen en cada uno de los pedidos.

*/

```
select p.codigopedido, sum(dp.cantidad), count(dp.codigopedido)
from detallepedidos dp, pedidos p
where p.codigopedido= dp.codigopedido
group by dp.codigopedido
order by p.codigopedido;
```


/*

53.Devuelve un listado de los 20 productos más vendidos y el número total de unidades que se han vendido de cada uno. El listado deberá estar ordenado por el número total de unidades vendidas.

*/

```
select rownum, nombre, suma, total
from (select p.nombre as nombre, sum(dp.cantidad) suma,
count(p.nombre) as total
      from pedidos pe, detallepedidos dp, productos p
      where pe.codigopedido= dp.codigopedido
      and p.codigoproducto= dp.codigoproducto
      group by p.nombre
      order by sum(dp.cantidad) desc)
where rownum <=20;
```

/*

54.La facturación que ha tenido la empresa en toda la historia, indicando la base imponible, el IVA y el total facturado. La base imponible se calcula sumando el coste del producto por el número de unidades vendidas de la tabla detalle pedido. El IVA es el 21 % de la base imponible, y el total la suma de los dos campos anteriores.

*/

```
SELECT C.NOMBRECLIENTE, SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS
BASE_IMPONIBLE,
      SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) * 0.21 AS IVA,
      SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) * 1.21 AS TOTAL_FACTURADO
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND DP.CODIGOPEDIDO = P.CODIGOPEDIDO
GROUP BY C.NOMBRECLIENTE;
```

/*

55.La misma información que en la pregunta anterior, pero agrupada por código de producto.

*/

```
SELECT PRO.CODIGOPRODUCTO, SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS
BASE_IMPONIBLE,
      SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) * 0.21 AS IVA,
      SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) * 1.21 AS TOTAL_FACTURADO
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP, PRODUCTOS PRO
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND DP.CODIGOPEDIDO = P.CODIGOPEDIDO
AND PRO.CODIGOPRODUCTO= DP.CODIGOPRODUCTO
GROUP BY PRO.CODIGOPRODUCTO;
```

/*

56.La misma información que en la pregunta anterior, pero agrupada por código de producto filtrada por los códigos que empiecen por OR.

*/

```
SELECT PRO.CODIGOPRODUCTO, SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS  
BASE_IMPONIBLE,  
SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) * 0.21 AS IVA,  
SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) * 1.21 AS TOTAL_FACTURADO  
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP, PRODUCTOS PRO  
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE  
AND DP.CODIGOPEDIDO = P.CODIGOPEDIDO  
AND PRO.CODIGOPRODUCTO= DP.CODIGOPRODUCTO  
AND UPPER(PRO.CODIGOPRODUCTO) LIKE 'OR%'  
GROUP BY PRO.CODIGOPRODUCTO;
```

--58. Muestre la suma total de todos los pagos que se realizaron

--para cada uno de los años que aparecen en la tabla pagos.

```
select sum(cantidad) total, to_char(fechapago, 'yyyy')
```

```
from pagos
```

```
group by to_char(fechapago, 'yyyy')
```

```
order by to_char(fechapago, 'yyyy');
```

-- OBTENER LA DESCRIPCIÓN TEXTUAL DE LAS GAMAS DE PRODUCTOS QUE HAN SIDO VENDIDOS POR EMPLEADOS QUE TIENE OFICINA EN BARCELONA

```
SELECT DISTINCT g.descripciontexto  
FROM OFICINAS O, EMPLEADOS E, CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP,  
PRODUCTOS PR, GAMASPRODUCTOS G  
WHERE O.CODIGOOFICINA = E.CODIGOOFICINA  
AND E.CODIGOEMPLEADO = C.CODIGOEMPLEADOREPVENTAS  
AND C.CODIGOCLIENTE = P.CODIGOCLIENTE  
AND P.CODIGOPEDIDO =DP.CODIGOPEDIDO  
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO  
AND PR.GAMA = G.GAMA  
AND UPPER(O.CIUDAD) = 'BARCELONA';
```

-- MOSTRAR EL NOMBRE DEL CLIENTE Y CUANTO SE HA GASTADO

-- GASTO DE CADA CLIENTE

```
SELECT C.NOMBRECLIENTE, SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) AS GASTADO  
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP  
WHERE C.CODIGOCLIENTE =P.CODIGOCLIENTE  
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO  
GROUP BY C.NOMBRECLIENTE;
```

--MOSTRAR SOLO AQUELLOS QUE SE HAN GASTADO MAS DE 50000

```
SELECT C.NOMBRECLIENTE, SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) AS GASTADO  
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP  
WHERE C.CODIGOCLIENTE =P.CODIGOCLIENTE
```

```
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
GROUP BY C.NOMBRECLIENTE
HAVING SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) >=1000;
```

```
-- DE TODOS LOS ANTERIORES SACAR EL QUE MAS SE HA GASTADO
-- SI EL VALOR MAYOR ES UNICO PUEDO UTILIZAR EL ROWNUM
SELECT *
FROM ( SELECT C.NOMBRECLIENTE, SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) AS
GASTADO
      FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP
      WHERE C.CODIGOCLIENTE =P.CODIGOCLIENTE
      AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
      GROUP BY C.NOMBRECLIENTE
      HAVING SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) >=1000
      ORDER BY 2 DESC)
WHERE ROWNUM =1;
```

```
--PERO SI NO ES UNICO--- MEJOR FORMA--PUES LOS DATOS PUEDEN CAMBIAR.
VISTA---(NOMBRECLIENTE, SALARIO)
SELECT C.NOMBRECLIENTE, SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) AS GASTADO
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP
WHERE C.CODIGOCLIENTE =P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
GROUP BY C.NOMBRECLIENTE
HAVING SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) >=1000;
```

```
--QUE TIO ES EL QUE MAS COBRA
SELECT MAX(SALARIO) FROM VISTA;
```

```
SELECT *
FROM VISTA
WHERE SALARIO = (SELECT MAX(SALARIO) FROM VISTA);
```

```
-- SUSTITUIMOS
SELECT C.NOMBRECLIENTE, SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) AS GASTADO
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP
WHERE C.CODIGOCLIENTE =P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
GROUP BY C.NOMBRECLIENTE
HAVING SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) = (SELECT MAX(SUM(DP.CANTIDAD
*DP.PRECIOUNIDAD))
      FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP
      WHERE C.CODIGOCLIENTE =P.CODIGOCLIENTE
      AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
      GROUP BY C.NOMBRECLIENTE
      );
```

```
-- CUANTO HA PAGADO CADA CLIENTE MOSTRAR EL NOMBRE, LA CIUDAD Y LO
QUE HA PAGADO,
-- MOSTRAR SOLO LOS QUE HA PAGADO MAS DE 5000 Y QUE VIVAN EN
BARCELONA, MADRID O GETAFE
-- Y MUESTRALOS ORDENADOS POR NOMBRE DE CIUDAD ALFABETICAMENTE
SELECT C.NOMBRECLIENTE, C.CIUDAD, SUM(PA.CANTIDAD)
FROM CLIENTES C, PAGOS PA
WHERE C.CODIGOCLIENTE = PA.CODIGOCLIENTE
AND UPPER(C.CIUDAD) IN ('BARCELONA','MADRID','GETAFE')
GROUP BY C.NOMBRECLIENTE, C.CIUDAD
HAVING SUM(PA.CANTIDAD) >= 5000
ORDER BY 2;
```

```
SELECT C.NOMBRECLIENTE, C.CIUDAD, SUM(PA.CANTIDAD)
FROM CLIENTES C, PAGOS PA
WHERE C.CODIGOCLIENTE = PA.CODIGOCLIENTE;
```

```
--DIME QUIENES SON LOS CLIENTES QUE VIVEN EN LA MISMA CIUDAD QUE
GOLDFISH GARDEN
SELECT *
FROM CLIENTES
WHERE CLIENTES.CIUDAD =V_CIDUDAD_GFG;
```

```
-- QUE ES EL ROWNUM, NO ES MAS QUE MOSTRA UN CONTADOR DE FILAS QUE
SALEN
SELECT C.NOMBRECLIENTE, C.CIUDAD
FROM CLIENTES C
WHERE UPPER(C.NOMBRECLIENTE) LIKE 'T%'
AND ROWNUM <=3;
```

```
SELECT * FROM (
SELECT C.NOMBRECLIENTE, SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) AS GASTADO
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP
WHERE C.CODIGOCLIENTE =P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
GROUP BY C.NOMBRECLIENTE
HAVING SUM(DP.CANTIDAD *DP.PRECIOUNIDAD) >=1000
ORDER BY 2 DESC
)
WHERE ROWNUM =1;
```

```
-- RECORDEMOS EL DECODE
SELECT E.ENAME, DECODE(E.SAL, 5000, 'SUPERSUELDO', 3000, 'PEDAZO
SUELDO','OTRO SUELDO')
FROM EMP E;
```

```
-- CATEGORIZAMOS EN BASE A SUELDO
-- SI COBRA MAS DE 3000--> SUELDO DE RICO
-- SI COBRA MÁS DE 2000 Y MENOS DE 3000-- SUELDO BUENO
-- SI COBRA ENTRE 1000 Y 2000 -- SUELDO MEDIO
-- MENOS DE MIL---SUELDO BASURA
```

```
SELECT E.ENAME, SAL,
       CASE
         --WHEN SAL >=10 THEN 'VENDO DROGA'
         WHEN SAL >=3000 AND UPPER(E.JOB) ='PRESIDENT' THEN 'QUE NO TE LO
DIGO'
         WHEN SAL >=3000 THEN 'SUELDO DE RICO'
         WHEN SAL <3000 AND SAL >=2000 THEN 'SUELDO BUENO'
         WHEN SAL <2000 AND SAL >=1000 THEN 'SUELDO MEDIO'
         ELSE 'SUELDO BASURA'
       END AS TIPIFICACION_SUELDO
FROM EMP E;
```

```
/*
3-mostrar el nombre y cargo de los empleados que no
sean directores de oficina
*/
select * from empleados;
```

```
select nombre, puesto
from empleados
where puesto in (select puesto
                 from empleados
                 where lower(puesto) <> 'director oficina');
```

```
select nombre, puesto
from empleados
where lower(puesto) <> 'director oficina';
```

```
/*
4-muestra el numero de empleados que hay en la empresa
*/
select COUNT(codigoempleado) total_empleados
from empleados;
```

```
/*
5-muestra el numero de clientes norteamericanos
*/
select * from clientes;
```

```
select COUNT(pais) as clientes_norteamericanos
from clientes
where upper(pais) = 'USA';
```

```
/*
6-NUMERO DE CLIENTES DE CADA PAIS
*/
select * from clientes;
```

```
select pais, COUNT(pais)
from clientes
group by pais;
```

```
/*
7- Muestra el nombre del cliente y el nombre de su
representante de ventas (si lo tiene).
*/
```

```
select c.nombrecliente as "Nombre cliente",
e.nombre as "Nombre representante"
from clientes c, empleados e
where c.codigoempleadorepventas=e.CODIGOEMPLEADO;
```

```
/*
8. Nombre de los clientes que hayan hecho un pago en 2007
*/
```

```
select distinct c.nombrecliente
from clientes c, pagos p
where c.codigocliente= p.codigocliente
and p.fechapago > to_date('01/01/2007', 'dd/mm/yyyy')
and p.fechapago < to_date('31/12/2007', 'dd/mm/yyyy');
```

```
/*
los posibles estados de un pedido
*/
select DISTINCT c.nombrecliente, p.estado
from pedidos p, clientes c
where p.codigopedido= c.codigocliente;
```

```
/*
10. Muestra el número de pedido,
el nombre del cliente, la fecha de entrega y
la fecha requerida de los pedidos que no
```

han sido entregados a tiempo.

*/

```
select p.codigopedido, c.nombrecliente, p.fechaentrega, p.fechaesperada
from clientes c, pedidos p
where c.codigocliente = p.CODIGOCLIENTE
and p.FECHAESPERADA < p.fechaentrega;
```

/*

13-muestra el codigo y la cantidad de veces que se ha pedido
un producto al menos una vez

*/

```
select p.codigoproducto, dp.cantidad as cantidad_pedida
from productos p, detallepedidos dp
where p.codigoproducto= dp.codigoproducto;
group by p.codigoproducto;
```

si entendemos por deuda lo que me he gastado menos lo que he pagado, calcular la deuda
de cada cliente mostrando su nombre y la deuda que tiene, ordenar descendientemente
por deuda

```
select gastado.codigocliente, gastado.gastado - pagado.pagado from (select
pa.codigocliente, sum(pa.cantidad) as pagado from pagos pa
                                group by pa.codigocliente) pagado,

                                (select
pa.codigocliente,sum(dp.preciounidad*dp.cantidad) as gastado from detallepedidos dp,
pedidos p, clientes c, pagos pa
                                where dp.codigopedido = p.codigopedido
                                and p.codigocliente = c.codigocliente
                                and c.codigocliente = pa.codigocliente
                                group by pa.codigocliente
                                order by 1 asc) gastado

where gastado.codigocliente = pagado.codigocliente
order by 2 desc;
```

/* JARDINERIA

mostrar el codigo de los pedidos donde se haya vendido el
producto de la gama 'aromaticas' mas caro

*/

```
select dp.codigopedido
from pedidos p, detallepedidos dp
where p.codigopedido= dp.codigopedido
and dp.codigoproducto in (select dp.codigoproducto
                        from detallepedidos dp
                        where dp.preciounidad = (select max(dp.preciounidad)
                                                from productos pro, detallepedidos dp
                                                where dp.codigoproducto = pro.codigoproducto
```

```
and upper(pro.gama)= 'AROMÁTICAS')  
group by dp.codigopedido;
```