-- \*\*\* BBDD 09\_jardineria \*\*\*

use 10\_jardineria;

**-- 1 Devuelve el nombre de los clientes que no hayan hecho pagos y el nombre de sus representantes junto con la ciudad de la oficina a la que pertenece el representante.**

select cl.nombre\_cliente,concat(emp.apellido1,' ', emp.apellido2,', ', emp.nombre) as empleado\_a\_cargo, ofi.ciudad as ciudad\_oficina

from clientes as cl

left join pagos pg on pg.fk\_cliente = cl.id\_cliente

join empleados as emp on cl.fk\_empleado\_rep\_ventas = emp.id\_empleado

join oficinas as ofi on emp.fk\_oficina = ofi.id\_oficina

where pg.fk\_cliente is null;

select distinct cl.id\_cliente, cl.nombre\_cliente, o.ciudad

from clientes cl

left join pagos pg on cl.id\_cliente = pg.fk\_cliente

join empleados em on cl.fk\_empleado\_rep\_ventas = em.id\_empleado

join oficinas o on em.fk\_oficina = o.id\_oficina

where pg.fk\_cliente is null;

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

select distinct(cl.id\_cliente)

from clientes as cl

left join pagos pg on pg.fk\_cliente = cl.id\_cliente

left join pedidos as pd on cl.id\_cliente = pd.fk\_cliente

where pg.id\_transaccion is null and pd.id\_pedido is null;

select distinct c.\*

from clientes c

left join pagos p on c.id\_cliente = p.fk\_cliente

left join pedidos ped on c.id\_cliente = ped.fk\_cliente

where p.id\_transaccion is null and ped.id\_pedido is null;

select c.\*

from clientes c

where c.id\_cliente not in (select distinct fk\_cliente from pagos) and

c.id\_cliente not in (select distinct fk\_cliente from pedidos);

**-- 3 Devuelve un listado que muestre solamente los clientes que sí han realizado algún pago.**

select distinct(cl.id\_cliente), cl.nombre\_cliente

from clientes as cl

join pagos as pg on cl.id\_cliente = pg.fk\_cliente;

select distinct c.\*

from clientes c

join pagos p on c.id\_cliente = p.fk\_cliente;

**-- 4 Calcula el número de clientes que tiene la empresa.**

select count(id\_cliente)

from clientes;

select count(\*) cant\_clientes

from clientes;

**-- 5 Devuelve el nombre del producto que tenga el precio de venta más caro.**

select prod.nombre, prod.precio\_venta

from productos as prod

where prod.precio\_venta = (select max(precio\_venta) from productos);

-- Mayor precio

select max(precio\_venta) from productos;

select p.\*

from productos p

where p.precio\_venta = (select max(precio\_venta) from productos);

**-- 6 Devuelve un listado indicando todas las ciudades donde hay oficinas y el número de empleados que tiene.**

select distinct(ciudad)

from oficinas;

oficinas y el número de empleados que tiene.

select o.ciudad, count(e.id\_empleado) cant

from oficinas o

join empleados e on o.id\_oficina = e.fk\_oficina

group by o.ciudad;

**-- 7 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 pd.id\_pedido, cl.id\_cliente, pd.fecha\_esperada, pd.fecha\_entrega

from pedidos as pd

join clientes as cl on pd.fk\_cliente = cl.id\_cliente

where pd.fecha\_entrega > pd.fecha\_esperada;

select p.id\_pedido, p.fecha\_esperada, p.fecha\_entrega, p.fk\_cliente

from pedidos p

where timestampdiff(day, p.fecha\_esperada, p.fecha\_entrega) > 0;

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

select prod.nombre, dp.fk\_pedido

from productos as prod

left join detalles\_pedido as dp on prod.id\_producto = dp.fk\_producto

where dp.fk\_pedido is null;

select p.\*

from productos p

left join detalles\_pedido dp on p.id\_producto = dp.fk\_producto

where dp.fk\_producto is null;

select \*

from productos p

where p.id\_producto not in (select fk\_producto from detalles\_pedido);

select \*

from productos p

where not exists (select \* from detalles\_pedido where fk\_producto = p.id\_producto);

**-- 9 Calcula el número de clientes que no tiene asignado representante de ventas.**

select count(cl.id\_cliente)

from clientes as cl

where cl.fk\_empleado\_rep\_ventas is null;

select count(\*)

from clientes

where fk\_empleado\_rep\_ventas is null;

**-- 10 Devuelve un listado con el nombre, apellidos y puesto de aquellos empleados que no sean representantes de ventas.**

select concat(emp.nombre,' ',emp.apellido1,', ',emp.apellido2) as n\_empleado, puesto

from empleados as emp

left join clientes as cl on emp.id\_empleado = cl.fk\_empleado\_rep\_ventas

where fk\_empleado\_rep\_ventas is null;

select \*

from empleados

where not exists (select \* from clientes where fk\_empleado\_rep\_ventas = id\_empleado);

**-- 11 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 id\_cliente,ciudad,fk\_empleado\_rep\_ventas

from clientes

where ciudad='Madrid' and (fk\_empleado\_rep\_ventas = 11 or fk\_empleado\_rep\_ventas = 30);

select \*

from clientes

where ciudad = 'Madrid' and fk\_empleado\_rep\_ventas in (11, 30);

**-- 12 Devuelve el nombre del puesto, nombre, apellidos y email del jefe de la empresa.**

select puesto, concat(nombre,' ', apellido1,' ', apellido2) as n\_jefe

from empleados

where puesto ='Director General';

select \*

from empleados

where fk\_jefe is null;

**-- 13 Devuelve un listado con los clientes que han realizado algún pedido pero no han realizado ningún pago, utilizando una subconsulta**

select id\_cliente

from clientes

where id\_cliente not in (select fk\_cliente from pagos) and id\_cliente in (select fk\_cliente from pedidos);

select distinct c.\*

from clientes c

join pedidos p on c.id\_cliente = p.fk\_cliente

where c.id\_cliente not in (select fk\_cliente from pagos);

-- sin subconsulta

select distinct c.\*

from clientes c

join pedidos p on c.id\_cliente = p.fk\_cliente

left join pagos pg on c.id\_cliente = pg.fk\_cliente

where pg.id\_transaccion is null;

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

select \*

from pagos

where year(fecha\_pago) = 2008 and forma\_pago = 'PayPal'

order by total desc;

select \*

from pagos

where year(fecha\_pago) = 2008 and forma\_pago like 'paypal';

**-- 15 Devuelve el nombre, apellidos, puesto y teléfono de la oficina de aquellos empleados que no sean representante de ventas de ningún cliente, utilizando una subconsulta**

select concat(emp.nombre,' ',emp.apellido1,' ', emp.apellido2) as n\_empleado, emp.puesto, ofi.telefono, emp.extension-- falta telefono

from empleados as emp

join oficinas as ofi on fk\_oficina = ofi.id\_oficina

where id\_empleado not in (select fk\_empleado\_rep\_ventas from clientes);

select e.\*, o.\*

from empleados e

join oficinas o on e.fk\_oficina = o.id\_oficina

where e.id\_empleado not in (select fk\_empleado\_rep\_ventas from clientes);

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

select cl.nombre\_cliente, emp.nombre as nombre\_representante, pg.id\_transaccion

from clientes as cl

join empleados as emp on cl.fk\_empleado\_rep\_ventas = emp.id\_empleado

join pagos as pg on cl.id\_cliente = pg.fk\_cliente

where pg.fk\_cliente is not null;

select distinct c.\*, e.\*

from clientes c

join pagos p on c.id\_cliente = p.fk\_cliente

join empleados e on c.fk\_empleado\_rep\_ventas = e.id\_empleado;

select \*

from clientes c

join empleados e on c.fk\_empleado\_rep\_ventas = e.id\_empleado

where c.id\_cliente in (select fk\_cliente from pagos);

**-- 17 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)**

-- MAL, faltaria ver la suma

select prod.id\_producto, prod.nombre, dp.cantidad

from productos as prod

join detalles\_pedido as dp on prod.id\_producto = dp.fk\_producto

where cantidad >=all (select cantidad from detalles\_pedido);

-- BIEN

-- cantidad vendida de cada producto

select sum(cantidad) cant

from detalles\_pedido

group by fk\_producto;

-- cantidad vendida maxima

select max(cant) cant\_max

from (select sum(cantidad) cant

from detalles\_pedido

group by fk\_producto) maxima;

select p.\*, sum(cantidad) cant

from productos p

join detalles\_pedido on p.id\_producto = fk\_producto

group by p.id\_producto

having cant = (select max(cant) cant\_max

from (select sum(cantidad) cant

from detalles\_pedido

group by fk\_producto) maxima);

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

select \*

from empleados as emp

where emp.fk\_oficina is null and id\_empleado not in (select fk\_empleado\_rep\_ventas from clientes);

select \*

from empleados

where fk\_oficina is not null

and id\_empleado not in (select fk\_empleado\_rep\_ventas from clientes);

**-- 20 ¿Cuántos clientes existen con domicilio en la ciudad de Madrid?**

select count(id\_cliente)

from clientes

where ciudad = 'Madrid';

select count(\*) cant

from clientes

where ciudad = 'Madrid';

**-- 21 Devuelve un listado con el código de oficina y la ciudad donde hay oficinas**.

select id\_oficina, codigo\_postal,ciudad

from oficinas;

select \*

from oficinas;

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

select \*

from empleados e

left join empleados j on e.fk\_jefe = j.id\_empleado

where e.id\_empleado not in (select fk\_empleado\_rep\_ventas from clientes);

select \*

from empleados e

left join empleados j on e.fk\_jefe = j.id\_empleado

where e.id\_empleado not in (select fk\_empleado\_rep\_ventas from clientes);

**-- 23 Devuelve el producto que más unidades tiene en stock.**

select \*

from productos

where cantidad\_en\_stock = (select max(cantidad\_en\_stock) cant\_max from productos);

-- cantidad maxima en stock

select max(cantidad\_en\_stock) cant\_max from productos;

select \*

from productos

where cantidad\_en\_stock = (select max(cantidad\_en\_stock) cant\_max from productos);

**-- 24 Muestra el nombre de los clientes que no hayan realizado pagos junto con el nombre de sus representantes de ventas.**

select cl.nombre\_cliente as n\_cliente, emp.nombre as n\_empleado

from clientes as cl

join empleados as emp on cl.fk\_empleado\_rep\_ventas = emp.id\_empleado

left join pagos as pg on cl.id\_cliente = pg.fk\_cliente

where pg.fk\_cliente is null;

select cli.id\_cliente, cli.nombre\_cliente, e.id\_empleado, e.nombre nombre\_representante

from (select \* from clientes c

left join pagos pg on c.id\_cliente = pg.fk\_cliente

where pg.fk\_cliente is null) cli

join empleados e on cli.fk\_empleado\_rep\_ventas = e.id\_empleado;

**-- 25 Devuelve un listado con la ciudad y el teléfono de las oficinas de España.**

select ciudad, telefono

from oficinas

where pais = 'España';

select \* from oficinas;

**-- 26 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.

-- base imponible

select cantidad \* precio\_unidad as base\_imponible

from detalles\_pedido;

-- IVA

select sum(cantidad \* precio\_unidad) as base\_imponible, round(sum(cantidad \* precio\_unidad )\* 0.21,2) as IVA

from detalles\_pedido;

-- 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 sum(cantidad \* precio\_unidad) base\_imponible, round(sum(cantidad \* precio\_unidad) \* 0.21, 2) iva, sum(cantidad \* precio\_unidad) + round(sum(cantidad \* precio\_unidad) \* 0.21, 2) total

from detalles\_pedido;

**-- 27 Devuelve un listado con el nombre, apellidos y email de los empleados cuyo jefe tiene un código de jefe igual a 7.**

select nombre, apellido1, apellido2, email, fk\_jefe

from empleados

where fk\_jefe = 7 ;

select \*

from empleados

where fk\_jefe = 7;

**-- 28 Lista la dirección de las oficinas que tengan clientes en Fuenlabrada.**

select ofi.linea\_direccion1, ofi.linea\_direccion2, cl.region

from oficinas as ofi

join empleados as emp on fk\_oficina = ofi.id\_oficina

join clientes as cl on fk\_empleado\_rep\_ventas = emp.id\_empleado

where cl.region = 'Fuenlabrada';

select distinct \*

from oficinas o

join empleados e on o.id\_oficina = e.fk\_oficina

join (select fk\_empleado\_rep\_ventas fk\_emp from clientes where ciudad = 'Fuenlabrada') cli on e.id\_empleado = cli.fk\_emp;

**-- 29 Devuelve un listado de todos los pedidos que fueron rechazados en 2009.**

select \*

from pedidos

where estado = 'Rechazado' and year(fecha\_entrega)= 2009;

select \*

from pedidos

where estado = 'Rechazado' and year(fecha\_pedido) = 2009;

**-- 30 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(forma\_pago)

from pagos;

select distinct forma\_pago

from pagos;

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

select \*

from clientes as cl

left join pagos as pg on cl.id\_cliente = pg.fk\_cliente

where pg.fk\_cliente is null;

select c.\*

from clientes c

where c.id\_cliente != all (select fk\_cliente from pagos);

select c.\*

from clientes c

where not exists (select \* from pagos where fk\_cliente = c.id\_cliente);

**-- 32 Devuelve un listado con los clientes que han realizado algún pedido pero no han realizado ningún pago.**

select distinct(cl.id\_cliente)

from clientes as cl

left join pagos as pg on cl.id\_cliente = pg.fk\_cliente

join pedidos as pd on cl.id\_cliente = pd.fk\_cliente

where pg.fk\_cliente is null;

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

select sum(dp.cantidad) \* count(pd.id\_pedido)

from detalles\_pedido as dp

join pedidos as pd on dp.fk\_pedido = pd.id\_pedido;

--

select dp.fk\_pedido, sum(dp.cantidad) cant

from detalles\_pedido dp

group by dp.fk\_pedido;

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

select distinct c.id\_cliente, c.nombre\_cliente, gp.id\_gama, gp.gama

from clientes c

join pedidos p on c.id\_cliente = p.fk\_cliente

join detalles\_pedido dp on p.id\_pedido = dp.fk\_pedido

join productos pro on dp.fk\_producto = pro.id\_producto

join gamas\_productos gp on pro.fk\_gama = gp.id\_gama

order by c.id\_cliente;

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

select \*

from empleados

where id\_empleado not in (select fk\_empleado\_rep\_ventas from clientes);

**-- 36 Devuelve un listado con los distintos estados por los que puede pasar un pedido.**

select distinct estado

from pedidos;

**-- 37 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.**

**-- El listado deberá estar ordenado por su precio de venta, mostrando en primer lugar los de mayor precio.**

select \*

from productos p

where fk\_gama in (select id\_gama from gamas\_productos where gama = 'Ornamentales')

and p.cantidad\_en\_stock > 100

order by p.precio\_venta desc;

select \*

from productos p

join gamas\_productos gp on p.fk\_gama = gp.id\_gama

where gp.gama = 'Ornamentales' and p.cantidad\_en\_stock > 100

order by p.precio\_venta desc;

**-- 38 Calcula el precio de venta del producto más caro y más barato en una misma consulta.**

select max(precio\_venta), min(precio\_venta)

from productos;

select \*

from productos

where precio\_venta = (select max(precio\_venta) from productos)

or precio\_venta = (select min(precio\_venta) from productos);

**-- 39 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.id\_cliente, c.nombre\_cliente, e.id\_empleado, e.nombre, o.oficina

from clientes c

join empleados e on c.fk\_empleado\_rep\_ventas = e.id\_empleado

join oficinas o on e.fk\_oficina = o.id\_oficina;

**-- 40 Devuelve el nombre de los representantes de ventas y el número de clientes al que atiende cada uno.**

select e.id\_empleado, e.nombre, count(c.id\_cliente)

from clientes c

join empleados e on c.fk\_empleado\_rep\_ventas = e.id\_empleado

group by e.id\_empleado;

**-- 41 Devuelve el nombre del cliente, el nombre y primer apellido de su representante de ventas y el número de teléfono de la oficina del representante de ventas, de aquellos clientes que no hayan realizado ningún pago.**

select cli.id\_cliente, cli.nombre\_cliente, e.id\_empleado, e.nombre, e.apellido1, o.telefono

from (select id\_cliente, nombre\_cliente, fk\_empleado\_rep\_ventas

from clientes left join pagos on id\_cliente = fk\_cliente

where id\_transaccion is null) cli

join empleados e on cli.fk\_empleado\_rep\_ventas = e.id\_empleado

join oficinas o on e.fk\_oficina = o.id\_oficina;

**-- 42 Devuelve el nombre de los clientes que hayan hecho pedidos en 2008 ordenados alfabéticamente de menor a mayor.**

-- sin subconsulta, solo con join

select distinct c.\*

from clientes c

join pedidos ped on c.id\_cliente = ped.fk\_cliente

where year(fecha\_pedido) = 2008

order by c.nombre\_cliente;

select \*

from clientes c

join (select distinct fk\_cliente from pedidos where year(fecha\_pedido) = 2008) ped on c.id\_cliente = ped.fk\_cliente

order by c.nombre\_cliente;

select \*

from clientes c

where id\_cliente in (select fk\_cliente from pedidos where year(fecha\_pedido) = 2008)

order by c.nombre\_cliente;

**-- 43 Devuelve el nombre, los apellidos y el email de los empleados que están a cargo de Alberto Soria.**

select \* from empleados where nombre = 'Alberto';

select emp.nombre, emp.apellido1, emp.apellido2, emp.email, jefe.nombre, jefe.apellido1

from empleados as emp

join empleados as jefe on emp.fk\_jefe = jefe.id\_empleado

where jefe.nombre = 'Alberto' and jefe.apellido1 = 'Soria';

**-- 44 Devuelve un listado de los productos que han aparecido en un pedido alguna vez.**

select \*

from productos as prod

where id\_producto in (select fk\_producto from detalles\_pedido);

**-- 45 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.**

**-- 46 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 prod.id\_producto,prod.nombre, sum(dp.cantidad)as cantidades

from productos as prod

join detalles\_pedido as dp on prod.id\_producto = dp.fk\_producto

group by id\_producto

order by cantidades desc limit 20;

**-- 47 Devuelve el listado de clientes donde aparezca el nombre del cliente, el nombre y primer apellido de su representante de ventas y la ciudad donde está su oficina.**

select cl.nombre\_cliente, emp.nombre,emp.apellido1, ofi.ciudad

from clientes as cl

join empleados as emp on cl.fk\_empleado\_rep\_ventas = emp.id\_empleado

join oficinas as ofi on emp.fk\_oficina = ofi.id\_oficina;

**-- 48 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: Utilizando la función YEAR de MySQL., Utilizando la función DATE\_FORMAT de MySQL., Sin utilizar ninguna de las funciones anteriores.**

select id\_cliente

from clientes

where id\_cliente in (select fk\_cliente from pagos where year(fecha\_pago) = 2008);

**-- 49 Obtén un listado con el nombre de cada cliente y el nombre y apellido de su representante de ventas.**

select cl.nombre\_cliente, emp.nombre, emp.apellido1

from clientes as cl

join empleados as emp on cl.fk\_empleado\_rep\_ventas = emp.id\_empleado;

**-- 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 c.nombre\_cliente, c.nombre\_contacto, c.apellido\_contacto, min(p.fecha\_pago) as primer\_pago, max(p.fecha\_pago) as ultimo\_pago

from clientes as c

join pagos as p on c.id\_cliente = p.fk\_cliente

group by c.id\_cliente, c.nombre\_cliente, c.nombre\_contacto, c.apellido\_contacto

order by c.nombre\_cliente;

**-- 51 Los clientes cuyo límite de crédito sea mayor que los pagos que haya realizado. (Sin utilizar INNER JOIN).**

select cl.nombre\_cliente, cl.limite\_credito

from clientes as cl

where cl.limite\_credito > (select sum(total) as total\_gasto

from pagos

where fk\_cliente = cl.id\_cliente);

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

select e.nombre, e.fk\_jefe, j.nombre

from empleados as e

join empleados as j on e.fk\_jefe = j.id\_empleado;

**-- 53 ¿Cuántos pedidos hay en cada estado? Ordena el resultado de forma descendente por el número de pedidos.**

select pd.estado, count(pd.id\_pedido)as n\_pedidos

from pedidos as pd

group by pd.estado

order by n\_pedidos desc;

**-- 54 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.**

**-- 55 Devuelve un listado con el nombre de los todos los clientes españoles.**

select nombre\_cliente, pais

from clientes

where pais = 'Spain';

**-- 56 Devuelve un listado que muestre solamente los empleados que**

**-- no tienen un cliente asociado junto con los datos de la oficina donde trabajan.**

select emp.id\_empleado, cl.id\_cliente, ofi.\*

from empleados as emp

left join clientes as cl on emp.id\_empleado = cl.fk\_empleado\_rep\_ventas

join oficinas as ofi on emp.fk\_oficina = ofi.id\_oficina

where cl.fk\_empleado\_rep\_ventas is null;

**-- 57 Devuelve el producto que menos unidades tiene en stock.**

select prod.nombre, prod.cantidad\_en\_stock

from productos as prod

where prod.cantidad\_en\_stock = (select min(cantidad\_en\_stock) from productos);

**-- 58 Calcula cuántos clientes tiene cada una de las ciudades que empiezan por M**

select distinct(count(cl.id\_cliente)), cl.ciudad

from clientes as cl

where cl.ciudad like'M%'

group by ciudad;

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

select \*

from pedidos

where month(fecha\_entrega) = 1;

select \* from pedidos;

**-- 60 ¿Cuál fue el pago medio en 2009?**

select avg(pg.total) as pago\_medio\_2009

from pagos as pg

where year(pg.fecha\_pago) = 2009;

**-- 61 Devuelve el nombre del cliente con mayor límite de crédito.**

select cl.nombre\_cliente, cl.limite\_credito

from clientes as cl

where cl.limite\_credito >= (select max(limite\_credito) from clientes);

**-- 62 Devuelve el listado de clientes indicando el nombre del cliente y cuántos pedidos ha realizado. Tenga en cuenta que pueden existir clientes que no han realizado ningún pedido.**

select cl.nombre\_cliente, count(pd.id\_pedido)

from clientes as cl

join pedidos as pd on cl.id\_cliente = pd.fk\_cliente

group by id\_cliente;

**-- 63 Devuelve un listado de los productos que nunca han aparecido en un pedido. El resultado debe mostrar el nombre y la descripción.**

select prod.nombre, dp.fk\_pedido

from productos as prod

left join detalles\_pedido as dp on prod.id\_producto = dp.fk\_producto

where fk\_producto is null;

**-- 64 Devuelve el nombre, apellidos, puesto y teléfono de la oficina de aquellos empleados que no sean representante de ventas de ningún cliente.**

**-- 65 Devuelve un listado que muestre el nombre de cada empleados, el nombre de su jefe y el nombre del jefe de sus jefe.**

select e.id\_empleado, e.nombre as nombre\_empleado,j.id\_empleado, j.nombre as nombre\_jefe,jj.id\_empleado, jj.nombre as jefe\_del\_jefe

from empleados as e

left join empleados as j on e.fk\_jefe = j.id\_empleado

left join empleados as jj on j.fk\_jefe = jj.id\_empleado

order by jefe\_del\_jefe;

**-- 66 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(total), year(fecha\_pago)

from pagos

group by year(fecha\_pago)

order by year(fecha\_pago);

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

**-- 68 Devuelve el nombre del cliente con mayor límite de crédito utilizando una subconsulta**

select nombre\_cliente, limite\_credito

from clientes

where limite\_credito = (select max(limite\_credito) from clientes);

**-- 69 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.**

**-- Utilizando la función ADDDATE de MySQL., Utilizando la función DATEDIFF de MySQL.,**

**-- ¿Sería posible resolver esta consulta utilizando el operador de suma + o resta -?**

select pd.id\_pedido, cl.id\_cliente, pd.fecha\_esperada, pd.fecha\_entrega

from pedidos as pd

join clientes as cl on pd.fk\_cliente = cl.id\_cliente

where adddate(pd.fecha\_entrega,2) <= pd.fecha\_esperada;

select pd.id\_pedido, cl.id\_cliente, pd.fecha\_esperada, pd.fecha\_entrega

from pedidos as pd

join clientes as cl on pd.fk\_cliente = cl.id\_cliente

where datediff(pd.fecha\_esperada,pd.fecha\_entrega) >= 2;

-- Aunque devuelve lo mismo, no es correcto restar fechas de la misma manera que lo es sumarlas, etc

-- datediff es la manera correcta de hacer la resta de dos fechas.

select pd.id\_pedido, cl.id\_cliente, pd.fecha\_esperada, pd.fecha\_entrega,fecha\_esperada + fecha\_entrega as incorrecto

from pedidos as pd

join clientes as cl on pd.fk\_cliente = cl.id\_cliente

where fecha\_esperada - fecha\_entrega >= 2;

**-- 70 Devuelve el nombre, apellido1 y cargo de los empleados que no representen a ningún cliente.**

select emp.nombre, emp.apellido1, emp.puesto

from empleados as emp

left join clientes as cl on emp.id\_empleado = cl.fk\_empleado\_rep\_ventas

where cl.fk\_empleado\_rep\_ventas is null;

**-- 71 Devuelve el nombre del producto que tenga el precio de venta más caro utilizando una subconsulta**

select prod.nombre, prod.precio\_venta

from productos as prod

where prod.precio\_venta = (select max(precio\_venta) from productos);

**-- 72 Devuelve un listado con los nombres de los clientes y el total pagado por cada uno de ellos.**

**-- Tenga en cuenta que pueden existir clientes que no han realizado ningún pago.**

select cl.id\_cliente, cl.nombre\_cliente, coalesce(sum(pg.total), 0) as total

from clientes as cl

left join pagos as pg on cl.id\_cliente = pg.fk\_cliente

group by id\_cliente

order by total;

**-- 73 ¿Cuántos clientes tiene cada país?**

select count(id\_cliente), pais

from clientes

group by pais;

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

**-- no se puede hacer, enunciado mal**

**-- 75 Lista las ventas totales de los productos que hayan facturado más de 3000 euros.**

**-- Se mostrará el nombre, unidades vendidas, total facturado y total facturado con impuestos (21% IVA).**

select dp.fk\_pedido, prod.nombre,dp.cantidad, sum(dp.cantidad) cant, sum(dp.cantidad \* dp.precio\_unidad) as total

from detalles\_pedido dp

join productos as prod on prod.id\_producto = dp.fk\_producto

group by id\_producto

having total > 3000;

**-- 76 Calcula el número de productos diferentes que hay en cada uno de los pedidos.**

select count(fk\_producto) as cant,fk\_pedido

from detalles\_pedido

group by fk\_pedido

order by cant desc;

**-- 77 ¿Cuántos empleados hay en la compañía?**

select count(id\_empleado)

from empleados;

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

select cl.nombre\_cliente, pd.fecha\_esperada,pd.fecha\_entrega

from clientes as cl

join pedidos as pd on cl.id\_cliente = pd.fk\_cliente

where datediff(pd.fecha\_entrega, pd.fecha\_esperada) >= 1;

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

-- NO HAY. LA FK\_OFICINA EN EMPLEADOS ES NOT NULL

-- \*\*\* Vistas \*\*\*

**-- 80 Escriba una vista que se llame listado\_pagos\_clientes que muestre un listado donde aparezcan todos los clientes y los pagos que ha realizado cada uno de ellos.**

**-- La vista deberá tener las siguientes columnas: nombre y apellidos del cliente concatenados, teléfono, ciudad, pais, fecha\_pago, total del pago, id de la transacción**

create or replace view listado\_pagos\_clientes as

select id\_cliente,concat(c.nombre\_contacto, c.apellido\_contacto)as n\_cliente,telefono, c.ciudad,c.pais, pg.fecha\_pago, pg.total, pg.id\_transaccion

from clientes as c

join pagos as pg on c.id\_cliente = pg.fk\_cliente;

**-- 81 Escriba una vista que se llame listado\_pedidos\_clientes que muestre un listado donde aparezcan todos los clientes y los pedidos que ha realizado cada uno de ellos.**

**-- La vista deáber tener las siguientes columnas: nombre y apellidos del cliente concatendados, teléfono, ciudad, pais, código del pedido, fecha del pedido, fecha esperada, fecha de entrega y**

**-- la cantidad total del pedido,**

**-- que será la suma del producto de todas las cantidades por el precio de cada unidad, que aparecen en cada línea de pedido.**

create or replace view listado\_pedidos\_clientes as

select id\_cliente, concat(c.nombre\_contacto, c.apellido\_contacto)as n\_cliente,telefono, c.ciudad,c.pais,pd.id\_pedido,

pd.fecha\_pedido,pd.fecha\_esperada, pd.fecha\_entrega,sum(dp.cantidad \* dp.precio\_unidad)as total,pd.estado

from clientes as c

join pedidos as pd on c.id\_cliente = pd.fk\_cliente

join detalles\_pedido as dp on pd.id\_pedido = dp.fk\_pedido

group by pd.id\_pedido;

**-- 82 Utilice las vistas que ha creado en los pasos anteriores para devolver**

**-- un listado de los clientes de la ciudad de Madrid que han realizado pagos.**

select id\_cliente, n\_cliente, ciudad, id\_transaccion

from listado\_pagos\_clientes

where ciudad = 'Madrid';

**-- 83 Utilice las vistas que ha creado en los pasos anteriores para devolver**

**-- un listado de los clientes que todavía no han recibido su pedido.**

select id\_cliente, n\_cliente,estado

from listado\_pedidos\_clientes

where estado = 'Entregado';

**-- 84 Utilice las vistas que ha creado en los pasos anteriores para calcular**

**-- el número de pedidos que se ha realizado cada uno de los clientes.**

select id\_cliente, count(\*) as cant\_pedidos

from listado\_pedidos\_clientes

group by id\_cliente;

**-- 85 Utilice las vistas que ha creado en los pasos anteriores para calcular el valor del pedido máximo y mínimo que ha realizado cada cliente.**

select id\_cliente, max(total) as mayor, min(total)as menor

from listado\_pedidos\_clientes

group by id\_cliente;