Base de datos

Jardinería

**Vistas**

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

* CREATE VIEW rechazados AS

SELECT \* FROM pedido

WHERE estado="rechazado";

select \* FROM rechazados;

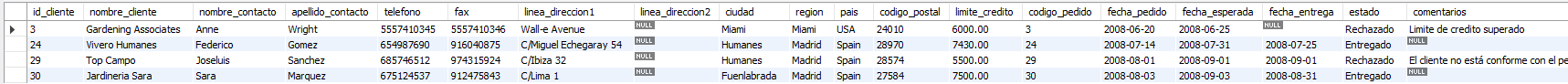


1. Listar todos los clientes que compraron en el mes de julio, junio y agosto del 2008

* CREATE VIEW compras\_meses AS

SELECT \* FROM cliente INNER JOIN pedido ON pedido.codigo\_pedido = cliente. id\_cliente where fecha\_pedido like "2008-06-\_\_" or fecha\_pedido like "2008-07-\_\_" or fecha\_pedido like "2008-08-\_\_";

select \* FROM compras\_meses;



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

* CREATE VIEW pedidos\_entregados\_enero AS

SELECT \* FROM pedido where estado="entregado" and fecha\_entrega like "\_\_\_\_-01-\_\_";

select \* FROM pedidos\_entregados\_enero;

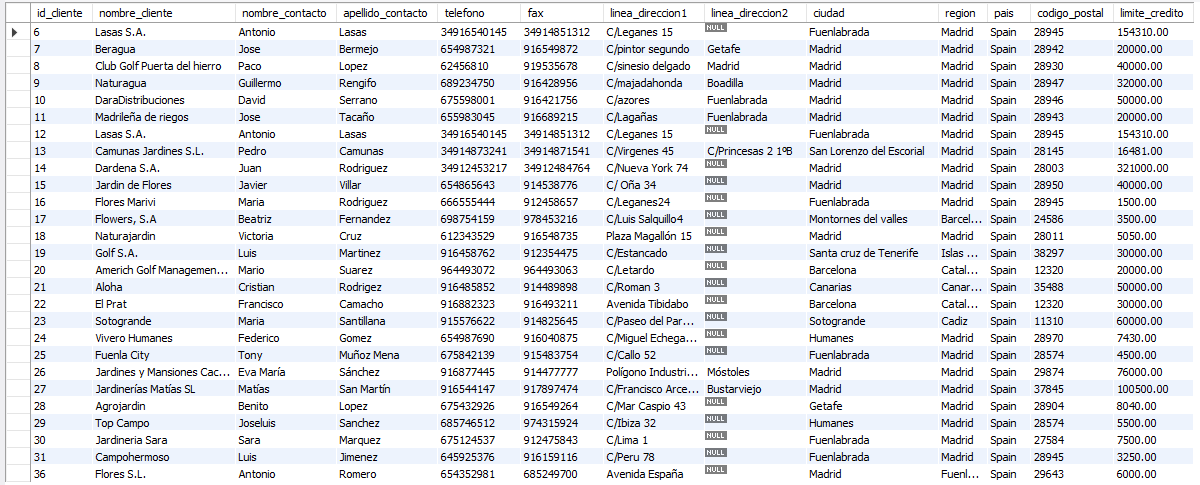


1. Devuelve un listado con el nombre de los todos los clientes españoles

* CREATE VIEW clientes\_españoles AS

SELECT \* FROM cliente where pais="spain";

select \* FROM clientes\_españoles;



1. Contar cuantos clientes no son españoles

* CREATE VIEW clientes\_no\_españoles AS

SELECT count(\*) FROM cliente where not pais="spain";

select \* FROM clientes\_no\_españoles;



1. Mostrar todos los pedidos del cliente “Flores Marivi” donde aparezca nombre del cliente, fecha del pedido, numero del pedido nombre de los productos del pedido cantidad y precio unitario

* CREATE VIEW Flores\_Marivi AS

SELECT cliente.nombre\_cliente, pedido.fecha\_pedido, detalle\_pedido.codigo\_pedido, producto.nombre,

detalle\_pedido.cantidad, producto.precio\_proveedor

FROM pedido

INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

where cliente.nombre\_cliente="Flores Marivi";

select \* FROM Flores\_Marivi;



1. Traer un reporte con el nombre del cliente, número de factura y el valor total del pedido que compró del código del cliente 4

* CREATE VIEW reporte AS

SELECT cliente.id\_cliente, cliente.nombre\_cliente, pedido.codigo\_pedido as "numero de factura", (detalle\_pedido.cantidad \* detalle\_pedido.precio\_unidad)"valor pedido"

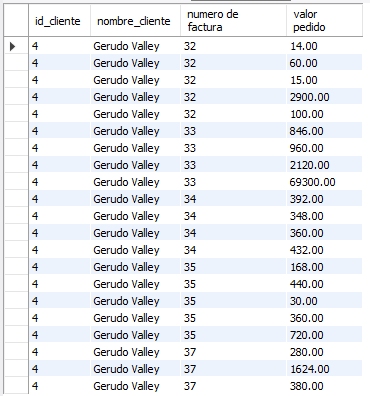
FROM pedido

INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

where cliente.id\_cliente=4;

select \* FROM reporte;



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

* CREATE VIEW no\_pedido AS

SELECT \*

FROM cliente where not exists(select \* from pedido where pedido.codigo\_cliente = cliente.id\_cliente)

select \* FROM no\_pedido;

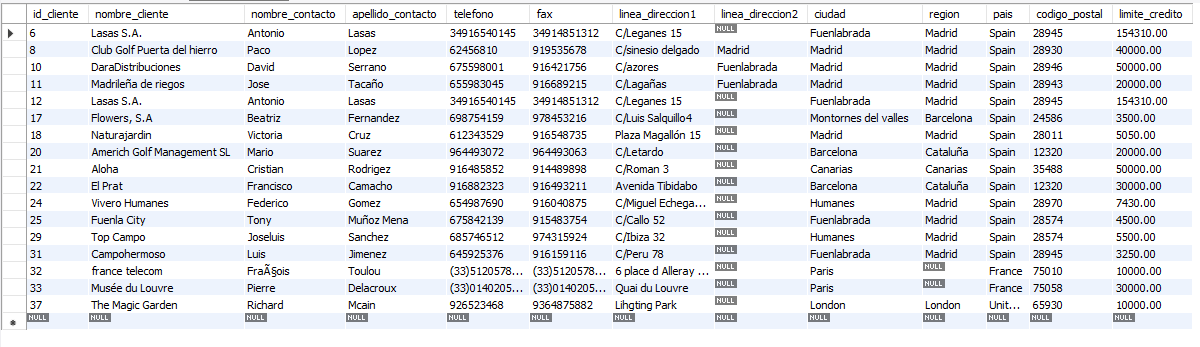
* CREATE VIEW no\_pedido2 AS

SELECT \*

FROM cliente

left join pedido on pedido.codigo\_cliente = cliente.id\_cliente where pedido.codigo\_cliente is null;

select \* FROM no\_pedido2;



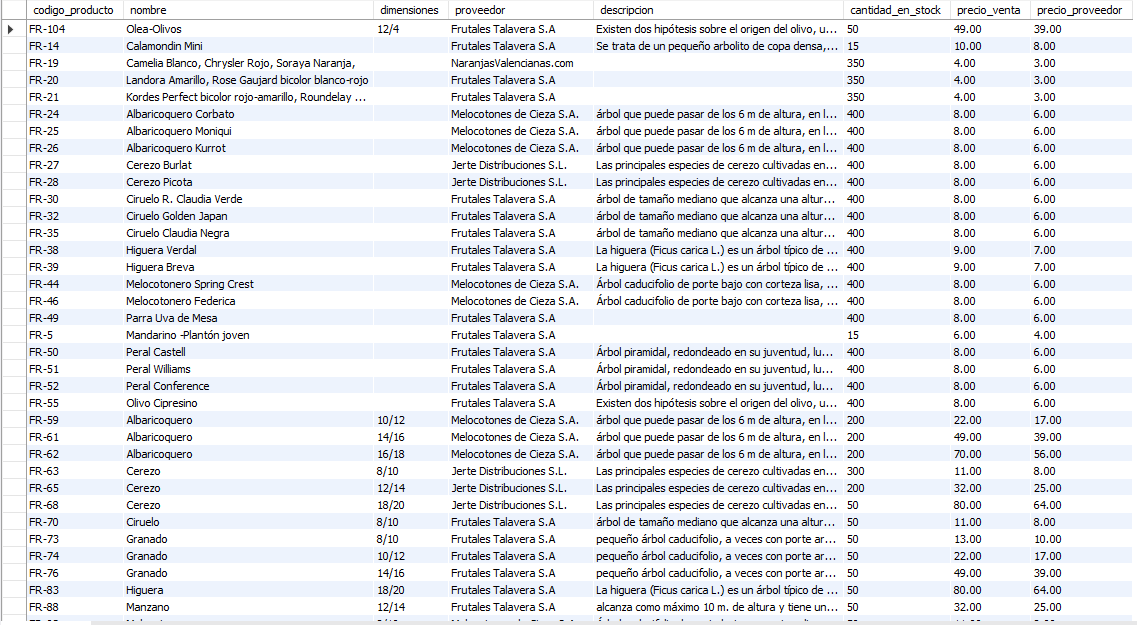
1. Devuelve un listado de los productos que nunca han aparecido en un pedido.

* CREATE VIEW no\_producto AS

SELECT \*

FROM producto where not exists(select \* from detalle\_pedido where producto.codigo\_producto = detalle\_pedido.codigo\_producto);

select \* FROM no\_producto;



1. ¿Cuántos clientes tienen cada país?

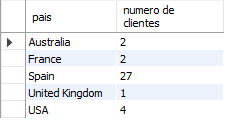
* CREATE VIEW contar\_paises AS

SELECT cliente.pais as pais, count(\*) as "numero de clientes"

FROM cliente

group by cliente.pais;

select \* FROM contar\_paises;



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

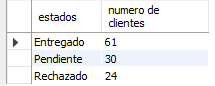
* CREATE VIEW contar\_estados AS

SELECT pedido.estado as estados, count(\*) as "numero de clientes"

FROM pedido

group by pedido.estado order by count(\*)desc;

select \* FROM contar\_estados;



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

* CREATE VIEW max\_and\_min\_producto AS

SELECT max(producto.precio\_venta) as "producto mas caro", min(producto.precio\_venta) as "producto mas barato"

FROM producto ;

select \* FROM max\_and\_min\_producto;



1. Traer un reporte con el nombre del cliente, número de factura y nombre del producto, precio del producto, cantidad del producto y crear una columna llamada subtotal que compró cliente de código 6 y el cliente llamado Pedro

* CREATE VIEW pedro\_6 AS

SELECT cliente.nombre\_contacto, detalle\_pedido.codigo\_pedido as "numero factura",

producto.nombre, producto.precio\_proveedor, detalle\_pedido.cantidad, (detalle\_pedido.cantidad\*producto.precio\_proveedor)subtotal

FROM pedido

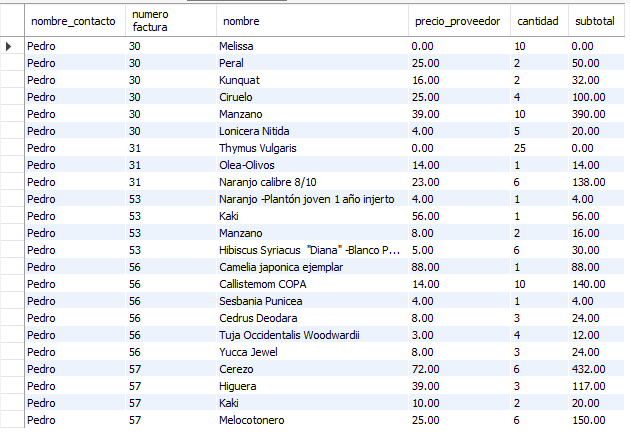
INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

where cliente.id\_cliente=6 or nombre\_contacto="Pedro";

select \* FROM pedro\_6;



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

* CREATE VIEW contar\_paises AS

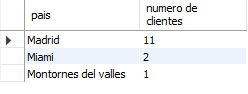
SELECT cliente.ciudad as pais, count(\*) as "numero de clientes"

FROM cliente

where cliente.ciudad like "M%"

group by cliente.ciudad;

select \* FROM contar\_paises;



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

* CREATE VIEW productos\_diferentes AS

SELECT producto.codigo\_producto, producto.nombre, count(producto.nombre)

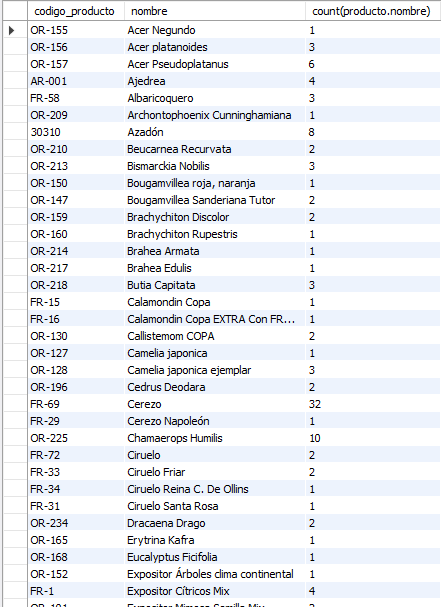
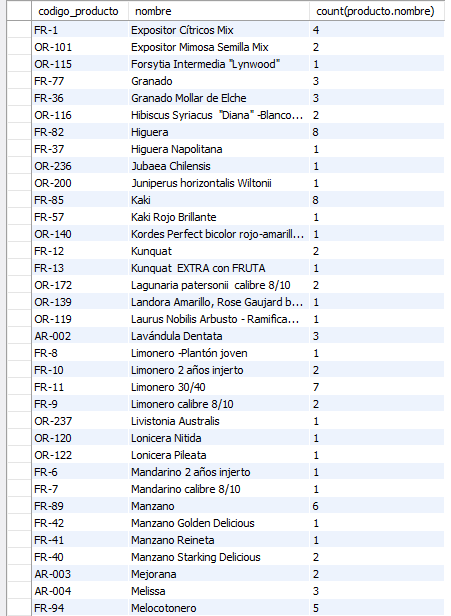
FROM pedido

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

group by producto.nombre;

select \* FROM productos\_diferentes;



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

* CREATE VIEW cant\_productos\_diferentes AS

SELECT count(producto.nombre)

FROM pedido

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto;

select \* FROM cant\_productos\_diferentes;



1. 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.

* CREATE VIEW productos\_mas\_vendidos AS

SELECT producto.codigo\_producto, producto.nombre, count(producto.nombre) as "Unidades vendidas"

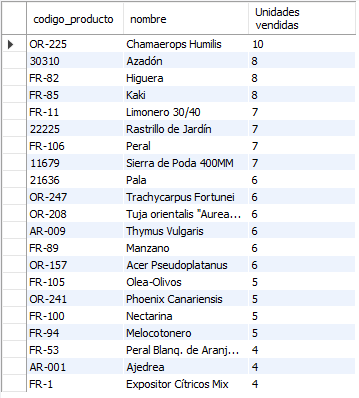
FROM pedido

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

group by producto.nombre order by count(producto.nombre) desc limit 1,21;

select \* FROM productos\_mas\_vendidos;



1. 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

* CREATE VIEW facturacion\_empresa AS

SELECT sum((detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad)) as "base imponible",

(sum((detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad))\*21/100)as iva,

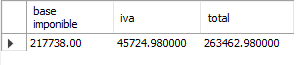
(sum((detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad))+(sum((detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad))\*21/100)) as total

FROM pedido

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto;

select \* FROM facturacion\_empresa;



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

* CREATE VIEW total\_con\_iva\_producto AS

SELECT producto.codigo\_producto, sum((detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad)) as "base imponible",

(sum((detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad))\*21/100)as iva,

(sum((detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad))+(sum((detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad))\*21/100)) as total

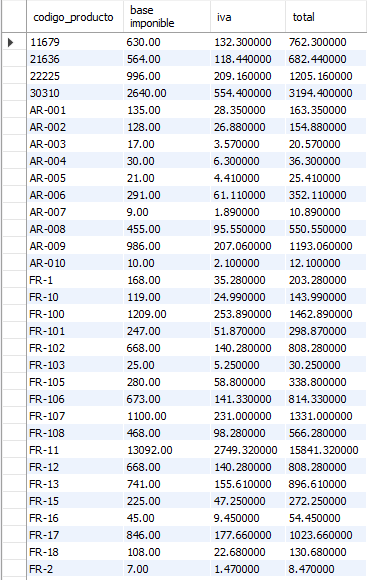
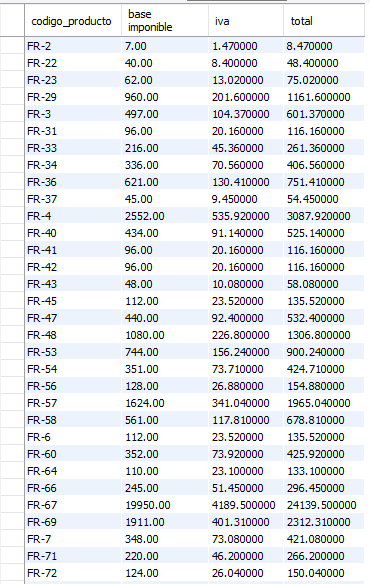
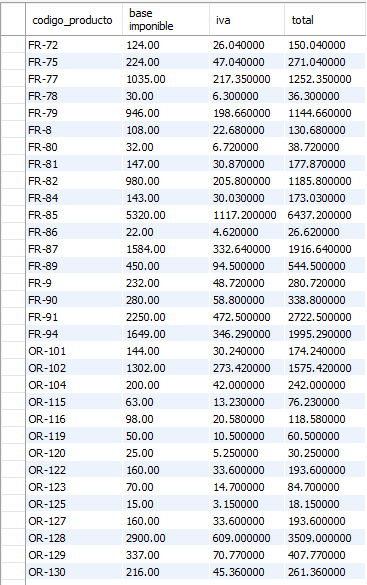
FROM pedido

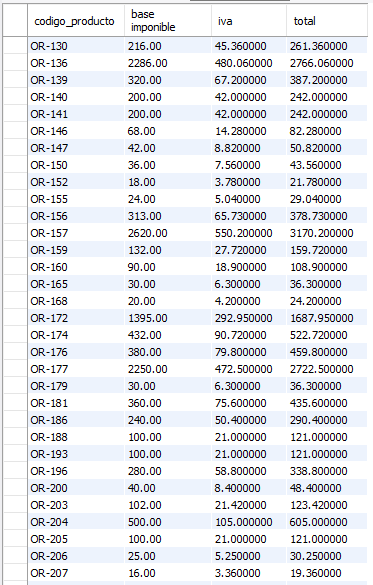
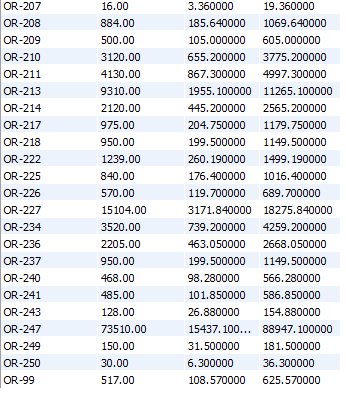
INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

group by producto.codigo\_producto;

select \* FROM total\_con\_iva\_producto;

**Procedimientos**

1. Traer un listado de los productos del pedido del id del cliente = <parámetro> de la fecha = <parámetro>

* delimiter //

CREATE PROCEDURE cliente\_fecha(in id INTEGER,fecha varchar(10))

begin

select \*

FROM pedido

INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

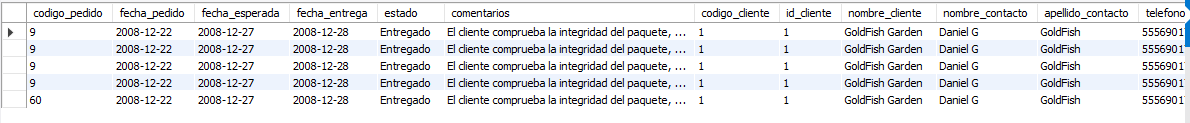
INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

where cliente.id\_cliente=id and pedido.fecha\_pedido=fecha;

end //

delimiter ;

call cliente\_fecha(1,"2008-12-22");



1. Traer el valor total comprado del producto “Cerezo” en todos los pedidos del año = <parámetro>

* delimiter //

CREATE PROCEDURE cerezo(in anio varchar(4),out total int)

begin

select (detalle\_pedido.cantidad\*detalle\_pedido.precio\_unidad) into total

FROM pedido

INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

where producto.nombre="Cerezo" and pedido.fecha\_pedido like CONCAT(anio , '%') group by producto.nombre;

end //

delimiter ;

call cerezo("2008",@totali);

select @totali;



1. Debido a los constantes reclamos en la jardinería crearemos un procedimiento que busque los pedidos de un cliente por nombre

* delimiter //

CREATE PROCEDURE nombre(in nombres varchar(40))

begin

select pedido.codigo\_pedido, cliente.nombre\_cliente

FROM pedido

INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

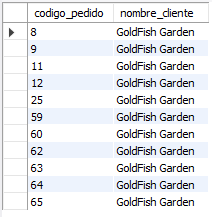
INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

where cliente.nombre\_cliente=nombres group by pedido.codigo\_pedido;

end //

delimiter ;

call nombre("GoldFish Garden");



1. Debido a los constantes reclamos en la jardinería crearemos un procedimiento que busque los pedidos de un cliente por id del cliente

* delimiter //

CREATE PROCEDURE id\_pedidos(in id varchar(3))

begin

select pedido.codigo\_pedido, cliente.nombre\_cliente, cliente.id\_cliente

FROM pedido

INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

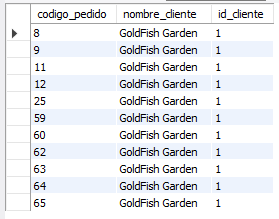
INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

where cliente.id\_cliente=id group by pedido.codigo\_pedido;

end //

delimiter ;

call id\_pedidos("1");



1. Debido a los constantes reclamos en la jardinería crearemos un procedimiento que busque los pedidos de un cliente por fecha.

* delimiter //

CREATE PROCEDURE fecha\_pedidos(in fechas varchar(12))

begin

select pedido.codigo\_pedido, cliente.nombre\_cliente, cliente.id\_cliente, pedido.fecha\_pedido

FROM pedido

INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

where pedido.fecha\_pedido=fechas group by pedido.codigo\_pedido;

end //

delimiter ;

call fecha\_pedidos("2006-01-17");



1. Crear un procedimiento que devuelva el valor total de todo el inventario de la jardinería

* delimiter //

CREATE PROCEDURE inventario()

begin

SELECT producto.codigo\_producto, sum(producto.cantidad\_en\_stock) as "inventario"

FROM pedido

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

group by producto.codigo\_producto;

end //

delimiter ;

call inventario();

* delimiter //

CREATE PROCEDURE inventario2()

begin

SELECT sum(producto.cantidad\_en\_stock) as "total productos en inventario"

FROM pedido

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto;

end //

delimiter ;

call inventario2();



1. Crear un procedimiento que devuelva el número total de los clientes que vengan en el mes=<parámetro>

* delimiter //

CREATE PROCEDURE cliente\_pedidos(in mes varchar(2))

begin

select count(cliente.nombre\_cliente) as total\_clientes

FROM pedido

INNER JOIN cliente ON pedido.codigo\_cliente = cliente.id\_cliente

INNER JOIN detalle\_pedido ON pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

INNER JOIN producto ON detalle\_pedido.codigo\_producto = producto.codigo\_producto

where pedido.fecha\_pedido like CONCAT('%', mes , '%');

end //

delimiter ;

call cliente\_pedidos("01");

