

JARDINERÍA

CHRISTIAN MILLÁN SORIA

1º DAW TARDE

1. Mostrar el nombre de un cliente dado su código.

```
drop procedure if exists nombre_cliente;
delimiter //
create procedure nombre_cliente(codigo int)
begin
select nombre_cliente, codigo_cliente from cliente where codigo_cliente=codigo;
end//
delimiter ;
call nombre_cliente(1);
```

nombre_cliente	codigo_cliente
GoldFish Garden	1

2. Mostrar el precioVenta y la gama de un producto dado su código.

```
drop procedure if exists precio_venta;
delimiter //
create procedure precio_venta(codigo int)
begin
select precio_venta from producto where codigo_producto=codigo;
end//
delimiter ;
call precio_venta(21636);
```

precio_venta
14.00

3. Mostrar toda la información de un pedido dado su código (fechaEsperada, fechaEntrega, fechaPedido, estado y comentarios).

```
drop procedure if exists info_pedido;
delimiter //
create procedure info_pedido(codigo int)
begin
select * from pedido where codigo_pedido=codigo;
end//
```

```
delimiter ;
call info_pedido(1);
```

codigo_pedido	fecha_pedido	fecha_esperada	fecha_entrega	estado	comentarios	codigo_cliente
1	2006-01-17	2006-01-19	2006-01-19	Entregado	Pagado a plazos	5

4. Realizar una función que me devuelva la suma de pagos que ha realizado un cliente. Pasa el código por parámetro.

```
drop procedure if exists suma_pago_cliente_controlado;
delimiter //
create procedure suma_pago_cliente_controlado(codigo int)
begin
declare suma_total decimal(10,2);
select sum(total) into suma_total
from pago
inner join cliente on pago.codigo_cliente=cliente.codigo_cliente
where cliente.codigo_cliente=codigo;
select coalesce(suma_total, -1) as total_pagos;
end //
delimiter ;
call suma_pago_cliente_controlado(1);
```

total_pagos
4000.00

5. Realizar un método o procedimiento que muestre el total en euros de un pedido. Pasa el código por parámetro.

```
drop procedure if exists total_euros;
delimiter //
create procedure total_euros(codigo int)
begin
select
end//
delimiter ;
call total_euros();
```



6. Mostrar el nombre de un cliente dado su código. Controla en caso de que no se encuentre, mostrando un mensaje por ejemplo.

```
drop procedure if exists nombre_cliente_controlado;
delimiter //
create procedure nombre_cliente_controlado(codigo int)
begin
if not exists(select 1 from cliente where codigo_cliente=codigo) then
select 'Error: El cliente no existe.' as mensaje;
else
select nombre_cliente from cliente where codigo_cliente=codigo;
end if;
end //
delimiter ;
call nombre_cliente_controlado(2);
```

mensaje
Error: El cliente no existe.

7. Realizar una función que me devuelva la suma de pagos que ha realizado un cliente. Pasa el código por parámetro. Controla en caso de que no se encuentre, en ese caso devuelve un -1.

```
drop procedure if exists suma_pago_cliente_controlado;
delimiter //
create procedure suma_pago_cliente_controlado(codigo int)
begin
declare suma_total decimal(10,2);
select sum(total) into suma_total
from pago
inner join cliente on pago.codigo_cliente=cliente.codigo_cliente
where cliente.codigo_cliente=codigo;
if suma_total is null then
select -1 as total_pagos;
else
select suma_total as total_pagos;
end if;
end //
delimiter ;
call suma_pago_cliente_controlado(2);
```

total_pagos
-1

8. Realizar un método o procedimiento que muestre el total en euros de un pedido. Pasa el código por parámetro. Controla en caso de que no se encuentre, devolviendo un 0. Pasa otro parámetro como límite, si lo supera, se lanza una excepción propia y devuelve un 0.


```
drop procedure if exists mostrar_total_pedido;
delimiter //
```

```
create procedure mostrar_total_pedido(in codigo int, in limite decimal(15,2))
begin
declare total_pedido decimal(15,2);
select sum(detalle_pedido.cantidad*producto.precio_venta) into total_pedido
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
where pedido.codigo_pedido=codigo;
if total_pedido is null then
select 0 as total_en_euros;
elseif total_pedido>limite then
signal sqlstate '45000' set message_text='El total en euros del pedido supera el
límite máximo permitido.';
select 0 as total_en_euros;
else
select total_pedido as total_en_euros;
end if;
end //
delimiter ;
call mostrar_total_pedido(13, 100);
```

Error

SQL query: [Copy](#)

```
call mostrar_total_pedido(13, 100);
```

MySQL said: 

#1644 - El total en euros del pedido supera el límite máximo permitido.

9. Realiza un resumen con informe de las estadísticas de los pedidos realizados por meses y por años.

```
drop procedure if exists ;
delimiter //
create procedure estadisticas_pedidos()
begin
select
year(fecha_pedido) as anio,
monthname(fecha_pedido) as mes,
count(*) as cantidad_pedidos,
sum(d.precio_unidad*d.cantidad) as monto_total
from pedido p
join detalle_pedido d on p.codigo_pedido=d.codigo_pedido
group by year(fecha_pedido), month(fecha_pedido)
order by anio desc, month(fecha_pedido) desc;
end //
```

```
delimiter ;  
call estadisticas_pedidos();
```

anio	mes	cantidad_pedidos	monto_total
2009	October	3	5759.00
2009	May	6	2659.00
2009	April	15	8440.00
2009	March	22	5699.00
2009	February	25	12503.00
2009	January	90	36905.00
2008	December	28	13541.00
2008	November	15	7023.00
2008	October	16	9224.00
2008	September	3	244.00
2008	August	15	2189.00
2008	July	7	912.00
2008	June	11	11114.00
2008	March	18	12982.00
2008	January	3	605.00
2007	October	13	7823.00
2007	June	4	1532.00
2007	May	4	73226.00
2007	March	3	149.00
2007	January	5	3089.00
2006	May	7	553.00
2006	January	5	1567.00