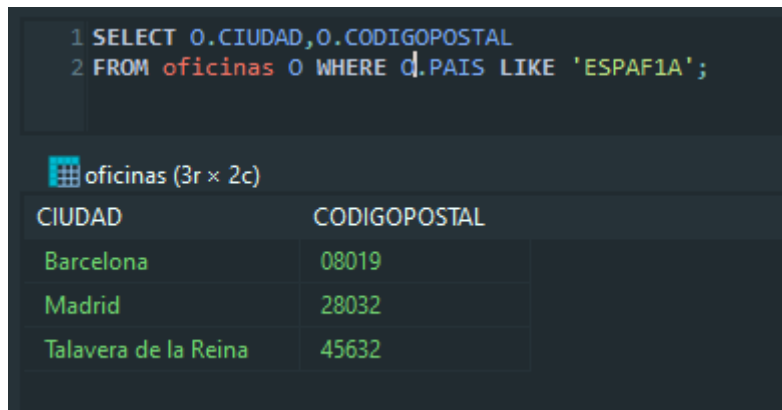


La base de datos Jardinería está hecha en Oracle, y usando la siguiente herramienta Online <http://www.sqlines.com/online> conviértela a MySQL. Sólo hay que pegar el código en la parte izquierda en Oracle, y aplicar el botón CONVERT para que salga en código en SQL de MySQL. Crea la BD Jardineria en tu entorno MySQL y ejecuta las siguientes actividades. Cada ejercicio tendrá que contener su sentencia SQL junto con una imagen que muestre la ejecución y su resultado:

### CONSULTAS SQL:

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

```
SELECT O.CIUDAD,O.CODIGOPOSTAL  
FROM oficinas O WHERE O.PAIS LIKE 'ESPAÑA';
```

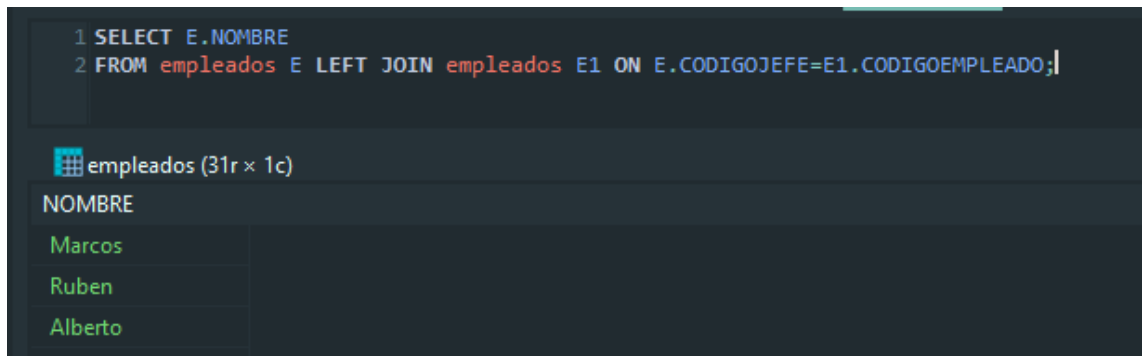


The screenshot shows the SQL query and its results in a dark-themed interface. The query is: `1 SELECT O.CIUDAD,O.CODIGOPOSTAL  
2 FROM oficinas O WHERE O.PAIS LIKE 'ESPAÑA';`. Below the query, there is a table icon and the text "oficinas (3r x 2c)". The table has two columns: "CIUDAD" and "CODIGOPOSTAL". The data rows are: Barcelona (08019), Madrid (28032), and Talavera de la Reina (45632).

CIUDAD	CODIGOPOSTAL
Barcelona	08019
Madrid	28032
Talavera de la Reina	45632

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

```
SELECT E.NOMBRE  
FROM empleados E LEFT JOIN empleados E1 ON  
E.CODIGOJEFE=E1.CODIGOEMPLEADO;
```



The screenshot shows the SQL query and its results in a dark-themed interface. The query is: `1 SELECT E.NOMBRE  
2 FROM empleados E LEFT JOIN empleados E1 ON E.CODIGOJEFE=E1.CODIGOEMPLEADO;`. Below the query, there is a table icon and the text "empleados (31r x 1c)". The table has one column: "NOMBRE". The data rows are: Marcos, Ruben, and Alberto.

NOMBRE
Marcos
Ruben
Alberto

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

```
SELECT E.NOMBRE,E.PUESTO
FROM empleados E
WHERE E.PUESTO NOT LIKE 'DIRECTOR OFICINA';
```

empleados (23r x 2c)

NOMBRE	PUESTO
Marcos	Director General
Ruben	Subdirector Marketing
Alberto	Subdirector Ventas
Maria	Secretaria
Felipe	Representante Ventas
Juan Carlos	Representante Ventas

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

```
SELECT COUNT(*)
FROM empleados E;
```

```
1 SELECT COUNT(*)
2 FROM empleados E;
```

Resultado #1 (1r x 1c)

COUNT(*)
31

5. Muestra el número de clientes norteamericanos.

```
SELECT C.NOMBRECLIENTE
FROM clientes C WHERE C.PAIS LIKE 'USA';
```

```
1 SELECT C.NOMBRECLIENTE
2 FROM clientes C WHERE C.PAIS LIKE 'USA';
```

clientes (4r x 1c)

NOMBRECLIENTE
DGPRODUCTIONS GARDEN
Gardening Associates
Gerudo Valley
Tendo Garden

6. Número de clientes de cada país.

```
SELECT COUNT(*),C.PAIS  
FROM clientes C GROUP BY C.PAIS;
```

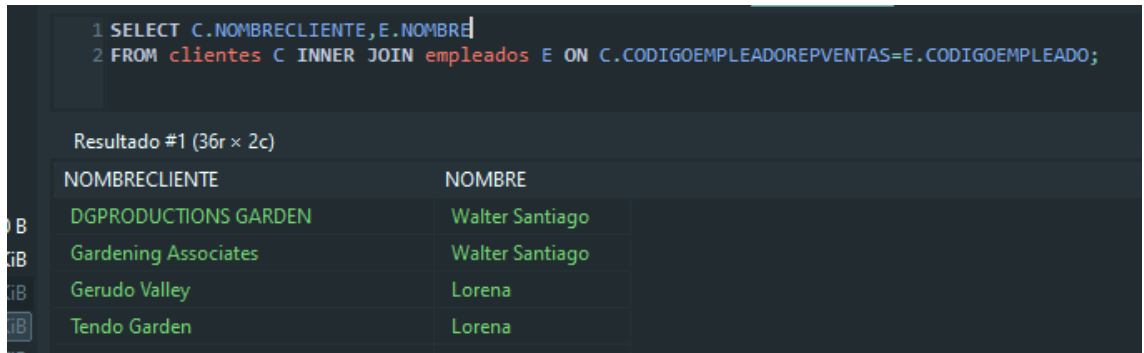


The screenshot shows a SQL query and its result. The query is: `1 SELECT COUNT(*),C.PAIS  
2 FROM clientes C GROUP BY C.PAIS;` The result is a table with two columns: `COUNT(*)` and `PAIS`. The data is as follows:

COUNT(*)	PAIS
4	USA
4	Spain
23	EspaF1a
2	France
2	Australia
1	United Kingdom

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

```
SELECT C.NOMBRECLIENTE,E.NOMBRE  
FROM clientes C INNER JOIN empleados E ON  
C.CODIGOEMPLEADOREPVENTAS=E.CODIGOEMPLEADO;
```



The screenshot shows a SQL query and its result. The query is: `1 SELECT C.NOMBRECLIENTE,E.NOMBRE  
2 FROM clientes C INNER JOIN empleados E ON C.CODIGOEMPLEADOREPVENTAS=E.CODIGOEMPLEADO;` The result is a table with two columns: `NOMBRECLIENTE` and `NOMBRE`. The data is as follows:

NOMBRECLIENTE	NOMBRE
DGPRODUCTIONS GARDEN	Walter Santiago
Gardening Associates	Walter Santiago
Gerudo Valley	Lorena
Tendo Garden	Lorena

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

```
SELECT C.NOMBRECLIENTE, YEAR(P.FECHAPAGO)  
FROM clientes C INNER JOIN pagos P ON  
C.CODIGOCLIENTE=P.CODIGOCLIENTE  
WHERE YEAR(P.FECHAPAGO)=2007;
```

```

1 SELECT C.NOMBRECLIENTE, YEAR(P.FECHAPAGO)
2 FROM clientes C INNER JOIN pagos P ON C.CODIGOCLIENTE=P.CODIGOCLIENTE
3 WHERE YEAR(P.FECHAPAGO)=2007;

```

clientes (6r × 2c)

NOMBRECLIENTE	YEAR(P.FECHAPAGO)
Gerudo Valley	2.007
Gerudo Valley	2.007
Gerudo Valley	2.007
Gerudo Valley	2.007
Gerudo Valley	2.007
Tutifruti S.A	2.007

9. Los posibles estados de un pedido.

```

SELECT P.ESTADO
FROM pedidos P GROUP BY P.ESTADO;

```

```

1 SELECT P.ESTADO
2 FROM pedidos P GROUP BY P.ESTADO;

```

pedidos (4r × 1c)

ESTADO
Entregado
Rechazado
Pendiente
Pendiente

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 pedidos P INNER JOIN clientes C ON
P.CODIGOCLIENTE=C.CODIGOCLIENTE;

```

```

1 SELECT P.CODIGOPEDIDO,C.NOMBRECLIENTE,P.FECHAENTREGA,P.FECHAESPERADA
2 FROM pedidos P INNER JOIN clientes C ON P.CODIGOCLIENTE=C.CODIGOCLIENTE;|

```

Resultado #1 (114r x 4c)

	CODIGOPEDIDO	NOMBRECLIENTE	FECHAENTREGA	FECHAESPERADA
1		Tendo Garden	2006-01-19 00:00:00	2006-01-19 00:00:00
2		Tendo Garden	2007-10-26 00:00:00	2007-10-28 00:00:00
3		Tendo Garden	(NULL)	2008-06-25 00:00:00
4		Tendo Garden	(NULL)	2009-01-26 00:00:00
8		DGPRODUCTIONS GARDEN	2008-11-14 00:00:00	2008-11-14 00:00:00
9		DGPRODUCTIONS GARDEN	2008-12-28 00:00:00	2008-12-27 00:00:00
10		Gardening Associates	(NULL)	2009-01-20 00:00:00

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

```

SELECT PR.NOMBRE,PR.GAMA,PR.CODIGOPRODUCTO
FROM detallepedidos DP INNER JOIN pedidos P ON
P.CODIGOPEDIDO=DP.CODIGOPEDIDO
INNER JOIN productos PR ON
DP.CODIGOPRODUCTO=PR.CODIGOPRODUCTO;

```

```

1 SELECT PR.NOMBRE,PR.GAMA,PR.CODIGOPRODUCTO
2 FROM detallepedidos DP INNER JOIN pedidos P ON P.CODIGOPEDIDO=DP.CODIGOPEDIDO
3 INNER JOIN productos PR ON DP.CODIGOPRODUCTO=PR.CODIGOPRODUCTO;|

```

productos (289r x 3c)

NOMBRE	GAMA	CODIGOPRODUCTO
Peral	Frutales	FR-106
Peral	Frutales	FR-108
Limonero 30/40	Frutales	FR-11
Ajedrea	Aromáticas	AR-001
Thymus Citriodora (Tomillo limF3n)	Aromáticas	AR-008

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

```

SELECT E.NOMBRE,CONCAT(E.APELLIDO1,' ',E.APELLIDO2) AS
'APELLIDOS',O.CIUDAD
FROM empleados E INNER JOIN oficinas O ON
E.CODIGOOFICINA=O.CODIGOOFICINA
WHERE O.CIUDAD LIKE 'BARCELONA';

```

```

1 SELECT E.NOMBRE,CONCAT(E.APELLIDO1,' ',E.APELLIDO2) AS 'APELLIDOS',O.CIUDAD
2 FROM empleados E INNER JOIN oficinas O ON E.CODIGOOFICINA=O.CODIGOOFICINA
3 WHERE O.CIUDAD LIKE 'BARCELONA';|

```

Resultado #1 (4r x 3c)

NOMBRE	APELLIDOS	CIUDAD
Emmanuel	MagaF1a Perez	Barcelona
JosE9 Manuel	Martinez De la Osa	Barcelona
David	Palma Aceituno	Barcelona
Oscar	Palma Aceituno	Barcelona

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

```
SELECT PR.CODIGOPRODUCTO, COUNT(*)
FROM detallepedidos DP INNER JOIN productos PR ON
DP.CODIGOPRODUCTO=PR.CODIGOPRODUCTO
INNER JOIN pedidos P ON DP.CODIGOPEDIDO=P.CODIGOPEDIDO
GROUP BY PR.NOMBRE;
```

```
1 SELECT PR.CODIGOPRODUCTO, COUNT(*)
2 FROM detallepedidos DP INNER JOIN productos PR ON DP.CODIGOPRODUCTO=PR.CODIGOPRODUCTO
3     INNER JOIN pedidos P ON DP.CODIGOPEDIDO=P.CODIGOPEDIDO
4 GROUP BY PR.NOMBRE;|
```

productos (116r × 2c)

CODIGOPRODUCTO	COUNT(*)
11679	7
21636	6
22225	7

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

```
SELECT C.NOMBRECLIENTE
FROM clientes C INNER JOIN pedidos P ON
C.CODIGOCLIENTE=P.CODIGOCLIENTE
WHERE C.CIUDAD LIKE 'MIAMI';
```

```
1 SELECT C.NOMBRECLIENTE
2 FROM clientes C INNER JOIN pedidos P ON C.CODIGOCLIENTE=P.CODIGOCLIENTE
3 WHERE C.CIUDAD LIKE 'MIAMI';|
```

clientes (14r × 1c)

NOMBRECLIENTE
Gardening Associates
Gardening Associates
Gardening Associates

15. Mostrar el precio final de cada pedido.

```
SELECT SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS
'PRECIO_TOTAL', P.CODIGOPEDIDO
FROM detallepedidos DP INNER JOIN pedidos P ON
DP.CODIGOPEDIDO=P.CODIGOPEDIDO
GROUP BY P.CODIGOPEDIDO;
```

```

1 SELECT SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS 'PRECIO_TOTAL',P.CODIGOPEDIDO
2 FROM detallepedidos DP INNER JOIN pedidos P ON DP.CODIGOPEDIDO=P.CODIGOPEDIDO
3 GROUP BY P.CODIGOPEDIDO;

```

pedidos (88r × 2c)

PRECIO_TOTAL	CODIGOPEDIDO
1.567,00	1
7.113,00	2
10.850,00	3

16. Mostrar lo que ha pagado cada cliente.

```

SELECT SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS
'PRECIO_TOTAL',C.NOMBRECLIENTE
FROM detallepedidos DP INNER JOIN pedidos P ON
DP.CODIGOPEDIDO=P.CODIGOPEDIDO
INNER JOIN clientes C ON P.CODIGOCLIENTE=C.CODIGOCLIENTE
GROUP BY C.CODIGOCLIENTE;

```

```

1 SELECT SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS 'PRECIO_TOTAL',C.NOMBRECLIENTE
2 FROM detallepedidos DP INNER JOIN pedidos P ON DP.CODIGOPEDIDO=P.CODIGOPEDIDO
3     INNER JOIN clientes C ON P.CODIGOCLIENTE=C.CODIGOCLIENTE
4 GROUP BY C.CODIGOCLIENTE;

```

clientes (18r × 2c)

PRECIO_TOTAL	NOMBRECLIENTE
6.165,00	DGPRODUCTIONS GARDEN
10.926,00	Gardening Associates
81.849,00	Gerudo Valley
23.794,00	Tendo Garden
2.390,00	Beragua

17. Mostrar el número de productos de cada gama.

```

SELECT COUNT(*) , P.GAMA
FROM productos P GROUP BY P.GAMA;

```

```

1 SELECT COUNT(*),P.GAMA
2 FROM productos P GROUP BY P.GAMA;

```

productos (4r × 2c)

COUNT(*)	GAMA
10	Aromáticas
108	Frutales
4	Herramientas
154	Ornamentales

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

```

SELECT MAX(PR.PRECIOVENTA), PR.NOMBRE, P.CODIGOPEDIDO
FROM productos PR INNER JOIN detallepedidos DP ON
PR.CODIGOPRODUCTO=DP.CODIGOPRODUCTO
INNER JOIN pedidos P ON DP.CODIGOPEDIDO=P.CODIGOPEDIDO
WHERE PR.GAMA LIKE 'AROMATICAS'
GROUP BY P.CODIGOPEDIDO;

```

```

1 SELECT MAX(PR.PRECIOVENTA), PR.NOMBRE, P.CODIGOPEDIDO
2 FROM productos PR INNER JOIN detallepedidos DP ON PR.CODIGOPRODUCTO=DP.CODIGOPRODUCTO
3 INNER JOIN pedidos P ON DP.CODIGOPEDIDO=P.CODIGOPEDIDO
4 WHERE PR.GAMA LIKE 'AROMATICAS'
5 GROUP BY P.CODIGOPEDIDO;

```

Resultado #1 (20r x 3c)

MAX(PR.PRECIOVENTA)	NOMBRE	CODIGOPEDIDO
1,00	Ajedrea	9
1,00	Ajedrea	40
1,00	Ajedrea	90
1,00	Ajedrea	116
1,00	LavEIndula Dentata	23
1,00	Mejorana	41

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

```

SELECT P.CODIGOPEDIDO
FROM pedidos P INNER JOIN detallepedidos DP ON
P.CODIGOPEDIDO=DP.CODIGOPEDIDO
INNER JOIN productos PR ON DP.CODIGOPRODUCTO=PR.CODIGOPRODUCTO
WHERE DP.CANTIDAD>6
GROUP BY DP.CODIGOPEDIDO;

```

```

1 SELECT P.CODIGOPEDIDO
2 FROM pedidos P INNER JOIN detallepedidos DP ON P.CODIGOPEDIDO=DP.CODIGOPEDIDO
3 INNER JOIN productos PR ON DP.CODIGOPRODUCTO=PR.CODIGOPRODUCTO
4 WHERE DP.CANTIDAD>6
5 GROUP BY DP.CODIGOPEDIDO;

```

pedidos (78r x 1c)

CODIGOPEDIDO
1
2

20. Mostrar el código 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);

```

```

1 SELECT PE.CODIGOPEDIDO
2 FROM PEDIDOS PE
3 WHERE
4 (SELECT SUM(DP.CANTIDAD * DP.PRECIOUNIDAD) AS TOTAL
5  FROM PEDIDOS P, DETALLEPEDIDOS DP
6   WHERE P.CODIGOPEDIDO = DP.CODIGOPEDIDO AND PE.CODIGOPEDIDO = P.CODIGOPEDIDO
7   GROUP BY P.CODIGOPEDIDO)
8 >
9 (SELECT AVG(T.TOTAL)
10 FROM (SELECT P.CODIGOPEDIDO, SUM(DP.CANTIDAD * DP.PRECIOUNIDAD) AS TOTAL
11        FROM PEDIDOS P, DETALLEPEDIDOS DP
12         WHERE P.CODIGOPEDIDO = DP.CODIGOPEDIDO
13         GROUP BY P.CODIGOPEDIDO) T);
14 |

```

pedidos (18r x 1c)

codigopedido

9

10

58

21. Muestra el país(cliente) donde menos pedidos se hacen.

```

SELECT COUNT(*),C.PAIS
FROM pedidos P INNER JOIN clientes C ON
P.CODIGOCLIENTE=C.CODIGOCLIENTE
GROUP BY C.PAIS LIMIT 1;

```

```

1 SELECT COUNT(*),C.PAIS
2 FROM pedidos P INNER JOIN clientes C ON P.CODIGOCLIENTE=C.CODIGOCLIENTE
3 GROUP BY C.PAIS LIMIT 1;

```

clientes (1r x 2c)

COUNT(\*) PAIS

30 USA

## FUNCIONES Y PROCEDIMIENTOS

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

```

CREATE PROCEDURE P1( IN codigo INT)
BEGIN
SELECT C.NOMBRECLIENTE
FROM clientes C
WHERE C.CODIGOCLIENTE=codigo;
END

```

```
CALL P1(1);
```

```

1 CREATE PROCEDURE P1( IN codigo INT)
2 BEGIN
3 SELECT C.NOMBRECLIENTE
4 FROM clientes C
5 WHERE C.CODIGOCLIENTE=codigo;
6 END
7
8 CALL P1(1);
9

```

clientes (1r x 1c)	
NOMBRECLIENTE	
DGPRODUCTIONS	GARDEN

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

```

CREATE PROCEDURE P2( IN codigo INT)
BEGIN
SELECT P.PRECIOVENTA,P.GAMA
FROM productos P
WHERE P.CODIGOPRODUCTO=codigo;
END

```

CALL P2 (11679);

```

1 CREATE PROCEDURE P2( IN codigo INT)
2 BEGIN
3 SELECT P.PRECIOVENTA,P.GAMA
4 FROM productos P
5 WHERE P.CODIGOPRODUCTO=codigo;
6 END
7
8 CALL P2(11679);
9

```

productos (1r x 2c)	
PRECIOVENTA	GAMA
14,00	Herramientas

24. Mostrar toda la información de un pedido dado su código (fechaEsperada, fechaEntrega, fechapedido, estado, comentarios)

```

CREATE PROCEDURE P3( IN codigo INT)
BEGIN
SELECT *
FROM pedidos P
WHERE P.CODIGOPEDIDO=codigo;
END

```

CALL P3 (1);

```

1 CREATE PROCEDURE P3( IN codigo INT)
2 BEGIN
3 SELECT *
4 FROM pedidos P
5 WHERE P.CODIGOPEDIDO=codigo;
6 END
7
8 CALL P3(1);
9

```

pedidos (1r x 7c)						
CODIGOPEDIDO	FECHAPEDIDO	FECHAESPERADA	FECHAENTREGA	ESTADO	COMENTARIOS	CODIGOCLIENTE
1	2006-01-17 00:00:00	2006-01-19 00:00:00	2006-01-19 00:00:00	Entregado	(NULL)	5

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

```

CREATE PROCEDURE P4( IN codigo INT)
BEGIN
SELECT SUM(P.CANTIDAD)
FROM pagos P
WHERE P.CODIGOCLIENTE=codigo
GROUP BY P.CODIGOCLIENTE;
END

```

```
CALL P4(1);
```

```

1 CREATE PROCEDURE P4( IN codigo INT)
2 BEGIN
3 SELECT SUM(P.CANTIDAD)
4 FROM pagos P
5 WHERE P.CODIGOCLIENTE=codigo
6 GROUP BY P.CODIGOCLIENTE;
7 END
8
9 CALL P4(1);
10

```

Resultado #1 (1r x 1c)	
SUM(P.CANTIDAD)	
4.000,00	

26. Realizar un método o procedimiento que muestre el total en euros de un pedido, pásale el código por parámetro.

```

CREATE PROCEDURE P5( IN codigo INT)
BEGIN
SELECT SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS 'TOTAL'
FROM detallepedidos DP
WHERE DP.CODIGOPEDIDO=codigo
GROUP BY DP.CODIGOPEDIDO;
END

```

```
CALL P5(1);
```

```

1 CREATE PROCEDURE P5( IN codigo INT)
2 BEGIN
3 SELECT SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) AS 'TOTAL'
4 FROM detallepedidos DP
5 WHERE DP.CODIGOPEDIDO=codigo
6 GROUP BY DP.CODIGOPEDIDO;
7 END
8
9 CALL P5(1);|
10

```

Resultado #1 (1r x 1c)

TOTAL
1.567,00

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

```

CREATE PROCEDURE NOMBRE_CLIENTE(IDCLIENTE DECIMAL)
BEGIN
DECLARE NOMBRE VARCHAR(50) DEFAULT NULL;

SELECT NOMBRECLIENTE INTO NOMBRE
FROM CLIENTES
WHERE CODIGOCLIENTE = IDCLIENTE;

IF NOMBRE IS NULL THEN
SELECT 'ESE CLIENTE NO EXISTE' AS MENSAJE;
ELSE
SELECT NOMBRE;
END IF;
END

CALL NOMBRE_CLIENTE(1);

```

```

3 DECLARE NOMBRE VARCHAR(50) DEFAULT NULL;
4
5 SELECT NOMBRECLIENTE INTO NOMBRE
6 FROM CLIENTES
7 WHERE CODIGOCLIENTE = IDCLIENTE;
8
9 IF NOMBRE IS NULL THEN
10 SELECT 'ESE CLIENTE NO EXISTE' AS MENSAJE;
11 ELSE
12 SELECT NOMBRE;
13 END IF;
14 END
15
16 CALL NOMBRE_CLIENTE(1);|
17

```

Resultado #1 (1r x 1c)

nombre
DGPRODUCTIONS GARDEN

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

```
CREATE FUNCTION P7(idcliente DECIMAL(38,0)) RETURNS INT
DETERMINISTIC
BEGIN
DECLARE suma DECIMAL(38,0) DEFAULT 0;
SELECT ROUND(SUM(cantidad),0) INTO suma FROM pagos
WHERE idcliente = codigocliente
GROUP BY codigocliente;
if suma = 0 then
RETURN -1;
END if;
RETURN suma;
END
SELECT P7(1);
```

```
1 CREATE FUNCTION P7(idcliente DECIMAL(38,0)) RETURNS INT
2 DETERMINISTIC
3 BEGIN
4 DECLARE suma DECIMAL(38,0) DEFAULT 0;
5 SELECT ROUND(SUM(cantidad),0) INTO suma FROM pagos
6 WHERE idcliente = codigocliente
7 GROUP BY codigocliente;
8 if suma = 0 then
9 RETURN -1;
10 END if;
11 RETURN suma;
12 END
13 SELECT P7(1);|
```

Resultado #1 (1r x 1c)

P7(1)
4.000

29. Realizar un método o procedimiento que muestre el total en euros de un pedido, pásale el código por parámetro. Controla en caso de que no se encuentre, en ese caso devuelve un 0. Pásale otro parámetro, si supera ese límite, lanzaremos una excepción propia y devolveremos un 0

```
CREATE PROCEDURE P8(IN codigo DECIMAL)

BEGIN

DECLARE CANTIDAD DECIMAL DEFAULT 0;
DECLARE MENSAJE VARCHAR(50) DEFAULT 'NO EXISTE';

IF EXISTS (SELECT SUM(DP.CANTIDAD*DP.PRECIOUNIDAD) FROM
detallepedidos DP
WHERE DP.CODIGOPEDIDO=codigo) THEN
SELECT COUNT(*) INTO CANTIDAD
FROM detallepedidos DP
WHERE DP.CODIGOPEDIDO=codigo
GROUP BY DP.CODIGOPEDIDO;

IF CANTIDAD=-1 THEN
SELECT MENSAJE;

END
```

```
CALL P8(1);
```

30. Realiza un resumen de las estadísticas de los pedidos realizados por meses y por años (usa cursores).

```
CREATE PROCEDURE P9 ()
BEGIN
DECLARE MES VARCHAR(40);
DECLARE CPEDIDOS INT;
DECLARE ANIO INT;
DECLARE SALIDA INTEGER DEFAULT 0;
DECLARE LISTA VARCHAR(4000) DEFAULT "";
DECLARE CURNAME

CURSOR FOR
SELECT MONTHNAME(P.FECHAPEDIDO) AS MES, YEAR(P.FECHAPEDIDO)
AS ANIO, COUNT(*) AS CANTIDAD
FROM PEDIDOS P
GROUP BY MONTH(P.FECHAPEDIDO), YEAR(P.FECHAPEDIDO)
ORDER BY YEAR(P.FECHAPEDIDO) ASC, MONTH(P.FECHAPEDIDO);

DECLARE CONTINUE HANDLER FOR NOT FOUND SET SALIDA = 1;

OPEN CURNAME;
SET LISTA= '-----PEDIDOS-----'\N';
FETCH CURNAME INTO MES, ANIO, CPEDIDOS;
SET LISTA=CONCAT (LISTA, MES, ' ', ANIO, ' : ', CPEDIDOS, '
PEDIDOS', '\N');

WHILE SALIDA=0 DO
FETCH CURNAME INTO MES, ANIO, CPEDIDOS;
SET LISTA=CONCAT (LISTA, MES, ' ', ANIO, ' : ', CPEDIDOS, '
PEDIDOS', '\N');
END WHILE;

SELECT LISTA;
CLOSE CURNAME;
END;
CALL P9();
```

```
27
28 SELECT LISTA;
29 CLOSE CURNAME;
30 END;
31 CALL P9();
32
```

Resultado #1 (1r x 1c)

LISTA
-----PEDIDOS-----\NJanuary 2006 : 1 PEDIDOSNMay ...