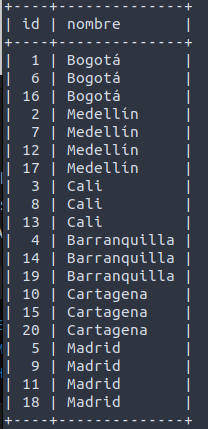
**Consultas sobre una tabla**

1. Devuelve un listado con el código de oficina y la ciudad donde hay oficinas.  
SELECT o.id, c.nombre

FROM oficina o

left JOIN ciudad c ON c.id = o.fkIdCiudad;



2. Devuelve un listado con la ciudad y el teléfono de las oficinas de España.

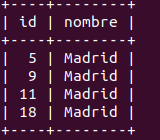
SELECT o.id, c.nombre FROM oficina o

JOIN ciudad c ON c.id = o.fkIdCiudad

JOIN region r ON r.id = c.fkIdRegion

JOIN pais p ON p.id = r.fkIdPais

WHERE p.nombre = 'España';

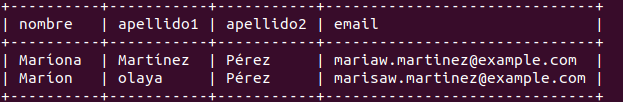


3. Devuelve un listado con el nombre, apellidos y email de los empleados cuyo

jefe tiene un código de jefe igual a 7.

select nombre, apellido1,apellido2,email from empleado

where fkCodigoJefe =7;

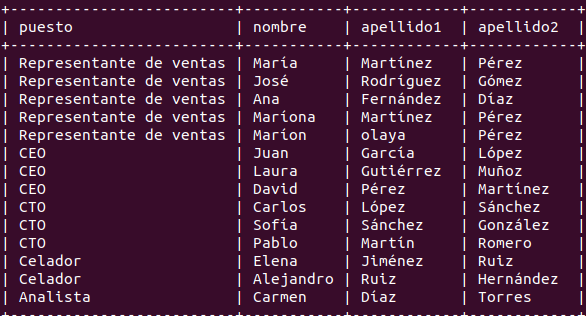


4. Devuelve el nombre del puesto, nombre, apellidos y email del jefe de la

empresa.

select p.puesto, e.nombre, e.apellido1,e.apellido2 from empleado e

join puesto p on p.id=e.fkPuesto;



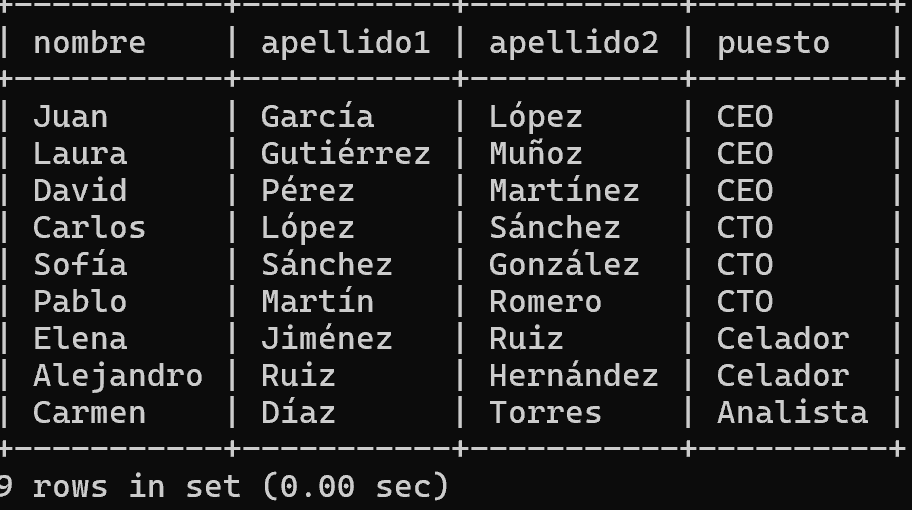
5. Devuelve un listado con el nombre, apellidos y puesto de aquellos

empleados que no sean representantes de ventas.

select e.nombre, e.apellido1,e.apellido2, p.puesto from empleado e

join puesto p on p.id=e.fkPuesto

where p.id !=1;



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

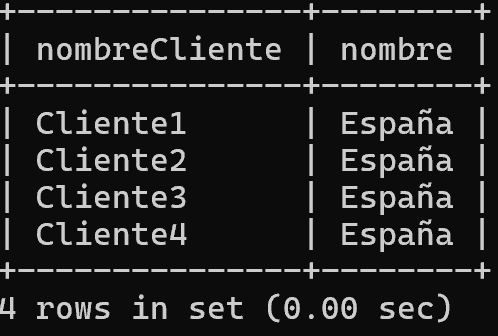
select c.nombreCliente, p.nombre from cliente c

join ciudad ci on ci.id=c.fkIdCiudad

join region r on r.id=ci.fkIdRegion

join pais p on p.id=r.fkIdPais

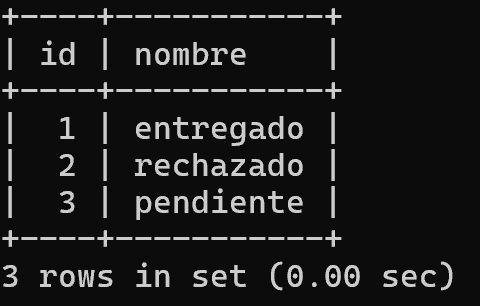
where p.nombre='España';



7. Devuelve un listado con los distintos estados por los que puede pasar un

pedido.

select id, nombre from estado;



8. Devuelve un listado con el código de cliente de aquellos clientes que

realizaron algún pago en 2008. Tenga en cuenta que deberá eliminar

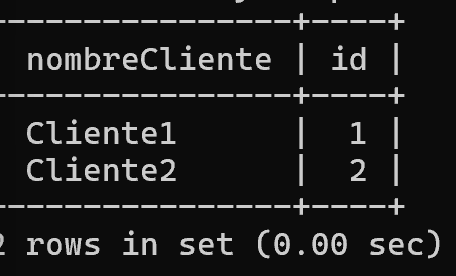
aquellos códigos de cliente que aparezcan repetidos. Resuelva la consulta:

**YEAR:**

select distinct c.nombreCliente, c.id from cliente c

join pago p on p.fkIdCliente=c.id

where year(p.fechaPago)=2008;



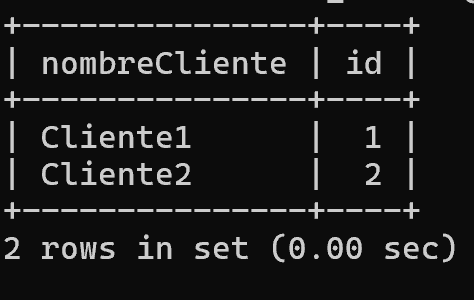
**DATE FORMAT**

SELECT DISTINCT c.nombreCliente, c.id

FROM cliente c

JOIN pago p ON p.fkIdCliente = c.id

WHERE DATE\_FORMAT(p.fechaPago, '%Y') = '2008';



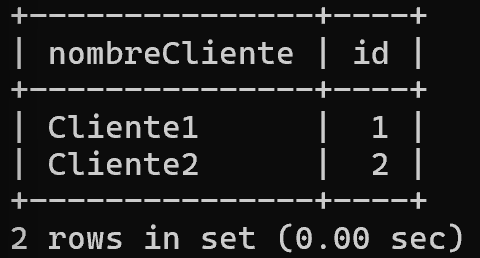
**Sin utilizar ninguna de las funciones anteriores**

SELECT DISTINCT c.nombreCliente, c.id

FROM cliente c

JOIN pago p ON p.fkIdCliente = c.id

WHERE p.fechaPago LIKE '2008%';



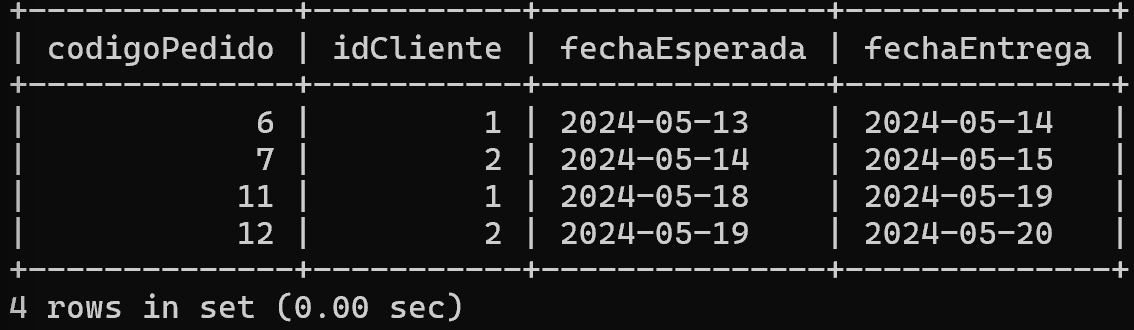
9. Devuelve un listado con el código de pedido, có digo de cliente, fecha

esperada y fecha de entrega de los pedidos que no han sido entregados a

tiempo.

select id as codigoPedido, fkIdCliente as idCliente, fechaEsperada, fechaEntrega from pedido

where fechaEntrega>fechaEsperada;



10. Devuelve un listado con el código de pedido, có digo de cliente, fecha

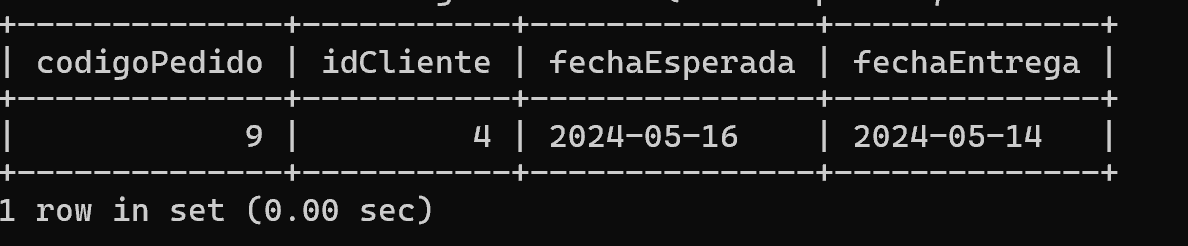
esperada y fecha de entrega de los pedidos cuya fecha de entrega ha sido al

menos dos dí as antes de la fecha esperada.

• Utilizando la funció n ADDDATE de MySQL.

select id as codigoPedido, fkIdCliente as idCliente, fechaEsperada, fechaEntrega from pedido

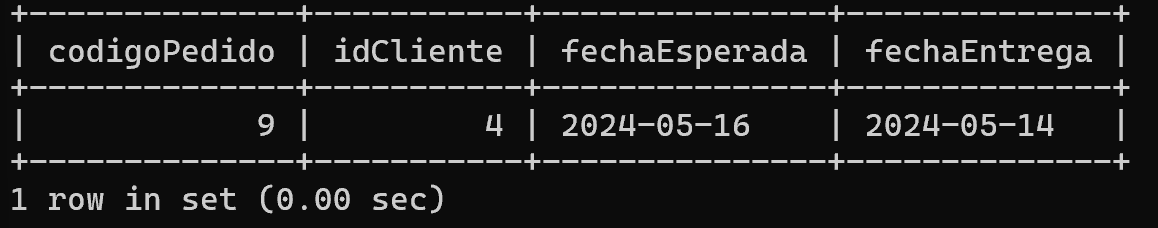
where fechaEntrega <=adddate(fechaEsperada,interval -2 day);



• Utilizando la funció n DATEDIFF de MySQL.

select id as codigoPedido, fkIdCliente as idCliente, fechaEsperada, fechaEntrega from pedido

where datediff(fechaEntrega,fechaEsperada) >2;



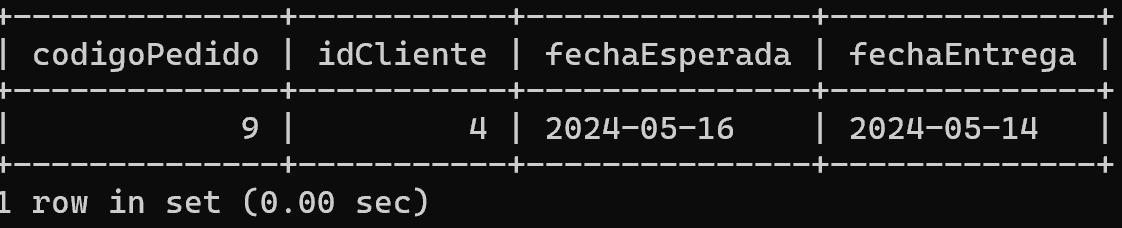
• ¿Sería posible resolver esta consulta utilizando el operador de suma + o

resta -?

SELECT id AS codigoPedido, fkIdCliente AS idCliente, fechaEsperada, fechaEntrega

FROM pedido

WHERE fechaEntrega <= fechaEsperada - INTERVAL 2 DAY;

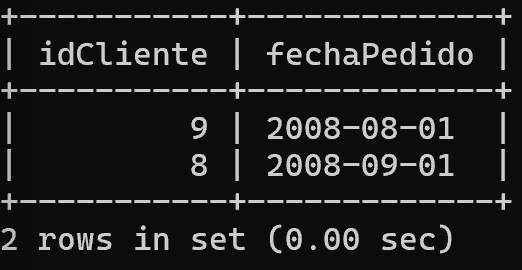


11. Devuelve un listado de todos los pedidos que fueron rechazados en 2009.

select p.fkIdCliente as idCliente, p.fechaPedido from pedido p

join estado e on p.fkEstado=e.id

where p.fechaPedido like '2008%' and e.id=2;



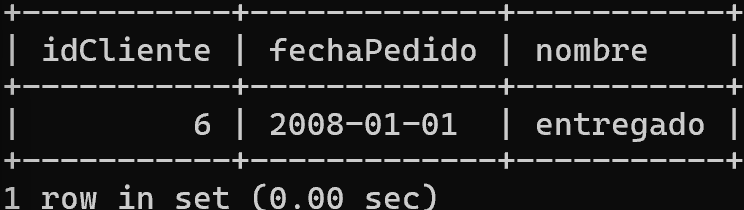
12. Devuelve un listado de todos los pedidos que han sido entregados en el

mes de enero de cualquier año.

select p.fkIdCliente as idCliente, p.fechaPedido, e.nombre from pedido p

join estado e on p.fkEstado=e.id

where e.id=1 and month(fechaEntrega)=01;



13. Devuelve un listado con todos los pagos que se realizaron en el

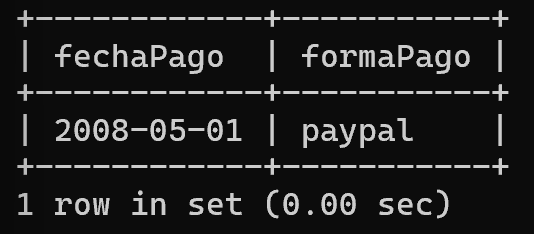
año 2008 mediante Paypal. Ordene el resultado de mayor a menor.

SELECT p.fechaPago, f.nombre as formaPago

FROM pago p

JOIN formaPago f ON f.id = p.fkFormaPago

WHERE YEAR(p.fechaPago) = 2008 AND f.nombre = 'paypal';



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 nombre from formaPago;



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

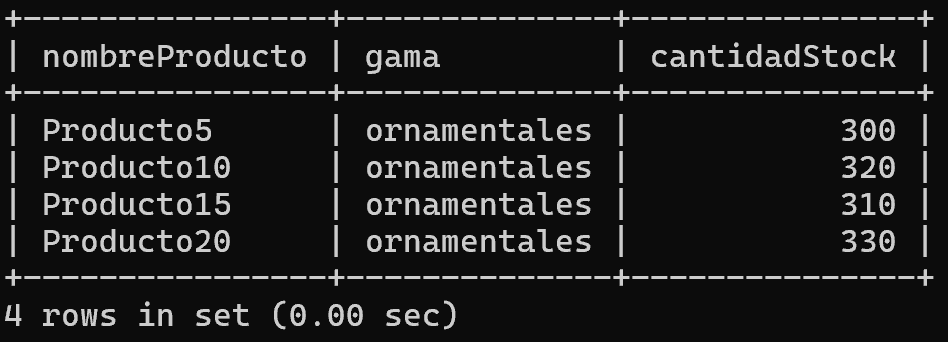
deberá estar ordenado por su precio de venta, mostrando en primer lugar

los de mayor precio.

select p.nombre as nombreProducto, g.gama, p.cantidadStock from producto p

join gamaproducto g on g.id=p.fkIdgama

where g.id=5 and p.cantidadStock>100;



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 c.nombreCliente, ci.nombre

FROM cliente c

JOIN ciudad ci ON ci.id = c.fkIdCiudad

WHERE c.fkIdCiudad = 6 AND (c.fkCodigoEmpleado = 11 OR c.fkCodigoEmpleado = 30);

**Consultas multitabla (Composición interna)**

Resuelva todas las consultas utilizando la sintaxis de SQL1 y SQL2. Las consultas con

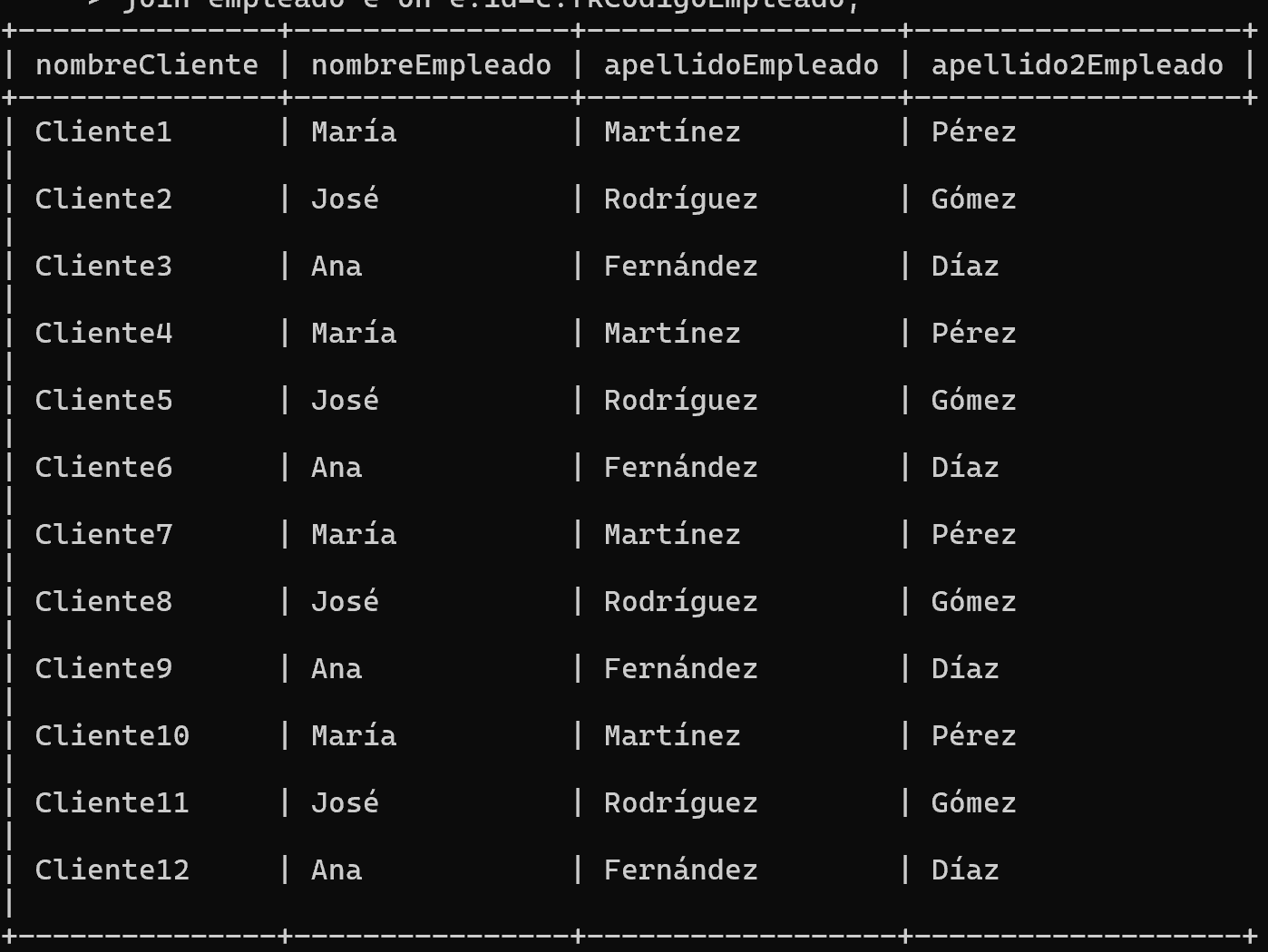
sintaxis de SQL2 se deben resolver con INNER JOIN y NATURAL JOIN.

1. Obtén un listado con el nombre de cada cliente y el nombre y apellido de su

representante de ventas.

select c.nombreCliente, e.nombre as nombreEmpleado,e.apellido1 as apellidoEmpleado,e.apellido2 as apellido2Empleado from cliente c

join empleado e on e.id=c.fkCodigoEmpleado;



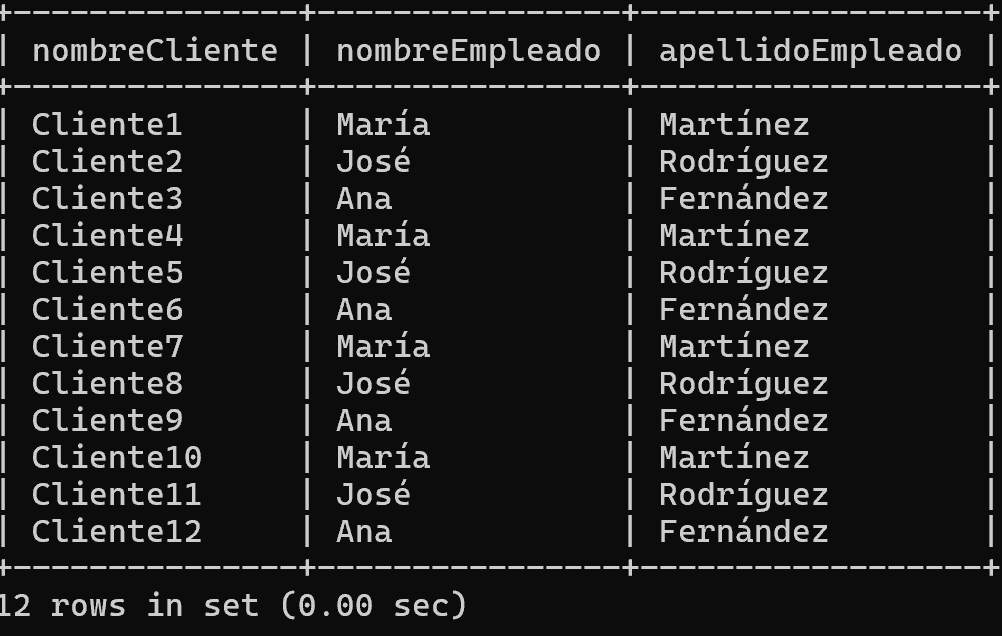
2. Muestra el nombre de los clientes que hayan realizado pagos junto con el

nombre de sus representantes de ventas.

select c.nombreCliente, e.nombre as nombreEmpleado, e.apellido1 as apellidoEmpleado from cliente c

right join pago p on p.fkIdCliente=c.id

join empleado e on e.id=c.fkCodigoEmpleado;



3. Muestra el nombre de los clientes que no hayan realizado pagos junto con

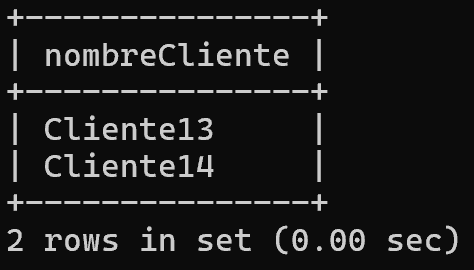
el nombre de sus representantes de ventas.

select c.nombreCliente from cliente c

left join pago p on p.fkIdCliente=c.id

join empleado e on e.id=c.fkCodigoEmpleado

where p.id is NULL;



4. 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 c.nombreCliente, e.nombre as nombreEmpleado, e.apellido1 as apellidoEmpleado, ci.nombre as ciudad from cliente c

right join pago p on p.fkIdCliente=c.id

join empleado e on e.id=c.fkCodigoEmpleado

join oficina o on o.id=e.fkIdOficina

join ciudad ci on ci.id=o.fkIdCiudad;



5. 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 c.nombreCliente, e.nombre as nombreEmpleado, e.apellido1 as apellidoEmpleado, ci.nombre as ciudad from cliente c

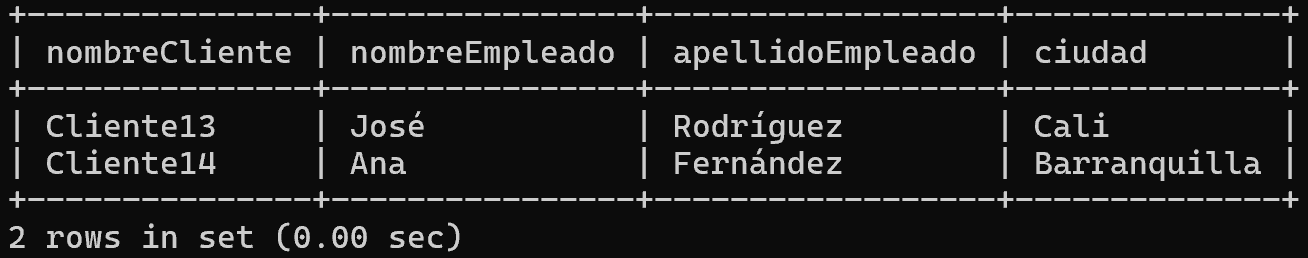
left join pago p on p.fkIdCliente=c.id

join empleado e on e.id=c.fkCodigoEmpleado

join oficina o on o.id=e.fkIdOficina

join ciudad ci on ci.id=o.fkIdCiudad

where p.id is null;



6. Lista la dirección de las oficinas que tengan clientes en Fuenlabrada.

select o.nombre from oficina o

join ciudad c on c.id=o.fkIdCiudad

where c.nombre = 'Fuenlabrada';

7. 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 as nombreEmpleado, e.apellido1 as apellidoEmpleado, ci.nombre as ciudad from cliente c

join empleado e on e.id=c.fkCodigoEmpleado

join oficina o on o.id=e.fkIdOficina

join ciudad ci on ci.id=o.fkIdCiudad;



8. Devuelve un listado con el nombre de los empleados junto con el nombre

de sus jefes.

SELECT

e1.nombre AS nombreEmpleado,

e1.apellido1 AS apellidoEmpleado,

e1.apellido2 AS apellido2Empleado,

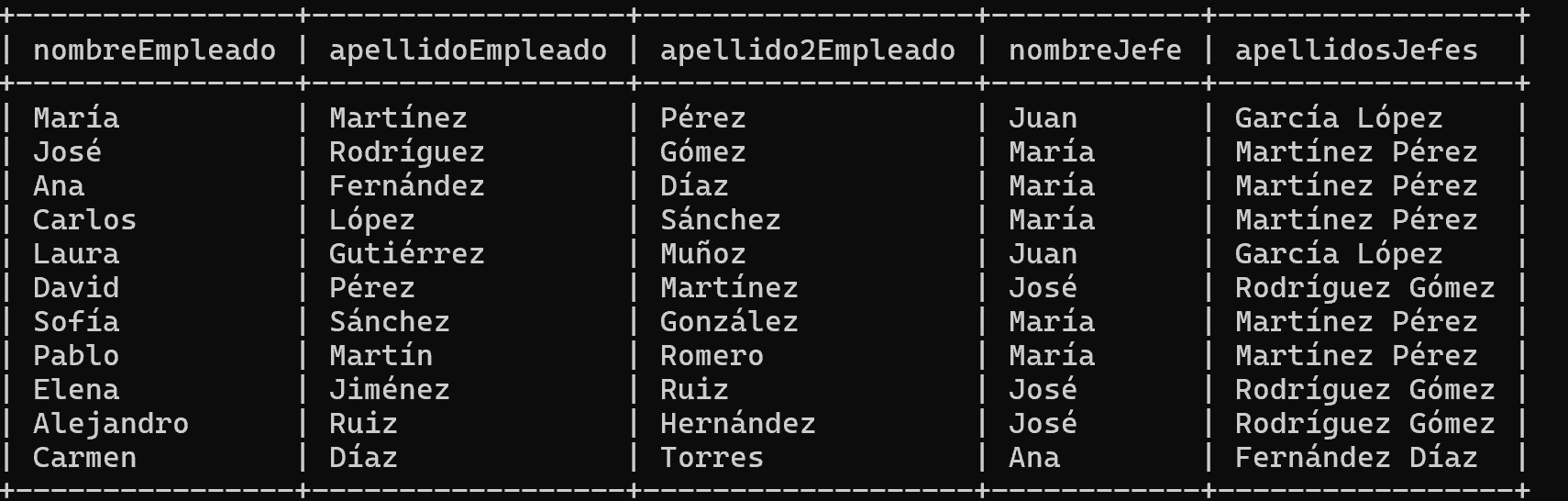
e2.nombre AS nombreJefe,

CONCAT(e2.apellido1, ' ', e2.apellido2) AS apellidosJefes

FROM empleado e1

JOIN empleado e2 ON e1.fkCodigoJefe = e2.id

WHERE e1.fkCodigoJefe IS NOT NULL;



9. Devuelve un listado que muestre el nombre de cada empleados, el nombre

de su jefe y el nombre del jefe de sus jefe.

SELECT

e1.nombre AS nombreEmpleado,

e1.apellido1 AS apellidoEmpleado,

e1.apellido2 AS apellido2Empleado,

e2.nombre AS nombreJefe,

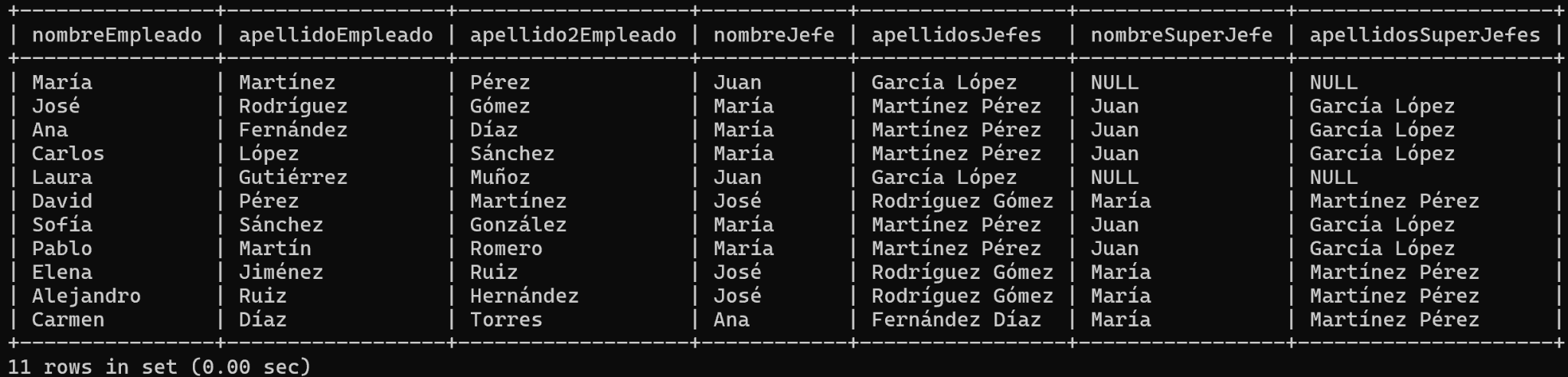
CONCAT(e2.apellido1, ' ', e2.apellido2) AS apellidosJefes

FROM empleado e1

JOIN empleado e2 ON e1.fkCodigoJefe = e2.id

left join empleado e3 on e3.id=e2.fkCodigoJefe

WHERE e1.fkCodigoJefe IS NOT NULL;



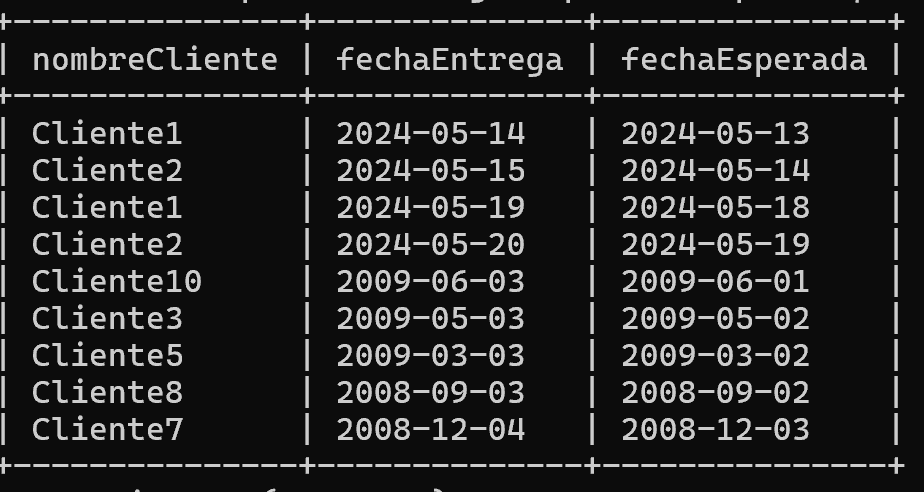
10. Devuelve el nombre de los clientes a los que no se les ha entregado a

tiempo un pedido.

select c.nombreCliente, p.fechaEntrega, p.fechaEsperada from cliente c

join pedido p on p.fkIdCliente=c.id

where p.fechaEntrega > p.fechaEsperada;



11. Devuelve un listado de las diferentes gamas de producto que ha comprado

cada cliente.

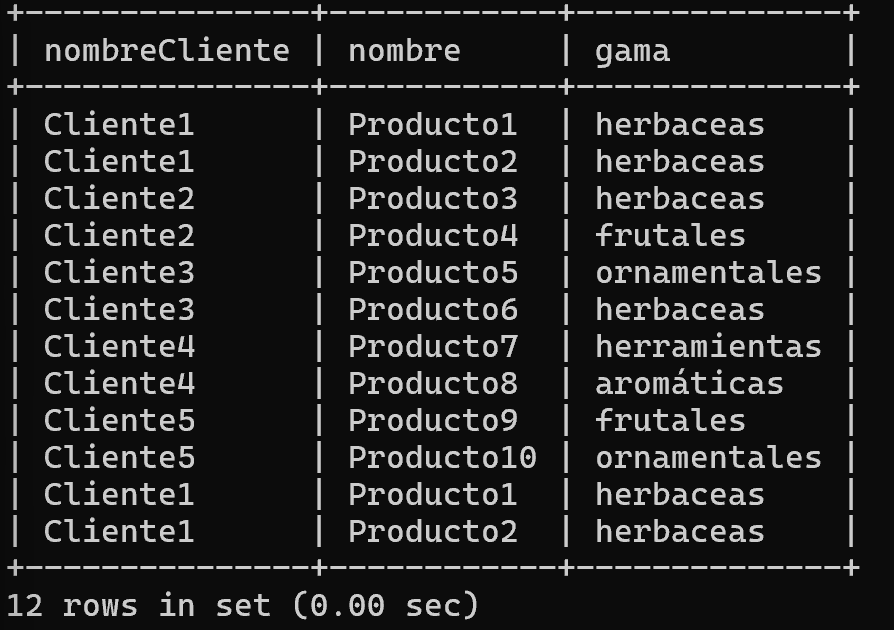
select c.nombreCliente, Pr.nombre, g.gama from cliente c

join pedido p on c.id=p.fkIdCliente

join detallepedido d p.id=d.fkIdpedido

join producto Pr on Pr.id=d.fkIdProducto

join gamaproducto g on g.id=Pr.fkIdgama;



**Consultas multitabla (Composición externa)**

Resuelva todas las consultas utilizando las cláusulas LEFT JOIN, RIGHT JOIN, NATURAL

LEFT JOIN y NATURAL RIGHT JOIN.

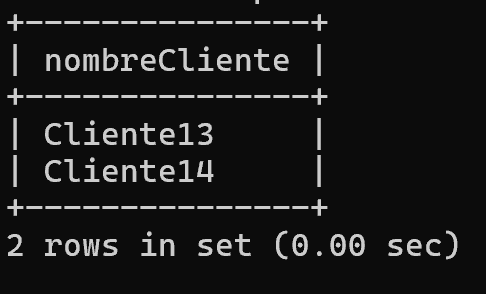
1. Devuelve un listado que muestre solamente los clientes que no han

realizado ningún pago.

select c.nombreCliente from cliente c

left join pago p on p.fkIdCliente=c.id

where p.id is NULL;



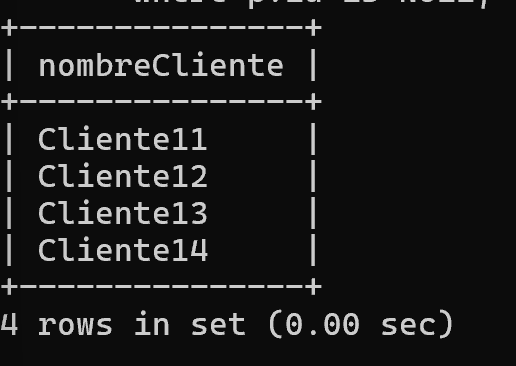
2. Devuelve un listado que muestre solamente los clientes que no han

realizado ningún pedido.

select c.nombreCliente from cliente c

left join pedido p on p.fkIdCliente=c.id

where p.id is NULL;



3. Devuelve un listado que muestre los clientes que no han realizado ningún

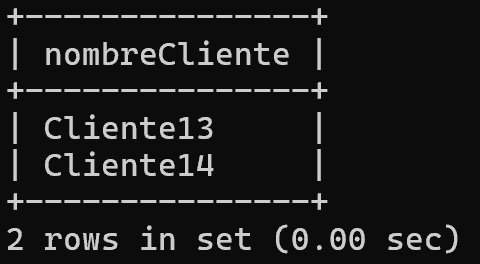
pago y los que no han realizado ningún pedido.

select c.nombreCliente from cliente c

left join pedido p on p.fkIdCliente=c.id

left join pago pa on pa.fkIdCliente=c.id

where p.id is NULL and pa.fkIdCliente is null;



4. Devuelve un listado que muestre solamente los empleados que no tienen

una oficina asociada.

select e.nombre from empleado e

where e.fkIdOficina is null;

5. Devuelve un listado que muestre solamente los empleados que no tienen un

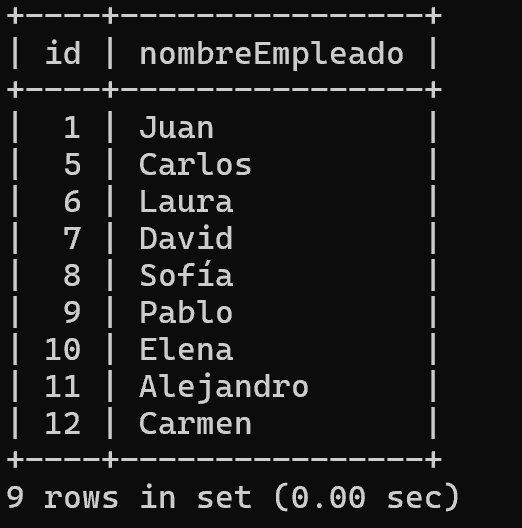
cliente asociado.

SELECT e.id, e.nombre AS nombreEmpleado

FROM empleado e

LEFT JOIN cliente c ON c.fkCodigoEmpleado = e.id

WHERE c.fkCodigoEmpleado IS NULL;



6. 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.id, e.nombre as nombreEmpleado, o.nombre as nombreOficina from empleado e

left join cliente c on c.fkCodigoEmpleado=e.id

join oficina o on o.id=e.fkIdOficina

where c.fkCodigoEmpleado is null;

7. Devuelve un listado que muestre los empleados que no tienen una oficina

asociada y los que no tienen un cliente asociado.

SELECT e.id, e.nombre, e.apellido1

FROM empleado e

LEFT JOIN cliente c ON c.fkCodigoEmpleado = e.id

LEFT JOIN oficina o ON o.id = e.fkIdOficina

WHERE c.fkCodigoEmpleado IS NULL AND e.fkIdOficina IS NULL;

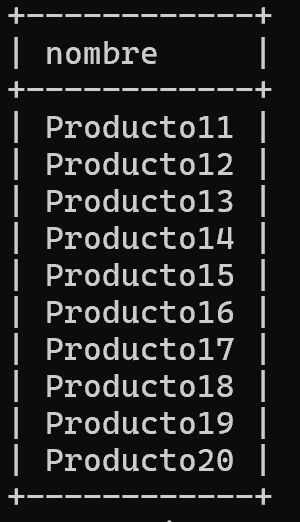
8. Devuelve un listado de los productos que nunca han aparecido en un

pedido.

select p.nombre from producto p

left join detallepedido d on d.fkIdproducto=p.id

where fkIdProducto is null;



9. 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

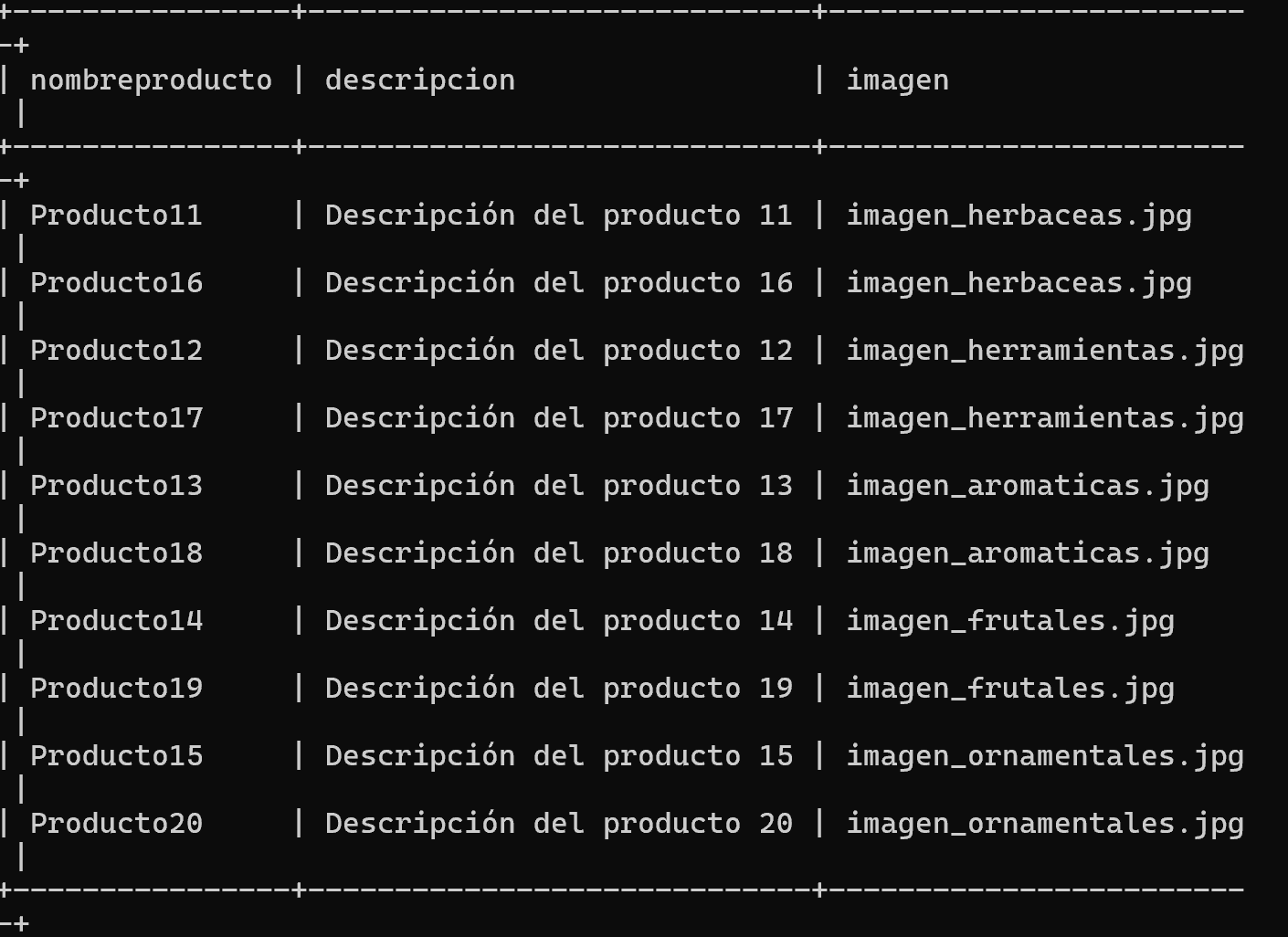
producto.

select p.nombre as nombreproducto, p.descripcion, g.imagen from producto p

left join detallepedido d on d.fkIdproducto=p.id

join gamaproducto g on g.id=p.fkIdgama

where fkIdProducto is null;



10. 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.

11. Devuelve un listado con los clientes que han realizado algún pedido pero no

han realizado ningún pago.

select c.nombreCliente from cliente c

right join pedido p on p.fkIdCliente=c.id

left join pago pa on pa.fkIdCliente=c.id

where p.fkIdCliente is null;

12. Devuelve un listado con los datos de los empleados que no tienen clientes

asociados y el nombre de su jefe asociado.

SELECT

e.id,

e.nombre AS nombreEmpleado,

e.apellido1 AS apellidoEmpleado,

e.apellido2 AS apellido2Empleado,

jefe.nombre AS nombreJefe,

jefe.apellido1 AS apellido1Jefe,

jefe.apellido2 AS apellido2Jefe

FROM

empleado e

LEFT JOIN

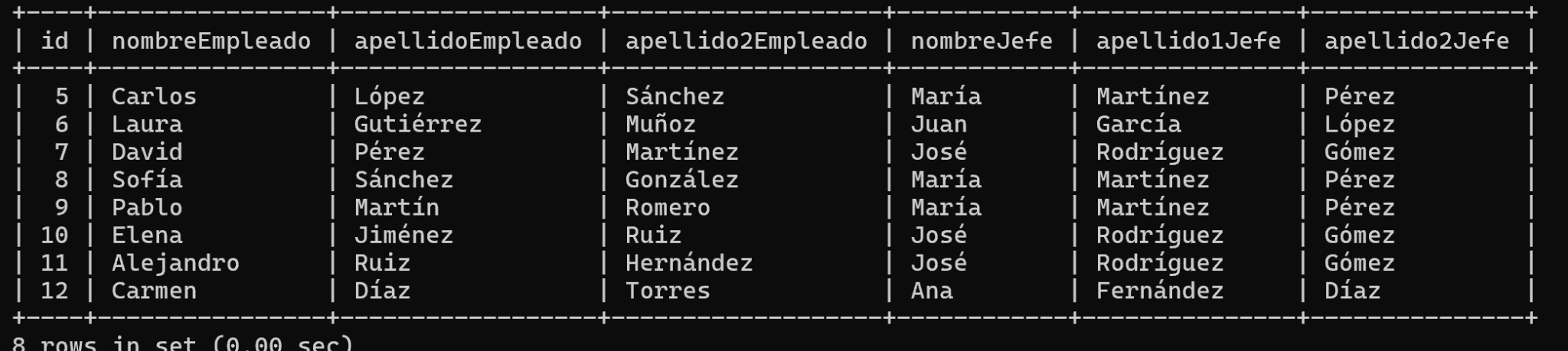
cliente c ON c.fkCodigoEmpleado = e.id

JOIN

empleado jefe ON e.fkCodigoJefe = jefe.id

WHERE

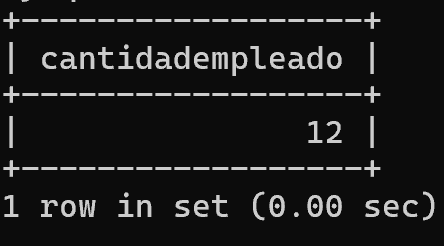
c.fkCodigoEmpleado IS NULL;



**Consultas resumen**

1. ¿Cuántos empleados hay en la compañía?

select count(id) as cantidadempleado from empleado;



2. ¿Cuántos clientes tiene cada país?

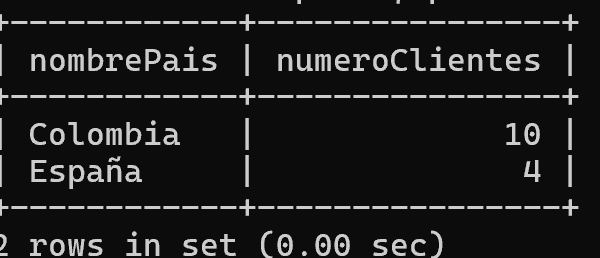
select pa.nombre AS nombrePais, COUNT(c.id) AS numeroClientes

join ciudad ci on ci.id=c.fkIdCiudad

join region r on r.id=ci.fkIdRegion

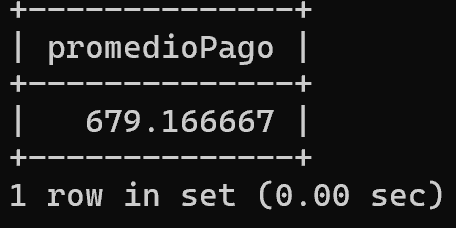
join pais pa on r.fkIdPais=pa.id

group by pa.id, pa.nombre;



3. ¿Cuál fue el pago medio en 2009?

select avg(total) as promedioPago from pago;



4. ¿Cuántos pedidos hay en cada estado? Ordena el resultado de forma

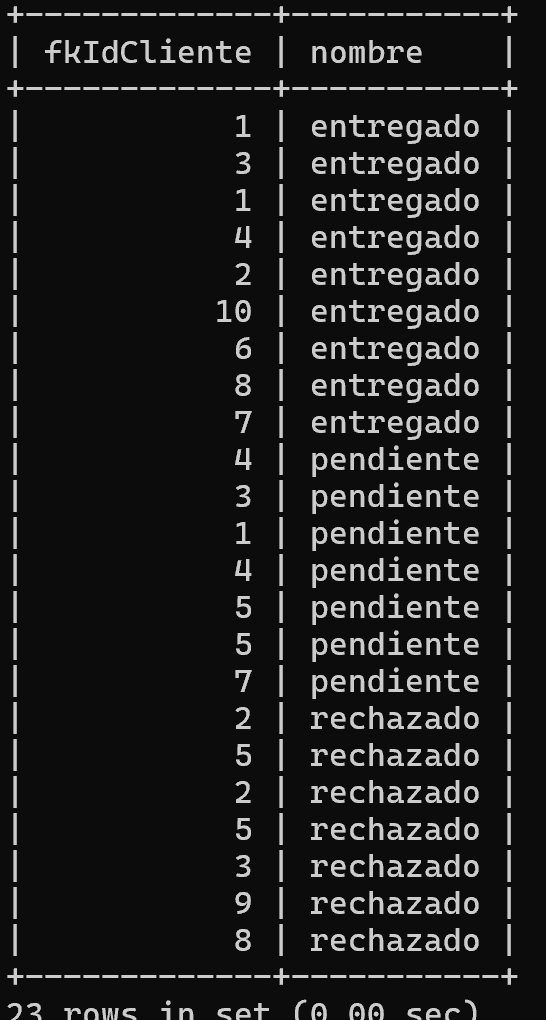
descendente por el número de pedidos.

SELECT p.fkIdCliente,e.nombre

FROM pedido p

JOIN estado e ON e.id = p.fkEstado

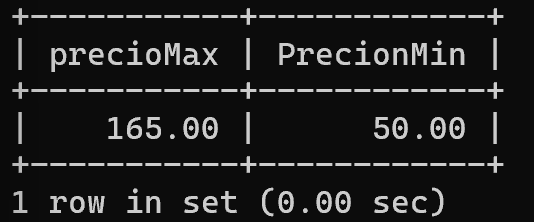
ORDER BY e.nombre ASC;



5. Calcula el precio de venta del producto más caro y más barato en una

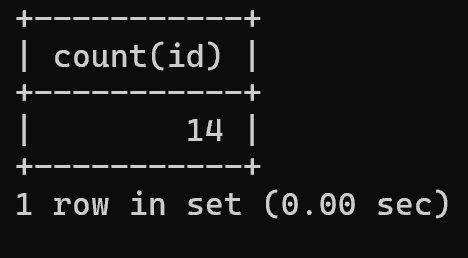
misma consulta.

select max(precioVenta) as precioMax,min(precioVenta) as PrecionMin from producto;



6. Calcula el número de clientes que tiene la empresa.

select count(id) from cliente;



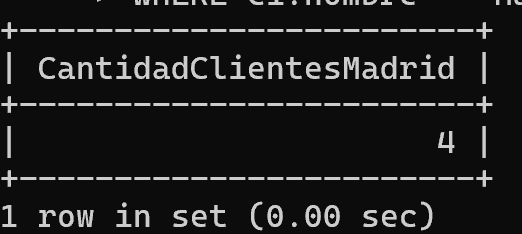
7. ¿Cuántos clientes existen con domicilio en la ciudad de Madrid?

SELECT COUNT(c.id) as CantidadClientesMadrid

FROM cliente c

JOIN ciudad ci ON ci.id = c.fkIdCiudad

WHERE ci.nombre = 'Madrid';



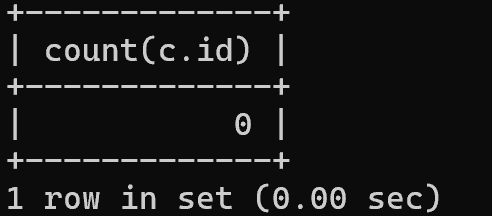
8. ¿Calcula cuántos clientes tiene cada una de las ciudades que empiezan

por M?

select count(c.id) from cliente c

join ciudad ci on ci.id=c.fkIdCiudad

where ci.nombre='m%';



9. Devuelve el nombre de los representantes de ventas y el número de clientes

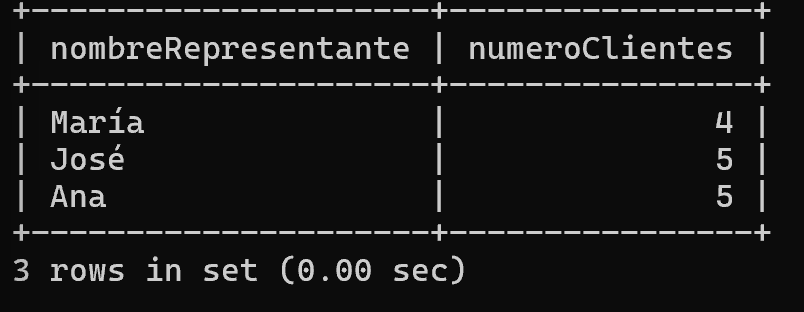
al que atiende cada uno.

SELECT e.nombre AS nombreRepresentante, COUNT(c.id) AS numeroClientes

FROM cliente c

JOIN empleado e ON e.id = c.fkCodigoEmpleado

GROUP BY e.id, e.nombre;

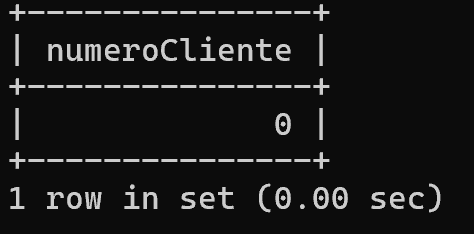


10. Calcula el número de clientes que no tiene asignado representante de

ventas.

select count(id) as numeroCliente from cliente

where fkCodigoEmpleado is null;



11. 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.nombreCliente,

c.nombreContacto,

c.apellidocontacto,

MAX(p.fechaPago) AS fechaUltimoPago,

MIN(p.fechaPago) AS fechaPrimerPago

FROM cliente c

JOIN pago p ON p.fkIdCliente = c.id

GROUP BY c.id, c.nombreCliente, c.nombreContacto, c.apellidocontacto;



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

pedidos.

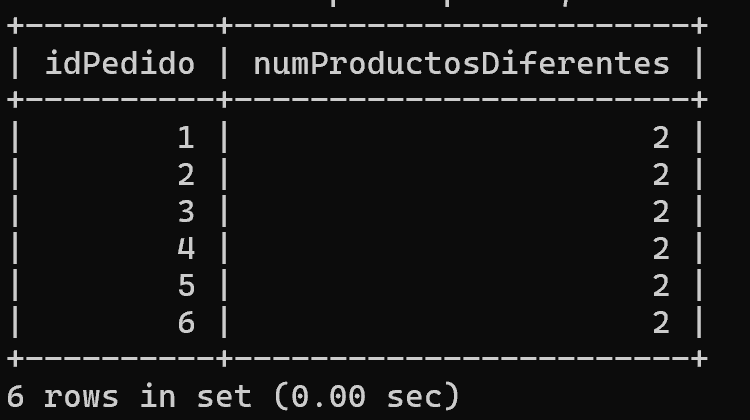
SELECT

dp.fkIdpedido AS idPedido,

COUNT(DISTINCT dp.fkIdProducto) AS numProductosDiferentes

FROM detallePedido dp

GROUP BY dp.fkIdpedido;



13. Calcula la suma de la cantidad total de todos los productos que aparecen en

cada uno de los pedidos.

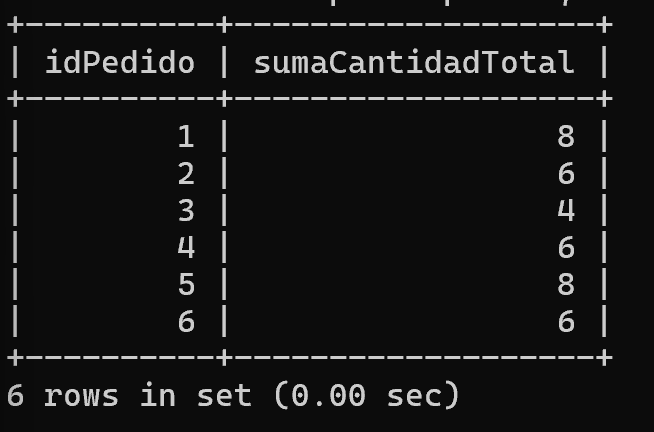
SELECT

dp.fkIdpedido AS idPedido,

SUM(dp.cantidad) AS sumaCantidadTotal

FROM detallePedido dp

GROUP BY dp.fkIdpedido;



14. Devuelve un listado de los 3 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

p.nombre AS nombreProducto,

SUM(d.cantidad) AS totalUnidades

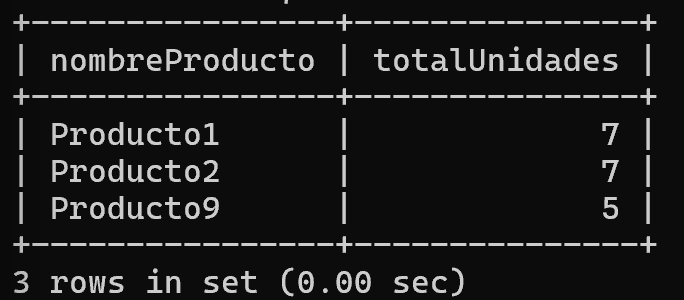
FROM detallepedido d

JOIN producto p ON d.fkIdProducto = p.id

GROUP BY p.id, p.nombre

ORDER BY totalUnidades DESC

LIMIT 3;



15. 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

SUM(base\_imponible) AS base\_imponible\_total,

SUM(iva) AS iva\_total,

SUM(total\_facturado) AS total\_facturado

FROM (

SELECT

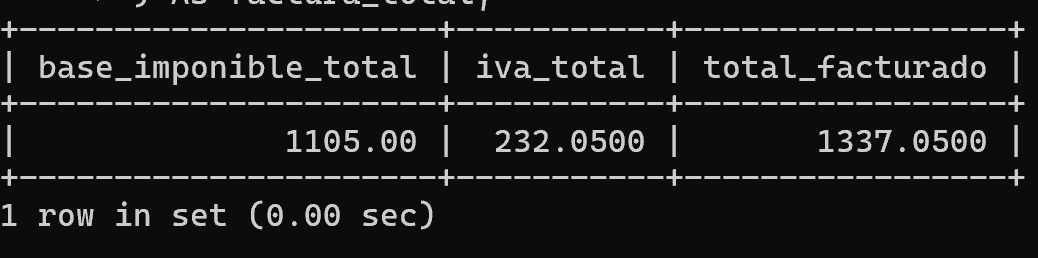
SUM(precioUnidad \* cantidad) AS base\_imponible,

SUM(precioUnidad \* cantidad) \* 0.21 AS iva,

SUM(precioUnidad \* cantidad) + (SUM(precioUnidad \* cantidad) \* 0.21) AS total\_facturado

FROM detallePedido

) AS factura\_total;



16. La misma información que en la pregunta anterior, pero agrupada por

código de producto.

SELECT

fkIdProducto AS codigo\_producto,

SUM(base\_imponible) AS base\_imponible\_total,

SUM(iva) AS iva\_total,

SUM(total\_facturado) AS total\_facturado

FROM (

SELECT

fkIdProducto,

SUM(precioUnidad \* cantidad) AS base\_imponible,

SUM(precioUnidad \* cantidad) \* 0.21 AS iva,

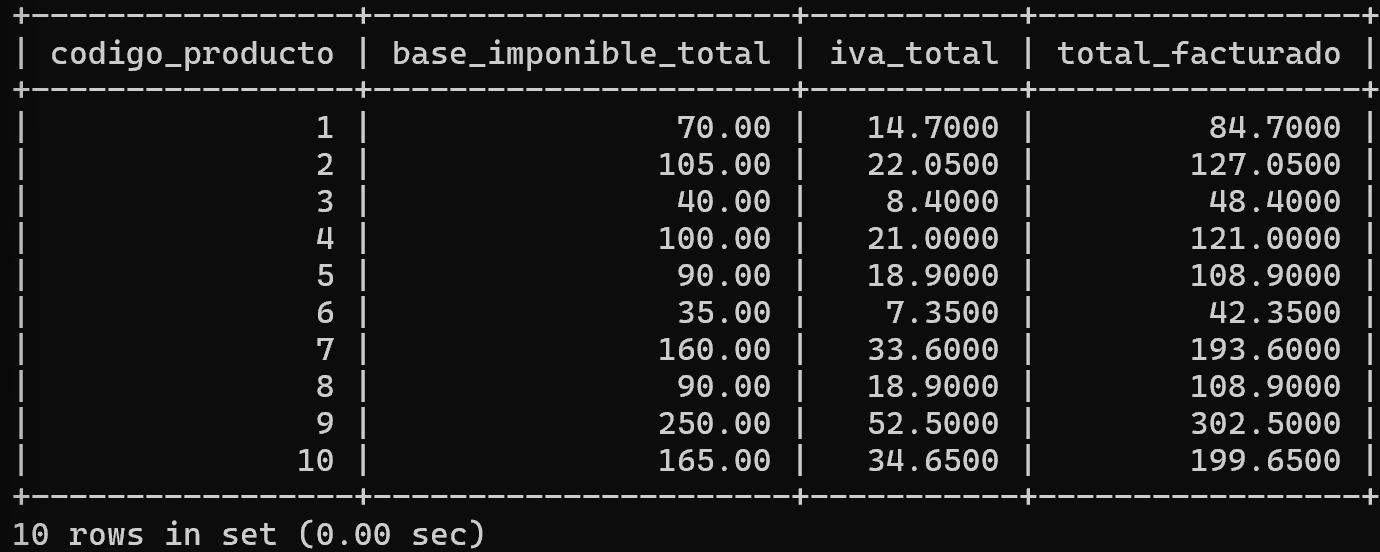
SUM(precioUnidad \* cantidad) + (SUM(precioUnidad \* cantidad) \* 0.21) AS total\_facturado

FROM detallePedido

GROUP BY fkIdProducto

) AS factura\_total\_por\_producto

GROUP BY fkIdProducto;



17. La misma información que en la pregunta anterior, pero agrupada por

código de producto filtrada por los códigos que empiecen por OR.

SELECT

fkIdProducto AS codigo\_producto,

SUM(base\_imponible) AS base\_imponible\_total,

SUM(iva) AS iva\_total,

SUM(total\_facturado) AS total\_facturado

FROM (

SELECT

fkIdProducto,

SUM(precioUnidad \* cantidad) AS base\_imponible,

SUM(precioUnidad \* cantidad) \* 0.21 AS iva,

SUM(precioUnidad \* cantidad) + (SUM(precioUnidad \* cantidad) \* 0.21) AS total\_facturado

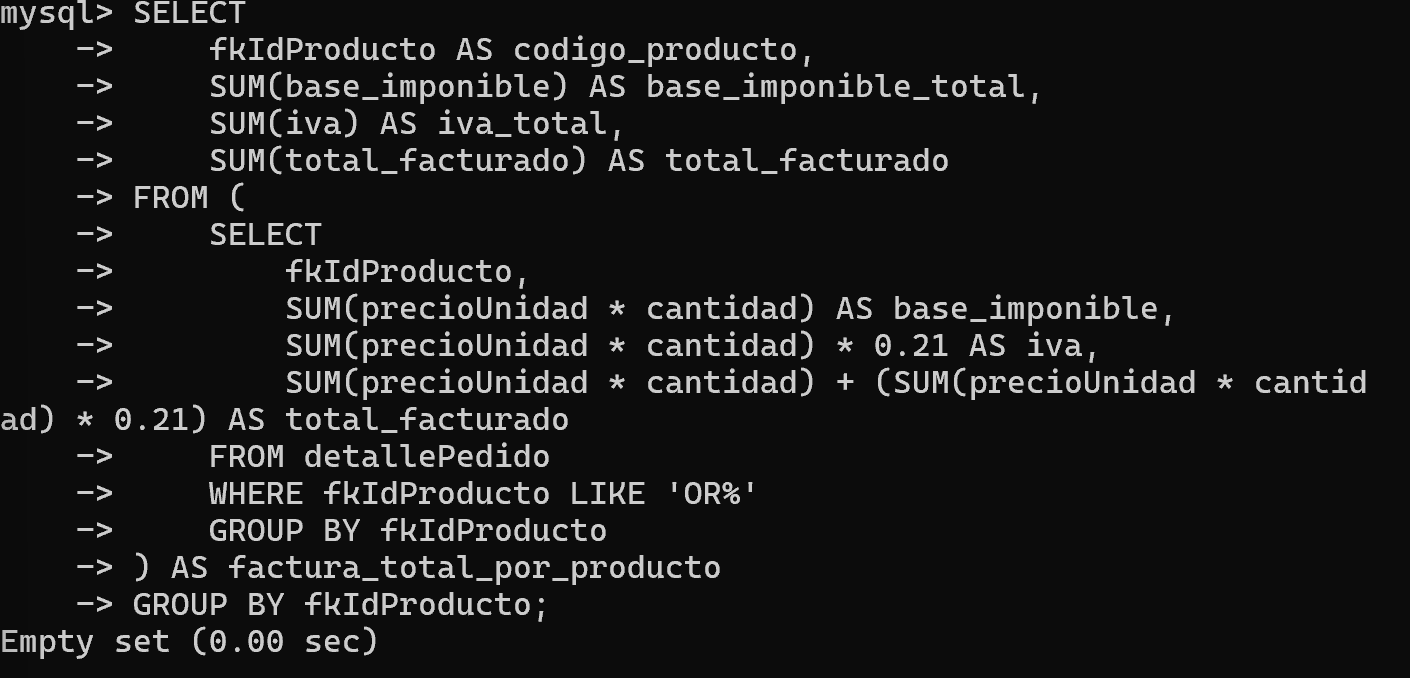
FROM detallePedido

WHERE fkIdProducto LIKE 'OR%'

GROUP BY fkIdProducto

) AS factura\_total\_por\_producto

GROUP BY fkIdProducto;



18. 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

nombre\_producto,

unidades\_vendidas,

total\_facturado,

total\_con\_iva

FROM (

SELECT

p.nombre AS nombre\_producto,

SUM(dp.cantidad) AS unidades\_vendidas,

SUM(dp.precioUnidad \* dp.cantidad) AS total\_facturado,

SUM(dp.precioUnidad \* dp.cantidad) \* 1.21 AS total\_con\_iva

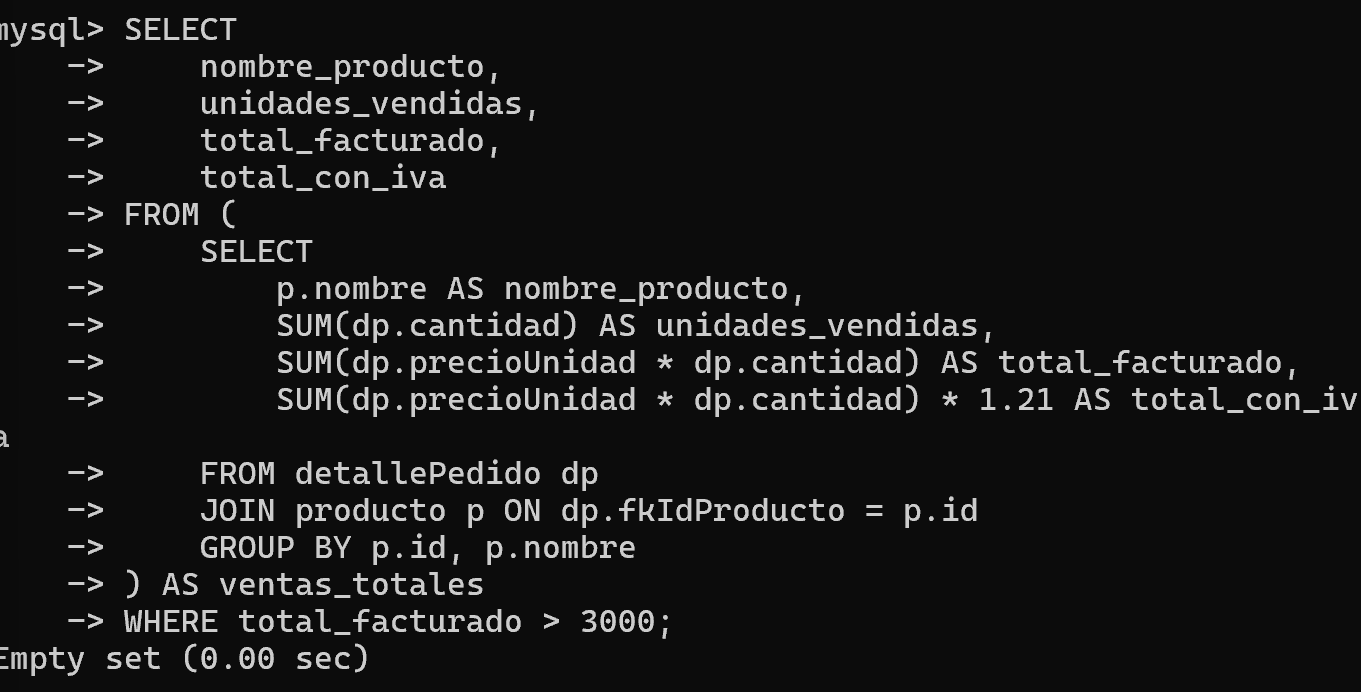
FROM detallePedido dp

JOIN producto p ON dp.fkIdProducto = p.id

GROUP BY p.id, p.nombre

) AS ventas\_totales

WHERE total\_facturado > 3000;



19. Muestre la suma total de todos los pagos que se realizaron para cada uno

de los años que aparecen en la tabla pagos.

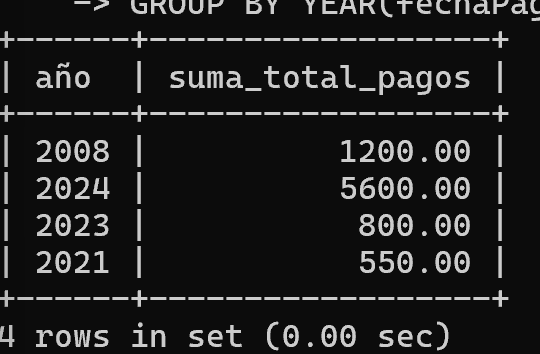
SELECT

YEAR(fechaPago) AS año,

SUM(total) AS suma\_total\_pagos

FROM pago

GROUP BY YEAR(fechaPago);



**Subconsultas**

**Con operadores básicos de comparación**

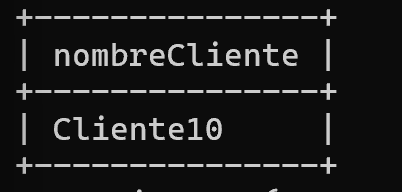
1. Devuelve el nombre del cliente con mayor límite de crédito.

SELECT nombreCliente

FROM cliente

ORDER BY limiteCredito DESC

LIMIT 1;

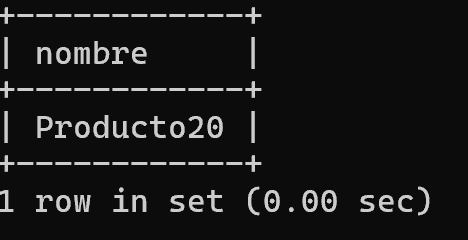


2. Devuelve el nombre del producto que tenga el precio de venta más caro.

select nombre from producto

order by precioVenta desc

limit 1;



3. 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 p.nombre AS nombre\_producto

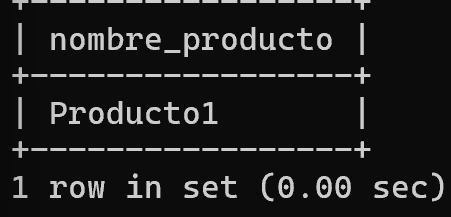
FROM detallePedido dp

JOIN producto p ON dp.fkIdProducto = p.id

GROUP BY dp.fkIdProducto, p.nombre

ORDER BY SUM(dp.cantidad) DESC

LIMIT 1;



4. Los clientes cuyo límite de crédito sea mayor que los pagos que haya

realizado. (Sin utilizar INNER JOIN).

SELECT nombreCliente

FROM cliente

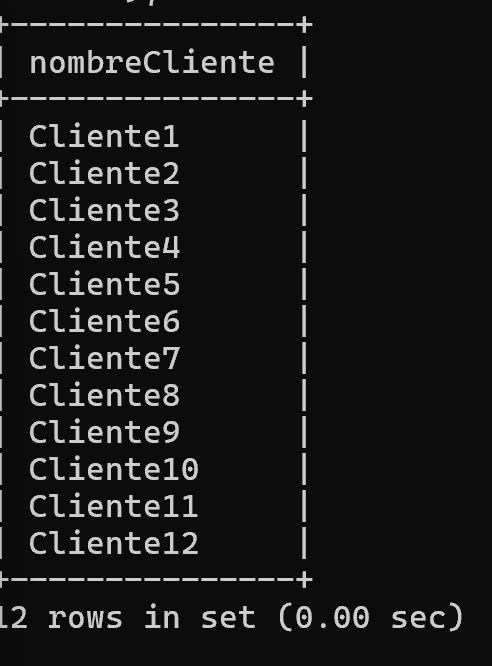
WHERE limiteCredito > (

SELECT SUM(total)

FROM pago

WHERE fkIdCliente = cliente.id

);

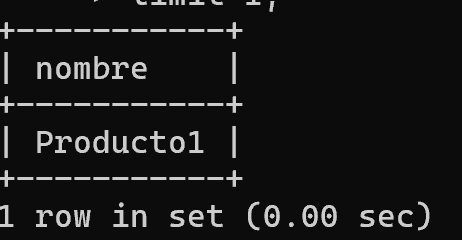


5. Devuelve el producto que más unidades tiene en stock.

select nombre from producto

order by cantidadStock asc

limit 1;

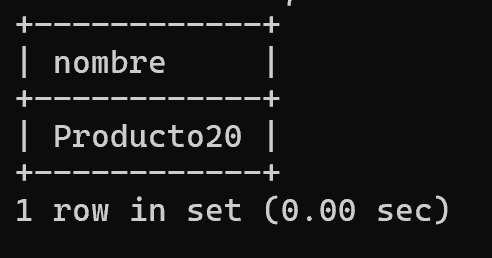


6. Devuelve el producto que menos unidades tiene en stock.

select nombre from producto

order by cantidadStock desc

limit 1;

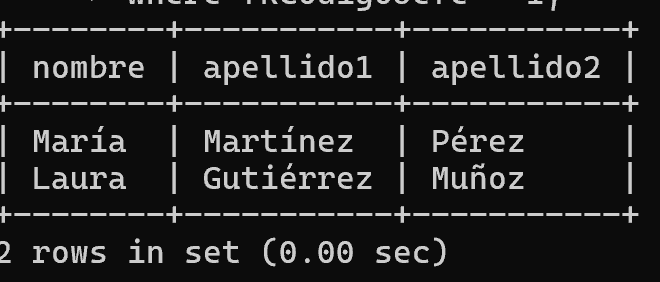


7. Devuelve el nombre, los apellidos y el email de los empleados que están a

cargo de Juan Garcia .

select nombre, apellido1,apellido2 from empleado

where fkCodigoJefe = 1;



**Subconsultas con ALL y ANY**

8. Devuelve el nombre del cliente con mayor límite de crédito.

SELECT nombreCliente

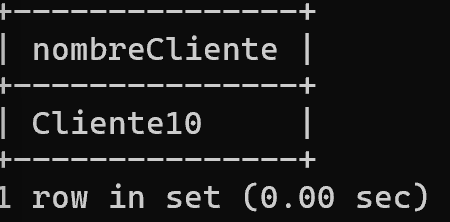
FROM cliente

WHERE limiteCredito >= ALL (

SELECT limiteCredito

FROM cliente

);



9. Devuelve el nombre del producto que tenga el precio de venta más caro.

SELECT nombre

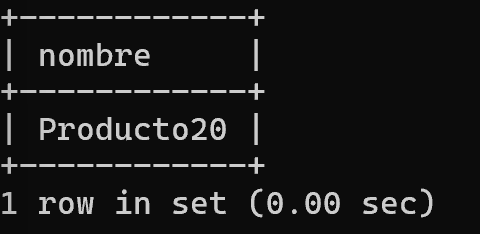
FROM producto

WHERE precioVenta >= ALL (

SELECT precioVenta

FROM producto

);



10. Devuelve el producto que menos unidades tiene en stock.

SELECT nombre

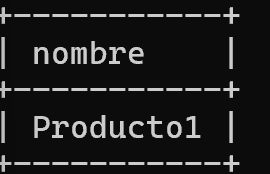
FROM producto

WHERE cantidadStock <= ALL (

SELECT cantidadStock

FROM producto

);



**Subconsultas con IN y NOT IN**

11. Devuelve el nombre, apellido1 y cargo de los empleados que no

representen a ningún cliente.

SELECT e.nombre, e.apellido1

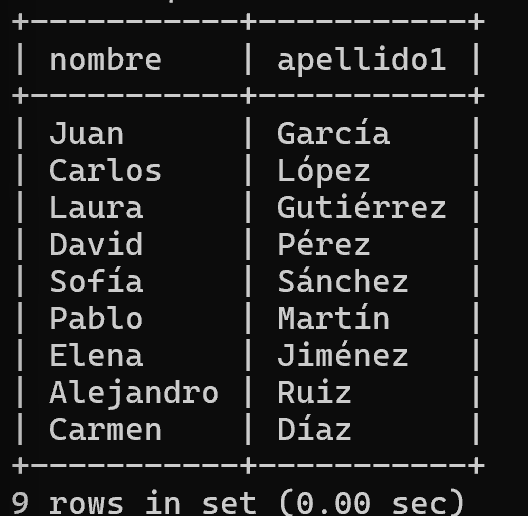
FROM empleado e

WHERE e.id NOT IN (

SELECT fkCodigoEmpleado

FROM cliente

);



12. Devuelve un listado que muestre solamente los clientes que no han

realizado ningún pago.

SELECT nombreCliente

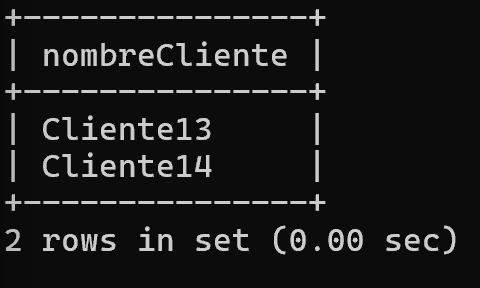
FROM cliente

WHERE id NOT IN (

SELECT fkIdCliente

FROM pago

);



13. Devuelve un listado que muestre solamente los clientes que sí han realizado

algún pago.

SELECT nombreCliente

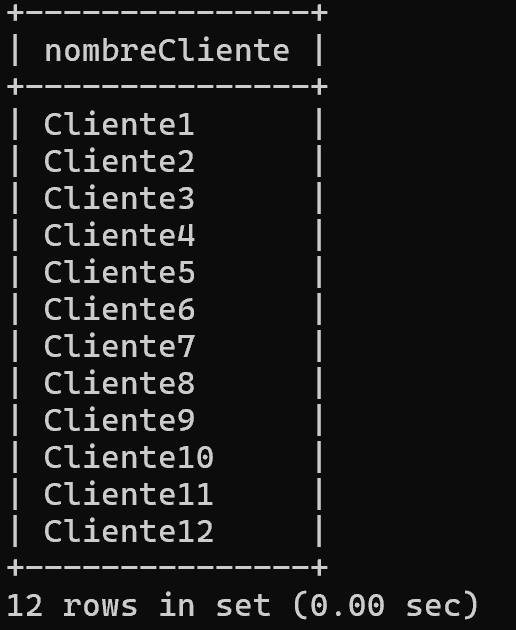
FROM cliente

WHERE id IN (

SELECT fkIdCliente

FROM pago

);



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

pedido.

SELECT nombre

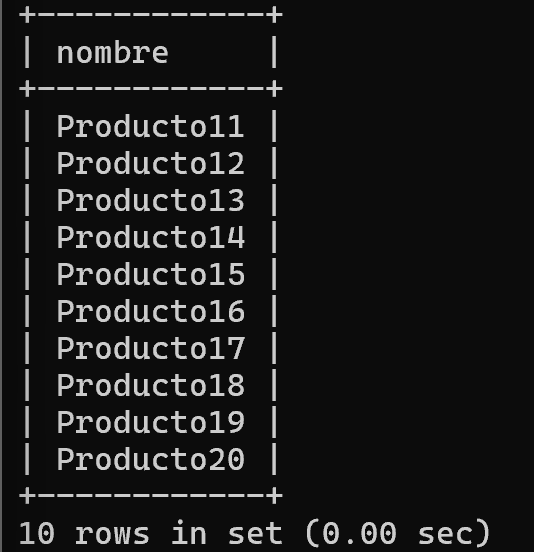
FROM producto

WHERE id NOT IN (

SELECT DISTINCT fkIdProducto

FROM detallePedido

);



15. Devuelve el nombre, apellidos, puesto y teléfono de la oficina de aquellos

empleados que no sean representante de ventas de ningún cliente.

SELECT e.nombre, e.apellido1, e.apellido2, p.puesto, t.telefono

FROM empleado e

JOIN puesto p ON e.fkPuesto = p.id

JOIN oficina o ON e.fkIdOficina = o.id

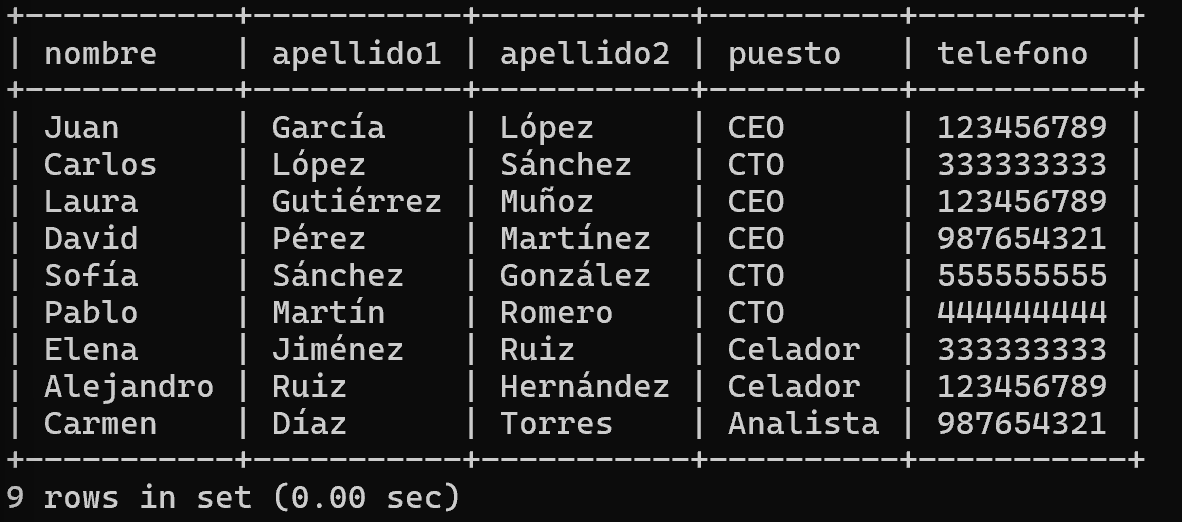
JOIN telefonoOficina t ON o.id = t.fkIdOficina

WHERE e.id NOT IN (

SELECT DISTINCT fkCodigoEmpleado

FROM cliente

);



16. 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.

17. Devuelve un listado con los clientes que han realizado algún pedido pero no

han realizado ningún pago.

SELECT nombreCliente

FROM cliente

WHERE id IN (

SELECT DISTINCT fkIdCliente

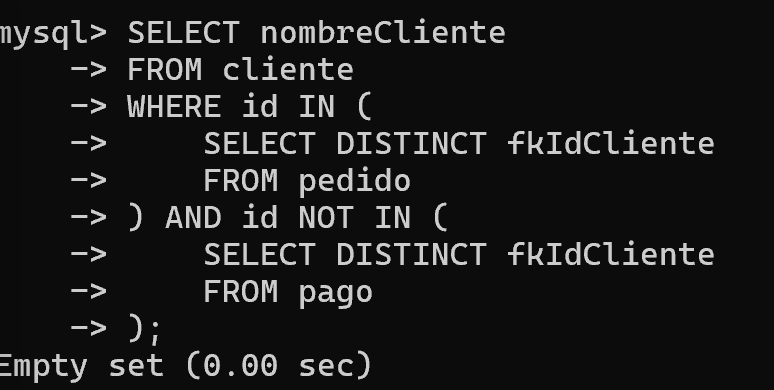
FROM pedido

) AND id NOT IN (

SELECT DISTINCT fkIdCliente

FROM pago

);



**Subconsultas con EXISTS y NOT EXISTS**

18. Devuelve un listado que muestre solamente los clientes que no han

realizado ningún pago.

SELECT nombreCliente

FROM cliente c

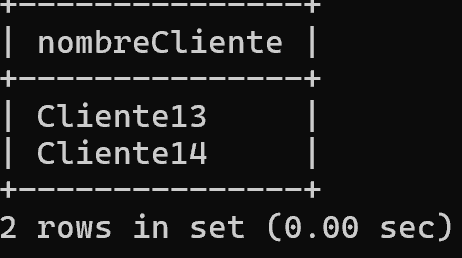
WHERE NOT EXISTS (

SELECT 1

FROM pago p

WHERE p.fkIdCliente = c.id

);



19. Devuelve un listado que muestre solamente los clientes que sí han realizado

algún pago.

SELECT nombreCliente

FROM cliente c

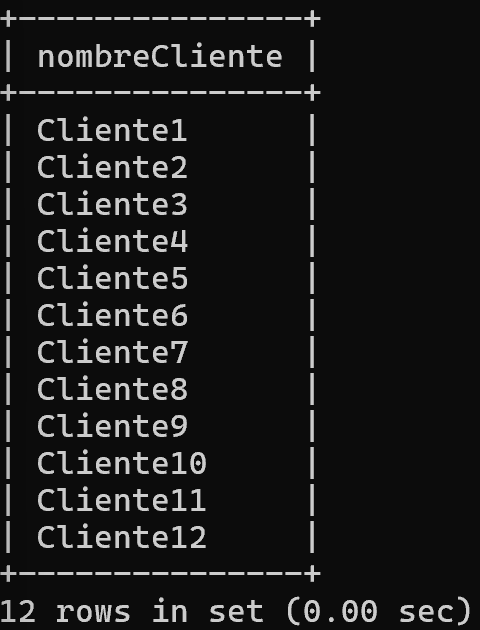
WHERE EXISTS (

SELECT 1

FROM pago p

WHERE p.fkIdCliente = c.id

);



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

pedido.

SELECT nombre

FROM producto p

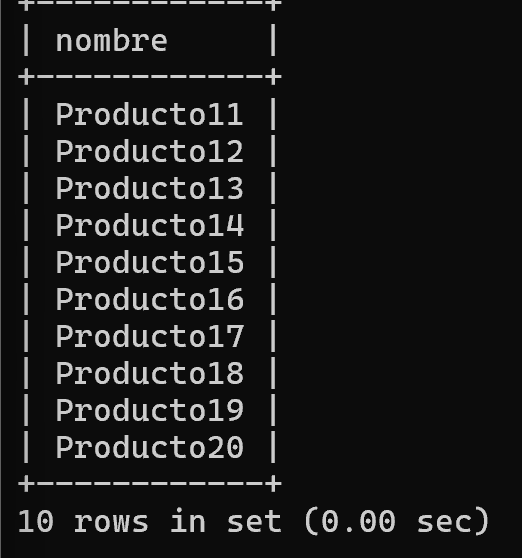
WHERE NOT EXISTS (

SELECT 1

FROM detallePedido dp

WHERE dp.fkIdProducto = p.id

);



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

alguna vez.

SELECT DISTINCT nombre

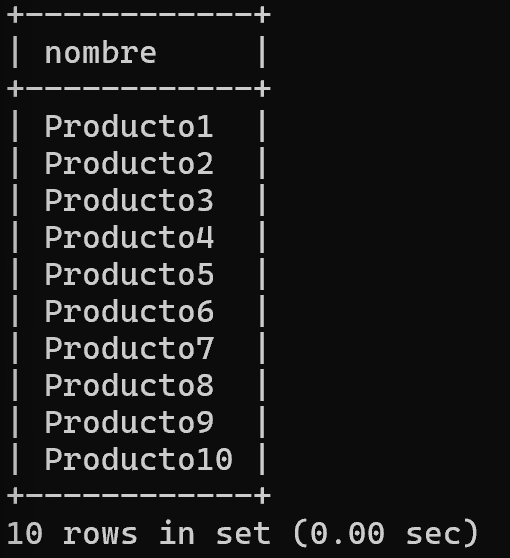
FROM producto

WHERE id IN (

SELECT DISTINCT fkIdProducto

FROM detallePedido

);



Subconsultas correlacionadas

Consultas variadas

1. Devuelve el listado de clientes indicando el nombre del cliente y cuántos

pedidos ha realizado. Tenga en cuenta que pueden existir clientes que no

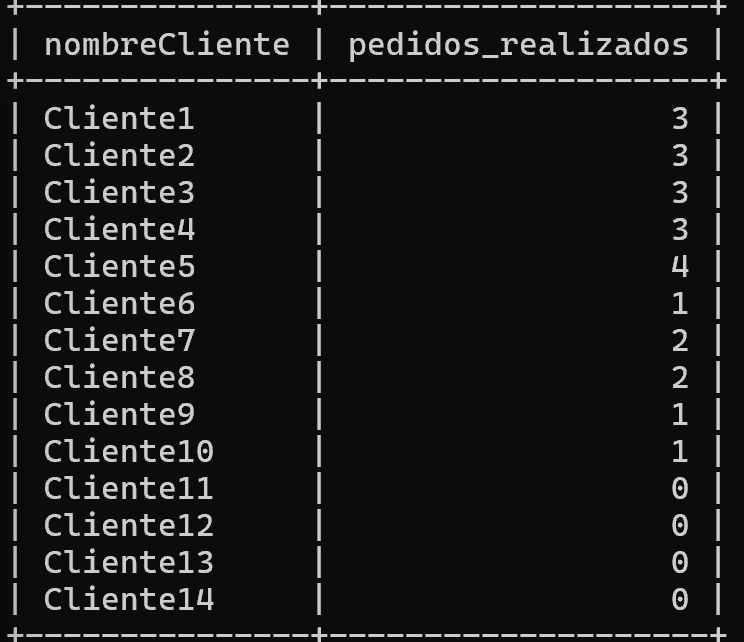
han realizado ningún pedido.

SELECT c.nombreCliente, COUNT(p.id) AS pedidos\_realizados

FROM cliente c

LEFT JOIN pedido p ON c.id = p.fkIdCliente

GROUP BY c.id, c.nombreCliente;



2. 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.

3. Devuelve el nombre de los clientes que hayan hecho pedidos en 2008

ordenados alfabéticamente de menor a mayor.

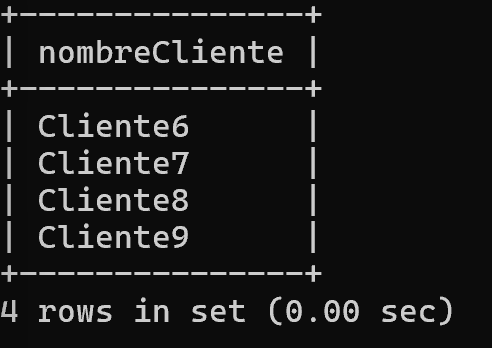
SELECT DISTINCT c.nombreCliente

FROM cliente c

JOIN pedido p ON c.id = p.fkIdCliente

WHERE YEAR(p.fechaPedido) = 2008

ORDER BY c.nombreCliente ASC;



4. 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 c.nombreCliente,

CONCAT(e.nombre, ' ', e.apellido1) AS nombre\_representante,

tof.telefono AS telefono\_oficina

FROM cliente c

LEFT JOIN empleado e ON c.fkCodigoEmpleado = e.id

LEFT JOIN telefonoOficina tof ON e.fkIdOficina = tof.fkIdOficina

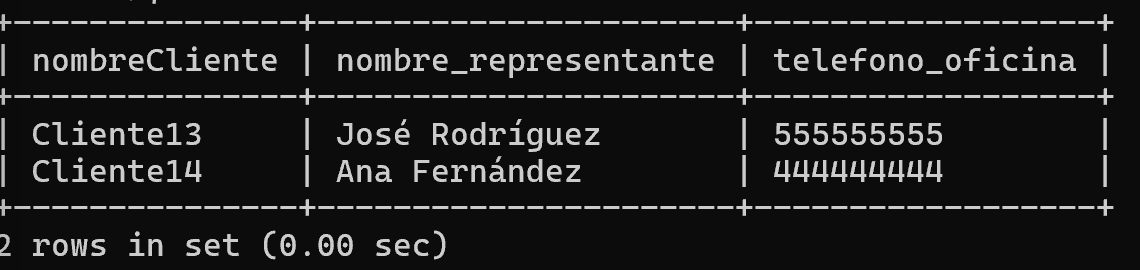
WHERE NOT EXISTS (

SELECT 1

FROM pago p

WHERE p.fkIdCliente = c.id

);



5. 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 c.nombreCliente,

CONCAT(e.nombre, ' ', e.apellido1) AS nombre\_representante,

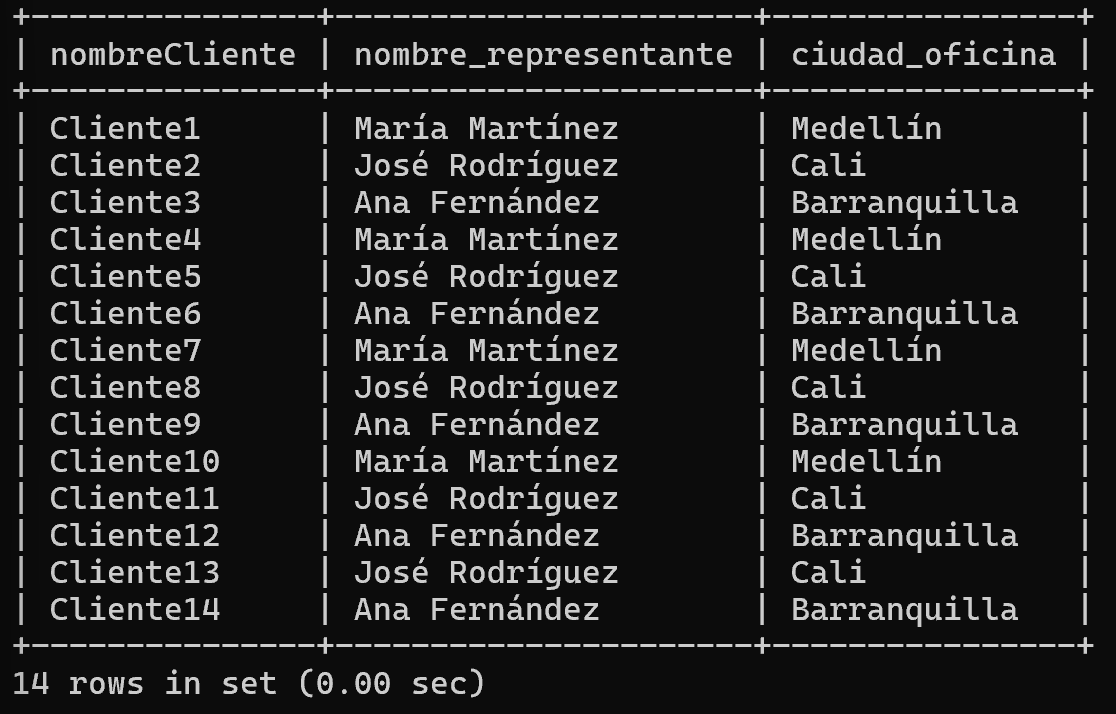
ci.nombre AS ciudad\_oficina

FROM cliente c

JOIN empleado e ON c.fkCodigoEmpleado = e.id

JOIN oficina o ON e.fkIdOficina = o.id

JOIN ciudad ci ON o.fkIdCiudad = ci.id;



6. Devuelve el nombre, apellidos, puesto y teléfono de la oficina de aquellos

empleados que no sean representante de ventas de ningún cliente.

SELECT e.nombre, e.apellido1, e.apellido2, p.puesto, tof.telefono

FROM empleado e

JOIN puesto p ON e.fkPuesto = p.id

JOIN oficina o ON e.fkIdOficina = o.id

JOIN telefonoOficina tof ON o.id = tof.fkIdOficina

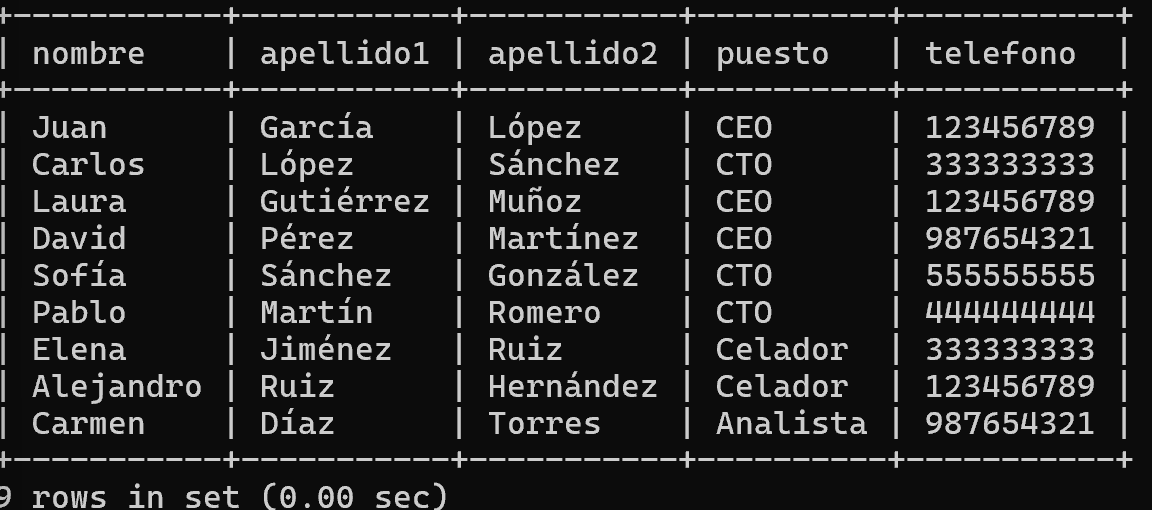
WHERE NOT EXISTS (

SELECT 1

FROM cliente c

WHERE c.fkCodigoEmpleado = e.id

);



7. Devuelve un listado indicando todas las ciudades donde hay oficinas y el

número de empleados que tiene.

SELECT ci.nombre AS ciudad, COUNT(e.id) AS numero\_empleados

FROM empleado e

JOIN oficina o ON e.fkIdOficina = o.id

JOIN ciudad ci ON o.fkIdCiudad = ci.id

GROUP BY ci.id, ci.nombre;

