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.nombre\_cliente,e.nombre,e.apellido1 **FROM** cliente c **INNER** **JOIN**

empleado e **ON** c.codigo\_empleado\_rep\_ventas=e.codigo\_empleado;



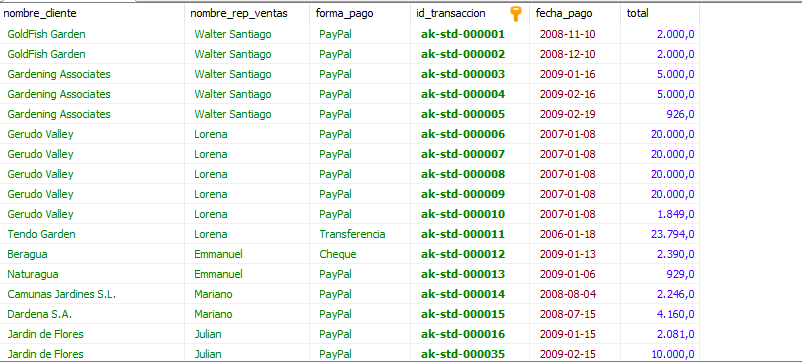
2. Muestra el nombre de los clientes que hayan realizado pagos junto con el

nombre de sus representantes de ventas.

**SELECT** c.nombre\_cliente, e.nombre **AS** nombre\_rep\_ventas,p.forma\_pago,p.id\_transaccion, p.fecha\_pago,p.total **FROM** cliente c

**INNER** **JOIN** empleado e **ON** c.codigo\_empleado\_rep\_ventas = e.codigo\_empleado **INNER** **JOIN** pago p

**ON** c.codigo\_cliente = p.codigo\_cliente;



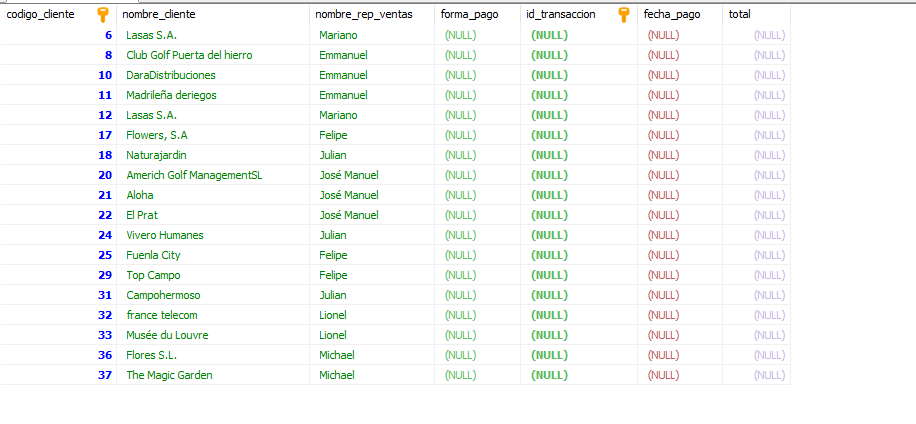
3. Muestra el nombre de los clientes que no hayan realizado pagos junto con el nombre

de sus representantes de ventas.

**SELECT** c.codigo\_cliente,c.nombre\_cliente, e.nombre **AS** nombre\_rep\_ventas,p.forma\_pago,p.id\_transaccion ,p.fecha\_pago,p.total

**FROM** cliente c **INNER** **JOIN** empleado e **ON** c.codigo\_empleado\_rep\_ventas = e.codigo\_empleado **LEFT** **JOIN** pago p

**ON** c.codigo\_cliente = p.codigo\_cliente **WHERE** p.codigo\_cliente **IS** **NULL**;



4. Devuelve el nombre de los clientes que han hecho pagos y el nombre de sus

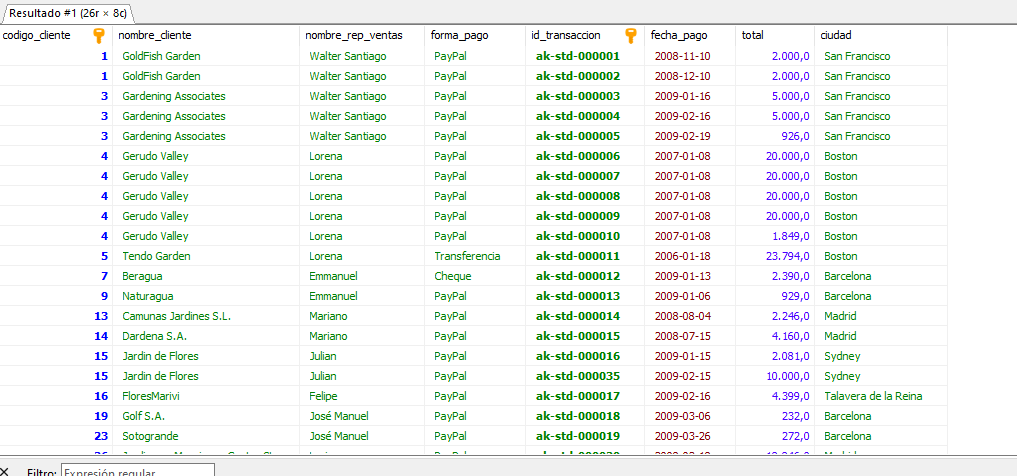
representantes junto con la ciudad de la oficina a la que pertenece el

representante.

**SELECT** c.codigo\_cliente,c.nombre\_cliente, e.nombre **AS** nombre\_rep\_ventas,p.forma\_pago,p.id\_transaccion, p.fecha\_pago,p.total ,uf.ciudad **FROM** cliente c

**INNER** **JOIN** empleado e **ON** c.codigo\_empleado\_rep\_ventas = e.codigo\_empleado **INNER** **JOIN** pago p

**ON** c.codigo\_cliente = p.codigo\_cliente **INNER** **JOIN** oficina uf **ON** e.codigo\_oficina=uf.codigo\_oficina;



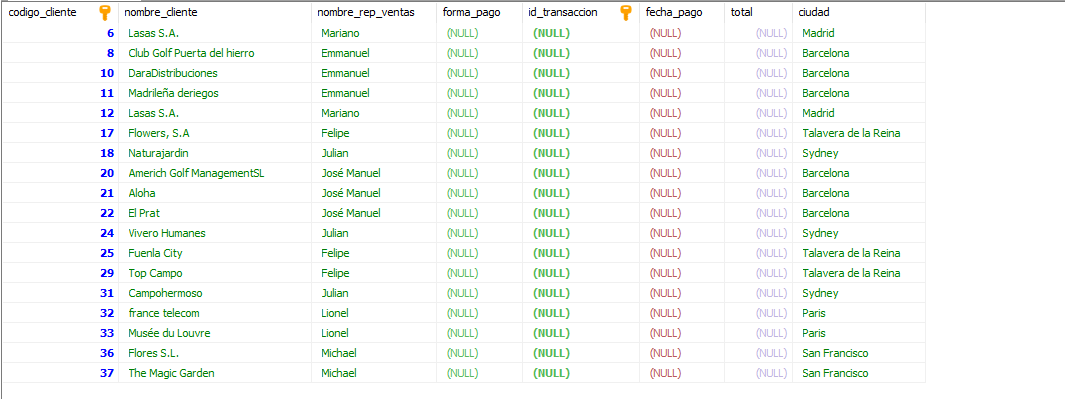
5. Devuelve el nombre de los clientes que no hayan hecho pagos y el nombre de sus

representantes junto con la ciudad de la oficina a la que pertenece el representante.

**SELECT** c.codigo\_cliente,c.nombre\_cliente, e.nombre **AS** nombre\_rep\_ventas,p.forma\_pago,p.id\_transaccion, p.fecha\_pago,p.total ,

uf.ciudad **FROM** cliente c **INNER** **JOIN** empleado e **ON** c.codigo\_empleado\_rep\_ventas = e.codigo\_empleado **LEFT** **JOIN** pago p

**ON** c.codigo\_cliente = p.codigo\_cliente **INNER** **JOIN** oficina uf **ON** e.codigo\_oficina=uf.codigo\_oficina **WHERE** p.codigo\_cliente **IS** **NULL** ;



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

**SELECT** **DISTINCT** o.linea\_direccion1, o.linea\_direccion2,c.ciudad **FROM** oficina o **INNER** **JOIN** empleado e **ON** o.codigo\_oficina = e.codigo\_oficina

**INNER** **JOIN** cliente c **ON** e.codigo\_empleado = c.codigo\_empleado\_rep\_ventas **WHERE** c.ciudad = 'Fuenlabrada';

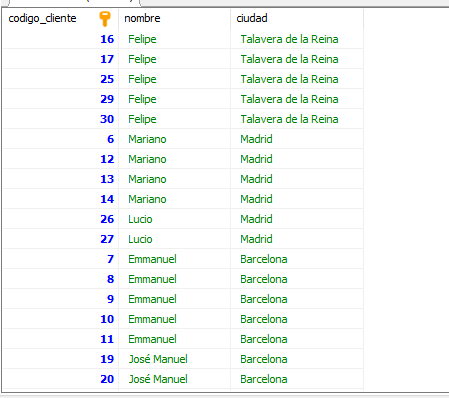


7. Devuelve el nombre de los clientes y el nombre de sus representantes junto con

la ciudad de la oficina a la que pertenece el representante.

**SELECT** c.codigo\_cliente,e.nombre ,o.ciudad **FROM** cliente c **INNER** **JOIN** empleado e **ON** c.codigo\_empleado\_rep\_ventas=e.codigo\_empleado

**INNER** **JOIN** oficina o **ON** e.codigo\_oficina=o.codigo\_oficina;



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

**SELECT** e.codigo\_empleado ,e.nombre **AS** "nombre empleado",j.nombre **AS** "nombre jefe"**FROM** empleado e **left** **JOIN** empleado j

**ON** e.codigo\_jefe=j.codigo\_empleado;



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** e.codigo\_empleado ,e.nombre **AS** "nombre empleado",j.nombre **AS** "nombre jefe",jj.nombre **AS**"jefe de su jefe"**FROM** empleado e **left** **JOIN** empleado j

**ON** e.codigo\_jefe=j.codigo\_empleado **LEFT** **JOIN** empleado jj **ON** j.codigo\_jefe=jj.codigo\_empleado;



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

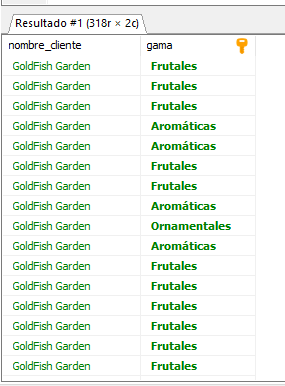
**SELECT** c.nombre\_cliente **FROM** cliente c **INNER** **JOIN** pedido p **ON** c.codigo\_cliente = p.codigo\_cliente

**WHERE** p.estado = 'Entregado' **AND** p.fecha\_entrega <= p.fecha\_esperada;

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

**SELECT** c.nombre\_cliente, gp.gama **FROM** cliente c **INNER** **JOIN** pedido p **ON** c.codigo\_cliente = p.codigo\_cliente **NATURAL** **JOIN**

detalle\_pedido dp **NATURAL** **JOIN** producto pr **INNER** **JOIN** gama\_producto gp **ON** pr.gama = gp.gama;



Consultas multitabla (Composición externa)

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

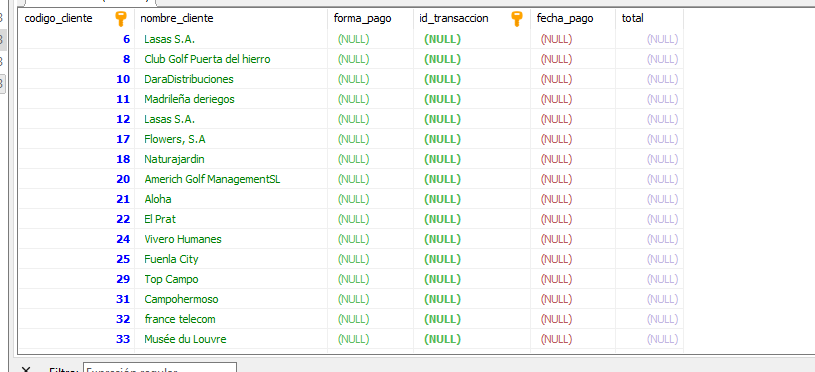
JOIN y NATURAL RIGHT JOIN.

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

ningún pago.

**SELECT** c.codigo\_cliente,c.nombre\_cliente,p.forma\_pago,p.id\_transaccion,p.fecha\_pago,p.total **FROM**

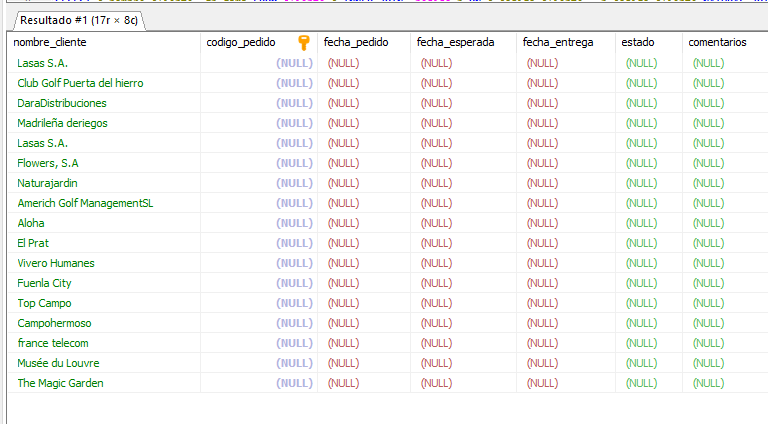
cliente c **LEFT** **JOIN** pago p **ON** c.codigo\_cliente = p.codigo\_cliente **WHERE** p.codigo\_cliente **IS** **NULL**;



13. Devuelve un listado que muestre solamente los clientes que no han realizado

ningún pedido.

**SELECT** c.nombre\_cliente,p.\* **FROM** cliente c **LEFT** **JOIN** pedido p **ON** c.codigo\_cliente = p.codigo\_cliente **WHERE** p.codigo\_pedido **IS** **NULL**;



14. Devuelve un listado que muestre los clientes que no han realizado ningún pago y

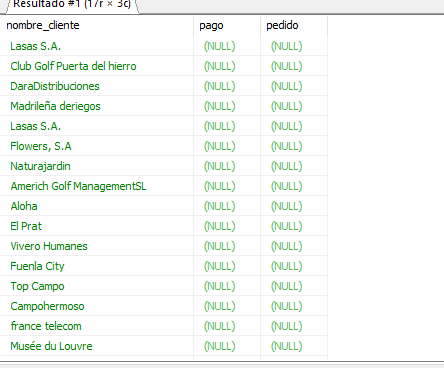
los que no han realizado ningún pedido.

**SELECT** c.nombre\_cliente ,p.forma\_pago **AS** "pago",pd.estado **as**"pedido"

**FROM** cliente c **LEFT** **JOIN** pago p **ON** c.codigo\_cliente = p.codigo\_cliente

**LEFT** **JOIN** pedido pd **ON** c.codigo\_cliente = pd.codigo\_cliente

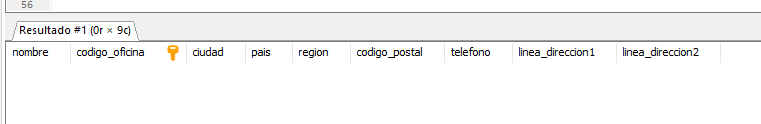
**WHERE** p.codigo\_cliente **IS** **NULL** **AND** pd.codigo\_pedido **IS** **NULL**;



15. Devuelve un listado que muestre solamente los empleados que no tienen una oficina asociada.

**SELECT** e.nombre,o.\* **FROM** empleado e **LEFT** **JOIN** oficina o **ON** e.codigo\_oficina=o.codigo\_oficina

**WHERE** o.codigo\_oficina **IS** **NULL**;

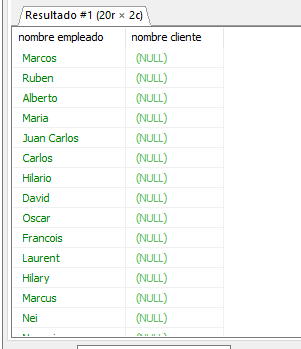


16. Devuelve un listado que muestre solamente los empleados que no tienen un

cliente asociado.

**SELECT** e.nombre **AS** "nombre empleado" ,c.nombre\_cliente **AS**"nombre cliente"

**FROM** empleado e **LEFT** **JOIN** cliente c **ON** e.codigo\_empleado = c.codigo\_empleado\_rep\_ventas **WHERE** c.codigo\_cliente **IS** **NULL**;



17. 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.nombre **AS** "nombre empleado" ,c.nombre\_cliente **AS**"nombre cliente",o.\*

**FROM** empleado e **LEFT** **JOIN** cliente c **ON** e.codigo\_empleado = c.codigo\_empleado\_rep\_ventas **LEFT** **JOIN** oficina o **ON** e.codigo\_oficina=o.codigo\_oficina

**WHERE** c.codigo\_cliente **IS** **NULL**;



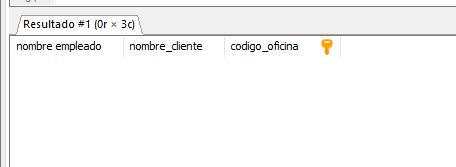
18. Devuelve un listado que muestre los empleados que no tienen una oficina

asociada y los que no tienen un cliente asociado.

**SELECT** e.nombre **AS**"nombre empleado",c.nombre\_cliente ,o.codigo\_oficina **FROM** empleado e **LEFT** **JOIN** oficina o **ON** e.codigo\_oficina=o.codigo\_oficina

**LEFT** **JOIN** cliente c **ON** e.codigo\_empleado=c.codigo\_empleado\_rep\_ventas

**WHERE** o.codigo\_oficina **IS** **NULL** **AND** c.codigo\_cliente **is** **null**;



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

**SELECT** pr.nombre **AS**"nombre\_producto",pd.estado **AS** " pedido" **FROM** producto pr **LEFT** **JOIN** detalle\_pedido dll **ON** pr.codigo\_producto=dll.codigo\_producto

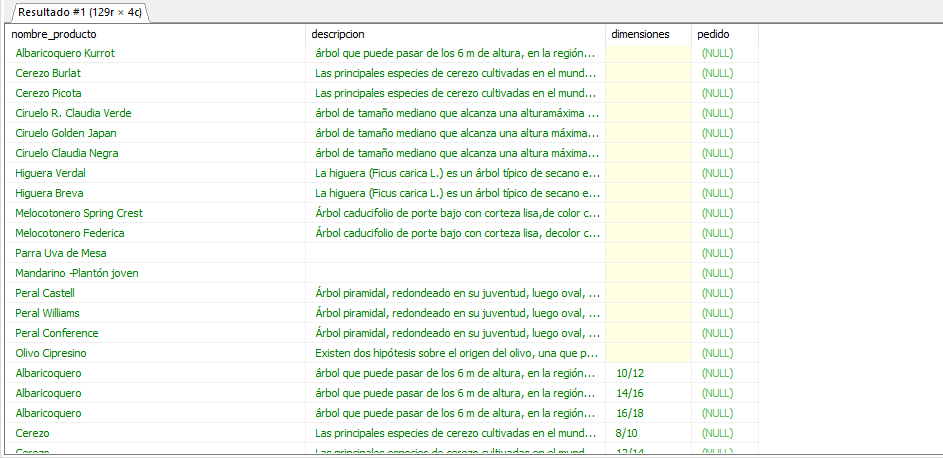
**LEFT** **JOIN** pedido pd **ON** dll.codigo\_pedido=pd.codigo\_pedido **WHERE** pd.codigo\_pedido **IS** **NULL**;



20. 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** pr.nombre **AS**"nombre\_producto",pr.descripcion,pr.dimensiones,pd.estado **AS** " pedido" **FROM** producto pr **LEFT** **JOIN** detalle\_pedido dll **ON** pr.codigo\_producto=dll.codigo\_producto

**LEFT** **JOIN** pedido pd **ON** dll.codigo\_pedido=pd.codigo\_pedido **WHERE** pd.codigo\_pedido **IS** **NULL**;



21. 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.

**SELECT** **DISTINCT** o.codigo\_oficina, o.ciudad, o.pais, o.region, o.codigo\_postal, o.telefono, o.linea\_direccion1, o.linea\_direccion2

**FROM** oficina o **LEFT** **JOIN** ( **SELECT** e.codigo\_oficina **FROM** empleado e

**NATURAL** **LEFT** **JOIN** cliente c **NATURAL** **LEFT** **JOIN** pedido p **NATURAL** **LEFT** **JOIN** detalle\_pedido dp **NATURAL** **LEFT** **JOIN** producto pr **WHERE** pr.gama = 'Frutales') subquery **ON** o.codigo\_oficina = subquery.codigo\_oficina

**WHERE** subquery.codigo\_oficina **IS** **NULL**;

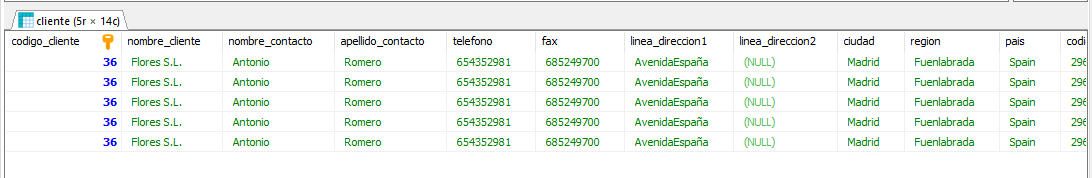


22. Devuelve un listado con los clientes que han realizado algún pedido pero no han

realizado ningún pago.

**SELECT** c.\* **FROM** cliente c **NATURAL** **LEFT** **JOIN** pedido p **LEFT** **JOIN** pago pg **ON** c.codigo\_cliente = pg.codigo\_cliente

**WHERE** p.codigo\_pedido **IS** **NOT** **NULL** **AND** pg.codigo\_cliente **IS** **NULL**;



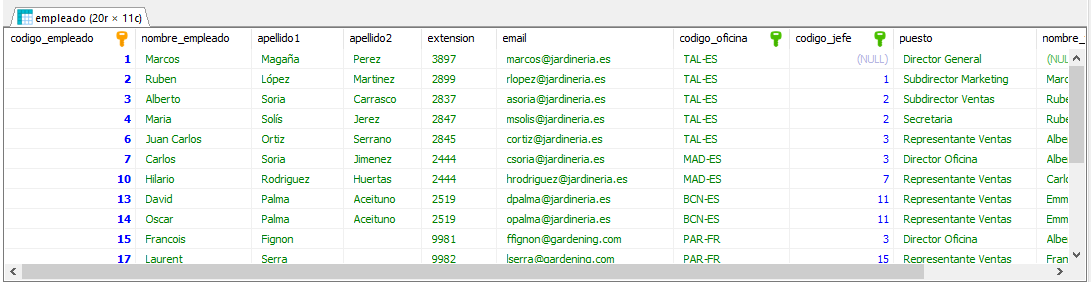
23. Devuelve un listado con los datos de los empleados que no tienen clientes asociados y el nombre de su jefe asociado. Consultas resumen

**SELECT** e1.codigo\_empleado, e1.nombre **AS** nombre\_empleado, e1.apellido1, e1.apellido2, e1.extension, e1.email, e1.codigo\_oficina,

e1.codigo\_jefe, e1.puesto, e2.nombre **AS** nombre\_jefe, e2.apellido1 **AS** apellido\_jefe

**FROM** empleado e1 **LEFT** **JOIN** cliente c **ON** e1.codigo\_empleado = c.codigo\_empleado\_rep\_ventas **LEFT** **JOIN** empleado e2 **ON** e1.codigo\_jefe = e2.codigo\_empleado

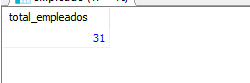
**WHERE** c.codigo\_cliente **IS** **NULL**;



24. ¿Cuántos empleados hay en la compañía?

**SELECT** **COUNT**(\*) **AS** total\_empleados

**FROM** empleado;



25. ¿Cuántos clientes tiene cada país?

**SELECT** pais, **COUNT**(\*) **AS** total\_clientes

**FROM** cliente

**GROUP** **BY** pais;



26. ¿Cuál fue el pago medio en 2009?

**SELECT** **AVG**(total) **AS** pago\_medio\_2009

**FROM** pago

**WHERE** **YEAR**(fecha\_pago) = 2009;



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

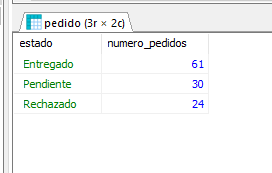
por el número de pedidos.

**SELECT** estado, **COUNT**(\*) **AS** numero\_pedidos

**FROM** pedido

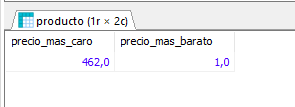
**GROUP** **BY** estado

**ORDER** **BY** numero\_pedidos **DESC**;



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

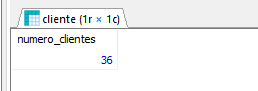
**SELECT** (**SELECT** **MAX**(precio\_venta) **FROM** producto) **AS** precio\_mas\_caro,(**SELECT** **MIN**(precio\_venta) **FROM** producto) **AS** precio\_mas\_barato;



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

**SELECT** **COUNT**(\*) **AS** numero\_clientes

**FROM** cliente;

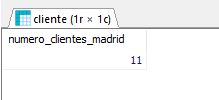


30. ¿Cuántos clientes existen con domicilio en la ciudad de Madrid?

**SELECT** **COUNT**(\*) **AS** numero\_clientes\_madrid

**FROM** cliente

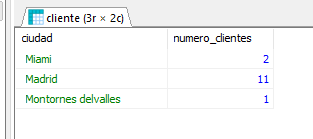
**WHERE** ciudad = 'Madrid';



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

**SELECT** ciudad, **COUNT**(\*) **AS** numero\_clientes

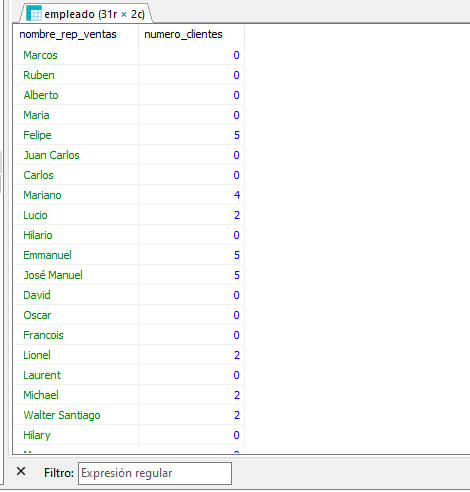
**FROM** cliente **WHERE** ciudad **LIKE** 'M%' **GROUP** **BY** ciudad;



32. Devuelve el nombre de los representantes de ventas y el número de clientes al

que atiende cada uno.

**SELECT** e.nombre **AS** nombre\_rep\_ventas, **COUNT**(c.codigo\_cliente) **AS** numero\_clientes **FROM** empleado e **LEFT** **JOIN** cliente c **ON** e.codigo\_empleado = c.codigo\_empleado\_rep\_ventas **GROUP** **BY** e.nombre;

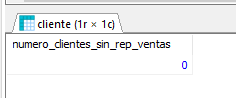


33. Calcula el número de clientes que no tiene asignado representante de ventas.

**SELECT** **COUNT**(\*) **AS** numero\_clientes\_sin\_rep\_ventas

**FROM** cliente

**WHERE** codigo\_empleado\_rep\_ventas **IS** **NULL**;



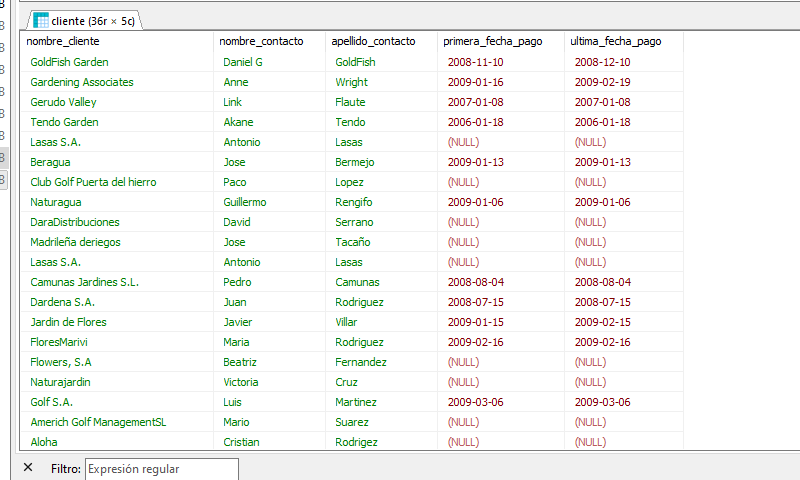
34. Calcula la fecha del primer y último pago realizado por cada uno de los clientes.

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** primera\_fecha\_pago,

**MAX**(p.fecha\_pago) **AS** ultima\_fecha\_pago **FROM** cliente c **LEFT** **JOIN** pago p **ON** c.codigo\_cliente = p.codigo\_cliente

**GROUP** **BY** c.codigo\_cliente, c.nombre\_cliente, c.nombre\_contacto, c.apellido\_contacto;

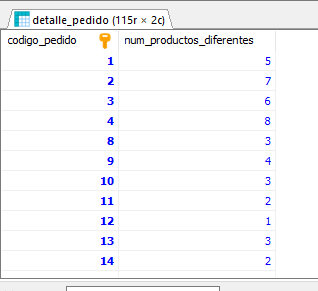


35. Calcula el número de productos diferentes que hay en cada uno de los pedidos.

**SELECT** codigo\_pedido, **COUNT**(**DISTINCT** codigo\_producto) **AS** num\_productos\_diferentes

**FROM** detalle\_pedido

**GROUP** **BY** codigo\_pedido;



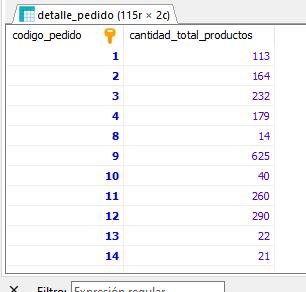
36. Calcula la suma de la cantidad total de todos los productos que aparecen en cada

uno de los pedidos.

**SELECT** codigo\_pedido, **SUM**(cantidad) **AS** cantidad\_total\_productos

**FROM** detalle\_pedido

**GROUP** **BY** codigo\_pedido;



37. 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.

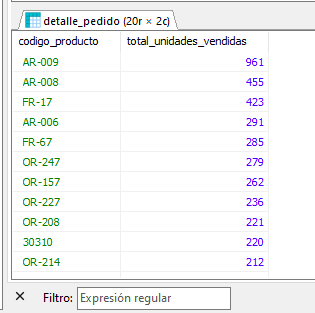
**SELECT** codigo\_producto, **SUM**(cantidad) **AS** total\_unidades\_vendidas

**FROM** detalle\_pedido

**GROUP** **BY** codigo\_producto

**ORDER** **BY** total\_unidades\_vendidas **DESC**

**LIMIT** 20;



38. 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** total\_base\_imponible,

**SUM**(iva) **AS** total\_iva,

**SUM**(total\_facturado) **AS** facturacion\_total **FROM** ( **SELECT**

**SUM**(precio\_unidad \* cantidad) **AS** base\_imponible,

**SUM**(precio\_unidad \* cantidad) \* 0.21 **AS** iva,

**SUM**(precio\_unidad \* cantidad) \* 1.21 **AS** total\_facturado **FROM** detalle\_pedido **GROUP** **BY** codigo\_pedido) **AS** facturacion\_por\_pedido;



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

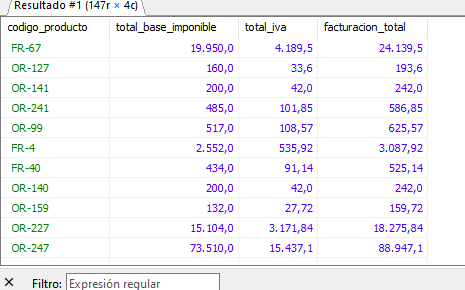
producto.

**SELECT** codigo\_producto, **SUM**(base\_imponible) **AS** total\_base\_imponible,**SUM**(iva) **AS** total\_iva,**SUM**(total\_facturado) **AS** facturacion\_total

**FROM** (**SELECT** codigo\_producto,**SUM**(precio\_unidad \* cantidad) **AS** base\_imponible, **SUM**(precio\_unidad \* cantidad) \* 0.21 **AS** iva,

**SUM**(precio\_unidad \* cantidad) \* 1.21 **AS** total\_facturado **FROM** detalle\_pedido **GROUP** **BY** codigo\_pedido, codigo\_producto)

**AS** facturacion\_por\_producto **GROUP** **BY** codigo\_producto;



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

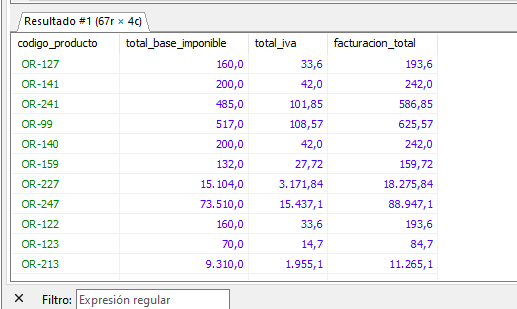
producto filtrada por los códigos que empiecen por OR.

**SELECT** codigo\_producto,**SUM**(base\_imponible) **AS** total\_base\_imponible,**SUM**(iva) **AS** total\_iva,**SUM**(total\_facturado) **AS** facturacion\_total

**FROM** ( **SELECT** codigo\_producto,**SUM**(precio\_unidad \* cantidad) **AS** base\_imponible,**SUM**(precio\_unidad \* cantidad) \* 0.21 **AS** iva,

**SUM**(precio\_unidad \* cantidad) \* 1.21 **AS** total\_facturado **FROM** detalle\_pedido **WHERE** codigo\_producto **LIKE** 'OR%' **GROUP** **BY** codigo\_pedido,

codigo\_producto) **AS** facturacion\_por\_producto **GROUP** **BY** codigo\_producto;



41. 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** p.nombre **AS** nombre\_producto, **SUM**(dp.cantidad) **AS** unidades\_vendidas, **SUM**(dp.precio\_unidad \* dp.cantidad) **AS** total\_facturado,

**SUM**(dp.precio\_unidad \* dp.cantidad) \* 1.21 **AS** total\_con\_iva **FROM** producto p

**INNER** **JOIN** detalle\_pedido dp **ON** p.codigo\_producto = dp.codigo\_producto **GROUP** **BY** p.codigo\_producto, p.nombre **HAVING** total\_facturado

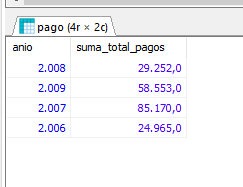
> 3000;



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

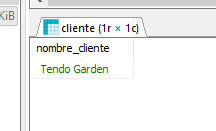
años que aparecen en la tabla pagos.Subconsultas Con operadores básicos de comparación

**SELECT** **YEAR**(fecha\_pago) **AS** anio,**SUM**(total) **AS** suma\_total\_pagos **FROM** pago **GROUP** **BY** **YEAR**(fecha\_pago);



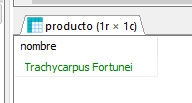
43. Devuelve el nombre del cliente con mayor límite de crédito.

**SELECT** nombre\_cliente **FROM** cliente **WHERE** limite\_credito = (**SELECT** **MAX**(limite\_credito)**FROM** cliente);



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

**SELECT** nombre **FROM** producto **WHERE** precio\_venta = (**SELECT** **MAX**(precio\_venta)**FROM** producto);



45. Devuelve el nombre del producto del que se han vendido más unidades. (Tenga

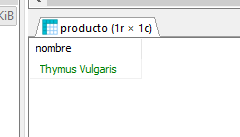
en cuenta que tendrá que calcular cuál es el número total de unidades que se han

vendido de cada producto a partir de los datos de la tabla detalle\_pedido)

**SELECT** nombre **FROM** producto **WHERE** codigo\_producto = (**SELECT** codigo\_producto **FROM** (**SELECT** codigo\_producto, **SUM**(cantidad)

**AS** total\_unidades\_vendidas **FROM** detalle\_pedido **GROUP** **BY** codigo\_producto **ORDER** **BY** total\_unidades\_vendidas **DESC**

**LIMIT** 1 ) **AS** subconsulta);



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

(Sin utilizar INNER JOIN).

**SELECT** nombre\_cliente **FROM** cliente **WHERE** limite\_credito > (**SELECT** **IFNULL**(**SUM**(total), 0)**FROM** pago

**WHERE** pago.codigo\_cliente = cliente.codigo\_cliente);



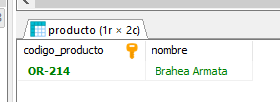
47. Devuelve el producto que más unidades tiene en stock.

**SELECT** codigo\_producto, nombre **FROM** producto **WHERE** cantidad\_en\_stock = (**SELECT** **MAX**(cantidad\_en\_stock)**FROM** producto);



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

**SELECT** codigo\_producto, nombre **FROM** producto **WHERE** cantidad\_en\_stock = (**SELECT** **MIN**(cantidad\_en\_stock)**FROM** producto);

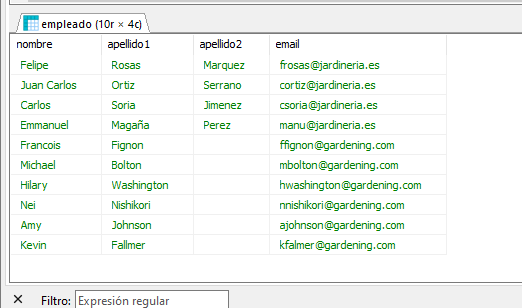


49. Devuelve el nombre, los apellidos y el email de los empleados que están a cargo

de Alberto Soria.

**SELECT** e.nombre, e.apellido1, e.apellido2, e.email **FROM** empleado e **WHERE** e.codigo\_jefe = (**SELECT** codigo\_empleado

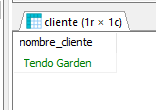
**FROM** empleado **WHERE** nombre = 'Alberto' **AND** apellido1 = 'Soria');



Subconsultas con ALL y ANY

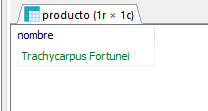
50. Devuelve el nombre del cliente con mayor límite de crédito.

**SELECT** nombre\_cliente **FROM** cliente **WHERE** limite\_credito >= **ALL** (**SELECT** limite\_credito **FROM** cliente);



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

**SELECT** nombre **FROM** producto **WHERE** precio\_venta >= **ALL** (**SELECT** precio\_venta **FROM** producto);



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

**SELECT** nombre **FROM** producto **WHERE** cantidad\_en\_stock <= **ANY** (**SELECT** cantidad\_en\_stock **FROM** producto);



Subconsultas con IN y NOT IN

53. Devuelve el nombre, apellido1 y cargo de los empleados que no representen a

ningún cliente.

**SELECT** nombre, apellido1, puesto **FROM** empleado **WHERE** codigo\_empleado **NOT** **IN** (**SELECT** **DISTINCT** codigo\_empleado\_rep\_ventas **FROM** cliente);



54. Devuelve un listado que muestre solamente los clientes que no han realizado

ningún pago.

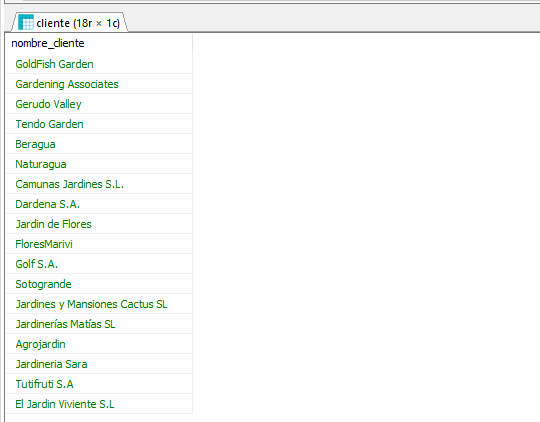
**SELECT** \* **FROM** cliente **WHERE** codigo\_cliente **NOT** **IN** (**SELECT** **DISTINCT** codigo\_cliente **FROM** pago);



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

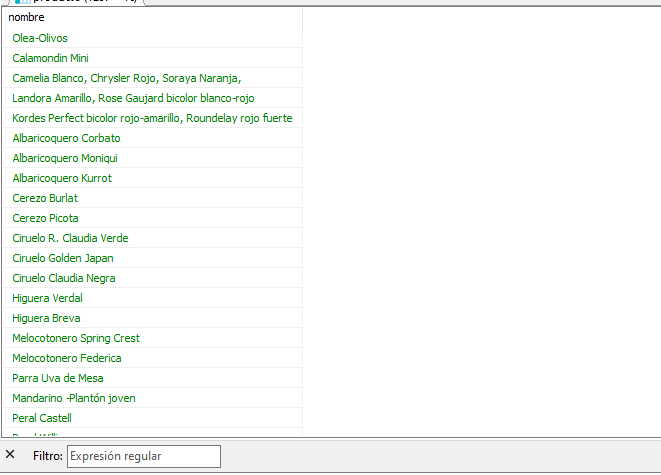
algún pago.

**SELECT** nombre\_cliente **FROM** cliente **WHERE** codigo\_cliente **IN** (**SELECT** **DISTINCT** codigo\_cliente **FROM** pago);



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

**SELECT** nombre **FROM** producto **WHERE** codigo\_producto **NOT** **IN** (**SELECT** **DISTINCT** codigo\_producto **FROM** detalle\_pedido);

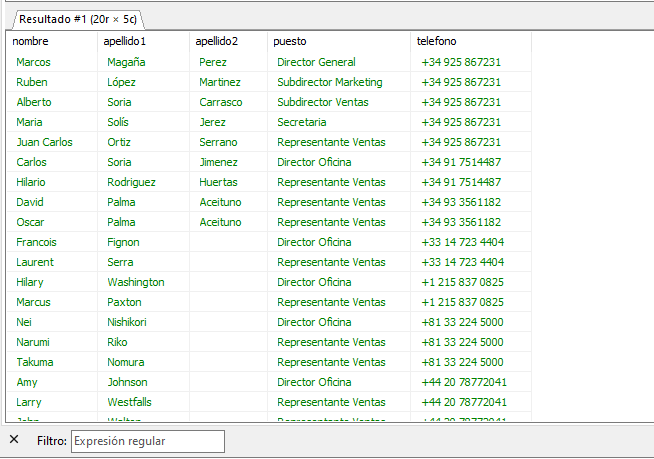


57. Devuelve el nombre, apellidos, puesto y teléfono de la oficina de aquellos

empleados que no sean representante de ventas de ningún cliente.

**SELECT** e.nombre, e.apellido1, e.apellido2, e.puesto, o.telefono **FROM** empleado e **LEFT** **JOIN** oficina o **ON**

e.codigo\_oficina = o.codigo\_oficina **WHERE** e.codigo\_empleado **NOT** **IN** (**SELECT** **DISTINCT** codigo\_empleado\_rep\_ventas **FROM** cliente);



58. 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.

**SELECT** o.\* **FROM** oficina o **WHERE** o.codigo\_oficina **NOT** **IN** (**SELECT** **DISTINCT** e.codigo\_oficina **FROM** empleado e **JOIN** cliente c **ON**

e.codigo\_empleado = c.codigo\_empleado\_rep\_ventas **JOIN** pedido p **ON** c.codigo\_cliente = p.codigo\_cliente **JOIN** detalle\_pedido dp **ON**

p.codigo\_pedido = dp.codigo\_pedido **JOIN** producto pr **ON** dp.codigo\_producto = pr.codigo\_producto **WHERE** pr.gama = 'Frutales');

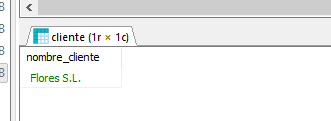


59. Devuelve un listado con los clientes que han realizado algún pedido pero no han

realizado ningún pago.

**SELECT** c.nombre\_cliente **FROM** cliente c **WHERE** c.codigo\_cliente **IN** (**SELECT** p.codigo\_cliente **FROM** pedido p) **AND**

c.codigo\_cliente **NOT** **IN** (**SELECT** pa.codigo\_cliente **FROM** pago pa);

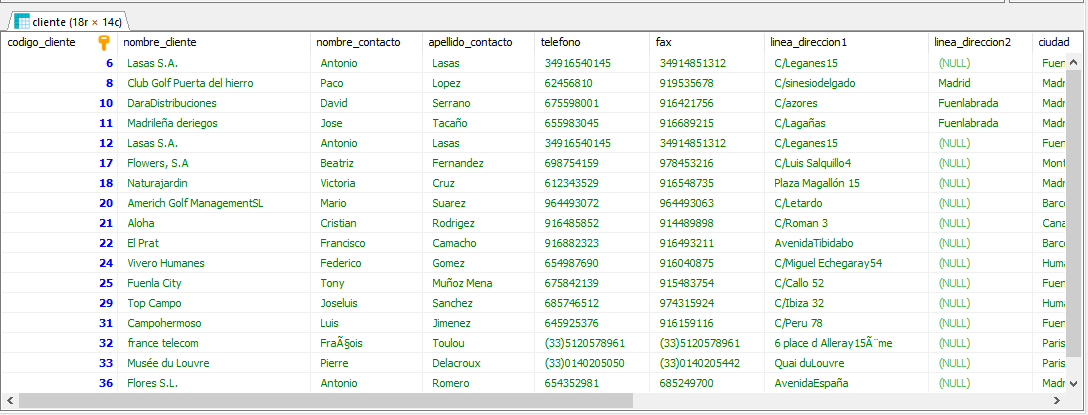


Subconsultas con EXISTS y NOT EXISTS

60. Devuelve un listado que muestre solamente los clientes que no han realizado

ningún pago.

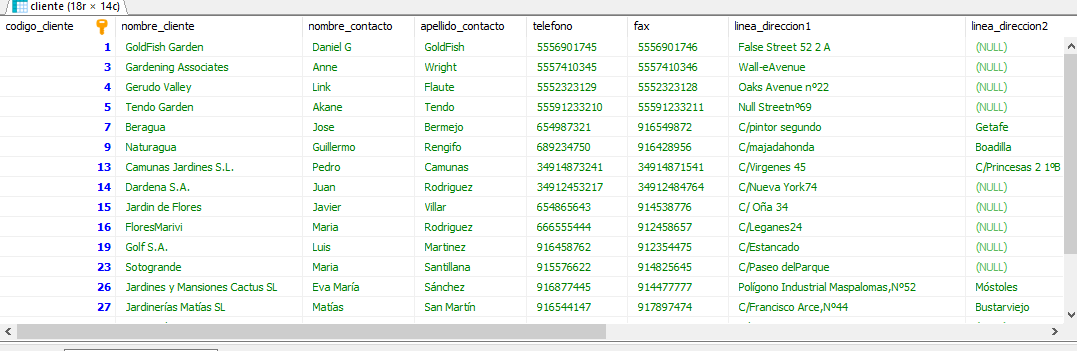
**SELECT** \* **FROM** cliente c **WHERE** **NOT** **EXISTS** (**SELECT** 1 **FROM** pago p **WHERE** p.codigo\_cliente = c.codigo\_cliente);



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

algún pago.

**SELECT** \* **FROM** cliente c **WHERE** **EXISTS** (**SELECT** 1 **FROM** pago p **WHERE** p.codigo\_cliente = c.codigo\_cliente);



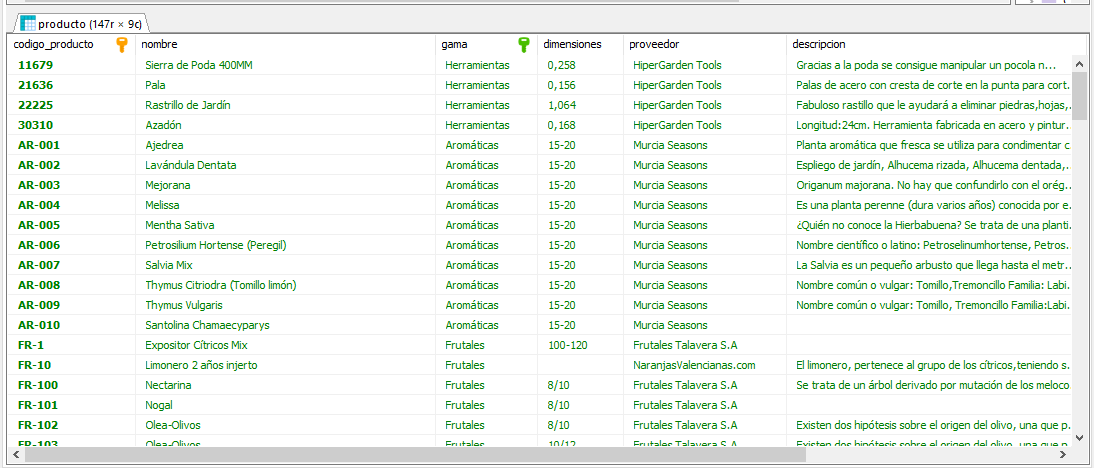
62. Devuelve un listado de los productos que nunca han aparecido en un pedido.

**SELECT** \* **FROM** producto pr **WHERE** **NOT** **EXISTS** (**SELECT** 1 **FROM** detalle\_pedido dp **WHERE** dp.codigo\_producto = pr.codigo\_producto);



63. Devuelve un listado de los productos que han aparecido en un pedido alguna vez.

**SELECT** \* **FROM** producto pr **WHERE** **EXISTS** ( **SELECT** 1 **FROM** detalle\_pedido dp **WHERE** dp.codigo\_producto = pr.codigo\_producto);



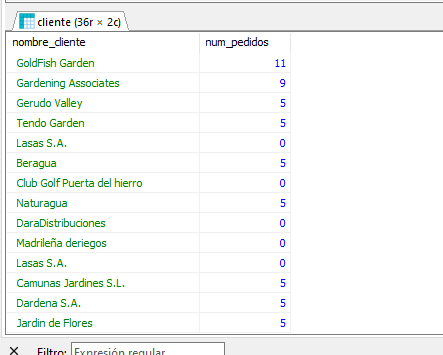
Subconsultas correlacionadas Consultas variadas

64. 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.nombre\_cliente, (**SELECT** **COUNT**(\*) **FROM** pedido p **WHERE** p.codigo\_cliente = c.codigo\_cliente) **AS** num\_pedidos **FROM** cliente c;



65. 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** c.nombre\_cliente, **COALESCE**((**SELECT** **SUM**(total) **FROM** pago p **WHERE** p.codigo\_cliente = c.codigo\_cliente), 0) **AS** total\_pagado

**FROM** cliente c;

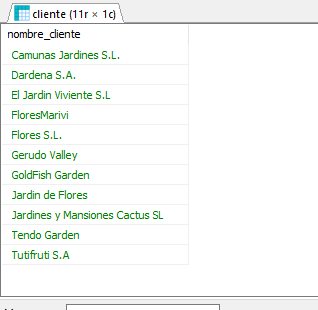


66. Devuelve el nombre de los clientes que hayan hecho pedidos en 2008 ordenados

alfabéticamente de menor a mayor.

**SELECT** nombre\_cliente **FROM** cliente **WHERE** codigo\_cliente **IN** (**SELECT** **DISTINCT** codigo\_cliente **FROM** pedido **WHERE**

**YEAR**(fecha\_pedido) = 2008)**ORDER** **BY** nombre\_cliente **ASC**;



67. 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.nombre\_cliente, e.nombre, e.apellido1, o.telefono **FROM** cliente c **INNER** **JOIN** empleado e **ON**

c.codigo\_empleado\_rep\_ventas = e.codigo\_empleado **INNER** **JOIN** oficina o **ON** e.codigo\_oficina = o.codigo\_oficina **WHERE**

c.codigo\_cliente **NOT** **IN** (**SELECT** **DISTINCT** codigo\_cliente **FROM** pago);



68. 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.nombre\_cliente, e.nombre, e.apellido1, o.ciudad **FROM** cliente c **INNER** **JOIN** empleado e **ON**

c.codigo\_empleado\_rep\_ventas = e.codigo\_empleado **INNER** **JOIN** oficina o **ON** e.codigo\_oficina = o.codigo\_oficina;

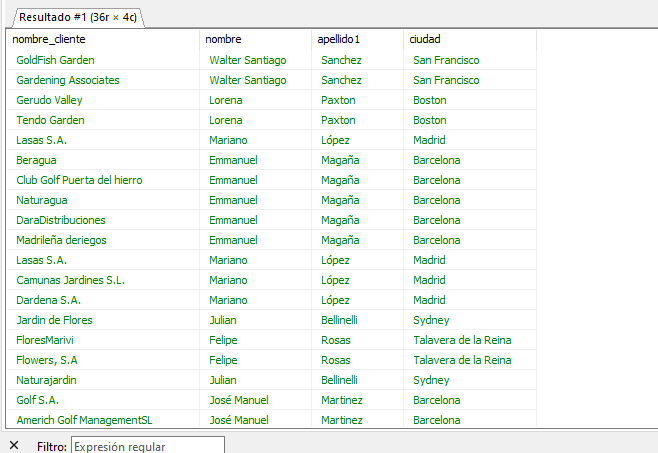


69. Devuelve el nombre, apellidos, puesto y teléfono de la oficina de aquellos

empleados que no sean representante de ventas de ningún cliente.

**SELECT** e.nombre, e.apellido1, e.apellido2, e.puesto, o.telefono **FROM** empleado e **INNER** **JOIN** oficina o **ON**

e.codigo\_oficina = o.codigo\_oficina **WHERE** **NOT** **EXISTS** (**SELECT** \***FROM** cliente c **WHERE** c.codigo\_empleado\_rep\_ventas = e.codigo\_empleado);



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

de empleados que tiene.

**SELECT** o.ciudad,**COUNT**(e.codigo\_empleado) **AS** numero\_empleados **FROM** oficina o **LEFT** **JOIN** empleado e **ON**

o.codigo\_oficina = e.codigo\_oficina **GROUP** **BY** o.ciudad;

