**Texto

Descripción generada automáticamente**

**Nombre y apellidos:**

**Fecha:**

**Actividad 4: Consultas resumen.**

1. **Script de la BD a ejecutar (no hay que crear un esquema):**

**DROP** **DATABASE** **IF** **EXISTS** jardineria;

**CREATE** **DATABASE** jardineria CHARACTER **SET** utf8mb4;

**USE** jardineria;

**CREATE** **TABLE** oficina (

codigo\_oficina VARCHAR(10) **NOT** **NULL**,

ciudad VARCHAR(30) **NOT** **NULL**,

pais VARCHAR(50) **NOT** **NULL**,

region VARCHAR(50) **DEFAULT** **NULL**,

codigo\_postal VARCHAR(10) **NOT** **NULL**,

telefono VARCHAR(20) **NOT** **NULL**,

linea\_direccion1 VARCHAR(50) **NOT** **NULL**,

linea\_direccion2 VARCHAR(50) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (codigo\_oficina)

);

**CREATE** **TABLE** empleado (

codigo\_empleado INTEGER **NOT** **NULL**,

nombre VARCHAR(50) **NOT** **NULL**,

apellido1 VARCHAR(50) **NOT** **NULL**,

apellido2 VARCHAR(50) **DEFAULT** **NULL**,

extension VARCHAR(10) **NOT** **NULL**,

email VARCHAR(100) **NOT** **NULL**,

codigo\_oficina VARCHAR(10) **NOT** **NULL**,

codigo\_jefe INTEGER **DEFAULT** **NULL**,

puesto VARCHAR(50) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (codigo\_empleado),

**FOREIGN** **KEY** (codigo\_oficina) **REFERENCES** oficina (codigo\_oficina),

**FOREIGN** **KEY** (codigo\_jefe) **REFERENCES** empleado (codigo\_empleado)

);

**CREATE** **TABLE** gama\_producto (

gama VARCHAR(50) **NOT** **NULL**,

descripcion\_texto TEXT,

descripcion\_html TEXT,

imagen VARCHAR(256),

**PRIMARY** **KEY** (gama)

);

**CREATE** **TABLE** cliente (

codigo\_cliente INTEGER **NOT** **NULL**,

nombre\_cliente VARCHAR(50) **NOT** **NULL**,

nombre\_contacto VARCHAR(30) **DEFAULT** **NULL**,

apellido\_contacto VARCHAR(30) **DEFAULT** **NULL**,

telefono VARCHAR(15) **NOT** **NULL**,

fax VARCHAR(15) **NOT** **NULL**,

linea\_direccion1 VARCHAR(50) **NOT** **NULL**,

linea\_direccion2 VARCHAR(50) **DEFAULT** **NULL**,

ciudad VARCHAR(50) **NOT** **NULL**,

region VARCHAR(50) **DEFAULT** **NULL**,

pais VARCHAR(50) **DEFAULT** **NULL**,

codigo\_postal VARCHAR(10) **DEFAULT** **NULL**,

codigo\_empleado\_rep\_ventas INTEGER **DEFAULT** **NULL**,

limite\_credito NUMERIC(15,2) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (codigo\_cliente),

**FOREIGN** **KEY** (codigo\_empleado\_rep\_ventas) **REFERENCES** empleado (codigo\_empleado)

);

**CREATE** **TABLE** pedido (

codigo\_pedido INTEGER **NOT** **NULL**,

fecha\_pedido date **NOT** **NULL**,

fecha\_esperada date **NOT** **NULL**,

fecha\_entrega date **DEFAULT** **NULL**,

estado VARCHAR(15) **NOT** **NULL**,

comentarios TEXT,

codigo\_cliente INTEGER **NOT** **NULL**,

**PRIMARY** **KEY** (codigo\_pedido),

**FOREIGN** **KEY** (codigo\_cliente) **REFERENCES** cliente (codigo\_cliente)

);

**CREATE** **TABLE** producto (

codigo\_producto VARCHAR(15) **NOT** **NULL**,

nombre VARCHAR(70) **NOT** **NULL**,

gama VARCHAR(50) **NOT** **NULL**,

dimensiones VARCHAR(25) **NULL**,

proveedor VARCHAR(50) **DEFAULT** **NULL**,

descripcion text **NULL**,

cantidad\_en\_stock SMALLINT **NOT** **NULL**,

precio\_venta NUMERIC(15,2) **NOT** **NULL**,

precio\_proveedor NUMERIC(15,2) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (codigo\_producto),

**FOREIGN** **KEY** (gama) **REFERENCES** gama\_producto (gama)

);

**CREATE** **TABLE** detalle\_pedido (

codigo\_pedido INTEGER **NOT** **NULL**,

codigo\_producto VARCHAR(15) **NOT** **NULL**,

cantidad INTEGER **NOT** **NULL**,

precio\_unidad NUMERIC(15,2) **NOT** **NULL**,

numero\_linea SMALLINT **NOT** **NULL**,

**PRIMARY** **KEY** (codigo\_pedido, codigo\_producto),

**FOREIGN** **KEY** (codigo\_pedido) **REFERENCES** pedido (codigo\_pedido),

**FOREIGN** **KEY** (codigo\_producto) **REFERENCES** producto (codigo\_producto)

);

**CREATE** **TABLE** pago (

codigo\_cliente INTEGER **NOT** **NULL**,

forma\_pago VARCHAR(40) **NOT** **NULL**,

id\_transaccion VARCHAR(50) **NOT** **NULL**,

fecha\_pago date **NOT** **NULL**,

total NUMERIC(15,2) **NOT** **NULL**,

**PRIMARY** **KEY** (codigo\_cliente, id\_transaccion),

**FOREIGN** **KEY** (codigo\_cliente) **REFERENCES** cliente (codigo\_cliente)

);

1. **Consultas:**

**Antes de empezar con las consultas, generar el modelo entidad-relación en MySQL.**

**2.0 Consultas sencillas:**

1. Devuelve un listado con el código de oficina y la ciudad donde hay oficinas.
2. Devuelve un listado con la ciudad y el teléfono de las oficinas de España.
3. Devuelve un listado con el nombre, apellidos y email de los empleados cuyo jefe tiene un código de jefe igual a 7.
4. Devuelve el nombre del puesto, nombre, apellidos y email del jefe de la empresa.
5. Devuelve un listado con el nombre, apellidos y puesto de aquellos empleados que no sean representantes de ventas.
6. Devuelve un listado con el nombre de los todos los clientes españoles.
7. Devuelve un listado con los distintos estados por los que puede pasar un pedido.
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:

* Utilizando la función [YEAR de MySQL](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function_year).
* Utilizando la función [DATE\_FORMAT de MySQL](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function_date-format).
* Sin utilizar ninguna de las funciones anteriores.

1. Devuelve un listado con el código de pedido, código de cliente, fecha esperada y fecha de entrega de los pedidos que no han sido entregados a tiempo.
2. Devuelve un listado con el código de pedido, código de cliente, fecha esperada y fecha de entrega de los pedidos cuya fecha de entrega ha sido al menos dos días antes de la fecha esperada.

* Utilizando la función [ADDDATE de MySQL](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function_adddate).
* Utilizando la función [DATEDIFF de MySQL](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function_datediff).
* ¿Sería posible resolver esta consulta utilizando el operador de suma + o resta -?

1. Devuelve un listado de todos los pedidos que fueron **rechazados** en 2009.
2. Devuelve un listado de todos los pedidos que han sido **entregados** en el mes de enero de cualquier año.
3. Devuelve un listado con todos los pagos que se realizaron en el año 2008 mediante Paypal. Ordene el resultado de mayor a menor.
4. 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.
5. 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.
6. 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.

**2.1 Consultas resumen:**

1. ¿Cuántos empleados hay en la compañía?
2. ¿Cuántos clientes tiene cada país?
3. ¿Cuál fue el pago medio en 2009?
4. ¿Cuántos pedidos hay en cada estado? Ordena el resultado de forma descendente por el número de pedidos.
5. Calcula el precio de venta del producto más caro y más barato en una misma consulta.
6. Calcula el número de clientes que tiene la empresa.
7. ¿Cuántos clientes existen con domicilio en la ciudad de Madrid?
8. ¿Calcula cuántos clientes tiene cada una de las ciudades que empiezan por M?
9. Devuelve el nombre de los representantes de ventas y el número de clientes al que atiende cada uno.
10. Calcula el número de clientes que no tiene asignado representante de ventas.
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.
12. Calcula el número de productos diferentes que hay en cada uno de los pedidos.
13. Calcula la suma de la cantidad total de todos los productos que aparecen en cada uno de los pedidos.
14. 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.
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.
16. La misma información que en la pregunta anterior, pero agrupada por código de producto.
17. La misma información que en la pregunta anterior, pero agrupada por código de producto filtrada por los códigos que empiecen por OR.
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).
19. Muestre la suma total de todos los pagos que se realizaron para cada uno de los años que aparecen en la tabla pagos.

**1.4.5 Consultas multitabla (Composición interna)**

Las consultas con sintaxis de SQL2 se deben resolver con INNER JOIN y NATURAL JOIN. (Mirar las diferencias en internet. Si se pueden hacer con INNER JOIN hacer con INNER que son las que se han visto más en profundidad)

1. Obtén un listado con el nombre de cada cliente y el nombre y apellido de su representante de ventas.
2. Muestra el nombre de los clientes que hayan realizado pagos junto con el nombre de sus representantes de ventas.
3. Muestra el nombre de los clientes que **no** hayan realizado pagos junto con el nombre de sus representantes de ventas.
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.
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.
6. Lista la dirección de las oficinas que tengan clientes en 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.
8. Devuelve un listado con el nombre de los empleados junto con el nombre de sus jefes.
9. Devuelve un listado que muestre el nombre de cada empleado, el nombre de su jefe y el nombre del jefe de sus jefes.
10. Devuelve el nombre de los clientes a los que no se les ha entregado a tiempo un pedido.
11. Devuelve un listado de las diferentes gamas de producto que ha comprado cada cliente.

**1.4.6 Consultas multitabla (Composición externa)**

Resolver todas las consultas utilizando las cláusulas LEFT JOIN, RIGHT JOIN.

1. Devuelve un listado que muestre solamente los clientes que no han realizado ningún pago.
2. Devuelve un listado que muestre solamente los clientes que no han realizado ningún pedido.
3. Devuelve un listado que muestre los clientes que no han realizado ningún pago y los que no han realizado ningún pedido.
4. Devuelve un listado que muestre solamente los empleados que no tienen una oficina asociada.
5. Devuelve un listado que muestre solamente los empleados que no tienen un cliente asociado.
6. Devuelve un listado que muestre solamente los empleados que no tienen un cliente asociado junto con los datos de la oficina donde trabajan.
7. Devuelve un listado que muestre los empleados que no tienen una oficina asociada y los que no tienen un cliente asociado.
8. Devuelve un listado de los productos que nunca han aparecido en un pedido.
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.
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.
12. Devuelve un listado con los datos de los empleados que no tienen clientes asociados y el nombre de su jefe asociado.