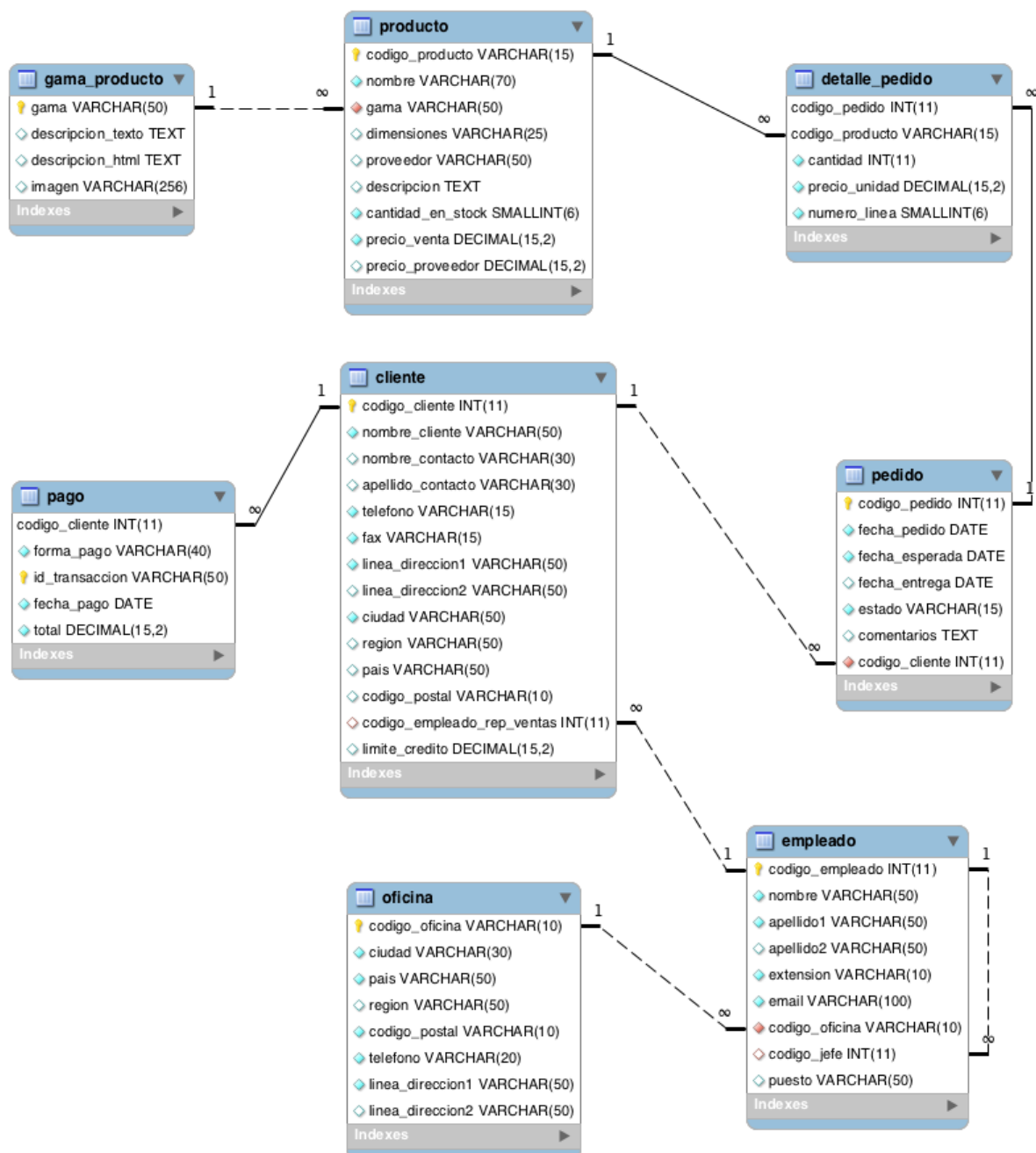


En base al DER, arme la base de datos correspondiente e inserte los siguientes datos, luego realice las consultas en SQL:

1.4 Jardinería- Modelo entidad/relación



1.4.2 Base de datos para MySQL

```
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)
```

);

1.4.3 Datos

Acceder al [script SQL](#) para la creación de la base de datos y la inserción de datos: [jardineria.sql](#).

1.4.4 Consultas sobre una tabla

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](#).
 - Utilizando la función [DATE_FORMAT de MySQL](#).
 - Sin utilizar ninguna de las funciones anteriores.
9. 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.
10. 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](#).
 - Utilizando la función [DATEDIFF de MySQL](#).
 - ¿Sería posible resolver esta consulta utilizando el operador de suma + o resta -?

11. Devuelve un listado de todos los pedidos que fueron rechazados en 2009.
12. Devuelve un listado de todos los pedidos que han sido entregados en el mes de enero de cualquier año.
13. Devuelve un listado con todos los pagos que se realizaron en el año 2008 mediante PayPal. Ordene el resultado de mayor a menor.
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.
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.
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.

1.4.5 Consultas multitable (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.
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 empleados, el nombre de su jefe y el nombre del jefe de su jefe.
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 multitable (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.
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.

1.4.7 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`.

1.4.8 Subconsultas

1.4.8.1 Con operadores básicos de comparación

1. Devuelve el nombre del cliente con mayor límite de crédito.
2. Devuelve el nombre del producto que tenga el precio de venta más caro.
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`)

4. Los clientes cuyo límite de crédito sea mayor que los pagos que haya realizado. (Sin utilizar `INNER JOIN`).
5. Devuelve el producto que más unidades tiene en stock.
6. Devuelve el producto que menos unidades tiene en stock.
7. Devuelve el nombre, los apellidos y el email de los empleados que están a cargo de Alberto Soria.

1.4.8.2 Subconsultas con ALL y ANY

8. Devuelve el nombre del cliente con mayor límite de crédito.
9. Devuelve el nombre del producto que tenga el precio de venta más caro.
10. Devuelve el producto que menos unidades tiene en stock.

1.4.8.3 Subconsultas con IN y NOT IN

11. Devuelve el nombre, apellido1 y cargo de los empleados que no representen a ningún cliente.
12. Devuelve un listado que muestre solamente los clientes que no han realizado ningún pago.
13. Devuelve un listado que muestre solamente los clientes que sí han realizado algún pago.
14. Devuelve un listado de los productos que nunca han aparecido en un pedido.
15. Devuelve el nombre, apellidos, puesto y teléfono de la oficina de aquellos empleados que no sean representante de ventas de ningún 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.

1.4.8.4 Subconsultas con EXISTS y NOT EXISTS

18. Devuelve un listado que muestre solamente los clientes que no han realizado ningún pago.
19. Devuelve un listado que muestre solamente los clientes que sí han realizado algún pago.
20. Devuelve un listado de los productos que nunca han aparecido en un pedido.
21. Devuelve un listado de los productos que han aparecido en un pedido alguna vez.

1.4.8.5 Subconsultas correlacionadas

1.4.9 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.**
- 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.**
- 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.**
- 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.**
- 6. Devuelve el nombre, apellidos, puesto y teléfono de la oficina de aquellos empleados que no sean representante de ventas de ningún cliente.**
- 7. Devuelve un listado indicando todas las ciudades donde hay oficinas y el número de empleados que tiene.**