

1.4 Jardinería

1.4.1 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, SELECT DISTINCT c2.codigo_cliente ,c2.nombre_cliente
FROM cliente c2 INNER JOIN pago
ON c2.codigo_cliente = pago.codigo_cliente
  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.

```

SELECT codigo_oficina, ciudad
FROM oficina;

```

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

```

SELECT ciudad, telefono
FROM oficina
WHERE pais = 'España';

SELECT ciudad, telefono
FROM oficina
WHERE pais LIKE '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, codigo_jefe
FROM empleado
WHERE codigo_jefe = 7;

```

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

```

SELECT puesto, nombre, apellido1, apellido2, email
FROM empleado
WHERE codigo_jefe IS NULL;

```

5. Devuelve un listado con el nombre, apellidos y puesto de aquellos empleados que no sean representantes de ventas.

```

SELECT nombre, apellido1, apellido2, puesto
FROM empleado
WHERE puesto != 'Representante Ventas';

SELECT nombre, apellido1, apellido2, puesto
FROM empleado
WHERE puesto NOT LIKE 'Representante Ventas';

```

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

```
SELECT nombre_cliente
FROM cliente
WHERE pais = 'Spain';
```

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

```
SELECT DISTINCT estado
FROM 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.

```
SELECT DISTINCT codigo_cliente
FROM pago
WHERE YEAR(fecha_pago) = 2008;

SELECT DISTINCT codigo_cliente
FROM pago
WHERE DATE_FORMAT(fecha_pago, '%Y') = 2008;

SELECT DISTINCT codigo_cliente
FROM pago
WHERE fecha_pago >= '2008-01-01' AND fecha_pago <= '2008-12-31';
```

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.

```
SELECT codigo_pedido, codigo_cliente, fecha_esperada, fecha_entrega
FROM pedido
WHERE fecha_entrega > fecha_esperada;
```

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](#).

```
SELECT codigo_pedido, codigo_cliente, fecha_esperada, fecha_entrega
FROM pedido
WHERE fecha_entrega <= ADDDATE(fecha_esperada, -2);

SELECT codigo_pedido, codigo_cliente, fecha_esperada, fecha_entrega
FROM pedido
WHERE DATEDIFF(fecha_esperada, fecha_entrega) >= 2;
```

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

```
SELECT *
FROM pedido
WHERE YEAR(fecha_pedido) = 2009 AND estado = 'Rechazado';
```

12. Devuelve un listado de todos los pedidos que han sido **entregados** en el mes de enero de cualquier año.

```
SELECT *
FROM pedido
WHERE MONTH(fecha_pedido) = 1 AND estado = 'Entregado';
```

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 *
FROM pago
WHERE forma_pago = 'Paypal' AND YEAR(fecha_pago) = 2008
ORDER BY total DESC;
```

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 forma_pago
FROM pago;
```

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 *
FROM producto
WHERE gama = 'Ornamentales' AND cantidad_en_stock >= 100
ORDER BY precio_venta DESC;
```

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 *
FROM cliente
WHERE
```