

TP MYSQL

La empresa Dulce Capricho se dedica a la comercialización de productos elaborados de pastelería y panadería. Se encuentra en un momento de crecimiento por lo que necesita gestionar las ordenes de pedidos y ventas de forma eficiente. Es clave tener información actualizada en tiempo y forma para poder tomar decisiones en cuanto a compra de insumos y maquinarias como también tener los stocks actualizados.

Se diseña la base de datos **Tienda** para la gestión integral de los pedidos y ventas que realicen los clientes desde la web dulcecapricho.com.ar.

Las tablas principales son:

- **Clientes:** En esta tabla se cargan los distintos clientes de la empresa

id: campo numérico que identifica a cada cliente

nombre: cadena de caracteres con nombre y apellido del cliente

direccion: cadena de caracteres que identifica domicilio del cliente

correo_electronico: cadena de caracteres con el mail de contacto

created_at: campo que identifica cuando fue dado de alta el cliente en la base de datos

- **Productos:** En esta tabla se cargan todos los productos que la empresa comercializa

id: campo numérico que identifica a cada producto

nombre: cadena de caracteres con el nombre que identifica a un producto

descripcion: texto en el que se puede agregar información adicional al nombre

precio: en este campo se define el precio unitario del producto

cantidad_stock: campo que identifica las existencias del producto

created_at: campo que identifica cuando fue dado de alta el producto en la base de datos

- **CategoriasProductos:** En esta tabla definimos las categorías o rubros en los que agrupamos los productos

id: campo numérico que identifica a una categoría

nombre: cadena de caracteres con el nombre de la categoría

descripcion: texto en el que se puede agregar información adicional al nombre

created_at: campo que identifica cuando fue dada de alta la categoría en la base de datos

- **Ordenes:** En esta tabla se cargan las ordenes de pedidos de los distintos clientes

id: campo numérico que identifica a una orden de pedido

cliente_id: campo numérico que identifica al cliente que realiza el pedido.

Tiene que existir en la tabla de Clientes

fecha_orden: campo fecha que identifica la fecha en la que se realizó el pedido

estado: cadena de caracteres que identifica el estado del pedido.

Los valores posibles son Pendiente, En preparación, Facturado y Cancelado

total: campo numérico que identifica el importe total del pedido

created_at: campo que identifica cuando fue dada de alta la categoría en la base de datos

- **DetallesOrdenes:** En esta tabla se carga el detalle de la orden de pedido identificando los productos, la cantidad, precio y total por cada renglón de la misma

id: campo numérico que identifica un renglón del pedido

orden_id: campo numérico que identifica la orden de pedido.

Tiene que existir en la tabla Ordenes

producto_id: campo numérico que identifica un producto del pedido.

Tiene que existir en la tabla Productos

cantidad: campo numérico que identifica la cantidad pedida de cada producto

precio_unitario: campo numérico con el precio de cada producto

total: campo numérico con el importe por renglón resultante del producto entre precio unitario y cantidad

created_at: campo que identifica cuando fue dada de alta el renglón del pedido en la base de datos

- **Empleados:** En esta tabla se cargan los empleados de la empresa con sus datos

id: campo numérico que identifica a un empleado

nombre: cadena de caracteres con el nombre del empleado

puesto: cadena de caracteres que identifica el puesto que ocupa el empleado en la empresa

fecha_contratacion: campo fecha con el día en que lo contrataron

created_at: campo que identifica cuando fue dada de alta el empleado en la base de datos

- **Ventas:** En esta tabla se registran todas las ventas realizadas por los empleados

id: campo numérico que identifica cada venta realizada

cliente_id: campo numérico que identifica al cliente.

Tiene que existir en la tabla de Clientes

empleado_id: campo numérico que identifica al empleado que realizo la venta.

Tiene que existir en la tabla de Empleados

fecha_venta: campo fecha que identifica la fecha en que se realizo la venta

total: campo numérico con el importe total de la venta realizada

created_at: campo que identifica cuando fue dada de alta la venta en la base de datos

Relaciones entre las tablas:

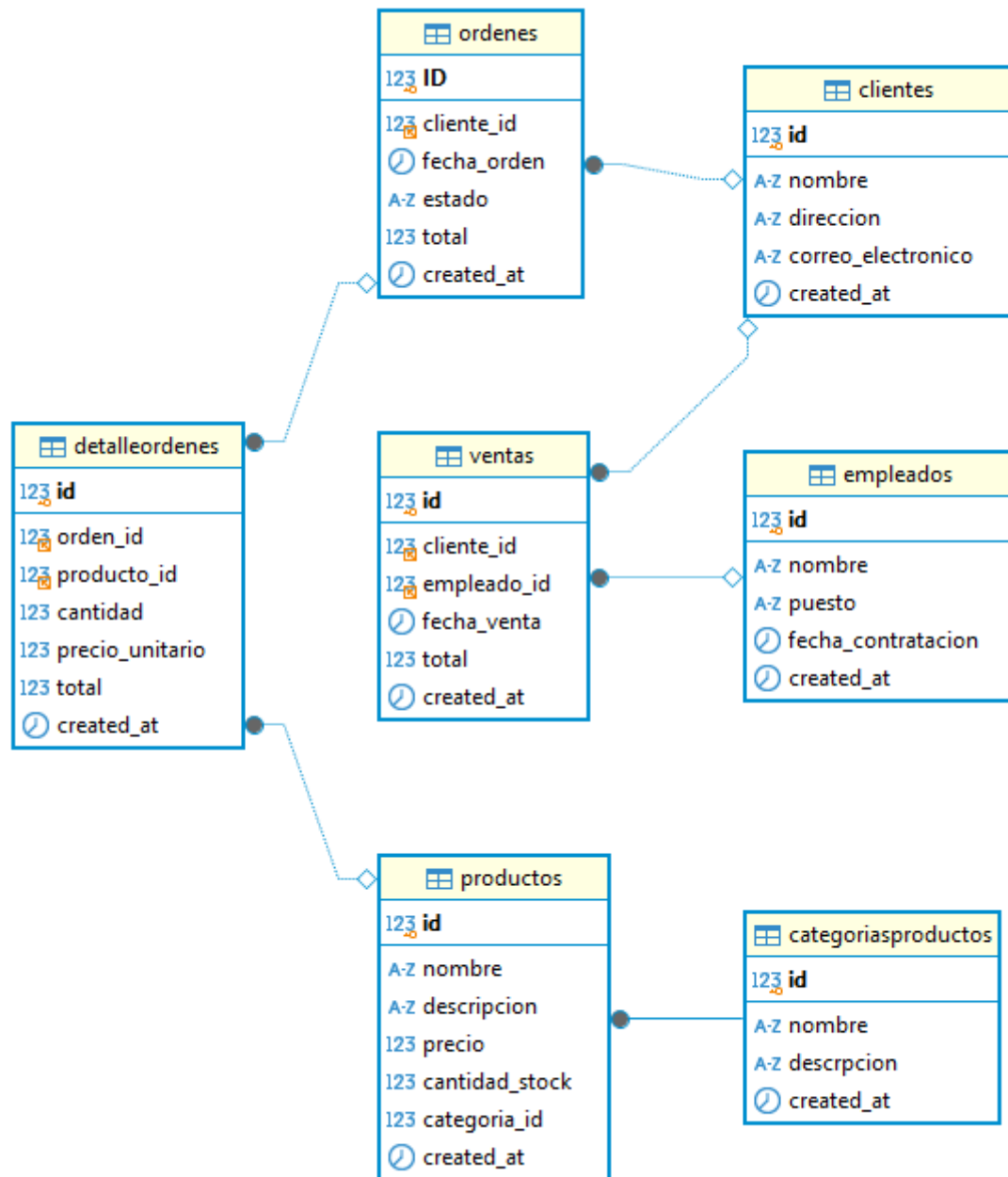
- Un producto pertenece a una única categoría de productos, pero una categoría puede tener varios productos.

- Una orden es realizada por un único cliente, pero un cliente puede realizar varias órdenes.

- Un detalle de orden está asociado a una única orden y a un único producto, pero una orden puede tener varios detalles de órdenes y un producto puede estar asociado a varios detalles de órdenes.

- Una venta es realizada por un único cliente y atendida por un único empleado, pero un cliente puede realizar varias ventas y un empleado puede atender a varios clientes en diferentes ventas.

Diagrama de entidad-relación (ER)



Definición SQL

//Crea Base de Datos

```
CREATE DATABASE tienda;
```

//Crea Tabla

```
CREATE TABLE clientes (  
    id INT PRIMARY KEY auto_increment not null,  
    nombre VARCHAR(50) not null,  
    direccion VARCHAR(50) not null,  
    correo_electronico VARCHAR(50),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

//Crea Tabla

```
CREATE TABLE categoriasproductos (  
    id INT PRIMARY KEY auto_increment not null,  
    nombre VARCHAR(50) not null,  
    descripcion TEXT(50),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

//Crea Tabla

```
CREATE TABLE productos (  
    id INT PRIMARY KEY auto_increment not null,  
    nombre VARCHAR(50) not null,  
    descripcion TEXT(50),  
    precio DECIMAL(10,2),  
    cantidad_stock INT,
```

```
categoria_id INT,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
constraint fk_categorias_productos  
foreign key (categoria_id) references categoriasproductos(id)  
);
```

//Crea Tabla

```
CREATE TABLE ordenes (  
    ID INT PRIMARY KEY auto_increment not null,  
    cliente_id INT,  
    fecha_orden DATE,  
    estado VARCHAR(15),  
    total DECIMAL (10,2),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    constraint fk_cliente  
    foreign key (cliente_id) references clientes(id)  
);
```

//Crea Tabla

```
CREATE TABLE detalleordenes (  
    id int PRIMARY KEY auto_increment not null,  
    orden_id int,  
    producto_id int,  
    cantidad INT,  
    precio_unitario DECIMAL(10,2),  
    total DECIMAL(10,2),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    constraint fk_orden  
    foreign key (orden_id) references ordenes(id),
```

```
constraint fk_producto  
foreign key (producto_id) references productos(id)  
);
```

//Crea Tabla

```
CREATE TABLE empleados (  
    id int PRIMARY KEY auto_increment not null,  
    nombre VARCHAR(50) not null,  
    puesto VARCHAR(50),  
    fecha_contratacion DATE,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

//Crea Tabla

```
CREATE TABLE ventas (  
    id int primary key auto_increment not null,  
    cliente_id int,  
    empleado_id int,  
    fecha_venta DATE,  
    total DECIMAL(10,2),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    foreign key (cliente_id) references clientes(id),  
    foreign key (empleado_id) references empleados(id)  
);
```