

# POS System SQL Project Documentation

## 1 Business Principles:

1. Each customer should order at least one or more to be considered a customer.
2. Every employee can be associated with many orders.
3. Each order is assigned to one and only one customer.
4. Each order is assigned to one and only one employee.
5. Each order is assigned to at least one or many orderProduct.
6. OrderProduct entity is a bridge entity that links a many-to-many relationship between order and product.
7. Each Product has zero or many orderProduct.
8. Each product is associated with many InventoryProduct.

## 2 Business Rules:

1. A customer can only order through an employee.
2. An employee processes the customer's orders by adding the products and details into the system.
3. An 'OrderProduct' record represents each item a customer ordered from the product entity with specific details and notes.
4. An item is added to the inventory through the inventory system.
5. 'InventoryProduct' bridges the products available on the menu to the inventory items.

### 3 Entity Relations:

1. Customer  $\rightarrow$  Order  $\rightarrow$  Employee
2. Employee  $\rightarrow$  OrderProduct
3. OrderProduct  $\rightarrow$  Product
4. Inventory  $\rightarrow$  InventoryProduct  $\rightarrow$  Product

### 4 Database Schema:

#### 4.1 1. Customer:

```
CREATE TABLE Customer (  
    CustomerID SERIAL PRIMARY KEY,  
    FirstName VARCHAR(45),  
    LastName VARCHAR(45),  
    Email VARCHAR(45),  
    PhoneName VARCHAR(10),  
    OrderCount INT DEFAULT 0  
);
```

#### 4.2 2. Employee:

```
CREATE TABLE Employee (  
    EmployeeID SERIAL PRIMARY KEY,  
    FirstName VARCHAR(45) NOT NULL,  
    LastName VARCHAR(45) NOT NULL,  
    Postion VARCHAR(45) NOT NULL,  
    Email VARCHAR(45),  
    Phone VARCHAR(10),  
    HireDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    Passcode VARCHAR(10) NOT NULL  
);
```

#### 4.3 3. Order:

```
CREATE TABLE Order (  
    OrderID SERIAL PRIMARY KEY,  
    CustomerID INT NOT NULL,  
    EmployeeID INT NOT NULL,  
    OrderDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    Total DECIMAL(10, 2) NOT NULL,  
    IsPending BOOL DEFAULT TRUE,  
    IsRefunded BOOL DEFAULT FALSE,  
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
```

```
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);
```

#### 4.4 4. OrderProduct:

```
CREATE TABLE OrderProduct (
    OrderProductID SERIAL PRIMARY KEY,
    ProductID INT NOT NULL,
    OrderID INT NOT NULL,
    Quantity INT NOT NULL,
    Note TEXT,
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID),
    FOREIGN KEY (OrderID) REFERENCES Order(OrderID)
);
```

#### 4.5 5. Product:

```
CREATE TABLE Product (
    ProductID SERIAL PRIMARY KEY,
    Name VARCHAR(45) NOT NULL,
    Description TEXT,
    Price DECIMAL(10, 2),
    Category VARCHAR(45) NOT NULL
);
```

#### 4.6 6. InventoryProduct:

```
CREATE TABLE InventoryProduct (
    InventoryProductID SERIAL PRIMARY KEY,
    ProductID INT NOT NULL,
    InventoryID INT NOT NULL,
    Quantity DECIMAL(10, 2),
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID),
    FOREIGN KEY (InventoryID) REFERENCES Inventory(InventoryID)
);
```

#### 4.7 7. Inventory:

```
CREATE TABLE Inventory (
    InventoryID SERIAL PRIMARY KEY,
    Name VARCHAR(45) NOT NULL,
    Details TEXT,
    Quantity INT DEFAULT 0,
    LastUpdated TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

## 5 Low-Level Design:

The low-level design of this system can be visualized with the above entity relationship model.

1. 'Customer' has a one-to-many relationship with 'Order'.
2. 'Employee' has a one-to-many relationship with 'Order'.
3. 'Order' has a one-to-many relationship with 'OrderProduct'.
4. 'Product' has a one-to-many relationship with 'OrderProduct' and a one-to-many relationship with 'InventoryProduct'.
5. 'Inventory' has a one-to-many relationship with 'InventoryProduct'.

This design allows for a modular and scalable system, where new customers, employees, products, and inventory can be easily added without disrupting the overall flow. The relationships ensure that the system maintains referential integrity and accurate data representation.