

LES Coffee Shop

Sam Aseoff, Erik Johnston, and Leah Nelson

Introduction

Scope:

- Domain: Coffee Shop
- Business Function: Customer Ordering



Why:

- Team experience
- Coffee culture
- Straight forward process
- Relatable process



Conceptual Design

Entities:

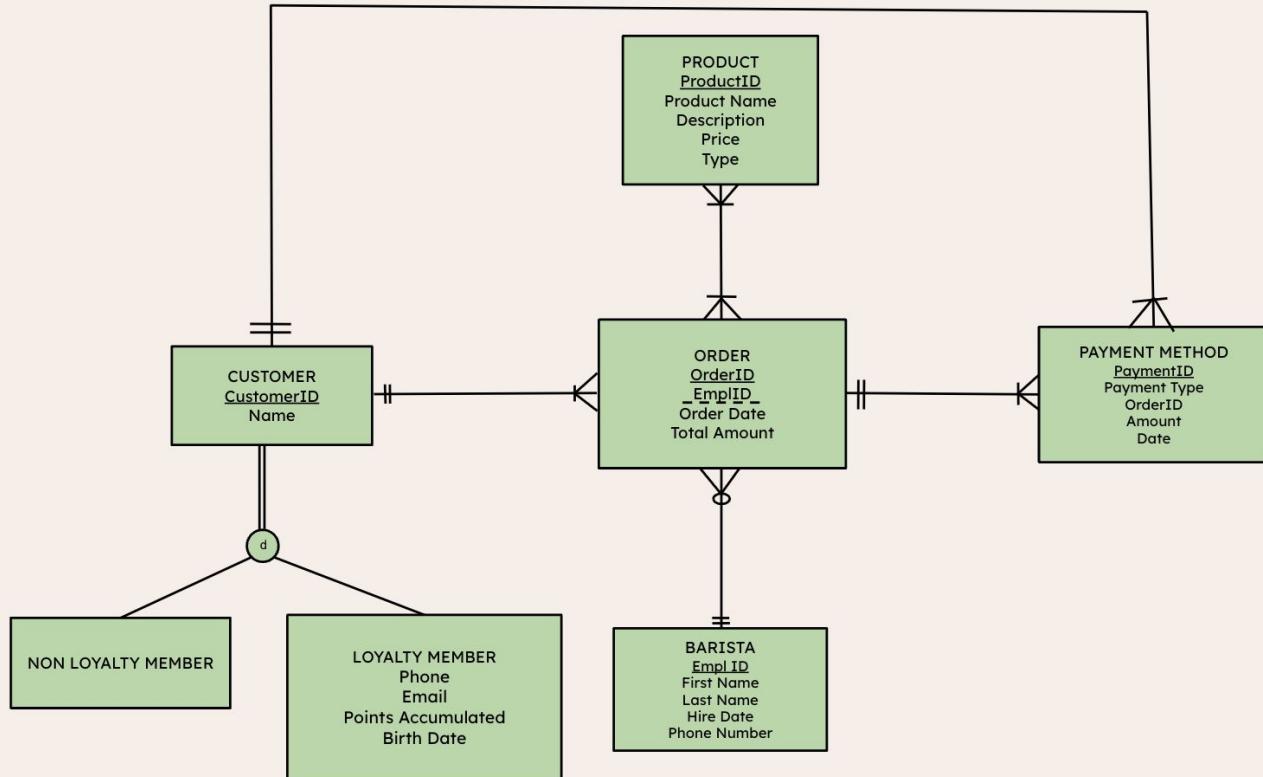
- Barista
- Customer
- Order
- Product
- Payment

Rules:

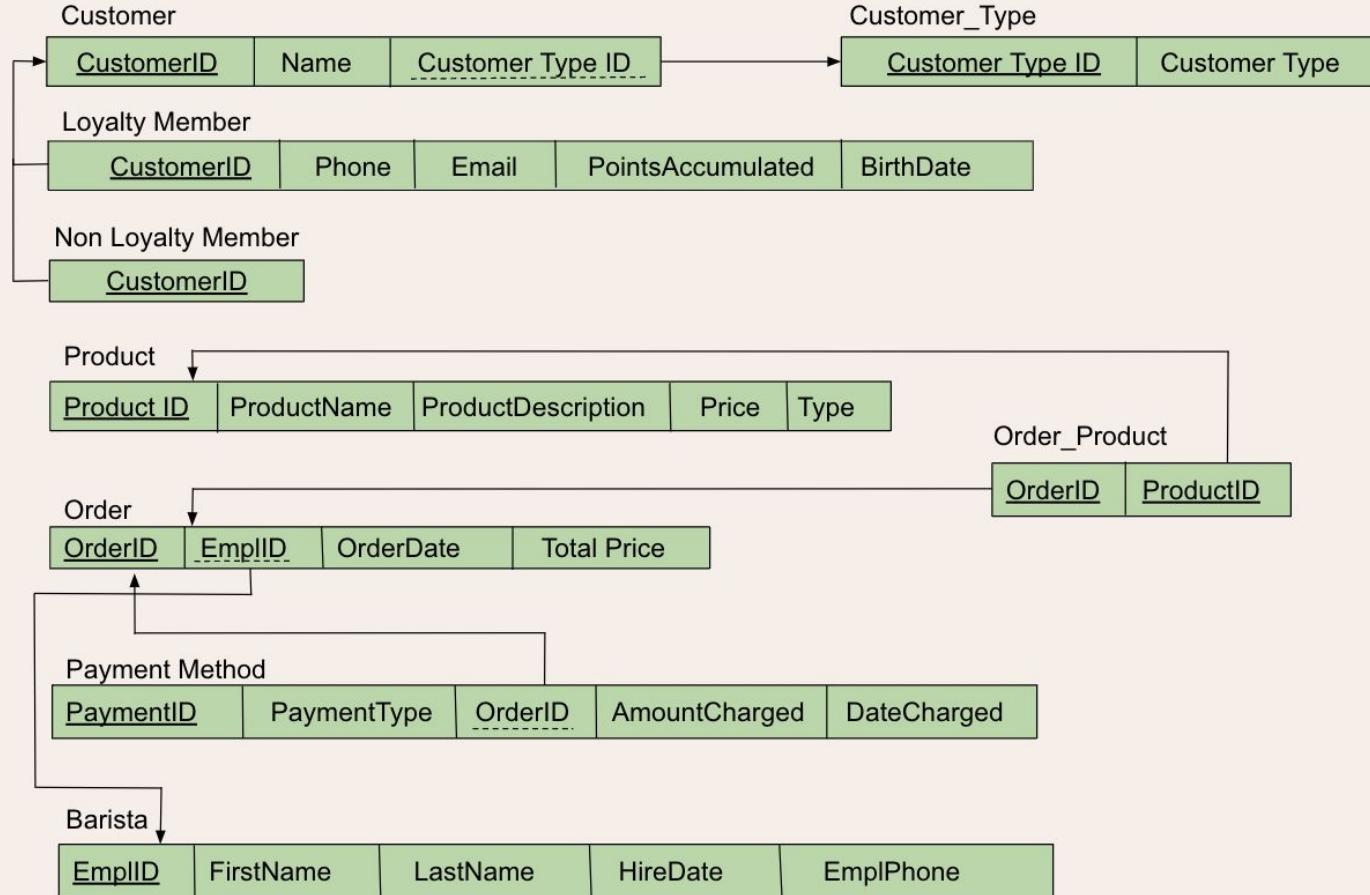
- Orders must be associated with at least one product and may have multiple payment methods.
- Each product has a name, price, description, and type, and may be part of multiple orders.
- Customers may be defined as a loyalty member and may place multiple orders. They may have one or many payment methods.
- A payment method can be associated with one or more orders.
- Barista can prepare multiple orders.

Logical Design

ER Diagram



Transforming the ER Model to Relations



Implementation

- Primary Keys:
 - INT AUTO_INCREMENT PRIMARY KEY
- Foreign Keys:
 - FOREIGN KEY (key) REFERENCES Parent_Table(key)
- Character Variables:
 - VARCHAR(255) NOT NULL
 - ENUM
- Integer Variables:
 - INT NOT NULL
 - Decimal Variables: DECIMAL(5, 2)
- Misc.
 - DATE

Database Demonstration

```
CREATE TABLE Customer_Type (
    customer_type_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_type ENUM('Loyalty', 'Non-Loyalty') NOT NULL
);

CREATE TABLE Customer (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255) NOT NULL,
    customer_type_id INT NOT NULL,
    FOREIGN KEY (customer_type_id) REFERENCES Customer_Type(customer_type_id)
);

CREATE TABLE Loyalty_Member (
    customer_loyalty_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT NOT NULL,
    phone VARCHAR(255) NOT NULL UNIQUE,
    email VARCHAR(255) NOT NULL UNIQUE,
    points_accumulated INT NOT NULL,
    birth_date DATE NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES Customer (customer_id)
);
```

```
-- INSERT STATEMENTS --
INSERT INTO Customer_Type (customer_type)
VALUES ('Loyalty'), ('Non-Loyalty');

SELECT * FROM Customer_Type;

INSERT INTO Customer (first_name, customer_type_id)
VALUES
    ('Alice', 1),
    ('Bob', 2),
    ('Charlie', 1),
    ('David', 2),
    ('Eve', 1),
    ('Frank', 2),
    ('Grace', 1),
    ('Hank', 2),
    ('Ivy', 1),
    ('Jack', 2);

INSERT INTO Loyalty_Member (customer_id, phone, email, points_accumulated, birth_date)
VALUES
    (1, '123-456-7890', 'alice@gmail.com', 100, '1990-01-01'),
    (3, '234-567-8901', 'charlie@hotmail.com', 200, '1985-08-02'),
    (5, '345-678-9012', 'eve@yahoo.com', 300, '1995-03-03'),
    (7, '805-789-0123', 'grace@calpoly.edu', 150, '1992-04-04'),
    (9, '567-890-1234', 'ivy@gmail.com', 250, '1988-05-05'),
    (2, '456-789-0123', 'bob@gmail.com', 50, '1987-06-15'),
    (4, '949-890-1234', 'david@hotmail.com', 75, '1979-09-20'),
    (6, '678-901-2345', 'frank@gmail.com', 100, '1992-03-10'),
    (8, '789-012-3456', 'hank@calpoly.cedu', 125, '1984-11-25'),
    (10, '890-123-4567', 'jack@calpoly.edu', 150, '1996-08-30');
```

Database Demonstration

#3. Top Performing Baristas by Total Sales

```
SELECT b.first_name, b.last_name, SUM(pi.total_price) AS total_sales
FROM Purchase_Instance pi JOIN Barista b
ON pi.empl_id = b.empl_id
GROUP BY b.first_name, b.last_name
ORDER BY total_sales DESC;
```

#5. Customers who have used gift cards

```
SELECT c.customer_id, c.first_name
FROM Customer c
WHERE c.customer_id IN (
    SELECT pm.order_id
    FROM Payment_Method pm
    WHERE pm.payment_type = 'Gift Card'
);
```

Discussion

- Test Queries
 1. Total sales by product type
 2. Average order value by month
 3. Top performing baristas by total sales
 4. Most loyal customers
 5. Customers who have used gift cards