Semester and Year

The customer loyalty program

COMP/MIS-302 Database Management Systems

Group Participant(s): Kulakov Dmitrii

*NOTE: This documentation report should be submitted within the project deadline in a compressed file along with the database file that proves the implementation of the database.*

Table of Contents

[The Case And Application Requirements 3](#_Toc39247962)

[1.1 The Case/Business 4](#_Toc39247963)

[1.2 Application Requirements 5](#_Toc39247964)

[Step 1 – Create Data Model From Application Requirements 6](#_Toc39247965)

[2.1 Data Model 7](#_Toc39247966)

[2.2 Supportive Documentation 8](#_Toc39247967)

[Step 2 – Transform Data Model Into Database Design 9](#_Toc39247968)

[2.1 Database Design 10](#_Toc39247969)

[2.2 Normalization 11](#_Toc39247970)

[2.3 Data Dictionary 12](#_Toc39247971)

[2.4 Minimum Cardinality Enforcement 13](#_Toc39247972)

[Step 3 – Database Implementation 14](#_Toc39247973)

[3.1 Database Creation 15](#_Toc39247974)

[3.2 Insertion of Sample/Test Data 16](#_Toc39247975)

[Step 4 – Query Processing 17](#_Toc39247976)

[4.1 Query Implementation 18](#_Toc39247977)

# The Case And Application Requirements

## 1.1 The Case/Business

The customer loyalty program aims to enhance customer retention and engagement by offering rewards for various interactions with the business. Customers can sign up for the loyalty program, receive rewards upon subscription, earn points through purchases or promotions, and redeem those points for discounts or gifts. The system should also allow customers to check their loyalty status, available points, and rewards online.

## 1.2 Application Requirements

User Registration: Customers can sign up for the loyalty program using their information such as phone number, email, etc.

Initial Rewards: Customers receive rewards upon subscribing to the loyalty program.

Earning Rewards: Customers earn rewards through purchases or promotional activities.

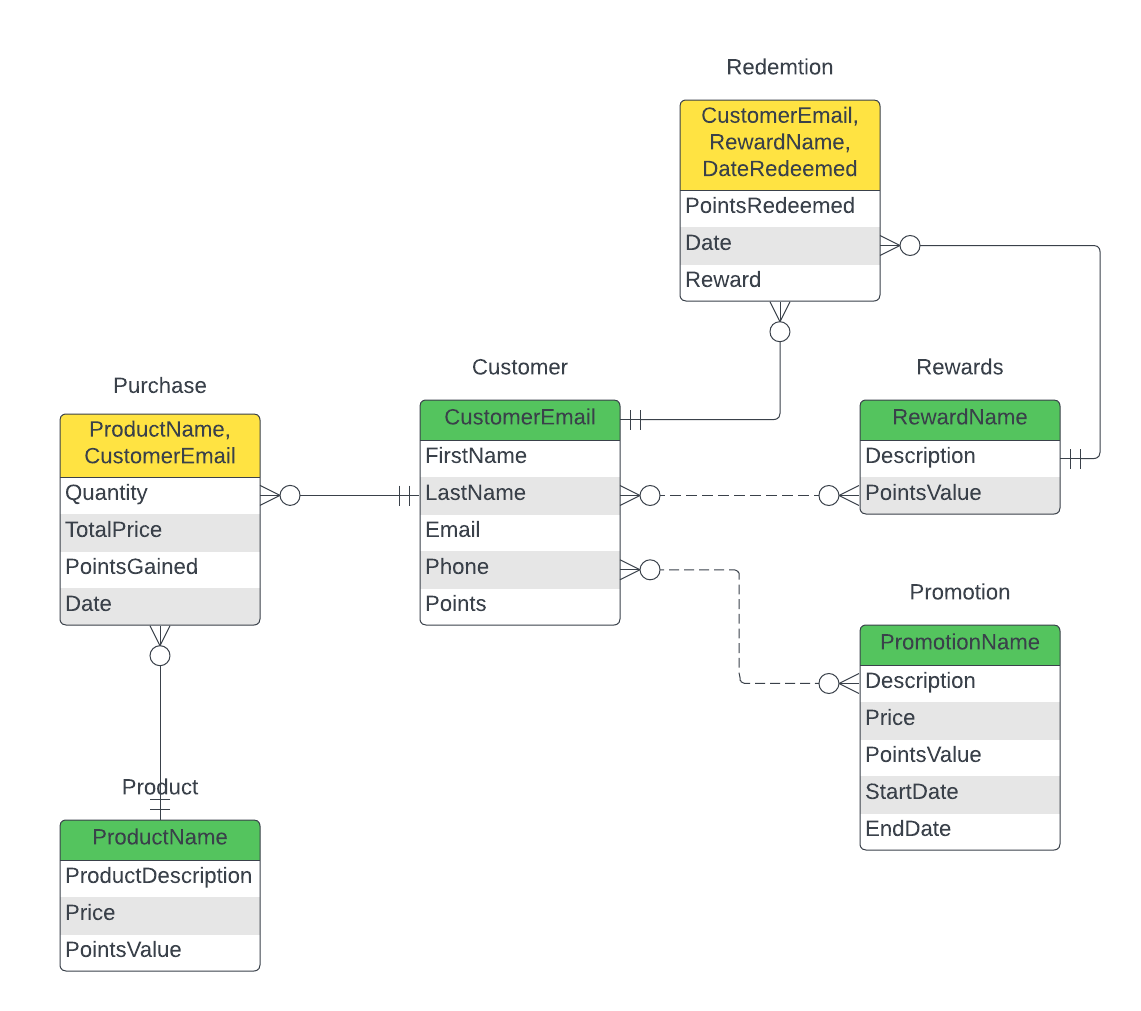
Points Redemption: Customers can redeem accumulated points for discounts or gifts.

Online Status Check: Customers should be able to check their loyalty status, available points, and rewards online.

Online Redemption: Customers can redeem points online by selecting available rewards.

# Step 1 – Create Data Model From Application Requirements

## 2.1 Data Model

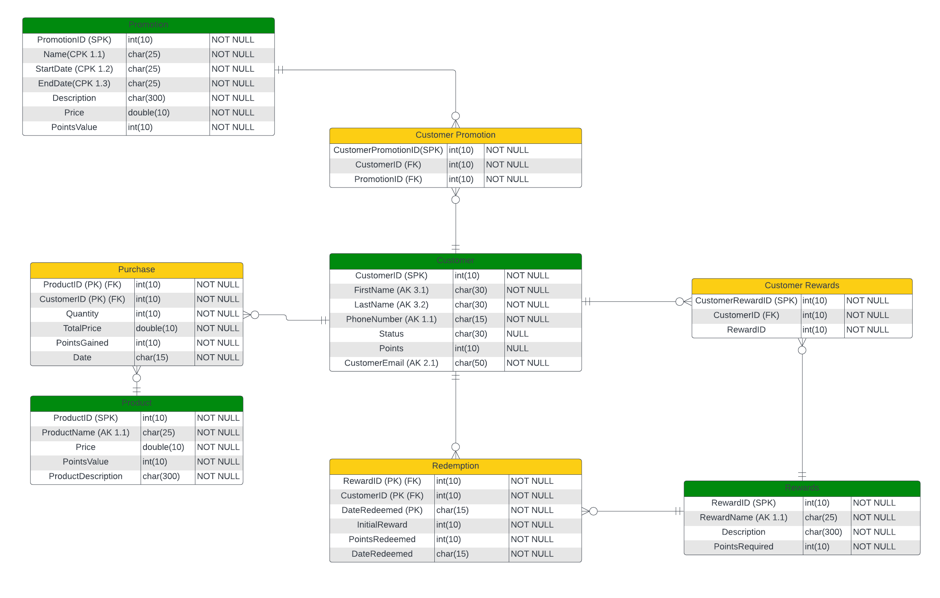


## 2.2 Supportive Documentation

The data model is designed to provide a structured framework for managing customer information, tracking purchases, offering rewards, and running promotional activities. It enables the business to enhance customer engagement, encourage repeat purchases, and analyze customer behavior for strategic decision-making. The system aims to build and maintain customer loyalty by rewarding and incentivizing customers based on their interactions with the business.

# Step 2 – Transform Data Model Into Database Design

## 2.1 Database Design



## 2.2 Normalization

Discuss normalization in relation to your relational schema.

## 2.3 Data Dictionary

|  |  |  |
| --- | --- | --- |
| Column | Data Type | NULL/NOT NULL |
| CustomerID | INT | NOT NULL |
| FirstName | VARCHAR | NOT NULL |
| LastName | VARCHAR | NOT NULL |
| PhoneNumber | VARCHAR | NOT NULL |
| Status | VARCHAR | NULL |
| Points | INT | NULL |
| CustomerEmail | VARCHAR | NOT NULL |

Table ‘Customer’

Table ‘Reward’

|  |  |  |
| --- | --- | --- |
| RewardID | INT | NOT NULL |
| RewardName | VARCHAR | NOT NULL |
| Description | VARCHAR | NOT NULL |
| PointsRequired | INT | NOT NULL |

Table ‘Promotion’

|  |  |  |
| --- | --- | --- |
| PromotionID | INT | NOT NULL |
| Name | VARCHAR | NOT NULL |
| StartDate | VARCHAR | NOT NULL |
| EndDate | VARCHAR | NOT NULL |
| Description | VARCHAR | NOT NULL |
| Price | DOUBLE | NOT NULL |
| PointsValue | INT | NOT NULL |

Table ‘Redemption’

|  |  |  |
| --- | --- | --- |
| RewardID | INT | NOT NULL |
| CustomerID | INT | NOT NULL |
| DateRedeemed | VARCHAR | NOT NULL |
| InitialReward | VARCHAR | NOT NULL |
| PointsRedeemed | INT | NOT NULL |
| DateRedeemed | VARCHAR | NOT NULL |

Table ‘Product’

|  |  |  |
| --- | --- | --- |
| ProductID | INT | NOT NULL |
| ProductName | VARCHAR | NOT NULL |
| Price | DOUBLE | NOT NULL |
| PointsValue | INT | NOT NULL |
| ProductDescription | VARCHAR | NOT NULL |

Table ‘Purchase’

|  |  |  |
| --- | --- | --- |
| ProductID | INT | NOT NULL |
| CustomerID | INT | NOT NULL |
| Quantity | INT | NOT NULL |
| TotalPrice | DOUBLE | NOT NULL |
| PointsGained | INT | NOT NULL |
| Date | VARCHAR | NOT NULL |

Table ‘Customer Promotion’

|  |  |  |
| --- | --- | --- |
| CustomerPromotionID | INT | NOT NULL |
| CustomerID | INT | NOT NULL |
| PromotionID | INT | NOT NULL |

Table ‘Customer Rewards’

|  |  |  |
| --- | --- | --- |
| CustomerRewardID | INT | NOT NULL |
| CustomerID | INT | NOT NULL |
| RewardID | INT | NOT NULL |

## 2.4 Minimum Cardinality Enforcement

Using a table for each relationship describe all of the necessary minimum cardinality enforcement actions. Refer to figures 6.28(a) and 6.28(b) (slides 45 and 46 of Chapter 6).

# Step 3 – Database Implementation

## 3.1 Database Creation

Create table of ‘Customer’:

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

PhoneNumber VARCHAR(20),

Status VARCHAR(20),

Points INT,

CustomerEmail VARCHAR(100) UNIQUE,

CONSTRAINT AK\_Customer\_FirstName\_LastName UNIQUE (FirstName, LastName),

CONSTRAINT AK\_Customer\_PhoneNumber UNIQUE (PhoneNumber),

CONSTRAINT AK\_Customer\_CustomerEmail UNIQUE (CustomerEmail)

);

Create table of ‘Rewards’:

CREATE TABLE Rewards (

RewardID INT PRIMARY KEY,

RewardName VARCHAR(50),

Description VARCHAR(255),

PointsRequired INT,

CONSTRAINT AK\_Rewards\_RewardName UNIQUE (RewardName)

);

Create table of ‘Promotion’:

CREATE TABLE Promotion (

PromotionID INT PRIMARY KEY,

Name VARCHAR(50),

StartDate DATE,

EndDate DATE,

Description VARCHAR(255),

Price DECIMAL(10,2),

PointsValue INT

);

Create table of ‘Redemption’:

CREATE TABLE Redemption (

RewardID INT,

CustomerID INT,

DateRedeemed DATE,

InitialReward VARCHAR(255),

PointsRedeemed INT,

PRIMARY KEY (RewardID, CustomerID, DateRedeemed),

FOREIGN KEY (RewardID) REFERENCES Rewards(RewardID),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

Create table of ‘Product’:

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(50),

Price DECIMAL(10,2),

PointsValue INT,

ProductDescription VARCHAR(255),

CONSTRAINT AK\_Product\_ProductName UNIQUE (ProductName)

);

Create table of ‘Purchase’:

CREATE TABLE Purchase (

ProductID INT,

CustomerID INT,

Quantity INT,

TotalPrice DECIMAL(10,2),

PointsGained INT,

Date DATE,

PRIMARY KEY (ProductID, CustomerID, Date),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

Create table of ‘CustomerPromotion’:

CREATE TABLE CustomerPromotion (

CustomerPromotionID INT PRIMARY KEY,

CustomerID INT,

PromotionID INT,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (PromotionID) REFERENCES Promotion(PromotionID)

);

Create table of ‘CustomerRewards’:

CREATE TABLE CustomerRewards (

CustomerRewardID INT PRIMARY KEY,

CustomerID INT,

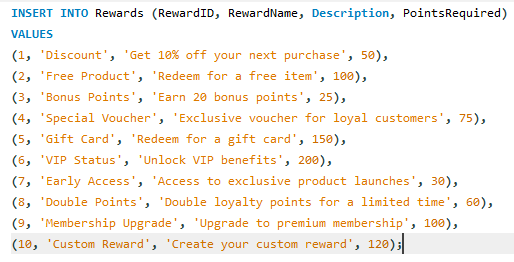
RewardID INT,

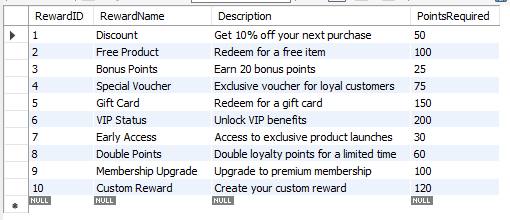
FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (RewardID) REFERENCES Rewards(RewardID)

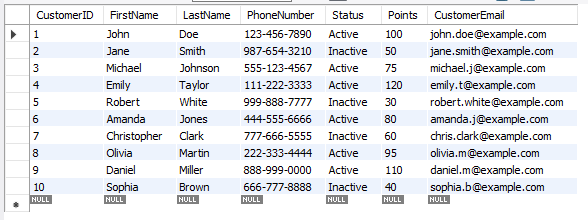
);

## 3.2 Insertion of Sample/Test Data

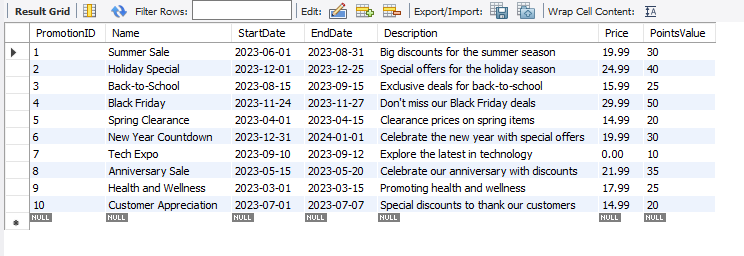


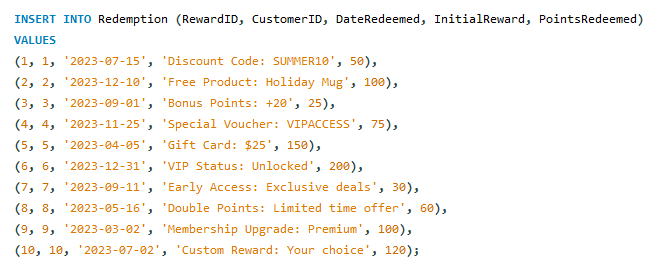


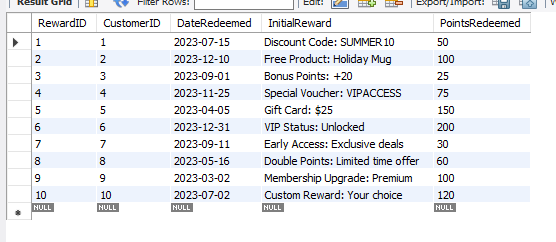


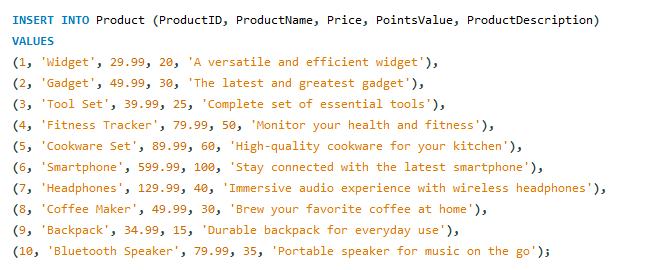


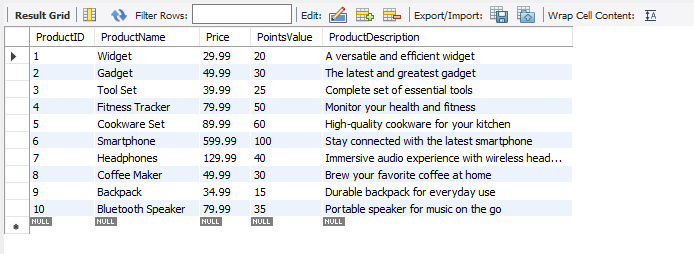


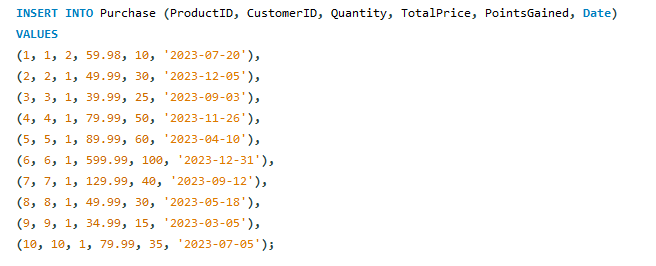


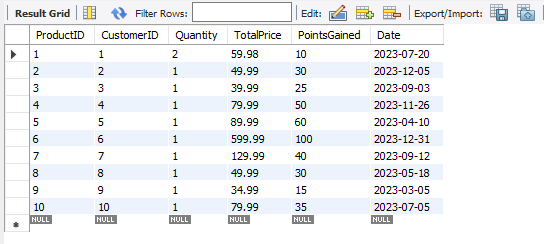


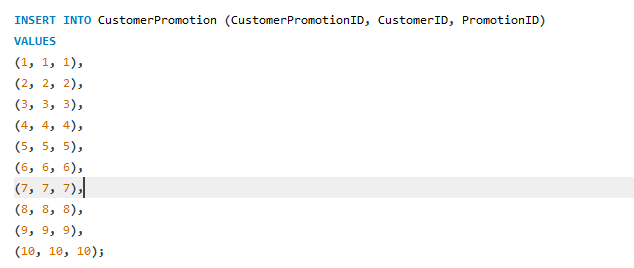


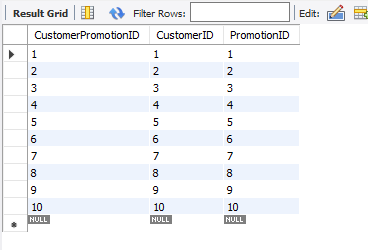


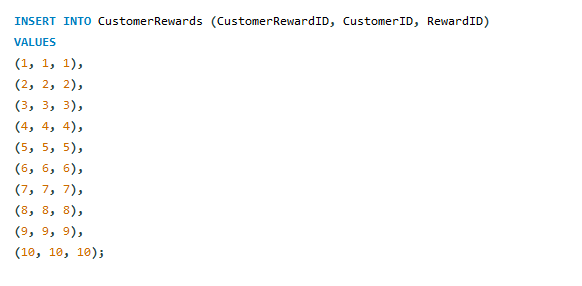


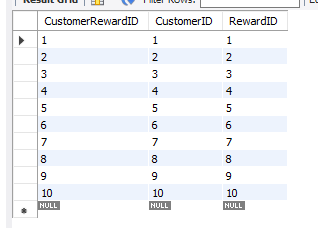










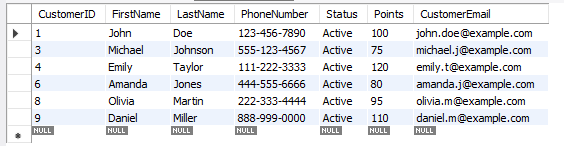


# Step 4 – Query Processing

## 4.1 Query Implementation

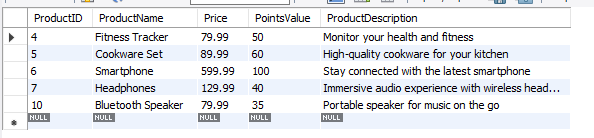
1. Retrieve details of all active customers.

SELECT \* FROM Customer WHERE Status = 'Active';



1. Retrieve products with a price higher than $50.

SELECT \* FROM Product WHERE Price > 50.00;



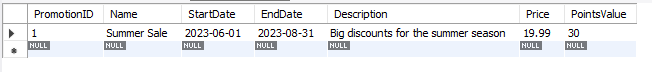
1. Find promotions that overlap with a given date.

SELECT \*

FROM Promotion

WHERE ('2023-07-10' BETWEEN StartDate AND EndDate)

OR ('2023-07-15' BETWEEN StartDate AND EndDate);



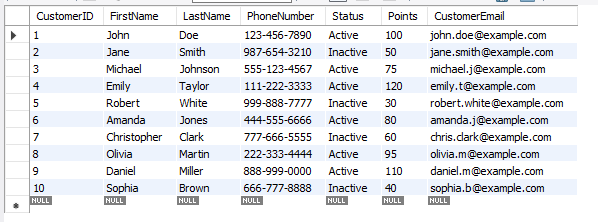
1. Count the total number of customers.

SELECT COUNT(\*) AS CustomerCount FROM Customer;



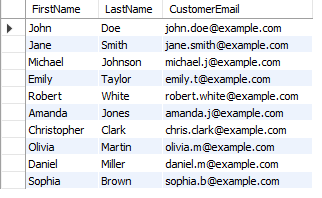
1. Retrieve Customer Information.

SELECT \* FROM Customer;



1. Retrieve customer names and email addresses.

SELECT FirstName, LastName, CustomerEmail FROM Customer;



1. Calculate the average points earned by customers.

SELECT AVG(Points) AS AveragePoints FROM Customer;



1. Display rewards earned by each customer.

SELECT Customer.FirstName, Customer.LastName, Rewards.RewardName

FROM Customer

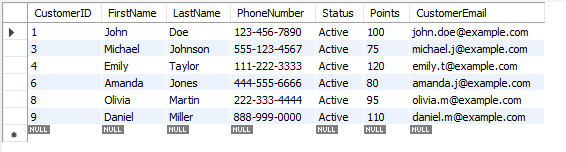
LEFT JOIN CustomerRewards ON Customer.CustomerID = CustomerRewards.CustomerID

LEFT JOIN Rewards ON CustomerRewards.RewardID = Rewards.RewardID;



1. Retrieve details of all active customers who has more than 40 points.

SELECT \* FROM Customer WHERE Status = 'Active' AND Points > 40;



1. Retrieve first name and last name of customer who has active status and 80 points.

SELECT FirstName, Lastname

FROM customer

WHERE Status = 'Active' AND Points = 80;

