**Database Management System**

**Project Report**

**Project name: Milkshake Stores Management**

**Course: BTech CSE AI (4th Semester)**

# Shoolini University of Biotechnology and Management Sciences

**Problem statement:** Design and implement a database system for a milkshake shop that includes entities such as Flavours, milkshakes, stores, customers, orders, and rewards. The system should enable tracking of milkshake flavours, store locations, customer orders, and offer rewards to customers based on certain criteria.

**Tables created:** Stores, Flavours, Milkshakes, Customers, Orders, and Rewards.

* The **Stores** table contains information about the name and location of the stores as there can be more than one store.
* The **Flavours** table contains information about the flavours.
* **Milkshakes** table contains data about the different types of shakes the store serves.
* **Customers** table contains the information of the customer whereas **Orders** table is used to store information about the orders given by the customers.
* It shows which rewards are given to which customers in the **Rewards** table.

**Primary keys:** StoreID, FlavourID, MilkshakeID, CustomerID, OrderID, and RewardID.

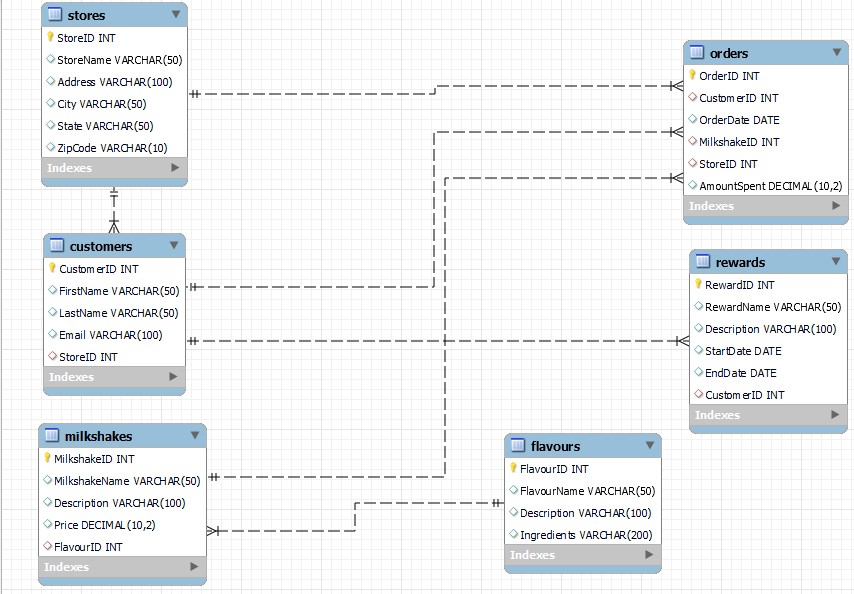
**Foreign keys:** StoreID, FlavourID, MilkshakeID, and CustomerID.

**Relationship between entities:**

* The Flavours entity and Milkshakes entity have a one to many relationship, where one flavour can be used in multiple milkshakes.
* The Customers entity and Orders entity have a one to many relationship, where one customer can place multiple orders.
* The Customers entity and Rewards entity have a one to many relationship, where one customer can have multiple rewards.

**Database Management System used:** MySQL

**Data flow diagram:**



**Performing queries on the database:**

1. **Amount spent by the customers:** The following query shows the amount customer spent based on which the rewards will be given to the customer.

**Query:**

SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName,

SUM(Orders.AmountSpent) AS TotalAmountSpent

FROM Customers

INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName;

**Output:**



1. **Accessing information about top customers:** The following query will return top 5 customers based on the amount they spend.

SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName,

SUM(Orders.AmountSpent) AS TotalAmountSpent

FROM Customers

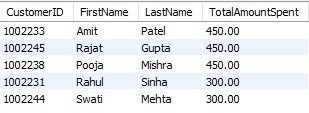
INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName

ORDER BY TotalAmountSpent DESC

LIMIT 5;

**Output:**



1. **Rewards for the customers:** The following query is used to access information about the reward given to a certain customer.

SELECT \* FROM Rewards WHERE CustomerID = 1002233;

**Output:**



1. **Updating price of the milkshake:** The following query is used to update the prices of the milkshakes.

UPDATE Milkshakes

SET Price = 200.00

WHERE MilkshakeID = 101;