**FOR MySQL ONLY**

1. **Sales Register:**
   * Query to retrieve the total number of sales transactions recorded in the Sales Register table.
2. **Purchase Register:**
   * Procedure to insert a new purchase record into the Purchase Register table for a specific product (Prod\_ID) with the quantity purchased and the purchase amount.
3. **Loyalty Program Member:**
   * Query to check if a specific customer (Cust\_ID) is enrolled in the loyalty program by searching for their ID in the Loyalty Program Members table.
4. **Customer:**
   * Procedure to update the customer's phone number (Cust\_Ph) in the Customer table based on their ID (Cust\_ID).

Write a PL/SQL block for MySQL to give 10% more discount on the original amount if one of the items in the bill is 'Books' and also display the updated table as the output, these are the values I have entered originally

--Bills

DELIMITER //

CREATE PROCEDURE ApplyExtraDiscount()

BEGIN

UPDATE Bills

SET Discount = Discount + (Amount \* 0.1)

WHERE Item1 = 'Books' OR Item2 = 'Books' OR Item3 = 'Books' OR Item4 = 'Books' OR Item5 = 'Books';

SELECT \* FROM Bills;

END //

DELIMITER ;

CALL ApplyExtraDiscount();

--Loyalty Program

DELIMITER //

CREATE PROCEDURE CheckLoyaltyEnrollment(IN p\_cust\_id INT)

BEGIN

DECLARE enrolled INT;

-- Check if the customer is enrolled in the loyalty program

SELECT COUNT(\*) INTO enrolled

FROM Loyalty\_members

WHERE Cust\_ID = p\_cust\_id;

IF enrolled > 0 THEN

SELECT 'Customer is enrolled in the loyalty program' AS Status;

ELSE

SELECT 'Customer is not enrolled in the loyalty program' AS Status;

END IF;

END;

//

DELIMITER ;

CALL CheckLoyaltyEnrollment(1); -- Assuming 1 is the customer ID you want to check  
  
--Sales Register  
  
DELIMITER //

CREATE PROCEDURE GetTotalSalesTransactions()

BEGIN

DECLARE total\_sales INT;

-- Get the total number of sales transactions

SELECT COUNT(\*) INTO total\_sales

FROM SalesRegister;

SELECT total\_sales AS Total\_Sales\_Transactions;

END;

//

DELIMITER ;

CALL GetTotalSalesTransactions();

--Purchase Register

DELIMITER //

CREATE PROCEDURE InsertPurchaseRecord(IN p\_prod\_id INT, IN p\_org\_name VARCHAR(50), IN p\_quantity INT, IN p\_amount DECIMAL(10, 2), IN p\_dop DATE)

BEGIN

DECLARE prod\_exists INT;

-- Check if the provided Prod\_ID exists in the Inventory table

SELECT COUNT(\*) INTO prod\_exists

FROM Inventory

WHERE Prod\_ID = p\_prod\_id;

-- If the provided Prod\_ID does not exist, return an error message

IF prod\_exists = 0 THEN

SELECT 'Error: Provided Prod\_ID does not exist in the Inventory table' AS Status;

ELSE

-- Insert a new purchase record into the Purchase Register table

INSERT INTO PurchaseRegister (Prod\_ID, Org\_name, Quantity, Amount, DoP)

VALUES (p\_prod\_id, p\_org\_name, p\_quantity, p\_amount, p\_dop);

SELECT 'New purchase record inserted successfully' AS Status;

END IF;

END;

//

DELIMITER ;

CALL InsertPurchaseRecord(6, 'ABC Suppliers', 30, 1200.00, '2024-03-25'); -- Example values for product ID, organization name, quantity, amount, and date of purchase  
  
  
--Customer  
DELIMITER //

CREATE PROCEDURE UpdateCustomerPhoneNumber(IN p\_cust\_id INT, IN p\_new\_ph VARCHAR(10))

BEGIN

-- Update the customer's phone number

UPDATE Customer

SET Cust\_Phone = p\_new\_ph

WHERE Cust\_ID = p\_cust\_id;

SELECT 'Customer phone number updated successfully' AS Status;

END;

//

DELIMITER ;

CALL UpdateCustomerPhoneNumber(1, '9876543210'); -- Example values for customer ID and new phone number