**MySQL Exercise 4**

1.

-- Create CUSTOMER table

CREATE TABLE CUSTOMER (

MeterNumber VARCHAR(4),

MeterType CHAR(1),

PreviousReading INT,

CurrentReading INT,

CustomerType CHAR(1),

LastBillPayment CHAR(1)

);

-- Insert sample data into CUSTOMER table

INSERT INTO CUSTOMER (MeterNumber, MeterType, PreviousReading, CurrentReading, CustomerType, LastBillPayment)

VALUES

('0001', 'S', 1000, 1200, 'R', 'Y'),

('0002', 'T', 1500, 1750, 'I', 'N'),

('0003', 'S', 300, 450, 'A', 'Y'),

('0004', 'T', 400, 800, 'C', 'N');

-- Create CustomerBill table

CREATE TABLE CustomerBill (

MeterNumber VARCHAR(4),

UnitsUsed INT,

Rate FLOAT,

Amount FLOAT,

Surcharge FLOAT,

ExciseDuty FLOAT,

NetAmount FLOAT

);

-- Create BillSummary table

CREATE TABLE BillSummary (

TotalAmount FLOAT,

TotalSurcharge FLOAT,

TotalExcise FLOAT,

TotalNet FLOAT

);

DELIMITER //

CREATE PROCEDURE calculate\_bills()

BEGIN

DECLARE meter\_no VARCHAR(4);

DECLARE meter\_type CHAR(1);

DECLARE prev\_reading INT;

DECLARE curr\_reading INT;

DECLARE customer\_type CHAR(1);

DECLARE last\_bill\_payment CHAR(1);

DECLARE units\_used INT;

DECLARE rate FLOAT;

DECLARE amount FLOAT;

DECLARE surcharge FLOAT;

DECLARE excise FLOAT;

DECLARE net\_amount FLOAT;

DECLARE total\_amount FLOAT DEFAULT 0;

DECLARE total\_surcharge FLOAT DEFAULT 0;

DECLARE total\_excise FLOAT DEFAULT 0;

DECLARE total\_net FLOAT DEFAULT 0;

DECLARE finished INTEGER DEFAULT 0;

DECLARE customer\_cursor CURSOR FOR

SELECT MeterNumber, MeterType, PreviousReading, CurrentReading, CustomerType, LastBillPayment

FROM CUSTOMER;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;

-- Drop temporary tables if they exist

DROP TABLE IF EXISTS CustomerBill;

DROP TABLE IF EXISTS BillSummary;

-- Create tables to store results

CREATE TABLE CustomerBill (

MeterNumber VARCHAR(4),

UnitsUsed INT,

Rate FLOAT,

Amount FLOAT,

Surcharge FLOAT,

ExciseDuty FLOAT,

NetAmount FLOAT

);

CREATE TABLE BillSummary (

TotalAmount FLOAT,

TotalSurcharge FLOAT,

TotalExcise FLOAT,

TotalNet FLOAT

);

-- Open the cursor

OPEN customer\_cursor;

read\_loop: LOOP

FETCH customer\_cursor INTO meter\_no, meter\_type, prev\_reading, curr\_reading, customer\_type, last\_bill\_payment;

IF finished THEN

LEAVE read\_loop;

END IF;

-- Calculate units used

SET units\_used = curr\_reading - prev\_reading;

-- Determine rate based on customer type

CASE customer\_type

WHEN 'A' THEN SET rate = 1.00;

WHEN 'I' THEN SET rate = 1.25;

WHEN 'C' THEN SET rate = 1.50;

WHEN 'R' THEN SET rate = 1.30;

END CASE;

-- Calculate amount

SET amount = rate \* units\_used;

-- Calculate surcharge based on meter type

IF meter\_type = 'S' THEN

SET surcharge = 0.05 \* amount;

ELSE

SET surcharge = 0.10 \* amount;

END IF;

-- Calculate excise

SET excise = 0.30 \* (amount + surcharge);

-- Calculate net amount

SET net\_amount = amount + surcharge + excise;

-- Insert into CustomerBill table

INSERT INTO CustomerBill (MeterNumber, UnitsUsed, Rate, Amount, Surcharge, ExciseDuty, NetAmount)

VALUES (meter\_no, units\_used, rate, amount, surcharge, excise, net\_amount);

-- Accumulate totals

SET total\_amount = total\_amount + amount;

SET total\_surcharge = total\_surcharge + surcharge;

SET total\_excise = total\_excise + excise;

SET total\_net = total\_net + net\_amount;

END LOOP;

-- Close the cursor

CLOSE customer\_cursor;

-- Insert totals into BillSummary table

INSERT INTO BillSummary (TotalAmount, TotalSurcharge, TotalExcise, TotalNet)

VALUES (total\_amount, total\_surcharge, total\_excise, total\_net);

END //

DELIMITER ;

CALL calculate\_bills();