**Exercise 1: Control Structures**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DOB DATE,

Balance NUMBER ,

LastModified DATE

);

CREATE TABLE Accounts(

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE,

FOREIGN KEY ( CustomerID ) REFERENCES Customers(CustomerID)

);

CREATE TABLE Transactions (

TransactionID NUMBER PRIMARY KEY,

AccountID NUMBER ,

TransactionDate DATE ,

Amount NUMBER,

TransactionType VARCHAR2(10),

FOREIGN KEY (AccountID) REFERENCES Accounts( AccountID )

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

InterestRate NUMBER,

StartDate DATE,

EndDate DATE,

FOREIGN KEY ( CustomerID ) REFERENCES Customers(CustomerID)

);

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY ,

Name VARCHAR2(100),

Position VARCHAR2(50) ,

Salary NUMBER,

Department VARCHAR2(50) ,

HireDate DATE

);

INSERT INTO Customers ( CustomerID , Name , DOB, Balance , LastModified )

VALUES (6,'Rajesh Kumar', TO\_DATE('1972-11-10','YYYY-MM-DD'),9500,SYSDATE);

INSERT INTO Customers (CustomerID,Name,DOB,Balance, LastModified)

VALUES(7,'Priya Sharma',TO\_DATE('1985-08-22','YYYY-MM-DD'),15500,SYSDATE);

INSERT INTO Customers( CustomerID, Name , DOB , Balance , LastModified)

VALUES(8,'Anil Verma',TO\_DATE('1960-04-05','YYYY-MM-DD'),20000,SYSDATE);

INSERT INTO Customers (CustomerID,Name, DOB, Balance, LastModified)

VALUES(9,'Neha Singh', TO\_DATE('1992-01-14','YYYY-MM-DD'),7500,SYSDATE);

INSERT INTO Customers(CustomerID, Name, DOB,Balance,LastModified)

VALUES(10,'Vikram Patel',TO\_DATE('1975-07-30','YYYY-MM-DD'), 5000,SYSDATE);

INSERT INTO Accounts(AccountID,CustomerID,AccountType,Balance,LastModified)

VALUES(5,6,'Savings',9500,SYSDATE);

INSERT INTO Accounts( AccountID , CustomerID , AccountType , Balance , LastModified )

VALUES (6 , 7 ,'Savings' , 15500 , SYSDATE );

INSERT INTO Accounts( AccountID, CustomerID, AccountType,Balance, LastModified)

VALUES(7,8,'Savings',20000,SYSDATE);

INSERT INTO Accounts(AccountID,CustomerID,AccountType,Balance,LastModified)

VALUES(8,9,'Checking',7500,SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID,AccountType,Balance,LastModified)

VALUES (9,10,'Savings',5000,SYSDATE);

INSERT INTO Transactions(TransactionID,AccountID,TransactionDate,Amount,TransactionType)

VALUES(8,5,SYSDATE,1000,'Deposit');

INSERT INTO Transactions(TransactionID,AccountID,TransactionDate,Amount,TransactionType)

VALUES(9,6,SYSDATE,500,'Withdrawal');

INSERT INTO Transactions(TransactionID,AccountID,TransactionDate,Amount,TransactionType)

VALUES(10,7,SYSDATE,2000,'Deposit');

INSERT INTO Transactions(TransactionID,AccountID,TransactionDate,Amount,TransactionType)

VALUES(11,8,SYSDATE,1000,'Withdrawal');

INSERT INTO Transactions(TransactionID,AccountID,TransactionDate,Amount,TransactionType)

VALUES(12,9,SYSDATE,500,'Deposit');

INSERT INTO Loans(LoanID,CustomerID,LoanAmount,InterestRate,StartDate,EndDate)

VALUES(5,6,10000,6.5,SYSDATE,ADD\_MONTHS(SYSDATE,48));

INSERT INTO Loans(LoanID,CustomerID,LoanAmount,InterestRate,StartDate,EndDate)

VALUES(6,7,25000,5.75,SYSDATE,ADD\_MONTHS(SYSDATE,60));

INSERT INTO Loans(LoanID,CustomerID,LoanAmount,InterestRate,StartDate,EndDate)

VALUES(7,8,50000,7.25,SYSDATE,ADD\_MONTHS(SYSDATE,72));

INSERT INTO Employees(EmployeeID,Name,Position,Salary,Department,HireDate)

VALUES(6,'Suresh Raina','Branch Manager',85000,'Operations',TO\_DATE('2010-06-01','YYYY-MM-DD'));

INSERT INTO Employees(EmployeeID,Name,Position,Salary,Department,HireDate)

VALUES(7,'Pooja Nair','Clerk',40000,'Customer Service',TO\_DATE('2018-12-15','YYYY-MM-DD'));

INSERT INTO Employees(EmployeeID,Name,Position,Salary,Department,HireDate)

VALUES(8,'Manoj Das','IT Officer',62000,'IT',TO\_DATE('2016-09-20','YYYY-MM-DD'));

**Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.

* + **Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

BEGIN

FOR rec IN (SELECT LoanID, CustomerID, InterestRate,

TRUNC(MONTHS\_BETWEEN(SYSDATE, DOB)/12) AS Age

FROM Loans

JOIN Customers USING (CustomerID))

LOOP

IF rec.Age > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = rec.LoanID;

DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount for Customer ID: ' || rec.CustomerID ||

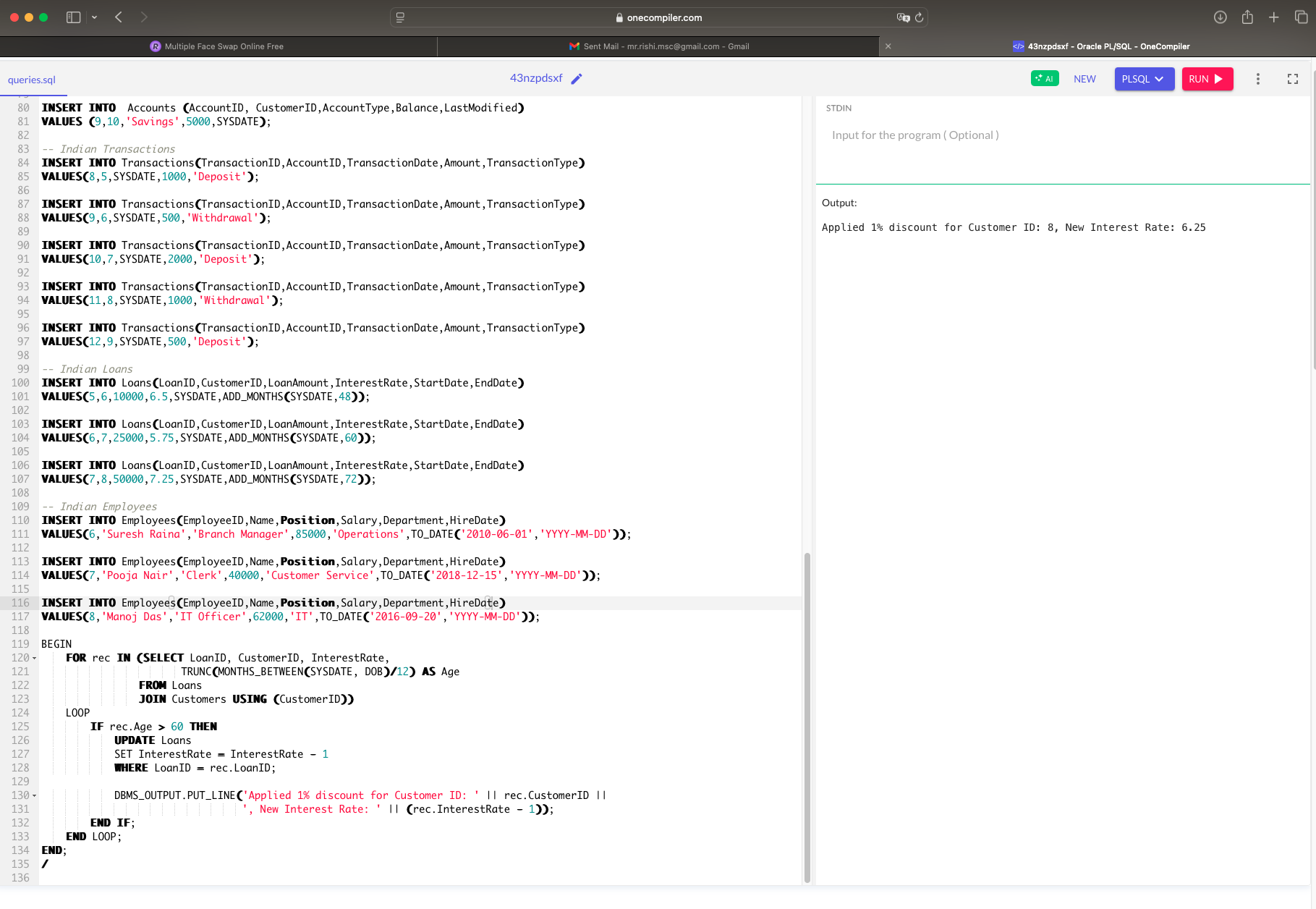
', New Interest Rate: ' || (rec.InterestRate - 1));

END IF;

END LOOP;

END;

/



**Scenario 2:** A customer can be promoted to VIP status based on their balance.

* + **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

ALTER TABLE Customers ADD IsVIP VARCHAR2(5);

BEGIN

FOR rec IN (SELECT CustomerID, Balance FROM Customers)

LOOP

IF rec.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

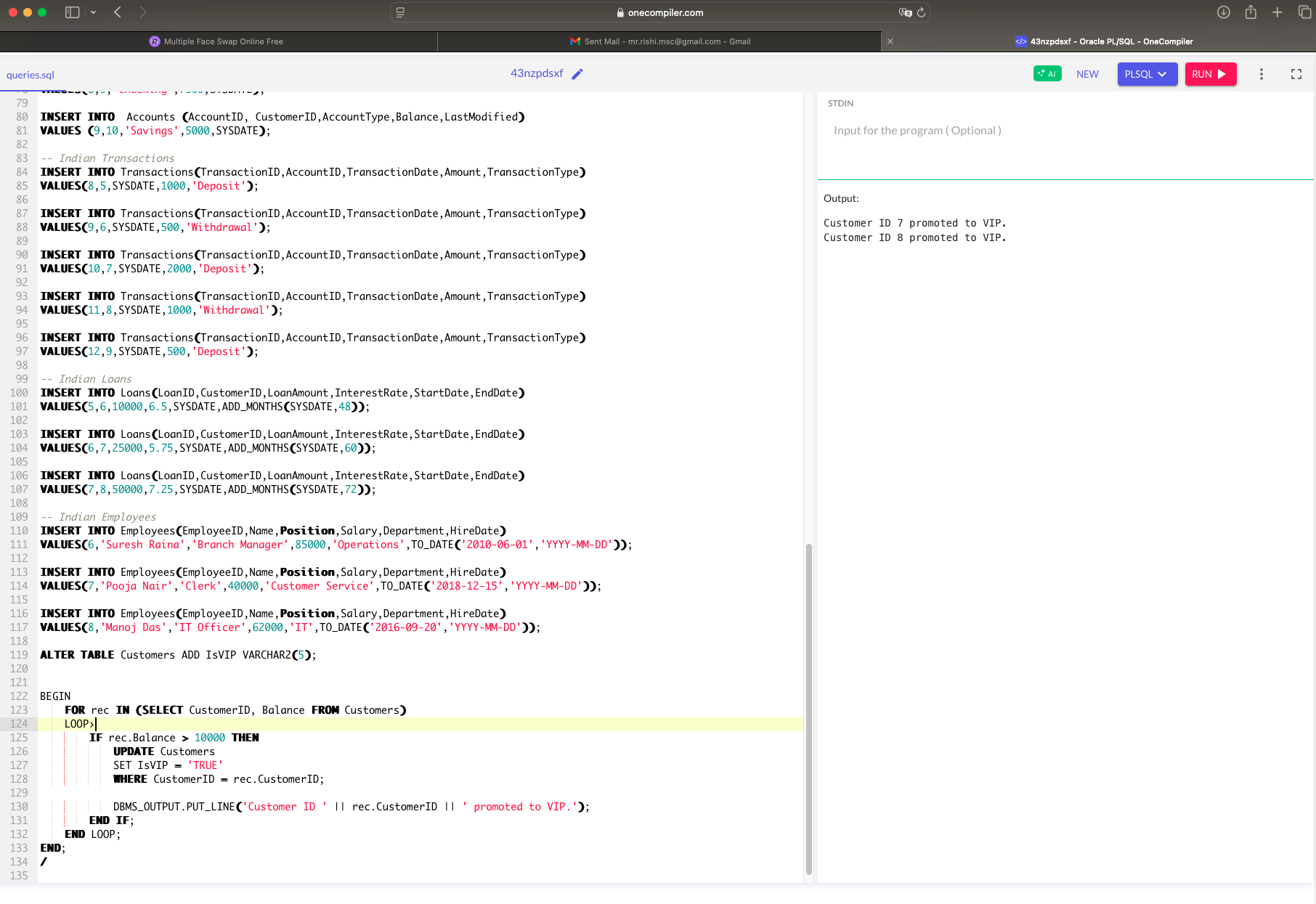
WHERE CustomerID = rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || rec.CustomerID || ' promoted to VIP.');

END IF;

END LOOP;

END;

/

**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

* + **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

BEGIN

FOR rec IN (SELECT LoanID, CustomerID, EndDate

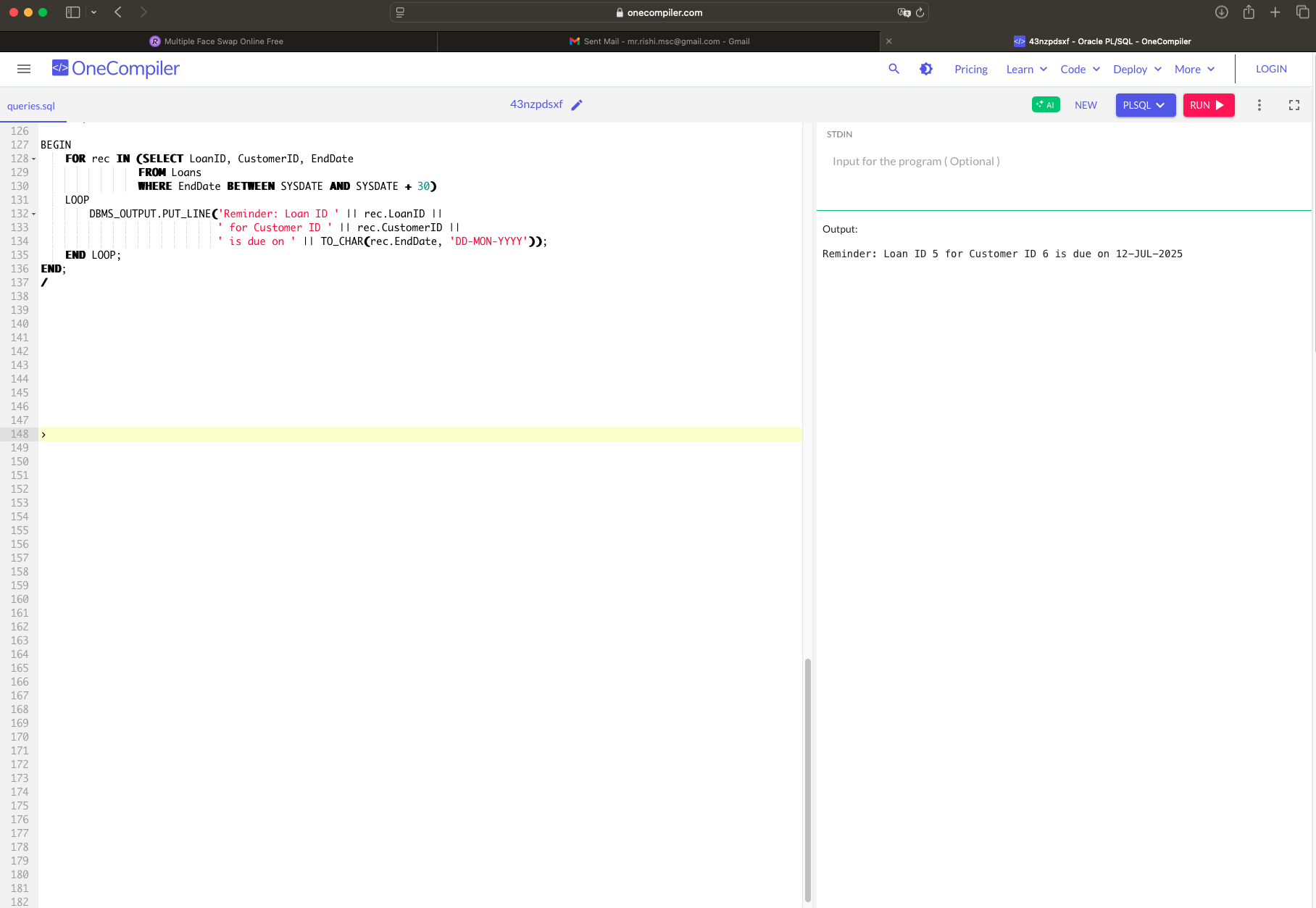
FROM Loans

WHERE EndDate BETWEEN SYSDATE AND SYSDATE + 30)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || rec.LoanID ||

' for Customer ID ' || rec.CustomerID ||

 ' is due on ' || TO\_CHAR(rec.EndDate, 'DD-MON-YYYY'));

END LOOP;

END;

/