**PL SQL\_EXERCISES**

**-Harini Baskar(6396726)**

**Exercise 1: Control Structures**

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

**Query:**

BEGIN

FOR cust IN (SELECT CustomerID, DOB FROM Customers) LOOP

IF MONTHS\_BETWEEN(SYSDATE, cust.DOB) / 12 > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE CustomerID = cust.CustomerID;

END IF;

END LOOP;

END;

**Output:**

****

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

**Query:**

UPDATE Customers

SET IsVIP = 'Y'

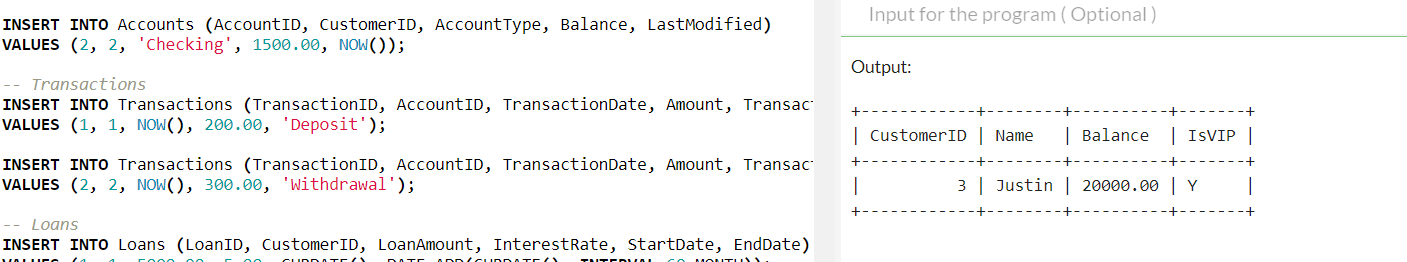
WHERE Balance > 10000;

SELECT CustomerID, Name, Balance, IsVIP

FROM Customers

WHERE IsVIP = 'Y';

**Output:**



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

**Query:**

SELECT

C.CustomerID,

C.Name,

L.LoanID,

L.EndDate,

CONCAT('Reminder: Loan ID ', L.LoanID, ' for ', C.Name,

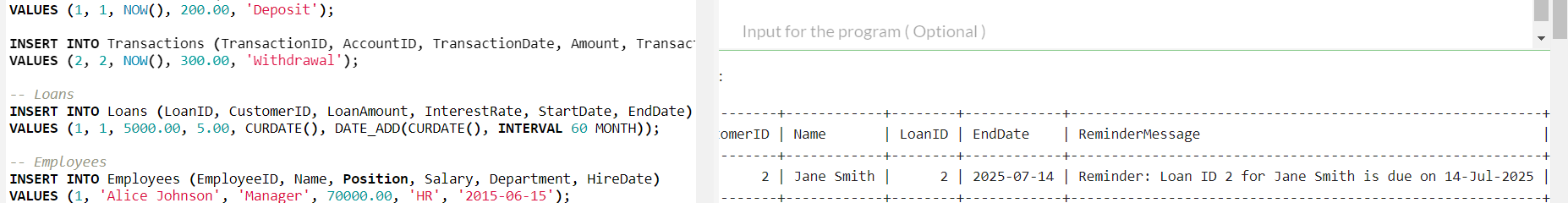
' is due on ', DATE\_FORMAT(L.EndDate, '%d-%b-%Y')) AS ReminderMessage

FROM Loans L

JOIN Customers C ON L.CustomerID = C.CustomerID

WHERE L.EndDate BETWEEN CURDATE() AND DATE\_ADD(CURDATE(), INTERVAL 30 DAY);

**Output:**

****

**Exercise 3: Stored Procedures**

**Scenario 1**: The bank needs to process monthly interest for all savings accounts.

**Query:**

UPDATE Accounts

SET Balance = Balance \* 1.01

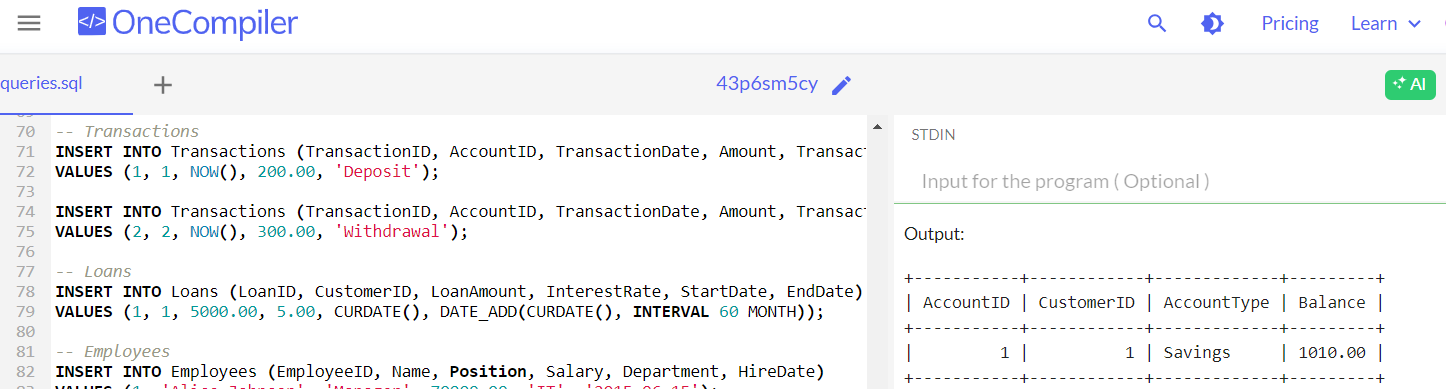
WHERE AccountType = 'Savings';

SELECT AccountID, CustomerID, AccountType, Balance

FROM Accounts

WHERE AccountType = 'Savings';

**Output:**

****

**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

**Query:**

UPDATE Employees

SET Salary = Salary + (Salary \* 0.10)

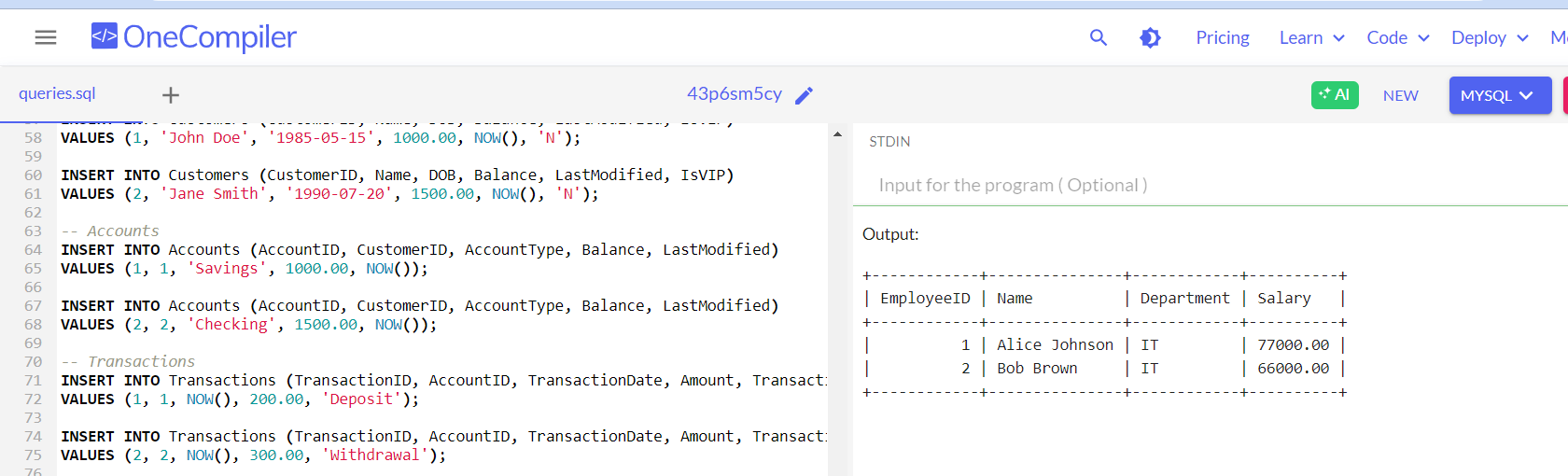
WHERE Department = 'IT';

SELECT EmployeeID, Name, Department, Salary

FROM Employees

WHERE Department = 'IT';

**Output:**

****

**Scenario 3:** Customers should be able to transfer funds between their accounts.

**Query:**

ALTER TABLE Transactions MODIFY TransactionID INT AUTO\_INCREMENT;

UPDATE Accounts

SET Balance = Balance - 500

WHERE AccountID = 1 AND Balance >= 500;

UPDATE Accounts

SET Balance = Balance + 500

WHERE AccountID = 2;

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

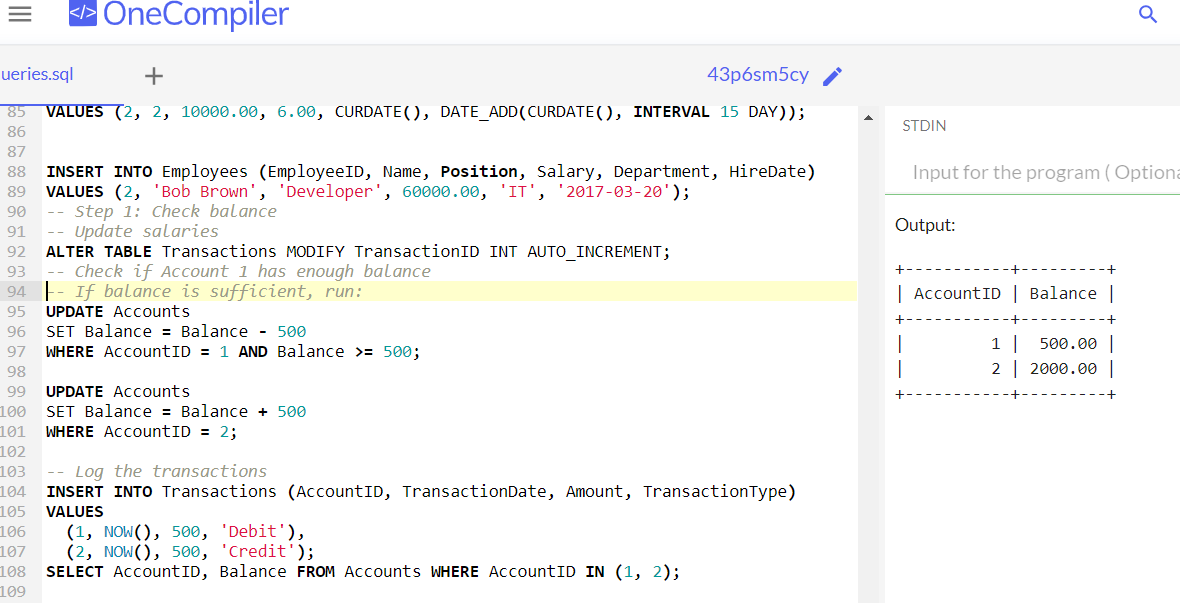
VALUES

(1, NOW(), 500, 'Debit'),

(2, NOW(), 500, 'Credit');

SELECT AccountID, Balance FROM Accounts WHERE AccountID IN (1, 2);

**Output:**

****