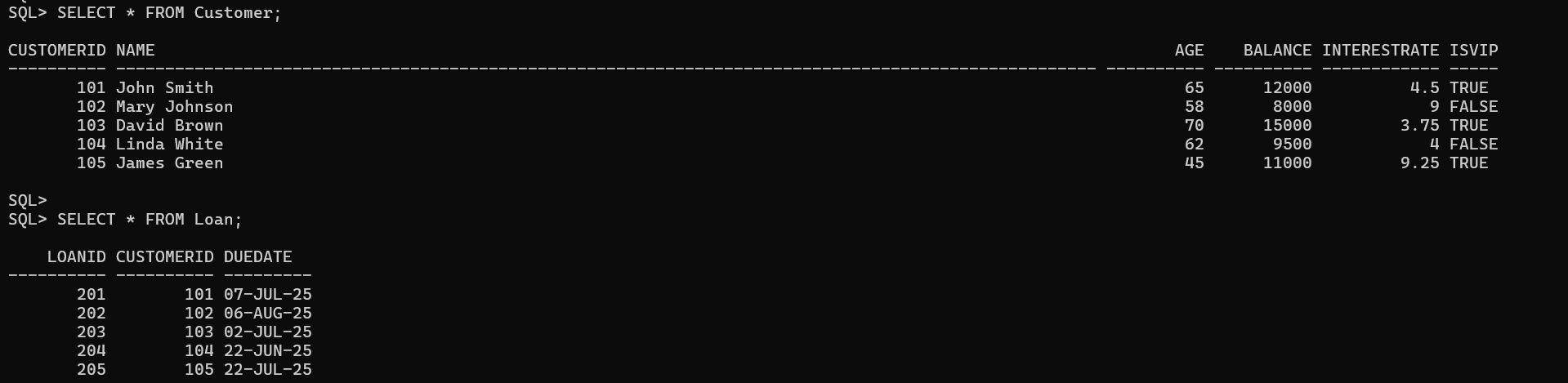
**PLSQL\_Exercises**

**Exercise 1: Control Structures**

****

**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.

**Code:**

BEGIN

FOR rec IN (SELECT CustomerID, InterestRate FROM Customer WHERE Age > 60) LOOP

UPDATE Customer

SET InterestRate = InterestRate - (InterestRate \* 0.01)

WHERE CustomerID = rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE(

'Updated InterestRate for Customer ID: ' || rec.CustomerID

);

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.

**Code:**

BEGIN

FOR rec IN (SELECT CustomerID, Balance FROM Customer WHERE Balance > 10000) LOOP

UPDATE Customer

SET IsVIP = 'TRUE'

WHERE CustomerID = rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE(

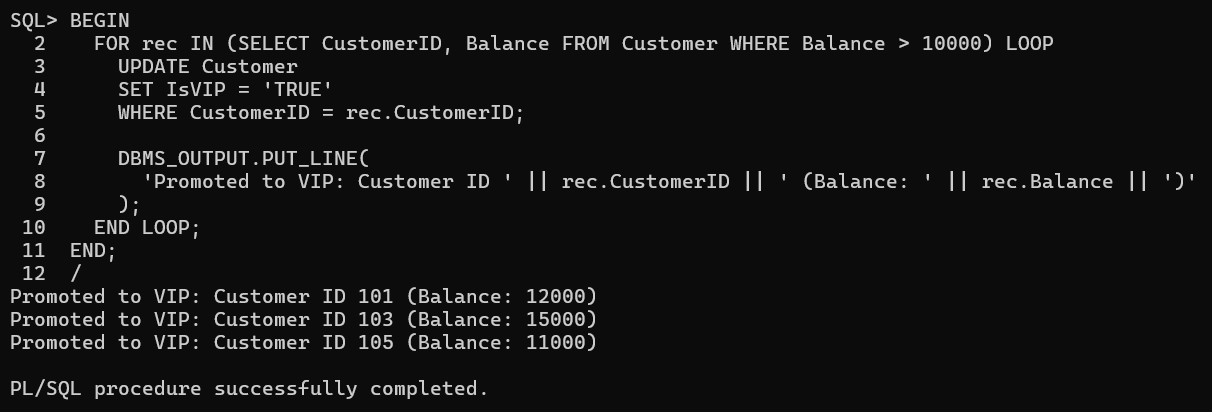
'Promoted to VIP: Customer ID ' || rec.CustomerID || ' (Balance: ' || rec.Balance || ')'

);

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.

**Code:**

BEGIN

FOR rec IN (

SELECT LoanID, CustomerID, DueDate

FROM Loan

WHERE DueDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: Loan ID ' || rec.LoanID ||

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

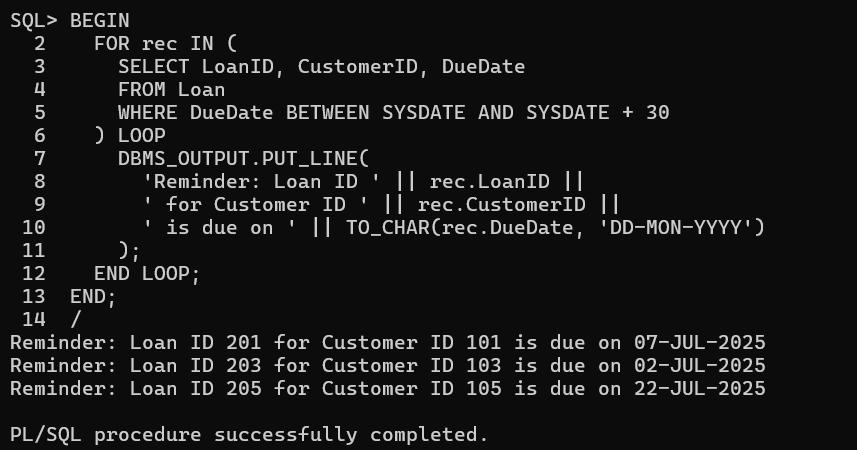
' is due on ' || TO\_CHAR(rec.DueDate, 'DD-MON-YYYY')

);

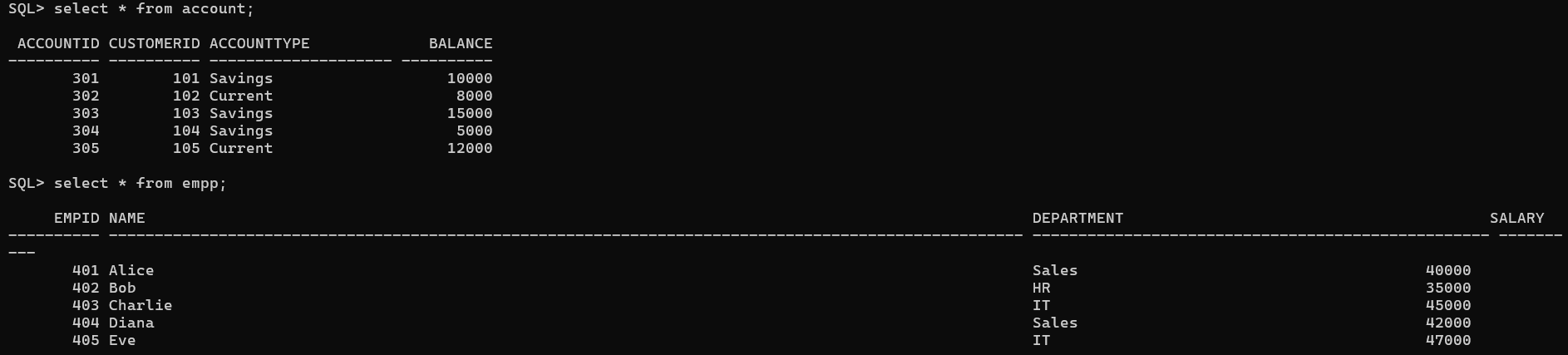
END LOOP;

END;

/



**Exercise 3: Stored Procedures**

****

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

* + **Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

**Code:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (SELECT AccountID, Balance FROM Account WHERE AccountType = 'Savings') LOOP

UPDATE Account

SET Balance = Balance + (Balance \* 0.01)

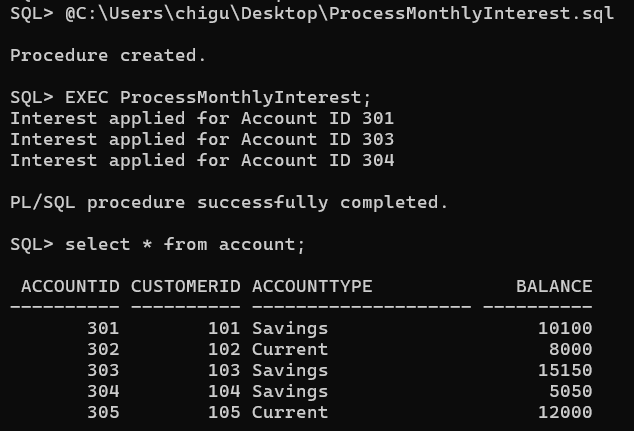
WHERE AccountID = acc.AccountID;

DBMS\_OUTPUT.PUT\_LINE('Interest applied for Account ID ' || acc.AccountID);

END LOOP;

END;

/



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

* + **Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

**Code:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

dept\_name IN VARCHAR2,

bonus\_percent IN NUMBER

) IS

BEGIN

FOR emp IN (SELECT EmpID, Salary FROM Empp WHERE Department = dept\_name) LOOP

UPDATE Empp

SET Salary = Salary + (Salary \* bonus\_percent / 100)

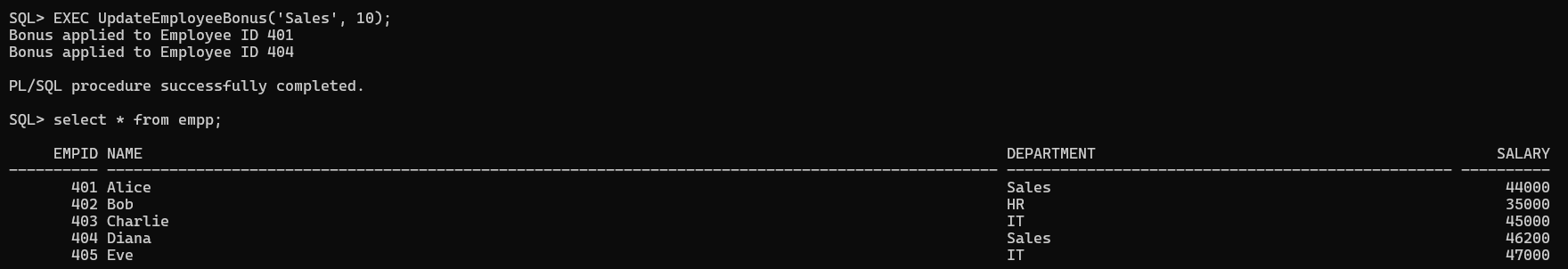
WHERE EmpID = emp.EmpID;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to Employee ID ' || emp.EmpID);

END LOOP;

END;

/



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

* + **Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**Code:**

CREATE OR REPLACE PROCEDURE TransferFunds(

from\_acct IN INT,

to\_acct IN INT,

amount IN NUMBER

) IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Account WHERE AccountID = from\_acct;

IF v\_balance < amount THEN

DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance.');

ELSE

UPDATE Account SET Balance = Balance - amount WHERE AccountID = from\_acct;

UPDATE Account SET Balance = Balance + amount WHERE AccountID = to\_acct;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || amount || ' from ' || from\_acct || ' to ' || to\_acct);

END IF;

END;

/

