```
SQL> DECLARE
  2      CURSOR cur_transactions IS
  3          SELECT c.CustomerID, c.Name, t.TransactionID, t.TransactionDate, t.Amount, t.Transactio
nType
  4          FROM Customers c
  5          JOIN Accounts a ON c.CustomerID = a.CustomerID
  6          JOIN Transactions t ON a.AccountID = t.AccountID
  7          WHERE EXTRACT(MONTH FROM t.TransactionDate) = EXTRACT(MONTH FROM SYSDATE)
  8            AND EXTRACT(YEAR FROM t.TransactionDate) = EXTRACT(YEAR FROM SYSDATE)
  9          ORDER BY c.CustomerID, t.TransactionDate;
 10
 11      v_customer_id Customers.CustomerID%TYPE;
 12      v_name Customers.Name%TYPE;
 13      v_transaction_id Transactions.TransactionID%TYPE;
 14      v_transaction_date Transactions.TransactionDate%TYPE;
 15      v_amount Transactions.Amount%TYPE;
 16      v_transaction_type Transactions.TransactionType%TYPE;
 17  BEGIN
 18      OPEN cur_transactions;
 19
 20      LOOP
 21          FETCH cur_transactions INTO v_customer_id, v_name, v_transaction_id, v_transaction_date
, v_amount, v_transaction_type;
 22          EXIT WHEN cur_transactions%NOTFOUND;
 23
 24          DBMS_OUTPUT.PUT_LINE('Customer ID: ' || v_customer_id || ' - ' || v_name);
 25          DBMS_OUTPUT.PUT_LINE('Transaction ID: ' || v_transaction_id);
 26          DBMS_OUTPUT.PUT_LINE('Date: ' || v_transaction_date || ' Amount: ' || v_amount || ' Typ
e: ' || v_transaction_type);
 27          DBMS_OUTPUT.PUT_LINE('-----------------------------------------');
 28      END LOOP;
 29
 30      CLOSE cur_transactions;
 31  END;
 32  /
Customer ID: 1 - John Doe
Transaction ID: 1
Date: 04-AUG-24 Amount: 200 Type: Deposit
-----------------------------------------
Customer ID: 2 - Jane Smith
Transaction ID: 2
Date: 04-AUG-24 Amount: 300 Type: Withdrawal
-----------------------------------------

PL/SQL procedure successfully completed.
```

```
SQL> DECLARE
  2      CURSOR cur_accounts IS
  3          SELECT AccountID, Balance
  4          FROM Accounts;
  5
  6      v_account_id Accounts.AccountID%TYPE;
  7      v_balance Accounts.Balance%TYPE;
  8      v_annual_fee CONSTANT NUMBER := 50; -- Define the annual fee amount
  9  BEGIN
 10      OPEN cur_accounts;
 11
 12      LOOP
 13          FETCH cur_accounts INTO v_account_id, v_balance;
 14          EXIT WHEN cur_accounts%NOTFOUND;
 15
 16          -- Deduct the annual fee
 17          UPDATE Accounts
 18          SET Balance = Balance - v_annual_fee
 19          WHERE AccountID = v_account_id;
 20
 21          DBMS_OUTPUT.PUT_LINE('Annual fee applied to Account ID: ' || v_account_id);
 22      END LOOP;
 23
 24      CLOSE cur_accounts;
 25      COMMIT;
 26  END;
 27  /
Annual fee applied to Account ID: 1
Annual fee applied to Account ID: 2

PL/SQL procedure successfully completed.
```

```
SQL> DECLARE
  2      CURSOR cur_loans IS
  3          SELECT LoanID, InterestRate
  4          FROM Loans;
  5
  6      v_loan_id Loans.LoanID%TYPE;
  7      v_interest_rate Loans.InterestRate%TYPE;
  8      v_new_interest_rate NUMBER;
  9  BEGIN
 10      OPEN cur_loans;
 11
 12      LOOP
 13          FETCH cur_loans INTO v_loan_id, v_interest_rate;
 14          EXIT WHEN cur_loans%NOTFOUND;
 15
 16          -- Apply new policy (e.g., increase by 0.5% for demonstration)
 17          v_new_interest_rate := v_interest_rate + 0.5;
 18
 19          -- Update the interest rate
 20          UPDATE Loans
 21          SET InterestRate = v_new_interest_rate
 22          WHERE LoanID = v_loan_id;
 23
 24          DBMS_OUTPUT.PUT_LINE('Updated Interest Rate for Loan ID: ' || v_loan_id || ' to ' || v_
new_interest_rate || '%');
 25      END LOOP;
 26
 27      CLOSE cur_loans;
 28      COMMIT;
 29  END;
 30  /
Updated Interest Rate for Loan ID: 1 to 5.5%

PL/SQL procedure successfully completed.
```