**PL/SQL PROGRAMMING**

**Exercise 1: Control Structures**

**Scenario 1-:**

|  |
| --- |
| **SQL> create table customers(**  **2 customer\_id number primary key,**  **3 name varchar2(100),**  **4 dob date,**  **5 balance number,**  **6 isVip char(1),**  **7 lastModified date**  **8 );**  **Table created.**  **SQL> INSERT INTO customers(customer\_id, name, dob, balance, lastModified, isVip) VALUES (1, 'Tripti', TO\_DATE('1955-06-15','YYYY-MM-DD'), 15000, SYSDATE, NULL);**  **1 row created.**  **SQL> INSERT INTO customers(customer\_id, name, dob, balance, lastModified, isVip) VALUES (2, 'Amrit', TO\_DATE('1955-06-15','YYYY-MM-DD'), 7000, SYSDATE, NULL);**  **1 row created.**  **SQL> INSERT INTO customers(customer\_id, name, dob, balance, lastModified, isVip) VALUES (3, 'Suhani', TO\_DATE('1955-06-15','YYYY-MM-DD'), 7000, SYSDATE, NULL);**  **1 row created.**  **SQL> INSERT INTO customers(customer\_id, name, dob, balance, lastModified, isVip) VALUES (4, 'Rachna', TO\_DATE('1955-06-15','YYYY-MM-DD'), 7000, SYSDATE, NULL);**  **1 row created.**  **SQL> INSERT INTO customers(customer\_id, name, dob, balance, lastModified, isVip) VALUES (5, 'Harsh', TO\_DATE('1955-06-15','YYYY-MM-DD'), 7000, SYSDATE, NULL);**  **1 row created.**  **SQL> CREATE TABLE Loans (**  **2 loan\_id NUMBER PRIMARY KEY,**  **3 customer\_id NUMBER,**  **4 loan\_amount NUMBER,**  **5 interest\_rate NUMBER,**  **6 StartDate DATE,**  **7 EndDate DATE,**  **8 FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)**  **9 );**  **Table created.**  **SQL> INSERT INTO Loans (loan\_id, customer\_id, loan\_amount, interest\_rate, StartDate, EndDate)**  **2 VALUES (201, 1, 50000, 8.5, TO\_DATE('2023-01-01', 'YYYY-MM-DD'), TO\_DATE('2025-07-15', 'YYYY-MM-DD'));**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Loans (loan\_id, customer\_id, loan\_amount, interest\_rate, StartDate, EndDate)**  **2 VALUES (202, 2, 30000, 7.5, TO\_DATE('2023-06-01', 'YYYY-MM-DD'), SYSDATE + 20);**  **1 row created.**  **SQL> INSERT INTO Loans (loan\_id, customer\_id, loan\_amount, interest\_rate, StartDate, EndDate)**  **2 VALUES (204, 1, 45000, 8.2, TO\_DATE('2023-08-15', 'YYYY-MM-DD'), SYSDATE + 10);**  **1 row created.**  **SQL> INSERT INTO Loans (loan\_id, customer\_id, loan\_amount, interest\_rate, StartDate, EndDate)**  **2 VALUES (205, 3, 55000, 9.0, TO\_DATE('2024-01-01', 'YYYY-MM-DD'), SYSDATE + 29);**  **1 row created.**  **SQL> INSERT INTO Loans (loan\_id, customer\_id, loan\_amount, interest\_rate, StartDate, EndDate)**  **2 VALUES (203, 4, 25000, 7.0, TO\_DATE('2023-11-01', 'YYYY-MM-DD'), SYSDATE + 60);**  **1 row created.**  **SQL> declare**  **2 cursor customer\_cursor is**  **3 select customer\_id**  **4 from customers**  **5 where MONTHS\_BETWEEN(SYSDATE, dob)/12 > 60;**  **6**  **7 begin**  **8 for cust in customer\_cursor loop**  **9 update Loans**  **10 set interest\_rate= interest\_rate- 1**  **11 where customer\_id= cust.customer\_id;**  **12 end loop;**  **13**  **14 commit;**  **15 dbms\_output.put\_line('Interest rates updated for customers above 60.');**  **16 end;**  **17 /** |

**OUTPUT-:**

|  |
| --- |
| **Interest rates updated for customers above 60.**  **PL/SQL procedure successfully completed.** |

**Scenario 2 -:**

|  |
| --- |
| **SQL> set serveroutput on;**  **SQL> BEGIN**  **2 UPDATE customers**  **3 SET isVip= 'Y'**  **4 WHERE balance > 10000;**  **5**  **6 COMMIT;**  **7**  **8 dbms\_output.put\_line('VIP status updated for eligible customers.');**  **9 END;** |

**OUPUT-:**

|  |
| --- |
| **VIP status updated for eligible customers.**  **PL/SQL procedure successfully completed.** |

**Scenario 3-:**

|  |
| --- |
| **SQL> SET SERVEROUTPUT ON;**  **SQL>**  **SQL> DECLARE**  **2 CURSOR loan\_cursor IS**  **3 SELECT l.loan\_id, l.EndDate, c.name, c.customer\_id**  **4 FROM Loans l**  **5 JOIN Customers c ON c.customer\_id = l.customer\_id**  **6 WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30;**  **7 BEGIN**  **8 FOR i IN loan\_cursor LOOP**  **9 DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || i.loan\_id ||**  **10 ' for customer ' || i.name ||**  **11 ' (CustomerID: ' || i.customer\_id ||**  **12 ') is due on ' || TO\_CHAR(i.EndDate, 'YYYY-MM-DD'));**  **13 END LOOP;**  **14 END;**  **15 /** |

**OUTPUT-:**

|  |
| --- |
| **Reminder: Loan ID 204 for customer Tripti (CustomerID: 1) is due on 2025-07-07**  **Reminder: Loan ID 201 for customer Tripti (CustomerID: 1) is due on 2025-07-15**  **Reminder: Loan ID 202 for customer Amrit (CustomerID: 2) is due on 2025-07-17**  **Reminder: Loan ID 205 for customer Suhani (CustomerID: 3) is due on 2025-07-26**  **PL/SQL procedure successfully completed.** |

**Exercise 3: Stored Procedures**

**Scenario 1-:**

|  |
| --- |
| **SQL> CREATE TABLE Accounts (**  **2 account\_id NUMBER PRIMARY KEY,**  **3 customer\_id NUMBER,**  **4 accountType VARCHAR2(20),**  **5 balance NUMBER,**  **6 LastModified DATE,**  **7 FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)**  **8 );**  **Table created.**  **SQL> INSERT INTO Accounts (account\_id, customer\_id, accountType, balance, LastModified) VALUES (101, 1, 'Savings', 15000, SYSDATE);**  **1 row created.**  **SQL> INSERT INTO Accounts (account\_id, customer\_id, accountType, balance, LastModified) VALUES (102, 2, 'Savings', 19000, SYSDATE);**  **1 row created.**  **SQL> INSERT INTO Accounts (account\_id, customer\_id, accountType, balance, LastModified) VALUES (103, 3, 'Savings', 5000, SYSDATE);**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Accounts (account\_id, customer\_id, accountType, balance, LastModified) VALUES (104, 4, 'Savings', 7000, SYSDATE);**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Accounts (account\_id, customer\_id, accountType, balance, LastModified) VALUES (105, 5, 'Savings', 7000, SYSDATE);**  **1 row created.**  **SQL> CREATE TABLE Transactions (**  **2 trans\_id NUMBER PRIMARY KEY,**  **3 account\_id NUMBER,**  **4 TransactionDate DATE,**  **5 amount NUMBER,**  **6 TransactionType VARCHAR2(10),**  **7 FOREIGN KEY (account\_id) REFERENCES Accounts(account\_id)**  **8 );**  **Table created.**  **SQL> INSERT INTO Transactions (trans\_id, account\_id, TransactionDate, amount, TransactionType)**  **2 VALUES (1, 101, SYSDATE - 5, 3000, 'Credit');**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Transactions (trans\_id, account\_id, TransactionDate, amount, TransactionType)**  **2 VALUES (2, 101, SYSDATE - 2, 1000, 'Debit');**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Transactions (trans\_id, account\_id, TransactionDate, amount, TransactionType)**  **2 VALUES (3, 102, SYSDATE - 3, 1500, 'Credit');**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Transactions (trans\_id, account\_id, TransactionDate, amount, TransactionType)**  **2 VALUES (4, 103, SYSDATE - 1, 2000, 'Debit');**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Transactions (trans\_id, account\_id, TransactionDate, amount, TransactionType)**  **2 VALUES (5, 103, SYSDATE, 500, 'Credit');**  **1 row created.**  **SQL> CREATE TABLE Employees (**  **2 emp\_id NUMBER PRIMARY KEY,**  **3 name VARCHAR2(100),**  **4 position VARCHAR2(50),**  **5 salary NUMBER,**  **6 dept VARCHAR2(50),**  **7 hireDate DATE**  **8 );**  **Table created.**  **SQL> INSERT INTO Employees (emp\_id, name, position, salary, dept, hireDate)**  **2 VALUES (1, 'Ravi Kumar', 'Manager', 90000, 'Finance', TO\_DATE('2020-01-15', 'YYYY-MM-DD'));**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Employees (emp\_id, name, position, salary, dept, hireDate)**  **2 VALUES (2, 'Anita Sharma', 'Analyst', 65000, 'Finance', TO\_DATE('2021-06-10', 'YYYY-MM-DD'));**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Employees (emp\_id, name, position, salary, dept, hireDate)**  **2 VALUES (3, 'Rahul Mehta', 'Developer', 72000, 'IT', TO\_DATE('2019-11-20', 'YYYY-MM-DD'));**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Employees (emp\_id, name, position, salary, dept, hireDate)**  **2 VALUES (4, 'Priya Das', 'Support Executive', 48000, 'Customer Service', TO\_DATE('2022-03-05', 'YYYY-MM-DD'));**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Employees (emp\_id, name, position, salary, dept, hireDate)**  **2 VALUES (5, 'Sonal Verma', 'Tester', 58000, 'IT', TO\_DATE('2020-08-30', 'YYYY-MM-DD'));**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Employees (emp\_id, name, position, salary, dept, hireDate)**  **2 VALUES (6, 'Amit Joshi', 'Sales Executive', 50000, 'Sales', TO\_DATE('2021-02-14', 'YYYY-MM-DD'));**  **1 row created.**  **SQL>**  **SQL> INSERT INTO Employees (emp\_id, name, position, salary, dept, hireDate)**  **2 VALUES (7, 'Neha Roy', 'HR Manager', 75000, 'Human Resources', TO\_DATE('2018-12-01', 'YYYY-MM-DD'));**  **1 row created.**  **SQL> set serveroutput on ;**  **SQL> CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS**  **2 begin**  **3 update Accounts**  **4 set balance = balance + (balance \* 0.01)**  **5 where accountType= 'Savings';**  **6**  **7 commit;**  **8**  **9 dbms\_output.put\_line('Monthly interest of 1% applied to all savings accounts.');**  **10 dbms\_output.put\_line('Updated Balances:');**  **11**  **12**  **13 for rec in (**  **14 select account\_id , balance**  **15 from Accounts**  **16 where accountType= 'Savings'**  **17 ) loop**  **18 dbms\_output.put\_line('Account ID: ' || rec.account\_id|| ' | New Balance: ' || TO\_CHAR(rec.balance, 'FM9999990.00'));**  **19 end loop;**  **20 end;**  **21 /**  **SQL> BEGIN**  **2 ProcessMonthlyInterest;**  **3 END;**  **4 /** |

**OUTPUT -:**

|  |
| --- |
| **Procedure created.**  **Monthly interest of 1% applied to all savings accounts.**  **Updated Balances:**  **Account ID: 101 | New Balance: 15150.00**  **Account ID: 102 | New Balance: 19190.00**  **Account ID: 103 | New Balance: 5050.00**  **Account ID: 104 | New Balance: 7070.00**  **Account ID: 105 | New Balance: 7070.00**  **PL/SQL procedure successfully completed.** |

**Scenario 2-:**

|  |
| --- |
| **SQL> CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(**  **2 p\_department IN VARCHAR2,**  **3 p\_bonus\_percent IN NUMBER**  **4 ) IS**  **5 BEGIN**  **6 UPDATE Employees**  **7 SET Salary = salary+ (salary \* (p\_bonus\_percent / 100))**  **8 WHERE dept = p\_department;**  **9**  **10 COMMIT;**  **11**  **12 dbms\_output.put\_line('Bonus of ' || p\_bonus\_percent || '% applied to department: ' || p\_department);**  **13 END;**  **14 /**  **SQL> BEGIN**  **2 UpdateEmployeeBonus('IT', 10);**  **3 END;**  **4 /** |

**OUTPUT -:**

|  |
| --- |
| **Procedure created.**  **Bonus of 10% applied to department: IT**  **PL/SQL procedure successfully completed.** |

**Scenario 3-:**

|  |
| --- |
| **SQL> CREATE OR REPLACE PROCEDURE TransferFunds(**  **2 p\_from\_account IN NUMBER,**  **3 p\_to\_account IN NUMBER,**  **4 p\_amount IN NUMBER**  **5 ) IS**  **6 v\_balance NUMBER;**  **7 v\_new\_trans\_id NUMBER;**  **8 BEGIN**  **9**  **10 SELECT balance INTO v\_balance**  **11 FROM Accounts**  **12 WHERE account\_id = p\_from\_account;**  **13**  **14**  **15 IF v\_balance < p\_amount THEN**  **16 DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance in source account.');**  **17 RETURN;**  **18 END IF;**  **19**  **20**  **21 UPDATE Accounts**  **22 SET balance = balance - p\_amount**  **23 WHERE account\_id = p\_from\_account;**  **24**  **25**  **26 UPDATE Accounts**  **27 SET balance = balance + p\_amount**  **28 WHERE account\_id = p\_to\_account;**  **29**  **30**  **31 SELECT NVL(MAX(trans\_id), 0) + 1 INTO v\_new\_trans\_id FROM Transactions;**  **32**  **33**  **34 INSERT INTO Transactions (trans\_id, account\_id, TransactionDate, amount, TransactionType)**  **35 VALUES (v\_new\_trans\_id, p\_from\_account, SYSDATE, p\_amount, 'Debit');**  **36**  **37**  **38 INSERT INTO Transactions (trans\_id, account\_id, TransactionDate, amount, TransactionType)**  **39 VALUES (v\_new\_trans\_id + 1, p\_to\_account, SYSDATE, p\_amount, 'Credit');**  **40**  **41**  **42 COMMIT;**  **43**  **44 DBMS\_OUTPUT.PUT\_LINE('Transfer of ' || p\_amount || ' from account ' || p\_from\_account || ' to account ' || p\_to\_account || ' completed.');**  **45 EXCEPTION**  **46 WHEN NO\_DATA\_FOUND THEN**  **47 DBMS\_OUTPUT.PUT\_LINE('Transfer failed: One of the accounts does not exist.');**  **48 WHEN OTHERS THEN**  **49 DBMS\_OUTPUT.PUT\_LINE('Transfer failed: ' || SQLERRM);**  **50 ROLLBACK;**  **51 END;**  **52 /**  **SQL> SET SERVEROUTPUT ON;**  **SQL>**  **SQL> BEGIN**  **2 TransferFunds(101, 102, 1000);**  **3 END;**  **4 /** |

**OUTPUT -:**

|  |
| --- |
| **Procedure created.**  **Transfer of 1000 from account 101 to account 102 completed.**  **PL/SQL procedure successfully completed.** |