**Exercise1 : Control Structures**

**Scenario1:** Loan Interest Reduction for Customers Over 60

**Objective**: The bank wants to apply a 1% discount on loan interest rates for customers who are elder than 60 years.

**Assumptions for Table Creation**:

* Table: customers
* Columns: customer\_id, customer\_name, age
* Table: loans
* Columns: loan\_id, customer\_id, interest\_rate

Relationship between the tables: customers.customer\_id = loans.customer\_id

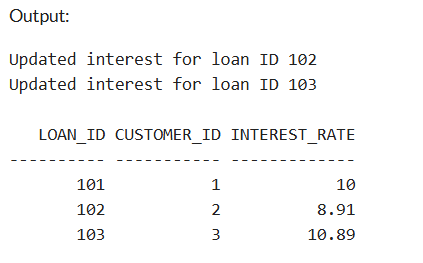
**Construction of PL/SQL block**:

* Firstly, loop through all the customers and their loans.
* If a customer’s age > 60, reduce their loan’s interest rate by 1% of its current value.
* Discount is applied using the formula:

interest\_rate = interest\_rate \* 0.9

interest\_rate.sql

|  |
| --- |
| CREATE TABLE customers (  customer\_id NUMBER PRIMARY KEY,  customer\_name VARCHAR2(100),  age NUMBER  );  CREATE TABLE loans (  loan\_id NUMBER PRIMARY KEY,  customer\_id NUMBER,  interest\_rate NUMBER,  FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)  );  select \* from customers;  select \* from loans;  INSERT INTO customers VALUES (1, 'John', 45);  INSERT INTO customers VALUES (2, 'Mary', 65);  INSERT INTO customers VALUES (3, 'David', 70);  INSERT INTO loans VALUES (101, 1, 10.0);  INSERT INTO loans VALUES (102, 2, 9.0);  INSERT INTO loans VALUES (103, 3, 11.0);  SET SERVEROUTPUT ON;  BEGIN  FOR rec IN (  SELECT cust.customer\_id, cust.age, l.loan\_id, l.interest\_rate  FROM customers cust  JOIN loans l ON cust.customer\_id = l.customer\_id  )  LOOP  IF rec.age > 60 THEN  UPDATE loans  SET interest\_rate = interest\_rate \* 0.99  WHERE loan\_id = rec.loan\_id;  DBMS\_OUTPUT.PUT\_LINE ('Updated interest for loan ID ' || rec.loan\_id);  END IF;  END LOOP;  END;  /  SELECT \* FROM loans; |

****

**Scenario2:** Customers should be promoted to VIP based on balance

**Objective:** Customers are promoted to VIP status by setting the is\_vip flag to TRUE for all the customers who have a balance greater than $10,000.

**Assumptions for Table Creation**:

* Table customers:
* Columns: customer\_id, name, balance, is\_vip

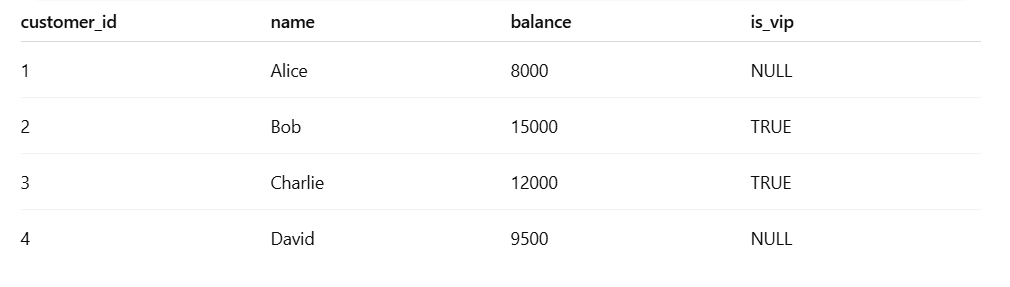
**Construction of PL/SQL Block:**

* Firstly, the loop goes through the customer id and balance in the customers table.
* Then, it checks the condition whether balance exists above 10000 or not.

vip\_status.sql:

|  |
| --- |
| CREATE TABLE customers (  customer\_id NUMBER PRIMARY KEY,  name VARCHAR2(100),  balance NUMBER, is\_vip VARCHAR2(5)  );  INSERT INTO customers VALUES (1, 'Alice', 8000, NULL);  INSERT INTO customers VALUES (2, 'Bob', 15000, NULL);  INSERT INTO customers VALUES (3, 'Charlie', 12000, NULL);  INSERT INTO customers VALUES (4, 'David', 9500, NULL);  BEGIN  FOR cust IN (SELECT customer\_id, balance FROM customers) LOOP  IF cust.balance > 10000 THEN  UPDATE customers  SET is\_vip = 'TRUE'  WHERE customer\_id = cust.customer\_id;  END IF;  END LOOP;  END;  /  SELECT \* FROM customers; |

Expected output:



**Scenario3**: Send Remainders within 30 days for loans due to the customers

**Objective:** Must retrieve reminder message for all the customers whose loans are due within the next 30 days.

**Assumptions:**

* Table loans:
* Columns: loan\_id, customer\_id, due\_date
* Table customers:
* Columns: customer\_id, name

**Construction of PL/SQL Block:**

* Firstly, declare the cursor as dueloan\_cursor.
* Then, begin the loop through the cursor and then print the reminder messages with customer names and loan id along with due date.

due\_loan.sql

|  |
| --- |
| CREATE TABLE customers (  customer\_id NUMBER PRIMARY KEY,  name VARCHAR2(100)  );  CREATE TABLE loans (  loan\_id NUMBER PRIMARY KEY,  customer\_id NUMBER,  due\_date DATE,  FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)  );  BEGIN  INSERT INTO customers VALUES (1, 'Alice');  INSERT INTO customers VALUES (2, 'Bob');  INSERT INTO customers VALUES (3, 'Charlie');  INSERT INTO loans VALUES (101, 1, TO\_DATE ('03-JUL-2025', 'DD-MON-YYYY'));  INSERT INTO loans VALUES (102, 2, TO\_DATE ('13-JUL-2025', 'DD-MON-YYYY'));  INSERT INTO loans VALUES (103, 3, TO\_DATE ('12-AUG-2025', 'DD-MON-YYYY'));  END;  /  SET SERVEROUTPUT ON;  BEGIN  FOR loan\_rec IN (  SELECT l.loan\_id, l.customer\_id, l.due\_date, c.name  FROM loans l  JOIN customers c ON l.customer\_id = c.customer\_id  WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30  )  LOOP  DBMS\_OUTPUT.PUT\_LINE (  'Reminder: Dear ' || loan\_rec.name ||  ', your loan (ID: ' || loan\_rec.loan\_id ||  ') is due on ' || TO\_CHAR (loan\_rec.due\_date, 'DD-MON-YYYY') || '.'  );  END LOOP;  END;  /  SELECT \* FROM loans; |

