**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.

**QUERY:**

DECLARE

v\_discount\_rate NUMBER := 0.01; -- 1% discount rate

BEGIN

FOR cust IN (SELECT \* FROM Customers) LOOP

IF cust.age > 60 THEN

UPDATE Loans

SET interest\_rate = interest\_rate - (interest\_rate \* v\_discount\_rate)

WHERE customer\_id = cust.customer\_id;

END IF;

END LOOP;

COMMIT;

END;

**Explanation:**

We declare a variable `v\_discount\_rate` to store the discount rate (1% in this case). We use a `FOR` loop to iterate through all customers in the Customers table. For each customer above 60 years old, we update the interest rate of their loans by reducing it by the discount rate. Finally, we commit the changes to the database using the `COMMIT` statement.

**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.

**QUERY:**

DECLARE

BEGIN

FOR cust IN (SELECT \* FROM Customers) LOOP

IF cust.balance > 10000 THEN

UPDATE Customers

SET IsVIP = TRUE

WHERE customer\_id = cust.customer\_id;

END IF;

END LOOP;

COMMIT;

END;

**Explanation:**

We use a `FOR` loop to iterate through all customers in the Customers table. For each customer with a balance over $10,000, we update the IsVIP flag to TRUE. Finally, we commit the changes to the database using the `COMMIT` statement.

**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.

**QUERY:**

DECLARE

v\_due\_date DATE;

BEGIN

FOR loan\_rec IN (SELECT \* FROM Loans WHERE due\_date BETWEEN SYSDATE AND SYSDATE + 30) LOOP

SELECT due\_date INTO v\_due\_date FROM Loans WHERE loan\_id = loan\_rec.loan\_id;

DBMS\_OUTPUT.PUT\_LINE('Reminder: Your loan with ID ' || loan\_rec.loan\_id || ' is due on ' || TO\_CHAR(v\_due\_date, 'DD-MON-YYYY'));

END LOOP;

END;

**Explanation:**

We declare a variable `v\_due\_date` to store the due date of each loan. We use a `FOR` loop to fetch all loans due within the next 30 days from the Loans table. For each loan due within the specified timeframe, we retrieve the due date and print a reminder message using `DBMS\_OUTPUT.PUT\_LINE`.