**Create Table :**

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(50),

age NUMBER,

interest\_rate NUMBER(5,2),

balance NUMBER(10,2),

IsVIP VARCHAR2(5)

);

**Insert Values :**

INSERT INTO customers VALUES (1, 'Alice', 65, 5.50, 12000, 'FALSE');

INSERT INTO customers VALUES (2, 'Bob', 45, 6.00, 8000, 'FALSE');

INSERT INTO customers VALUES (3, 'Carol', 70, 4.75, 11000, 'FALSE');

**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.

**Ans :**

BEGIN

FOR cust IN (

SELECT customer\_id, interest\_rate

FROM customers

WHERE age > 60

) LOOP

UPDATE customers

SET interest\_rate = interest\_rate - 1

WHERE customer\_id = cust.customer\_id;

END LOOP;

COMMIT;

END;

**Output :**

| customer\_id | name | age | interest\_rate | balance | IsVIP |
| --- | --- | --- | --- | --- | --- |
| 1 | Alice | 65 | 4.50 | 12000 | FALSE |
| 2 | Bob | 45 | 6.00 | 8000 | FALSE |
| 3 | Carol | 70 | 3.75 | 11000 | FALSE |

**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.

**Ans :**

BEGIN

FOR cust IN (

SELECT customer\_id

FROM customers

WHERE balance > 10000

) LOOP

UPDATE customers

SET IsVIP = 'TRUE'

WHERE customer\_id = cust.customer\_id;

END LOOP;

COMMIT;

END;

**Output :**

| customer\_id | name | age | interest\_rate | balance | IsVIP |
| --- | --- | --- | --- | --- | --- |
| 1 | Alice | 65 | 4.50 | 12000 | TRUE |
| 2 | Bob | 45 | 6.00 | 8000 | FALSE |
| 3 | Carol | 70 | 3.75 | 11000 | TRUE |

**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.

**Ans :**

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

due\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

INSERT INTO loans VALUES (101, 1, SYSDATE + 15); -- due soon

INSERT INTO loans VALUES (102, 2, SYSDATE + 40); -- not due soon

INSERT INTO loans VALUES (103, 3, SYSDATE + 5); -- due soon

BEGIN

FOR loan\_rec IN (

SELECT l.loan\_id, l.customer\_id, c.name, l.due\_date

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE l.due\_date <= SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ' || loan\_rec.loan\_id ||

' for customer ' || loan\_rec.name ||

' is due on ' || TO\_CHAR(loan\_rec.due\_date, 'DD-MON-YYYY'));

END LOOP;

END;

**Output :**

Reminder: Loan 101 for customer Alice is due on 14-JUL-2025

Reminder: Loan 103 for customer Carol is due on 04-JUL-2025