**Control Structures**

**Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.

CREATE TABLE customers (

customer\_id INTEGER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

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

);

BEGIN

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

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

INSERT INTO customers VALUES (3, 'Charlie', 70, 15000);

INSERT INTO customers VALUES (4, 'Diana', 99, 500);

INSERT INTO loans VALUES (101, 1, 5.5);

INSERT INTO loans VALUES (102, 2, 6.0);

INsERT INTO loans VALUES (103, 3, 4.8);

INSERT INTO loans VALUES (104, 4, 7.0);

COMMIT;

END;

/

BEGIN

FOR rec IN (

SELECT l.loan\_id, l.interest\_rate

FROM loans l

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

WHERE c.age > 60

) LOOP

UPDATE loans

SET interest\_rate = interest\_rate - 0.01

WHERE loan\_id = rec.loan\_id;

DBMS\_OUTPUT.PUT\_LINE('Discount applied to Loan ID ' || rec.loan\_id);

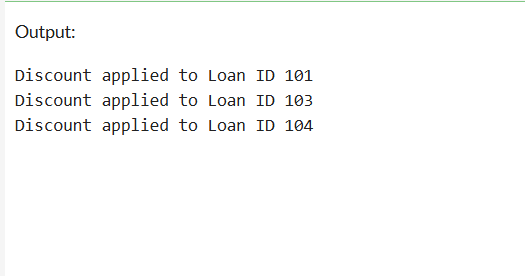
END LOOP;

COMMIT;

END;

/

OUTPUT:



**Scenario 2:** A customer can be promoted to VIP status based on their balance.

CREATE TABLE customers (

customer\_id INTEGER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER,

is\_vip CHAR(1) DEFAULT 'N'

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

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

);

BEGIN

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

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

INSERT INTO customers VALUES (3, 'Charlie', 70, 15000,'N');

INSERT INTO customers VALUES (4, 'Diana', 99, 500,'N');

INSERT INTO loans VALUES (101, 1, 5.5);

INSERT INTO loans VALUES (102, 2, 6.0);

INsERT INTO loans VALUES (103, 3, 4.8);

INSERT INTO loans VALUES (104, 4, 7.0);

COMMIT;

END;

/

BEGIN

FOR rec IN (

SELECT customer\_id FROM customers WHERE balance > 10000

) LOOP

UPDATE customers

SET is\_vip = 'Y'

WHERE customer\_id = rec.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || rec.customer\_id || ' set as VIP');

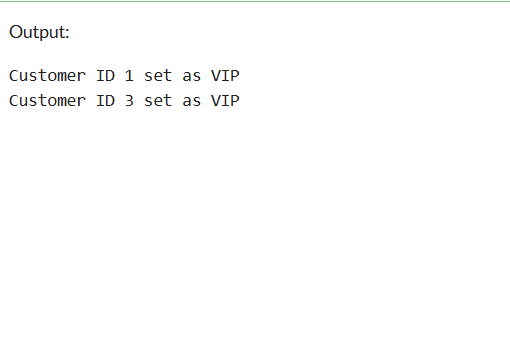
END LOOP;

COMMIT;

END;

/

OUTPUT:



**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days  
CREATE TABLE customers (

customer\_id INTEGER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER,

is\_vip CHAR(1) DEFAULT 'N'

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

due\_date DATE,

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

);

BEGIN

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

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

INSERT INTO customers VALUES (3, 'Charlie', 70, 15000,'N');

INSERT INTO customers VALUES (4, 'Diana', 99, 500,'N');

INSERT INTO loans VALUES (101, 1, 5.5, SYSDATE + 10); -- Due soon

INSERT INTO loans VALUES (102, 2, 6.0, SYSDATE + 40); -- Not due soon

INSERT INTO loans VALUES (103, 3, 4.8, SYSDATE + 5); -- Due soon

INSERT INTO loans VALUES (104, 4, 7.0, SYSDATE + 25); -- Due soon

COMMIT;

END;

/

BEGIN

FOR rec IN (

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

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: Loan ' || rec.loan\_id ||' for customer ' || rec.name ||' is due on ' || TO\_CHAR(rec.due\_date, 'YYYY-MM-DD'));

END LOOP;

END;

/

