**Stored Procedures**

**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

account\_type VARCHAR2(20),

balance NUMBER(12,2)

);

CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

name VARCHAR2(50),

department\_id NUMBER,

salary NUMBER(12,2)

);

INSERT INTO accounts VALUES (1, 'SAVINGS', 5000);

INSERT INTO accounts VALUES (2, 'SAVINGS', 10000);

INSERT INTO accounts VALUES (3, 'CHECKING', 7500);

INSERT INTO accounts VALUES (4, 'SAVINGS', 2500);

INSERT INTO employees VALUES (1, 'Alice', 10, 50000);

INSERT INTO employees VALUES (2, 'Bob', 20, 60000);

INSERT INTO employees VALUES (3, 'Carol', 10, 55000);

COMMIT;

BEGIN

UPDATE accounts

SET balance = ROUND(balance \* 1.01, 2)

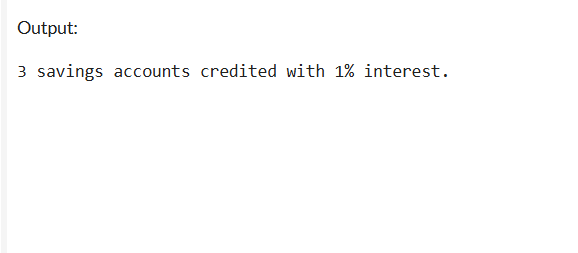
WHERE account\_type = 'SAVINGS';

DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT || ' savings accounts credited with 1% interest.');

COMMIT;

END;

/



**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

name VARCHAR2(50),

department\_id NUMBER,

salary NUMBER(12,2)

);

INSERT INTO employees VALUES (1, 'Alice', 10, 50000);

INSERT INTO employees VALUES (2, 'Bob', 20, 60000);

INSERT INTO employees VALUES (3, 'Carol', 10, 55000);

COMMIT;

DECLARE

p\_dept\_id NUMBER := 10; -- Department to apply bonus

p\_bonus\_pct NUMBER := 10; -- Bonus percentage

BEGIN

UPDATE employees

SET salary = ROUND(salary \* (1 + p\_bonus\_pct / 100), 2)

WHERE department\_id = p\_dept\_id;

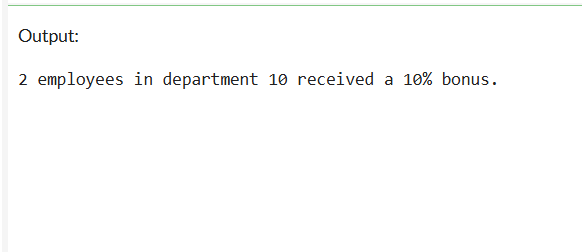
DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT || ' employees in department ' ||

p\_dept\_id || ' received a ' || p\_bonus\_pct || '% bonus.');

COMMIT;

END;

/



**Scenario 3:** Customers should be able to transfer funds between their accounts.

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

account\_type VARCHAR2(20),

balance NUMBER(12,2)

);

INSERT INTO accounts VALUES (1, 'SAVINGS', 5000);

INSERT INTO accounts VALUES (2, 'SAVINGS', 8000);

INSERT INTO accounts VALUES (3, 'CHECKING', 3000);

COMMIT;

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_acct IN NUMBER,

p\_to\_acct IN NUMBER,

p\_amount IN NUMBER

) IS

v\_balance accounts.balance%TYPE;

BEGIN

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = p\_from\_acct

FOR UPDATE;

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001,

'Insufficient funds in account ' || p\_from\_acct);

END IF;

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_acct;

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_acct;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE(

'Transferred ' || p\_amount ||

' from account ' || p\_from\_acct ||

' to account ' || p\_to\_acct

);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20002,

'One or both accounts not found.');

WHEN OTHERS THEN

ROLLBACK;

RAISE;

END TransferFunds;

/

BEGIN

TransferFunds(1, 2, 1500);

END;

/

SELECT \* FROM accounts ORDER BY account\_id;

