WEEK : 2

**Exercise 3: Stored Procedures**

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

* + **Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

CREATE TABLE savings\_accounts (

account\_id NUMBER PRIMARY KEY,

balance NUMBER

);

INSERT INTO savings\_accounts VALUES (1, 1000);

INSERT INTO savings\_accounts VALUES (2, 2000);

INSERT INTO savings\_accounts VALUES (3, 500);

COMMIT;

CREATE OR REPLACE PROCEDURE process\_monthly\_interest IS

BEGIN

UPDATE savings\_accounts

SET balance = balance \* 1.01;

END;

/

BEGIN

process\_monthly\_interest;

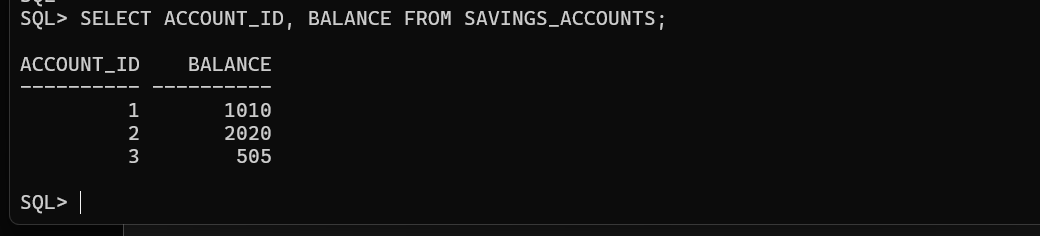
END;

/

SELECT account\_id, balance

FROM savings\_accounts;

Output:



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

**Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

dept\_id NUMBER,

salary NUMBER

);

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

INSERT INTO employees VALUES (2, 10, 60000);

INSERT INTO employees VALUES (3, 20, 55000);

COMMIT;

CREATE OR REPLACE PROCEDURE update\_employee\_bonus (

p\_dept\_id NUMBER,

p\_bonus\_pct NUMBER

) IS

BEGIN

UPDATE employees

SET salary = salary + (salary \* p\_bonus\_pct / 100)

WHERE dept\_id = p\_dept\_id;

END;

/

BEGIN

update\_employee\_bonus(10, 5);

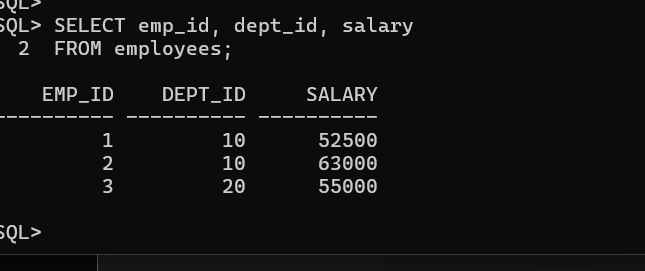
END;

/

SELECT emp\_id, dept\_id, salary

FROM employees;

Output:



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

**Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

balance NUMBER

);

INSERT INTO accounts VALUES (101, 1000);

INSERT INTO accounts VALUES (102, 500);

COMMIT;

CREATE OR REPLACE PROCEDURE transfer\_funds (

p\_from\_acc NUMBER,

p\_to\_acc NUMBER,

p\_amount NUMBER

) IS

v\_balance NUMBER;

BEGIN

SELECT balance

INTO v\_balance

FROM accounts

WHERE account\_id = p\_from\_acc

FOR UPDATE;

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds');

END IF;

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_acc;

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_acc;

END;

/

BEGIN

transfer\_funds(101, 102, 300);

END;

/

SELECT account\_id, balance

FROM accounts;

Output:

