**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.

**Query**

1. **Create the Table**

CREATE TABLE SAVINGS\_ACCOUNTS (

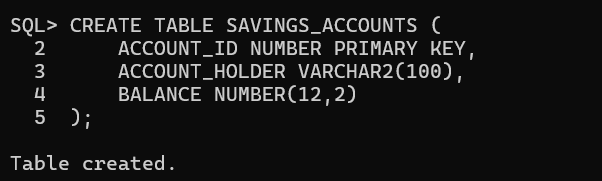
ACCOUNT\_ID NUMBER PRIMARY KEY,

ACCOUNT\_HOLDER VARCHAR2(100),

BALANCE NUMBER(12,2)

);

**Output**

****

**Query**

1. **Insert Sample Data**

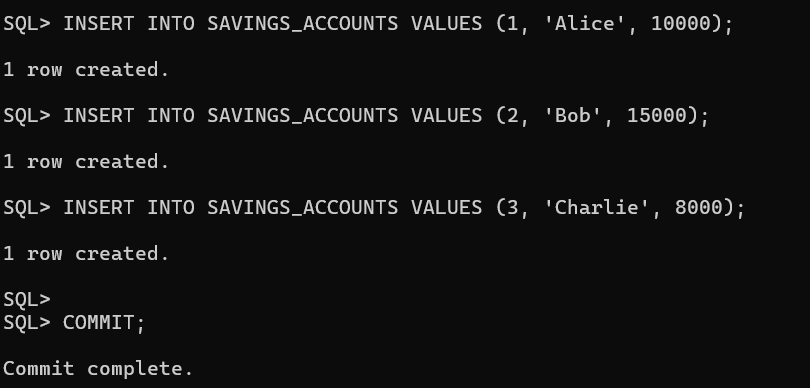
INSERT INTO SAVINGS\_ACCOUNTS VALUES (1, 'Alice', 10000);

INSERT INTO SAVINGS\_ACCOUNTS VALUES (2, 'Bob', 15000);

INSERT INTO SAVINGS\_ACCOUNTS VALUES (3, 'Charlie', 8000);

COMMIT;

**Output**



**Query**

1. **Create the Stored Procedure**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE SAVINGS\_ACCOUNTS

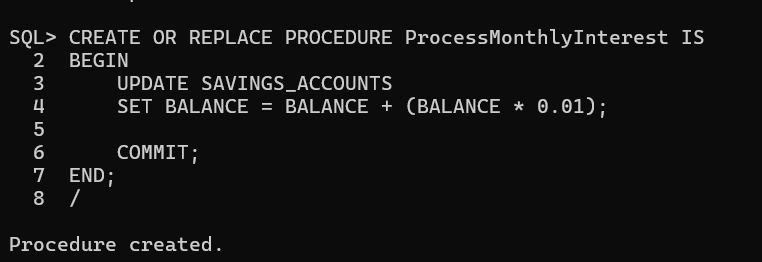
SET BALANCE = BALANCE + (BALANCE \* 0.01);

COMMIT;

END;

/

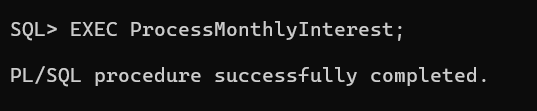
**Output**



**Query**

1. Execute the Procedure

**Output**

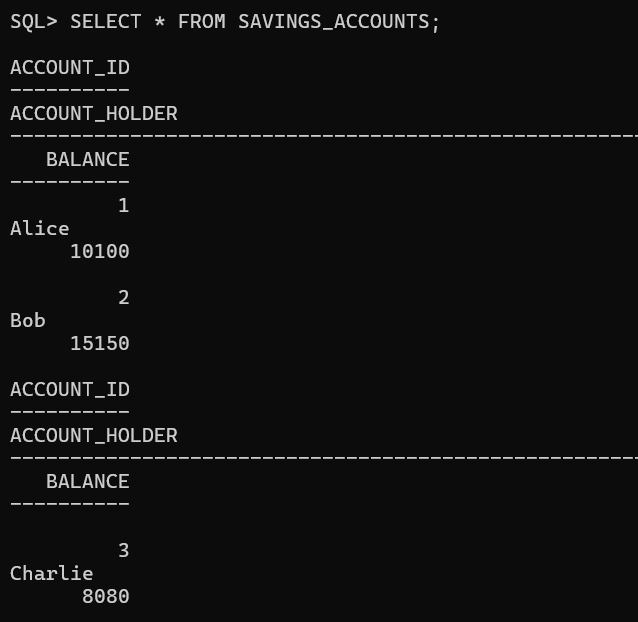


**Query**

1. **View Updated Balances**

SELECT \* 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.

**Query**

1. **Create EMPLOYEES Table**

CREATE TABLE EMPLOYEES (

EMP\_ID NUMBER PRIMARY KEY,

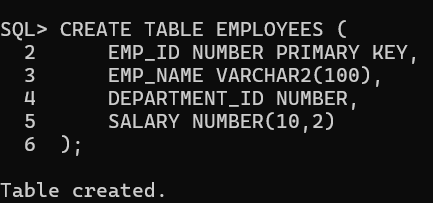
EMP\_NAME VARCHAR2(100),

DEPARTMENT\_ID NUMBER,

SALARY NUMBER(10,2)

);

**Output**



**Query**

1. **Insert Sample Data**

INSERT INTO EMPLOYEES VALUES (1, 'Alice', 101, 50000);

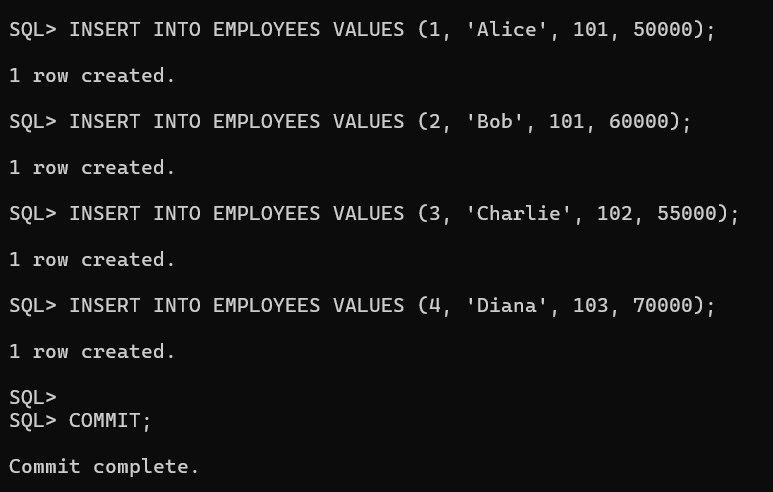
INSERT INTO EMPLOYEES VALUES (2, 'Bob', 101, 60000);

INSERT INTO EMPLOYEES VALUES (3, 'Charlie', 102, 55000);

INSERT INTO EMPLOYEES VALUES (4, 'Diana', 103, 70000);

COMMIT;

**Output**



**Query**

1. **Create the Stored Procedure**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department\_id IN NUMBER,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

UPDATE EMPLOYEES

SET SALARY = SALARY + (SALARY \* p\_bonus\_percent / 100)

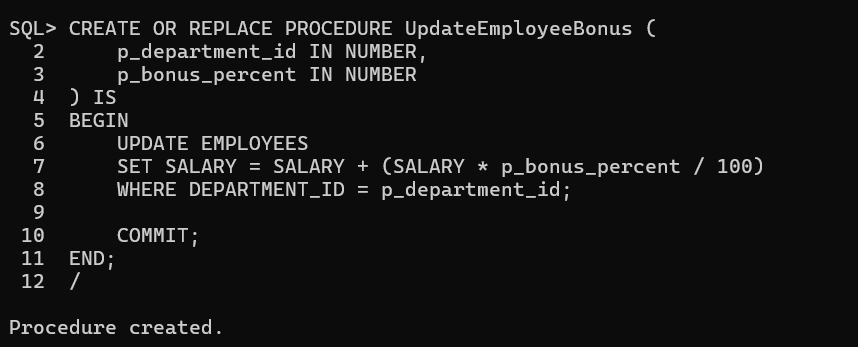
WHERE DEPARTMENT\_ID = p\_department\_id;

COMMIT;

END;

/

**Output**

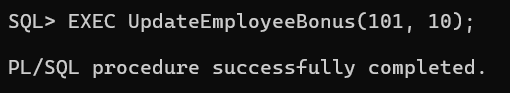


**Query**

1. **Execute the Procedure (Example: 10% bonus to dept 101)**

EXEC UpdateEmployeeBonus(101, 10);

**Output**

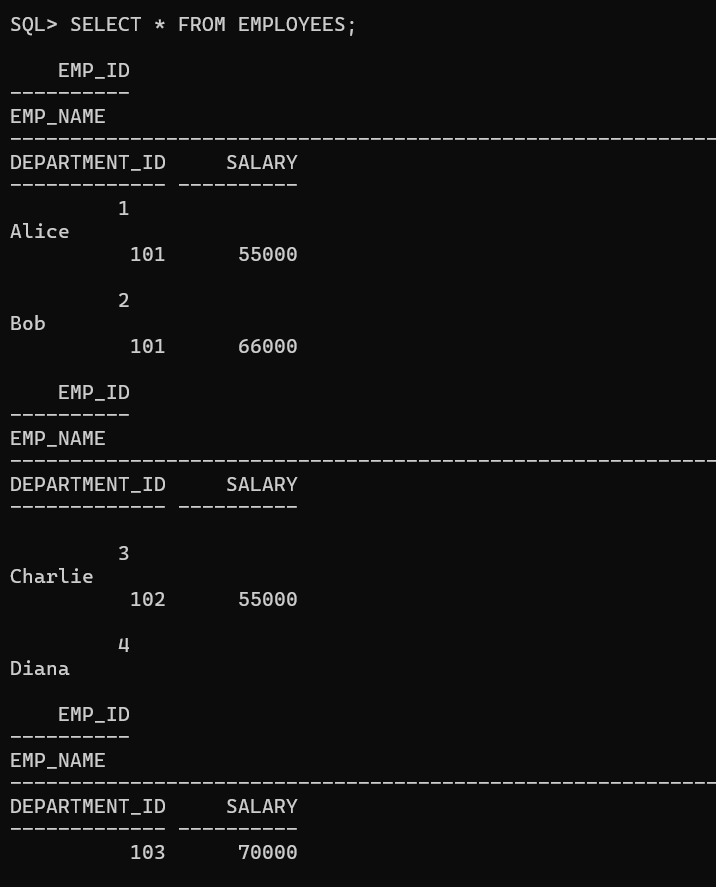


**Query**

1. **View Updated Salaries**

SELECT \* 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.

**Query**

1. **Create ACCOUNTS Table**

CREATE TABLE ACCOUNTS (

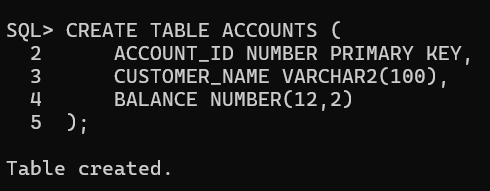
ACCOUNT\_ID NUMBER PRIMARY KEY,

CUSTOMER\_NAME VARCHAR2(100),

BALANCE NUMBER(12,2)

);

**Output**



**Query**

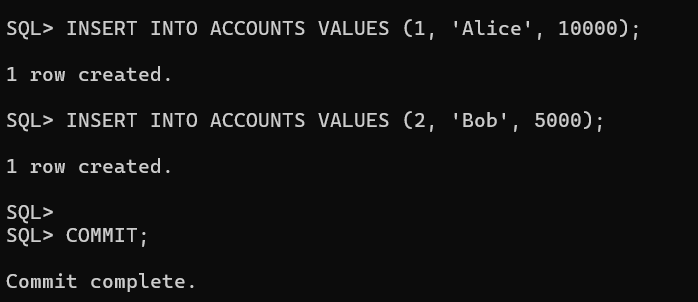
1. **Insert Sample Data**

INSERT INTO ACCOUNTS VALUES (1, 'Alice', 10000);

INSERT INTO ACCOUNTS VALUES (2, 'Bob', 5000);

COMMIT;

**Output**



**Query**

1. **Create the Stored Procedure**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) IS

v\_from\_balance NUMBER;

BEGIN

SELECT BALANCE INTO v\_from\_balance

FROM ACCOUNTS

WHERE ACCOUNT\_ID = p\_from\_account\_id

FOR UPDATE;

IF v\_from\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');

END IF;

UPDATE ACCOUNTS

SET BALANCE = BALANCE - p\_amount

WHERE ACCOUNT\_ID = p\_from\_account\_id;

UPDATE ACCOUNTS

SET BALANCE = BALANCE + p\_amount

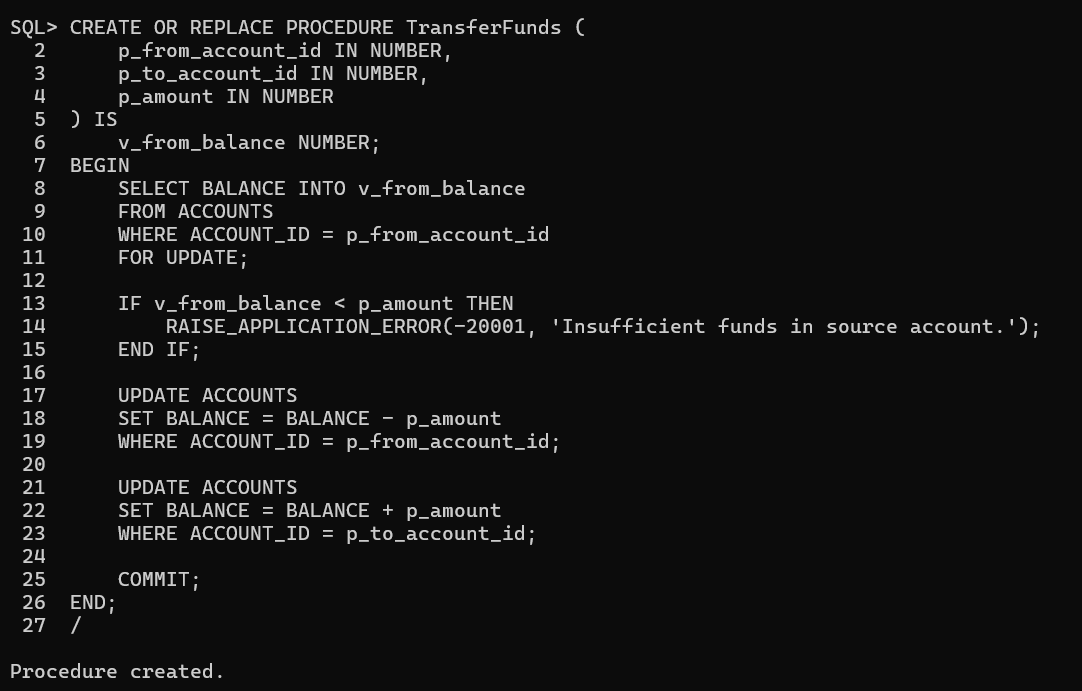
WHERE ACCOUNT\_ID = p\_to\_account\_id;

COMMIT;

END;

/

**Output**

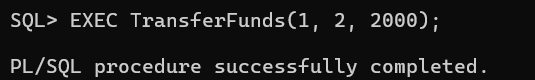


**Query**

1. **Execute the Procedure (e.g., Transfer 2000 from Alice to Bob)**

EXEC TransferFunds(1, 2, 2000);

**Output**

****

**Query**

1. **View Updated Balances**

SELECT \* FROM ACCOUNTS;

**Output**

