**Scenario 1**

CODE

create or replace PROCEDURE process\_monthly\_interest

IS

BEGIN

UPDATE ACCOUNTS

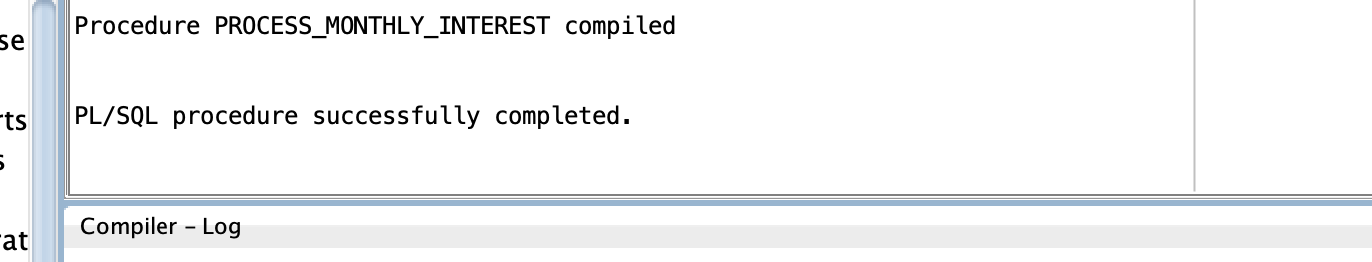
SET BALANCE = BALANCE \* 1.01

WHERE ACCOUNTTYPE = 'SAVINGS';

END;

exec process\_monthly\_interest;

OUTPUT



**Scenario 2**

CODE

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_deptno IN EMP.DEPTNO%TYPE,

p\_bonus\_percent IN NUMBER

)

AS

BEGIN

UPDATE EMP

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

WHERE DEPTNO = p\_deptno;

—COMMIT; -- Optional: remove if you want to control transactions from the caller

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

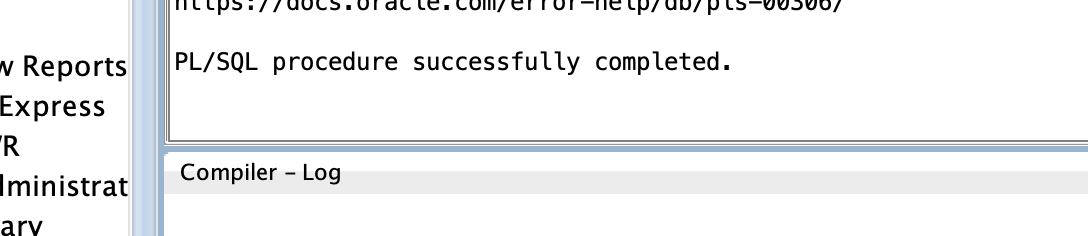
DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

exec UpdateEmployeeBonus(5);

OUTPUT



**Scenario 3**

CODE

create or replace PROCEDURE transferfunds (

from\_account INTEGER,

to\_account INTEGER,

amount NUMBER )

IS

fromacc\_balance customers.balance%TYPE;

fromacc\_cid customers.customerid%TYPE;

toacc\_cid customers.customerid%TYPE;

BEGIN

SELECT c1.customerid, c1.balance, c2.customerid

INTO fromacc\_cid, fromacc\_balance, toacc\_cid

FROM accounts a1

JOIN customers c1 ON a1.customerid = c1.customerid

JOIN accounts a2 ON a2.accountid = to\_account

JOIN customers c2 ON a2.customerid = c2.customerid

WHERE a1.accountid = from\_account

FOR UPDATE;

IF fromacc\_balance >= amount

THEN

UPDATE customers

SET balance = CASE

WHEN customerid = from\_account THEN balance - amount

WHEN customerid = to\_account THEN balance + amount

END

WHERE customerid IN (from\_account,to\_account);

ELSE

RAISE\_APPLICATION\_ERROR(-20001,'BALANCE IS TOO LOW!');

END IF;

END;

exec transferfunds;

OUTPUT

