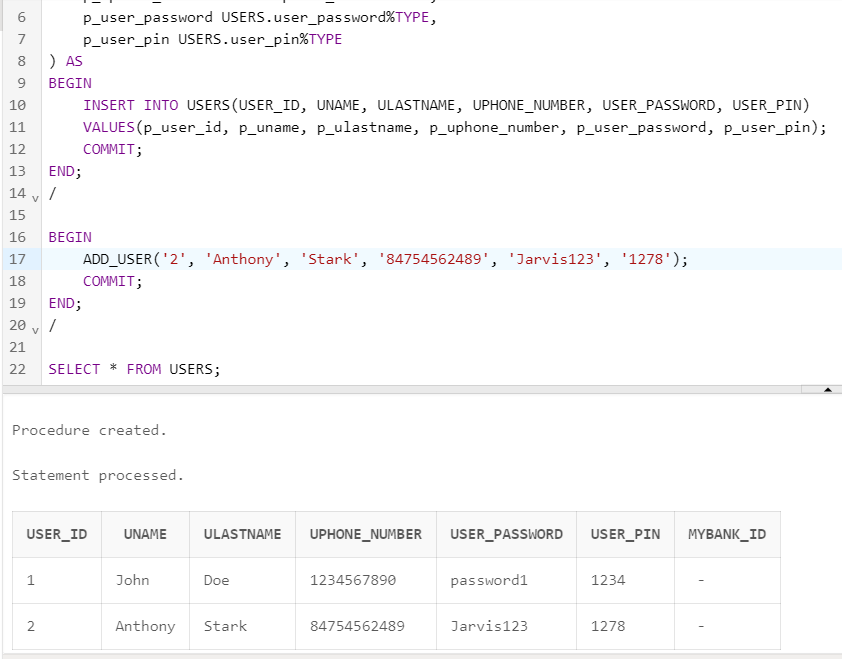
**Lab4. Creating Procedures**

1. Create at least 4 procedures for your database. You may use the anonymous blocks you created in Lab 2 as a basis.
2. At least 3 procedures must be with parameters.
3. During the defense you must show the execution of the procedures.



CREATE OR REPLACE PROCEDURE ADD\_USER (

p\_user\_id USERS.USER\_ID%TYPE,

p\_uname USERS.UNAME%TYPE,

p\_ulastname USERS.ULASTNAME%TYPE,

p\_uphone\_number USERS.uphone\_number%TYPE,

p\_user\_password USERS.user\_password%TYPE,

p\_user\_pin USERS.user\_pin%TYPE

) AS

BEGIN

INSERT INTO USERS(USER\_ID, UNAME, ULASTNAME, UPHONE\_NUMBER, USER\_PASSWORD, USER\_PIN)

VALUES(p\_user\_id, p\_uname, p\_ulastname, p\_uphone\_number, p\_user\_password, p\_user\_pin);

COMMIT;

END;

/

BEGIN

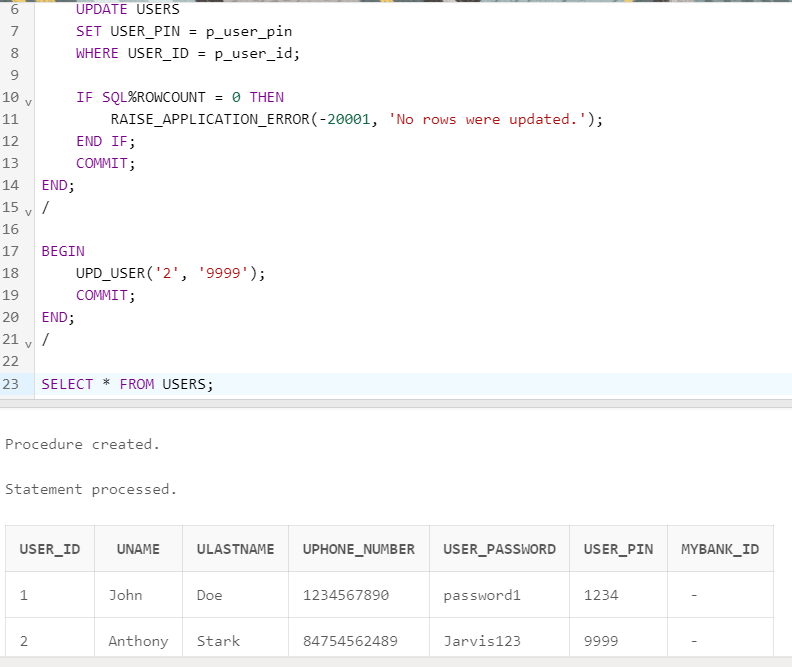
ADD\_USER('2', 'Anthony', 'Stark', '84754562489', 'Jarvis123', '1278');

COMMIT;

END;

/

SELECT \* FROM USERS;



CREATE OR REPLACE PROCEDURE UPD\_USER(

p\_user\_id USERS.USER\_ID%TYPE,

p\_user\_pin USERS.USER\_PIN%TYPE

) AS

BEGIN

UPDATE USERS

SET USER\_PIN = p\_user\_pin

WHERE USER\_ID = p\_user\_id;

IF SQL%ROWCOUNT = 0 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'No rows were updated.');

END IF;

COMMIT;

END;

/

BEGIN

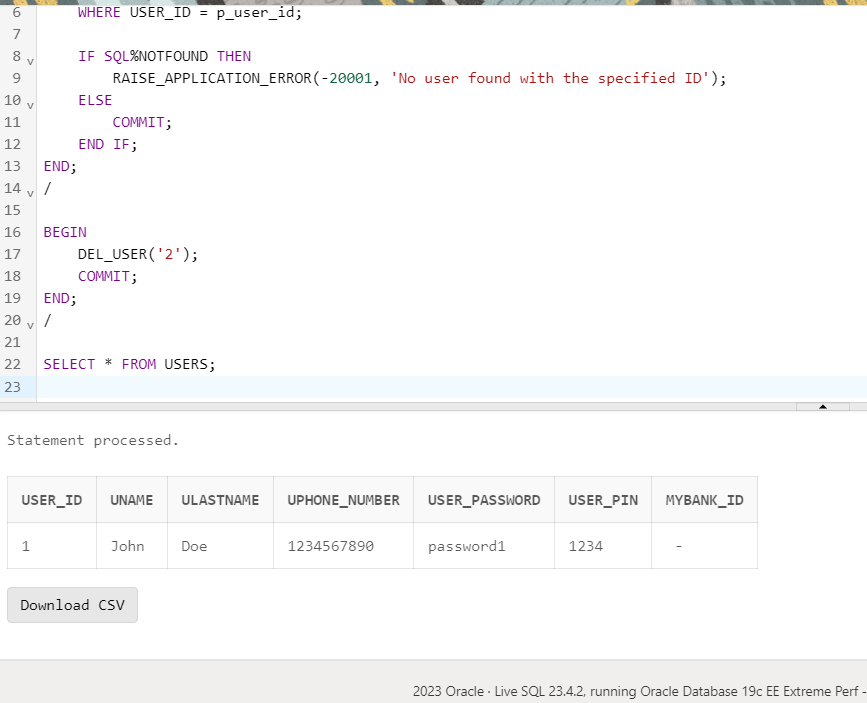
UPD\_USER('2', '9999');

COMMIT;

END;

/

SELECT \* FROM USERS;



CREATE OR REPLACE PROCEDURE DEL\_USER (

p\_user\_id USERS.USER\_ID%TYPE

) AS

BEGIN

DELETE FROM USERS

WHERE USER\_ID = p\_user\_id;

IF SQL%NOTFOUND THEN

RAISE\_APPLICATION\_ERROR(-20001, 'No user found with the specified ID');

ELSE

COMMIT;

END IF;

END;

/

BEGIN

DEL\_USER('2');

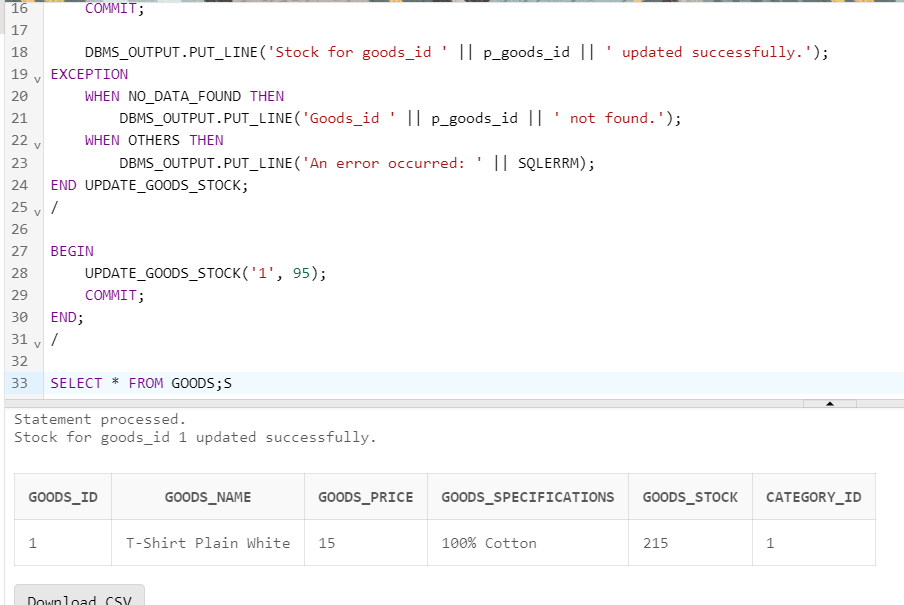
COMMIT;

END;

/

SELECT \* FROM USERS;





CREATE OR REPLACE PROCEDURE UPDATE\_GOODS\_STOCK(

p\_goods\_id IN GOODS.GOODS\_ID%TYPE,

p\_stock\_change IN NUMBER

) AS

v\_current\_stock NUMBER;

BEGIN

SELECT GOODS\_STOCK

INTO v\_current\_stock

FROM GOODS

WHERE GOODS\_ID = p\_goods\_id;

UPDATE GOODS

SET GOODS\_STOCK = v\_current\_stock + p\_stock\_change

WHERE GOODS\_ID = p\_goods\_id;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Stock for goods\_id ' || p\_goods\_id || ' updated successfully.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Goods\_id ' || p\_goods\_id || ' not found.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END UPDATE\_GOODS\_STOCK;

/

BEGIN

UPDATE\_GOODS\_STOCK('1', 95);

COMMIT;

END;

/

SELECT \* FROM GOODS;

Questions:

1. Define a procedure in PL/SQL
2. Describe the functionality of procedures
3. What types of parameters exist?
4. Explain the functionality of IN parameters
5. Explain the functionality of OUT parameters
6. Explain the functionality of IN OUT parameters
7. What are formal parameters?
8. What are actual parameters?
9. What types of notations for passing actual parameters exist in PL/SQL?
10. Describe positional notation for passing actual parameters
11. Describe named notation for passing actual parameters
12. Describe mixed notation for passing actual parameters