Name : Siva Rama Krishna Nallapati
Id No : 190031154

DATABASE & SYSTEM SECURITY PRACTICAL – 6

In-Lab:

Q1. Create a table 'app_users' with the following columns: id, username, password. 'id' is the primary key and 'username' is unique.

Sol)

CREATE TABLE app_users (id NUMBER(10) PRIMARY KEY, username VARCHAR2(30)UNIQUE, password VARCHAR2(40) NOT NULL);

```
SQL> CREATE TABLE app_users(id NUMBER(10) PRIMARY KEY, username VARCHAR2(30) UNIQUE, password VARCHAR2(40) NOT NULL);

Table created.

SQL>
```

Q2. Create a sequence 'app_users_seq'.

Sol)

1. Connect to system account.

Type '**CREATE SEQUENCE app_users_seq start with 3001;**' and press enter

```
SQL> CREATE SEQUENCE app_users_sequence START WITH 3001;

Sequence created.

SQL> _
```

Name : Siva Rama Krishna Nallapati
Id No : 190031154

Q3. Grant execute on dbms_crypto package to the user to be able to use that to hash.

Sol)

1. Connect to system as sysdba using 'connect system/system as sysdba'.
2. Now grant the execute privilege on dbms_crypto to system.

   grant execute on sys.dbms_crypto to system;

```
SQL> CONNECT System/root as SYSDBA;
Connected.
SQL> GRANT EXECUTE ON SYS.DBMS_CRYPTO TO System;

Grant succeeded.

SQL>
```

Q4. Create a function 'get_hash' (which uses dbms_crypto) in PL/SQL to hashthe given username and password using SH1 algorithm.

Sol)

1. Connect to system account using '**connect system/system**'.
2. Type '**ed**' in sql command line. A notepad file with the '.buf' extension will open up.
3. Type your function in that file and press 'ALT+F+X'. Save the file when prompted.

```
CREATE OR REPLACE FUNCTION get_hash (p_username IN VARCHAR2,
                p_password IN  VARCHAR2)
  RETURN VARCHAR2 AS
  l_salt VARCHAR2(30) := 'mysaltvalue';
 BEGIN
   RETURN DBMS_CRYPTO.HASH
(UTL_RAW.CAST_TO_RAW(UPPER(p_username) || l_salt ||
UPPER(p_password)),DBMS_CRYPTO.HASH_SH1);
 END;
```

Type '/' (slash) in the command line.

Name : Siva Rama Krishna Nallapati
Id No : 190031154

```
Run SQL Command Line

SQL> CREATE SEQUENCE app_users_sequence START WITH 3001;

Sequence created.

SQL> CONNECT System/root as SYSDBA;
Connected.
SQL> GRANT EXECUTE ON SYS.DBMS_CRYPTO TO System;

Grant succeeded.

SQL> CONNECT System/root;
Connected.
SQL> ed
Wrote file afiedt.buf
```

```
*afiedt.buf - Notepad
File Edit Format View Help
CREATE OR REPLACE FUNCTION get_hash (p_username IN VARCHAR2, p_password IN
RETURN VARCHAR2 AS
l_salt VARCHAR2(30) := 'mysaltvalue';
BEGIN
 RETURN DBMS_CRYPTO.HASH
(UTL_RAW.CAST_TO_RAW(UPPER(p_username) || l_salt || UPPER(p_password)),DBMS
END;

Ln 5, Col 2   100%   Windows (CRLF)   UTF-8
```

```
SQL> CONNECT System/root;
Connected.
SQL> ed
Wrote file afiedt.buf

  1  CREATE OR REPLACE FUNCTION get_hash (p_username IN VARCHAR2, p_password IN VARCHAR2)
  2  RETURN VARCHAR2 AS
  3  l_salt VARCHAR2(30) := 'mysaltvalue';
  4  BEGIN
  5   RETURN DBMS_CRYPTO.HASH
  6  (UTL_RAW.CAST_TO_RAW(UPPER(p_username) || l_salt || UPPER(p_password)),DBMS_CRYPTO.HASH_SH1);
  7* END;
SQL> /

Function created.

SQL> _
```

Q5. Create a procedure 'add_user' which executes the 'get_hash' function and stores the given username and hashed password in the table 'app_users'.

Sol)

1. Make sure you're connected to the system account.
2. Type '**ed**' in sql command line. A notepad file with the '.buf' extension will open up.
3. Type your procedure in that file and press 'ALT+F+X'. Save the file when prompted.

CREATE OR REPLACE PROCEDURE add_user (p_username IN VARCHAR2,p_password IN VARCHAR2) AS

BEGIN

INSERT INTO app_users

(id,username,password)

Name : Siva Rama Krishna Nallapati
Id No : 190031154

VALUES (

app_users_seq.NEXTVAL, UPPER(p_username),

get_hash(p_username, p_password)

);

```
SQL> ed
Wrote file afiedt.buf

  1  CREATE OR REPLACE PROCEDURE add_user (p_username IN VARCHAR2, p_password IN VARCHAR2) AS
  2  BEGIN
  3  INSERT INTO app_users ( id,username, password)
  4  VALUES (app_users_sequence.NEXTVAL, UPPER(p_username), get_hash(p_username, p_password));
  5  COMMIT;
  6* END;
  7  /

Procedure created.

SQL>
```

Q6. Execute the function 'add_user' with inputs as 'labtest' (for username) and 'labtest1' (for password) and display the table 'app_users' from Q5.

Sol)

execute add_user('labtest','labtest1');

select * from app_users;

```
SQL> EXECUTE add_user('labtest2', 'labtest12');

PL/SQL procedure successfully completed.

SQL> SELECT * FROM app_users;

        ID USERNAME
---------- ------------------------------
PASSWORD
----------------------------------------
      3001 LABTEST2
ED1080B40461DC76F2C1C1110C1B3B9437495E64


SQL>
```

The given username and hashed password will be displayed

Name : Siva Rama Krishna Nallapati
Id No : 190031154

Post-Lab:

Q1. Create a function 'get_hash' (which uses dbms_crypto) in PL/SQL to hash the given username and password using MD4 algorithm.

Sol)

1. Connect to system account using '**connect system/system**'.
2. Type '**ed**' in sql command line. A notepad file with the '.buf' extension will open up.
3. Type your function in that file and press 'ALT+F+X'. Save the file when prompted.

```
CREATE OR REPLACE FUNCTION get_hash (p_username IN VARCHAR2,
                p_password IN  VARCHAR2)
  RETURN VARCHAR2 AS
  l_salt VARCHAR2(30) := 'mysaltvalue';
 BEGIN
   RETURN
DBMS_CRYPTO.HASH(UTL_RAW.CAST_TO_RAW(UPPER(p_username) ||
l_salt || UPPER(p_password)),DBMS_CRYPTO.HASH_MD4);
 END;
```

4. Type '/' (slash) in the command line.
5. The function has been created.

```
SQL> CONNECT System/root;
Connected.
SQL> ed
Wrote file afiedt.buf

  1  CREATE OR REPLACE FUNCTION get_hash (p_username IN VARCHAR2, p_password IN VARCHAR2)
  2  RETURN VARCHAR2 AS
  3  l_salt VARCHAR2(30) := 'mysaltvalue'; BEGIN
  4  RETURN DBMS_CRYPTO.HASH(UTL_RAW.CAST_TO_RAW(UPPER(p_username) ||
  5* l_salt || UPPER(p_password)),DBMS_CRYPTO.HASH_MD5); END;
SQL> /

Function created.

SQL> _
```

Q2. Execute the function 'add_user' with inputs as 'labtest2' (for username) and

Name : Siva Rama Krishna Nallapati
Id No : 190031154

‘labtest12’ (for password) and display the table ‘app_users’.

Sol)

```
execute add_user('labtest2','labtest12');
select * from app_users;
```

Q3. Create a function ‘get_hash’ (which uses dbms_crypto) in PL/SQL to hash the given username and password using MD5 algorithm.

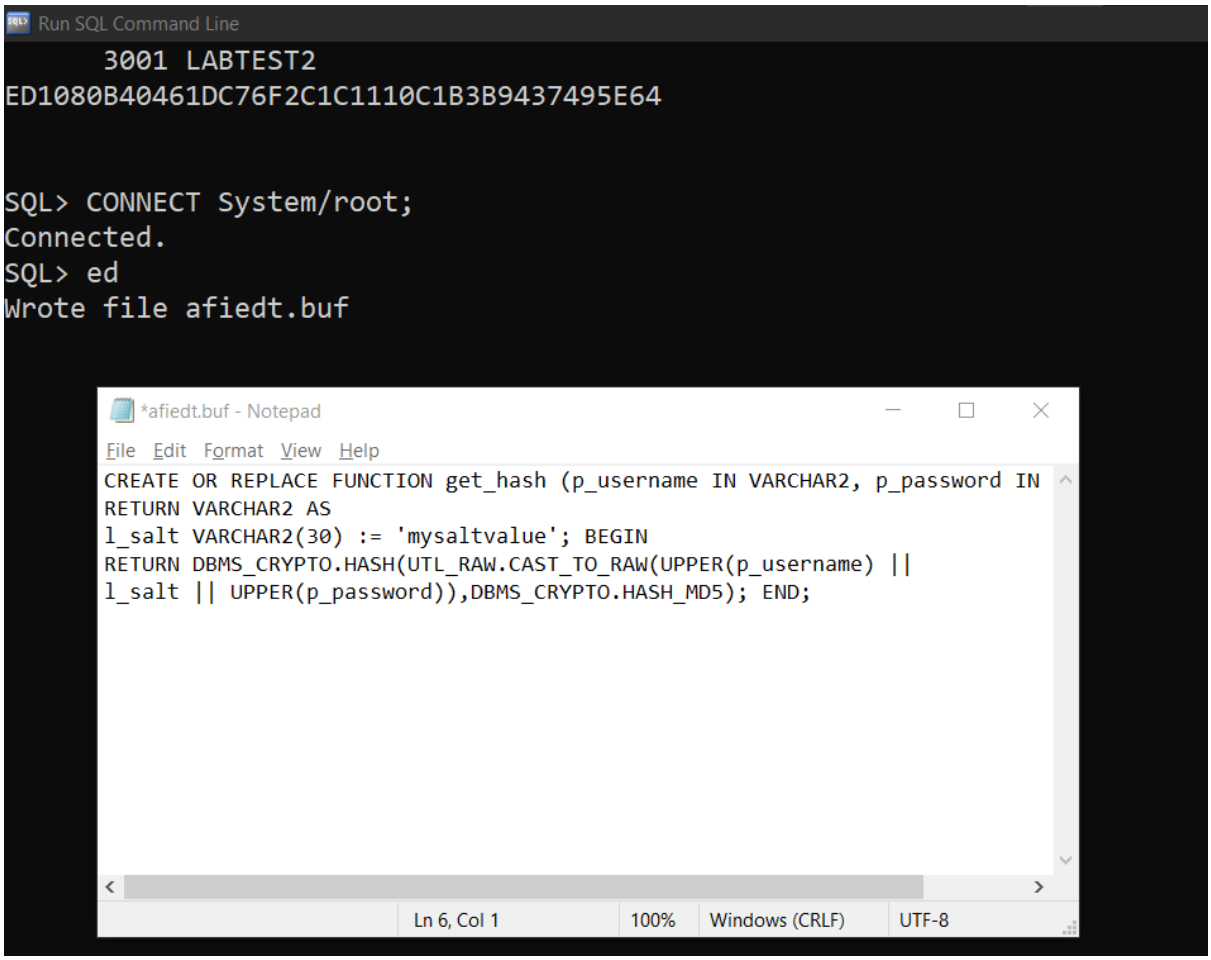
Sol)

1. Connect to system account using ‘**connect system/system**’.
2. Type ‘**ed**’ in sql command line. A notepad file with the ‘.buf’ extension will open up.
3. Type your function in that file and press ‘ALT+F+X’. Save the file when prompted.

Name : Siva Rama Krishna Nallapati
Id No : 190031154

```
CREATE OR REPLACE FUNCTION get_hash (p_username IN
                VARCHAR2,p_password IN  VARCHAR2)
   RETURN VARCHAR2 AS
   l_salt VARCHAR2(30) :=
  'mysaltvalue';BEGIN
    RETURN
DBMS_CRYPTO.HASH(UTL_RAW.CAST_TO_RAW(UPPER(p_username) ||
l_salt ||
  UPPER(p_password)),DBMS_CRYPTO.HASH_MD
  5);END;
```

4. Type '/' (slash) in the command line.
5. The function has been created.



Name : Siva Rama Krishna Nallapati
Id No : 190031154

Q4. Execute the function ‘add_user’ with inputs as ‘labtest3’ (for username) and‘labtest123’ (for password) and display the table ‘app_users’.

Sol)

select * from app_users;

```
SQL> EXECUTE add_user('labtest3', 'labtest123');

PL/SQL procedure successfully completed.

SQL>
```

```
SQL> SELECT * FROM app_users;

        ID USERNAME
---------- ------------------------------
PASSWORD
----------------------------------------
      3001 LABTEST2
ED1080B40461DC76F2C1C1110C1B3B9437495E64

      3002 LABTEST3
2ECFFC053F8D662C4C5370858B3EB5DA


SQL>
```