CREATE OR REPLACE FUNCTION ADD\_NEW\_Department( Department\_Name IN VARCHAR2 )

RETURN NUMBER IS

JDATE DATE ;

YEARS NUMBER ;

MSG VARCHAR2(100) ;

BEGIN

INSERT INTO DEPT VALUES( NUMBER\_ID\_SEQ.NEXTVAL, Department\_Name, NULL );

RETURN 1 ; --return the message

EXCEPTION

--you must return value from this section also

WHEN OTHERS THEN

RETURN 0 ;

END ;

/

CREATE OR REPLACE FUNCTION ADD\_NEW\_Teacher( Username IN VARCHAR2, Pass in VARCHAR2, FIRST\_NAME IN VARCHAR2, LAST\_NAME IN VARCHAR2,

Address in VARCHAR2, PHone\_Number IN VARCHAR2, Email\_Address IN VARCHAR2, JOIN\_DATE IN DATE, Dept\_Name IN VARCHAR2 )

RETURN NUMBER IS

Dept\_ID NUMBER;

SALARY NUMBER;

BEGIN

/\*SELECT Department\_ID INTO DEPT\_ID

FROM DEPT

WHERE DEPT.DEPARTMENT\_NAME = Dept\_Name;

\*/

--RETURN Dept\_ID;

Dept\_ID := Get\_Dept\_ID\_From\_Dept\_Name(Dept\_Name );

--RETURN Dept\_ID;

SALARY := ( (SYSDATE-JOIN\_DATE)\*10 + 50000);

INSERT INTO TEACHER VALUES( NUMBER\_ID\_SEQ.NEXTVAL, USERNAME, PASS, FIRST\_NAME, LAST\_NAME, ADDRESS, PHONE\_NUMBER, EMAIL\_ADDRESS,

SALARY, JOIN\_DATE, DEPT\_ID);

RETURN 1 ; --return the message

EXCEPTION

--you must return value from this section also

WHEN NO\_DATA\_FOUND THEN

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

RETURN -2;

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN -3;

WHEN INVALID\_NUMBER THEN

RETURN -4;

WHEN VALUE\_ERROR THEN

RETURN -5;

WHEN ZERO\_DIVIDE THEN

RETURN -6;

WHEN OTHERS THEN

RETURN -7;

END ;

/

SELECT \*

FROM TEACHER;

CREATE OR REPLACE FUNCTION Get\_Dept\_ID\_From\_Dept\_Name(Dept\_Name IN VARCHAR2 )

RETURN NUMBER IS

RET NUMBER;

BEGIN

RET := 0;

SELECT DEPARTMENT\_ID INTO RET

FROM DEPT

WHERE DEPT.DEPARTMENT\_NAME = Dept\_Name;

RETURN RET;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

RETURN -2;

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN -3;

END;

/

--It is showing compilation error in Navicat

--But running nicely in SQL Plus

CREATE OR REPLACE FUNCTION Do\_Username\_Password\_Exist(un IN VARCHAR2, pa IN VARCHAR2, userType IN VARCHAR2)

RETURN NUMBER IS

RET NUMBER;

BEGIN

RET := 0;

IF LOWER(userType) LIKE 'teacher' THEN

SELECT COUNT(\*) INTO RET

FROM TEACHER

WHERE USERNAME = un AND PASS = pa;

END IF;

IF LOWER(userType) LIKE 'student' THEN

SELECT COUNT(\*) INTO RET

FROM STUDENT

WHERE USERNAME = un AND PASS = pa;

END IF;

RETURN RET;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

RETURN -2;

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN -3;

END;

/

CREATE OR REPLACE FUNCTION ADD\_NEW\_Student( Username IN VARCHAR2, Pass in VARCHAR2, FIRST\_NAME IN VARCHAR2, LAST\_NAME IN VARCHAR2,

Address in VARCHAR2, PHone\_Number IN VARCHAR2, Email\_Address IN VARCHAR2, LEV IN NUMBER, TERM IN NUMBER, Dept\_Name IN VARCHAR2 )

RETURN NUMBER IS

Dept\_ID NUMBER;

BEGIN

/\*SELECT Department\_ID INTO DEPT\_ID

FROM DEPT

WHERE DEPT.DEPARTMENT\_NAME = Dept\_Name;

\*/

--RETURN Dept\_ID;

Dept\_ID := Get\_Dept\_ID\_From\_Dept\_Name(Dept\_Name );

--RETURN Dept\_ID;

INSERT INTO STUDENT

VALUES( NUMBER\_ID\_SEQ.NEXTVAL, USERNAME, PASS, FIRST\_NAME, LAST\_NAME, ADDRESS, PHONE\_NUMBER, EMAIL\_ADDRESS,

LEV, TERM, DEPT\_ID);

RETURN 1 ; --return the message

EXCEPTION

--you must return value from this section also

WHEN NO\_DATA\_FOUND THEN

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

RETURN -2;

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN -3;

WHEN INVALID\_NUMBER THEN

RETURN -4;

WHEN VALUE\_ERROR THEN

RETURN -5;

WHEN ZERO\_DIVIDE THEN

RETURN -6;

WHEN OTHERS THEN

RETURN -7;

END ;

/

CREATE OR REPLACE FUNCTION Get\_Dept\_Name\_From\_Dept\_ID(Dept\_ID IN NUMBER )

RETURN VARCHAR2 IS

RET VARCHAR2(200);

BEGIN

SELECT DEPARTMENT\_NAME INTO RET

FROM DEPT

WHERE DEPT.DEPARTMENT\_ID = Dept\_ID;

RETURN RET;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'NO\_DATA\_FOUND';

WHEN TOO\_MANY\_ROWS THEN

RETURN 'TOO\_MANY\_ROWS';

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN 'DUP\_VAL\_ON\_INDEX';

END;

/

CREATE OR REPLACE FUNCTION Get\_Head\_ID\_From\_Dept\_Name(Dept\_NAME IN VARCHAR2 )

RETURN NUMBER IS

RET NUMBER;

BEGIN

SELECT HEAD\_ID INTO RET

FROM DEPT

WHERE DEPT.DEPARTMENT\_NAME = Dept\_Name;

RETURN RET;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

RETURN -2;

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN -3;

END;

/

CREATE OR REPLACE FUNCTION Get\_DEPT\_ID\_From\_TEACHER\_ID(T\_ID IN NUMBER )

RETURN NUMBER IS

RET NUMBER;

BEGIN

SELECT DEPARTMENT\_ID INTO RET

FROM TEACHER

WHERE TEACHER\_ID = T\_ID;

RETURN RET;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

RETURN -2;

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN -3;

END;

/

CREATE OR REPLACE FUNCTION COUNT\_OF\_DEPT\_ID\_IN\_LOG(D\_ID IN NUMBER )

RETURN NUMBER IS

RET NUMBER;

BEGIN

SELECT COUNT(\*) INTO RET

FROM DEPARTMENT\_HEAD\_LOG

GROUP BY DEPARTMENT\_ID

HAVING DEPARTMENT\_ID = D\_ID;

RETURN RET;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

RETURN -2;

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN -3;

END;

/

CREATE OR REPLACE FUNCTION Get\_T\_ID\_From\_TEACHER\_UN(T\_UN IN VARCHAR2 )

RETURN NUMBER IS

RET NUMBER;

BEGIN

SELECT TEACHER\_ID INTO RET

FROM TEACHER

WHERE USERNAME = T\_UN;

RETURN RET;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

RETURN -2;

WHEN DUP\_VAL\_ON\_INDEX THEN

RETURN -3;

END;

/