2)

CREATE OR REPLACE PROCEDURE insert\_zip

(I\_ZIPCODE IN zipcode.zip%TYPE,

I\_CITY IN zipcode.city%TYPE,

I\_STATE IN zipcode.state%TYPE)

AS

v\_zipcode zipcode.zip%TYPE;

v\_city zipcode.city%TYPE;

v\_state zipcode.state%TYPE;

v\_dummy zipcode.zip%TYPE;

BEGIN

v\_zipcode := i\_zipcode;

v\_city := i\_city;

v\_state := i\_state;

SELECT zip

INTO v\_dummy

FROM zipcode

WHERE zip = v\_zipcode;

DBMS\_OUTPUT.PUT\_LINE('The zipcode '||V\_ZIPCODE||

' is already in the database. Try again.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

INSERT INTO ZIPCODE

VALUES (v\_zipcode, v\_city, v\_state, user, sysdate,

user, sysdate);

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('There is an error '||

'in insert\_zip.');

END;

/

SET SERVEROUTPUT ON

BEGIN

insert\_zip (48104, 'Ann Arbor', 'MI');

END;

/

BEGIN

insert\_zip (18104, 'Chicago', 'MI');

END;

/

SELECT \* FROM ZIPCODE

where state = 'MI';

ROLLBACK;

3)

CREATE OR REPLACE FUNCTION EXIST\_ZIP

(I\_ZIPCODE IN ZIPCODE.ZIP%TYPE)

RETURN BOOLEAN

AS

V\_DUMMY CHAR(1);

BEGIN

SELECT NULL

INTO V\_DUMMY

FROM ZIPCODE

WHERE ZIP = I\_ZIPCODE;

RETURN FALSE;

EXCEPTION

WHEN OTHERS THEN

RETURN TRUE;

END EXIST\_ZIP;

4)

CREATE OR REPLACE FUNCTION INSTRUCT\_STATUS

(P\_FIRST\_NAME IN INSTRUCTOR.FIRST\_NAME%TYPE,

P\_LAST\_NAME IN INSTRUCTOR.LAST\_NAME%TYPE)

RETURN VARCHAR2

AS

V\_INSTRUCTOR\_ID INSTRUCTOR.INSTRUCTOR\_ID%TYPE;

V\_SECTION\_COUNT NUMBER;

V\_STATUS VARCHAR2(100);

BEGIN

SELECT INSTRUCTOR\_ID

INTO V\_INSTRUCTOR\_ID

FROM INSTRUCTOR

WHERE FIRST\_NAME = P\_FIRST\_NAME

AND LAST\_NAME = P\_LAST\_NAME;

SELECT COUNT(\*)

INTO V\_SECTION\_COUNT

FROM SECTION

WHERE INSTRUCTOR\_ID = V\_INSTRUCTOR\_ID;

IF V\_SECTION\_COUNT >= 10 THEN

V\_STATUS :=

P\_LAST\_NAME||' This Instructor will teach '||V\_SECTION\_COUNT||

' courses and needs a vacation';

ELSIF V\_SECTION\_COUNT <= 9 AND V\_SECTION\_COUNT >0 THEN

V\_STATUS :=

P\_LAST\_NAME||' This Instructor will teach '||V\_SECTION\_COUNT||

' courses';

ELSE

V\_STATUS :=

P\_LAST\_NAME||' This Instructor is NOT scheduled teach ';

END IF;

RETURN V\_STATUS;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

V\_STATUS :=

'There is NO such instructor.';

RETURN V\_STATUS;

WHEN OTHERS THEN

V\_STATUS :=

'There is an error in the function.';

RETURN V\_STATUS;

END;

/

SELECT INSTRUCT\_STATUS(FIRST\_NAME, LAST\_NAME)

FROM INSTRUCTOR

ORDER by LAST\_NAME;

/