Q1

SET SERVEROUTPUT ON

SET VERIFY OFF

VARIABLE LNAME VARCHAR2(225);

VARIABLE PNUM VARCHAR2(25);

VARIABLE ZIP VARCHAR2(25);

CREATE OR REPLACE PROCEDURE find\_stud

(

STUD\_ID IN NUMBER,

LASTNAME OUT VARCHAR2,

PHONENUM OUT VARCHAR2,

ZIPZ OUT VARCHAR2

)

IS

V\_STUDID NUMBER;

BEGIN

SELECT ZIP INTO ZIPZ FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

SELECT LAST\_NAME INTO LASTNAME FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

SELECT PHONE INTO PHONENUM FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

SELECT STUDENT\_ID INTO V\_STUDID FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

DBMS\_OUTPUT.PUT\_LINE('Student with the Id of: '||STUD\_ID||' is '||LASTNAME||' with the phone# '||PHONENUM||' and who belongs to zip code '||ZIPZ);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('There is NO Student with the Id of: '||V\_STUDID);

END;

/

EXECUTE find\_stud(110, :LNAME, :PNUM, :ZIP);

PRINT :LNAME;

PRINT :PNUM;

PRINT :ZIP;

Q2

SET SERVEROUTPUT ON

SET VERIFY OFF

CREATE OR REPLACE PROCEDURE drop\_stud

(

STUD\_ID IN NUMBER,

FLAG IN CHAR

) IS

v\_flag char(1);

v\_rows NUMBER;

v\_tot NUMBER;

noStudent EXCEPTION;

BEGIN

v\_rows := 0;

v\_tot := 0;

SELECT COUNT(\*) INTO v\_rows FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

IF (v\_rows = 0) THEN

raise noStudent;

END IF;

v\_rows := 0;

IF (FLAG = '') THEN

v\_flag := 'R';

ELSE

v\_flag := FLAG;

END IF;

IF (v\_flag = 'R') THEN

SELECT count(\*) into v\_rows from enrollment where STUDENT\_ID = STUD\_ID;

IF (v\_rows = 0) THEN

DELETE FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

DBMS\_OUTPUT.PUT\_LINE('Student with ID : '||STUD\_ID||' is removed. He/she was NOT enrolled in any courses');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Student with ID: '||STUD\_ID||' is enrolled in one or more courses. His/her removal is denied.');

END IF;

ELSIF (v\_flag = 'C') THEN

SELECT COUNT(\*) INTO v\_rows FROM GRADE WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM GRADE WHERE STUDENT\_ID = STUD\_ID;

SELECT COUNT(\*) INTO v\_rows FROM ENROLLMENT WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM ENROLLMENT WHERE STUDENT\_ID = STUD\_ID;

SELECT COUNT(\*) INTO v\_rows FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

DBMS\_OUTPUT.PUT\_LINE('Student with the Id of : '||STUD\_ID||' is removed. Total # of rows deleted: '||v\_tot);

END IF;

EXCEPTION

WHEN noStudent THEN

DBMS\_OUTPUT.PUT\_LINE('Student with the Id of: '||STUD\_ID||' does NOT exist. Try again.');

END;

/

EXECUTE drop\_stud(210, 'R');

EXECUTE drop\_stud(410, 'R');

EXECUTE drop\_stud(310, 'R');

ROLLBACK;

EXECUTE drop\_stud(110, 'C');

ROLLBACK;

Q3

SET SERVEROUTPUT ON

SET VERIFY OFF

VARIABLE LNAME VARCHAR2(225);

VARIABLE PNUM VARCHAR2(25);

VARIABLE ZIP VARCHAR2(25);

CREATE OR REPLACE PACKAGE manage\_stud IS

PROCEDURE find\_stud

(

STUD\_ID IN NUMBER,

LASTNAME OUT VARCHAR2,

PHONENUM OUT VARCHAR2,

ZIPZ OUT VARCHAR2

);

PROCEDURE drop\_stud

(

STUD\_ID IN NUMBER,

FLAG IN CHAR

);

END manage\_stud;

/

CREATE OR REPLACE PACKAGE BODY manage\_stud IS

PROCEDURE find\_stud

(

STUD\_ID IN NUMBER,

LASTNAME OUT VARCHAR2,

PHONENUM OUT VARCHAR2,

ZIPZ OUT VARCHAR2

)

IS

V\_STUDID NUMBER;

BEGIN

SELECT ZIP INTO ZIPZ FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

SELECT LAST\_NAME INTO LASTNAME FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

SELECT PHONE INTO PHONENUM FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

SELECT STUDENT\_ID INTO V\_STUDID FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

DBMS\_OUTPUT.PUT\_LINE('Student with the Id of: '||STUD\_ID||' is '||LASTNAME||' with the phone# '||PHONENUM||' and who belongs to zip code '||ZIPZ);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('There is NO Student with the Id of: '||V\_STUDID);

END find\_stud;

PROCEDURE drop\_stud

(

STUD\_ID IN NUMBER,

FLAG IN CHAR

) IS

v\_flag char(1);

v\_rows NUMBER;

v\_tot NUMBER;

noStudent EXCEPTION;

BEGIN

v\_rows := 0;

v\_tot := 0;

SELECT COUNT(\*) INTO v\_rows FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

IF (v\_rows = 0) THEN

raise noStudent;

END IF;

v\_rows := 0;

IF (FLAG = '') THEN

v\_flag := 'R';

ELSE

v\_flag := FLAG;

END IF;

IF (v\_flag = 'R') THEN

SELECT count(\*) into v\_rows from enrollment where STUDENT\_ID = STUD\_ID;

IF (v\_rows = 0) THEN

DELETE FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

DBMS\_OUTPUT.PUT\_LINE('Student with ID : '||STUD\_ID||' is removed. He/she was NOT enrolled in any courses');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Student with ID: '||STUD\_ID||' is enrolled in one or more courses. His/her removal is denied.');

END IF;

ELSIF (v\_flag = 'C') THEN

SELECT COUNT(\*) INTO v\_rows FROM GRADE WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM GRADE WHERE STUDENT\_ID = STUD\_ID;

SELECT COUNT(\*) INTO v\_rows FROM ENROLLMENT WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM ENROLLMENT WHERE STUDENT\_ID = STUD\_ID;

SELECT COUNT(\*) INTO v\_rows FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

DBMS\_OUTPUT.PUT\_LINE('Student with the Id of : '||STUD\_ID||' is removed. Total # of rows deleted: '||v\_tot);

END IF;

EXCEPTION

WHEN noStudent THEN

DBMS\_OUTPUT.PUT\_LINE('Student with the Id of: '||STUD\_ID||' does NOT exist. Try again.');

END drop\_stud;

END manage\_stud;

/

EXECUTE find\_stud(110, :LNAME, :PNUM, :ZIP);

PRINT :LNAME;

PRINT :PNUM;

PRINT :ZIP;

EXECUTE drop\_stud(310, 'R');

ROLLBACK;

EXECUTE drop\_stud(110, 'C');

ROLLBACK;

Q4

SET SERVEROUTPUT ON

SET VERIFY OFF

VARIABLE LNAME VARCHAR2(225);

VARIABLE FNAME VARCHAR2(225);

CREATE OR REPLACE FUNCTION valid\_stud (

StudID IN STUDENT.STUDENT\_ID%TYPE

)

RETURN BOOLEAN IS

v\_rows NUMBER;

BEGIN

v\_rows := 0;

SELECT COUNT(\*) INTO v\_rows FROM STUDENT WHERE STUDENT\_ID = StudID;

if(v\_rows = 0) THEN

return false;

else

return true;

end if;

END valid\_stud;

/

CREATE OR REPLACE PACKAGE manage\_stud IS

PROCEDURE find\_stud

(

ID IN NUMBER,

LASTNAME OUT VARCHAR2,

FIRSTNAME OUT VARCHAR2

);

PROCEDURE drop\_stud

(

STUD\_ID IN NUMBER,

FLAG IN CHAR

);

END manage\_stud;

/

CREATE OR REPLACE PACKAGE BODY manage\_stud IS

PROCEDURE find\_stud

(

ID IN NUMBER,

LASTNAME OUT VARCHAR2,

FIRSTNAME OUT VARCHAR2

)

IS

V\_STUDID NUMBER;

BEGIN

SELECT LAST\_NAME INTO LASTNAME FROM STUDENT WHERE STUDENT\_ID = ID;

SELECT FIRST\_NAME INTO FIRSTNAME FROM STUDENT WHERE STUDENT\_ID = ID;

SELECT STUDENT\_ID INTO V\_STUDID FROM STUDENT WHERE STUDENT\_ID = ID;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('There is NO Student with the Id of: '||V\_STUDID);

END find\_stud;

PROCEDURE drop\_stud

(

STUD\_ID IN NUMBER,

FLAG IN CHAR

) IS

v\_flag char(1);

v\_rows NUMBER;

v\_tot NUMBER;

noStudent EXCEPTION;

BEGIN

v\_rows := 0;

v\_tot := 0;

SELECT COUNT(\*) INTO v\_rows FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

IF (v\_rows = 0) THEN

raise noStudent;

END IF;

v\_rows := 0;

IF (FLAG = '') THEN

v\_flag := 'R';

ELSE

v\_flag := FLAG;

END IF;

IF (v\_flag = 'R') THEN

SELECT count(\*) into v\_rows from enrollment where STUDENT\_ID = STUD\_ID;

IF (v\_rows = 0) THEN

DELETE FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

DBMS\_OUTPUT.PUT\_LINE('Student with ID : '||STUD\_ID||' is removed. He/she was NOT enrolled in any courses');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Student with ID: '||STUD\_ID||' is enrolled in one or more courses. His/her removal is denied.');

END IF;

ELSIF (v\_flag = 'C') THEN

SELECT COUNT(\*) INTO v\_rows FROM GRADE WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM GRADE WHERE STUDENT\_ID = STUD\_ID;

SELECT COUNT(\*) INTO v\_rows FROM ENROLLMENT WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM ENROLLMENT WHERE STUDENT\_ID = STUD\_ID;

SELECT COUNT(\*) INTO v\_rows FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

v\_tot := v\_tot + v\_rows;

DELETE FROM STUDENT WHERE STUDENT\_ID = STUD\_ID;

DBMS\_OUTPUT.PUT\_LINE('Student with the Id of : '||STUD\_ID||' is removed. Total # of rows deleted: '||v\_tot);

END IF;

EXCEPTION

WHEN noStudent THEN

DBMS\_OUTPUT.PUT\_LINE('Student with the Id of: '||STUD\_ID||' does NOT exist. Try again.');

END drop\_stud;

END manage\_stud;

/

EXECUTE manage\_stud.find\_stud(110, :LNAME, :FNAME);

PRINT :LNAME;

PRINT :FNAME;

Q5

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT Reg\_ID NUMBER PROMPT 'Enter Region Id: ';

DECLARE

TYPE countryNames\_type IS TABLE OF COUNTRIES.COUNTRY\_NAME%TYPE

INDEX BY BINARY\_INTEGER;

countryNames countryNames\_type;

CURSOR c1 IS

SELECT region\_id, COUNTRIES.country\_id, COUNTRY\_NAME FROM countries LEFT JOIN LOCATIONS ON COUNTRIES.COUNTRY\_ID = LOCATIONS.COUNTRY\_ID

WHERE LOCATIONS.COUNTRY\_ID IS NULL

ORDER BY COUNTRY\_NAME;

noCity c1%ROWTYPE;

CURSOR c2 IS

SELECT region\_id, COUNTRIES.country\_id, country\_name

FROM COUNTRIES

LEFT JOIN LOCATIONS

ON COUNTRIES.COUNTRY\_ID = LOCATIONS.COUNTRY\_ID

WHERE LOCATIONS.COUNTRY\_ID IS NULL AND COUNTRIES.REGION\_ID = &Reg\_ID ORDER BY COUNTRY\_NAME;

noRegion c2%ROWTYPE;

CURSOR c3 IS

SELECT region\_id, COUNTRIES.country\_id, COUNTRY\_NAME FROM countries LEFT JOIN LOCATIONS ON COUNTRIES.COUNTRY\_ID = LOCATIONS.COUNTRY\_ID

WHERE LOCATIONS.COUNTRY\_ID IS NULL for update

ORDER BY COUNTRY\_NAME;

cityUpdate c3%ROWTYPE;

v\_rows NUMBER;

v\_regID varchar2(7);

v\_valueGet NUMBER;

v\_2ndval varchar2(40);

v\_2ndlast varchar2(40);

BEGIN

SELECT COUNT(\*) INTO v\_rows FROM REGIONS WHERE REGION\_ID = &Reg\_ID;

OPEN c3;

LOOP

FETCH c3 INTO noRegion;

EXIT WHEN c3%NOTFOUND;

UPDATE COUNTRIES

SET

flag = ('EMPTY\_' + noRegion.REGION\_ID)

WHERE

CURRENT OF c3;

END LOOP;

CLOSE c3;

v\_rows := 1;

OPEN c1;

LOOP

FETCH c1 INTO noCity;

EXIT WHEN c1%NOTFOUND;

countryNames(v\_rows) := noCity.COUNTRY\_NAME;

v\_rows := v\_rows + 5;

END LOOP;

CLOSE c1;

v\_rows := 0;

v\_valueGet := countryNames.FIRST;

WHILE v\_valueGet IS NOT NULL LOOP

v\_rows := v\_rows + 1;

DBMS\_OUTPUT.PUT\_LINE('Index Table Key: '||v\_valueGet||' has a value of '||countryNames(v\_valueGet));

v\_valueGet := countryNames.NEXT(v\_valueGet);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('=====================================================');

DBMS\_OUTPUT.PUT\_LINE('Total number of elements in index tabl or Number of countries with NO cities is: '||v\_rows);

v\_2ndval := countryNames.FIRST;

v\_2ndval := countryNames.NEXT(v\_2ndval);

DBMS\_OUTPUT.PUT\_LINE('Second element in index table is: '||v\_2ndval);

v\_2ndlast := countryNames.LAST;

v\_2ndlast := countryNames.PRIOR(v\_2ndval);

DBMS\_OUTPUT.PUT\_LINE('Second element in index table is: '||v\_2ndlast);

DBMS\_OUTPUT.PUT\_LINE('=====================================================');

v\_rows := 0;

OPEN c2;

LOOP

FETCH c2 INTO noRegion;

EXIT WHEN c2%NOTFOUND;

v\_rows := v\_rows + 1;

DBMS\_OUTPUT.PUT\_LINE('In the region '||&Reg\_ID||' There is country '||noRegion.country\_name||' with NO city.');

END LOOP;

CLOSE c2;

DBMS\_OUTPUT.PUT\_LINE('=====================================================');

DBMS\_OUTPUT.PUT\_LINE('Total # of countries with NO cities listed in the Region '||&Reg\_ID||' is: '||v\_rows);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('This region ID does NOT exist: '||&Reg\_ID);

END;

/

SELECT \* FROM COUNTRIES;

ROLLBACK;