Instructor: Nebojsa Conkic  
Date: 10/19/2017

Assignment1

**DBS501**

Group Member:

Shan Zhang, #113004154

XiaoChen Wang, #015297153

Anthony Nguyen

Contents

Answer:

[Q1 2](#_Toc496379733)

[Q2 4](#_Toc496379734)

[Q3 6](#_Toc496379735)

[Q4 8](#_Toc496379736)

[Answer: 10](#_Toc496379737)

[Q1 – output 10](#_Toc496379738)

[Q2-output 12](#_Toc496379739)

[Q3-output 14](#_Toc496379740)

[Q4-ouput 18](#_Toc496379741)

# Q1

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT ctyid PROMPT 'Enter value for country:'

DECLARE

v\_ctyid VARCHAR2(30) := '&ctyid';

v\_count NUMBER(5);

v\_ctyname countries.country\_name%TYPE;

v\_ctynameb VARCHAR2(5);

v\_charprovince VARCHAR2(50);

v\_char VARCHAR2(5);

r\_locations locations%ROWTYPE;

BEGIN

SELECT LENGTH(street\_address), city INTO v\_count, v\_ctyname FROM locations WHERE state\_province IS NULL AND UPPER(country\_id) = UPPER(v\_ctyid);

SELECT substr(v\_ctyname,1,1) INTO v\_ctynameb FROM dual;

IF v\_ctynameb in ('A', 'B', 'E', 'F') THEN

v\_char := '\*';

ELSIF v\_ctynameb in ('C', 'D', 'E', 'H') THEN

v\_char := '&';

ELSE

v\_char := '#';

END IF;

FOR i IN 1..v\_count LOOP

v\_charprovince := v\_charprovince || v\_char;

END LOOP;

UPDATE locations SET state\_province = v\_charprovince WHERE state\_province IS NULL AND UPPER(country\_id) = UPPER(v\_ctyid);

DBMS\_OUTPUT.PUT\_LINE('City '|| v\_ctyname || ' has modified its province to ' || v\_charprovince);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('This country has NO cities listed.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('This country has MORE THAN ONE City without province listed.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Other errors');

END;

/

SELECT \* FROM locations WHERE substr(state\_province,1,1) in ('\*', '&', '#');

ROLLBACK;

/

# Q2

--1, add column

-- ALTER TABLE countries ADD (flag VARCHAR2(7));

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT inputregid PROMPT 'Enter value for region: ';

DECLARE

v\_cid countries.country\_id%TYPE;

v\_cname countries.country\_name%TYPE;

v\_rid countries.region\_id%TYPE;

v\_regid NUMBER := &inputregid;

v\_cnt NUMBER;

v\_flag VARCHAR2(30);

cursor c\_rid is

SELECT c.country\_id, c.REGION\_ID

FROM countries c LEFT JOIN locations l ON c.country\_id = l.country\_id

WHERE l.city IS NULL AND c.region\_id in (select region\_id from regions)

order by c.region\_id, c.country\_name;

BEGIN

SELECT c.country\_id, c.country\_name, c.REGION\_ID

INTO v\_cid, v\_cname, v\_rid

FROM countries c

LEFT JOIN locations l ON c.country\_id = l.country\_id

WHERE l.city IS NULL AND region\_id = v\_regid;

DBMS\_OUTPUT.PUT\_LINE('In the region '|| v\_regid || ' there is ONE country ' || v\_cname || ' with NO city.');

SELECT count(\*) INTO v\_cnt

FROM countries c LEFT JOIN locations l ON c.country\_id = l.country\_id

WHERE l.city IS NULL;

DBMS\_OUTPUT.PUT\_LINE('Number of countries with NO cities listed is: ' || v\_cnt);

FOR r in c\_rid LOOP

v\_flag := 'Empty\_' || r.region\_id;

UPDATE countries SET flag = v\_flag WHERE region\_id = r.region\_id AND country\_id = r.country\_id;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('This region ID does NOT exist: '|| v\_regid);

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('This region ID has MORE THAN ONE country without cities listed: '|| v\_regid);

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Other errors');

END;

/

SELECT c.\*

FROM countries c

LEFT JOIN locations l ON c.country\_id = l.country\_id

WHERE l.city IS NULL and c.flag is not null

order by c.region\_id, c.country\_name;

ROLLBACK;

/

# Q3

--1, add column

-- ALTER TABLE countries ADD (flag VARCHAR2(7));

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT inputregid PROMPT 'Enter value for region: ';

DECLARE

v\_regid NUMBER := &inputregid;

v\_cnt NUMBER;

v\_flag VARCHAR2(30);

cnt NUMBER := 0;

CURSOR c1 IS

SELECT c.country\_id, c.country\_name, c.REGION\_ID

FROM countries c LEFT JOIN locations l ON c.country\_id = l.country\_id

WHERE l.city IS NULL AND region\_id = v\_regid;

CURSOR c2 IS

SELECT c.country\_id, c.country\_name, c.REGION\_ID

FROM countries c LEFT JOIN locations l ON c.country\_id = l.country\_id

WHERE l.city IS NULL AND c.region\_id in (select region\_id from regions)

ORDER BY c.country\_name;

TYPE countryname IS TABLE OF countries.country\_name%TYPE INDEX BY VARCHAR2(30);

countryname\_list countryname;

idx integer := -4;

BEGIN

FOR r in c1 LOOP

cnt := cnt + 1;

END LOOP;

IF cnt = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This region ID does NOT exist: '|| v\_regid);

ELSE

FOR r2 in c2 LOOP

v\_flag := 'Empty\_' || r2.region\_id;

UPDATE countries SET flag = v\_flag WHERE region\_id = r2.region\_id AND country\_id = r2.country\_id;

idx := idx + 5;

countryname\_list(idx) := r2.country\_name;

DBMS\_OUTPUT.PUT\_LINE('Index Table Key: '||idx||' has a value of '||countryname\_list(idx));

END LOOP;

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

DBMS\_OUTPUT.PUT\_LINE('Total number of elements in the Index Table or Number of countries with NO cities listed is: ' || countryname\_list.COUNT);

DBMS\_OUTPUT.PUT\_LINE('Second element (Country) in the Index Table is: '|| countryname\_list(countryname\_list.FIRST + 5));

DBMS\_OUTPUT.PUT\_LINE('Before the last element (Country) in the Index Table is: '|| countryname\_list(46));

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

FOR r3 in c1 LOOP

DBMS\_OUTPUT.PUT\_LINE('In the region '|| v\_regid || ' there is ONE country ' || r3.country\_name || ' with NO city.');

END LOOP;

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

DBMS\_OUTPUT.PUT\_LINE('Total Number of countries with NO cities listed in the Region '||v\_regid||' is: '|| cnt);

END IF;

END;

/

SELECT c.\*

FROM countries c LEFT JOIN locations l ON c.country\_id = l.country\_id

WHERE l.city IS NULL and c.flag is not null

ORDER BY c.region\_id, c.country\_name;

ROLLBACK;

/

# Q4

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT inputdesc PROMPT 'Enter the piece of the course description in UPPER case: ';

ACCEPT inputlname PROMPT 'Enter the beginning of the Instructor last name in UPPER case: ';

DECLARE

CURSOR c1 is

SELECT course.course\_no cno, course.DESCRIPTION cdes, section.section\_id secid, instructor.last\_name ilname, section.section\_no secno

FROM course, section, instructor

WHERE course.course\_no = section.course\_no

AND section.instructor\_id = instructor.instructor\_id

AND UPPER(DESCRIPTION) LIKE UPPER('%&inputdesc%')

AND last\_name LIKE UPPER('&inputlname%')

ORDER BY 1 desc;

v\_count\_secid NUMBER;

CURSOR c2(v\_section\_id NUMBER) is

SELECT COUNT(\*) ecount

FROM enrollment

WHERE section\_id = v\_section\_id;

row\_cnt NUMBER := 0;

BEGIN

FOR r1 IN c1

LOOP

DBMS\_OUTPUT.PUT\_LINE('Course No: ' || r1.cno ||' '|| r1.cdes ||' with Section Id: '

|| r1.secid || ' is taught by '|| r1.ilname || ' in the Course Section: ' || r1.secno);

FOR r2 in c2(r1.secid)

LOOP

DBMS\_OUTPUT.PUT\_LINE(' This Section Id has an enrollment of: '|| r2.ecount);

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

row\_cnt := row\_cnt + r2.ecount;

END LOOP;

END LOOP;

IF row\_cnt = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('There is NO data for this input match between the course description piece and the surname start of Instructor. Try again!');

ELSE

DBMS\_OUTPUT.PUT\_LINE('This input match has a total enrollment of: ' || row\_cnt || ' students.');

END IF;

END;

/

# Answer:

# Q1 – output



City Hiroshima has modified its province to &&&&&&&&&&&&&&&

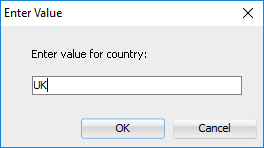
PL/SQL procedure successfully completed.

LOCATION\_ID STREET\_ADDRESS POSTAL\_CODE CITY STATE\_PROVINCE CO

----------- ---------------------------------------- ------------ ------------------------------ ------------------------- --

1300 9450 Kamiya-cho 6823 Hiroshima &&&&&&&&&&&&&&& JP

Rollback complete.



City London has modified its province to ##############

PL/SQL procedure successfully completed.

LOCATION\_ID STREET\_ADDRESS POSTAL\_CODE CITY STATE\_PROVINCE CO

----------- ---------------------------------------- ------------ ------------------------------ ------------------------- --

2400 8204 Arthur St London ############## UK

Rollback complete.

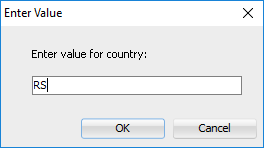


This country has MORE THAN ONE City without province listed.

PL/SQL procedure successfully completed.

no rows selected

Rollback complete.



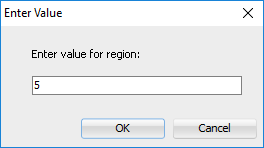
This country has NO cities listed.

PL/SQL procedure successfully completed.

no rows selected

Rollback complete.

# Q2-output

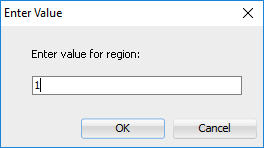


This region ID does NOT exist: 5

PL/SQL procedure successfully completed.

no rows selected

Rollback complete.

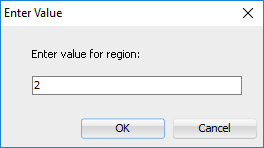


This region ID has MORE THAN ONE country without cities listed: 1

PL/SQL procedure successfully completed.

no rows selected

Rollback complete.



In the region 2 there is ONE country Argentina with NO city.

Number of countries with NO cities listed is: 11

PL/SQL procedure successfully completed.

CO COUNTRY\_NAME REGION\_ID FLAG

-- ---------------------------------------- ---------- -------

BE Belgium 1 Empty\_1

DK Denmark 1 Empty\_1

FR France 1 Empty\_1

AR Argentina 2 Empty\_2

HK HongKong 3 Empty\_3

EG Egypt 4 Empty\_4

IL Israel 4 Empty\_4

KW Kuwait 4 Empty\_4

NG Nigeria 4 Empty\_4

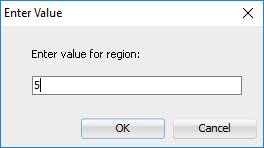
ZM Zambia 4 Empty\_4

ZW Zimbabwe 4 Empty\_4

11 rows selected.

Rollback complete.

# Q3-output

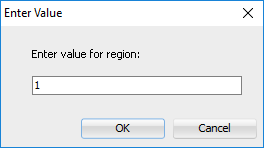


This region ID does NOT exist: 5

PL/SQL procedure successfully completed.

no rows selected

Rollback complete.



Index Table Key: 1 has a value of Argentina

Index Table Key: 6 has a value of Belgium

Index Table Key: 11 has a value of Denmark

Index Table Key: 16 has a value of Egypt

Index Table Key: 21 has a value of France

Index Table Key: 26 has a value of HongKong

Index Table Key: 31 has a value of Israel

Index Table Key: 36 has a value of Kuwait

Index Table Key: 41 has a value of Nigeria

Index Table Key: 46 has a value of Zambia

Index Table Key: 51 has a value of Zimbabwe

======================================================================

Total number of elements in the Index Table or Number of countries with NO cities listed is: 11

Second element (Country) in the Index Table is: Belgium

Before the last element (Country) in the Index Table is: Zambia

======================================================================

In the region 1 there is ONE country France with NO city.

In the region 1 there is ONE country Belgium with NO city.

In the region 1 there is ONE country Denmark with NO city.

======================================================================

Total Number of countries with NO cities listed in the Region 1 is: 3

PL/SQL procedure successfully completed.

CO COUNTRY\_NAME REGION\_ID FLAG

-- ---------------------------------------- ---------- -------

BE Belgium 1 Empty\_1

DK Denmark 1 Empty\_1

FR France 1 Empty\_1

AR Argentina 2 Empty\_2

HK HongKong 3 Empty\_3

EG Egypt 4 Empty\_4

IL Israel 4 Empty\_4

KW Kuwait 4 Empty\_4

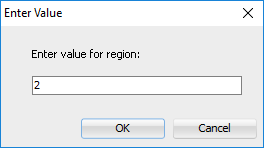
NG Nigeria 4 Empty\_4

ZM Zambia 4 Empty\_4

ZW Zimbabwe 4 Empty\_4

11 rows selected.

Rollback complete.



Index Table Key: 1 has a value of Argentina

Index Table Key: 6 has a value of Belgium

Index Table Key: 11 has a value of Denmark

Index Table Key: 16 has a value of Egypt

Index Table Key: 21 has a value of France

Index Table Key: 26 has a value of HongKong

Index Table Key: 31 has a value of Israel

Index Table Key: 36 has a value of Kuwait

Index Table Key: 41 has a value of Nigeria

Index Table Key: 46 has a value of Zambia

Index Table Key: 51 has a value of Zimbabwe

======================================================================

Total number of elements in the Index Table or Number of countries with NO cities listed is: 11

Second element (Country) in the Index Table is: Belgium

Before the last element (Country) in the Index Table is: Zambia

======================================================================

In the region 2 there is ONE country Argentina with NO city.

======================================================================

Total Number of countries with NO cities listed in the Region 2 is: 1

PL/SQL procedure successfully completed.

CO COUNTRY\_NAME REGION\_ID FLAG

-- ---------------------------------------- ---------- -------

BE Belgium 1 Empty\_1

DK Denmark 1 Empty\_1

FR France 1 Empty\_1

AR Argentina 2 Empty\_2

HK HongKong 3 Empty\_3

EG Egypt 4 Empty\_4

IL Israel 4 Empty\_4

KW Kuwait 4 Empty\_4

NG Nigeria 4 Empty\_4

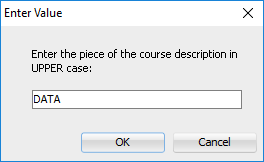
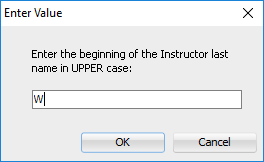
ZM Zambia 4 Empty\_4

ZW Zimbabwe 4 Empty\_4

11 rows selected.

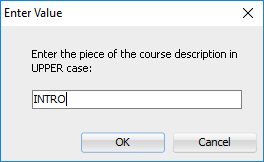
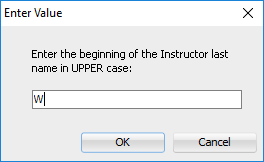
Rollback complete.

# Q4-ouput

There is NO data for this input match between the course description piece and the surname start of Instructor. Try again!

PL/SQL procedure successfully completed.

Course No: 240 Intro to the Basic Language with Section Id: 102 is taught by Wojick in the Course Section: 2

This Section Id has an enrollment of: 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Course No: 120 Intro to Java Programming with Section Id: 149 is taught by Wojick in the Course Section: 4

This Section Id has an enrollment of: 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

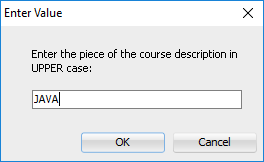
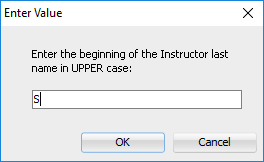
Course No: 25 Intro to Programming with Section Id: 88 is taught by Wojick in the Course Section: 4

This Section Id has an enrollment of: 5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This input match has a total enrollment of: 7 students.

PL/SQL procedure successfully completed.

Course No: 124 Advanced Java Programming with Section Id: 127 is taught by Schorin in the Course Section: 2

This Section Id has an enrollment of: 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Course No: 122 Intermediate Java Programming with Section Id: 153 is taught by Smythe in the Course Section: 2

This Section Id has an enrollment of: 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Course No: 120 Intro to Java Programming with Section Id: 150 is taught by Schorin in the Course Section: 5

This Section Id has an enrollment of: 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This input match has a total enrollment of: 7 students.

PL/SQL procedure successfully completed.