Assignment 1 PL/SQL Group 9

2019-06-05

Paul:

Zoey:

Graham:

Haanee:

1. ACCEPT country\_code PROMPT 'Please input two letter country code: ';

DECLARE

v\_country\_code varchar2(2):=UPPER('&country\_code');

v\_province varchar2(25);

v\_country\_id\_count number(5);

v\_city number(5);

v\_location locations%ROWTYPE;

BEGIN

SELECT count(country\_id)

INTO v\_country\_id\_count

FROM countries

WHERE UPPER(country\_id)=UPPER('&country\_code');

IF v\_country\_id\_count=0 THEN

RAISE NO\_DATA\_FOUND;

END IF;

SELECT count(city)

INTO v\_city

FROM locations

WHERE state\_province IS null

AND UPPER(country\_id)=v\_country\_code;

IF v\_city > 1 THEN

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

ELSIF v\_city = 0 THEN

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

ELSE

UPDATE locations

SET state\_Province = CASE

WHEN substr(City,1,1) IN ('A', 'B', 'E', 'F' ) THEN rpad('\*',Length(street\_address),'\*')

WHEN substr(City,1,1) IN ('C', 'D', 'G', 'H' ) THEN rpad('&',Length(street\_address),'&')

ELSE rpad('#',Length(street\_address),'#')

END

WHERE state\_province is null

AND country\_id=v\_country\_code;

select \*

INTO v\_location

from locations

where UPPER(country\_id)= v\_country\_code

and substr(state\_Province,1,1) in ('\*','&','#');

DBMS\_OUTPUT.PUT\_LINE('City '||v\_location.city||' has modified its province to '||v\_location.STATE\_PROVINCE);

Rollback;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

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

END;

/

Output:

**Please input two letter country code: JP**

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

**PL/SQL procedure successfully completed.**

**SQL> @as1**

**Please input two letter country code: UK**

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

**PL/SQL procedure successfully completed.**

**SQL> @as1**

**Please input two letter country code: IT**

**This country has MORE THAN ONE City without province listed.**

**PL/SQL procedure successfully completed.**

**SQL> @as1**

**Please input two letter country code: RS**

**This country has NO cities listed**

**PL/SQL procedure successfully completed.**

1. SET SERVEROUTPUT ON

SET VERIFY OFF

SET FEEDBACK OFF

/\*

ALTER TABLE countries

SET UNUSED (flag);

ALTER TABLE countries

DROP UNUSED COLUMNS;

ALTER TABLE countries

ADD flag CHAR(7);

\*/

-- get user selected region

ACCEPT user\_region\_id PROMPT 'Enter a Region ID: ';

DECLARE

v\_userInput VARCHAR2 (2); -- get user input

v\_count NUMBER; -- used to hold counted values

v\_regionID countries.region\_id%TYPE; -- will be user entered region

v\_noCityCount NUMBER; -- count coutries with no city

v\_countryName countries.country\_name%TYPE; -- to hold name of countries

BEGIN

v\_userInput := '&user\_region\_id'; --put the user entered region into var

v\_regionID := v\_userInput; -- throws VALUE\_ERROR if not a number (should anyways)

-- check if region exists

SELECT COUNT(\*)

INTO v\_count

FROM regions

WHERE region\_id = v\_regionID;

IF v\_count = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Region ' || v\_regionID || ' does not exist.');

ELSE

-- count countries with no city for entered region\_id

SELECT COUNT(\*)

INTO v\_noCityCount

FROM countries

WHERE country\_id NOT IN(

SELECT country\_id

FROM locations)

AND

region\_id = &&user\_region\_id;

-- display the count of countries without a city

CASE v\_noCityCount

WHEN 0 THEN

DBMS\_OUTPUT.PUT\_LINE('All countries have at least 1 city');

WHEN 1 THEN

-- get the country name

SELECT country\_name

INTO v\_countryName

FROM countries

WHERE country\_id NOT IN(

SELECT country\_id

FROM locations)

AND

region\_id = v\_regionID;

-- Update the flag column for countries with no city

UPDATE countries

SET flag = 'Empty\_' || region\_id

WHERE country\_id NOT IN (

SELECT country\_id

FROM locations);

-- count the modified cities

SELECT COUNT(\*)

INTO v\_count

FROM countries

WHERE flag LIKE 'Empty\_%';

DBMS\_OUTPUT.PUT\_LINE('This region ' || v\_regionID || ' there is ONE country ' || v\_countryName || ' with NO city.');

DBMS\_OUTPUT.PUT\_LINE('Number of countries with NO city is: ' || v\_count);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Region ID ' || v\_regionID || ' has MORE THAN ONE country without cities listed');

END CASE;

END IF;

EXCEPTION

WHEN VALUE\_ERROR THEN

DBMS\_OUTPUT.PUT\_LINE('Please enter a number. Try again.');

END;

/

SET FEEDBACK ON

select \* from countries where flag like 'Empty\_%' order by flag;

ROLLBACK;

Output:

**Enter a Region ID: 5**

**Region 5 does not exist.**

**Enter a Region ID: 1**

**Region ID 1 has MORE THAN ONE country without cities listed**

**Enter a Region ID: 2**

**This region 2 there is ONE country Argentina with NO city.**

**Number of countries with NO city is: 11**

**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**

**ZW Zimbabwe 4 Empty\_4**

**IL Israel 4 Empty\_4**

**KW Kuwait 4 Empty\_4**

**NG Nigeria 4 Empty\_4**

**ZM Zambia 4 Empty\_4**

**EG Egypt 4 Empty\_4**

**11 rows selected.**

**Rollback complete.**

1. ACCEPT user\_region\_id PROMPT 'Enter a Region ID: ';

DECLARE

v\_userInput VARCHAR2 (2); -- get user input

v\_count NUMBER; -- used to hold counted values

v\_regionID countries.region\_id%TYPE; -- will be user entered region

-- region cursor

CURSOR c\_region\_cursor IS

select country\_name

from countries

where region\_id = v\_regionID

AND

country\_id NOT IN (

SELECT country\_id

FROM locations)

ORDER BY 1;

-- for making index table

CURSOR c\_countries\_cursor IS

select country\_name

from countries

where country\_id NOT IN (

SELECT country\_id

FROM locations)

ORDER BY 1;

TYPE country\_name\_table\_type IS TABLE OF

countries.country\_name%TYPE

INDEX BY PLS\_INTEGER;

country\_name\_table country\_name\_table\_type;

v\_counter INTEGER (3) := 0;

v\_index INTEGER(3);

-- end of making index by table

BEGIN

v\_userInput := '&user\_region\_id'; --put the user entered region into var

v\_regionID := v\_userInput; -- throws VALUE\_ERROR if not a number (should anyways)

-- check if region exists

SELECT COUNT(\*)

INTO v\_count

FROM regions

WHERE region\_id = v\_regionID;

IF v\_count = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Region ' || v\_regionID || ' does not exist.');

ELSE

-- Update the flag column for countries with no city

UPDATE countries

SET flag = 'Empty\_' || region\_id

WHERE country\_id NOT IN (

SELECT country\_id

FROM locations);

-- create and display index by table values`

FOR int IN c\_countries\_cursor

LOOP

v\_index := v\_counter \* 5 + 1;

country\_name\_table(v\_index) := int.country\_name;

DBMS\_OUTPUT.PUT\_LINE('Index Table Key: ' || v\_index || ' has a value of ' || country\_name\_table(v\_index));

v\_counter := v\_counter + 1;

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: '|| v\_counter);

DBMS\_OUTPUT.PUT\_LINE('Second element (Country) in Index Table is:' || country\_name\_table(country\_name\_table.NEXT(country\_name\_table.first)));

DBMS\_OUTPUT.PUT\_LINE('Before the last element (Country) in the Index Table is: ' || country\_name\_table(country\_name\_table.PRIOR(country\_name\_table.last)));

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

-- display region cities with ni city

v\_counter := 0;

FOR int IN c\_region\_cursor

LOOP

v\_counter := v\_counter + 1;

DBMS\_OUTPUT.PUT\_LINE('In the region '|| v\_regionID || ' there is country ' || int.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\_regionID || ' is: ' || v\_counter);

END IF;

EXCEPTION

WHEN VALUE\_ERROR THEN

DBMS\_OUTPUT.PUT\_LINE('Please enter a number. Try again.');

END;

/

SET FEEDBACK ON

select \* from countries where flag like 'Empty\_%' order by flag, country\_name;

ROLLBACK;

Output:

**Enter a Region ID: 1**

**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 Index Table is:Belgium**

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

**======================================================================**

**In the region 1 there is country Belgium with NO city**

**In the region 1 there is country Denmark with NO city**

**In the region 1 there is country France with NO city**

**======================================================================**

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

**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.**

**Enter a Region ID: 2**

**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 Index Table is:Belgium**

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

**======================================================================**

**In the region 2 there is country Argentina with NO city**

**======================================================================**

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

**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.**

1. ACCEPT descPiece PROMPT 'Enter the piece of the course description in UPPER case: '

ACCEPT instrChar PROMPT 'Enter the beginning of Instructor last name in UPPER CASE: '

DECLARE

CURSOR c\_course\_cursor IS

SELECT c.DESCRIPTION, s.SECTION\_ID, c.COURSE\_NO, i.LAST\_NAME, s.SECTION\_NO

FROM COURSE c

INNER JOIN SECTION s ON c.COURSE\_NO = s.COURSE\_NO

INNER JOIN INSTRUCTOR i ON i.INSTRUCTOR\_ID = s.INSTRUCTOR\_ID

WHERE UPPER(c.DESCRIPTION) LIKE '%' || '&descPiece' || '%' AND i.LAST\_NAME LIKE '&instrChar' || '%'

ORDER BY c.DESCRIPTION;

CURSOR c\_enroll\_cursor (sec\_id SECTION.SECTION\_ID%TYPE) IS

SELECT COUNT(\*) AS enroll\_count

FROM ENROLLMENT

WHERE SECTION\_ID = sec\_id;

v\_enrollCount NUMBER(4, 0) := 0;

v\_total NUMBER(8, 0) := 0;

v\_matches NUMBER (3, 0) := 0;

BEGIN

FOR courseInd IN c\_course\_cursor LOOP

v\_matches := v\_matches + 1;

DBMS\_OUTPUT.PUT\_LINE('Course No: ' || courseInd.course\_no || ' ' ||

courseInd.description || ' with Section Id: ' || courseInd.section\_id ||

' is taught by ' || courseInd.last\_name || ' in the Course Section: ' ||

courseInd.section\_no || CHR(10));

FOR enrollInd IN c\_enroll\_cursor(courseInd.section\_id) LOOP

v\_enrollCount := enrollInd.enroll\_count;

v\_total := v\_total + v\_enrollCount;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('This Section Id has an enrollment of: ' || v\_enrollCount || CHR(10));

DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*' || CHR(10));

END LOOP;

IF v\_matches > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('The input match has a total enrollment of: ' || v\_total || ' students.');

ELSE

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!');

END IF;

END;

/

Output:

**Enter the piece of the course description in UPPER case: DATA**

**Enter the beginning of Instructor last name in UPPER CASE: W**

**There is NO data for this input match between the course description piece and**

**the surname start of Instructor. Try again!**

**Enter the piece of the course description in UPPER case: INTRO**

**Enter the beginning of Instructor last name in UPPER CASE: W**

**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**

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

**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**

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

**The input match has a total enrollment of: 7 students.**

**Enter the piece of the course description in UPPER case: JAVA**

**Enter the beginning of Instructor last name in UPPER CASE: S**

**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**

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

**The input match has a total enrollment of: 7 students.**