**1.**

SET SERVEROUTPUT ON;

SET VERIFY OFF;

ACCEPT V\_COUNTRYID PROMPT 'Enter value for country';

DECLARE

V\_CID VARCHAR2(2):= UPPER('&v\_countryId');

V\_LENGTH NUMBER;

V\_STRING VARCHAR(45);

V\_LOCATIONID NUMBER(10);

V\_STADD VARCHAR2(25);

V\_CITY VARCHAR2 (25);

BEGIN

  SELECT STREET\_ADDRESS, CITY, LOCATION\_ID INTO V\_STADD, V\_CITY, V\_LOCATIONID

    FROM LOCATIONS

    WHERE COUNTRY\_ID = V\_CID AND STATE\_PROVINCE IS NULL

    GROUP BY STREET\_ADDRESS, CITY, LOCATION\_ID;

    V\_LENGTH := LENGTH(V\_STADD);

    IF SUBSTR(V\_CITY, 1, 1) IN ('A','B','E','F') THEN

     UPDATE LOCATIONS

     SET STATE\_PROVINCE = RPAD('\*', v\_length, '\*')

     WHERE LOCATION\_ID = V\_LOCATIONID;

    ELSIF SUBSTR(V\_CITY, 1, 1) IN ('C','D','G','H') THEN

     UPDATE LOCATIONS

     SET STATE\_PROVINCE = RPAD('&', v\_length, '&')

     WHERE LOCATION\_ID = V\_LOCATIONID;

    ELSE

     UPDATE LOCATIONS

     SET STATE\_PROVINCE = RPAD('#', v\_length, '#')

     WHERE LOCATION\_ID = V\_LOCATIONID;

    END IF;

    SELECT STATE\_PROVINCE INTO V\_STRING FROM LOCATIONS WHERE LOCATION\_ID = V\_LOCATIONID;

    DBMS\_OUTPUT.PUT\_LINE('City '|| V\_CITY || ' has modified its province to ' || V\_STRING);

    ROLLBACK;

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.');

END;

/

**2.**

ALTER TABLE COUNTRIES

ADD FLAG CHAR(7);

ACCEPT reg\_id PROMPT 'Enter value for region: '

DECLARE

    V\_CNAME COUNTRIES.COUNTRY\_NAME%TYPE;

    V\_REGI\_ID REGIONS.REGION\_ID%TYPE := &reg\_id;

    V\_COUNT NUMBER(6);

BEGIN

    SELECT COUNTRY\_NAME

    INTO V\_CNAME

    FROM COUNTRIES

    WHERE REGION\_ID = V\_REGI\_ID

    AND COUNTRY\_ID NOT IN (SELECT COUNTRY\_ID From LOCATIONS );

    SELECT COUNT(COUNTRY\_ID)

    INTO V\_COUNT

    FROM COUNTRIES

    WHERE COUNTRY\_ID NOT IN (SELECT COUNTRY\_ID From LOCATIONS);

    UPDATE COUNTRIES

    SET FLAG = ('EMPTY\_' ||REGION\_ID)

    WHERE COUNTRY\_ID NOT IN (SELECT COUNTRY\_ID From LOCATIONS);

    DBMS\_OUTPUT.PUT\_LINE('In the region ' || V\_REGI\_ID || ' there is ONE country '|| V\_CNAME || ' with NO city');

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

    ROLLBACK;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

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

WHEN TOO\_MANY\_ROWS THEN

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

END;

**3.**

ALTER TABLE COUNTRIES

ADD FLAG VARCHAR(7);

CREATE SEQUENCE COUNTRY\_FLAG\_INDEX

START WITH 1

INCREMENT BY 5;

ACCEPT reg\_id PROMPT 'Enter value for region: '

DECLARE

     TYPE   c\_tab\_type IS TABLE OF COUNTRIES.COUNTRY\_ID%TYPE

                      INDEX   BY  BINARY\_INTEGER;

     TYPE   c\_countryid\_type IS TABLE OF COUNTRIES.COUNTRY\_NAME%TYPE

                      INDEX   BY  BINARY\_INTEGER;

     TYPE   c\_countryrow\_type IS TABLE OF COUNTRIES%ROWTYPE

                      INDEX   BY  BINARY\_INTEGER;

    TYPE NO\_CITY\_TABLE IS TABLE OF COUNTRIES.COUNTRY\_NAME%TYPE INDEX BY BINARY\_INTEGER;

    NCITY\_TABLE NO\_CITY\_TABLE;

    regi\_id REGIONS.REGION\_ID%TYPE := &reg\_id;

    c\_tab c\_tab\_type;

    c\_countryid c\_countryid\_type;

    c\_countryrow c\_countryrow\_type;

    V\_SECOND COUNTRIES.COUNTRY\_NAME%TYPE;

    V\_SECOND\_LAST COUNTRIES.COUNTRY\_NAME%TYPE;

    V\_ORIGINAL\_MOD INTEGER := 0;

CURSOR  c1  IS

      SELECT  COUNTRY\_ID,country\_name

      FROM countries

      where region\_id = regi\_id

      AND country\_id NOT IN (SELECT COUNTRY\_ID

                           FROM LOCATIONS)

                           ORDER BY country\_name ASC;

CURSOR  c2 IS

      SELECT  COUNTRY\_ID ,

               COUNTRY\_NAME ,

               REGION\_ID ,

               FLAG

      FROM countries

      where country\_id NOT IN (SELECT COUNTRY\_ID

                           FROM LOCATIONS)

      ORDER BY country\_name ASC;

    total INTEGER := 0;

    counter INTEGER := 0;

    SEQCOUNTER INTEGER := 0;

    reg\_id\_check REGIONS.REGION\_ID%TYPE := 0;

BEGIN

    --CHECKS IF REGION ID EXISTS

    SELECT REGION\_ID INTO reg\_id\_check FROM REGIONS WHERE REGION\_ID = regi\_id;

    For i IN c1 LOOP

       counter := counter +1;

       c\_tab(counter):= i.country\_id;

       c\_countryid(counter) := i.country\_name;

       UPDATE COUNTRIES

       SET FLAG = ('EMPTY\_' || regi\_id)

       WHERE c\_tab(counter) = country\_id;

    END LOOP;

    V\_ORIGINAL\_MOD := counter;

    counter := 0;

    SELECT  count(COUNTRY\_ID) into total

      FROM countries

      where country\_id NOT IN (SELECT COUNTRY\_ID

                           FROM LOCATIONS);

    For i IN c2 LOOP

       counter := counter + 1;

       IF(counter = 2)THEN

           V\_SECOND := I.COUNTRY\_NAME;

       END IF;

       IF(counter = total - 1)THEN

           V\_SECOND\_LAST := I.COUNTRY\_NAME;

       END IF;

       SEQCOUNTER := COUNTRY\_FLAG\_INDEX.NEXTVAL;

       NCITY\_TABLE(SEQCOUNTER):= i.COUNTRY\_NAME;

       DBMS\_OUTPUT.PUT\_LINE('Index Table Key: ' || SEQCOUNTER ||' has a value of ' || NCITY\_TABLE(SEQCOUNTER));

    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: ' || total);

    DBMS\_OUTPUT.PUT\_LINE('Second element (Country) in the Index Table is: ' || V\_SECOND);

    DBMS\_OUTPUT.PUT\_LINE('Before the last element (Country) in the Index Table is: ' || V\_SECOND\_LAST);

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

    FOR i IN 1..V\_ORIGINAL\_MOD LOOP

       DBMS\_OUTPUT.PUT\_LINE('In the region ' || regi\_id || ' there is country ' || c\_countryid(i) || ' with NO city.');

    END LOOP;

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

    DBMS\_OUTPUT.PUT\_LINE('Total Number of countries with NO cities listed in the Region ' || regi\_id || ' is: ' || V\_ORIGINAL\_MOD);

ROLLBACK;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

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

END;

**4.**

SET SERVEROUTPUT ON;

SET VERIFY OFF;

ACCEPT V\_COURSE PROMPT "Enter the piece of the course description in UPPER case:";

ACCEPT V\_INSTRUCTOR PROMPT "Enter the beginning of Instructor last name in UPPER case:";

DECLARE

 V\_C\_SEARCH VARCHAR2(10):= '&V\_COURSE';

 V\_I\_SEARCH VARCHAR2(2):= '&V\_INSTRUCTOR';

 V\_COUNT NUMBER(6);

 V\_ENTRY NUMBER := 0;

 V\_E\_COUNTER NUMBER := 0;

 CURSOR C1 IS

 SELECT CO.COURSE\_NO, CO.DESCRIPTION, S.SECTION\_ID, S.SECTION\_NO, I.LAST\_NAME

 FROM COURSE CO, SECTION S ,INSTRUCTOR I

 WHERE UPPER(CO.DESCRIPTION) LIKE '%'|| V\_C\_SEARCH ||'%'

 AND UPPER(I.LAST\_NAME) LIKE V\_I\_SEARCH ||'%'

 AND CO.COURSE\_NO = S.COURSE\_NO

 AND I.INSTRUCTOR\_ID = S.INSTRUCTOR\_ID;

 V\_C1  C1%ROWTYPE;

BEGIN

OPEN C1;

FETCH C1 INTO V\_C1;

 WHILE C1%FOUND

    LOOP

    EXIT  WHEN  C1%NOTFOUND;

    V\_ENTRY := 1;

    DBMS\_OUTPUT.PUT\_LINE('Course No: ' || V\_C1.COURSE\_NO || ' ' || V\_C1.DESCRIPTION || ' with Section Id: ' || V\_C1.SECTION\_ID || ' is taught by ' || V\_C1.LAST\_NAME || ' in the Course Section: ' || V\_C1.SECTION\_NO);

    SELECT COUNT(SECTION\_ID) INTO V\_COUNT FROM ENROLLMENT WHERE SECTION\_ID = V\_C1.SECTION\_ID;

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

    V\_E\_COUNTER := V\_COUNT + V\_E\_COUNTER;

    FETCH C1 INTO V\_C1;

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

  END LOOP;

  CLOSE C1;

  CASE

  WHEN V\_ENTRY = 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: ' || V\_E\_COUNTER || ' students.');

  END CASE;

END;