PL/SQL part 1 of 3 Notes adapted from Nebojsa October 2020

How to build a Block without an Exception Handler and then how to transform it to a Stored Procedure with one IN parameter

Problem: For a given CITY name, you need to find out Department\_id and Department\_name that exists in that city. There are 3 scenarios here;

1. In a given CITY, there is a SINGLE department
2. In a given CITY, there is a MORE THAN ONE department
3. In a given CITY, there is NO department

**PART 1**

1. Our first Code Example is a Block without Exception Handler. It will work properly only if in the given City is ONLY ONE department

Here is the BLOCK

**DECLARE**

**v\_city locations.city%TYPE := 'SOUTHLAKE';**

**v\_dept# departments.department\_id%TYPE ;**

**v\_dname departments.department\_name%TYPE;**

**v\_loc# departments.location\_id %TYPE;**

**BEGIN**

**SELECT location\_id INTO v\_loc#**

**FROM locations**

**WHERE UPPER(city) = v\_city;**

**SELECT department\_id, department\_name**

**INTO v\_dept#, v\_dname**

**FROM departments**

**WHERE location\_id = v\_loc# ;**

**DBMS\_OUTPUT.PUT\_LINE('Department ID for chosen city is ' || v\_dept# );**

**DBMS\_OUTPUT.PUT\_LINE('and your department name is ' || v\_dname);**

**END;**

**Test 1: Using TORONTO This city location has only one department**

**OUTPUT is**

**Department ID for chosen city is 20**

**and your department name is Marketing**

Test 2: Using SEATTLE which has several departments located there

OUTPUT:

**ORA-01422: exact fetch returns more than requested number of rows**

ORA-06512: at line 11

Test 3: Using VENICE

**OUTPUT:**

**ORA-01403: no data found**

ORA-06512: at line 11

Because test 2 and test 3 generated errors, you need to be able to handle those errors and produce a message for the user. At this stage, the messages are simple text. Not now, but as you go deeper into learning the error messages will be more complicated. For example, if you cannot find the student you entered you do not as a user want a message then close the program. You probably want a message and re prompt the user for a name or an alternate search method for the student.

**PART 2**

Our second Code Example is a Block with Exception Handler that deals with BOTH exceptions, so you will not get Error messages

DECLARE

v\_city location.city%TYPE := 'SEATTLE';

v\_dept# department.department\_id%TYPE ;

v\_dname department.department\_name%TYPE;

v\_loc# department.location\_id %TYPE;

BEGIN

SELECT location\_id INTO v\_loc#

FROM location

WHERE UPPER(city) = v\_city;

SELECT department\_id, department\_name

INTO v\_dept#, v\_dname

FROM department

WHERE location\_id = v\_loc# ;

DBMS\_OUTPUT.PUT\_LINE('In the chosen city your department id is ' || v\_dept# || ' and your department name is ' || v\_dname);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('In the chosen city ' || v\_city || ' there is NO department.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('In the chosen city ' || v\_city || ' there is MORE THAN ONE department.');

END;

Test 4: Using SEATTLE again

**In the chosen city SEATTLE there is MORE THAN ONE department.**

PL/SQL procedure successfully completed.

Test 5: VENICE

**In the chosen city VENICE there is NO department.**

PL/SQL procedure successfully completed.

**PART 3**

Our third Code Example is a Stored Procedure that accepts one IN parameter, CITY name. Watch how the code has changed, when using p\_city and not v\_city

**CREATE OR REPLACE PROCEDURE find\_dept (p\_city locations.city%TYPE)**

IS

v\_dept# departments.department\_id%TYPE ;

v\_dname departments.department\_name%TYPE;

v\_loc# departments.location\_id %TYPE;

BEGIN

SELECT location\_id INTO v\_loc#

FROM locations

WHERE UPPER(city) = p\_city;

SELECT department\_id, department\_name

INTO v\_dept#, v\_dname

FROM departments

WHERE location\_id = v\_loc# ;

DBMS\_OUTPUT.PUT\_LINE('Department id is ' || v\_dept#);

DBMS\_OUTPUT.PUT\_LINE('Name is ' || v\_dname);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('City ' || p\_city || ' there is NO department.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('City ' || p\_city || ' there is MORE THAN ONE department.');

END;

To run the stored Procedure

**EXECUTE find\_dept('OXFORD');**

**OUTPUT:**

Department id is 80

Name is Sales

**EXECUTE find\_dept('SEATTLE');**

**City SEATTLE there is MORE THAN ONE department.**

PL/SQL procedure successfully completed.

SQL> **EXECUTE find\_dept('VENICE');**

**City VENICE there is NO department.**

PL/SQL procedure successfully completed.