

Database Programming with PL/SQL

7-1: Handling Exceptions

Practice Activities

**Vocabulary**

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| An exception handler | Code that defines the recovery actions to be performed when execution-time errors occur. |
| Exception | Occurs when an error is discovered during the execution of a program that disrupts the normal operation of the program. |

**Try It / Solve It**

1. What happens when Oracle encounters a runtime problem while executing a PL/SQL block?

Oracle va afisa un mesaj corespunzator erorii, iar blocul de executie nu se va mai executa.

1. What do you need to add to your PL/SQL block to address these problems?

Putem adauga o exceptie care sa ne indrume inspre partea de cod de unde a survenit eroarea.



1. List three advantages of handling exceptions within a PL/SQL block.

Putem preveni oprirea subita a programului cauzata de o eroare, putem afisa un mesaj corespunzator unei exceptii acolo unde ne asteptam sa apara, putem depana mai usor programul.

1. Run this PL/SQL code and then answer the questions that follow.

DECLARE

v\_jobid employees.job\_id%TYPE;

BEGIN

SELECT job\_id INTO v\_jobid

FROM employees

WHERE department\_id = 80; END;

* 1. What happens when you run the block? In your own words, explain what you can do to fix this problem.

Blocul returneaza mai mult decat numar de linii asteptate. Putem rezolva aceasta problema cu exceptie TOO\_MANY\_ROWS .

* 1. Modify the code to fix the problem. Use a TOO\_MANY\_ROWS exception handler.

DECLARE

v\_jobid employees.job\_id%TYPE;

BEGIN

SELECT job\_id INTO v\_jobid

FROM employees

WHERE department\_id = 80;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Your select statement retived multiple rows. Consier using a cursor.');

END;

* 1. Run your modified code. What happens this time?

Se afiseaza mesajul corespunzator exceptiei.

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

1. Run the following PL/SQL block, which tries to insert a new row (with department\_id = 50) into the departments table. What happens and why?

BEGIN

INSERT INTO departments (department\_id, department\_name, manager\_id, location\_id)

VALUES (50, 'A new department', 100, 1500);

DBMS\_OUTPUT.PUT\_LINE('The new department was inserted');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('An exception has occurred.'); END;

In momentul inserarii are loc violarea cheii primare ceea ce duce la aparitia exceptiei si afisarea mesajului corespunzator in consola.

1. **Enter the following code to create a copy of the employees table for this and the next question.**

CREATE TABLE emp\_temp AS SELECT \* FROM employees;

**In the new emp\_temp table, delete all but one of the employees in department 10.**

SELECT \* FROM emp\_temp WHERE department\_id = 10;

DELETE FROM emp\_temp WHERE employee\_id = …; (repeat as necessary)

**Employee\_id sunt 200, 225,226, 227.**

Enter the following PL/SQL block, which tries to SELECT all the employees in a specific department. Run it three times, using department\_ids 10, 20, and 30. What happens and why?

DECLARE

v\_employee\_id emp\_temp.employee\_id%TYPE; v\_last\_name emp\_temp.last\_name%TYPE;

BEGIN

SELECT employee\_id, last\_name INTO v\_employee\_id, v\_last\_name FROM emp\_temp

WHERE department\_id = 10; -- run with values 10, 20, and 30

DBMS\_OUTPUT.PUT\_LINE('The SELECT was successful');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An exception has occurred'); END;

Pentru cazul 10 si 30 nu avem date, iar pentru cazul 20 se returneaza mai mult decat o linie, ca urmare apare exceptia mentionata in program.

1. Modify your code from question 6 to add two more exception handlers to trap the possible exceptions individually. Use NO\_DATA\_FOUND and TOO\_MANY\_ROWS. Re-run the block three times, using 10, 20, and 30 as before. Observe the message displayed in each case.

DECLARE

v\_employee\_id emp\_temp.employee\_id%TYPE;

v\_last\_name emp\_temp.last\_name%TYPE;

BEGIN

SELECT employee\_id, last\_name INTO v\_employee\_id, v\_last\_name FROM emp\_temp

WHERE department\_id = 10; -- run with values 10, 20, and 30

DBMS\_OUTPUT.PUT\_LINE('The SELECT was successful');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No data found');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('TOO many rows');

END;

In cazul 10 si 30 apare No data found, iar in cazul 20 apare Too many rows .

**When finished, remember to delete the emp\_temp table.**

DROP TABLE emp\_temp;

1. List three guidelines for trapping exceptions.

1.Mereu sa adaugam o exceptie unde exista posibilitatea aparitiei unei erori.

2.Este de preferat sa numim exceptiile in loc sa utilizam doar OTHERS.

3.Sa testam codul cu diferite combinatii de bad data pt a vedea posibilele erori.

2

1. Enter and run the following PL/SQL block. Explain the output. Note: the WHEN OTHERS handler successfully handles any type of exception which occurs.

DECLARE

v\_number NUMBER(2); BEGIN

v\_number := 9999;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An exception has occurred');

END;

Din cauza unei erori are loc afisarea mesajului din exceptie.

1. Modify the block in question 9 to omit the exception handler, then re-run the block. Explain the output.

DECLARE

v\_number NUMBER(2); BEGIN

v\_number := 9999;

END;

🡪la executarea acestui cod apare o eroare pt ca a doua atribuire are o valoare mult mai mare decat cea declara pt variabila respectiva: number precision too large .

1. Enter and run the following code and explain the output.

DECLARE

v\_number NUMBER(4); BEGIN

v\_number := 1234; DECLARE

v\_number NUMBER(4); BEGIN

v\_number := 5678;

v\_number := 'A character string';

END;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An exception has occurred');

DBMS\_OUTPUT.PUT\_LINE('The number is: ' || v\_number); END;

In acest caz se va intra pe ramura de exceptie si se vor afisa mesajele din exceptie. Pentru v\_number se va afisa prima valoarea atribuita lui, adica 1234, mai apoi avand loc eroare de conversii, iar ca rezultat al acestei exceptii se va declansa exceptia.