Handling Exceptions

Terminology

Directions: Identify the vocabulary word for each definition below:

1. _____ Code that defines the recovery actions to be performed when execution-time errors occur.

2. _____ Occurs when an error occurs 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?

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

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

4. 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;
```

   A. What happens when you run the block?

   B. In your own words, explain what can you do to fix this problem.

   C. Modify the code to fix the problem. Use a TOO_MANY_ROWS exception handler.

   D. Run your modified code. What happens this time?

5. 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;
```

6. 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   employees.employee_id%TYPE;
  v_last_name     employees.last_name%TYPE;
BEGIN
  SELECT employee_id, last_name INTO v_employee_id, v_last_name
    FROM employees
    WHERE department_id = <enter a value>;
  DBMS_OUTPUT.PUT_LINE('The SELECT was successful');
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An exception has occurred');
END;
```

7. 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.