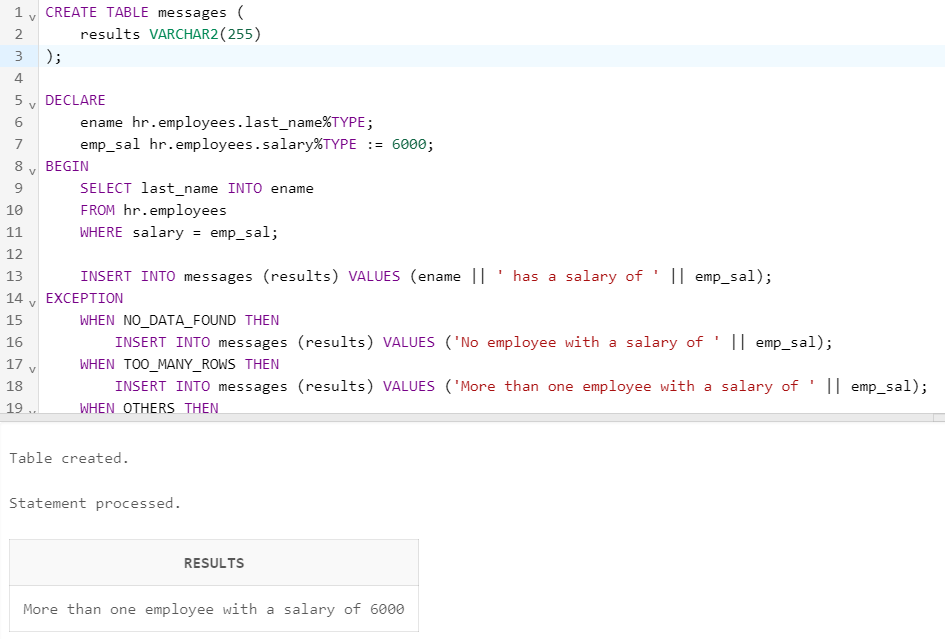
**Practice 5**

1. The purpose of this example is to show the usage of predefined exceptions. Write a PL/SQL block to select the name of the employee with a given salary value.
   * + 1. Create a table messages with one column **results** of the VARCHAR(50) type. If this table already exists, just delete all the records in this table.
       2. In the declarative section declare two variables: ename of type employees.last\_name and emp\_sal of type employees.salary. Give the value to emp\_sal.
       3. In the executable section, retrieve the last names of employees whose salaries are equal to the value in emp\_sal. **Note:** Do not use explicit cursors. If the salary entered returns only one row, insert into the messages table the employee’s name and the salary amount.
       4. If the salary entered does not return any rows, handle the exception with an appropriate exception handler and insert into the messages table the message “No employee with a salary of <*salary>*.”
       5. If the salary entered returns more than one row, handle the exception with an appropriate exception handler and insert into the messages table the message “More than one employee with a salary of <*salary*>.”

f. Handle any other exception with an appropriate exception handler and insert into the messages table the message “Some other error occurred.”

g. Display the rows from the messages table to check whether the PL/SQL block has executed successfully. Sample output is shown below.

D:\PL_SQL\NEW\Lessons\Graphics\Les08\sol1.gif



CREATE TABLE messages (

results VARCHAR2(255)

);

DECLARE

ename hr.employees.last\_name%TYPE;

emp\_sal hr.employees.salary%TYPE := 6000;

BEGIN

SELECT last\_name INTO ename

FROM hr.employees

WHERE salary = emp\_sal;

INSERT INTO messages (results) VALUES (ename || ' has a salary of ' || emp\_sal);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

INSERT INTO messages (results) VALUES ('No employee with a salary of ' || emp\_sal);

WHEN TOO\_MANY\_ROWS THEN

INSERT INTO messages (results) VALUES ('More than one employee with a salary of ' || emp\_sal);

WHEN OTHERS THEN

INSERT INTO messages (results) VALUES ('Some other error occured.');

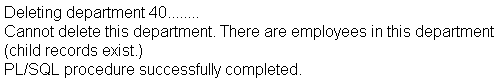
END;

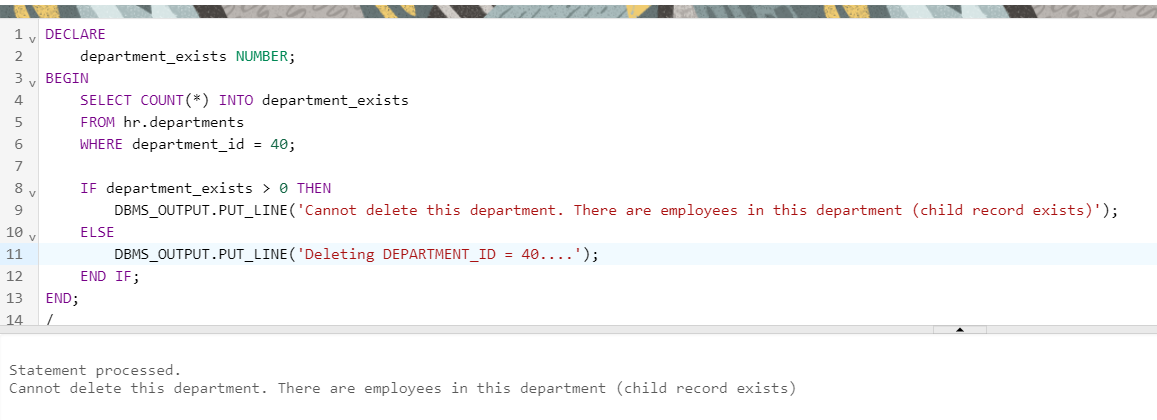
/

SELECT \* FROM messages;

2. The purpose of this task is to show how to declare exceptions with a standard Oracle server error. Use the Oracle server error ORA-02292 (integrity constraint violated – child record found).

* + - 1. In the declarative section, declare an exception childrecord\_exists. Associate the declared exception with the standard Oracle server error –02292.
      2. In the executable section, display ‘Deleting department 40.....’. Include a DELETE statement to delete the department with department\_id 40.
      3. Include an exception section to handle the childrecord\_exists exception and display the appropriate message. Sample output is shown below.





DECLARE

childrecord\_exists EXCEPTION;

PRAGMA EXCEPTION\_INIT (childrecord\_exists, -02292);

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Deleteting DEPARTMENT\_ID = 40....');

DELETE FROM hr.departments WHERE department\_id = 40;

EXCEPTION

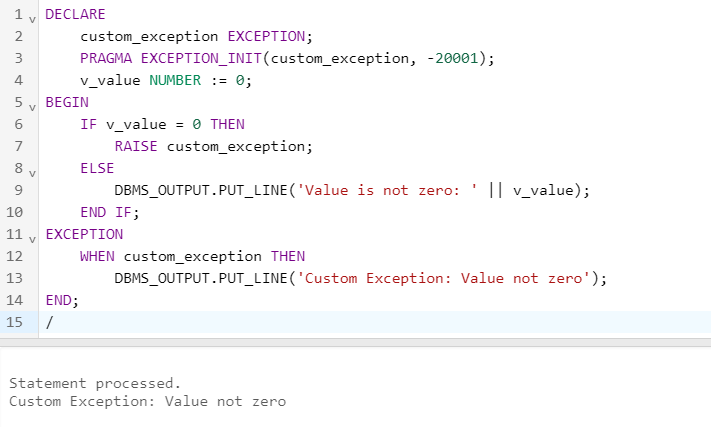
WHEN childrecord\_exists THEN

DBMS\_OUTPUT.PUT\_LINE('Cannot delete this department. There are employees in this department (childrecord exist)');

END;

/

1. Create any PL/SQL block that handles any user-defined exception.



DECLARE

custom\_exception EXCEPTION;

PRAGMA EXCEPTION\_INIT(custom\_exception, -20001);

v\_value NUMBER := 0;

BEGIN

IF v\_value = 0 THEN

RAISE custom\_exception;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Value is not zero: ' || v\_value);

END IF;

EXCEPTION

WHEN custom\_exception THEN

DBMS\_OUTPUT.PUT\_LINE('Custom Exception: Value cannot be zero');

END;

/