Database Programming with PL/SQL
7-4: Recognizing the Scope of Exceptions
Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

Propagation of exceptions	The inner block terminates unsuccessfully, and PL/SQL passes the exception to the outer block.
Exception Visibility	The portion of the program where the exception can be accessed without using a qualifier.
Exception Scope	The portion of a program in which the exception is declared and is accessible.

Try It / Solve It

1. Enter and run the following code twice, once for each of the two country_ids, 5 (which does not exist) and 672 (Antarctica, which does exist but has no currency).

```
DECLARE
  v_country_name          countries.country_name%TYPE;
  v_currency_code         countries.currency_code%TYPE;
BEGIN
  DECLARE
    e_no_currency         EXCEPTION;
  BEGIN
    SELECT country_name, currency_code INTO v_country_name, v_currency_code
      FROM countries
        WHERE country_id = 5;   -- repeat with 672
    IF v_currency_code = 'NONE' THEN
      RAISE e_no_currency;
    END IF;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('This country does not exist');
      WHEN e_no_currency THEN
        DBMS_OUTPUT.PUT_LINE('This country exists but has no currency');
    END;
  EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Another type of error occurred');
END;
```

This country does not exist

Statement processed.
no data found exception is raised before getting to raise e_no_currency

This country exists but has no currency

A. Explain the output. Save your code.

B. Modify the code to move the two exception handlers to the outer block. Leave the declaration of e_no_currency in the inner block. Execute twice, again using country_ids 5 and 672. Now what happens and why? Save your code.

C. Modify the code again to move the declaration of e_no_currency to the outer block. Execute the code again using country_ids 5 and 672. Now what happens and why?

B

```
DECLARE
 v_country_name wf_countries.country_name%TYPE;
 v_currency_code wf_countries.currency_code%TYPE;
BEGIN
 DECLARE
 e_no_currency EXCEPTION;
 BEGIN
 SELECT country_name, currency_code INTO
v_country_name, v_currency_code
 FROM wf_countries
 WHERE country_id = 672; -- repeat with 672
 IF v_currency_code = 'NONE' THEN
 RAISE e_no_currency;
 END IF;
 END;
 EXCEPTION
 WHEN NO_DATA_FOUND THEN
 DBMS_OUTPUT.PUT_LINE('This country does not exist');
 WHEN e_no_currency THEN
 DBMS_OUTPUT.PUT_LINE('This country exists but has
no currency');
 WHEN OTHERS THEN
 DBMS_OUTPUT.PUT_LINE('Another type of error
occurred');
END;
```

Like this, it will give an error because it is declared in the inner block,
so it will be out of scope in the outer block.
We need to also declare it where the other variables are.

C.

```
DECLARE
 v_country_name wf_countries.country_name%TYPE;
 v_currency_code wf_countries.currency_code%TYPE;
  e_no_currency EXCEPTION;
BEGIN
 BEGIN
 SELECT country_name, currency_code INTO v_country_name,
v_currency_code
 FROM wf_countries
 WHERE country_id = 672; -- repeat with 672
 IF v_currency_code = 'NONE' THEN
 RAISE e_no_currency;
 END IF;
 END;
 EXCEPTION
 WHEN NO_DATA_FOUND THEN
 DBMS_OUTPUT.PUT_LINE('This country does not exist');
 WHEN e_no_currency THEN
 DBMS_OUTPUT.PUT_LINE('This country exists but has no
currency');
 WHEN OTHERS THEN
 DBMS_OUTPUT.PUT_LINE('Another type of error occurred');
END;
```

It's working like it was on point A.

.