

# PL/SQL

( Oracle University - Oracle Database 10g: PL/SQL Fundamentals O'Reilly - Oracle PL/SQL Programming 5th edition .)

first name Michael last name . . . . . , last\_name SELECT lastname , first\_name='Michael' , . (compile-time) , (run-time) (Exceptions) . PL/SQL , , PL/SQL .

?

, . . . . .

```

PROCEDURE minhang
IS
    new_value VARCHAR2(20)
BEGIN
    new_value := old_val || '.co.kr';
    IF new_value LIKE 'sysdba%'
    THEN
        ...
    END IF;
EXCEPTION
    WHEN VALUE_ERROR
    THEN
        ...
END;
```

PL/SQL . " . . . . (Handler) , . . . .

EXCEPTION , CASE . WHEN exceptionN ( , ) . OR . statementN . WHEN exceptionN WHEN OTHERS .

PL/SQL . . . . . successfully . Your select statement retrieved multiple rows. Consider using a cursor. .

<b>Exception section</b>	PL/SQL . . . CASE
<b>=</b>	
<b>Raise</b>	PL/SQL PL/SQL . RAISE, RAISE_APPLICATION_ERROR .

=	
<b>Handle, handler</b>	(Trap) ". - , , .
= ,	
<b>Scope</b>	. . .
=	
<b>Propagation</b>	, . .
=	
<b>Unhandled exception</b>	, . . , / .
=	
<b>Un-named or anonymous exception</b>	, RAISE , WHEN .
= ,	
<b>Named exception</b>	, . EXCEPTION_INIT , .
=	

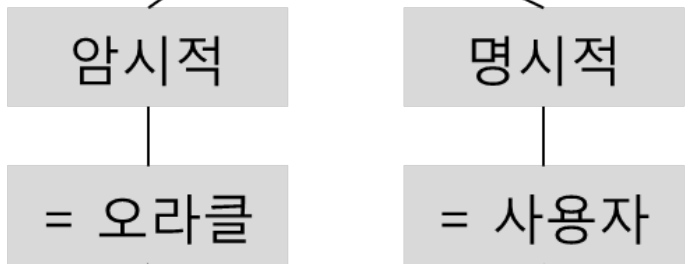
- : , , , DBMS . . . .
- :

- . (trap) : (handler) .
- . (propagation) : , PL/SQL . . . . , PL/SQL (, SQL\*Plus. SQL Developer) .

- :
- : , , .

- : , RAISE, RAISE\_APPLICATION\_ERROR .

# 예외 발생



오라클이 사전 정의 O  
오라클이 사전 정의 X

사용자 정의

이렇게 세가지 형태로 예외가 발생하고,  
각각의 형태에 따라 처리 방법이 달라짐

( ( ), , , , , )

PL/SQL , , . STANDARD PL/SQL Default . . .

```

SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  vempno NUMBER(4);
  3  vename VARCHAR(20);
  4  vsal NUMBER(10);
  5  BEGIN
  6  SELECT empno, ename, sal*12+NVL(comm, 0)
  7  INTO vempno, vename, vsal
  8  FROM emp
  9  WHERE empno=&vno;
 10  DBMS_OUTPUT.PUT_LINE('Employee# : ' || vempno );
 11  DBMS_OUTPUT.PUT_LINE('NAME : ' || vename);
 12  DBMS_OUTPUT.PUT_LINE('SALARY : ' || vsal);
 13  EXCEPTION
 14  WHEN NO_DATA_FOUND THEN
 15  DBMS_OUTPUT.PUT_LINE('Cannot find the employee');
 16  END;
 17  /
Enter VALUE FOR vno: 200
old  9: WHERE EMPNO=&VNO;
NEW  9: WHERE EMPNO=200;
Cannot find the employee

PL/SQL PROCEDURE successfully completed.

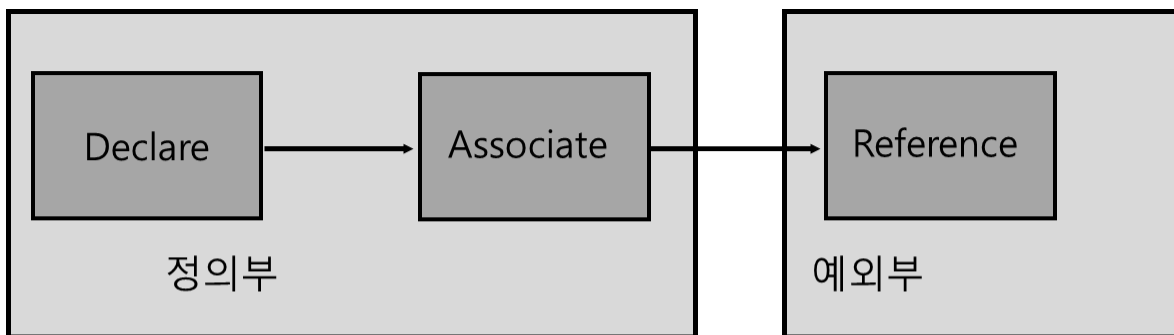
SQL>

```

*PL/SQL User's Guide and Reference*

ACCESS_INTO_NULL	ORA-06530	Attempted to assign values to the attributes of an uninitialized object
CASE_NOT_FOUND	ORA-06592	None of the choices in the WHEN clauses of a CASE statement are selected, and there is no ELSE clause.
COLLECTION_IS_NULL	ORA-06531	Attempted to apply collection methods other than EXISTS to an uninitialized nested table or VARRAY
CURSOR_ALREADY_OPEN	ORA-06511	Attempted to open an already-open cursor
DUP_VAL_ON_INDEX	ORA-00001	Attempted to insert a duplicate value
INVALID_CURSOR	ORA-01001	Illegal cursor operation occurred.
INVALID_NUMBER	ORA-01722	Conversion of character string to number fails.
LOGIN_DENIED	ORA-01017	Logging on to the Oracle server with an invalid username or password
NO_DATA_FOUND	ORA-01403	Single row SELECT returned no data.
NOT_LOGGED_ON	ORA-01012	PL/SQL program issues a database call without being connected to the Oracle server.
PROGRAM_ERROR	ORA-06501	PL/SQL has an internal problem.
ROWTYPE_MISMATCH	ORA-06504	Host cursor variable and PL/SQL cursor variable involved in an assignment have incompatible return types.
STORAGE_ERROR	ORA-06500	PL/SQL ran out of memory, or memory is corrupted.

SUBSCRIPT_BEYOND_COUNT	ORA-06533	Referenced a nested table or VARRAY element by using an index number larger than the number of elements in the collection
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532	Referenced a nested table or VARRAY element by using an index number that is outside the legal range (for example, -1)
SYS_INVALID_ROWID	ORA-01410	The conversion of a character string into a universal ROWID fails because the character string does not represent a valid ROWID.
TIMEOUT_ON_RESOURCE	ORA-00051	Time-out occurred while the Oracle server was waiting for a resource.
TOO_MANY_ROWS	ORA-01422	Single-row SELECT returned more than one row.
VALUE_ERROR	ORA-06502	Arithmetic, conversion, truncation, or size-constraint error occurred.
ZERO_DIVIDE	ORA-01476	Attempted to divide by zero



예외의 이름을 선언한다

PRAGMA EXCEPTION\_INIT문을 이용하여 오라클 에러코드와 예외 이름을 연결한다.

발생한 예외를 처리한다.

```

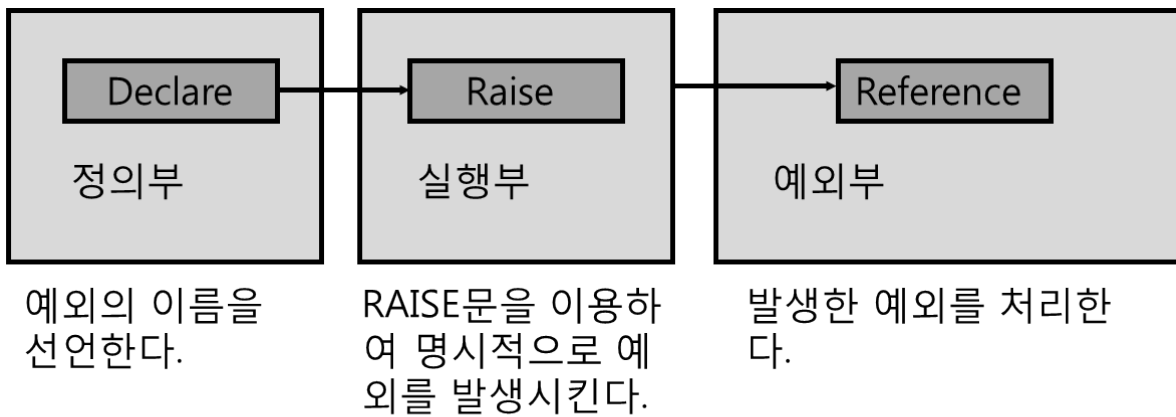
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  insert_excep EXCEPTION;
  3  PRAGMA EXCEPTION_INIT
  4  (insert_excep, -01400);
  5  BEGIN
  6  INSERT INTO departments
  7  (department_name) VALUES (NULL);
  8  EXCEPTION
  9  WHEN insert_excep THEN
 10  DBMS_OUTPUT.PUT_LINE('INSERT OPERATION FAILED');
 11  DBMS_OUTPUT.PUT_LINE(SQLERRM);
 12  END;
 13  /
INSERT OPERATION FAILED
ORA-01400: cannot INSERT NULL INTO ("HR"."DEPARTMENTS"."DEPARTMENT_ID")

PL/SQL PROCEDURE successfully completed.

SQL>

```

1. .()
2. PRAGMA EXCEPTION\_INIT .()
3. .()



```

SQL> ACCEPT deptno PROMPT 'Please enter the department number:'
Please enter the department NUMBER:9821
SQL> ACCEPT name PROMPT 'Please enter the department name:'
Please enter the department name:testing
SQL> DECLARE
  2   invalid_department EXCEPTION;
  3   name VARCHAR2(20):='&name';
  4   deptno NUMBER :=&deptno;
  5 BEGIN
  6   UPDATE departments
  7   SET    department_name = name
  8   WHERE department_id = deptno;
  9   IF SQL%NOTFOUND THEN
 10     RAISE invalid_department;
 11   END IF;
 12   COMMIT;
 13 EXCEPTION
 14   WHEN invalid_department THEN
 15     DBMS_OUTPUT.PUT_LINE('No such department id.');
```

old 3: name VARCHAR2(20):='&name';  
NEW 3: name VARCHAR2(20):='testing';  
old 4: deptno NUMBER :=&deptno;  
NEW 4: deptno NUMBER :=9821;  
No such department id.

PL/SQL PROCEDURE successfully completed.

1. .()
2. RAISE, RAISE\_APPLICATION\_ERROR .()
3. .()

SQLCODE	.
SQLERRM	.

SQLCODE

SQL CODE	
0	
1	
+100	NO_DATA_FOUND SQLCODE 100.
	.

```
SQL> CREATE TABLE errors (  
2 e_user VARCHAR2(20),  
3 e_date DATE,  
4 error_code NUMBER(20),  
5 error_message VARCHAR2(100)  
6 );
```

TABLE created.

```
SQL> DECLARE  
2 error_code NUMBER;  
3 error_message VARCHAR2(255);  
4 lastname VARCHAR2(20);  
5 insert_excep EXCEPTION;  
6 PRAGMA EXCEPTION_INIT  
7 (insert_excep, -01400);  
8 BEGIN  
9 SELECT last_name INTO lastname FROM employees WHERE  
10 first_name='Michael';  
11 DBMS_OUTPUT.PUT_LINE ('Michael''s last name is : '||lastname);  
12 EXCEPTION  
13 WHEN insert_excep THEN  
14 DBMS_OUTPUT.PUT_LINE('INSERT OPERATION FAILED');  
15 DBMS_OUTPUT.PUT_LINE(SQLERRM);  
16 WHEN OTHERS THEN  
17 ROLLBACK;  
18 error_code := SQLCODE;  
19 error_message := SQLERRM;  
20 INSERT INTO errors (e_user, e_date, error_code,error_message)  
21 VALUES(USER,SYSDATE,error_code,error_message);  
22 END;  
23 /
```

PL/SQL PROCEDURE successfully completed.

```
SQL> SELECT * FROM errors;
```

E_USER	E_DATE	ERROR_CODE	ERROR_MESSAGE
HR	08-FEB-12	-1422	ORA-01422: exact FETCH returns more than requested NUMBER OF

```
SQL>
```