

#Section4

예외처리 (Exception)

Exception - *Predefined Exceptions*

예외 처리

시스템 정의 예외처리(*Predefined Exceptions*)

사용자 정의 예외처리(User-Defined Exceptions)

전체 구문 Exception 처리

DECLARE

-- 변수선언 , 서브 프로그램

BEGIN

-- Statements

EXCEPTION When 에러유형 then

-- Error 메시지

When 에러유형 then

-- Error 메시지

When others then

-- Error 메시지

END;

Declare

v_name cst_info.name%type;

v_mnu_nm menu.mnu_nm%type;

BEGIN

-- 메뉴명 가져오기

select mnu_nm

into v_mnu_nm

from menu

where mnu_id='M001'

;

-- 고객명 가져오기

select name

into v_name

from cst_info

where cst_id='C001'

;

dbms_output.put_line('이름 : '||v_name);

dbms_output.put_line('메뉴 : '||v_mnu_nm);

EXCEPTION When 에러유형 then

-- Error 메시지

When others then

-- Error 메시지

END;

Exception - *Predefined Exceptions*

<https://docs.oracle.com/en/database/oracle/oracle-database/19/lnpls/plsql-error-handling.html#GUID-8C327B4A-71FA-4CFB-8BC9-4550A23734D6>

Exception Name	Error Code	Explain
ACCESS_INTO_NULL	-6530	초기화되지 않은 조합을 참조합니다
CASE_NOT_FOUND	-6592	CASE 문을 실행하는 중 CASE를 찾을 수 없음
COLLECTION_IS_NULL	-6531	초기화되지 않은 모음을 참조합니다
CURSOR_ALREADY_OPEN	-6511	커서가 이미 열려있습니다
DUP_VAL_ON_INDEX	-1	무결성 제약 조건(.)에 위배됩니다
INVALID_CURSOR	-1001	커서가 부적합합니다
INVALID_NUMBER	-1722	수치가 부적합합니다
LOGIN_DENIED	-1017	사용자명/비밀번호가 부적합, 로그인할 수 없습니다.
NO_DATA_FOUND	+100	데이터를 찾을 수 없습니다.
NO_DATA_NEEDED	-6548	행이 더 이상 필요 없음
NOT_LOGGED_ON	-1012	로그온되지 않았습니다.
PROGRAM_ERROR	-6501	PL/SQL 프로그램 오류
ROWTYPE_MISMATCH	-6504	PL/SQL: Result Set 변수 또는 질의의 리턴 유형이 일치하지 않습니다
SELF_IS_NULL	-30625	NULL SELF 인수에 디스패치 방식이 허용되지 않습니다
STORAGE_ERROR	-6500	PL/SQL: 저장 장치 오류입니다.
SUBSCRIPT_BEYOND_COUNT	-6533	첨자가 개수를 넘었습니다
SUBSCRIPT_OUTSIDE_LIMIT	-6532	첨자가 한계치를 넘었습니다
SYS_INVALID_ROWID	-1410	ROWID가 부적합합니다
TIMEOUT_ON_RESOURCE	-51	리소스를 기다리는 동안 시간 초과가 생겼습니다
TOO_MANY_ROWS	-1422	실제 인출은 요구된 것보다 많은 수의 행을 추출합니다
VALUE_ERROR	-6502	PL/SQL: 수치 또는 값 오류
ZERO_DIVIDE	-1476	제수가 0 입니다

Exception - *Predefined Exceptions*

예외 처리

시스템 정의 예외처리(*Predefined Exceptions*)

사용자 정의 예외처리(User-Defined Exceptions)

블록단위 Exception 처리

DECLARE

-- 변수선언 , 서브 프로그램

BEGIN

Declare

Begin

-- Statements

Exception when 에러유형 then

-- Error 메시지

End;

EXCEPTION When 에러유형 then

-- Error 메시지

When 에러유형 then

-- Error 메시지

When others then

-- Error 메시지

END;

Declare

v_name cst_info.name%type;

v_mnu_nm menu.mnu_nm%type;

BEGIN

Declare

Begin

-- 메뉴명 가져오기

select mnu_nm

into v_mnu_nm

from menu

where mnu_id='M001'

;

Exception when 에러유형 then

-- Error 메시지

End;

-- 고객명 가져오기

select name

into v_name

from cst_info

where cst_id='C001'

;

dbms_output.put_line('이름 : '||v_name);

dbms_output.put_line('메뉴 : '||v_mnu_nm);

EXCEPTION When 에러유형 then

-- Error 메시지

END;

SQLCODE , SQLERRM

SQLCODE Function

In an exception handler, the SQLCODE function returns the numeric code of the exception being handled. (Outside an exception handler, SQLCODE returns 0.)

For an internally defined exception, the numeric code is the number of the associated Oracle Database error. This number is negative except for the error "no data found", whose numeric code is +100.

For a user-defined exception, the numeric code is either +1 (**default**) or the error code associated with the exception by the EXCEPTION_INIT pragma.

예외 처리기에서 SQLCODE 함수는 처리 중인 예외의 숫자 코드를 반환합니다. (예외 핸들러 외부에서 **SQLCODE는 0**을 반환합니다.) 내부적으로 정의된 예외의 경우 숫자 코드는 연관된 Oracle 데이터베이스 오류의 번호입니다. **숫자 코드가 100인 "데이터를 찾을 수 없음" 오류를 제외하고 이 숫자는 음수입니다.**

SQLERRM Function

The SQLERRM function returns the error message associated with an error code.

Note:

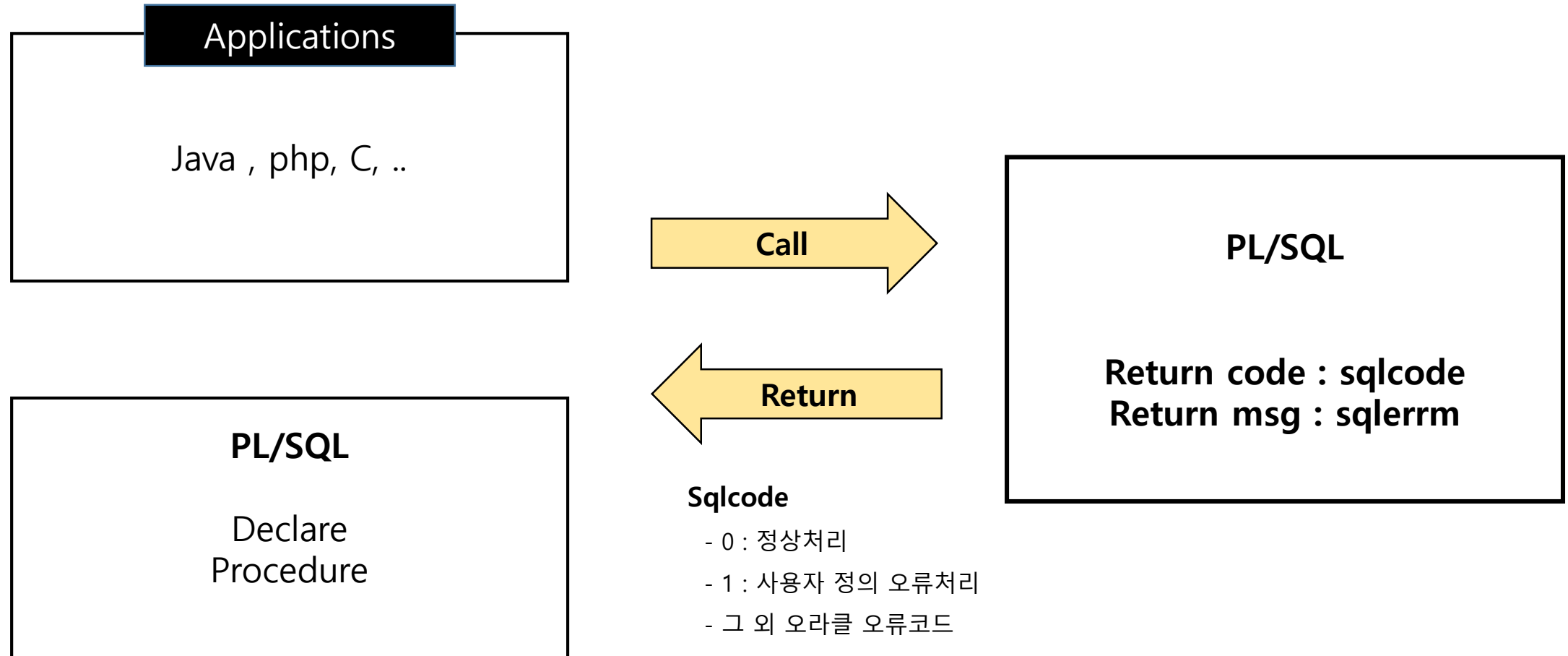
DBMS_UTILITY.FORMAT_ERROR_STACK is recommended over SQLERRM, unless you use the FORALL statement with its SAVE EXCEPTIONS clause. For more information, see "[Retrieving Error Code and Error Message](#)".

SQLERRM 함수는 오류 코드와 관련된 **오류 메시지**를 반환합니다.

Note:

SAVE EXCEPTIONS 절과 함께 FORALL 문을 사용하지 않는 한 SQLERRM보다 **DBMS_UTILITY.FORMAT_ERROR_STACK**을 사용하는 것이 좋습니다.

SQLCODE , SQLERRM



Exception - User-Defined Exceptions (RAISE , RAISE_APPLICATION_ERROR)

예외 처리

시스템 정의 예외처리(*Predefined Exceptions*)

사용자 정의 예외처리(User-Defined Exceptions)

전체 구문 Exception 처리

DECLARE

-- 변수선언 , 서브 프로그램

e_nocoffee EXCEPTION; -- 커피는 예외처리

BEGIN

-- Statements

RAISE past_due; -- 사용자 정의 오류 발생

EXCEPTION When 사용자정의 오류 then

-- Error 메시지

When 에러유형 then

-- Error 메시지

When others then

-- Error 메시지

END;

Declare

v_name cst_info.name%type;

v_mnu_nm menu.mnu_nm%type;

e_nocoffee EXCEPTION; -- 커피는 예외처리

BEGIN

-- 메뉴명 가져오기

select mnu_nm

into v_mnu_nm

from menu

where mnu_id='M001'

;

-- 커피는 안돼..

If v_mnu_nm ='아메리카노' Then

raise v_exception;

End If;

dbms_output.put_line('메뉴 : '||v_mnu_nm);

EXCEPTION When v_exception then

-- 사용자 정의 예외 처리

When others then

-- Error 메시지

END;

Exception - User-Defined Exceptions (RAISE , RAISE_APPLICATION_ERROR)

RAISE Statement

The `RAISE` statement explicitly raises an exception. Outside an exception handler, you must specify the exception name. Inside an exception handler, if you omit the exception name, the `RAISE` statement reraises the current exception.

`RAISE` 문은 명시적으로 예외를 발생시킵니다. 예외 처리기 외부에서는 예외 이름을 지정해야 합니다.

예외 처리기 내에서 예외 이름을 생략하면 `RAISE` 문이 현재 예외를 다시 발생시킵니다.

RAISE_APPLICATION_ERROR Procedure

You can invoke the `RAISE_APPLICATION_ERROR` procedure (defined in the `DBMS_STANDARD` package) only from a stored subprogram or method. Typically,

you invoke this procedure to raise a user-defined exception and return its error code and error message to the invoker.

`RAISE_APPLICATION_ERROR` 프로시저(`DBMS_STANDARD` 패키지에 정의됨)는 저장된 하위 프로그램이나 메서드에서만 호출할 수 있습니다.

일반적으로 이 프로시저를 호출하여 사용자 정의 예외를 발생시키고 해당 오류 코드와 오류 메시지를 호출자에게 반환합니다.

The `error_code` is an integer in the range -20000..-20999

❖ 예외 처리는 어떤 상황에 어떻게 발생시킬 것인가?

- 예외처리는 항상 필요한 것일까?
- RAISE , RAISE_APPLICATION_ERROR 을 활용한 무조건적인 오류 발생

❖ 리턴값 파라미터의 의 위치는 어디가 좋을까?

- 파라미터는 고정된 값일까?
- 파라미터가 늘어나도 리턴 값의 위치는 고정되게

❖ Transaction 처리의 주체는 누가 될 것인가?

- Application vs. PL/SQL
- 단독 실행일 경우 vs. subprogram 으로 실행될 경우