

다양한 예제로 쉽게 배우는

# 오라클 SQL 과 PL/SQL

# 18장 PL/SQL 예외처리

# 18. PL/SQL 예외처리

## 1. PL/SQL 예외란?

- PL/SQL 블록이 PARSE 되는 동안에 오타 등으로 인하여 발생하는 에러를 컴파일 에러(Compilation Error)라고 부르며, PL/SQL 블록이 실행되는 동안에 발생하는 에러를 런타임 에러 (Run-Time Error)라고 부르는데, 이 런타임 에러를 오라클에서는 **예외(Exception)**라고 칭함.
- 오라클의 예외 종류는 두 가지 이며 오라클에서 제공하는 **오라클 예외(ORACLE Exception)**와 다른 하나는 사용자에게 의해 정의되는 **사용자 정의 예외(User-defined Exception)**가 있음.
- 오라클 예외 : **Predefined ORACLE Exception**  
**Non-predefined ORACLE Exception'**

# 18. PL/SQL 예외처리

- **Predefined ORACLE Exception** (미리 정의되어 있는 오라클 예외들)

- 예외명: **ACCESS\_INTO\_NULL**

예외번호: ORA-06530

설명: 정의되지 않은 오브젝트 속성에 값을 할당하고자 했을 때 발생하는 예외.

- 예외명: **CASE\_NOT\_FOUND**

예외번호: ORA-06592

설명: CASE 문의 WHEN 절에 해당되는 조건이 없고 ELSE 절도 없을 경우에 발생하는 예외.

- 예외명: **COLLECTION\_IS\_NULL**

예외번호: ORA-06531

설명: 선언되지 않은 컬렉션 (nested table, varray)에 EXISTS 이외의 메소드를 사용했을 때 발생하는 예외.

# 18. PL/SQL 예외처리

- 예외명: **CURSOR\_ALREADY\_OPEN**

예외번호: ORA-06511

설명: 이미 열려진 커서를 열려고 시도 했을 때 발생하는 예외.

- 예외명: **DUP\_VAL\_ON\_INDEX**

예외번호: ORA-00001

설명: 유일인덱스에 중복값을 입력했을 경우 발생하는 예외.

- 예외명: **INVALID\_CURSOR**

예외번호: ORA-01001

설명: 잘못된 커서 조작이 실행될 때 발생하는 예외.

- 예외명: **INVALID\_NUMBER**

예외번호: ORA-01722

설명: 문자를 숫자로의 변환 시 실패가 될 때 발생하는 예외.

# 18. PL/SQL 예외처리

예외명: **LOGIN\_DENIED**

예외번호: ORA-01017

설명: 잘못된 사용자명 이나 암호로 로그인을 시도했을 때 발생하는 예외.

예외명: **NO\_DATA\_FOUND**

예외번호: ORA-01403

설명: PL/SQL SELECT 문이 한 건도 리턴 하지 못했을 경우 발생하는 예외

예외명: **NOT\_LOGGED\_ON**

예외번호: ORA-01012

설명: 접속되지 않은 상태에서 데이터베이스에 대한 요청이 PL/SQL 프로그램으로 실행된 경우 발생하는 예외.

예외명: **PROGRAM\_ERROR**

예외번호: ORA-06501

설명: PL/SQL 이 내부적인 문제를 가지고 있는 경우 발생하는 예외

# 18. PL/SQL 예외처리

예외명: **ROWTYPE\_MISMATCH**

예외번호: ORA-06504

설명: 할당문에서 호스트 커서 변수와 PL/SQL 커서 변수의 데이터 형이 불일치 할 때 발생하는 예외

예외명: **STORAGE\_ERROR**

예외번호: ORA-06500

설명: PL/SQL 이 실행될 때 메모리가 부족하거나 메모리상에 문제가 일어났을 때 발생하는 예외

예외명: **SUBSCRIPT\_BEYOND\_COUNT**

예외번호: ORA-06533

설명: 컬렉션의 요소 개수보다 더 큰 첨자 값으로 참조한 경우 발생하는 예외.

예외명: **SUBSCRIPT\_OUTSIDE\_LIMIT**

예외번호: ORA-06532

설명: 컬렉션의 첨자의 한계를 벗어난 참조가 일어났을 때 발생하는 예외

# 18. PL/SQL 예외처리

예외명: **SYS\_INVALID\_ROWID**

예외번호: ORA-01410

설명: 문자열을 ROWID 로 변환할 때 무효한 문자열의 표현일 경우 발생하는 예외

예외명: **TIMEOUT\_ON\_RESOURCE**

예외번호: ORA-00051

설명: 자원에 대한 대기시간이 초과했을 때 발생하는 예외

예외명: **TOO\_MANY\_ROWS**

예외번호: ORA-01422

설명: PL/SQL SELECT 문이 두 건 이상의 행을 리턴 했을 때 발생하는 예외

예외명: **VALUE\_ERROR**

예외번호: ORA-06502

설명: 산술, 변환, 절삭 또는 크기 제약에 에러가 생겼을 때 발생하는 예외

예외명: **ZERO\_DIVIDE**

예외번호: ORA-01476

설명: 0으로 나누려 했을 때 발생하는 예외.



# 18. PL/SQL 예외처리

## 2. PL/SQL 블록내의 예외 처리부

```
EXCEPTION
WHEN exception1 [OR exception2 ...] THEN
    statement1 ;
    statement2 ;
...
[ WHEN exception3 [OR exception4 ...] THEN
    statement3 ;
    statement4 ;
... ]
[WHEN OTHERS THEN
    statementN ;
    statementN+1 ;
... ]
```

# 18. PL/SQL 예외처리

## 예외처리 사용 예 1: 오라클에서 사전 정의된 예외 처리하기

사원명이 'A' 로 시작하는 사원을 조회하는 하되 여러 건의 데이터가 나올 경우 에러를 발생시키는 예외 처리부분을 포함하세요

```
SQL> SELECT ename
2  FROM emp
3  WHERE ename LIKE 'A%';
```

ENAME

-----

ALLEN

ADAMS -- A 로 시작하는 사원이 2명 검색됩니다.

이 2건의 데이터를 커서에 넣으려면 명시적 커서를 선언하고 사용해야 하는데 묵시적 커서를 쓰게 하여 에러를 발생시켜 예외처리 실습을 하겠습니다.

# 18. PL/SQL 예외처리

```
SQL> DECLARE
  2   v_ename   emp.ename%TYPE;
  3 BEGIN
  4   SELECT ename INTO v_ename
  5   FROM emp
  6   WHERE ename LIKE 'A%';
  7   DBMS_OUTPUT.PUT_LINE ('사원명은' ||v_ename||' 입니다');
  8 EXCEPTION
  9   WHEN NO_DATA_FOUND THEN
 10     DBMS_OUTPUT.PUT_LINE(' 해당 사원이 없습니다');
 11   WHEN TOO_MANY_ROWS THEN
 12     DBMS_OUTPUT.PUT_LINE(' 사원이 두 명 이상입니다');
 13 END;
 14 /
```

# 18. PL/SQL 예외처리

## 예외처리 사용 예 2. 사용자 정의 예외처리

부서번호를 입력 받아 해당 부서를 삭제하되 해당 부서에 소속된 사원이 있을 경우 '사원이 존재합니다' 라는 예외를 발생시키세요.

```
SQL> DECLARE
  2  sawon_exist EXCEPTION;
  3  PRAGMA EXCEPTION_INIT(sawon_exist,-2292);
  4  BEGIN
  5  DELETE FROM dept
  6  WHERE deptno =&dno;
  7  COMMIT;
  8  EXCEPTION
  9  WHEN sawon_exist THEN
 10  DBMS_OUTPUT.PUT_LINE('사원이 존재합니다');
 11  END ;
 12  /
Enter value for dno: 10
사원이 존재합니다
```

# 18. PL/SQL 예외처리

```
SCOTT>CREATE TABLE t_pragma  
2 (no NUMBER PRIMARY KEY ,  
3 name VARCHAR2(10));
```

Table created.

```
SCOTT>INSERT INTO t_pragma  
2 VALUES(1,'AAA') ;
```

1 row created.

```
SCOTT>INSERT INTO t_pragma  
2 VALUES(1,'BBB');  
INSERT INTO t_pragma  
*
```

ERROR at line 1:

**ORA-00001: unique constraint (SCOTT.SYS\_C0013626) violated**

PRAGMA 기능 이해하기

다음 장에 계속.....

# 18. PL/SQL 예외처리

```
SCOTT> DECLARE
2  new_msg EXCEPTION;
3  PRAGMA EXCEPTION_INIT(new_msg, -1);
4  BEGIN
5  INSERT INTO t_pragma
6  VALUES(1,'CCC');
7  EXCEPTION
8  WHEN new_msg THEN
9  DBMS_OUTPUT.PUT_LINE('존재하는 번호입니다!');
10 END ;
11 /
```

# 18. PL/SQL 예외처리

```
SCOTT>CREATE TABLE t_con
 2 ( no NUMBER NOT NULL ,
 3   name VARCHAR2(10));
```

Table created.

```
SCOTT>INSERT INTO t_con
 2 VALUES(1,'AAA');
```

1 row created.

PRAGMA 테스트 2

```
SCOTT>INSERT INTO t_con
 2 VALUES(null,'BBB');
VALUES(null,'BBB')
```

\*

ERROR at line 2:

**ORA-01400: cannot insert NULL into ("SCOTT"."T\_CON"."NO")**

다음 장에 계속....

# 18. PL/SQL 예외처리

```
SCOTT> DECLARE
  2 new_msg EXCEPTION;
  3 PRAGMA EXCEPTION_INIT(new_msg,-1400);
  4
  5 BEGIN
  6   INSERT INTO t_con
  7     VALUES(null,'BBB');
  8
  9 EXCEPTION
 10   WHEN new_msg THEN
 11     DBMS_OUTPUT.PUT_LINE('null 입력불가');
 12 END;
 13 /
```

null 입력불가

PL/SQL procedure successfully completed.



# 18. PL/SQL 예외처리

## - 예외 처리 사용 예 3: RAISE 를 사용하여 예외 처리하기

empno 를 입력 받은 후 emp 테이블에서 해당 사원을 지우는 작업을 수행합니다. 단 없는 사원번호를 입력할 경우 '사원이 없습니다' 라는 예외 메시지를 출력하게 하세요.

```
SCOTT>DECLARE
2     v_empno  emp.empno%TYPE;
3     v_name   emp.ename%TYPE;
4     no_empno EXCEPTION ;
5 BEGIN
6     DELETE FROM EMP
7     WHERE empno =&empno;
8     IF SQL%NOTFOUND THEN
9         RAISE no_empno ;
10    END IF;
11 EXCEPTION
12    WHEN no_empno THEN
13        DBMS_OUTPUT.PUT_LINE('사원이 없습니다');
14 END ;
15 /
```

# 18. PL/SQL 예외처리

## - 예외 처리 사용 예 4 : RAISE\_APPLICATION\_ERROR 프로시저 사용하기

RAISE\_APPLICATION\_ERROR 프로시저를 사용하여 사용자가 에러를 정의하고 즉시 예외를 처리하는 방식입니다.

이때 사용 가능한 에러 번호는 20000 번부터 20999 번 까지 입니다.

```
SCOTT> DECLARE
2   v_empno  emp.empno%TYPE;
3   v_name   emp.ename%TYPE;
4
5 BEGIN
6   DELETE FROM EMP
7   WHERE empno = &empno;
8   IF SQL%NOTFOUND THEN
9     RAISE_APPLICATION_ERROR(-20100, '존재 하지 않는 사번입니다');
10  END IF;
11  END ;
12  /
```

# 18. PL/SQL 예외처리

```
SCOTT> DECLARE
  2   v_empno  emp.empno%TYPE;
  3   v_name   emp.ename%TYPE;
  4
  5 BEGIN
  6   DELETE FROM EMP
  7   WHERE empno = &empno;
  8   IF SQL%NOTFOUND THEN
  9     RAISE_APPLICATION_ERROR(-20100, '존재 하지 않는 사번입니다');
 10   END IF;
 11 END ;
 12 /
```

정확한 데이터 입력 후 테스트

아무 이상 없이 잘 수행됩니다

Enter value for empno: 7900

PL/SQL procedure successfully completed.