

REM Oracle **EXCEPTION**

1. What?

- 1) **PL/SQL** 블록이 **PARSE** 되는 동안에 오라 등으로 인하여 발생하는 에러를 컴파일 에러라고 한다.
- 2) **PL/SQL** 블록이 실행되는 동안에 발생하는 에러를 런타임 에러라고 부르는데, 이 런타임 에러를 예외(**EXCEPTION**)이라고 한다.
- 3) 즉 **PL/SQL** 블록 내의 **SQL**문이 정상적으로 실행되지 못할 때 예외가 발생함
- 4) 예외가 발생하면 **EXCEPTION** 절에 의해 처리할 수 있다.
- 5) 자주 발생하는 예외처리기는 Oracle 사에서 기본적으로 제공하며, 자주 발생하지 않는 예외는 사용자가 직접 정의가능
- 6) 종류
 - a. 크게 두가지
 - Oracle **EXCEPTION** : Oracle 에서 제공
 - User-defined EXCEPTION** : 사용자 정의 예외
 - b. **Predefined (Internal) EXCEPTION** : 이름이 정해진 오라클 예외
 - c. **Non-defined EXCEPTION** : 오라클 예외이지만 이름이 명명되지 않은 예외, 암시적으로 발생, 즉 특정 상황이 되면 자동으로 발생

2. 사용자 정의 예외

- 1) 오라클 예외와 달리 선언부에서 **EXCEPTION** 형으로 예외를 선언한다음, 실행부 또는 예외 처리부에서 **RAISE** 문을 통해 명시적으로 예외를 발생시킨다.
- 2) 또 다른 방법은, **RAISE_APPLICATION_ERROR** 프로시저를 사용하기도 한다. 이 프로시저를 사용하면 예외를 정확히 호출환경으로 전달할 수 있다는 장점
- 3) 예외가 발생하면 예외가 발생한 **SQL**문은 처리가 중단되며, 그 후 예외처리가 있다면 예외 처리부로 넘어가서 예외처리부에서 기술된 조건으로 진행되고 만일 예외처리부가 없다면 실패한 상태로 **PL/SQL**블록을 종료하게 된다.

3. Syntax

```
DECLARE
...
BEGIN
    SELECT ...
    EXCEPTION
        WHEN exception1 [OR exception2...] THEN
            statement1;
            statement2;
        WHEN exception3 [OR exception4...] THEN
            statement3;
            statement4;
        WHEN exception5 [OR exception6...] THEN
            statement5;
            statement6;
        WHEN OTHERS THEN
            statement7;
            statement8;
END ;
/
```

-**OTHERS** : 위에 기술된 어느 예외에도 속하지 않는 기타 예외를 뜻하며 가장 마지막에 기술되는 **WHEN** 에만 사용한다.

REM Predefined **EXCEPTION**

1. Oracle 사에서 제공하는 미리 정의된 에러 처리 조건

- 1) **NO_DATA_FOUND**(Error Code : **ORA-01403**)
 - SQL**문에 의한 검색 조건을 만족하는 결과가 전혀 없는 조건의 경우
- ```
CREATE OR REPLACE PROCEDURE sp_exception_test
```

```

49 IS
50 v_empno emp.empno%TYPE;
51 v_ename emp.ename%TYPE;
52 v_sal emp.sal%TYPE;
53 BEGIN
54 SELECT empno, ename, sal
55 INTO v_empno, v_ename, v_sal
56 FROM emp
57 WHERE deptno = 99;
58
59 EXCEPTION
60 WHEN NO_DATA_FOUND THEN
61 DBMS_OUTPUT.PUT_LINE('만족하는 검색 조건이 없습니다 !!');
62 END ;
63 /

```

```

65 SQL>SET SERVEROUT ON
66 SQL>EXEC sp_exception_test;
67 만족하는 검색 조건이 없습니다 !!

```

## 2)NOT\_LOGGED\_ON(Error Code : ORA-01012)

-데이터베이스에 연결되지 않은 상태에서 **SQL**문을 실행하려는경우 발생

```

71 CREATE OR REPLACE PROCEDURE sp_exception_test
72 IS
73 v_empno emp.empno%TYPE;
74 v_ename emp.ename%TYPE;
75 v_sal emp.sal%TYPE;
76 BEGIN
77 SELECT empno, ename, sal
78 INTO v_empno, v_ename, v_sal
79 FROM emp
80 WHERE deptno = 99;
81
82 EXCEPTION
83 WHEN NOT_LOGGED_ON THEN
84 DBMS_OUTPUT.PUT_LINE('DB접속이 해제 되었습니다 !!');
85 END ;
86 /

```

## 3)TOO\_MANY\_ROWS(Error Code : ORA-01422)

-**SQL** 문의 실행 결과가 여러 개의 행을 리턴하는 경우 스칼라 변수에 저장하려고 할 때 발생

```

90 CREATE OR REPLACE PROCEDURE sp_exception_test
91 IS
92 v_empno emp.empno%TYPE;
93 v_ename emp.ename%TYPE;
94 v_sal emp.sal%TYPE;
95 BEGIN
96 SELECT empno, ename, sal
97 INTO v_empno, v_ename, v_sal
98 FROM emp
99 WHERE deptno = 20;
100
101 EXCEPTION
102 WHEN TOO_MANY_ROWS THEN
103 DBMS_OUTPUT.PUT_LINE('조건을 만족하는 데이터가 너무 많습니다 !!');
104 END ;

```

```

105 /
106
107 4) DUP_VAL_ON_INDEX
108 -UNIQUE 제약을 갖는 컬럼에 중복되는 데이터가 INSERT 될 때
109 5) ZERO_DIVIDE
110 -0으로 나눌 때
111 6) INVALID_CURSOR
112 -잘못된 커서 연산
113
114 CREATE OR REPLACE PROCEDURE PreException_test
115 (v_deptno IN emp.deptno%TYPE)
116 IS
117 v_emp emp%ROWTYPE;
118 BEGIN
119 DBMS_OUTPUT.ENABLE;
120 SELECT empno, ename, deptno
121 INTO v_emp.empno, v_emp.ename, v_emp.deptno
122 FROM emp
123 WHERE deptno = v_deptno ;
124
125 DBMS_OUTPUT.PUT_LINE('사번 : ' || v_emp.empno);
126 DBMS_OUTPUT.PUT_LINE('이름 : ' || v_emp.ename);
127 DBMS_OUTPUT.PUT_LINE('부서번호 : ' || v_emp.deptno);
128
129 EXCEPTION
130 WHEN DUP_VAL_ON_INDEX THEN
131 DBMS_OUTPUT.PUT_LINE('데이터가 존재 합니다.');
```

132 DBMS\_OUTPUT.PUT\_LINE('DUP\_VAL\_ON\_INDEX 에러 발생');

```

133 WHEN TOO_MANY_ROWS THEN
134 DBMS_OUTPUT.PUT_LINE('TOO_MANY_ROWS에러 발생');
```

135 WHEN NO\_DATA\_FOUND THEN

```

136 DBMS_OUTPUT.PUT_LINE('NO_DATA_FOUND에러 발생');
```

137 WHEN OTHERS THEN

```

138 DBMS_OUTPUT.PUT_LINE('기타 에러 발생');
```

```

139 END;
140 /
141
142 -- DBMS_OUTPUT.PUT_LINE을 출력하기 위해 사용
143 SQL> SET SERVEROUTPUT ON ;
144
145 -- 프로시저 실행
146 SQL> EXECUTE PreException_Test(20);
147
148 TOO_MANY_ROWS에러 발생
149
150 -- TOO_MANY_ROWS 에러가 발생하는 이유?
151 - SELECT문의 결과가 1개 이상의 행을 리턴하기 때문이다..
152 - TOO_MANY_ROWS를 피하기 위해서는 FOR문이나 LOOP문으로 SELECT문을 처리해야 한다.
```

153

```

154 --아래와 같이 변경하면 에러가 발생하지 않는다.
155
156 FOR emp_list IN
157 (SELECT empno, ename, deptno
158 FROM emp
159 WHERE deptno = v_deptno) LOOP
```

```

160
161 DBMS_OUTPUT.PUT_LINE('사번 : ' || emp_list.empno);
162 DBMS_OUTPUT.PUT_LINE('이름 : ' || emp_list.ename);
163 DBMS_OUTPUT.PUT_LINE('부서번호 : ' || emp_list.deptno);
164 END LOOP;
165
166

```

167 REM 미리 정의되지 않은 예외(Undefined **Exception**)

168 1. 미리정의되지 않은 예외는 사용자가 선언절에서 예외명을 정의하고, Oracle Server에서 제공하는 Error 번호를 사용하여, 정의된 예외와 연결한 후 **EXCEPTION**절에서 정의된 예외를 사용한다.

169 2. Steps

170 1) STEP 1 : 예외의 이름을 선언 (선언절)

171 2) STEP 2 : **PRAGMA** EXCEPTION\_INIT문장으로 예외의 이름과 오라클 서버 오류 번호를 결합 (선언절)

172 3) STEP 3 : 예외가 발생할 경우 해당 예외를 참조한다(예외절)

```

173
174 CREATE OR REPLACE PROCEDURE NonPreException_Test
175 IS
176 not_null_test EXCEPTION; -- STEP 1
177 /* not_null_test는 선언된 예외 이름
178 -1400 Error 처리번호는 표준 Oracle7 Server Error 번호 */
179 PRAGMA EXCEPTION_INIT(not_null_test, -1400); -- STEP 2
180 BEGIN
181 DBMS_OUTPUT.ENABLE;
182 -- empno를 입력하지 않아서 NOT NULL 에러 발생
183 INSERT INTO emp(ename, deptno)
184 VALUES('tiger', 30);
185
186 EXCEPTION
187 WHEN not_null_test THEN -- STEP 3
188 DBMS_OUTPUT.PUT_LINE('not null 에러 발생 ');
189 END;
190 /
191
192 -- DBMS_OUTPUT.PUT_LINE을 출력하기 위해 사용
193 SQL> SET SERVEROUTPUT ON ;
194 -- 실행 결과
195 SQL> EXECUTE NonPreException_Test;
196 not null 에러 발생

```