

## 1. 예외처리

일반적으로 프로그램을 작성하고 실행할 때, 두 가지 종류의 오류(Error)가 발생한다. PL/SQL에서는 블록의 실행 오류를 예외(Exception)라 부른다.

### ☞ 문법적인 오류

문법적으로 잘못된 명령문이 입력되었을 때 발생하는 오류이다. 예를 들어, SELECT문이 'SELET', 'SSELECT'등의 철자가 잘못 입력되었거나, INSERT (A, B) VALUES(111, 'AAA'); 문이 'INSERT (A, B) (111, 'AAA);' 등으로 예약어가 누락된 경우 오류가 발생한다. 이런 문법적인 오류는 철자를 수정하거나, 누락된 예약어를 추가하여 다시 실행한다.

### ☞ 실행 오류

프로그램의 로직상 오류나 데이터 입력 오류를 발생한다. 예를 들어, 블록 실행시 처리할 데이터나 데이터타입 오류, 저장할 변수나 칼럼의 자릿수 부족, 명령문 누락이나 순서 바뀜 등 실행이 되지 않거나, 잘못된 결과가 발생하는 오류를 말한다. 이 실행 오류는 찾아내기가 어려우나, 오류 메시지 또는 실행 결과를 검사하여 찾아낸다.

예외(Exception)란 PL/SQL 블록의 실행절에 기술된 명령문이 실행될 때 오라클 데이터베이스가 발생하는 오류를 말한다. 예외는 자동으로 발생할 수 있고, 사용자가 예외를 발생시킬 수도 있다. 예외가 발생되면 실행(BEGIN)절의 실행은 무조건 중단하고, 예외처리(EXCEPTION) 절로 제어를 이동하여 예외명에 해당되는 명령문을 실행하고 블록은 종료된다.

### 1.1. 예외처리절 구문

문법	EXCEPTION
	WHEN 예외명1 THEN
	명령문1; .....; 명령문n;
	WHEN 예외명2 THEN
	명령문1; .....; 명령문n;
	.....
	WHEN OTHERS THEN
	명령문1; .....; 명령문n;

※ 기술 방법

- 예외명1, 예외명2, .... : 오라클 데이터베이스가 자동으로 발생하거나, 사용자가 예외를 발생시키는 예외명을 기술한다.
- 명령문1; .....; 명령문n; : 예외명에 대한 처리 명령문들을 기술한다.
- OTHERS : 예외처리절에서 가장 마지막에 기술되는 것으로, 예외 발생시 예외처리절에서 예외

명을 발견하지 못할 때, 실행되는 예외명이다.

#### ☞ SELECT문 실행시 발생하는 예외

PL/SQL 블록에서 SELECT문이 실행될 때, 결과 행이 1행이면 정상적으로 실행되고, 0행이나 다수 행이면 자동적으로 예외가 발생된다. 이 때, 오라클 데이터베이스는

- 0행이면 NO\_DATA\_FOUND 예외가 자동으로 발생되고,
- 다수 행이면 TOO\_MANY\_ROWS 예외가 자동으로 발생한다.

**예제) Professor 테이블과 Course 테이블을 이용하여 'P12' 교수가 담당하는 과목을 검색하여 출력하는 익명의 블록을 만들어 보자. 예외처리절을 기술하지 않고 만들어서 실행해보기**

**이후 예외처리를 기술해서 다시 만들고 실행해보자.**

## 1.2. 오라클 데이터베이스의 예외

PL/SQL 블록을 실행할 때 발생하는 예외는 미리 정의된 예외와 사용자 정의 예외로 구분한다.

#### ☞ 미리 정의된 예외명

오라클 데이터베이스가 예외를 자동으로 발생하는 미리 정의된 예외명은 다음과 같다.

예 외 명	예외 발생 요인
TOO_MANY_ROWS	SELECT문의 결과가 하나 이상의 행을 반환되면 발생
NO_DATA_FOUND	SELECT문의 결과가 0행을 반환하면 발생
DUP_VAL_ON_INDEX	고유 인덱스를 갖는 칼럼에 중복되는 값을 삽입할 때 발생
VALUE_ERROR	산술, 변환, 절삭, 제한 오류가 발생
INVALID_NUMBER	변환할 수 없는 문자를 숫자로 변환할 때 발생
ZERO_DIVIDE	제수가 0일 때 발생( $a/b=c$ 일 때 $b$ 가 제수)
OTHERS	EXCEPTION절에 정의된 예외가 없을 때 발생

**예제) 1. Department 테이블에 학과코드를 '컴공', 학과명을 '컴퓨터공학과', 전화번호를 '765-4100'으로 입력하는 블록을 작성하여 실행해보자.[기본 키 중복 예외]**

**2. Course 테이블의 과목코드가 'L1031'에 대하여 추가 수강료(Course\_Fees)를 '삼만원'으로 수정해보자.[숫자형 데이터타입의 데이터 오류]**

#### ☞ 사용자 정의 예외

오라클 데이터베이스가 자동으로 발생하지 않는 예외에 대하여 사용자가 발생 가능한 예외명을 블록에 선언하고 예외를 발생시키는 것을 말한다.

- 예외명 선언 구문

문법	예외명	EXCEPTION;
----	-----	------------

- 예외를 발생시키는 명령문

RAISE문으로 실행절에서 사용자가 정의한 예외를 발생시킨다.

문법	RAISE	예외명;
----	-------	------

**예제)** SG\_Scores 테이블에 저장된 Score 컬럼의 점수가 100점이 초과되는 값이 있는지 조사하는 블록을 작성하시오. 단, 100점 초과시 Over\_Score 예외를 선언한다.

### 1.3. 오류 메시지 확인

발생된 오류와 오류 메시지를 확인할 수 있는 방법이 있다.

- SQLCODE는 사전에 정의된 기호가 저장되고, 정상적일 때 0이 된다.
- SQLERRM은 SQLCODE 기호에 대한 메시지를 저장한다. 정상적으로 실행되었을 때, 메시지는 'ORA-0000: normal, successful, completion'이라는 문자열이 자동으로 저장된다.

**예제)** 예외코드와 예외 메시지를 출력하는 익명의 블록을 만들고 실행해보자.

## 2. 커서를 이용한 복수행 처리

PL/SQL에서 SELECT문은 단일의 결과 행을 INTO절의 변수에 저장하고, 0행이나 복수 행이 검색되면 에러가 발생한다. 커서는 임의의 복수 행을 검색하기 위한 방법으로 사용한다. 이 기법은 PL/SQL의 강력한 기능의 하나이다.

### 2.1. 커서

SELECT문으로 임의의 복수 행을 검색하기 위한 메커니즘으로, 실행절에서 액세스할 수 있는 데이터의 집합을 선언절에서 오라클 데이터베이스의 메모리 영역에 사용자가 선언하는 하나의 객체이다.

즉, 커서는 질의의 결과로 얻어진 여러 행이 저장된 메모상의 위치(포인터)를 말한다.

커서는 커서의 선언과 처리 방법에 따라 명시적인 커서(explicit cursor)와 암시적인 커서(implicit cursor)로 구분한다. 커서의 선언 방법은 동일하나 검색하여 처리하는 방법이 다르다.

## 2.2. 명시적인 커서

다음의 네 가지 단계에 의해서 처리된다.

- 1단계 : 커서를 선언한다.
- 2단계 : 커서를 연다.
- 3단계 : 커서로부터 한 행을 가져온다.
- 4단계 : 커서를 닫는다.

1단계는 DECLARE절에 선언하고, 2단계, 3단계, 4단계는 BEGIN절에서 처리한다.

### (1) 커서의 선언

커서는 DECLARE 선언절에 다음과 같이 선언한다.

문법	CURSOR 커서명 [ (형식인자, ... ) ] IS SELECT문;
----	---

※ 기술 방법

- 커서명은 선언할 커서명이며, 일반적인 객체의 이름 정의 규칙에 따른다.
- 형식인자는 커서가 동적으로 실행하기 위해 값을 전달받는 인자이다.

형식인자명    데이터타입    [ := 기본 값 ];
--------------------------------

- SELECT문은 INTO절이 없는 SQL의 SELECT문으로, 하나 이상의 행을 반환하는 SELECT문을 기술한다.

### (2) 커서를 연다.

BEGIN절에서 DECLARE절에 선언된 커서를 OPEN문으로 연다. 오픈(Open)된 커서의 작업 포인트는 테이블에 저장된 순서의 첫 번째 행에 위치한다.

문법	OPEN    커서명;
----	--------------

※ 커서명은 1단계의 DECLARE절에 선언한 커서명이다.

### (3) 커서로부터 한 행을 인출한다.

BEGIN절에서, 오픈(Open)된 커서로부터 FETCH문으로 한 행을 인출한다. FETCH문이 실행될 때마다 커서로부터 한 행씩 인출된 데이터를 INTO절의 변수명1, 변수명2, ..... 에 기억한다.

문법	FETCH    커서명    INTO    변수명1, 변수명2, .....
----	---

또, 인출된 커서의 작업 포인터는 다음 포인터로 자동 이동한다.

- 커서는 순차적으로만 인출할 수 있다.
- 커서에서 인출되는 칼럼의 수와 변수의 수가 동일해야 한다.
- 커서에서 인출되는 칼럼과 변수의 데이터타입이 동일해야 한다.

#### (4) 커서를 닫는다.

커서의 사용이 끝나면, CLOSE문으로 커서를 닫는다.

문법	CLOSE    커서명;
----	---------------

### 2.3. 커서의 속성변수

커서를 사용할 때 커서가 오픈되었는지, 인출할 행이 존재하는지 커서의 속성을 확인하는 변수들이 있다. 커서의 속성을 관리하기 위한 속성변수는 다음과 같다.

속성 변수	내 용	사 용 예
%ISOPEN	커서가 오픈 상태인가를 검사, 오픈 되었으면 참이 되고, 그렇지 않으면 거짓	커서명%ISOPEN
%FOUND	커서로부터 행을 인출하여, 성공하면 참이 되고, 실패하면 거짓	커서명%FOUND
%NOTFOUND	커서로부터 행을 인출하여, 실패하면 참이 되고, 성공하면 거짓	커서명%NOTFOUND
%ROWCOUNT	인출된 행의 수를 반환	커서명%ROWCOUNT

커서의 속성 변수들은 다음과 같이 사용한다.

문법	커서명%속성변수;
----	-----------

**예제) Department 테이블의 모든 행을 출력하는 PL/SQL 블록을 만들어보자.**

### 2.4. 암시적 커서

오라클 데이터베이스나 PL/SQL 실행 메커니즘에 의해 처리되는 SQL문장이 처리되는 곳에 대한 익명의 주소이다. 커서 FOR~LOOP문으로 커서에 있는 모든 행들을 반복적으로 인출하는 것을 말한다. 커서는 선언되나 OPEN, FETCH, CLOSE가 자동으로 실행된다.

#### ☞ 암시적 커서는 명령문이 실행될 때

- 커서는 자동으로 오픈(open)되고

- 커서 FOR~LOOP문이 반복할 때마다 커서로부터 한 행씩 인출된다. 인출된 값은 '레코드변수. 커서칼럼명'으로 사용할 수 있고, 커서의 모든 행이 인출되면 커서 FOR~LOOP문은 자동으로 종료된다.
- 모든 행이 인출되면 커서는 자동으로 닫히게 된다.

문법	<pre> FOR 레코드변수 IN 커서명 LOOP     처리명령문1; ....; 처리명령문N; END LOOP; </pre>
----	--

**예제) Department 테이블의 모든 행을 출력하는 PL/SQL 블록을 암시적인 커서로 만들어보자.**