

6-4 PL/SQL CURSOR

1. SQL 커서란?
2. 묵시적 커서
3. 명시적 커서
4. 명시적 커서 처리 단계
5. Cursor FOR Loop 문 활용하기

1. SQL 커서란?

오라클 서버에서는 SQL 문을 실행할 때마다 처리(Parse, Execution)를 위한 메모리공간 (이 공간을 이후부터 SQL 커서라고 부르겠습니다)을 사용합니다. 즉 사용자가 요청하는 데이터를 데이터베이스 버퍼 캐쉬에서 커서로 복사해 온 후 커서에서 원하는 데이터를 추출하여(Fetch) 후속 작업을 하게 된다는 뜻입니다. 이 메모리 공간을 Private SQL Area 라고도 부르며, 오라클의 서버 프로세스 구성이 Dedicated Server 환경이냐 또는 MTS(Multi-Threaded Server)환경이냐에 따라 서버 내에 위치되는 곳이 다릅니다.

SQL 커서는 크게 **묵시적 커서(Implicit Cursor)**와 **명시적 커서(Explicit Cursor)**로 나눌 수 있습니다

2. 묵시적 커서(Implicit Cursor)

- 묵시적 커서는 오라클에서 자동적으로 선언해주는 SQL 커서로서, 사용자는 생성 유무를 알 수 없습니다.
- 묵시적 커서에 저장되는 데이터는 1 행만 가능합니다

•묵시적 커서 속성(Cursor Attribute)

- **SQL%ROWCOUNT**
- **SQL%FOUND**
- **SQL%NOTFOUND**
- **SQL%ISOPEN**

3. 명시적 커서(Explicit Cursor)

명시적 커서는 사용자가 선언하여 생성 후 사용하는 SQL 커서로, 주로 여러 개의 행을 처리하고자 할 경우 사용합니다.

- 명시적 커서 속성(Cursor Attribute)

- 커서이름%ROWCOUNT
- 커서이름%FOUND
- 커서이름%NOTFOUND
- 커서이름%ISOPEN

4. 명시적 커서(Explicit Cursor) 처리 단계



4. 명시적 커서(Explicit Cursor) 처리 단계

1) 명시적 커서 선언(Declaration)

```
CURSOR 커서명  
IS  
커서에 담고 싶은 내용을 가져오는 서브쿼리
```

2) 명시적 커서 열기(Open)

커서 선언 시 기술했던 서브쿼리를 수행해서 데이터를 커서로 가져 오는 과정

```
OPEN 커서 이름 ;
```

4. 명시적 커서(Explicit Cursor) 처리 단계

3) 명시적 커서로부터 데이터 읽어서 변수로 할당하기(Fetch)

```
FETCH 커서_이름 INTO 변수들 ;
```

4) 명시적 커서 닫기(Close)

```
CLOSE 커서_이름;
```

4. 명시적 커서(Explicit Cursor) 처리 단계

SCOTT>declare

2 vempno number(4);

3 vename varchar2(20);

4 vsal number(7);

5

6 cursor c1 is

7 select empno , ename , sal

8 from emp

9 where deptno=20;

10

명시적 커서 사용 예

C1 이라는 이름의 명시적 커서를 선언한 부분. 이 커서가 open 될 때 마치 view처럼 서브쿼리가 수행됩니다

다음 장에 계속....

4. 명시적 커서(Explicit Cursor) 처리 단계

```
11 begin
12  open c1;
13
14  dbms_output.put_line('번호   이름   급여');
15
16  loop
17    fetch c1 into vempno , vename , vsal ;
18    exit when c1%notfound;
19
20    dbms_output.put_line(vempno||' '||vename||' '||vsal);
21  end loop;
22  close c1;
23 end;
24 /
```

커서 open 합니다

커서에 있는 데이터를
Fetch해서 변수로 할당합니다.

커서를 close 합니다

5. Cursor FOR Loop 문 활용하기

```
FOR record_name IN cursor_name LOOP
```

```
-- 명시적 커서의 OPEN, FETCH 가 자동적으로 수행됨.
```

```
  statement1 ;
```

```
  statement2 ;
```

```
....
```

```
END LOOP ; -- 루프문을 빠져 나갈 때 자동적으로 커서가 CLOSE 됨.
```

5. Cursor FOR Loop 문 활용하기

- CURSOR FOR LOOP 예 1 :

: emp 테이블에서 empno, ename 값을 가져와서 커서에 저장 후 출력하세요.

```
SCOTT>DECLARE
2  CURSOR emp_cur IS -- 커서 선언됩니다
3      SELECT empno ,ename
4      FROM emp;
5  BEGIN
6  FOR emp_rec IN emp_cur -- 커서의 데이터를 저장할 emp_rec 변수 선언됨
7  LOOP
8      DBMS_OUTPUT.PUT_LINE(emp_rec.empno||' '||emp_rec.ename);
9  END LOOP;
10 END;
11 /
```

5. Cursor FOR Loop 문 활용하기

- CURSOR FOR LOOP 예 2 :

아래 예는 사용자가 직접 vemp 라는 emp%ROWTYPE 변수를 선언하고 FOR 문장에서 사용하는 예입니다.

```
SCOTT>DECLARE
  2  vemp emp%ROWTYPE;
  3
  4  CURSOR c1 IS
  5      SELECT empno,ename,sal
  6      FROM emp ;
  7
  8  BEGIN
  9      FOR vemp IN c1 LOOP
 10          EXIT WHEN c1%NOTFOUND ;
 11          DBMS_OUTPUT.PUT_LINE(vemp.empno||' '||vemp.ename||' '||vemp.sal);
 12      END LOOP;
 13  END ;
 14  /
```

5. Cursor FOR Loop 문 활용하기

- CURSOR FOR LOOP 예 3 :

이번 예는 Sub query 를 사용하여 작업을 하는 예 입니다.
이 방법은 DECLARE 부분에 커서를 선언 할 필요가 없습니다.
대신 커서에 들어가는 서브쿼리 부분을 IN 뒷부분에 바로 사용하는 방법입니다.

```
SCOTT>DECLARE
2 BEGIN
3 FOR emp_rec IN (SELECT empno, ename, sal
4                 FROM emp)
5 LOOP
6   DBMS_OUTPUT.PUT_LINE(emp_rec.empno||' '||emp_rec.ename||' '||emp_rec.sal);
7 END LOOP;
8 END ;
9 /
```