커서(CURSOR)

SELECT 문의 수행 결과가**여러개의로우(행,레코드)**로 구해지는 경우에 모든 행에 대해 어떤 처리를 하고싶을때 사용함

커서를 하나의 저장공간이라 생각하고 저장공간의 특성은 FIFO 구조라고 할 경우 가장 윗단에는 시작하는 블록이 있고 마지막에는 종료 블록이 있다. 그 사이에 결과값을 저장하게 된다.

커서는 CURSOR, OPEN, FETCH, CLOSE 4 단계 명령에 의해서 사용됨.

커서에는 명시적 커서와 묵시적 커서 두가지 유형이 있다.

명시적 커서

명시적커서(explicit cursor)란 사용자가 직접 쿼리의 결과에 접근해서 이를 사용하기 위해 명시적으로 선언한 커서

커서의 작업 순서 : 커서의 선언(CURSOR) -> 커서열기 (OPEN) -> 커서의 패치 (FETCH) -> 커서 닫기 (CLOSE)

형식

DECLARE -- (1) 커서 선언 : 커서에 이름을 주고, 이 커서가 접근하려는 쿼리를 정의

CURSOR *cursor_name* IS *select statement;*

BEGIN

OPEN *cursor_name;* --(2) 커서 열기 : 커서로 정의된 쿼리를 실행하는 역할을 한다.

LOOP

FETCH *cursor_name* INTO *variable_name*;

--(3) 패치(fetch) : 쿼리의 결과에 접근, 보통 한개이상의 결과를 반환하므로 루프를 돌며 개별 값들에 접근해서 임의의 처리를 하게됨.

EXIT WHEN [조건];

END LOOP;

CLOSE *cursor_name*; --(4) 커서 닫기 : 패치작업이 끝나면 사용된 커서를 닫는다. 닫는다는 의미는 커서처리를 끝내고 메모리상에 존재하는 쿼리의 결과를 소멸한다는 의미.

END;

- 커서의 속성

*cursor_name***%ROWCOUNT** : 카운터(counter) 역할, 커서가 막 오픈되었을때는 0, 패치될때마다 1 씩 증가

cursor_name%**FOUND** : 커서가 막 오픈된 상태에서는 NULL 값을 반환,

오픈되고첫번째패치가발생한 이후부터는 TRUE, 더이상패치할로우가 없을 경우 FALSE 를 반환,

cursor_name%**NOTFOUND** : %FOUND 와 반대개념, 더이상패치할로우가 없으면 TRUE

cursor_name%**ISOPEN** : 커서가 오픈된 상태일 경우 TRUE 값을 반환

- for 문을 활용한 커서

for 문을 활용하여 커서를 사용할 경우 명시적 커서에서의 열기, 패치, 닫기를 자동으로 적용시켜 주게 된다. 따라서 커서의 작업순서를 명시하지 않아도 된다. 단, 사용하기 위해서는 참조하는 커서가 있어야 하며, FOR UPDATE 절이 커서 선언 query 문장 안에 있어야 한다.

형식
CREATE OR REPLACE PROCEDURE insaSelect -- create 프로시저 (--매개변수 선언)

```
IS
  -- 커서의 선언
  CURSOR cursor_name IS select statement;
BEGIN
  FOR record_name IN cursor_name loop -- cursor_name 커서에 저장된 결과를 하나씩 불러온다. (레코드단위로)
    statement;
  END loop;
END;
```

묵시적 커서

묵시적커서(implicit cursor)란 오라클 내부에서 각각의 쿼리 결과에 접근하여 사용하기 위한 내부적 커서라할수있다. 묵시적 커서는 모든 쿼리가 실행될때마다오픈된다. 오라클 내부에서 접그하고 사용되는 커서이므로 선언, 오픈등의 작업을 할필요가 없다.

묵시적 커서의 경우 항상 가장 최근에 실행된 SQL 문장에 대한 커서를 내부적으로 갖고 있는데 이를 SQL 커서라하며 SQL 이라는 이름으로 속성에 접근 가능.

- 암시적 커서의 속성

SQL%ROWCOUNT : 해당 SQL 문에 영향을 받는 행의 수

SQL%FOUND : 해당 SQL 영향을 받는 행의 수가 1 개 이상일 경우 TRUE

SQL%NOTFOUND : 해당 SQL 문에 영향을 받는 행의 수가 없을 경우 TRUE

SQL%ISOPEN : 항상 FALSE, 암시적 커서가 열려 있는지의 여부 검색

```
-- 커서의 속성

-- %NOTFOUND : 커서의 영역의 자료가 모두 FETCH 되었다면 TRUE

-- %FOUND : 커서 영역에 FETCH 되지 않은 자료가 있다면 TRUE
-- %ISOPEN : 커서 OPEN 된 상태이면 TRUE
-- %ROWCOUNT : 커서가 얻어온 레코드의 개수



-- 커서로 테이블의 모든내용 조회하기
DECLARE
   v_deptdepartment%rowtype;
   cursor C1
  IS
  SELECT * FROM department ;
BEGIN
   dbms_output.put_line('부서번호 부서명 지역명');
   dbms_output.put_line('-------------------');
  OPEN c1;
   loop
      fetch c1 INTO v_dept.dno, v_dept.dname, v_dept.loc;
      exit WHEN c1%notfound;
      dbms_output.put_line(v_dept.dno||'     '||v_dept.dname||'     '||v_dept.loc);
  END loop;
   close c1;
END;
```

-- FOR 문을 활용한 커서

-- CUROSR FOR LOOP 는 명시적 CURSOR 에서 로우를 처리한다. CURSOR 가 내부적으로 OPEN 되고 , FETCH 되고 나서 CLOSE 되기 때문에 OPEN~FETCH~CLOS 과정없이 DECLARE 절에서 선언만하고 바로 사용할수있어 간편하다.
-- CURSOR 의 데이터를 읽어와서 저장할 변수도 따로 선언할 필요가 없다.

-- 형식
```
FOR record_name IN cursor_name LOOP

  statement1;
  statement2;
  ..
END LOOP;
```

--커서로 테이블의 모든내용 조회하기
```
DECLARE
  cursor C1
 IS
 SELECT * FROM department ;
BEGIN
  dbms_output.put_line('부서번호 부서명 지역명');
  dbms_output.put_line('--------------------');
 FOR v_dept IN c1 loop
```

```
    exit WHEN c1%notfound;
    dbms_output.put_line(v_dept.dno||'    '||v_dept.dname||'    '||v_dept.loc);
  END loop;
END;
```

--q. 사원테이블에서 커미션이 NULL이 아닌 상태의 사원번호, 이름 , 급여를 이름 기준 오름차순으로 정렬한 결과를 출력하시오.

```
SET serveroutput ON


DECLARE
  v_empemployee%rowtype;
  cursor c1 -- 커서 선언부
 IS
 SELECT * FROM employee WHERE commission IS NOT NULL ORDER BY ename
ASC;
BEGIN
```

```
  dbms_output.put_line('사번          이름          급여');
  dbms_output.put_line('-------------------------');
  OPEN c1 ;   -- 커서 오픈
  loop
    fetch c1 INTO v_emp; -- 커서 패치
    exit WHEN c1%notfound;
    dbms_output.put_line(v_emp.eno||'        '||v_emp.ename||'      '||v_emp.salary);
  END loop;
  close c1; -- 커서 닫기
END;
/









DECLARE
  cursor C1
  IS
  SELECT * FROM employee WHERE commission IS NOT NULL ORDER BY ename
ASC;
BEGIN
  dbms_output.put_line('사번          이름          급여');
  dbms_output.put_line('-------------------------');
```

```
  FOR v_emp IN c1 loop
    exit WHEN c1%notfound;
    dbms_output.put_line(v_emp.eno||'        '||v_emp.ename||'      '||v_emp.salary);
  END loop;
END;
```

ㄹ.커서(Cursor)

query 에 의해 반환되는 결과가 메모리에 위치하게 되는데 pl/sql 에서는 커서를 통하여 결과에 접근할 수 있습니다.

커서의 종류로는 묵시적 커서, 명시적 커서 이렇게 2 가지로 존재합니다.

	묵시적 커서	명시적 커서
정의	쿼리 수행시 ROW 수가 한 개만 반환하는 하는 모든 SQL 문에 대한 접근	쿼리 수행시 ROW 수가 여러 개를 반환하는 모 대한 접근
문법사용여부	오라클 내부에서 지원하므로 커서 불필요	커서 사용

- 묵시적 커서

묵시적 커서는 자동적으로 선언해주는 SQL 커서로서 사용자 입장에서 생성유무를 알 수 없으며 SELECT, DML(INSERT, DELETE, UPDATE)문이 실행 될 때마다 묵시적 커서가 실행됩니다.

-명시적 커서

명시적 커서는 사용자가 선언하여 생성 후 사용하는 커서로 주로 커서를 사용했다고 하는경우는 명시적 커서를 의미한다구 합니다.

<커서> CURSOR

- SELECT 의 결과가 2 개행 이상일 때 명시적으로 사용.

형식)

```
DECLARE
    --변수선언, 커서정의
    CURSOR 커서명 IS select 문장;
BEGIN
    OPEN 커서명;
    FETCH 커서명 INTO 변수명;
    CLOSE 커서명;
END;
/
```

?

```
--cursor_test.sql
CREATEORREPLACEPROCEDURESP_DEPT_INFO
IS
    VDEPT DEPT%ROWTYPE;
    CURSORCUR ISSELECTDEPTNO, DNAME, LOC FROMDEPT;
BEGIN
    --OPEN CUR;
    DBMS_OUTPUT.PUT_LINE('부서번호/부서명/부서위치');
    DBMS_OUTPUT.PUT_LINE('==========================');
    LOOP
        FETCHCUR INTOVDEPT; --VDEPTNO, VDNAME, VLOC
        EXIT WHENCUR%NOTFOUND; --커서에 인출된 행이 없다면

        DBMS_OUTPUT.PUT_LINE(VDEPT.DEPTNO ||' / '|| VDEPT.DNAME ||' / '||
VDEPT.LOC);
    ENDLOOP;
    CLOSECUR;
END;
/
```

?

```
--cursor_test2.sql
CREATEORREPLACEPROCEDURESP_DEPT_INFO2
IS
    VDEPT DEPT%ROWTYPE;
    CURSORCUR ISSELECTDEPTNO,DNAME,LOC FROMDEPT;
BEGIN
    DBMS_OUTPUT.PUT_LINE('부서번호/부서명/부서위치');
    DBMS_OUTPUT.PUT_LINE('==========================');
    -- FOR 변수 IN 최소값..최대값
    FORVDEPT INCUR -- 자동 OPEN, FETCH
    LOOP
```

```
12        --EXIT WHEN CUR%NOTFOUND; -- 커서에 인출된 행이 없다면
13        DBMS_OUTPUT.PUT_LINE(VDEPT.DEPTNO ||'  /  '|| VDEPT.DNAME||
14                    '  /  '||VDEPT.LOC);
15    ENDLOOP;
16END;
17/
18
19부서번호/부서명/부서위치
20============================
2110  /  ACCOUNTING  /  NEW YORK
2220  /  RESEARCH  /  DALLAS
2330  /  SALES  /  CHICAGO
2440  /  OPERATIONS  /  BOSTON
25
26PL/SQL 처리가 정상적으로 완료되었습니다.
```

문제) 특정부서(10,20,30,40)에 있는 사원의 사원번호, 사원명, 급여, 입사일을 출력하는 프로시저를 작성하시오.(SP_DEPT_MEMBER)

```
?

1  CREATEORREPLACEPROCEDURESP_DEPT_MEMBER (VDEPTNO NUMBER)
2  IS
3      VEMP EMP%ROWTYPE;
4      CURSORCUR IS(
5          SELECTEMPNO
6                , ENAME
7                , SAL
8                , HIREDATE
9            FROMEMP
10          WHEREDEPTNO = VDEPTNO
11     );
12BEGIN
13     DBMS_OUTPUT.PUT_LINE('사원번호/사원명/급여/입사일');
14     DBMS_OUTPUT.PUT_LINE('==============================');
```

```
15    FORVEMP INCUR
16    LOOP
17        DBMS_OUTPUT.PUT_LINE(VEMP.EMPNO ||' / '|| VEMP.ENAME ||' / '||
18VEMP.SAL ||' / '|| VEMP.HIREDATE);
19    ENDLOOP;
20END;
21/
22
23프로시저가 생성되었습니다.
24
25SQL> EXECUTESP_DEPT_MEMBER(10);
26사원번호/사원명/급여/입사일
27==============================
287782 / CLARK / 2450 / 81/06/09
297839 / KING / 5000 / 81/11/17
307934 / MILLER / 1300 / 82/01/23
31
32PL/SQL 처리가 정상적으로 완료되었습니다.
33
34SQL> EXECUTESP_DEPT_MEMBER(20);
35사원번호/사원명/급여/입사일
36==============================
377369 / SMITH / 800 / 80/12/17
387566 / JONES / 2975 / 81/04/02
397788 / SCOTT / 3000 / 87/04/19
407876 / ADAMS / 1100 / 87/05/23
417902 / FORD / 3000 / 81/12/03
42
43PL/SQL 처리가 정상적으로 완료되었습니다.
44
45SQL> EXECUTESP_DEPT_MEMBER(30);
46사원번호/사원명/급여/입사일
```

47==========
487499 / ALLEN / 1600 / 81/02/20
497521 / WARD / 1250 / 81/02/22
507654 / MARTIN / 1250 / 81/09/28
517698 / BLAKE / 2850 / 81/05/01
527844 / TURNER / 1500 / 81/09/08
537900 / JAMES / 950 / 81/12/03
54
55PL/SQL 처리가 정상적으로 완료되었습니다.
56
57SQL> EXECUTESP_DEPT_MEMBER(40);
58사원번호/사원명/급여/입사일
59==========
60
PL/SQL 처리가 정상적으로 완료되었습니다.

커서란, 사용자가 실행한 SQL 문의 단위를 의미한다.
또한, 오라클 서버는 모든 문장을 CURSOR 단위로 처리하고 그 정보를 저장 관리한다.

커서는 암시적 커서(IMPLICT CURSOR)와 명시적 커서(EXPLICT CURSOR) 2 가지 종류가 있다.

1. 암시적 커서
- 일반적으로 사용되는 SQL 문을 암시적 커서라고 한다. 한번 실행에 하나의 결과를 리턴하는 SQL 문이다.

2. 명시적 커서

- SQL 문을 실행했을 때 그 결과가 여러개인 경우에 암시적 커서를 사용하면 해당 SQL 문은 에러가 발생한다. 이유는 암시적 커서에 사용되는 스칼라 변수는 한번에 하나의 값만을 저장할 수 있기 때문이다.

이렇게 여러개의 행이 리턴되는질의문을 실행하는 경우에는 반드시 명시적 커서를 사용해야 한다.

구문>

```
CURSOR  cursor_name  IS
  [SELECT 문]
OPEN  cursor_name;
LOOP
   FETCH  cursor_name INTO 변수;
   EXIT  THEN  [처리내용];
END LOOP;
CLOSE  cursor_name;
```

예>

```
CREATE  OR  REPLECE  PROCEDURE  emp_process
IS
    v_empno        emp.empno%TYPE;
    v_ename        emp.ename%TYPE;
    v_sal           NUMBER(7, 2);
    CURSOR  emp_cursor  IS
        SELECT empno, ename, sal
        FROM  emp
        WHERE  deptno = 20;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH  emp_cursor  INTO  v_empno, v_ename, v_sal;
                EXIT  WHEN  emp_cursor%ROWCOUNT>  5  OR
emp_cursor%NOTFOUND;
                DBMS_OUTPUT.PUT_LINE(v_empno ||' '||v_ena'v_sal);
```

```
        END LOOP;
        CLOSE emp_cursor;
    END   emp_process;
```

3. 커서의 속성

- %ROWCOUNT : 커서의 현재 Row Count 를 리턴

```
        SELECT empno, ename, deptno
        INTO  v_empno, v_name, v_deptno
        FROM   emp
        WHERE   deptno = 10;
        DBMS_OUTPUT.PUT_LINE('검색된 행 수는 '||%ROWCOUNT||' 입니다.');
```

- %FOUND : 커서가 현재 조건을 만족하는지를 리턴

```
        SELECT empno, ename, deptno
        INTO  v_empno, v_name, v_deptno
        FROM   emp
        WHERE   deptno = 10;
        IF  %FOUND  THEN
            DBMS_OURPUT.PUT_LINE('조건을 만족하는 행이 발견되었습니다.');
        END IF;
```

- %NOTFOUND : 커서가 현재조건을 만족하지 않는지 리턴

```
        SELECT empno, ename, deptno
        INTO  v_empno, v_name, v_deptno
        FROM   emp
        WHERE   deptno = 10;
        IF  %FOUND  THEN
            DBMS_OURPUT.PUT_LINE('조건을 만족하는 행이 발견되지
않았습니다.');
        END IF;
```

- %ISOPEN : 커서가 현재 open 되어 있는지를 리턴

```
CURSOR  c1  IS SELECT  empno, ename, sal,  FROM  emp
                    WHERE  deptno = 10;
                    ...
                    ...
IF  %ISOPEN  THEN
   FETCH c1 INTO  v_empno, v_ename, v_sal
ELSE
   OPEN c1;
ENT IF;
```

4. WHERE CURRENT OF 절

- 명시적 커서를 사용하는경우, FETCH 절 내에서 DML(UPDATE, INSERT, DELETE)문을 실행할 때

현재 커서에 의해 읽혀진 행을 조작해야 하는 경우 WHERE CURRENT OF [CURSOR 명] 절을 사용하면

읽혀진 행을 모두 변경할 수 있다.

구문>

```
CURSOR  [커서명]  ←------------------------------
 IS                                                    |
    [SELECT {column_list}                              |
     FROM       table_name;                            |
 OPEN  [커서명];                                        |
 FOR                                                   |
      FETCH  [커서명]  INTO  [변수명];                  |
         UPDATE  [table_name]                          |
         SET        column=value                       |
         WHERE  CURRENT  OF  [커서명]; ---------
      EXIT  WHEN  [조건절];
 END  LOOP;
 CLOSE  [커서명];
```

예문>

```
CREATE OR REPLECE PROCEDURE emp_process
IS
    v_empno         emp.empno%TYPE;
    v_ename         emp.ename%TYPE;
    v_sal           NUMBER(7, 2);
CURSOR  emp_cursor  IS
    SELECT  empno, ename, sal
    FROM  emp
    WHERE deptno = 20
    FOR  UPDATE;
 BEGIN
 OPEN  emp_cursor;
 LOOP
    FETCH  emp_cursor  INTO  v_empno, v_ename, v_sal;
       UPDATE  emp
       SET  sal = sal * 1.1
       WHERE  CURRENT  OF  emp_cursor;

                    EXIT        WHEN        emp_cursor%ROWCOUNT>
5  OR  emp_cursor%NOTFOUND;
                DBMS_OUTPUT.PUT_LINE(v_empno||' '||v_ename||' '||v)sal);
 END LOOP;
 CLOSE  emp_cursor;
 END  emp_process;
```