

6-3. PL/SQL 제어문 익히기

1. 조건문(IF 문)
2. CASE 문과 CASE 식
3. 반복문

1. 조건문(IF 문)

- 조건문과 반복 문으로 나눌 수 있음.
- 조건 문 : IF 문 , CASE 문
- 반복 문 : Basic Loop 문 , While 문 – 반복 횟수를 알 수 없을 경우
FOR 문 – 반복 횟수를 지정할 경우

1. 조건 문 (IF 문) :

유형 1. IF~END IF 문장

IF (조 건) THEN

실행 문장;

END IF ;

1. 조건문(IF 문)

유형 1번의 예:

EMP 테이블에서 EMPNO , ENAME, DEPTNO , DNAME 을 출력하세요.

단 DNAME 의 값은 아래와 같습니다.

DEPTNO 가 10 이면 'ACCOUNT' ,

DEPTNO 가 20 이면 'RESEARCH' ,

DEPTNO 가 30 이면 'SALES' ,

DEPTNO 가 40 이면 'OPERATIONS' 로 출력하세요.

```
SQL> DECLARE
2   vempno emp.empno%TYPE;
3   vename emp.ename%TYPE;
4   vdeptno emp.deptno%TYPE;
5   vdname VARCHAR2(20) := null ;
```

다음 장에 계속...

1. 조건문(IF 문)

```
6 BEGIN
7  SELECT empno , ename , deptno
8  INTO vempno , vename , vdeptno
9  FROM emp
10 WHERE empno=7900;
11 IF (vdeptno = 10) THEN  -- IF 문이 시작됩니다.
12     vname := 'ACCOUNT';
13 END IF ; -- IF 문이 끝나면 반드시 END IF 로 마쳐야 합니다.
14 IF (vdeptno = 20) THEN
15     vname := 'RESEARCH';
16 END IF ;
17 IF (vdeptno = 30) THEN
18     vname := 'SALES';
19 END IF ;
20 IF (vdeptno = 40) THEN
21     vname := 'OPERATIONS';
22 END IF ;
23 DBMS_OUTPUT.PUT_LINE (vempno||'  '||vename||'  '||vdeptno||'  '||vname);
24 END ;
25 /
```

앞 장에서 계속....

1. 조건문(IF 문)

유형 2. IF ~ THEN ~ELSIF ~END IF 문장 (조건이 여러 개 일 경우 사용)

```
IF ( 조 건 ) THEN  
    실행 문장 ;  
ELSIF ( 조 건 ) THEN  
    실행 문장 ;  
ELSIF ( 조 건 ) THEN  
    실행 문장 ;  
END IF ;
```

예: EMP 테이블에서 EMPNO , ENAME, DEPTNO , DNAME 을 출력하세요.

단 DNAME 의 값은 아래와 같습니다.

DEPTNO 가 10 이면 'ACCOUNT' ,

DEPTNO 가 20 이면 'RESEARCH' ,

DEPTNO 가 30 이면 'SALES' ,

DEPTNO 가 40 이면 'OPERATIONS' 로 출력하세요

1. 조건문(IF 문)

```
12 IF (vdeptno = 10) THEN
13     vname := 'ACCOUNT';
14 ELSIF (vdeptno = 20) THEN
15     vname := 'RESEARCH';
16 ELSIF (vdeptno = 30) THEN
17     vname := 'SALES';
18 ELSIF (vdeptno = 40) THEN
19     vname := 'OPERATIONS';
20 END IF;
21
22 DBMS_OUTPUT.PUT_LINE (vempno||' '||vname||' '||vdeptno||' '||vname);
23 END ;
24 /
```

앞 장에서 계속....

1. 조건문(IF 문)

유형 3. IF~THEN~ELSE~END IF (조건이 2개 일 경우 사용)

```
IF ( 조 건 ) THEN  
    실행 문장 ;  
ELSE  
    실행 문장 ;  
END IF ;
```

예 : EMP 테이블에서 사원번호가 7900 번인 사원의 EMPNO , ENAME , COMM 을 출력하되 해당 사원의 COMM 이 0보다 크면 COMM 의 금액을 출력하고, 0 보다 작으면 '사원의 보너스는 없습니다' 라는 문장을 출력하세요.

1. 조건문(IF 문)

```
SQL> DECLARE
2  vempno emp.empno%TYPE;
3  vename emp.ename%TYPE;
4  vcomm emp.comm%TYPE := NULL;
5
6  BEGIN
7  SELECT empno , ename , comm INTO vempno , vename , vcomm
8  FROM emp
9  WHERE empno=7900;
10
11  IF vcomm > 0 THEN
12      DBMS_OUTPUT.PUT_LINE (vename||' 사원의 보너스는 '||vcomm||'입니
다');
13  ELSE
14      DBMS_OUTPUT.PUT_LINE (vename||' 사원의 보너스는 없습니다');
15  END IF ;
16 END ;
17 /
```


1. 조건문(IF 문)

< IF 문 연습문제 >

1. 사용자로부터 사원번호를 입력 받아 해당 사원번호의 사원의 보너스가 0 보다 클 경우 해당 사원의 보너스 금액을 출력하고 만약 보너스가 0 보다 작을 경우 '보너스가 없습니다' 라는 문장을 출력하세요(조건이 2가지 이므로 IF - THEN - ELSE - END IF 문장을 사용하세요).

2. 사 번을 입력 받아 사원(EMP)테이블에서 그 해당 사원의 이름과 연봉, 세금을 출력하세요.

(단, 세금은 연봉액수에 따라 5000 만원 이상이면 5%,
4000만원부터 4999 만원까지는 4% ,
3000만원부터 3999 만원까지는 3% ,
2000만원부터 2999 만원까지는 2% ,
나머지는 1% 로 차등 계산하세요)

조건이 여러 가지 이므로 IF - THEN - ELSIF - ELSE - END IF 문장을 사용하세요

2. CASE 문 과 CASE 식

CASE [조건]

WHEN 조건 1 **THEN** 결과 1

WHEN 조건 2 **THEN** 결과 2

...

WHEN 조건 n **THEN** 결과 n

[**ELSE** 기본값]

END ;

2. CASE 문 과 CASE 식

- CASE 문 사용 예 1:

사용자로부터 사원 번호를 입력 받은 후 EMP 테이블에서 EMPNO ,
ENAME, DEPTNO , DNAME 을 출력하세요.

단 DNAME 의 값은 아래와 같습니다.

DEPTNO 가 10 이면 'ACCOUNT' ,

DEPTNO 가 20 이면 'RESEARCH' ,

DEPTNO 가 30 이면 'SALES' ,

DEPTNO 가 40 이면 'OPERATIONS' 로 출력하세요

```
SCOTT> DECLARE
2     v_empno    emp.empno%TYPE ;
3     v_ename    emp.ename%TYPE ;
4     v_deptno   emp.deptno%TYPE ;
5     v_dname    varchar2(10) ;
```

다음 장에 계속....

2. CASE 문 과 CASE 식

```
6 BEGIN
7     SELECT empno , ename , deptno
8     INTO v_empno , v_ename , v_deptno
9     FROM emp
10    WHERE empno = &empno ;
11    v_dname := CASE v_deptno
12        WHEN 10 THEN 'ACCOUNT'
13        WHEN 20 THEN 'RESEARCH'
14        WHEN 30 THEN 'SALES'
15        WHEN 40 THEN 'OPERATIONS'
16    END ;
17    DBMS_OUTPUT.PUT_LINE (v_empno||' '||v_ename||' '||v_deptno||'
'||v_dname);
18 END ;
19 /
```

앞 장에서 계속....

Enter value for empno: 7900
7900 JAMES 30 SALES

```
11 v_dname := CASE
12     WHEN v_deptno = 10 THEN 'ACCOUNT'
13     WHEN v_deptno = 20 THEN 'RESEARCH'
14     WHEN v_deptno = 30 THEN 'SALES'
15     WHEN v_deptno = 40 THEN 'OPERATIONS'
16     END ;
```

PL/SQL procedure successfully completed.

2. CASE 문 과 CASE 식

- CASE 문 사용 예 2:

EMP 테이블을 사용하여 사용자로부터 사원 번호를 입력 받아 해당 사원의 empno , ename , sal , deptno, 인상후 연봉(up_sal) 을 출력하세요.

단 부서번호가 10 번 부서는 현재 연봉의 10% 인상하고,
부서번호가 20 번 , 30 번 부서는 20 % 인상하고 ,
부서번호가 30 번 보다 클 경우 30% 를 인상하세요.

```
SCOTT> DECLARE
2      v_empno    emp.empno%TYPE ;
3      v_ename    emp.ename%TYPE ;
4      v_sal      emp.sal%TYPE ;
5      v_deptno   emp.deptno%TYPE ;
6      v_up_sal   emp.sal%TYPE ;
```

다음 장에 계속....

2. CASE 문 과 CASE 식

앞 장에서 계속....

```
7 BEGIN
8     SELECT empno , ename , sal , deptno
9     INTO v_empno , v_ename , v_sal , v_deptno
10    FROM emp
11   WHERE empno = &empno ;
12   v_up_sal := CASE
13       WHEN v_deptno = 10 THEN v_sal * 1.1
14       WHEN v_deptno IN (20,30) THEN v_sal*1.2
15       WHEN v_deptno > 30 THEN v_sal*1.3
16       ELSE v_sal
17   END ;
18   DBMS_OUTPUT.PUT_LINE ('사원번호'|| '||'이름'|| '||'급여'|| '||'부서번호'|| ' '
||'인상후급여');
19   DBMS_OUTPUT.PUT_LINE (v_empno||'    '||v_ename||'    '||v_sal||'    '||v_deptno||'
'||v_up_sal);
20 END ;
21 /
```

Enter value for empno: 7900

사원번호	이름	급여	부서번호	인상후급여
7900	JAMES	950	30	1140

3. 반복문

반복문은 반복 횟수를 알 수 없는 경우에 사용하는 BASIC LOOP 문과 WHILE 문이 있고 반복횟수를 지정하는 FOR 문이 있습니다

1) BASIC LOOP 반복문

LOOP

PL/SQL 문장 ;

PL/SQL 문장 ;

EXIT [조건] ;

END LOOP ;

문장을 먼저 수행 후 조건을 확인함

3. 반복문

BASIC LOOP 문 사용 예 1 :

Loop 문을 사용하여 화면에 0 부터 5 까지의 숫자를 아래와 같이 출력하세요.

출력
예시

0
1
2
3
4
5

```
SCOTT>DECLARE
2      num number := 0 ;
3  BEGIN
4      LOOP
5          DBMS_OUTPUT.PUT_LINE(num) ;
6          num := num + 1 ;
7          EXIT WHEN num > 5;
8      END LOOP ;
9  END ;
10 /
```

PL/SQL procedure successfully completed.

3. 반복문

2) WHILE 반복문

While 조건 LOOP

PL/SQL 문장

PL/SQL 문장

END LOOP ;

조건을 먼저 확인 후 문장을 수행함

BASIC LOOP 문은 나중에 조건을 검색하지만 WHILE 문은 시작부터 조건을 먼저 검사한 후 PL/SQL 문장을 수행하게 됩니다. 즉 BASIC LOOP 문은 조건이 틀려도 PL/SQL 문장이 1회는 실행이 되지만 WHILE 문은 아예 실행이 되지 않습니다.

3. 반복문

WHILE 반복문 사용 예 1:

WHILE 반복문을 사용하여 아래의 출력 예시와 같이 출력하세요.

출력
예시

0
1
2
3
4
5

```
SCOTT> DECLARE
2   num number := 0 ;
3 BEGIN
4   WHILE num < 6 LOOP
5     DBMS_OUTPUT.PUT_LINE(num) ;
6     num := num + 1 ;
7   END LOOP ;
8 END ;
9 /
```

3. 반복문

< BASIC LOOP 문과 WHILE 반복문 연습 문제 >

BASIC LOOP 문과 WHILE 문을 사용하여 아래의 결과처럼 나오도록 각각 작성하세요.

출력 결과 화면

5
4
3
2
1
0

3. 반복문

BASIC LOOP 문 사용 시

```
SCOTT>DECLARE
2      num number := 5 ;
3 BEGIN
4      LOOP
5          DBMS_OUTPUT.PUT_LINE(num) ;
6          num := num - 1 ;
7          EXIT WHEN num = - 1;
8      END LOOP ;
9 END ;
10 /
```

WHILE 문 사용 시

```
SCOTT>DECLARE
2      num number := 5 ;
3 BEGIN
4      WHILE num >= 0 LOOP
5          DBMS_OUTPUT.PUT_LINE(num) ;
6          num := num - 1 ;
7      END LOOP ;
8 END ;
9 /
```

3. 반복문

3) FOR 반복문 – 반복 횟수를 지정 가능함

```
FOR counter IN [REVERSE] start .. end LOOP  
    Statement1 ;  
    Statement2 ;  
    ...  
END LOOP ;
```

위 문법에서 counter 는 반복을 카운트 할 변수입니다. 원래 PL/SQL 에서는 사용할 변수를 사용하기 전에 미리 DECLARE 부분에서 선언해야 하지만 FOR 반복문 내에서 사용할 변수는 미리 선언을 하지 않아도 사용 가능합니다.

그리고 IN 다음에 start 부분에 시작번호를 쓰고 .. (점 두 개) 뒤에 end 부분에 끝 숫자를 적으시면 됩니다.

만약 역순으로 반복하려면 IN 다음에 REVERSE 를 쓰고 반복할 횟수를 쓰면 되는데 주의 사항은 REVERSE 를 쓰고 시작번호와 끝 번호는 작은 숫자부터 써야 한다는 것입니다.

3. 반복문

- FOR 반복문 예 1:

0 – 5 까지 숫자를 화면에 출력하세요.

출력
예시

0
1
2
3
4
5

```
SCOTT>BEGIN
2  FOR i IN 0..5 LOOP
3    DBMS_OUTPUT.PUT_LINE(i) ;
4  END LOOP ;
5 END ;
6 /
```

3. 반복문

- FOR 반복문 예 2:

0 – 5 까지의 숫자를 역순으로 출력하세요.

출력
예시

5
4
3
2
1
0

```
SCOTT>BEGIN
2  FOR i IN REVERSE 0..5 LOOP
3    DBMS_OUTPUT.PUT_LINE(i) ;
4  END LOOP ;
5  END ;
6  /
```

3. 반복문

사원번호를 입력 받아 사원 이름의 글자수 만큼 화면에 * 를 찍는 문장을 작성하세요

출력 예

Enter value for eno: 7900

*

**

```
SQL> DECLARE
  2  v_empno    emp.empno%TYPE := &eno;
  3  v_ename    emp.ename%TYPE;
  4  v_number   NUMBER;
  5  v_char     VARCHAR2(100);
  6 BEGIN
  7  SELECT  ename, length(ename) INTO v_ename, v_number
  8  FROM emp
  9  WHERE empno = v_empno;
 10  FOR i IN 1..v_number LOOP
 11    v_char := v_char || '*';
 12    DBMS_OUTPUT.PUT_LINE(v_char);
 13  END LOOP;
 14 END;
 15 /
```


3. 반복문

4) CONTINUE 문 (11g 에서 추가된 기능입니다)

```
SCOTT> DECLARE
2      total  number := 0;
3  BEGIN
4      FOR i in 1..10 LOOP
5          total := total +1 ;
6          DBMS_OUTPUT.PUT_LINE('Total :'||total) ;
7          CONTINUE WHEN i > 5 ;
8          total := total + i ;
9          DBMS_OUTPUT.PUT_LINE('Out total:' || total );
10     END LOOP ;
11     END ;
12     /
```

```
Total :1
Out total:2
Total :3
Out total:5
Total :6
Out total:9
Total :10
Out total:14
Total :15
Out total:20
Total :21
Total :22
Total :23
Total :24
Total :25
```

3. 반복문

1. total 변수를 0 으로 초기화합니다. (2번 라인)
2. total 변수에 $0 + 1$ 의 식으로 1을 할당합니다 (5번 라인)
3. 1 을 출력합니다 (6번 라인 , 결과화면의 첫 번째 출력결과)
4. CONTINUE 문의 조건을 확인합니다 (i 가 5보다 큰가?)
5. 아니라고 판단되면 total 변수에 있는 값에 i 의 값을 더합니다 (8번 라인)
6. 더해진 값을 Out total 이라는 내용과 함께 화면에 출력합니다 (9번 라인)
기존 total 에 1이 있었고 i 변수에도 1이므로 화면에 2가 출력됩니다.
7. 다시 5번 라인으로 돌아가서 total 변수 값 $2+1$ 을 하고 화면에 3을 출력합니다 (6번라인)
- 8 . CONTINUE 문의 조건을 다시 확인한 후 (7번라인) total 변수값 $3 + 2$ (i 변수값) 을 계산해
화면에 5를 출력합니다 (출력화면 4번째 줄)
9. 위와 같은 방식으로 계속 루프를 돌다가 i 값이 5가 되면 CONTINUE 조건에 걸리게 되므로
8번, 9번 줄의 연산을 하지 않게 됩니다.
그래서 출력 화면을 보면 Total 값만 나오고 Out total 값은 나오지 않게 됩니다.