

제 7 장

PL/SQL



목차

7.1 PL/SQL의 구성

7.2 PL/SQL 생성 및 실행

7.3 변수

7.4 선택처리문

7.5 반복처리문

7.6 PL/SQL에서 SQL

7.1 PL/SQL의 구성

- PL/SQL은 SQL에 절차적 언어의 요소를 더한 기능
- 절차적 언어 요소
 - 선택처리(IF)
 - 반복처리(LOOP, WHILE, FOR)
- 향상된 데이터 처리 능력을 부여함
- 3장에서 실습했던 학사 DB

7.1 PL/SQL의 구성

DECLARE	선언부 <i>모든 변수, 상수, 커서 등을 선언하는 부분</i>
BEGIN	실행부 <i>SQL에 선택처리, 반복처리 등을 포함한 실제 로직이 표현되는 부분</i>
EXCEPTION	예외처리부 <i>실행부 로직 처리 중 에러 또는 비정상적인 상황을 처리하기 위한 작업을 기술하는 부분</i>

7.1 PL/SQL의 종류

- **PROCEDURE** : 프로시저 처리 후 반환 값의 제한이 없음
- **FUNCTION** : 함수 처리 후 반환 하나의 값 반환
- **PACKAGE** : 프로시저나 함수의 그룹
- **TRIGGER** : 조건에 맞는 이벤트가 일어나면 후속 이벤트가 발생

7.2 PL/SQL의 생성 및 실행

➤ 과목 테이블(subject)에 한 과목 삽입

```
create procedure test1
(v_sub_name in subject.sub_name%type,
 v_sub_prof in subject.sub_prof%type,
 v_sub_grade in subject.sub_grade%type,
 v_sub_dept in subject.sub_dept%type)
is
  v_sub_no subject.sub_no%type;
begin
  select max(sub_no)
    into v_sub_no
    from subject;
  v_sub_no := to_number(v_sub_no)+1;

  insert
    into subject
    values(v_sub_no,v_sub_name,v_sub_prof,v_sub_grade,
          v_sub_dept);
  commit;
end test1;
/
```

7.2 PL/SQL의 생성 및 실행

➤ 과목 테이블(subject)에 한 과목 삽입

```
SQL> @test1.sql
```

```
SQL> select * from subject;
```

```
SQL> execute test1( '컴퓨터구조' , '강주봉' ,2, '컴퓨터정보' );
```

```
SQL> select * from subject;
```

7.3 변수

- 매개변수, 자체변수

- IS로 구분됨

```
variable_name [mode] data_type;
```

- IS 다음의 자체변수는 mode 생략

- 매개변수는 역할에 따라

- IN : 프로저서가 호출될 때 전달되는 값을 처리하기 위한 변수
- OUT : 프로시저가 처리된 후 반환되는 값을 처리하기 위한 변수
- INOUT : 두가지 기능을 갖는 변수

7.3 변수 타입

- 스칼라(scalar) 타입

- NUMBER, CHAR, VARCHAR2, DATE, BOOLEAN 등

- 참조(references) 타입을 많이 사용함

- %TYPE

```
v sub no enrol.sub no%type  
v stu no enrol.stu no%type  
v cour grade enrol.cour grade%type
```

- %ROWTYPE

```
v_enrol enrol%rowtype
```

7.4 선택처리문(IF ~ ENDIF)

- if ~~ end if

SQL> set serveroutput on

SQL> declare

```
2  v_condition number := 1;  
3  begin  
4  if v_condition = 1 then  
5      dbms_output.put_line( '대한' );  
6  end if;  
7  end;  
8  /
```

7.4 선택처리문(IF ~ ELSE ~ ENDIF)

- if ~~ else ~~ end if

SQL> set serveroutput on

SQL> declare

```
2  v_condition number := 1;
3  begin
4  if v_condition = 1 then
5      dbms_output.put_line( '대한' );
6  else
7      dbms_output.put_line( '우리' );
8  end if;
9  end;
10 /
```

7.4 선택처리문(IF ~ ELSIF ~ ELSE ~ ENDIF)

- if ~~ elsif ~~ else ~~ end if

SQL> set serveroutput on

SQL> declare

```
2  v_condition number := 2;
3  begin
4  if v_condition < 1 then
5      dbms_output.put_line( '대한' );
6  elsif v_condition = 1 then
7      dbms_output.put_line( '민국' );
8  else
9      dbms_output.put_line( '만세' );
10 end if;
11 end;
12 /
```

7.5 반복처리문

- LOOP ~ END LOOP

```
SQL> declare
  2  v_cnt  number := 0;
  3  v_sum  number := 0;
  4  begin
  5      loop
  6          v_cnt := v_cnt + 1;
  7          v_sum := v_sum + v_cnt;
  8          dbms_output.put_line(v_cnt|| " " ||v_sum);
  9          exit when v_cnt = 10;
 10      end loop;
 11  end;
 12  /
```

7.5 반복처리문

- FOR ~ END LOOP

```
SQL> declare
  2  v_cnt number;
  3  v_sum number := 0;
  4  begin
  5    for v_cnt in 1..10 loop
  6      v_sum := v_sum + v_cnt;
  7      dbms_output.put_line(v_cnt||'    '||v_sum);
  8    end loop;
  9 end;
10 /
```

7.5 반복처리문

- WHILE ~ END LOOP

```
SQL> declare
  2   v_cnt number := 0;
  3   v_sum number := 0;
  4   begin
  5       while v_cnt < 10 loop
  6           v_cnt := v_cnt + 1;
  7           v_sum := v_sum + v_cnt;
  8           dbms_output.put_line(v_cnt||'    '||v_sum);
  9       end loop;
 10   end;
 11   /
```