



PL/SQL 개요 및 출력

PL/SQL 기본적으로 프로그래밍 언어처럼 쓰겠다는 이야기다.
그렇다면 당연히 메소드 같은 블록처리가 기본이다.

Anonymous Block (익명 블록)

Procedure (프로시저)

Function (함수)

1. 프로시저에 출력하기 (자체소)
2. DECLARE 변수 사용을 위한 구역
3. SQL Plus 에서 확인하기
4. 파일 직접실행

PL/SQL 장점

- **SQL과의 긴밀한 통합**

- PL/SQL을 사용하면 모든 SQL 데이터 조작, 커서 제어 및 트랜잭션 제어 문과 모든 SQL 함수, 연산자 및 의사 열을 사용할 수 있습니다.

- **고성능**

- PL/SQL을 사용하면 명령문 블록을 데이터베이스로 보낼 수 있으므로 애플리케이션과 데이터베이스 간의 트래픽을 크게 줄일 수 있습니다.

<< 레이블 >> (선택 사항)

DECLARE -- 선언 부분(선택 사항)

-- 지역 유형, 변수 및 하위 프로그램의 선언

```
BEGIN -- 실행 부분(필수)
    -- 명령문(선언 부분에서 선언된 항목을 사용할 수 있음)

[EXCEPTION -- 예외 처리 부분(선택)
    -- 실행 부분에서 발생한 예외(오류)에 대한 예외 처리기]
END;
```

PL/SQL 기본적으로 프로그래밍 언어처럼 쓰겠다는 이야기다.
그렇다면 당연히 메소드 같은 블록처리가 기본이다.

Anonymous Block (익명 블록)

이름이 없는 블록을 의미 하며, 실행하기 위해 프로그램 안에서 선언 되고 실행시에 실행을 위해 PL/SQL 엔진으로 전달 된다.

선행 컴파일러 프로그램과 SQL*Plus 또는 서버 관리자에서 익명의 블록을 내장 할 수 있다.

Procedure (프로시저)

특정 작업을 수행할수 있는 이름이 있는 PL/SQL 블록으로서, 매개 변수를 받을수 있고, 반복적으로 사용할수 있다.

보통 연속 실행 또는 구현이 복잡한 트랜잭션을 수행하는 PL/SQL블록을 데이터베이스에 저장하기 위해 생성 한다.

Function (함수)

보통 값을 계산하고 결과값을 반환하기 위해서 함수를 많이 사용 한다.

대부분 구성이 프로시저와 유사하지만 IN 파라미터만 사용 할 수 있고, 반드시 반환 될 값의 데이터 타입을 RETURN문에 선언해야 한다.

또한 PL/SQL블록 내에서 RETURN문을 통해서 반드시 값을 반환 해야 한다.

1. 프로시저에 출력하기 (자체소)

```
BEGIN
    dbms_output.put_line('my frist block');
END;

BEGIN
    dbms_output.put_line('my frist block');
    dbms_output.put_line('this is the second lint ');
END;

DECLARE    ##변수선언을 구역을 넣고 하여도 문제없
BEGIN
    dbms_output.put_line('hello world');
END;
```

2. DECLARE 변수 사용을 위한 구역

```
DECLARE    ##변수선언구역에서는 선언만 해주단 초기값을 세팅X
    v number ;    #변수선언
BEGIN
    v:=5; #변수 초기값
    dbms_output.put_line('hello world');
    dbms_output.put_line(v);
END;
```

3. SQL Plus 에서 확인하기

```
##sqlplus 에 접속후

BEGIN
dbms_output.put_line('my frist block');
END;
/
```

```
set serveroutput: on  
/
```

4. 파일 직접실행

```
@C:\Users\ket36\test.sql;
```

```
cd C:\Users\ket36;  
@test.sql;
```