

저장 프로시저 (Stored Procedure)

오라클은 사용자가 만든 PL/SQL 문을 데이터베이스에 저장할 수 있도록 저장 프로시저라는 것을 제공하고 있음.

자주 사용되는 복잡한 DML(INSERT/UPDATE/DELETE 문), DQL(SELECT 문)을 모듈화 시켜서 DB 에 저장하였다가 필요할 때 호출해서 사용하는 객체임.

작성한 PL/SQL 을 저장해 놓고, 여러 번 사용할 수 있도록 할 수 있음.

저장 프로시저를 사용하면 성능도 향상되고, 호환성 문제도 해결됨.

장점

- 여러 클라이언트가 업무 규칙을 공유할 수 있음 : 여러 클라이언트가 서버에 저장해 둔 SQL 문을 함께 공유할 수 있음.
- 저장 프로시저는 SQL 문을 서버에서 미리 구문 검사가 끝난 상태에서 대기하고 있다가, 호출만 되면 바로 실행하므로 속도가 빠름.

■ 프로시저 작성 및 실행 방법

```
CREATE [OR REPLACE] PROCEDURE 프로시저이름 (매개변수명 1 [IN | OUT|INOUT]
자료형, 매개변수명 2 [MODE] 자료형, .....)
```

```
--IN : 데이터를 전달 받을 때
```

```
--OUT : 수행된 결과를 받아갈 때
```

```
--INOUT : 두 가지 목적에 모두 사용
```

```
IS
```

```
    지역변수 선언;
```

```
BEGIN
```

```
    실행할 문장 1;
```

```
    실행할 문장 2;
```

```
    실행할 문장 3;
```

```
-- 함수호출, 초기화, 절차적 요소, SQL 문
```

```
END;
```

```
/
```

실행방법)

```
SQL> EXECUTE 프로시저이름[(전달값 1, 전달값 2, .....)];
```

제거>

```
DROP PROCEDURE 프로시저이름;
```

예> EMP01 테이블의 데이터를 모두 삭제하는 프로시저를 생성하시오.

프로시저 이름 : DEL_ALL

```
CREATE OR REPLACE PROCEDURE DEL_ALL
```

```
IS
```

```
BEGIN
```

```
    DELETE FROM EMP01;
```

```
    COMMIT;
```

```
END;
```

```
/
```

실행>

```
EXECUTE DEL_ALL;
```

확인>

```
SELECT * FROM EMP01;
```

■ 저장 프로시저 조회하기

프로시저 객체는 데이터 디렉터리 USER_SOURCE 에 저장됨.

```
DESC USER_SOURCE;
```

```
--NAME 컬럼 : 저장 프로시저 이름
```

```
--TEXT 컬럼 : 프로시저 소스 저장
```

```
SELECT NAME, TEXT FROM USER_SOURCE;
```

■ 저장 프로시저 매개변수

프로시저 실행시 자바의 메소드처럼 해당 프로시저로 값을 전달할 수도 있음.

예제> 매개변수가 있는 저장 프로시저 만들기

```
CREATE OR REPLACE PROCEDURE
    DEL_ENAME(VENAME IN EMP01.ENAME%TYPE)
IS
BEGIN
    DELETE FROM EMP01 WHERE ENAME LIKE VENAME;
    COMMIT;
END;
/
```

실행>

```
EXECUTE DEL_ENAME('이%');
```

■ IN 모드, OUT 모드, INOUT 모드 매개변수

● IN 모드 매개변수

프로시저 실행시 전달된 값을 받아서, 프로시저 안에서 그 값을 사용하도록 하는 변수

```
EXECUTE 프로시저이름(전달값);
```

```
CREATE OR REPLACE PROCEDURE 프로시저이름(변수명 IN 자료형)
```

● OUT 모드 매개변수

프로시저에서 구한 결과값을 받기 위한 변수 지정시 적용함.

프로시저 실행시 값 받을 변수명 앞에 ':'(콜론) 붙임

값 받을 바인드변수 선언>

```
SQL> VARIABLE 변수명 자료형(바이트);
```

```
EXECUTE 프로시저이름(전달값, :값 받을 변수명);
```

```
CREATE OR REPLACE PROCEDURE 프로시저이름(변수명 OUT 자료형)
```

```
--바인드변수와 매개변수의 자료형은 반드시 같아야 함.
```

받은 값 출력확인>

```
SQL> PRINT 변수명;
```

예제> 매개 변수가 있는 저장 프로시저 만들기

```
CREATE OR REPLACE PROCEDURE SEL_EMPNO ( VEMPNO IN EMP.EMPNO%TYPE,  
                                           VENAME OUT EMP.ENAME%TYPE,  
                                           VSAL OUT EMP.SAL%TYPE,  
                                           VJOB OUT EMP.JOB%TYPE)
```

```
IS
```

```
BEGIN
```

```
    SELECT ENAME, SAL, JOB  
    INTO VENAME, VSAL, VJOB  
    FROM EMP  
    WHERE EMPNO = VEMPNO;
```

```
END;
```

```
/
```

바인드변수 선언하기>

```
VARIABLE VAR_ENAME VARCHAR2(15);
```

```
VARIABLE VAR_SAL NUMBER;
```

```
VARIABLE VAR_JOB VARCHAR2(10);
```

저장 프로시저 실행>

```
EXECUTE SEL_EMPNO(7499, :VAR_ENAME, :VAR_SAL, :VAR_JOB);
```

바인드변수 출력하기>

```
PRINT VAR_ENAME;
```

```
PRINT VAR_SAL;
```

```
PRINT VAR_JOB;
```

● INOUT 모드 매개변수

IN 과 OUT 이 동시에 적용되는 모드임.

실습> 부서 테이블(DEPT02)에서 부서번호를 입력 받아 매개변수로 넘겨 해당 부서를 삭제하는 프로시저를 만드시오.

DEPT 테이블의 내용을 복사한 DEPT02 테이블을 만드시오.

프로시저 이름 : DEL_DEPTNO

```
CREATE OR REPLACE PROCEDURE DEL_DEPTNO(VDEPTNO IN NUMBER)
IS
BEGIN
    DELETE
    FROM DEPT02
    WHERE DEPTNO = VDEPTNO;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(VDEPTNO || ' 부서를 삭제하였습니다!!');
END;
/
```

```
SQL> EXECUTE SP_DEL_DEPTNO(&DEPTNO);
```

실습> EMP 테이블을 복제한 EMP02 테이블을 만들고 사원의 이름을 입력받아 그 사원의 정보를 삭제하는 프로시저를 생성하시오.

프로시저 이름 : DEL_ENAME(매개변수 있음)

```
CREATE OR REPLACE PROCEDURE SP_DEL_ENAME (P_NAME IN VARCHAR2)
IS
BEGIN
    DELETE
    FROM EMP02
    WHERE ENAME = P_NAME;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('# ' || P_NAME || '이 퇴사하였습니다');
END;
/

EXECUTE DEL_ENAME(&ENAME);
```