



PROCEDURE

PL/SQL문을 저장하는 객체이다. 필요할 때마다 복잡한 구문을 다시 입력 할 필요 없이 간단하게 호출해서 실행결과를 얻을 수 있다.

CREATE TABLE EMP_DUP
AS SELECT * FROM EMPLOYEE;



SELECT * FROM EMP_DUP;

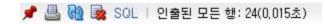
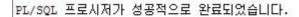


Table EMP_DUP() (가) 생성되었습니다.

CREATE OR REPLACE
PROCEDURE DEL_ALL_EMP
IS
BEGIN
DELETE FROM EMP_DUP;
COMMIT;
END;
/

Procedure DEL_ALL_EMPO (가) 컴파일되었습니다.

EXEC DEL_ALL_EMP;





SELECT * FROM EMP_DUP;







PROCEDURE

DESC USER_SOURCE;

 SELECT * FROM USER_SOURCE;

	NAME	↑ TYPE	♦ LINE	♦ TEXT
1	DEL_ALL_EMP	PROCEDURE	1	PROCEDURE DEL_ALL_EMP
2	DEL_ALL_EMP	PROCEDURE	2	IS
3	DEL_ALL_EMP	PROCEDURE	3	BEGIN
4	DEL_ALL_EMP	PROCEDURE	4	DELETE FROM EMP_DUP;
5	DEL_ALL_EMP	PROCEDURE	5	
6	DEL_ALL_EMP	PROCEDURE	6	COMMIT;
7	DEL ALL EMP	PROCEDURE	7	END;



PROCEDURE

[매개변수 있는 프로시져]

```
CREATE OR REPLACE PROCEDURE DEL_EMP_ID

(V_EMP_ID EMPLOYEE.EMP_ID%TYPE)

IS

BEGIN

DELETE FROM EMPLOYEE WHERE EMP_ID = V_EMP_ID;

COMMIT;

END;

/
```

Procedure DEL EMP IDOI(가) 컴파일되었습니다.

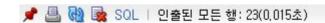
EXECUTE DEL_EMP_ID('&EMP_ID');



이전:EXECUTE DEL_EMP_ID('&EMP_ID') 신규:EXECUTE DEL_EMP_ID('900')

PL/SQL 프로시저가 성공적으로 완료되었습니다.

SELECT * FROM EMPLOYEE;





PROCEDURE

[IN/OUT 매개변수 있는 프로시져]

```
CREATE OR REPLACE PROCEDURE SELECT_EMPID(
    V_EMP_ID IN EMPLOYEE.EMP_NO%TYPE,
    V_EMP_NAME OUT EMPLOYEE.EMP_NAME%TYPE,
    V_SALARY OUT EMPLOYEE.SALARY%TYPE,
    V_BONUS OUT EMPLOYEE.BONUS%TYPE
)
IS
BEGIN
    SELECT EMP_NAME, SALARY, NVL(BONUS, 0)
    INTO V_EMP_NAME, V_SALARY, V_BONUS
    FROM EMPLOYEE
    WHERE EMP_ID = V_EMP_ID;
END;
/
```

Procedure SELECT EMPIDOI(가) 컴파일되었습니다.

```
VARIABLE VAR_EMP_NAME VARCHAR2(30);
VARIABLE VAR_SALARY NUMBER;
VARIABLE VAR_BONUS NUMBER;
```

** 바인드변수 선언





PROCEDURE

EXEC SELECT EMPID(200,: VAR EMP NAME,: VAR SALARY,: VAR BONUS);

PL/SQL 프로시저가 성공적으로 완료되었습니다.

PRINT VAR_EMP_NAME; PRINT VAR_SALARY; PRINT VAR_BONUS; VAR_EMP_NAME
--선동일
VAR_SALARY
----8000000
VAR_BONUS

** 프로시져 실행과 동시에 모든 바인딩 변수를 출력하기 위해서는 SET AUTOPRINT ON; 을 실행시킨다.



FUNCTION

프로시져와 거의 유사한 용도로 사용하지만, 실행 결과를 되돌려 받을 수 있는 점에서 프로시져와 다르다.

```
CREATE OR REPLACE FUNCTION
BONUS CALC(V EMPID EMPLOYEE.EMP ID%TYPE)
RETURN NUMBER
V SAL EMPLOYEE.SALARY%TYPE;
V BONUS EMPLOYEE.BONUS%TYPE;
 CALC SAL NUMBER;
BEGIN
 SELECT SALARY, NVL(BONUS, 0)
 INTO V SAL, V_BONUS
 FROM EMPLOYEE
 WHERE EMP ID = V EMPID;
 CALC SAL := (V SAL + (V SAL * V BONUS)) * 12;
 RETURN CALC SAL;
END;
```

VARIABLE VAR_CALC NUMBER;

EXEC :VAR_CALC :=
 BONUS_CALC('&EMP_ID');



PL/SQL 프로시저가 성공적으로 완료되었습니다.

CALC_SAL ------

Function BONUS_CALCO(가) 컴파일되었습니다.



FUNCTION

SELECT EMP_ID, EMP_NAME, BONUS_CALC(EMP_ID) FROM EMPLOYEE;



		⊕ EMP_NAME	BONUS_CALC(EMP_ID)
1	200	선동일	124800000
2	201	송종기	72000000
3	217	전지연	57096000
4	204	유재식	48960000
5	209	심봉선	48300000
6	205	정중하	46800000
7	215	대북혼	45120000
8	202	노옹철	44400000
9	216	차태연	40032000
10	222	이태림	39467088
11	212	장쯔위	38250000
12	218	이오리	34680000
13	203	송은희	33600000
14	213	하동운	30624000
15	208	김해술	30000000



CURSOR

처리 결과가 여러 개의 행으로 구해지는 SELECT문을 처리하기 위해 처리 결과를 저장해 놓은 객체이다. CURSOR~ OPEN~ FETCH~ CLOSE 단계로 진행된다.

CURSOR의 상태

속 성	설 명			
%NOTFOUND	커서 영역의 자료가 모두 FETCH되어 다음 영역이 존재하지 않으면 TRUE			
%FOUND	커서 영역에 아직 FETCH되지 않은 자료가 있으면 TRUE			
%ISOPEN	커서가 OPEN된 상태이면 TRUE			
%ROWCOUNT	커서가 얻어 온 레코드의 갯수			



CURSOR

```
CREATE OR REPLACE PROCEDURE CURSOR DEPT
 V DEPT DEPARTMENT%ROWTYPE;
                                                PL/SOL 프로시저가 성공적으로 완료되었습니다.
 CURSOR C1
                                                부서코드 : D1 , 부서명 : 인사관리부 , 지역 : L1
 SFLECT * FROM DEPARTMENT:
                                                부서코드 : D2 , 부서명 : 회계관리부 , 지역 : L1
BEGIN
                                                부서코드 : D3 , 부서명 : 마케팅부 , 지역 : L1
 OPEN C1;
                                                부서코드 : D4 , 부서명 : 국내영업부 , 지역 : L1
 IOOP
                                                부서코드 : D5 , 부서명 : 해외영업1부 , 지역 : L2
  FETCH C1 INTO V DEPT. DEPT ID.
                                                부서코드 : D6 , 부서명 : 해외영업2부 , 지역 : L3
                                                부서코드 : D7 , 부서명 : 해외영업3부 , 지역 : L4
                 V DEPT.DEPT TITLE.
                                                부서코드 : D8 , 부서명 : 기술지원부 , 지역 : L5
                 V DEPT.LOCATION ID;
                                                부서코드 : D9 , 부서명 : 총무부 , 지역 : L1
   EXIT WHEN C1%NOTEOUND:
  DBMS OUTPUT.PUT LINE('부서코드:' | V DEPT.DEPT ID
                         ||', 부서명:'|| V_DEPT.DEPT_TITLE
                         II'. 지역:'IIV DEPT.LOCATION_ID);
 FND LOOP:
 CLOSE C1;
END;
```



CURSOR

```
PL/SOL 프로시저가 성공적으로 완료되었습니다.
CREATE OR REPLACE PROCEDURE CURSOR DEPT
                                                      부서코드 : D1 , 부서명 : 인사관리부 , 지역 : L1
IS
                                                      부서코드 : D2 , 부서명 : 회계관리부 , 지역 : L1
 V DEPT DEPARTMENT%ROWTYPE;
                                                      부서코드 : D3 , 부서명 : 마케팅부 , 지역 : L1
 CURSOR C1
                                                      부서코드 : D4 , 부서명 : 국내영업부 , 지역 : L1
                                                      부서코드 : D5 , 부서명 : 해외영업1부 , 지역 : L2
                                                      부서코드 : D6 , 부서명 : 해외영업2부 , 지역 : L3
 SFLECT * FROM DEPARTMENT:
                                                      부서코드 : D7 , 부서명 : 해외영업3부 , 지역 : L4
BEGIN
                                                      부서코드 : D8 , 부서명 : 기술지원부 , 지역 : L5
 FOR V DEPT IN C1 LOOP
                                                      부서코드 : D9 , 부서명 : 총무부 , 지역 : L1
   DBMS OUTPUT.PUT LINE('부서코드:' | V DEPT.DEPT ID
                          || '. 부서명 : '|| V_DEPT.DEPT_TITLE
                          || ', 지역 : ' || V_DEPT.LOCATION_ID);
 FND LOOP:
END;
```

** FOR IN LOOP를 이용하면 LOOP 반복시 자동으로 CURSOR를 OPEN하고 행을 인출(FETCH)한다. LOOP 종료시 자동으로 CURSOR를 CLOSE한다.



CURSOR

```
CREATE OR REPLACE PROCEDURE CURSOR_DEPT IS

V_DEPT DEPARTMENT%ROWTYPE;
BEGIN

FOR V_DEPT IN (SELECT * FROM DEPARTMENT) LOOP

DBMS_OUTPUT.PUT_LINE('부서코드:'|| V_DEPT.DEPT_ID

||', 부서명:'|| V_DEPT.DEPT_TITLE

||', 지역:'|| V_DEPT.LOCATION_ID);
END LOOP;
END;
/
```

** FOR IN 문을 사용하면 커서의 선언도 생략할 수 있다.

PL/SQL 프로시저가 성공적으로 완료되었습니다.

```
부서코드 : D1 , 부서명 : 인사관리부 , 지역 : L1
부서코드 : D2 , 부서명 : 회계관리부 , 지역 : L1
부서코드 : D3 , 부서명 : 마케팅부 , 지역 : L1
부서코드 : D4 , 부서명 : 국내영업부 , 지역 : L1
부서코드 : D5 , 부서명 : 해외영업1부 , 지역 : L2
부서코드 : D6 , 부서명 : 해외영업2부 , 지역 : L3
부서코드 : D7 , 부서명 : 해외영업3부 , 지역 : L4
부서코드 : D8 , 부서명 : 기술지원부 , 지역 : L5
부서코드 : D9 , 부서명 : 총무부 , 지역 : L1
```