```
drop table cust_info;


CREATE TABLE CUST_INFO(
 ID VARCHAR2(13) NOT NULL PRIMARY KEY ,
 FIRST_NM VARCHAR2(10) ,
 LAST_NM VARCHAR2(10) ,
 ANNL_PERF NUMBER(10,2) ,
 SALARY  NUMBER(6)
);

INSERT INTO CUST_INFO VALUES ('100' , '허' , '재혁', 330.08 ,5000);
INSERT INTO CUST_INFO VALUES ('200' , '표' , '준태', 857.61,5500);
INSERT INTO CUST_INFO VALUES ('110' , '조' , '은경', -76.77,5800);
INSERT INTO CUST_INFO VALUES ('220' , '이' , '정훈', 468.54,5900);
INSERT INTO CUST_INFO VALUES ('400' , '이' , '윤정', -890,5700);
INSERT INTO CUST_INFO VALUES ('300' , '이' , '윤', 47.44,5500);
COMMIT;
```

PL/SQL
프로시저와 함수

프로시저 (Procedure)

:넓은 의미는 어떤 업무를 처리하기 위한 절차
 결과값 반환없이 특정 로직을 처리
 질의의 집합으로 어떤 동작을 일괄 처리
 테이블에서 데이터 추출 및 조작 결과를 다른 테이블에 저장하거나 갱신

 프로시저 문법

```
CREATE OR REPLACE PROCEDURE   프로시저 이름
(   매개변수 이름 1 [ IN | OUT | IN OUT ] 데이터 타입,
    매개변수 이름 2 [ IN | OUT | IN OUT ] 데이터 타입, … )
IS | AS
      변수 및 상수 선언
BEGIN
     실행 문장
     EXCEPTION 문장
END;
```

프로시저실행

```
EXECUTE 프로시저 이름();
EXEC 프로시저 이름();
```

PL/SQL변수 종류

• 일반 변수

```
count NUMBER();
emp_name VARCHAR2(10);
```

• 상수: CONSTANT 키워드 사용 (변경 불가)

```
count CONSTANT NUMBER();
emp_name CONSTANT VARCHAR2(10);
```

• %TYPE: 테이블 열 1개의 데이터 형식에 접근

```
emp_name EMPLOYEES.EMPLOYEE_ID%TYPE
emp_email EMPLOYEES.EMAIL%TYPE
```

테이블의 특정컬럼과 같은자료형으로 선언하겠다.

• %ROWTYPE: 테이블 전체 열의 데이터 형식에 접근

```
emp EMPLOYEES%ROWTYPE
dept DEPARTMENTS%ROWTYPE
```

• 레코드(Record): 여러 개의 열의 데이터 형식을 지정

```
TYPE user_type IS RECORD (name VARCHAR2(10), email VARCHAR2(20));
user user_type;
```

```
## 직원 출력 프로시저
CREATE OR REPLACE PROCEDURE first_emp AS
        emp_name VARCHAR2(20);
BEGIN
        SELECT first_nm || ' ' || last_nm INTO emp_name
        FROM CUST_INFO WHERE ID =  '8301111567897' ;
        DBMS_OUTPUT.PUT_LINE(emp_name);
END;
```

서버 출력 허용

```
SET SERVEROUTPUT ON;
```

프로시저 실행

```
EXECUTE first_emp();
```

```
#평균수익율 구하기
CREATE OR REPLACE PROCEDURE emp_avg_ANNL_PERF (
        avg_salary OUT NUMBER
) AS
BEGIN
        SELECT AVG(ANNL_PERF) INTO avg_salary
        FROM CUST_INFO ;
END ;
```

```
DECLARE
        avg_ANNL_PERF NUMBER;
BEGIN
        emp_avg_ANNL_PERF (avg_ANNL_PERF );
        DBMS_OUTPUT.PUT_LINE(avg_ANNL_PERF );
END;


#주언진 연봉이 평균이상 인지 판별
CREATE OR REPLACE PROCEDURE if_salary (
        salary IN NUMBER
) AS
        avg_salary NUMBER;
BEGIN
        SELECT AVG(salary) INTO avg_salary FROM cust_info;
        IF salary >= avg_salary THEN
                DBMS_OUTPUT.PUT_LINE('평균 이상');
        ELSE
                DBMS_OUTPUT.PUT_LINE('평균 미만');
        END IF;
END;

//// in , out 변수 사용해 보기
#직원조회하기
CREATE OR REPLACE PROCEDURE out_emp (
        aid IN cust_info.id%TYPE,
        out_str OUT VARCHAR2
) AS
        emp_name VARCHAR2(20);
BEGIN
        SELECT first_nm || ' ' || last_nm INTO emp_name
        FROM cust_info WHERE id = aid;
        out_str := '직원: ' || emp_name;
END;


DECLARE
        out_str VARCHAR2(30);
BEGIN
        out_emp( '100', out_str);
        DBMS_OUTPUT.PUT_LINE(out_str);
END;
```

IN, OUT 변수선언하기, 생략하면 IN , IN OUT 같이 사용할 수 있다

```
out_str := '직원'     => out_str변수에 직원값을 할당
a='직원' 과 같은 의미입니다. ! 어렵지않아요
```

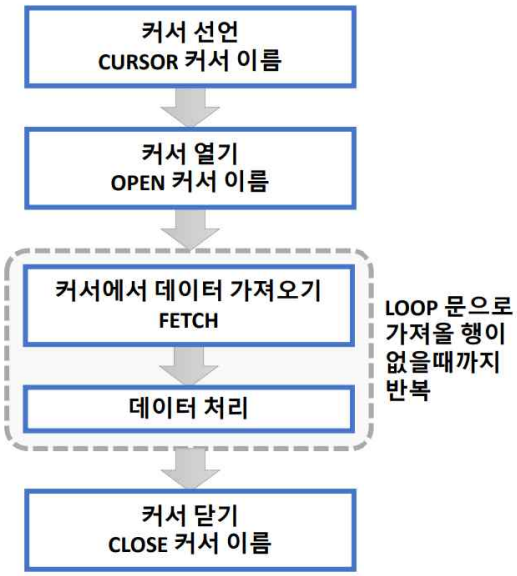
```
## 직원조회하기 ,예외처리하기
CREATE OR REPLACE PROCEDURE out_emp (
        emp_id IN cust_info.id%TYPE,
        out_str OUT VARCHAR2
) AS
        emp_name VARCHAR2(20);
BEGIN
        SELECT first_nm || ' ' || last_nm INTO emp_name
        FROM cust_info WHERE id = emp_id;
        out_str := '직원: ' || emp_name;
EXCEPTION
        WHEN NO_DATA_FOUND THEN
        out_str := '직원: 없음';
END;


DECLARE
        out_str VARCHAR2(30);
BEGIN
        out_emp(300, out_str);
        DBMS_OUTPUT.PUT_LINE(out_str);
END;
```

커서

- 커서(Cursor)
- 일반 프로그래밍 언어의 파일 처리 방법과 유사
- 행의 집합을 다룰 수 있는 편리한 기능 제공
- 테이블에서 여러 개의 행을 질의 후, 질의 결과인 행 집합을 한 행씩 처리
- 프로시저 내부에서 커서 사용
- 커서 처리 순서

- 커서(Cursor)
  - 일반 프로그래밍 언어의 파일 처리 방법과 유사
  - 행의 집합을 다룰 수 있는 편리한 기능 제공
  - 테이블에서 여러 개의 행을 질의 후, 질의 결과인 행 집합을
  - 프로시저 내부에서 커서 사용

- 커서 처리 순서



cust_info 테이블로부터 salary를 조회한 결과를 하나씩 읽어서 총샐러리와 카운트를 구한다음 평균을 구한것임
(커서 어렵지 않아요 ! 조회결과집합을 하나씩 읽어서 처리하는 것입니다 !)

```
CREATE OR REPLACE PROCEDURE cursor_salary AS
        sal NUMBER := 0;                    (변수선언과 초기값 할당   int a=0 과 동일함)
        cnt NUMBER := 0;
        total NUMBER := 0;
        CURSOR emp_cursor IS SELECT salary FROM cust_info;
BEGIN
        OPEN emp_cursor;
                LOOP
                        FETCH emp_cursor INTO sal;
                        EXIT WHEN emp_cursor%NOTFOUND;
                        total := total + sal;
                        cnt := cnt + 1;
                END LOOP;
        CLOSE emp_cursor;
        DBMS_OUTPUT.PUT_LINE('평균 SALARY: ' || (total / cnt));
END;
```

```
// 커서실행
EXECUTE cursor_salary();
```

함수(Function)
오라클에 지원되는 함수들이 있습니다. 사용자가 직접 함수를 만들 수 있는 사용자정의 함수라고 생각하시면 됩니다

- 프로시저의 각 프로세스를 수행하기 위해 필요한 기능
- 일반적인 프로그래밍 언어에서 사용되는 함수와 같이 복잡한 프로그래밍지원

함수 문법

```
CREATE OR REPLACE FUNCTION 함수 이름
( 매개변수 이름 1 데이터 타입,
매개변수 이름 2 데이터 타입, … )
RETURN 데이터 타입
IS | AS
변수 및 상수 선언
BEGIN
실행 문장
RETURN 반환값
EXCEPTION 문장
END;
```

• 프로시저와 함수 차이

프로시저 함수
• 특정 작업 수행 • 특정 계산 수행
• 리턴값이 없을수도 있음 • 리턴값이 반드시 존재해야 함
• 리턴값을 여러개 가질 수 있음 • 리턴값을 하나만 가질 수 있음
• 서버(DB)에서 기술 • 클라이언트에서 기술
• 수식내에서 사용 불가 • 수식내에서만 사용 가능
• 단독으로 문장 구성 가능 • 단독으로 문장 구성 불가

```
 #  나이 구하는  함수만들기
CREATE OR REPLACE FUNCTION get_age(date Date)
        RETURN NUMBER
IS
        age NUMBER;
BEGIN
        age := TRUNC(MONTHS_BETWEEN(TRUNC(SYSDATE), to_yyyymmdd(date))/ 12);
RETURN age;
END;

CREATE OR REPLACE FUNCTION to_yyyymmdd( date Date)
        RETURN VARCHAR2
IS
        char_date VARCHAR2(20);
BEGIN
        char_date := TO_CHAR(date, 'YYYYMMDD' );
END;


SELECT get_age('20010101') FROM dual;


 # 1~10 출력하는 반복문 수행하는 프로시저
CREATE OR REPLACE PROCEDURE while_print AS
        str VARCHAR(100);
        i NUMBER;
BEGIN
        i := 1;
        WHILE (i <= 10) LOOP
                str := '반복 횟수:' || '(' || i || ')';
                DBMS_OUTPUT.PUT_LINE(str);
                i := i + 1;
        END LOOP;
END;


EXECUTE while_print();
```

For반복문 사용하기

```
CREATE OR REPLACE PROCEDURE for_print AS
BEGIN
  FOR i IN 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE('반복 횟수:(' || i || ')');
  END LOOP;
END;


EXECUTE for_print  ;
```

#문제

For문을 이용하여 2 단 출력하기

For문을 이용하여 전체구구단 출력하기

#문제

Coursor를 이용해 cust_info 테이블에서 id가 '300' 인 직원의 first name와 last name을 결합한 출력을 하는 프로시저정의