

# PL/SQL 수업 목차

2018년 4월 27일 금요일    오전 10:07

- 1. PL/SQL 을 배워야하는 이유?
- 2. PL/SQL의 기본 Block구조
- 3. PL/SQL 내에서의 DML문
- 4. PL/SQL 내에서의 IF 문
- 5. PL/SQL 내에서의 Loop 문
  - 5-1. Basic Loop 문
  - 5-2 while loop 문
  - 5-3 for loop문
- 6. PL/SQL 내에서의 커서문
- 7. 예외 처리
- 8. 사용자 정의 함수
- 9. 프로시저

# 1. PL/SQL 을 배워야하는 이유?

2018년 4월 27일 금요일 오전 9:53

## 1. 데이터 분석을 하기 위해서는 SQL은 필수로 알고 있어야 한다.

SQL로 반복해서 해야하는 작업을 PL/SQL로 단순하게 처리할 수 있다.

딥 러닝 데이터 분류를 하기 위해서 데이터를 저장할 때 큰 규모의 회사는 데이터를 전부 오라클에 저장한다.

그런데 SQL 만으로 데이터를 분석하고 처리하려면 노가다를 많이 해야 한다.

노가다를 줄이면서 성능을 높일 수 있다.

## 2. R로 데이터 분석을 하기 위해서 엑셀로 DB의 데이터를 내려서 분석하는게 아니라,

바로 DB에 PL/SQL을 이용해서 분석을 할 수 있다.(회귀 분석, 의사 결정 트리)

## 3. SQL 튜닝방법으로 PL/SQL을 사용할 수 있다.

문제1. 부서번호, 부서번호별 토달월급을 가로로 출력 하시오.(sum함수와 decode를 이용해서)

```
SELECT SUM(DECODE(deptno,10,sal,0)) AS "10",  
       SUM(DECODE(deptno,20,sal,0)) AS "20",  
       SUM(DECODE(deptno,30,sal,0)) AS "30"  
FROM EMP;
```

	10	20	30
1	8750	10875	9400

문제2. 아래의 결과를 출력 하시오.

	JOB	10	20	30
1	SALESMAN	0	0	5600
2	CLERK	1300	1900	950
3	PRESIDENT	5000	0	0
4	MANAGER	2450	2975	2850
5	ANALYST	0	6000	0

```
SELECT job, SUM(DECODE(deptno,10,sal,0)) AS "10",  
       SUM(DECODE(deptno,20,sal,0)) AS "20",  
       SUM(DECODE(deptno,30,sal,0)) AS "30"  
FROM EMP  
GROUP BY job;
```

문제3. 문제2번의 결과를 PL/SQL로 수행 하시오.

```
create or replace procedure  
get_data(p_x out sys_refcursor)
```

```

as
  l_query varchar2(400) := 'select deptno '
begin
  for x in (select distinct job from emp order by 1)
  loop
    l_query := l_query || replace(' , sum(decode(job, '$X', sal)) as $X '
    , '$X', x.job );
  end loop;
  l_query := l_query || ' from emp group by deptno '
  open p_x for l_query;
end;
/

```

```
variable x refcursor;
```

```
exec get_data(:x);
```

```
print x;
```

을 cmd에서 수행한다.

```

프로시저가 생성되었습니다.

SQL>
SQL>
SQL>
SQL> variable x refcursor;
SQL>
SQL>
SQL>
SQL> exec get_data(:x);

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL>
SQL>
SQL>
SQL> print x;

```

DEPTNO	ANALYST	CLERK	MANAGER	PRESIDENT	SALESMAN
30		950	2850		5600
20	6000	1900	2975		
10		1300	2450	5000	

문제4. 통신사, 나이, 통신사별 나이별 인원수를 출력하시오.

```

SELECT age, sum(DECODE(telecom,'sk',1,0)) AS "sk",
       sum(DECODE(telecom,'kt',1,0)) AS "kt",
       sum(DECODE(telecom,'lg',1,0)) AS "lg",
       sum(DECODE(telecom,'cjh',1,0)) AS "cjh"
FROM EMP2
GROUP BY age;

```

	AGE	sk	kt	lg	cjh
1	25	2	2	1	0
2	28	2	3	2	1

<input type="checkbox"/>	AGE	sk	kt	lg	cjh
1	25	2	2	1	0
2	28	2	3	2	1
3	29	0	0	1	0
4	26	1	2	2	0
5	24	1	2	0	0
6	32	0	0	1	0
7	27	2	3	2	0

## 2. PL/SQL의 기본 Block구조

2018년 4월 27일 금요일 오전 10:59

**Declare** : 선언절 ( 변수, 상수, 예외 등을 선언한다.)

**Begin** : 실행절 ( 실행 할 SQL문, IF문, Loop문을 실행 한다.)

**End;** : 종료절

**/** <-- PL/SQL을 종료하겠다.

문제5. 사원번호를 물어보게하고 사원번호를 입력하면 해당 사원의 월급이 출력되도록 PL/SQL을 작성하시오.

```
SQL> ed p5.sql
```

Cmd에서 ed p5.sql을 출력후에 메모장에

```
set serveroutput on      // on 으로 켜야지 아래의 Dbms_output 패키지가 작동한다
```

```
accept p_empno prompt '사원번호를 입력 하세요'      // p_empno 변수(빈칸) 에 ' '의 메시지를 야기시켜라
```

```
declare                // v_sal 변수(빈칸)은 숫자 10자리만 들어올 수 있게한다.
  v_sal number(10);
```

```
begin
  select sal into v_sal  // sal 을 v_sal에 넣겠다
  from emp
  where empno = &p_empno; // & : 치환 변수 : 반드시 써줘야함
```

```
dbms_output.put_line('해당 사원의 월급은' || v_sal);      // ()안에 결과를 출력하는 패키지
```

```
end;
```

```
/
```

입력하고 저장 후에 종료하고 @p5.sql을 입력 하면 된다.

```
사원번호를 입력 하세요7788
구   7:      where empno = &p_empno;
신   7:      where empno = 7788;
해당 사원의 월급은3000
```

7788이 P\_empno에 들어간다.

**P와 v의 의미**

Declare 와 end; 내부에 있는건 v, 외부에 있는건 p 이여서 외부변수를 내부에서 쓸려면 &(치환변수들) 꼭 붙여줘야한다.

문제6. 사원번호를 물어보게하고 사원번호를 입력하면 해당사원의 직업이 출력되게 PL/SQL을 작성하시오.

```
set serveroutput on
```

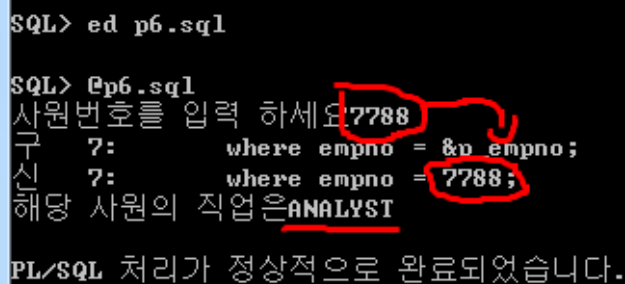
```
accept p_empno prompt '사원번호를 입력 하세요'
```

```
declare  
  v_job varchar2(10);
```

```
begin  
  select job into v_job  
    from emp  
   where empno = &p_empno;
```

```
  dbms_output.put_line('해당 사원의 직업은' || v_job);
```

```
end;  
/
```



```
SQL> ed p6.sql  
  
SQL> @p6.sql  
사원번호를 입력 하세요7788  
구   7:      where empno = &p_empno;  
신   7:      where empno = 7788;  
해당 사원의 직업은ANALYST  
  
PL/SQL 처리가 정상적으로 완료되었습니다.
```

문제7. 사원이름을 물어보게하고, 사원이름을 입력하면 해당 사원의 직업이 출력되게하는  
PL/SQL을 작성하시오.

```
set serveroutput on
```

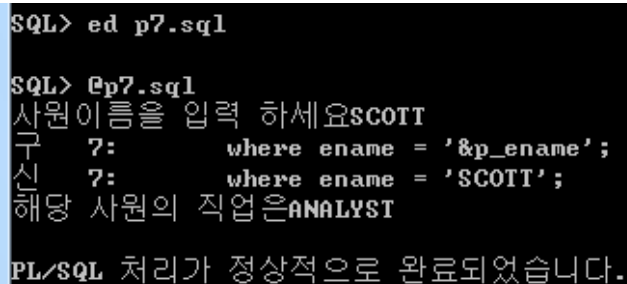
```
accept p_ename prompt '사원이름을 입력 하세요'
```

```
declare  
  v_job varchar2(20);
```

```
begin  
  select job into v_job  
    from emp  
   where ename = '&p_ename';
```

```
  dbms_output.put_line('해당 사원의 직업은' || v_job);
```

```
end;  
/
```



```
SQL> ed p7.sql  
  
SQL> @p7.sql  
사원이름을 입력 하세요SCOTT  
구   7:      where ename = '&p_ename';  
신   7:      where ename = 'SCOTT';  
해당 사원의 직업은ANALYST  
  
PL/SQL 처리가 정상적으로 완료되었습니다.
```

문제8. (점심시간)문제7번에서 소문자를 입력하던 대문자를 입력하던 상관없이 출력되게 하시오.

```

set serveroutput on

accept p_ename prompt '사원이름을 입력 하세요'

declare
    v_job varchar2(20);

begin
    select job into v_job
    from emp
    where ename = upper('&p_ename');

    dbms_output.put_line('해당 사원의 직업은' || v_job);

end;
/

```

```

SQL> ed p8.sql

SQL> @p8.sql
사원이름을 입력 하세요scott
구   7:      where ename = upper('&p_ename');
신   7:      where ename = upper('scott');
해당 사원의 직업은ANALYST

PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제9. 문제8번을 다시 수행하는데 결과가 출력될때에 아래의 메시지가 안나오게 하시오.

```

SQL> ed p8.sql

SQL> @p8.sql
사원이름을 입력 하세요scott
구   7:      where ename = upper('&p_ename');
신   7:      where ename = upper('scott');
해당 사원의 직업은ANALYST

PL/SQL 처리가 정상적으로 완료되었습니다.

```

답)

코드 맨위에 set verify off를 쓴다.

```

SQL> @p9.sql
사원이름을 입력 하세요scott
해당 사원의 직업은ANALYST

PL/SQL 처리가 정상적으로 완료되었습니다.

```

%% PL/SQL의 정보

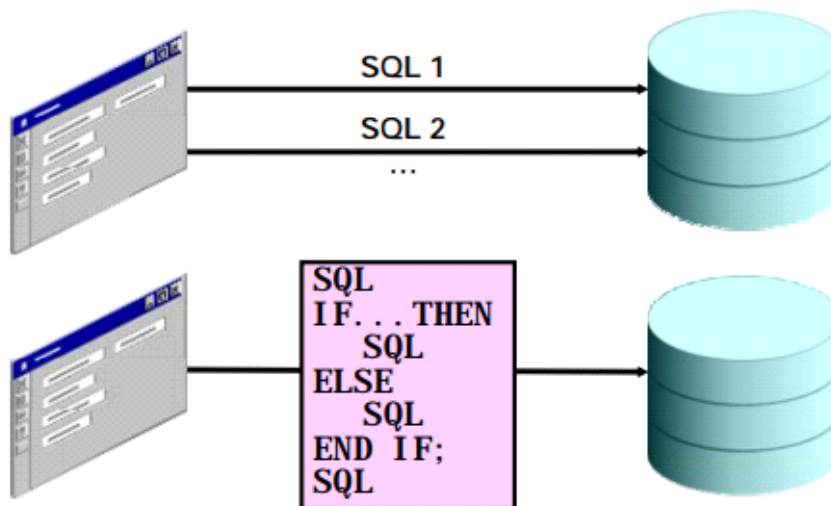
## PL/SQL:

- "SQL을 확장한 절차적 언어(Procedural Language)"를 나타냅니다.
- 관계형 데이터베이스에서 사용되는 Oracle의 표준 데이터 액세스 언어입니다.
- 프로시저 생성자를 SQL과 완벽하게 통합합니다.



## %%PL/SQL의 이점

- 프로시저 생성자와 SQL의 통합
- 성능 향상



문제10. 이름이 SCOTT인 사원의 이름과 부서위치를 출력하시오.

```
Select e.ename, d.loc
from emp e, dept d
where e.deptno = d.deptno
and ename = 'SCOTT';
```

문제11. 이름을 물어보게하고 이름을 입력하면 해당 사원의 부서위치가 출력되게하는 PLSQL을 작성하시오.

```
set verify off
set serveroutput on      <-- 한번 해놓으면 끝때까지 안써줘도됨

accept p_ename prompt '사원이름을 입력 하세요'

declare
v_loc varchar2(20);

begin
select d.loc into v_loc
```



```

from emp e, dept d
where e.deptno = d.deptno
and ename = upper('&p_ename');

dbms_output.put_line('해당 사원의 부서위치는은' || v_loc);

end;
/

```

```

SQL> @p11.sql
사원이름을 입력 하세요scott
해당 사원의 부서위치는은DALLAS
PL/SQL 처리가 정상적으로 완료되었습니다.

```

**문제12. 이름, 월급, 급여등급(grade)를 출력하시오.**

```

Select e.ename, e.sal, s.grade
from emp e, salgrade s
where e.sal between s.losal and s.hisal;

```

**문제13. 이름을 물어보게하고 이름을 입력하면 해당 사원의 월급과 급여등급이 아래와 같이 출력되게 하시오.**

```

SQL> @p13.sql
사원이름을 입력 하세요scott
해당 사원의 월급은3000이고 등급은4입니다.
PL/SQL 처리가 정상적으로 완료되었습니다.

```

accept p\_ename prompt '사원이름을 입력 하세요'

```

declare
v_sal number(10);
v_grade number(5);

```

```

begin
select e.sal, s.grade into v_sal, v_grade
from emp e, salgrade s
where e.sal between s.losal and s.hisal
and ename = upper('&p_ename');

```

```

dbms_output.put_line('해당 사원의 월급은' || v_sal || '이고 ' || '등급은' || v_grade || '입니다.');
```

```

end;
/

```

**문제14. 이름, 월급, 순위를 출력하는데 순위가 월급이 높은 순서대로 출력하시오.**

```

SELECT *
FROM (
SELECT ename, sal, dense_RANK()OVER(ORDER BY sal desc) 순위
FROM EMP
)
ORDER BY 순위 ASC;

```

**문제15. 이름을 물어보게 하고 순위를 입력하면 해당 사원의 월급의 순위가 출력되게 PL/SQL하시오.**

```

set serveroutput on

```

```
accept p_ename prompt '이름을 입력하세요 '
```

```
declare
```

```
    v_rnk varchar2(20);
```

```
begin
```

```
    select rnk into v_rnk
```

```
        from ( select ename, sal, dense_rank () over (order by sal desc) rnk
                from emp )
```

```
        where ename = upper('&p_ename');
```

```
    dbms_output.put_line('해당 사원의 월급 순위는 ' || v_rnk || ' 입니다.');
```

```
end;
```

```
/
```

**문제16.** 두개의 숫자를 각각 물어보게하고 두개의 숫자를 각각 입력하면 두개의 숫자의 합이 출력되게 하시오.

```
accept p_num1 prompt '첫 번째 숫자를 입력하세요.'
```

```
accept p_num2 prompt '두 번째 숫자를 입력하세요.'
```

```
declare
```

```
    v_sum number(20);
```

```
begin
```

```
    v_sum := &p_num1 + &p_num2;    := 할당 연산자
```

```
    dbms_output.put_line('총합은 : ' || v_sum);
```

```
end;
```

```
/
```

```
SQL> @p16.sql
첫 번째 숫자를 입력하세요.12
두 번째 숫자를 입력하세요.1
총합은 : 13
PL/SQL 처리가 정상적으로 완료되었습니다.
```

**문제17.** 수학자 가우스가 초등학교때 알아낸 방법을 사용해서 1부터 10까지 다 더하면 몇인지 출력하시오.

```
accept p_num1 prompt '첫 번째 숫자를 입력하세요.'
```

```
accept p_num2 prompt '두 번째 숫자를 입력하세요.'
```

```
declare
```

```
    v_sum number(20);
```

```

begin
  v_sum := (&p_num1 + &p_num2) * (&p_num2 / 2);

  dbms_output.put_line(&p_num1 || '부터' || &p_num2 || '까지 합은' || v_sum);

end;
/

```

```

SQL> @p17.sql
첫 번째 숫자를 입력하세요.1
두 번째 숫자를 입력하세요.10
1부터10까지 합은55
PL/SQL 처리가 정상적으로 완료되었습니다.

```

**문제18.** 첫번째 숫자를 물어보게 하고 두번째 숫자를 물어보게 해서 가우스가 알아낸 방법으로 숫자의 합을 출력하도록 PLSQL을 작성하시오.

```

accept p_num1 prompt '첫 번째 숫자를 입력하세요.'
accept p_num2 prompt '두 번째 숫자를 입력하세요.'

```

```

declare
  v_sum number(20);

begin
  v_sum := (&p_num1 + &p_num2) * (&p_num2 / 2);

  dbms_output.put_line(&p_num1 || '부터' || &p_num2 || '까지 합은' || v_sum);

end;
/

```

```

SQL> @p17.sql
첫 번째 숫자를 입력하세요.1
두 번째 숫자를 입력하세요.10
1부터10까지 합은55
PL/SQL 처리가 정상적으로 완료되었습니다.

```

**문제19.** log(2,10)이 무엇인지 SQL로 출력하시오.

```

SELECT log(2,10)
FROM dual;

```

**문제20.** 밑수를 물어보게하고 진수를 물어보게해서 그 로그값이 무엇인지 출력하는 PLSQL을 작성하시오.  
(소수점 4자리까지)

```

accept p_num1 prompt '밑수를 입력하세요.'
accept p_num2 prompt '진수를 입력하세요.'

```

```

declare
  v_log number(20,4);

begin
  v_log := log(&p_num1,&p_num2);

  dbms_output.put_line('입력한 로그 값은 : ' || v_log);

```

```
end;
```

```
/
```

```
SQL> @p20.sql  
밑수를 입력하세요.2  
진수를 입력하세요.5  
입력한 로그 값은2.3219
```

문제21. 수학책의 예제에는  $\frac{1}{3}\log 125$  식을 PLSQL로 구현하시오.

실수를 입력하세요 1/3

밑수를 입력하세요 2

진수를 입력하세요 125

```
accept p_num1 prompt '실수를 입력하세요.'
```

```
accept p_num2 prompt '밑수를 입력하세요.'
```

```
accept p_num3 prompt '진수를 입력하세요.'
```

```
declare
```

```
  v_log number(20,4);
```

```
begin
```

```
  v_log := &p_num1*log(&p_num2,&p_num3);
```

```
  dbms_output.put_line('입력한 로그 값은 : ' || v_log);
```

```
end;
```

```
/
```

문제22. 수학책의 문제4번을 PLSQL로 아래와 같이 구현하시오.

실수를 입력하시오 : 1

실수를 입력하시오 : log(3,7)

실수를 입력하시오 : log(9,100)

이 결과를 작은수 부터 나열하시오.

```
accept p_num1 prompt '실수를 입력하세요.'
```

```
accept p_num2 prompt '실수를 입력하세요.'
```

```
accept p_num3 prompt '실수를 입력하세요.'
```

```
declare
```

```
  v_ee1 number(10,4);
```

```
  v_ee2 number(10,4);
```

```
  v_ee3 number(10,4);
```

```
begin
```

```
  v_ee1 := &p_num1;
```

```
  v_ee2 := &p_num2;
```

```
  v_ee3 := &p_num3;
```

```
if v_ee1 < v_ee2 then
```

```
  if v_ee2 < v_ee3 then
```

```
    dbms_output.put_line(v_ee1 || ',' || v_ee2 || ',' || v_ee3);
```

```
  end if;
```

```
end if;
```

```

if v_ee1 < v_ee3 then
  if v_ee3 < v_ee2 then
    dbms_output.put_line(v_ee1 || ',' || v_ee3 || ',' || v_ee2);
  end if;
end if;

if v_ee2 < v_ee3 then
  if v_ee3 < v_ee1 then
    dbms_output.put_line(v_ee2 || ',' || v_ee3 || ',' || v_ee1);
  end if;
end if;

if v_ee2 < v_ee1 then
  if v_ee1 < v_ee3 then
    dbms_output.put_line(v_ee2 || ',' || v_ee1 || ',' || v_ee3);
  end if;
end if;

if v_ee3 < v_ee1 then
  if v_ee1 < v_ee2 then
    dbms_output.put_line(v_ee3 || ',' || v_ee1 || ',' || v_ee2);
  end if;
end if;

if v_ee3 < v_ee2 then
  if v_ee2 < v_ee1 then
    dbms_output.put_line(v_ee3 || ',' || v_ee2 || ',' || v_ee1);
  end if;
end if;

end;
/

```

```

SQL> @p22.sql
실수를 입력하세요.1
실수를 입력하세요.log(3,7)
실수를 입력하세요.log(9,100)
1, 1.7712, 2.0959

PL/SQL 처리가 정상적으로 완료되었습니다.

```

### 3. PL/SQL 내에서의 DML문

2018년 4월 30일 월요일 오전 10:12

- 데이터 전처리를 위한 PLSQL 사용

Source table								Target table		
7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000	(null)	10	번호	부서번호	토탈월급
7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	(null)	30	1	10	8,750
7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	(null)	10	2	20	10,875
7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	(null)	20	3	30	9,400
7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	1400	30	4	40	0
7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600	300	30			
7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500	0	30			
7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950	(null)	30			
7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	1250	500	30			
7902	FORD	ANALYST	7566	1981-12-11 오전 12:00:00	3000	(null)	20			
7369	SMITH	CLERK	7902	1980-12-09 오전 12:00:00	800	(null)	20			
7788	SCOTT	ANALYST	7566	1982-12-22 오전 12:00:00	3000	(null)	20			
7876	ADAMS	CLERK	7788	1983-01-15 오전 12:00:00	1100	(null)	20			
7934	MILLER	CLERK	7782	1982-01-11 오전 12:00:00	1300	(null)	10			

==> source table의 데이터를 가공해서 target table로 데이터를 이행

- DML문의 종류 4가지

1. Insert
2. Update
3. Delete
4. merge

문제23. 부서번호, 부서번호별 토탈 월급을 담은 아래의 구조의 테이블을 생성 하시오.

번호	부서번호	토탈월급
1	10	8,750
2	20	10,875
3	30	9,400
4	40	0

```
CREATE TABLE dept_sumsal
( no number(10),
  deptno number(10),
  sumsal NUMBER(20,2) );
```

문제24. 아래의 테이블의 10번 부서번호의 토탈월급을 아래와 같이 번호와 함께 입력되게 insert문장을 작성하시오.(서브쿼리를 사용한 insert문)

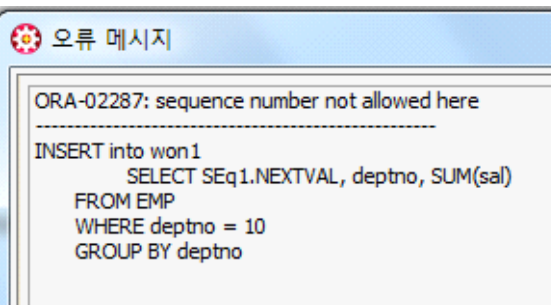
번호	부서번호	토탈월급
1	10	8,750

```
CREATE SEQUENCE seq1
START WITH 1
INCREMENT BY 1;
```

```
INSERT into dept_sumsal
SELECT seq1.NEXTVAL, deptno, SUM(sal)
FROM EMP
WHERE deptno = 10
GROUP BY deptno;
```

```
CREATE SEQUENCE seq1
START WITH 1
INCREMENT BY 1;
```

```
INSERT into won1
SELECT seq1.NEXTVAL, deptno, SUM(sal)
FROM EMP
WHERE deptno = 10
GROUP BY deptno;
```



시퀀스 번호는 이 위치에 사용할 수 없다.

문제25. 문제 24번을 PLSQL로 구현하시오.

```
set verify off
set serveroutput on

accept p_deptno prompt '번호를 입력하세요'

declare
v_deptno number(10);
v_sumsal number(10);

begin

select deptno, sum(sal)
into v_deptno, v_sumsal
from emp
where deptno = &p_deptno
group by deptno;

insert into dept_sumsal
values(seq1.nextval, v_deptno, v_sumsal);

commit;

end;
/
```

```
SQL> @p25.sql
번호를 입력하세요10
PL/SQL 처리가 정상적으로 완료되었습니다.
```

Select \* from dept\_sumsal;

<input type="checkbox"/>	NO	DEPTNO	SUMSAL
1	2	10	8750

이렇게하면 no가 2부터 들어가지는데,

DROP TABLE dept\_sumsal; 을 하지말고

delete FROM DEPT\_SUMSAL;

DROP SEQUENCE seq1;

이 두개만 하고 다시 시퀀스만들고 하면

<input type="checkbox"/>	NO	DEPTNO	SUMSAL
1	1	10	8750

1부터 나오기는 하는데 이유는 ? .....

**문제26. 문제25번의 코드에 암시적 커서를 사용해서 PLSQL 내에서 데이터를 입력할 때 입력된 건수가 화면에 출력되게 하시오.**

accept p\_deptno prompt '번호를 입력하세요'

declare

v\_deptno number(10);

v\_sumsal number(10);

begin

```
select deptno, sum(sal)
  into v_deptno, v_sumsal
 from emp
 where deptno = &p_deptno
 group by deptno;
```

```
insert into dept_sumsal
 values(seq1.nextval, v_deptno, v_sumsal);
```

```
dbms_output.put_line(SQL%rowcount || '행이 입력되었습니다.');
```

commit;

end;

/

```
SQL> @p26.sql
번호를 입력하세요20
1행이 입력되었습니다.
PL/SQL 처리가 정상적으로 완료되었습니다.
```



<input type="checkbox"/>	NO	DEPTNO	SUMSAL
1	1	10	8750
2	2	20	10875

-- SQL%rowcount : 암시적 커서로 앞의 DML문의 처리된 건수를 출력할 때 사용한다.

문제27. 통신사, 통신사별 학생 이름을 가로로 출력하시오.(SQL문제)

```
SELECT telecom,
       listagg(ename,',') within group(ORDER BY ename)
FROM EMP2
GROUP BY telecom;
```

<input type="checkbox"/>	TELECOM	LISTAGG(ENAME,',')WITHINGROUP(ORDERBYENAMI
1	cjh	이근호
2	kt	김근아,김영토,김지우,방송준,백광훈,송윤호,신
3	lg	김광록,도웅,유혜린,은해찬,이광훈,이상민,이천
4	sk	김건태,김대경,김동윤,김원섭,윤동환,윤진민,이

문제28. 통신사, 통신사별 학생이름, 통신사별 인원수가 출력되게 하시오.

```
SELECT telecom,
       listagg(ename,',') within group(ORDER BY ename), COUNT(*)
FROM EMP2
GROUP BY telecom;
```

<input type="checkbox"/>	TELECOM	LISTAGG(ENAME,',')WITHINGROUP(ORDERBYENAMI	COUNT(*)
1	cjh	이근호	1
2	kt	김근아,김영토,김지우,방송준,백광훈,송윤호,신	12
3	lg	김광록,도웅,유혜린,은해찬,이광훈,이상민,이천	9
4	sk	김건태,김대경,김동윤,김원섭,윤동환,윤진민,이	8

문제29. 문제28번의 데이터를 담은 테이블을 생성하시오.

```
CREATE TABLE emp2_telecom_ename
( telecom VARCHAR2(10),
  ename VARCHAR2(1000),
  cnt number(10) );
```

문제30. 통신사를 물어보게하고, 통신사를 입력하면 해당 통신사와 통신사의 학생 이름과 인원수가 Emp2\_telecom\_ename 테이블의 입력되게 하는 PLSQL을 작성하시오.

```
accept p_telecom prompt '통신사를 입력하세요'
```

```
declare
  v_telecom varchar2(10);
  v_ename varchar2(1000) :=&p_tel
  v_cnt number(15);

begin

  select telecom, listagg(ename,',') within group(ORDER BY ename), COUNT(*)
    into v_telecom, v_ename, v_cnt
  from emp2
  where telecom = &p_telecom
  group by telecom;

  insert into emp2_telecom_ename
    values(v_telecom, v_ename, v_cnt);

commit;
```

```
end;
/
```

```
SQL> @p30.sql
통신사를 입력하세요'sk'

PL/SQL 처리가 정상적으로 완료되었습니다.
```

```
SQL> @p30.sql
통신사를 입력하세요'lg'

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p30.sql
통신사를 입력하세요'kt'

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p30.sql
통신사를 입력하세요'cjh'

PL/SQL 처리가 정상적으로 완료되었습니다.
```

	TELECOM	ENAME	CNT
1	sk	김건태,김대경,김동윤,김원섭,윤동환,윤진민,이	8
2	lg	김광록,도웅,유혜린,은해찬,이광훈,이상민,이천	9
3	kt	김근아,김영토,김지우,방승준,백광흠,송윤희,신	12
4	cjh	이근호	1

문제31. 가우스 공식을 적용할 테이블을 생성하고 번호를 물어보게 하고 번호를 입력하면 해당번호의 total 컬럼이 업데이트 되는 PLSQL을 작성 하시오.

```
create table gauss
(num number(10),
num1 number(10),
num2 number(10),
total number(10));
```

```
insert into gauss values(1,1,10,null);
insert into gauss values(2,1,20,null);
insert into gauss values(3,1,30,null);
insert into gauss values(4,1,40,null);
insert into gauss values(5,1,50,null);
insert into gauss values(6,1,60,null);
```

```
commit;
```

	NUM	NUM1	NUM2	TOTAL
1	1	1	10	(null)
2	2	1	20	(null)
3	3	1	30	(null)
4	4	1	40	(null)
5	5	1	50	(null)
6	6	1	60	(null)

답)

```
accept p_num prompt '번호를 입력하세요.'
```

```
declare  
  v_total number(10);
```

```
begin
```

```
  update gauss  
    set total = (num1 + num2) * (num2/2)  
    where num = &p_num;
```

```
commit;  
end;  
/
```

**문제32. 문제31번을 수행하는데 몇 개의 행이 갱신되었다는 메시지도 같이 출력되게 하시오.**

```
accept p_num prompt '번호를 입력하세요.'
```

```
declare  
  v_total number(10);
```

```
begin
```

```
  update gauss  
    set total = (num1 + num2) * (num2/2)  
    where num = &p_num;
```

```
  dbms_output.put_line(SQL%rowcount || '행이 입력되었습니다.');
```

```
commit;  
end;  
/
```

**문제33. 부서번호를 물어보게하고 부서번호를 입력하면 해당 부서번호인 직원들이 삭제되게하는 PLSQL을 작성하시오.**

```
accept p_deptno prompt '번호를 입력하세요.'
```

```
declare  
  v_deptno number(10);
```

```
begin  
  delete from emp  
    where deptno = p_deptno;
```

```
  dbms_output.put_line(SQL%rowcount || ' 행이 삭제됨');
```

```
commit;  
  
end;  
/
```

**문제34. 직원 테이블에 LOC컬럼을 추가하고 해당 직원의 부서위치로 값을 갱신하시오.(merge문)**

```
ALTER TABLE EMP  
  ADD loc VARCHAR2(20);
```

```

MERGE INTO EMP e
  USING DEPT d
    ON(e.deptno = d.deptno)
  WHEN matched THEN
    UPDATE SET e.loc = d.loc;

```

Select loc from emp;

	LOC
1	NEW YORK
2	CHICAGO
3	NEW YORK
4	DALLAS
5	CHICAGO
6	CHICAGO
7	CHICAGO
8	CHICAGO
9	CHICAGO
10	DALLAS
11	DALLAS
12	DALLAS
13	DALLAS
14	NEW YORK

**문제35. 이름을 물어보게하고, 이름을 입력하면 해당사원의 부서위치로 loc컬럼이 갱신되게 하는 PLSQL을 작성하시오.**

```
accept p_ename prompt '이름을 입력하세요.'
```

```

declare
  v_loc varchar2(10);
  v_ename varchar2(10);

begin
  select e.ename, d.loc into v_ename, v_loc
    from emp e, dept d
   where e.deptno= d.deptno
      and e.ename = upper('&p_ename');

  update emp
    set loc = v_loc
   where ename = v_ename;

commit;

end;
/

```

**문제36. 부서 테이블에 sumsal컬럼을 추가하고 부서번호를 물어보면, 해당 부서번호의 토탈 월급으로 값이 갱신되게 하는데 토탈월급이 9000이상이어야 갱신되게 하고, 9000보다 작으면 갱신되지 않도록 PLSQL을 작성하시오.**

```

ALTER TABLE DEPT
  ADD sumsal NUMBER(10);

```

```
accept p_deptno prompt '부서번호입력'
```

```

declare
  v_deptno number(20);

```

```

v_sumsal number(10);

begin
  select deptno, sum(sal) into v_deptno, v_sumsal
    from emp
   where deptno = &p_deptno
  group by deptno;

  if v_sumsal >= 9000 then

    update dept
      set sumsal = v_sumsal
     where deptno = v_deptno;

  commit;

end if;

end;
/

```

```

SQL> @p36.sql
부서번호입력20

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p36.sql
부서번호입력10

PL/SQL 처리가 정상적으로 완료되었습니다.

```

SELECT \* FROM DEPT;

	DEPTNO	DNAME	LOC	SUMSAL
1	10	ACCOUNTING	NEW YORK	(null)
2	20	RESEARCH	DALLAS	10875
3	30	SALES	CHICAGO	(null)
4	40	OPERATIONS	BOSTON	(null)

처리는 되지만 결과가 안 들어간다.

## 4. PL/SQL 내에서의 IF 문

2018년 4월 30일 월요일    오후 3:10

### IF 문

#### Syntax:

```
IF condition THEN
    statements;
[ELSIF condition THEN
    statements; ]
[ELSE
    statements; ]
END IF;
```

if 문의 구조

If 조건 then

실행문 ;

Elsif 조건 then

실행문 ;

Else

실행문 ;

End if ;

문제37. 이름을 물어보게하고 이름을 입력하면 해당 사원이 고소득자인지, 저소득자인지 출력되게 하시오.

월급 > 3000 고소득자 입니다.

월급 >= 2000 적당합니다.

월급 < 2000 저소득자입니다.

accept p\_ename prompt '이름을 입력'

declare

v\_sal number(10);

v\_ename varchar2(10) := upper('&p\_ename');

begin

select sal, ename into v\_sal , v\_ename

from emp

where ename = v\_ename;

if v\_sal > 3000 then

dbms\_output.put\_line('고소득자입니다.');

```

elseif v_sal between 2000 and 3000 then
    dbms_output.put_line('적당합니다.');
```

```

else
    dbms_output.put_line('저소득자입니다.');
```

```

end if;

commit;

end;
/

```

**문제38. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자가 짝수인지 홀수인지 출력되게 하시오.**

```

accept p_num prompt '숫자를 입력'

declare
    v_num number(10) := &p_num;

begin
    if mod(v_num,2) = 0 then
        dbms_output.put_line('짝수 입니다');
```

```


    else
        dbms_output.put_line('홀수 입니다');
```

```

    end if;

commit;
end;
/

```



```

SQL> @p38.sql
숫자를 입력10
짝수 입니다

PL/SQL 처리가 정상적으로 완료되었습니다.

```

**문제39. 피타고라스 직각 삼각형 공식을 PLSQL로 구현 하시오.**

가로의 길이를 입력하세요. 3  
 세로의 길이를 입력하세요. 4  
 빗변의 길이를 입력하세요. 5

**직각 삼각형이 맞습니다.**

```

accept p_garo prompt '가로를 입력'
accept p_sero prompt '세로를 입력'
accept p_bitbyeon prompt '빗변을 입력'

declare
    v_garo number(10) := &p_garo;
```

```

v_sero number(10) := &p_sero;
v_bitbyeon number(10) := &p_bitbyeon;

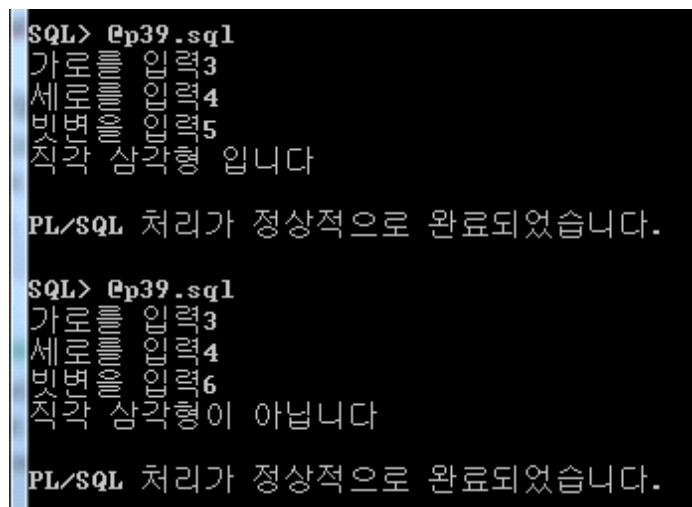
begin
  if power(v_garo,2) + power(v_sero,2) = power(v_bitbyeon,2) then
    dbms_output.put_line('직각 삼각형 입니다');

  else
    dbms_output.put_line('직각 삼각형이 아닙니다');

  end if;

end;
/

```



```

SQL> @p39.sql
가로를 입력3
세로를 입력4
빗변을 입력5
직각 삼각형 입니다

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p39.sql
가로를 입력3
세로를 입력4
빗변을 입력6
직각 삼각형이 아닙니다

PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제40. 두개의 숫자를 물어보게하고, 아래의 메시지가 출력되게 하시오.

숫자를 입력하세요. - 2

숫자를 입력하세요. - 5

5는 2보다 큽니다.

숫자를 입력하세요. - 5

숫자를 입력하세요. - 2

5는 2보다 큽니다.

-----

```

accept p_num1 prompt '첫 번째 숫자를 입력'

```

```

accept p_num2 prompt '두 번째 숫자를 입력'

```

```

declare

```

```

  v_num1 number(10) := &p_num1;

```

```

  v_num2 number(10) := &p_num2;

```

```

begin

```

```

  if v_num1 > v_num2 then

```

```

    dbms_output.put_line(v_num1 || '는' || v_num2 || '보다 큽니다');

```

```

  elsif v_num2 > v_num1 then

```

```

    dbms_output.put_line(v_num2 || '는' || v_num1 || '보다 큽니다');

```

```

  else

```



```

        dbms_output.put_line('두 수는 같습니다');

    end if;

end;
/

```

```

SQL> @p40.sql
첫 번째 숫자를 입력2
두 번째 숫자를 입력5
5는2보다 큼니다

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p40.sql
첫 번째 숫자를 입력3
두 번째 숫자를 입력5
5는3보다 큼니다

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p40.sql
첫 번째 숫자를 입력2
두 번째 숫자를 입력2
두 수는 같습니다

```

## 5. PL/SQL 내에서의 Loop 문

2018년 5월 1일 화요일 오전 9:41

- Loop 문은 언제 사용하는가?  
- 특정 실행문을 반복 실행하고자 할 때 사용

- Loop문의 종류 3가지

1. Basic Loop문
2. While Loop문
3. For Loop문

### 5-1. Basic Loop 문

```
begin
    loop
        실행 문;
        루프 문을 종료시키는 문법;    <- 반드시 있어야한다.
    end loop;

end;
/
```

ex)

```
declare
    v_count number(10) := 0;

begin
    loop
        v_count := v_count + 1;    <- 할당연산자를 이부분에 쓰면 오른쪽(v_count + 1) 부터 실행된다.
        dbms_output.put_line(v_count);
        exit when v_count = 1000;    <- loop종료 시키는 문법
    end loop;
end;
/
```

문제41. basic loop문으로 구구단 2단을 출력하시오.

```
begin
    loop
        v_num := v_num + 1;
        dbms_output.put_line ('2*' || v_num || '=' || 2*v_num);
        exit when v_num = 9;
    end loop;
end;
/
```

```
SQL> @p41.sql
2*1=2
2*2=4
2*3=6
2*4=8
2*5=10
2*6=12
2*7=14
2*8=16
2*9=18
PL/SQL 처리가 정상적으로 완료되었습니다.
```

문제42. 아래와 같이 결과를 출력하시오.

```
SQL> @p42.sql
1 2 3 4 5 6 7 8 9 10
PL/SQL 처리가 정상적으로 완료되었습니다.
```

```
declare
    v_num number(10) := 1;

begin
    loop
        dbms_output.put(v_num || ' ');
        v_num := v_num + 1;
        exit when v_num = 11;
    end loop;

    dbms_output.new_line;

end;
/
```

**\_Line 을 붙이면 바로 출력해라.**  
**\_Line이 없으면 ( )안의 있는 내용을 버퍼(메모리)에 올림.**  
**<-- 버퍼에 있는 내용을 출력**

문제43. 아래의 결과를 출력하시오.

```
2 x 1 = 2 3 x 1 = 3 4 x 1 = 4 5 x 1 = 5 ... 9 x 1 = 9
declare
    v_num number(10) := 1;

begin
    loop

        v_num := v_num + 1;
        dbms_output.put(v_num || ' x 1 = ' || 1*v_num || ' ');

        exit when v_num = 9;
    end loop;
    dbms_output.new_line;
end;
/
```

문제44.(점심시간) 아래와 같이 결과를 출력하시오.

```
2 x 1 = 2 3 x 1 = 3 4 x 1 = 4 5 x 1 = 5 ... 9 x 1 = 9
2 x 2 = 4 3 x 2 = 6 4 x 2 = 8 5 x 2 = 10 ... 9 x 2 = 18
.
.
.
2 x 9 = 18 3 x 9 = 27 4 x 9 = 36 5 x 9 = 45 ... 9 x 9 = 81
```

문제45. basic loop문을 이용해서 아래의 결과를 출력하시오. (v\_star := v\_star || '★')

```
★
★★
★★★
★★★★
★★★★★
★★★★★★
★★★★★★★
★★★★★★★★
★★★★★★★★★
★★★★★★★★★★
★★★★★★★★★★★
```

-----  
set verify off

set serveroutput on

declare

  v\_star varchar2(100) := '★';

begin

  loop

    dbms\_output.put\_line(v\_star);

    v\_star := v\_star || '★';

    exit when length(v\_star) = 11;

  end loop;

end;

/



문제46. 아래와 같이 출력하시오.

```
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
★★★★★★★★★★★
```

★

```
-----
set verify off
set serveroutput on

declare
  v_star varchar2(100) := '★★★★★★★★★★★★';

begin
  loop

    v_star := substr(v_star, 1, length(v_star)-1);
    dbms_output.put_line(v_star);
    exit when length(v_star) = 1;
  end loop;
end;
/
```

문제47. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자까지 별을 찍어서 아래와 같이 하나씩 늘어나게 하시오.

숫자를 입력하세요 5

★

★★

★★★

★★★★

★★★★★

답)

```
set verify off
set serveroutput on

accept p_star prompt '숫자를 입력하세요.'

declare
  v_star varchar2(100) := '★';

begin
  loop
    dbms_output.put_line(v_star);
    v_star := v_star || '★';
    exit when length(v_star) = &p_star + 1;
  end loop;
end;
/
```

문제48. (점심시간)숫자를 물어보게하고 입력하면 아래와 같이 별이 출력되게 하시오.

숫자를 입력하세요 10

★★★★★★★★★★

★★★★★★★★★★

★★★★★★★★★★

★★★★★★★★★★

★★★★★★

★★★★★

★★★★

★★★

★★

★

accept p\_num prompt '숫자를 입력하세요'

declare

v\_star varchar2(200) := '★';

begin

v\_star := rpad(v\_star, (&p\_num + 1) \* 2, '★');

loop

v\_star := substr(v\_star, 1, length(v\_star) - 1);

dbms\_output.put\_line(v\_star);

exit when length(v\_star) = 1;

end loop;

end;

/

## • 이중 Loop문

Loop

실행 문 ;

loop

실행 문;

inner loop 문을 종료하기 위한 코드;

end loop;

outer loop 문을 종료하기 위한 코드;

End loop;

**문제49. 구구단 2단과 3단을 출력하시오.**

set verify off

set serveroutput on

declare

v\_num2 number(10) := 0;

v\_num3 number(10) := 0;

begin

loop

v\_num2 := v\_num2 + 1;

dbms\_output.put\_line ('2\*' || v\_num2 || '=' || 2\*v\_num2);

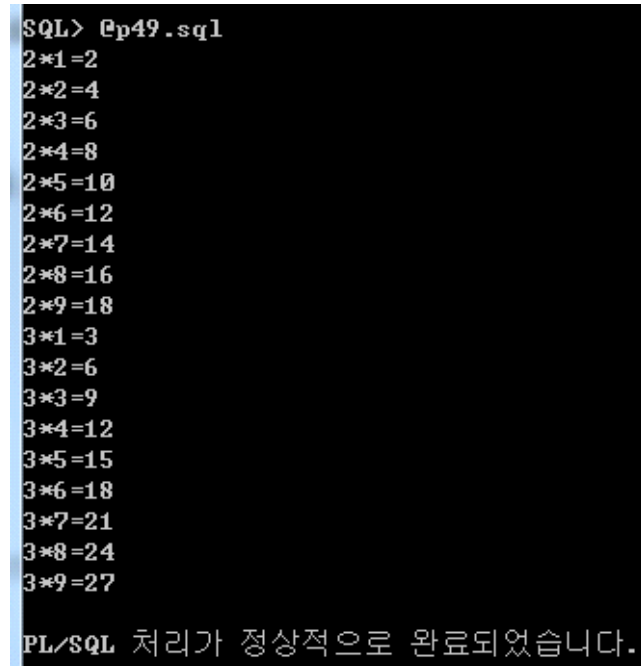
exit when v\_num2 = 9;

end loop;

```

loop
  v_num3 := v_num3+1;
  dbms_output.put_line ('3*' || v_num3 || '=' || 3*v_num3);
  exit when v_num3 = 9;
end loop;
end;
/

```



```

SQL> @p49.sql
2*1=2
2*2=4
2*3=6
2*4=8
2*5=10
2*6=12
2*7=14
2*8=16
2*9=18
3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27
PL/SQL 처리가 정상적으로 완료되었습니다.

```

이러면 구구단이 많아지면 무식한 방법이라서 loop 안에 loop를 넣어서 해야한다.

문제50. 구구단을 2단과 3단을 출력하는데 루프문을 중첩해서 수행하시오.

```

set verify off
set serveroutput on

declare
  v_num2 number(10) := 0;
  v_num3 number(10) := 0;

begin
  loop
    v_num2 := v_num2 + 1;
    v_num3 := 0;
    loop
      v_num3 := v_num3 + 1;
      dbms_output.put_line (v_num2 || 'x' || v_num3 || '=' || v_num2*v_num3);
      exit when v_num3 = 9;
    end loop;

    exit when v_num2 = 3;
  end loop;
end;
/

```

문제51. 문제44번을 다시 수행해보세요.

```

declare
  v_num2 number(10) := 1;
  v_num3 number(10) := 0;

begin

```

```

loop
  v_num2 := v_num2 + 1;
  v_num3 := 0;
  loop
    v_num3 := v_num3 + 1;
    dbms_output.put(v_num2 || 'x' || v_num3 || '=' || v_num2*v_num3 || ' ');
    exit when v_num3 = 9;
  end loop;
  dbms_output.new_line;

  exit when v_num2 = 9;

end loop;

end;
/

```

```

SQL> @p51.sql
2x1=2 2x2=4 2x3=6 2x4=8 2x5=10 2x6=12 2x7=14 2x8=16 2x9=18
3x1=3 3x2=6 3x3=9 3x4=12 3x5=15 3x6=18 3x7=21 3x8=24 3x9=27
4x1=4 4x2=8 4x3=12 4x4=16 4x5=20 4x6=24 4x7=28 4x8=32 4x9=36
5x1=5 5x2=10 5x3=15 5x4=20 5x5=25 5x6=30 5x7=35 5x8=40 5x9=45
6x1=6 6x2=12 6x3=18 6x4=24 6x5=30 6x6=36 6x7=42 6x8=48 6x9=54
7x1=7 7x2=14 7x3=21 7x4=28 7x5=35 7x6=42 7x7=49 7x8=56 7x9=63
8x1=8 8x2=16 8x3=24 8x4=32 8x5=40 8x6=48 8x7=56 8x8=64 8x9=72
9x1=9 9x2=18 9x3=27 9x4=36 9x5=45 9x6=54 9x7=63 9x8=72 9x9=81

```

문제52. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자만큼 ★가 가로로 출력 되게하시오.

```
accept p_num prompt '숫자를 입력하세요'
```

```

declare
  v_num number(10) := 0;

begin
  loop
    v_num := v_num + 1;
    dbms_output.put('★');

    exit when &p_num = v_num;
  end loop;

  dbms_output.new_line;
end;
/

```

```

SQL> @p52.sql
숫자를 입력하세요4
★★★★
PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제53. factorial 을 PLSQL로 구현하시오.

숫자를 입력하세요 5

5! = 120

```
accept p_num prompt '숫자를 입력하세요'
```

```

declare
  v_num number(10) := &p_num;

```



```

v_result number(10) := 1;
begin
    loop

        dbms_output.put(v_num || 'x');
        v_result := v_result * v_num;
        v_num := v_num -1;

        exit when 1 = v_num;

    end loop;
    dbms_output.put(v_num || '=' || v_result*v_num);
    dbms_output.new_line;
end;
/

```

**문제54. power 함수를 PLSQL로 구현하시오.(power 함수 쓰지말고)**

**밑수를 입력하세요. 2**

**지수를 입력하세요. 3**

**8 입니다**

accept p\_num1 prompt '밑수를 입력하세요'

accept p\_num2 prompt '지수를 입력하세요'

```

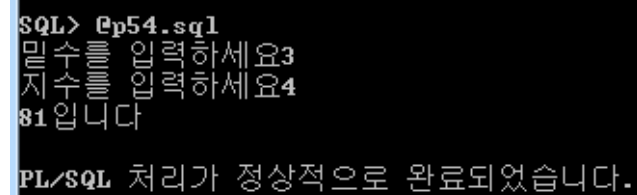
declare
    v_result number(10) := &p_num1;
    v_cnt number(10) := &p_num2;

begin
    loop
        v_result := v_result * &p_num1;
        v_cnt := v_cnt -1;

        exit when v_cnt = 1;
    end loop;

    dbms_output.put_line(v_result || '입니다');
end;
/

```



```

SQL> @p54.sql
밑수를 입력하세요3
지수를 입력하세요4
81입니다
PL/SQL 처리가 정상적으로 완료되었습니다.

```

**문제55. (마지막)mod 함수를 PLSQL로 구현하시오.**

**숫자를 입력하세요 10**

**숫자를 입력하세요 3**

**10을 3으로 나눈 나머지 값은 1입니다.**

```

-----
set serveroutput on
set verify off

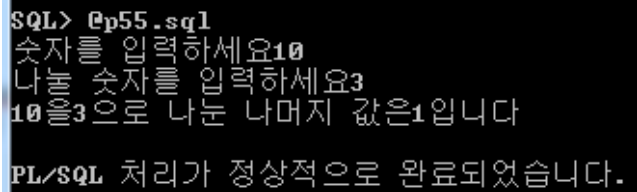
accept p_num1 prompt '숫자를 입력하세요'
accept p_num2 prompt '나눌 숫자를 입력하세요'

```

```

declare
    v_num1 number(10) := &p_num1;
    v_cnt number(10) := &p_num2;
begin
    loop
        v_num1 := v_num1 - v_cnt;
        exit when := &p_num2 > v_num1;
    end loop;
    dbms_output.put_line(&p_num1 || '을' || &p_num2 || '으로 나눈 나머지 값은' || v_num1 || '입니다');
end;
/

```



```

SQL> @p55.sql
숫자를 입력하세요10
나눌 숫자를 입력하세요3
10을3으로 나눈 나머지 값은1입니다
PL/SQL 처리가 정상적으로 완료되었습니다.

```

## 5-2 while loop 문

### basic loop문

```

Declare
    v_cnt number(10) := 0;
Begin
    loop
        v_cnt := v_cnt + 1;
        실행문;
        exit when v_cnt = 10;
    end loop;
End;
/

```

### while loop문

```

Declare
    v_cnt number(10) := 0;
Begin
    while v_cnt < 10 loop
        v_cnt := v_cnt + 1;
        실행문;
    end loop;
End;
/

```

**문제56. 구구단 2단을 while loop 문으로 구현 하시오.**

```

set verify off
set serveroutput on

```

```

declare
    v_num number(10) := 0;

begin
    while v_num < 9 loop
        v_num := v_num + 1;
        dbms_output.put_line('2 x ' || v_num || '=' || 2*v_num);
    end loop;
end;
/

```

```

SQL> @p56.sql
2 x 1= 2
2 x 2= 4
2 x 3= 6
2 x 4= 8
2 x 5= 10
2 x 6= 12
2 x 7= 14
2 x 8= 16
2 x 9= 18

PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제57. 아래와 같이 출력하시오.(while loop문으로)

1 2 3 4 5 6 7 8 9 10

```

declare
    v_num number(10) := 1;

begin
    while v_num < 11 loop
        dbms_output.put(v_num || ' ');
        v_num := v_num + 1;
    end loop;
    dbms_output.new_line;
end;
/

```

```

SQL> @p57.sql
1 2 3 4 5 6 7 8 9 10

PL/SQL 처리가 정상적으로 완료되었습니다.

```

## 중첩 basic loop

```

loop
    v_cnt1 := v_cnt1 + 1;
    v_cnt2 := 0;    // 초기화
    loop
        v_cnt2 := v_cnt2 + 1;
        실행문;
        exit when v_cnt2 = 10;
    end loop;

    exit when v_cnt1 = 10;
end loop;

```

## 중첩 while loop

```

While v_cnt1 < 10 loop
    v_cnt1 := v_cnt1 + 1;
    v_cnt2 := 0; // 초기화

    while v_cnt2 < 10 loop
        v_cnt2 := v_cnt2 + 1;
        실행문;
    end loop;

end loop;

```

문제58. 중첩 while loop 문으로 구구단 전체를 세로로 구현하시오.

```

declare
    v_num1 number(10) := 1;
    v_num2 number(10) := 2;

begin
    while v_num2 < 10 loop
        v_num1 := 1;
        while v_num1 < 10 loop
            dbms_output.put_line(v_num2 || 'x' || v_num1 || '=' || v_num2 * v_num1);
            v_num1 := v_num1 + 1;
        end loop;
        v_num2 := v_num2 + 1;
    end loop;
end;

```

```

        end loop;
        v_num2 := v_num2 + 1;
    end loop;
end;
/

```

```

2x1=2
2x2=4
2x3=6
2x4=8
2x5=10
2x6=12
2x7=14
2x8=16
2x9=18
3x1=3

```

```

8x8=64
8x9=72
9x1=9
9x2=18
9x3=27
9x4=36
9x5=45
9x6=54
9x7=63
9x8=72
9x9=81
PL/SQL 처리가 정상적으로 완료되었습니다.

```

**문제59. 중첩 while loop문으로 구구단 전체를 가로로 구현하시오.**

```

declare
    v_num1 number(10) := 1;
    v_num2 number(10) := 2;

begin
    while v_num1 < 10 loop
        v_num2 := 2;

        while v_num2 < 10 loop
            dbms_output.put(v_num2 || 'x' || v_num1 || '=' || v_num2 * v_num1 || ' ');
            v_num2 := v_num2 + 1;
        end loop;

        dbms_output.new_line;
        v_num1 := v_num1 + 1;

    end loop;
end;
/

```

```

SQL> @p59.sql
2x1=2 3x1=3 4x1=4 5x1=5 6x1=6 7x1=7 8x1=8 9x1=9
2x2=4 3x2=6 4x2=8 5x2=10 6x2=12 7x2=14 8x2=16 9x2=18
2x3=6 3x3=9 4x3=12 5x3=15 6x3=18 7x3=21 8x3=24 9x3=27
2x4=8 3x4=12 4x4=16 5x4=20 6x4=24 7x4=28 8x4=32 9x4=36
2x5=10 3x5=15 4x5=20 5x5=25 6x5=30 7x5=35 8x5=40 9x5=45
2x6=12 3x6=18 4x6=24 5x6=30 6x6=36 7x6=42 8x6=48 9x6=54
2x7=14 3x7=21 4x7=28 5x7=35 6x7=42 7x7=49 8x7=56 9x7=63
2x8=16 3x8=24 4x8=32 5x8=40 6x8=48 7x8=56 8x8=64 9x8=72
2x9=18 3x9=27 4x9=36 5x9=45 6x9=54 7x9=63 8x9=72 9x9=81

PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제60. while loop 문을 이용해서 아래의 결과를 출력하시오.

```

★
★★
★★★
★★★★
★★★★★
★★★★★★
★★★★★★★
★★★★★★★★
★★★★★★★★★
★★★★★★★★★★
★★★★★★★★★★★
-----
declare
    v_star varchar2(100) := '★';

begin
    while length(v_star) < 11 loop
        dbms_output.put_line(v_star);
        v_star := v_star || '★';
    end loop;
end;
/

```

문제61. 숫자를 물어보게 하고 숫자를 입력하면 ★을 출력하시오.(while loop문으로)

```

숫자를 입력하세요 5
accept p_num prompt '숫자를 입력하세요'

declare
    v_star varchar2(100) := '★';

begin
    while length(v_star) < &p_num+1 loop
        dbms_output.put_line(v_star);
        v_star := v_star || '★';
    end loop;
end;
/

```

```
SQL> ep61.sql
숫자를 입력하세요5
```

PL/SQL 처리가 정상적으로 완료되었습니다.

```
accept p_num prompt '숫자를 입력하세요'
```

```

declare
  v_num1 number(10) := &p_num;
  v_num2 number(10) := 1;
begin
  while v_num2 <= &p_num loop

    v_num1 := &p_num;

    while v_num2 <= v_num1 loop
      dbms_output.put('★');
      v_num1 := v_num1 - 1;
    end loop;

    v_num2 := v_num2 + 1;
    dbms_output.new_line;
  end loop;
end;
/

```

문제63. while loop문으로 마름모를 출력하시오.

★  
★ ★  
★ ★ ★  
★ ★ ★ ★  
★ ★ ★ ★ ★  
★ ★ ★ ★ ★  
★ ★ ★ ★  
★ ★ ★  
★ ★  
★

## 5-3 for loop문

## basic loop

```
loop
  v_cnt1 := v_cnt1 + 1;
```

## for loop

For i in 1 .. 10 loop

loop	For i in 1 .. 10 loop
v_cnt1 := v_cnt1 + 1;	
실행문;	실행문;
exit when v_cnt1 = 10;	end loop;
end loop;	

문제64. for loop 문을 이용해서 1부터 10까지 출력하시오.

```

1
2
3
4
5
6
7
8
9
10
-----
begin
  for i in 1 .. 10 loop

    dbms_output.put_line(i);
  end loop;
end;
/

```

문제65. for loop 문을 이용해서 1부터 10까지 출력하시오.(가로로)

```

begin
  for i in 1 .. 10 loop
    dbms_output.put(i || ' ');
  end loop;
  dbms_output.new_line;
end;
/

```



```

SQL> @p65.sql
1 2 3 4 5 6 7 8 9 10
PL/SQL 처리가 정상적으로 완료되었습니다.

```

## 중첩 for loop문

### 중첩 basic loop

```

loop
  v_cnt1 := v_cnt1 + 1;
  v_cnt2 := 0;    // 초기화
  loop
    v_cnt2 := v_cnt2 + 1;
    실행문;
    exit when v_cnt2 = 10;
  end loop;

```

### 중첩 for loop

```

For i in 1 .. 10 loop
  for j in 1 .. 10 loop
    실행문;
  end loop;
end loop;

```

```

        exit when v_cnt2 = 10;
    end loop;

    exit when v_cnt1 = 10;
end loop;

```

**문제66. 중첩 for loop 이용해서 구구단 전체를 세로로 출력하시오.**

```

begin
    for i in 2 .. 9 loop

        for j in 1 .. 9 loop
            dbms_output.put_line(i || 'x' || j || '=' || i*j);
        end loop;

    end loop;
end;
/

```

**문제67. 숫자를 물어보게하고 숫자를 입력하면 ★을 출력하시오.(for loop문)**

```

set verify off
set serveroutput on

accept p_num prompt '숫자 입력 : '

begin

    for i in 1..&p_num loop
        for j in 1..i loop
            dbms_output.put('★');
        end loop;
        dbms_output.new_line;
    end loop;

end;
/

```



**문제68. 문제67번을 반대로**

```

accept p_num prompt '숫자입력'

declare
    v_star varchar2(100) := '★';

begin
    for i in 1 .. &p_num loop
        v_star := v_star || '★';
    end loop;

    for j in 1 .. &p_num loop

```



```

        v_star := substr(v_star, 1, length(v_star) - j);
        dbms_output.put_line(v_star);
    end loop;

```

```

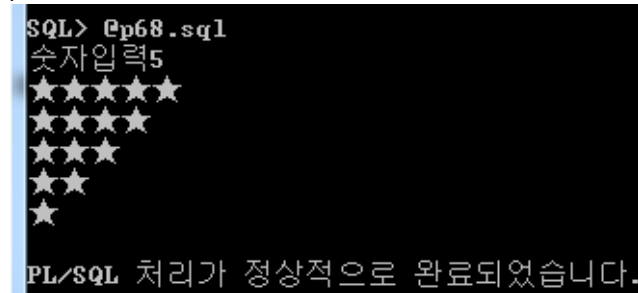
end;

```

```

/

```



문제69. 아래와 같이 가로, 세로를 각각 물어보게 하고 ★ 모양이 출력되게 하시오.

가로 숫자를 입력하세요 5

세로 숫자를 입력하세요 5

```

★★★★★
★★★★★
★★★★★
★★★★★
★★★★★

```

```

accept p_num1 prompt '가로 숫자 입력'

```

```

accept p_num2 prompt '세로 숫자 입력'

```

```

begin for i in 1 .. &p_num1 loop
    for j in 1 .. &p_num2 loop
        dbms_output.put('★');
    end loop;
    dbms_output.new_line;
end loop;
end;
/

```

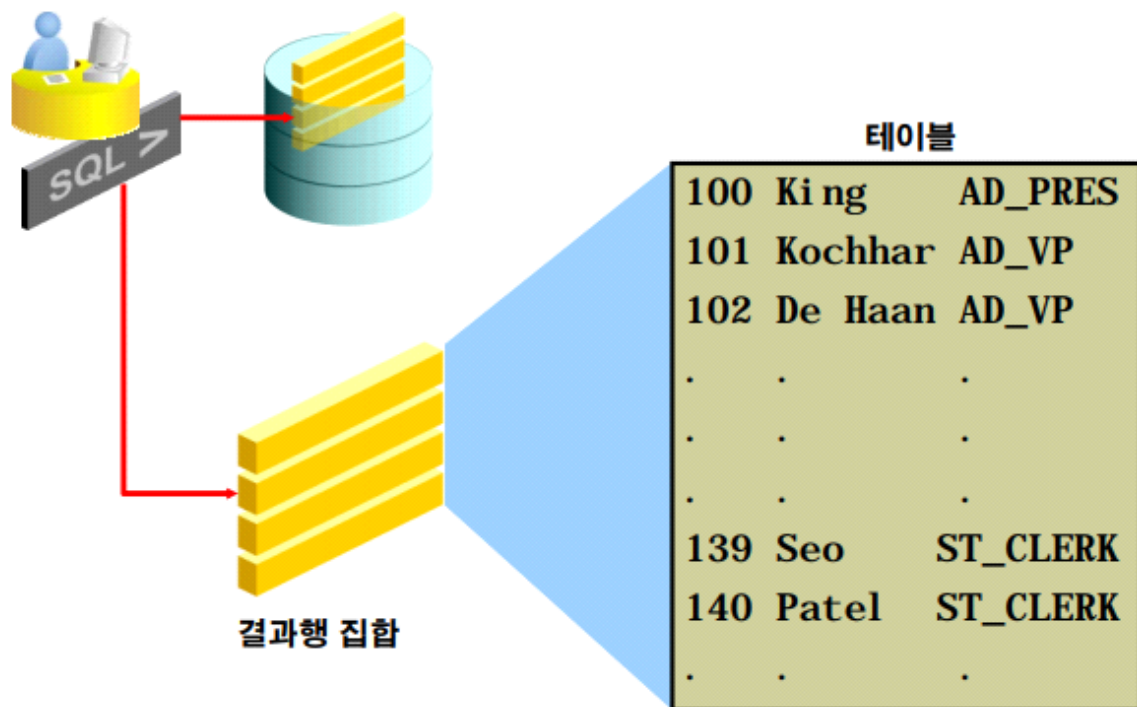
## 6. PL/SQL 내에서의 커서문

2018년 5월 2일 수요일    오후 3:14

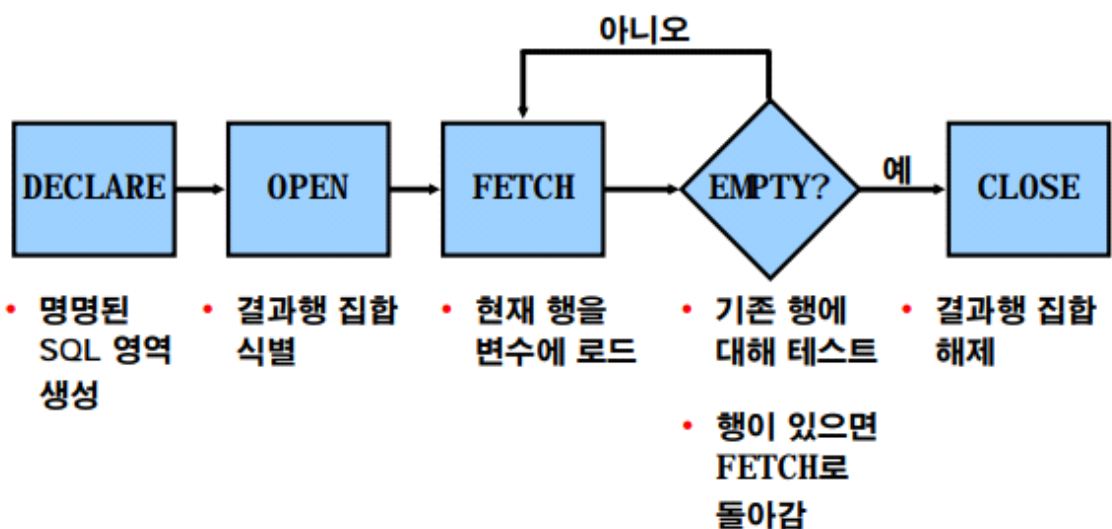
### 커서문을 사용 하면 이점?

- 처리해야 할 데이터를 메모리에 올려놓고 작업 할 수 있기 때문에 한번에 여러 개의 작업을 할 수 있다.

### 명시적 커서 작업



### • 명시적 커서의 제어 순서



문제70. 사원 테이블의 grade 라는 컬럼을 추가하고, grade 컬럼의 데이터를 아래의 내용으로 갱신되게

하는데, 이름을 물어보게하고 이름을 입력하면 아래와 같이 갱신되게 하시오.

월급 >= 3000 이면 A

월급 >= 2500 이면 B

월급 >= 2000 이면 C

월급 >= 1000 이면 D

나머지 월급은 F로 갱신되게 하시오.

```
-----  
ALTER TABLE EMP  
ADD grade VARCHAR2(10);
```

```
accept p_ename prompt '이름을 입력'
```

```
declare  
  v_sal emp.sal%type;  
  v_ename emp.ename%type := upper('&p_ename');  
  v_grade emp.grade%type;
```

Emp 테이블의 sal 컬럼의 데이터  
타입을 그대로 따르겠다.

```
begin  
  select sal into v_sal  
    from emp  
   where ename = v_ename;  
  
  v_grade := case when v_sal >= 3000 then 'A'  
                  when v_sal >= 2500 then 'B'  
                  when v_sal >= 2000 then 'C'  
                  when v_sal >= 1000 then 'D'  
                  else 'F' end;  
  
  update emp  
    set grade = v_grade  
   where ename = v_ename;  
  
  commit;  
end;  
/
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	LOC	GRADE
7839	KING	PRESIDENT		81/11/17	5000		10		A
7698	BLAKE	MANAGER	7839	81/05/01	2850		30		

문제71. 문제70번을 일일이 이름을 물어보지 않고 14명의 직원들의 grade가 동시에 갱신되게 기존 코드에 loop문만 추가해서 모두 작성 하시오.

```
declare  
  v_sal emp.sal%type;  
  
  v_grade emp.grade%type;  
  
begin  
  for i in 7000 .. 8000 loop  
  
    select sal into v_sal  
      from emp  
     where empno = i;
```

```

v_grade := case when v_sal >= 3000 then 'A'
              when v_sal >= 2500 then 'B'
              when v_sal >= 2000 then 'C'
              when v_sal >= 1000 then 'D'
              else 'F' end;

update emp
  set grade = v_grade
  where empno = i;

commit;
end loop;
end;
/

```

이렇게 하면 비어있는 empno에 데이터를 찾을 수 없다고 나온다.

## 문제72. 문제71번을 커서문을 이용해서 작성하시오.

```

declare
  v_sal emp.sal%type;
  v_ename emp.ename%type;
  v_grade emp.grade%type;

  cursor emp_cursor is
    select ename, sal      <-- 메모리에 sql문 내용을 emp_cursor 라는 이름으로 올려 놓음.
    from emp;

begin
  open emp_cursor;      <-- 커서 열기

  loop
    fetch emp_cursor into v_ename, v_sal; <-- 위에 메모리에서 데이터(ename, sal 순서 맞춰야함)를
                                          하나씩 가져온다.
    Exit when emp_cursor%notfound; <-- 커서에서 데이터가 발견되지 않을 때 loop를 종료 해라.

    v_grade := case when v_sal >= 3000 then 'A'
                    when v_sal >= 2500 then 'B'
                    when v_sal >= 2000 then 'C'
                    when v_sal >= 1000 then 'D'
                    else 'F' end;

    update emp
      set grade = v_grade
      where ename = v_ename;

    commit;

  end loop;
  close emp_cursor; <-- 커서 닫기(반드시 넣어줘야함 나중에 메모리가 부족함)

end;
/

```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	LOC	GRADE
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000	(null)	10	(null)	A
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	(null)	30	(null)	B
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	(null)	10	(null)	C
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	(null)	20	(null)	B
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	1400	30	(null)	D
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600	300	30	(null)	D
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500	0	30	(null)	D
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950	(null)	30	(null)	F
9	7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	1250	500	30	(null)	D
10	7902	FORD	ANALYST	7566	1981-12-11 오전 12:00:00	3000	(null)	20	(null)	A
11	7369	SMITH	CLERK	7902	1980-12-09 오전 12:00:00	800	(null)	20	(null)	F
12	7788	SCOTT	ANALYST	7566	1982-12-22 오전 12:00:00	3000	(null)	20	DALLAS	A
13	7876	ADAMS	CLERK	7788	1983-01-15 오전 12:00:00	1100	(null)	20	(null)	D
14	7934	MILLER	CLERK	7782	1982-01-11 오전 12:00:00	1300	(null)	10	(null)	D

문제73. 우리반 테이블에 grade 라는 컬럼을 추가하고 grade 라는 컬럼에 데이터가 아래의 내용으로 자동으로 갱신되게 하시오.

나이 >= 30 A

나이 >= 28 B

나이 >= 26 C

나이 >= 25 D

나머지 나이는 F 로 갱신

```

declare
v_age emp2.age%type;
v_ename emp2.ename%type;
v_grade emp2.grade%type;

cursor emp2_cursor is
select ename, age
from emp2;

begin
open emp2_cursor;

loop
fetch emp2_cursor into v_ename, v_age;
exit when emp2_cursor%notfound;

v_grade := case when v_age >= 30 then 'A'
when v_age >= 28 then 'B'
when v_age >= 26 then 'C'
when v_age >= 25 then 'D'
else 'F' end;

update emp2
set grade = v_grade
where ename = v_ename;
commit;
end loop;
close emp2_cursor;
end;
/

```

	ENAME	AGE	GRADE
1	김원섭	26	C
2	김광록	27	C
3	유혜린	26	C
4	이유진	25	D
5	윤진민	27	C
6	백광훈	26	C
7	장은희	24	F
8	김영토	28	B
9	차호성	28	B
10	방송준	26	C
11	이한새	25	D
12	이근호	28	B
13	신영근	25	D
14	신현수	27	C
15	이찬준	28	B

문제74. 이름, 월급, 순위를 출력하는데 순위가 월급이 높은 사원순으로 순위를 부여하시오.

```
SELECT ename, sal, DENSE_RANK() OVER(ORDER BY sal desc) rnk
FROM EMP;
```

	ENAME	SAL	RNK
1	KING	5000	1
2	SCOTT	3000	2
3	FORD	3000	2
4	JONES	2975	3
5	BLAKE	2850	4
6	CLARK	2450	5
7	ALLEN	1600	6
8	TURNER	1500	7
9	MILLER	1300	8
10	WARD	1250	9
11	MARTIN	1250	9
12	ADAMS	1100	10
13	JAMES	950	11
14	SMITH	800	12

문제75. 사원테이블의 rnk라는 컬럼을 추가하고 rnk 컬럼에 월급에 대한 순위로 값을 갱신하는 PLSQL 프로그래밍을 작성하시오.

```
ALTER TABLE EMP
ADD rnk NUMBER(10);
```

```
set verify off
set serveroutput on
```

```
declare
v_rnk emp.rnk%type;
v_sal emp.sal%type;
v_ename emp.ename%type;
```

```
cursor emp_cursor is
select ename, sal, rank() over (order by sal desc)
from emp;
```

```
begin
open emp_cursor;
```

```

loop
  fetch emp_cursor into v_ename, v_sal, v_rnk;
  exit when emp_cursor%notfound;

  update emp
  set rnk = v_rnk
  where ename = v_ename;

  commit;
end loop;
close emp_cursor;
end;
/

```

SELECT ename, sal, rnk FROM EMP;

	ENAME	SAL	RNK
1	KING	5000	1
2	BLAKE	2850	5
3	CLARK	2450	6
4	JONES	2975	4
5	MARTIN	1250	10
6	ALLEN	1600	7
7	TURNER	1500	8
8	JAMES	950	13
9	WARD	1250	10
10	FORD	3000	2
11	SMITH	800	14
12	SCOTT	3000	2
13	ADAMS	1100	12
14	MILLER	1300	9

#### • 커서문 작성 순서

1. Declare 에 cursor 선언 : cursor 커서명 is selcet문
2. Begin 에서 커서 테이블 사용한다고 open 선언
3. 커서값을 한 로우씩 가져옴 : fetch 커서명 into 변수1, 변수2
4. 커서값이 더 있는지 없는지 확인 %notfound
5. 커서사용이 끝나면 반드시 커서를 close 해줘야한다.!!

문제76. 사원 테이블 loc컬럼을 추가하고 해당 사원의 부서위치로 값을 갱신하는 PLSQL 프로그램을 작성하시오.

```

ALTER TABLE EMP
ADD loc VARCHAR2(20);

```

/\* 프로그램 설명 : 사원테이블의 loc컬럼에 해당 사원의 부서위치로 갱신 \*/

/\* 프로그램 생성자 : 김원섭 \*/

```

declare
  cursor emp_cursor is
    select e.ename, d.loc
    from emp e, dept d
    where e.deptno = d.deptno;

  v_ename emp.ename%type;

```

```

v_loc emp.loc%type;

begin
  open emp_cursor;

  loop
    fetch emp_cursor into v_ename, v_loc;
    exit when emp_cursor%notfound;

    update emp
      set loc = v_loc
      where ename = v_ename;

    commit;
  end loop;
  close emp_cursor;

end;
/

```

```
SELECT ename, loc FROM EMP;
```

	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	CHICAGO
3	CLARK	NEW YORK
4	JONES	DALLAS
5	MARTIN	CHICAGO
6	ALLEN	CHICAGO
7	TURNER	CHICAGO
8	JAMES	CHICAGO
9	WARD	CHICAGO
10	FORD	DALLAS
11	SMITH	DALLAS
12	SCOTT	DALLAS
13	ADAMS	DALLAS
14	MILLER	NEW YORK

**문제77. 학생이름, 생일, 태어난 날짜의 요일을 출력하시오.**

```
SELECT ename, birth, TO_CHAR(birth, 'day')
FROM EMP2;
```

**문제78. 우리반 테이블에 birth\_day 라는 컬럼을 추가하고 해당 학생이 태어난 요일로 값을 갱신하는 PLSQL 프로그래밍을 작성하시오.**

```

Alter table emp2
  add birth_day varchar2(100);

declare
  cursor emp2_cursor is
    select ename, birth, to_char(birth, 'day')
    from emp2;

  v_ename emp2.ename%type;
  v_birth emp2.birth%type;
  v_birth_day emp2.birth_day%type;

begin

```



```

open emp2_cursor;
loop
  fetch emp2_cursor into v_ename, v_birth, v_birth_day;
  exit when emp2_cursor%notfound;

  update emp2
    set birth_day = v_birth_day
    where ename = v_ename;

  commit;
end loop;
close emp2_cursor;
end;
/

```

```

SELECT ename, birth, birth_day
FROM EMP2;

```

	ENAME	BIRTH	BIRTH_DAY
1	김원섭	1994-02-08 오전 12:00:00	화요일
2	김광록	1992-12-16 오전 12:00:00	수요일
3	유혜린	1993-04-22 오전 12:00:00	목요일
4	이유진	1994-12-06 오전 12:00:00	화요일
5	윤진민	1992-07-15 오전 12:00:00	수요일
6	백광훈	1993-06-18 오전 12:00:00	금요일
7	장은희	1996-01-07 오전 12:00:00	일요일
8	김영토	1991-10-13 오전 12:00:00	일요일
9	차호성	1991-07-01 오전 12:00:00	월요일
10	방송준	1993-12-13 오전 12:00:00	월요일
11	미한새	1994-01-19 오전 12:00:00	수요일
12	이근호	1991-04-24 오전 12:00:00	수요일
13	신영근	1994-02-03 오전 12:00:00	목요일
14	신현수	1992-05-04 오전 12:00:00	월요일

## • For loop 문을 이용한 커서문

### Basic loop

Declare

커서 선언;

Begin

커서 오픈;

Loop

커서 패치;

End loop;

커서 닫기;

End;

### for loop

Declare

커서 선언;

Begin

패치

For 레코드이름 in 커서이름 loop

End loop;

End;

Ex) 문제76번을 for loop 문으로 바꿔보세요.

declare

```

cursor emp_cursor is
    select e.ename, d.loc
    from emp e, dept d
    where e.deptno = d.deptno;

begin
    for emp_record in emp_cursor loop

        update emp
        set loc = emp_record.loc
        where ename = emp_record.ename;

    end loop;
    commit;

end;
/

```

==> emp\_cursor 에서 emp\_record으로 패치가 된다.



커서에 컬럼의 이름과 개수에 따라 방의 크기가 결정된다.

**문제79. 이름, 월급, 급여등급(grade)를 출력하시오.**

```

SELECT e.ename, e.sal, s.grade
FROM EMP e, SALGRADE s
WHERE e.sal BETWEEN s.LOSAL AND s.HISAL;

```

**문제80. 사원테이블에 grade라는 컬럼을 추가하고 해당 사원의 급여등급으로 값을 갱신하는 PLSQL을 프로그래밍을 작성하시오.(for loop)**

```

ALTER TABLE EMP
ADD grade NUMBER(10);

declare

cursor emp_cursor is
    select e.ename, e.sal, s.grade
    from emp e, salgrade s
    where e.sal between s.losal and s.hisal;

begin
    for emp_record in emp_cursor loop

        update emp
        set grade = emp_record.grade
        where ename = emp_record.ename;

    end loop;
    commit;

end;
/
SELECT ename, sal, grade FROM EMP;

```

□	ENAME	SAL	GRADE
1	KING	5000	5
2	BLAKE	2850	4
3	CLARK	2450	4
4	JONES	2975	4
5	MARTIN	1250	2
6	ALLEN	1600	3
7	TURNER	1500	3
8	JAMES	950	1
9	WARD	1250	2
10	FORD	3000	4
11	SMITH	800	1
12	SCOTT	3000	4
13	ADAMS	1100	1
14	MILLER	1300	2

위의 결과를 간결하게 하는 방법.

```

begin
  for emp_record in ( select e.ename, e.sal, s.grade
                      from emp e, salgrade s
                      where e.sal between s.losal and s.hisal ) loop

    update emp
      set grade = emp_record.grade
      where ename = emp_record.ename;

  end loop;
  commit;

end;
/
SELECT ename, sal, grade FROM EMP;

```

**문제81. 우리반 테이블에 rnk라는 컬럼을 추가하고 나이 순서대로 rnk에 순위를 갱신하는 PLSQL 프로그래밍을 작성 하시오. (간결하게)**

```

ALTER TABLE EMP2
  ADD rnk NUMBER(10);

begin
  for emp2_record in ( select ename, age, dense_rank() over(order by age desc) rnk
                      from emp2 ) loop

    update emp2
      set rnk = emp2_record.rnk
      where ename = emp2_record.ename;
  end loop;
  commit;
end;
/

```

□	ENAME	AGE	RNK
1	김원섭	26	5
2	김광록	27	4
3	유혜린	26	5
4	이유진	25	6
5	윤진민	27	4
6	백광훈	26	5
7	장은희	24	7
8	김영토	28	3
9	차호성	28	3
10	방송준	26	5
11	이한새	25	6
12	이근호	28	3
13	신영근	25	6
14	신현수	27	4

문제82. (점심시간)통신사를 물어보게하고, 통신사를 입력하면 해당 통신사인 학생들의 정보가 출력되게 하시오.

```
accept p_telecom prompt '통신사 입력'
```

```
begin
```

```
  for emp2_record in (select ename, age, major
                      from emp2
```

```
                        where telecom = lower('&p_telecom')) loop
```

```
    dbms_output.put_line(emp2_record.ename || ' ' || emp2_record.age || ' ' || emp2_record.major);
```

```
  end loop;
```

```
end;
```

```
/
```

```
SQL> @p82.sql
통신사 입력sk
김원섭 26 컴퓨터공학
이유진 25 시스템경영공학부
윤진민 27 경제학
김대경 27 통계학
김동윤 28 경영학
이동환 25 컴퓨터정보시스템과
김건태 28 화학공학
한지윤 24 경영학

PL/SQL 처리가 정상적으로 완료되었습니다.
```

문제83. 부서번호를 물어보게하고 부서번호를 입력하면 해당 부서번호인 직원들의 이름과 월급과 부서번호가 아래와 같이 출력 하도록 하세요.

부서번호를 입력하세요 10

□	ENAME	SAL	DEPTNO
1	KING	5000	10
2	CLARK	2450	10
3	MILLER	1300	10

```
accept p_deptno prompt '부서번호 입력'
```

```

begin
  for emp_record in ( select ename, sal, deptno
                      from emp
                      where deptno = 10 ) loop
    dbms_output.put_line(emp_record.ename || ' ' || emp_record.sal || ' ' || emp_record.deptno );
  end loop;

end;
/

```

```

SQL> @p83.sql
부서번호 입력10
KING 5000 10
CLARK 2450 10
MILLER 1300 10

```

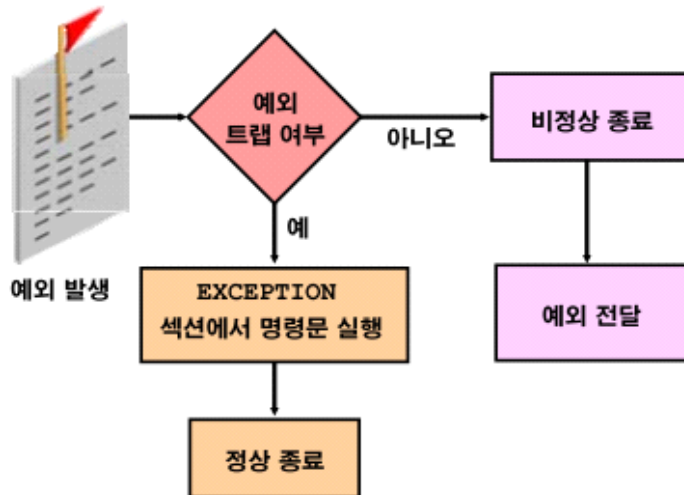
## 7. 예외 처리

2018년 5월 3일 목요일    오후 2:14

### 예외처리를 해야하는 이유?

- 프로그램 오류에 대한 사용자의 눈높이를 맞춰주기 위해서
- 프로그램이 비정상적으로 처리되어서 잘못된 결과를 출력하지 않도록 하기위해서

### 예외 처리



### 예외의 종류 3가지

1. 오라클에서 미리 정의한 예외
2. 오라클에서 미리 정의하지 않은 예외
3. 사용자 정의 예외

Ex) 사원번호를 물어보게 하고 사원번호를 입력하면 해당 사원의 월급이 출력되게 하는 PLSQL을 작성 하시오.

```
accept p_empno prompt '사원번호를 입력'
```

```
declare  
  v_sal emp.sal%type;
```

```
begin  
  select sal into v_sal  
  from emp  
  where empno = &p_empno;  
  
  dbms_output.put_line(v_sal);  
end;  
/
```

```

SQL> @except1.sql
사원번호를 입력7788
3000
PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @except1.sql
사원번호를 입력3001
declare
~
1 행에 오류:
ORA-01403: 데이터를 찾을 수 없습니다.
ORA-06512: 5행

```

문제84. 사원번호를 물어보게 하고 사원번호를 입력하면 해당 사원의 월급이 출력되게 하는 PLSQL을 작성하는데 없는 사원번호를 입력하면 해당 사원은 없습니다 라는 메시지가 출력되게 하시오.

```
accept p_empno prompt '사원번호를 입력'
```

```
declare
v_sal emp.sal%type;
```

```
begin
select sal into v_sal
from emp
where empno = &p_empno;

dbms_output.put_line(v_sal);
```

exception

```
when no_data_found then // no_data_found : 오라클에서 미리 만들어 놓은 예외
```

```
dbms_output.put_line('해당 사원은 없습니다');
```

```
end;
```

```
/
```

```

SQL> ed p84.sql

SQL> @p84.sql
사원번호를 입력1111
해당 사원은 없습니다
PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제85. 10번 부서번호의 토탈월급을 출력하시오.

```

Select sum(sal)
from emp
where deptno = 10;

```

문제86. 부서번호를 물어보게하고 부서번호를 입력하면 해당 부서번호의 토탈월급이 출력되는 PLSQL 프로그램을 작성하는데 없는 부서번호를 입력하면 해당 부서번호는 없습니다 라는 메시지가 출력되게하시오.

```
accept p_deptno prompt '부서번호 입력'
```

```
declare
v_sumsal number(10);
```

```
begin
```

```

select sum(sal) into v_sumsal
  from emp
 where deptno = &p_deptno
group by deptno;      <-- 묶어줘야지 된다.

dbms_output.put_line ( v_sumsal);

exception
  when no_data_found then

    dbms_output.put_line('해당 사원은 없습니다');

end;
/

```

```

SQL> @p86.sql
부서번호 입력10
8750

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p86.sql
부서번호 입력40
해당 사원은 없습니다

PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제87. 숫자 2개를 아래와 같이 각각 물어보게하고 첫번째 숫자를 두번째 숫자가 나눈 값이 출력되게 하는 PLSQL을 작성하시오.

첫 번째 숫자를 입력하세요 10

두 번째 숫자를 입력하세요 2

10을 2로 나눈 값은 5입니다.

```

-----
accept p_num1 prompt '첫 번째 숫자 입력'
accept p_num2 prompt '두 번째 숫자 입력'

declare
  v_num1 number(10) := &p_num1;
  v_num2 number(10) := &p_num2;

begin
  dbms_output.put_line ( v_num1 || '을' || v_num2 || '로 나눈 값은' || v_num1/v_num2 || '입니다. ');
end;
/

```

문제88. 문제87번을 다시 수행하는데 두번째 숫자를 0으로 입력하면 에러가 나는데 아니라 0으로 나눌 수 없습니다 라는 메시지가 출력되게 하시오.

/\* 프로그램 설명 : 숫자 2개를 물어보게하고 첫번째 숫자를 두번째 숫자가 나눈 값을 출력 (예외처리) \*/

/\* 프로그램 생성자 : 김원섭 \*/

```

accept p_num1 prompt '첫 번째 숫자 입력'
accept p_num2 prompt '두 번째 숫자 입력'

```



```

declare
  v_num1 number(10) := &p_num1;
  v_num2 number(10) := &p_num2;

begin
  dbms_output.put_line ( v_num1 || '을' || v_num2 || '로 나눈 값은' || v_num1/v_num2 || '입니다. ');

exception
  when zero_divide then
    dbms_output.put_line('0을 나눌 수 없음');

end;
/

```

```

SQL> @p88.sql
첫 번째 숫자 입력5
두 번째 숫자 입력0
0을 나눌 수 없음
PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제89. 부서번호를 물어보게하고 부서번호를 입력하면 해당 부서번호인 직원들의 이름과 월급과 부서번호가 아래와 같이 출력 하도록 하세요.(없는번호는 해당 번호가 없습니다 출력)

accept p\_deptno prompt '부서번호 입력'

```

declare
  v_deptno emp.deptno%type :=&p_deptno;
  v_exception exception; --v_exception 이라는 예외를 선언
  v_cnt number(10);

begin

  select count(*) into v_cnt
    from emp
   where deptno = v_deptno;

  if v_cnt = 0 then
    raise v_exception;
  else
    for emp_record in ( select ename, sal, deptno
                        from emp
                        where deptno = v_deptno ) loop
      dbms_output.put_line(emp_record.ename || ' ' || emp_record.sal || ' ' || emp_record.deptno );
    end loop;
  end if;

exception
  when v_exception then
    dbms_output.put_line ('해당 부서번호는 없음');

end;
/

```

*v\_cnt = 0이면 v\_exception을 일으켜라*

```

SQL> @p89.sql
부서번호 입력10
KING 5000 10
CLARK 2450 10
MILLER 1300 10

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p89.sql
부서번호 입력50
해당 부서번호는 없음

PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제90. 통신사를 물어보게하고, 통신사를 입력하면 해당 통신사인 학생들의 정보가 출력되게 하는데 없는 통신사를 넣으면 '해당 통신사는 없습니다' 라는 메시지가 출력되게 하시오.

```
accept p_telecom prompt '통신사 입력'
```

```
declare
```

```
    v_telecom emp2.telecom%type := '&p_telecom';
```

```
    v_exception exception;
```

```
    v_cnt number(10);
```

```
begin
```

```
    select count(*) into v_cnt
```

```
        from emp2
```

```
        where telecom = v_telecom;
```

```
    if v_cnt = 0 then
```

```
        raise v_exception;
```

```
    else
```

```
        for emp2_record in (select ename, age, major
```

```
                             from emp2
```

```
                             where telecom = lower('&p_telecom')) loop
```

```
            dbms_output.put_line(emp2_record.ename || ' ' || emp2_record.age || ' ' || emp2_record.major);
```

```
        end loop;
```

```
    end if;
```

```
exception
```

```
    when v_exception then
```

```
        dbms_output.put_line ('해당 통신사 없음');
```

```
end;
```

```
/
```

```

SQL> @p90.sql
통신사 입력sk
김원섭26 컴퓨터공학
이유진25 시스템경영공학부
유진민27 경제학
김대경27 통계학
김동훈28 경영학
유동환25 컴퓨터정보시스템과
김건태28 화학공학
한지훈24 경영학

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> @p90.sql
통신사 입력qq
해당 통신사 없음

PL/SQL 처리가 정상적으로 완료되었습니다.

```

## 8. 사용자 정의 함수

2018년 5월 3일 목요일    오후 4:11

### 함수의 종류 2가지

1. 오라클 내장 함수  
: 문자, 숫자, 날짜, 변환, 일반, 그룹함수, 분석함수
2. 사용자 정의 함수  
: 직접 우리 회사에 데이터 분석에 필요한 함수를 만들어서 사용할 때 필요함.

**Tip) show err // 어디에 에러가 났는지 확인**

문제91. 부서번호를 입력하면 부서위치가 출력되는 함수를 생성하시오.

```
CREATE OR REPLACE FUNCTION get_loc // 함수 이름
(p_deptno IN dept.deptno%type) <<-- (p_deptno = accept prompt 라고 생각
                                in == input )
RETURN dept.loc%TYPE // return 결과 리턴한 data유형
IS
  v_loc dept.loc%TYPE;
BEGIN
  SELECT loc INTO v_loc
  FROM DEPT
  WHERE deptno = p_deptno;
  RETURN v_loc;
END;
/
```

```
SQL> @p91.sql
함수가 생성되었습니다.
```

Dbms\_output.put 대신 return을 쓴다

```
SELECT ename, deptno, get_loc(deptno)
FROM EMP;
```

	ENAME	DEPTNO	GET_LOC(DEPTNO)
1	KING	10	NEW YORK
2	BLAKE	30	CHICAGO
3	CLARK	10	NEW YORK
4	JONES	20	DALLAS
5	MARTIN	30	CHICAGO
6	ALLEN	30	CHICAGO
7	TURNER	30	CHICAGO
8	JAMES	30	CHICAGO
9	WARD	30	CHICAGO
10	FORD	20	DALLAS
11	SMITH	20	DALLAS

	ENAME	DEPTNO	GET_LOC(DEPTNO)
1	KING	10	NEW YORK
2	BLAKE	30	CHICAGO
3	CLARK	10	NEW YORK
4	JONES	20	DALLAS
5	MARTIN	30	CHICAGO
6	ALLEN	30	CHICAGO
7	TURNER	30	CHICAGO
8	JAMES	30	CHICAGO
9	WARD	30	CHICAGO
10	FORD	20	DALLAS
11	SMITH	20	DALLAS
12	SCOTT	20	DALLAS
13	ADAMS	20	DALLAS
14	MILLER	10	NEW YORK

문제92. 부서번호를 입력하면 해당 부서번호의 토탈 월급이 출력되는 함수를 생성하고 아래와 같이 실행하시오.

```
Select ename, deptno, deptno_sum(deptno)
from emp;
```

```
create or replace function deptno_sum
( p_deptno in emp.deptno%type)
```

```
return emp.sal%type
is
v_sumsal emp.sal%type;
```

```
begin
SELECT sum(sal) into v_sumsal
from emp
where deptno = p_deptno;
```

```
return v_sumsal;
```

```
end;
```

```
/
```

	ENAME	DEPTNO	DEPTNO_SUM(DEPTNO)
1	KING	10	8750
2	BLAKE	30	9400
3	CLARK	10	8750
4	JONES	20	10875
5	MARTIN	30	9400
6	ALLEN	30	9400
7	TURNER	30	9400
8	JAMES	30	9400
9	WARD	30	9400
10	FORD	20	10875
11	SMITH	20	10875
12	SCOTT	20	10875
13	ADAMS	20	10875
14	MILLER	10	8750

문제93. (마지막) gauss 테이블의 데이터를 아래와 같이 검색해서 아래와 같이 결과가 출력되는 함수를 생성하시오.

```
Select num1, num2, func_gauss(num1, num2)
```

```

from gauss;
1  10  55
1  20 210
1  30 465
.
.

```

	NUM1	NUM2	FUNC_GAUSS(NUM1,NUM2)
1	1	10	55
2	1	20	210
3	1	30	465
4	1	40	820
5	1	50	1275
6	1	60	1830

```

create or replace function func_gauss
( p_num1 in number , p_num2 in number )

return gauss.num1%type
is
    v_result gauss.num1%type;

begin
    select (p_num1 + p_num2) * (p_num2/2) into v_result
    from dual;

    return v_result;
end;
/

```

문제94. log\_example 테이블을 아래와 같이 생성하고 num1 이라는 컬럼에 loop문을 이용해서 1부터 10까지 데이터를 입력하세요.

```

Create table log_example
(num1 number(10), num2 number(10) );

```

```

set verify off
set serveroutput on

```

```

begin
    for i in 1 .. 10 loop
        insert into log_example(num1)
        values(i);
    end loop;
end;
/

```

	NUM1	NUM2
1	1 (null)	
2	2 (null)	
3	3 (null)	
4	4 (null)	
5	5 (null)	
6	6 (null)	
7	7 (null)	
8	8 (null)	
9	9 (null)	
10	10 (null)	

10	10 (null)
----	-----------

문제95. (수학문제 19p) 예제 3번 구간  $[4,10]$  에서  $y=\log_3(x-1)$ 의 최대값과 최소값을 각각 구하시오.  
위의 수학문제를 풀기위한 함수를 생성하세요.

```

Select num1, log_value(num1)
  from log_example;

create or replace function log_value
  (p_num1 in log_example.num1%type )

return number
is
  v_value number(10,8);

begin
  v_value := log(3, p_num1 - 1);

  return v_value;

end;
/

```

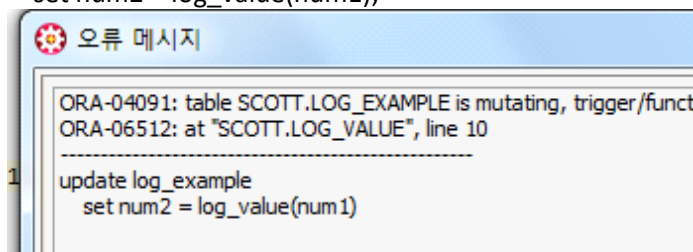
문제96. 문제95번에서 만든 log\_value 함수를 이용해서 log\_example 테이블의 num2의 값을 갱신하는 Update 문을 작성하세요.

	NUM1	NUM2
1		2 (null)
2		3 (null)
3		4 (null)
4		5 (null)
5		6 (null)
6		7 (null)
7		8 (null)
8		9 (null)
9		10 (null)

```

update log_example
  set num2 = log_value(num1);

```



같은 테이블에서는 데이터를 이동할 수 없다

문제97. 수학문제 p.19 문제4번인 다음 수를 작은 것 부터 차례대로 나열하는 문제를 풀기위해서 아래의 테이블을 생성하고 데이터를 입력하시오.

1,  $\log_3 7$ ,  $\log_9 100$

Num1	Num2
1	

Log(3,7)	
Log(9,100)	

```
create table math_ex
( num1 varchar2(50), num2 number(10,7) );
```

```
INSERT INTO math_ex(num1)
VALUES('1');
```

```
INSERT INTO math_ex(num1)
VALUES('log(3,7)');
```

```
INSERT INTO math_ex(num1)
VALUES('log(9,100)');
```

	NUM1	NUM2
1	1	(null)
2	log(3,7)	(null)
3	log(9,100)	(null)

문제98. 문제97번의 테이블의 num1의 값이 계산되는 함수를 생성하세요..

```
create or replace function log_value
(p_num1 in math_ex.num1%type )
```

```
return number
is
```

```
  v_stmt  varchar2(100);      -- SQL문장을 넣을 곳.
  v_value  number(10,8);
```

```
begin
```

```
  v_stmt := 'select ' || p_num1 || ' from dual';
```

```
  execute immediate v_stmt into v_value; -- 즉시 실행해서 v_stmt의 SQL 값을 v_value에 넣어라.
```

```
return v_value;
```

```
end;
/
```

문제99. 아래의 SQL을 다시 수행하는데 log\_value(num1)이 높은 순서대로 순위가 부여되게 순위 컬럼을 출력하세요.

```
select num1, log_value(num1),
       rank() over(order by log_value(num1) desc) rnk
from math_ex;
```

문제100. 아래의 테이블을 생성하세요.

col1	col2
Max(sal)	
Min(sal)	
Sum(sal)	
Avg(sal)	

```
Create table emp100
```



```
(col1 varchar2(20), col2 number(10,7) );
INSERT INTO EMP100(col1)
VALUES( 'MAX(sal)');

INSERT INTO EMP100(col1)
VALUES( 'min(sal)');

INSERT INTO EMP100(col1)
VALUES( 'sum(sal)');

INSERT INTO EMP100(col1)
VALUES( 'avg(sal)');
```

```
select * FROM emp100;
```

	COL1	COL2
1	MAX(sal)	(null)
2	min(sal)	(null)
3	sum(sal)	(null)
4	avg(sal)	(null)

**문제101.** emp100 테이블을 select 해서 아래의 결과가 출력되게 하는 함수를 생성하세요.

```
Select col1, func_100(col1)
from emp100;
```

col1	col2
Max(sal)	5000
Min(sal)	700
Sum(sal)	8950
Avg(sal)	2975

```
create or replace function func_100
(p_col1 in emp100.col1%type)
```

```
return number
```

```
is
```

```
v_stmt varchar2(100);
```

```
v_col2 number(10);
```

```
begin
```

```
v_stmt := 'select' || p_col1 || 'from dual';
```

```
execute immediate v_stmt into v_col2;
```

```
return v_value;
```

```
end;
```

```
/
```

**문제102.** emp100 테이블의 col2컬럼의 데이터가 아래와 같이 갱신되게 하시오.

```
UPDATE emp100
SET col2 = func_100(col1);
```

**문제103.** SCOTT 유저가 가지고 있는 함수 리스트를 조회 하시오.

```
Select * from user_objects
WHERE object_type = 'FUNCTION';
```

	OBJECT_NAME	SUBOBJECT_NAME	OBJECT_ID	DATA_OBJECT_ID	OBJECT_TYPE
1	LOG_VALUE	(null)	74875	(null)	FUNCTION
2	GET_SUMSAL	(null)	74870	(null)	FUNCTION
3	GET_LOC	(null)	74869	(null)	FUNCTION
4	FUNC_GAUSS	(null)	74871	(null)	FUNCTION
5	FUNC_100	(null)	74880	(null)	FUNCTION
6	DEPTNO_SUM	(null)	74872	(null)	FUNCTION

**문제104. 생성한 함수의 스크립트를 확인하시오.**

```
Select * from user_source
WHERE NAME = 'LOG_VALUE';
```

	NAME	TYPE	LINE	TEXT
1	LOG_VALUE	FUNCTION	1	function log_value
2	LOG_VALUE	FUNCTION	2	(p_num1 in math_ex.num1%type )
3	LOG_VALUE	FUNCTION	3	
4	LOG_VALUE	FUNCTION	4	return number
5	LOG_VALUE	FUNCTION	5	is
6	LOG_VALUE	FUNCTION	6	v_stmt varchar2(100);
7	LOG_VALUE	FUNCTION	7	v_value number(10,8);
8	LOG_VALUE	FUNCTION	8	
9	LOG_VALUE	FUNCTION	9	begin
10	LOG_VALUE	FUNCTION	10	v_stmt := 'select '    p_num1    ' fr
11	LOG_VALUE	FUNCTION	11	execute immediate v_stmt into v_
12	LOG_VALUE	FUNCTION	12	
13	LOG_VALUE	FUNCTION	13	return v_value;
14	LOG_VALUE	FUNCTION	14	
15	LOG_VALUE	FUNCTION	15	end;

**문제105. SCOTT 유저가 생성한 함수들을 모두 삭제하시오.**

```
SELECT 'drop function ' || object_name || ';'
FROM user_objects
WHERE object_type = 'FUNCTION';
```

## 9. 프로시저

2018년 5월 4일 금요일    오후 2:11

1. Anonymous PLSQL    <- 이름이 없다
2. 함수    <- 이름이 있다
3. 프로시저    <- 이름이 있다

Anonymous PLSQL 은 @이름을 써서

```
C:\WUsers\Witwill>
```

에 데이터를 저장한것이고,

함수랑 프로시저는 database에 PLSQL 코드를 저장한다.

### • 프로시저와 함수의 차이

프로시저	함수
Return 절이 <b>없다</b> .	Return 절이 <b>있다</b> .
결과를 반드시 <b>출력 하</b> <b>지 않아도됨</b> .	결과를 반드시 <b>출력 해</b> <b>야함</b> .

문제106. 문제6번을 프로시저로 작성하시오.(사원번호를 물어보게하고 사원번호를 입력하면 해당사원의 직업

이 출력되게 PL/SQL을 작성하시오.)

```
set serveroutput on
```

```
accept p_empno prompt '사원번호를 입력 하세요'
```

```
create or replace procedure pro106
( p_empno in emp.empno%type )
is
  v_job varchar2(20);
```

```
begin
  select job into v_job
  from emp
  where empno = &p_empno;
```

```
  dbms_output.put_line('해당 사원의 직업은' || v_job);
```

```
end;
/
```

```
Exec pro106(7788);
```

```
프로시저가 생성되었습니다.
```

```
SQL> exec pro106<7788>;  
해당 사원의 직업은ANALYST
```

```
PL/SQL 처리가 정상적으로 완료되었습니다.
```

문제107. 아래와 같이 이름을 입력해서 프로시저를 실행하면 해당 사원의 월급이 출력되게 하는 프로시저를

작성하시오. (이름을 소문자로 넣어도 가능하게)

```
Exec get_sal('scott');
```

3000 입니다.

```
create or replace procedure get_sal  
(p_ename in emp.ename%type)  
is  
    v_sal number(10);
```

```
begin  
    select sal into v_sal  
    from emp  
    where ename = upper(p_ename);
```

```
dbms_output.put_line(v_sal || '입니다');
```

```
end;
```

```
/
```

```
SQL> @p107.sql
```

```
프로시저가 생성되었습니다.
```

```
SQL> exec get_sal('scott');  
3000입니다
```

문제108. 아래와 같이 이름을 입력해서 프로시저를 실행하면 해당 사원의 월급이 출력되게 하는 프로시저를

작성하시오.

```
Exec get_sal_rank('scott');
```

```
create or replace procedure get_sal_rank  
(p_ename in emp.ename%type)  
is  
    v_sal_rank number(10);  
  
begin  
    select rn into v_sal_rank  
    from( select ename, dense_rank() over(order by sal desc) rn  
    from emp)  
    where ename = upper(p_ename);
```

```
dbms_output.put_line(v_sal_rank || ' 등 입니다');
```

```

end;
/
SQL> exec get_sal_rank('scott');
2 등 입니다
PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제109. 아래와 같이 프로시저를 수행하면 1부터 10까지 출력되게하는 프로시저를 생성하십시오.

```

Exec pro109;
1
2
3
.
.
9
10
create or replace procedure pro109
is

```

```

begin
  for i in 1 .. 10 loop
    dbms_output.put_line(i);
  end loop;
end;
/

```

```

SQL> exec pro109;
1
2
3
4
5
6
7
8
9
10
PL/SQL 처리가 정상적으로 완료되었습니다.

```

문제110. 아래와 같이 구구단의 해당 단이 출력되는 프로시저를 생성하세요.

```

Exec gugu_dan(2); 2단출력
Exec gugu_dan(3); 3단출력

```

문제111. demobld를 돌리고 grade 컬럼을 추가한 후에 해당 사원의 급여등급으로 값을 갱신하는 PLSQL을

코드인 문제 72번을 프로시저 생성해서 수행하세요.

```

Alter table emp
add grade varchar2(20);

```

```

Update emp
set grade = null;

```

create or replace procedure pro111      만 붙여주면된다

is

```
v_sal emp.sal%type;  
v_ename emp.ename%type;  
v_grade emp.grade%type;
```

```
cursor emp_cursor is  
    select ename, sal  
    from emp;
```

```
begin
```

```
    open emp_cursor;
```

```
    loop
```

```
        fetch emp_cursor into v_ename, v_sal;  
        exit when emp_cursor%notfound;
```

```
        v_grade := case when v_sal >= 3000 then 'A'  
            when v_sal >= 2000 then 'B'  
            when v_sal >= 1500 then 'C'  
            when v_sal >= 1000 then 'D'  
            else 'F' end;
```

```
        update emp  
            set grade = v_grade  
            where ename = v_ename;
```

```
    commit;
```

```
    end loop;
```

```
    close emp_cursor;
```

```
end;
```

```
/
```

	ENAME	GRADE
1	KING	A
2	BLAKE	B
3	CLARK	B
4	JONES	B
5	MARTIN	D
6	ALLEN	C
7	TURNER	C
8	JAMES	F
9	WARD	D
10	FORD	A
11	SMITH	F
12	SCOTT	A
13	ADAMS	D
14	MILLER	D

문제112. 사원번호를 물어보게하고 사원번호를 입력했을때 해당 사원의 월급을 출력되는 문제84번을 프로시

저로 생성하고 수행하시오.

**Exec pro112(3030);**

**해당 사원은 없습니다.**

```
create or replace procedure pro112
(p_empno in emp.empno%type)
is
    v_sal emp.sal%type;

begin
    select sal into v_sal
    from emp
    where empno = p_empno;

    dbms_output.put_line(v_sal);

exception
    when no_data_found then

        dbms_output.put_line('해당 사원은 없습니다');

end;
```

```
SQL> @p112.sql

프로시저가 생성되었습니다.

SQL> exec pro112(3030);
해당 사원은 없습니다

PL/SQL 처리가 정상적으로 완료되었습니다.
```

문제113. 아래의 두개의 함수식으로 함수를 각각 만들고,

$$f(x) = \log_a(x-m) + n$$

$$g(x) = a^{x-n} + m$$

**Select f(g(1)) from dual; 의 결과가 1인지 확인해보시오.**

```
-----
create or replace function fx
(x in number)
return number
is
    fx_a1 number;
```

```
begin
    fx_a1 := log(10,x-2) + 3;

    return fx_a1;
end;
```

```
-----
create or replace function gx
(x in number)
return number
```

is

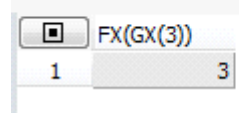
```

gx_b1 number;

begin
  gx_b1 := power(10,x-3) + 2;

  return gx_b1;
end;
/
SELECT fx(gx(3)) FROM dual;

```



FX(GX(3))
3

**문제114. 직업, 부서번호, 직업별 부서번호별 토달월급을 출력하시오.**

```

select job, sum(decode(deptno,10,sal,0) ) "10",
           sum(decode(deptno,20,sal,0) ) "20",
           sum(decode(deptno,30,sal,0) ) "30",
from emp
group by job;

```

**문제115. 아래의 SQL문이 출력되는 프로시저를 생성하시오.**

```

sum(decode(deptno,10,sal,0) )
-----
create or replace procedure pro115
as
  v_stmt varchar2(200);

begin
  v_stmt := 'sum(decode(deptno,10,sal,0) )';

  dbms_output.put_line (v_stmt);

end;
/

```

**문제116. 문제115번 프로시저에 for loop문을 추가해서 아래의 SQL문이 출력되는 프로시저를 생성하시오.**

```

sum(decode(deptno,10,sal,0) ),

```



```
sum(decode(deptno,20,sal,0) ),  
sum(decode(deptno,30,sal,0) )
```

---

```
create or replace procedure pro116  
is  
    v_stmt varchar2(200);  
    v_cnt number(10) := 0;  
  
begin  
    loop  
        v_cnt := v_cnt +10;  
  
        v_stmt := v_stmt || 'sum(decode(deptno, ' || v_cnt || ', sal,0) )';  
  
        exit when v_cnt = 30;  
    end loop  
  
    dbms_output.put_line(v_stmt);  
  
end;  
/
```

문제117. 문제116번 프로시저가 아래와 같이 출력될 수 있도록 코드를 수정하시오.

```
select  
sum(decode(deptno,10,sal,0) ),  
sum(decode(deptno,20,sal,0) ),  
sum(decode(deptno,30,sal,0) )
```

---

```
create or replace procedure pro116  
is  
    v_stmt varchar2(200) := 'select';  
    v_cnt number := 0;  
  
begin  
    loop  
        v_cnt := v_cnt +10;  
  
        v_stmt := v_stmt || 'sum(decode(deptno,' || v_cnt || ',sal,0) )';
```

```

        exit when v_cnt = 30;
    end loop;

    dbms_output.put_line(v_stmt);

end;
/

```

**문제118. 문제117번 프로시저에서 뒤에 ,가 없어지게 하시오.**

```

create or replace procedure pro116
is
    v_stmt varchar2(200) := 'select';
    v_cnt number := 0;

begin
    loop
        v_cnt := v_cnt + 10;

        v_stmt := v_stmt || 'sum(decode(deptno,' || v_cnt || ',sal,0) ),';

        exit when v_cnt = 30;
    end loop;

    v_stmt := rtrim(v_stmt, ',');

    dbms_output.put_line(v_stmt);

end;
/

```

**문제119. 문제118번 프로시저가 아래와 같이 출력될 수 있도록 코드를 수정하시오.**

```

select
    sum(decode(deptno,10,sal,0) ),
    sum(decode(deptno,20,sal,0) ),
    sum(decode(deptno,30,sal,0) )
from emp

```

-----

```

create or replace procedure pro119
is
    v_stmt varchar2(200) := 'select';

```

```

v_cnt number := 0;

begin
  loop
    v_cnt := v_cnt + 10;

    v_stmt := v_stmt || 'sum(decode(deptno,' || v_cnt || ',sal,0) ),' ;

    exit when v_cnt = 30;
  end loop;

  v_stmt := v_stmt || 'from emp';

  dbms_output.put_line(v_stmt); <-- emp 뒤에 ; 붙이면 안된다.!!!

end;
/

```

문제120. 문제119번 프로시저를 아래와 같이 출력될 수 있도록 코드를 수정하시오.

```

select
  sum(decode(deptno,10,sal,0) ) "10",
  sum(decode(deptno,20,sal,0) ) "20",
  sum(decode(deptno,30,sal,0) ) "30"
from emp
-----
create or replace procedure pro120
is
  v_stmt varchar2(200) := 'select';
  v_cnt number := 0;

begin
  loop
    v_cnt := v_cnt + 10;

    v_stmt := v_stmt || 'sum(decode(deptno,' || v_cnt || ',sal,0) )' || ' ' || v_cnt || ' ' || ' ';

    exit when v_cnt = 30;
  end loop;

  v_stmt := v_stmt || 'from emp';

```

```

        dbms_output.put_line(v_stmt);
    end;
/

```

위의 결과를 복사 해서 cmd 창에 붙여넣어 보세요.

```

SQL> select sum(decode(deptno,10,sal,0) ) "10",
              10              20              30              40
-----
8750      10875      9400      0

```

**문제121.** 문제120번에서 완성한 아래의 SQL을 프로시저에서 실행되게 하시오.

```

create or replace procedure pro121
    (v_result out sys_refcursor)    // out = 프로시저에 대한 결과를 컵(v_result)에다가
    담겠다.
is
    v_stmt varchar2(200) := 'select';
    v_cnt number := 0;

begin
    loop
        v_cnt := v_cnt +10;

        v_stmt := v_stmt || ' sum(decode(deptno,' || v_cnt || ',sal,0) )' || ' ' || v_cnt || ' ' || ',';

        exit when v_cnt = 30;
    end loop;

    v_stmt := rtrim(v_stmt, ',');
    v_stmt := v_stmt || 'from emp';

    dbms_output.put_line(v_stmt);

    open v_result for v_stmt;  // SQL문장의 결과를 담을 메모리를 v_result로 열겠다.
end;
/

```

실행하는 방법

```
variable v_result2 refcursor;    // 변수(v_result2) 를 선언하고 refcursor를 실행
```

```
exec pro121(:v_result2);
```

```
print v_result2;
```

문제122. 아래의 결과를 출력하는 프로시저를 생성하시오.

문제123. 통신사를 출력하는데 중복 제거해서 출력하시오.

```
select distinct telecom
from emp2;
```

문제124. 아래와 같이 프로시저를 생성하시오.

AGE	SK	LG	CJH	KT
25	2	1	0	2
28	2	2	1	3
29	0	1	0	0
26	1	2	0	2
24	1	0	0	2
32	0	1	0	0
27	2	2	0	3

```
create or replace procedure pro124
( v_result out sys_refcursor)
as
    v_stmt varchar2(2000) := 'select age, ' ;

begin

    for v_cnt in ( select distinct telecom
                    from emp2 ) loop

        v_stmt := v_stmt || ' sum(decode( telecom , ' ||
            ''' || v_cnt.telecom || ''' || ', 1, 0) ) ' ||
            ' as ' || v_cnt.telecom || ',';

    end loop;

    v_stmt := rtrim( v_stmt, ',');
    v_stmt := v_stmt ||' from emp2 group by age';

    dbms_output.put_line (v_stmt );
```

```
open v_result for v_stmt ;
```

```
end;
```

```
/
```