/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

조건문

- 조건에 따라 실행하는 구문이 다를 경우 사용.

- IF 문과 CASE문 이 있다.

- IF 문 구문

- [조건이 1개인 경우]

IF 조건 THEN 처리구문 END IF;

- [참인경우 거짓인 경우의 두 조건]

IF 조건 THEN 처리구문\_1

ELSE 처리구문\_2

END IF

- [조건이 여러개인 겨우]

IF 조건\_1 THEN 처리구문\_1

ELSIF 조건\_2 THEN 처리구문\_2

ELSIF 조건\_3 THEN 처리구문\_3

...

ELSE 처리구문\_N

END IF

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

declare

v\_num1 binary\_integer := 10;

v\_num2 binary\_integer := 0;

begin

if v\_num1 > v\_num2 then

dbms\_output.put\_line('num1이 크다');

elsif v\_num1 = v\_num2 then

dbms\_output.put\_line('num1과 num2는 같다.');

else

dbms\_output.put\_line('num2가 크다');

end if;

end;

/

-- TODO : 직원 ID가 110 인 직원의 급여(salary)를 조회한 뒤 5000 미만이면 'LOW', 5000이상 10000이하면 'MIDDLE', 1000 초과이면 'HIGH'를 출력한다.

declare

v\_salary emp.salary%type;

v\_grade varchar(6);

begin

select salary

into v\_salary

from emp

where emp\_id = 110;

if v\_salary < 5000 then

v\_grade := 'LOW';

elsif v\_salary <10000 then

v\_grade := 'MIDDLE';

else

v\_grade := 'HIGH';

end if;

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

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

end;

/

-- TODO: 직원 ID가 100 인 직원의 SALARY와 COMM\_PCT를 조회한다. COMM\_PCT가 NULL이 아니면 SALARY + SALARY \* COMM\_PCT의 값을 출력하고

--NULL 이면 COMM\_PCT가 없음과 SALARY만 출력한다.

--직원ID 100과 145 두명을 해본다.

declare

v\_salary emp.salary%type;

v\_comm\_pct emp.comm\_pct%type;

begin

select salary, comm\_pct

into v\_salary, v\_comm\_pct

from emp

where emp\_id = 145;

if v\_comm\_pct is not null then

dbms\_output.put\_line(v\_salary+v\_salary\*v\_comm\_pct);

else

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

end if;

end;

/

-- TODO: emp\_id를 bind변수로 받아서 그 직원이 커미션이 있는지 조회후(emp.comm\_pct) 커미션이 있는 직원이면 commission 테이블에 emp\_id와

--실행시점 일시와 커미션(salary \* comm\_pct)을 저장한다.

--커미션 저장 테이블

create table commission(

emp\_id number(6),

curr\_date date,

commission number

);

declare

v\_emp\_id emp.emp\_id%type := :id;

v\_comm\_pct emp.comm\_pct%type;

v\_salary emp.salary%type;

begin

select comm\_pct, salary

into v\_comm\_pct, v\_salary

from emp

where emp\_id = v\_emp\_id;

if v\_comm\_pct is not null then

insert into commission values(v\_emp\_id, sysdate, v\_salary\*v\_comm\_pct);

dbms\_output.put\_line('커미션 등록');

else

dbms\_output.put\_line('comm\_pct가 null임');

end if;

end;

/

select \* from commission;

select emp\_id, comm\_pct, salary from emp;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CASE 문 : SQL의 CASE 문과 (거의)동일

[유형 1]

CASE 표현식

WHEN 결과1 THEN 처리구문\_1;

WHEN 결과2 THEN 처리구문\_2;

...

ELSE 기타 처리구문;

END CASE;

[유형 2]

CASE WHEN 표현식1 THEN 처리구문\_1;

WHEN 표현식2 THEN 처리구문\_2;

...

ELSE 기타 처리구문;

END CASE;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

declare

v\_num binary\_integer := &num;

v\_result varchar2(20);

begin

case v\_num

when 10 then

v\_result := '십';

when 20 then

v\_result := '이십';

else

v\_result := '기타';

end case;

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

end;

/

-- TODO : 직원 ID가 110 인 직원의 급여(salary)를 조회한 뒤 5000 미만이면 'LOW', 5000이상 10000이하면 'MIDDLE', 1000 초과이면 'HIGH'를 출력한다.

-- CASE 문을 이용해 작성

declare

v\_salary emp.salary%type;

v\_emp\_id emp.emp\_id%type := &emp\_id;

begin

select salary

into v\_salary

from emp

where emp\_id = v\_emp\_id;

case when v\_salary<5000 then dbms\_output.put\_line('LOW');

when v\_salary<10000 then dbms\_output.put\_line('MIDDLE');

else dbms\_output.put\_line('HIGH');

end case;

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

end;

/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

반복문

- 특정 구문을 반복해서 실행하는 구문

- LOOP, WHILE, FOR 세가지 구문이 있다.

- EXIT WHEN 조건

- 반복문을 멈추고 빠져나온다.

- CONTINUE WHEN 조건

- 반복문의 이하부분을 실행하지 않고 처리구문 처음으로 가서 실행한다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LOOP 문 : 단순 반복처리 > 무한반복(exit조건을 명시해야함)

구문

LOOP

반복처리할 구문

EXIT WHEN 멈출조건

END LOOP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

-- LOOP 문을 이용해 0 ~ 10까지 누적합계 구하기

declare

v\_num binary\_integer := 1;

v\_result binary\_integer :=0;

begin

loop

v\_result := v\_result + v\_num;

dbms\_output.put\_line(v\_num||' 더한 결과 : '||v\_result);

v\_num := v\_num+1;

exit when v\_num = 11;

end loop;

end;

/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WHILE 문: 조건이 참(TRUE)일 동안 반복

구문

WHILE 반복조건

LOOP

처리구문

END LOOP;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

-- WHILE 문을 이용한 0 ~ 10까지 누적합계

declare

v\_num binary\_integer := 1;

v\_result binary\_integer :=0;

begin

while v\_num != 11 --반복구문을 실행할 조건.

loop

v\_result := v\_result + v\_num;

dbms\_output.put\_line(v\_num||' 더한 결과 : '||v\_result);

v\_num := v\_num+1;

end loop;

end;

/

-- TODO: hello1, hello2, hello3 ... hello10을 출력하는 코드를 작성 (LOOP와 WHILE문으로 작성)

--loop

declare

v\_num binary\_integer := 1;

begin

loop

dbms\_output.put\_line('hello'||v\_num);

v\_num := v\_num + 1;

exit when v\_num = 11;

end loop;

end;

/

--while

declare

v\_num binary\_integer := 1;

begin

while v\_num < 11

loop

dbms\_output.put\_line('hello'||v\_num);

v\_num := v\_num + 1;

end loop;

end;

/

-- TODO: sequence t\_seq 의 값이 20이 될때 까지 조회해 출력하는 코드를 작성. (LOOP 와 WHILE문으로 작성)

drop sequence t\_seq;

create sequence t\_seq;

--loop

begin

loop

dbms\_output.put\_line(t\_seq.nextval);

exit when t\_seq.currval = 20;

end loop;

end;

/

--while

declare

v\_num binary\_integer := 1;

begin

while v\_num<20

loop

dbms\_output.put\_line(t\_seq.nextval);

v\_num := t\_seq.currval;

end loop;

end;

/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR 문: 일정 횟수를 반복할 때 사용

FOR 인덱스 IN [REVERSE] 초깃값..최종값

LOOP

반복 처리 구문

END LOOP;

-- 인덱스: 반복 시마다 1씩 증가하는 값. 참조는 가능하나 변경은 안된다.

-- REVERSE : 최종값->초깃값

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

-- FOR 문을 이용한 0 ~ 10까지 누적합계

declare

v\_result binary\_integer := 0;

begin

for i in reverse 1..10

loop

v\_result := v\_result +i;

dbms\_output.put\_line(i);

end loop;

dbms\_output.put\_line('합계: '||v\_result);

end;

/

begin

for e\_row in (select \* from emp where dept\_id = 60)

loop

dbms\_output.put\_line(e\_row.emp\_id||e\_row.emp\_name);

end loop;

end;

/

-- TODO: hello1, hello2, hello3 ... hello10을 출력하는 코드를 작성 (FOR문으로 작성)

begin

for i in 1..10

loop

dbms\_output.put\_line('hello'||i);

end loop;

end;

/

-- TODO: sequence t\_seq 의 값이 20이 될때 까지 조회해 출력하는 코드를 작성. (FOR문으로 작성)

drop sequence t\_seq;

create sequence t\_seq;

declare

v\_num binary\_integer;

begin

for i in 1..20

loop

exit when v\_num>20;

dbms\_output.put\_line(t\_seq.nextval);

v\_num := t\_seq.currval;

end loop;

end;

/

--TODO 구구단 5단을 출력하는 코드를 작성 (loop, for, while으로 작성)

--for

declare

v\_dan pls\_integer:=&dan;

begin

for i in 1..9

loop

dbms\_output.put\_line(v\_dan||' \* '|| i || ' = ' || v\_dan\*i);

end loop;

end;

/

--loop

declare

v\_num binary\_integer := 1;

begin

loop

exit when v\_num >=10;

dbms\_output.put\_line('5 \* '|| v\_num || ' = ' || 5\*v\_num);

v\_num := v\_num + 1;

end loop;

end;

/

--while

declare

v\_num binary\_integer := 1;

begin

while v\_num < 10

loop

dbms\_output.put\_line('5 \* '|| v\_num || ' = ' || 5\*v\_num);

v\_num := v\_num + 1;

end loop;

end;

/

--TODO 1 ~ 100 사이의 숫자 중에 5의 배수만 출력하도록 작성. (MOD(A, B): A를 B로 나눈 나머지 반환 함수)

begin

for i in 1..100

loop

if mod(i,5)=0 then

dbms\_output.put\_line(i);

end if;

end loop;

end;

/