PL/SQL

가. PL/SQL 이란?

PL/SQL: Oracle’s Procedural Language extension to SQL

오라클에 내장되어 있는 절차형 언어

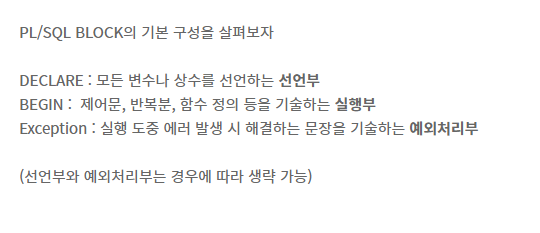
프로그램을 논리적인 블록으로 나누는 구조화된 블록 언어

변수 선언문, 조건문, 반목문(loop, while, for) 등을 지원

나. PL/SQL Block Structure

선언부(선택), 실행부(필수), 예회 처리부(선택)로 구성됨

begin 과 end 키워드는 반드시 기술



1) Declareative Section(선언부, 선택)

변수, 상수, Cursor등을 선언

2) Executable Section(실행부, 필수)

sql문장, 반복문 ,조건문 등을 실행

begin으로 시작하고 end로 끝남

3) Exception Handling Section(예회처리부, 선택)

예외에 대한 처리

다. Block type(pl/sql 블럭의 유형)

[Anonymous]

[Procedure]

[Function]



1) Anonymous (익명 블록)

이름이 없는 블록

2) Procedure(프로시저)

특정 작업을 수행할 수 있고, 이름이 있는 PL/SQL 블록

매개 변수를 입력받을 수 있음

매개 변수를 입력받을 수 있음

DB에 저장되어 반복적으로 사용할 수 있음 (저장 프로시저)

배치 작업(일괄처리) 또는 구현이 복잡한 트랜잭션을 수행하는 용도로 사용함

ex) 어떤 건에 대해서 자료를 입력하는데 기존에 자료가 있으면 insert를 하고, 기존에 자료가 있으면 update를 한다

-> count함수를 통해 레코드의 개수를 세어서 레코드 개수가 0이면 insert하고 레코드 개수가 1이면 update 한다

-> 이렇게 복잡한 sql를 프로시저 내에서 한꺼번에 처리가능 하다

3) Function(함수)

값을 계산하고 결과값을 반환하기 위해서 사용

저장 프로시저와의 차이점: 입력매개변수만 사용 할 수 있고 리턴 타입을 반드시 지정해야 함

프로시저- 입력매개변수와 출력매개변수가 있음

라. 저장프로시저

1) Stored Procedure(SP, 저장 프로시저)

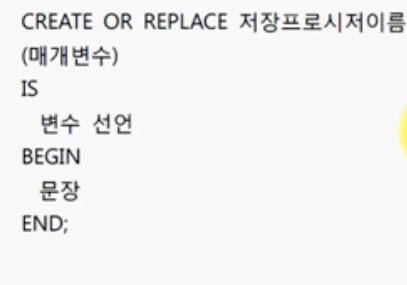
가) 특정 작업을 수핼할 수 있고, 이름이 있는 PL/SQL블록

나) 매개 변수를 입력받을 수 있음

다)DB에 저장되어 반복적으로 사용할 수 있음

라) 배치 작업 또는 구현이 복잡한 트랜잭션을 수행하는 용도로 사용함

2) 형식



3) 저장 프로시저 실습 예제

가) 급여 인상 저장 프로시저 실습

====================================

--create or replace procedure 프로시저이름(매개변수)

--입력매개변수: 변수명 in 자료형

--출력매개변수: 변수명 out 자료형

====================================

CREATE OR REPLACE PROCEDURE UPDATE\_SAL

(V\_EMPNO IN NUMBER ) --입력매개변수

is--선언부

BEGIN --실행부

update employees

set salary=salary\*1.1

where employee\_id=v\_empno;

commit;

END;

/

====================================

--저장 프로시저의 실행 방법

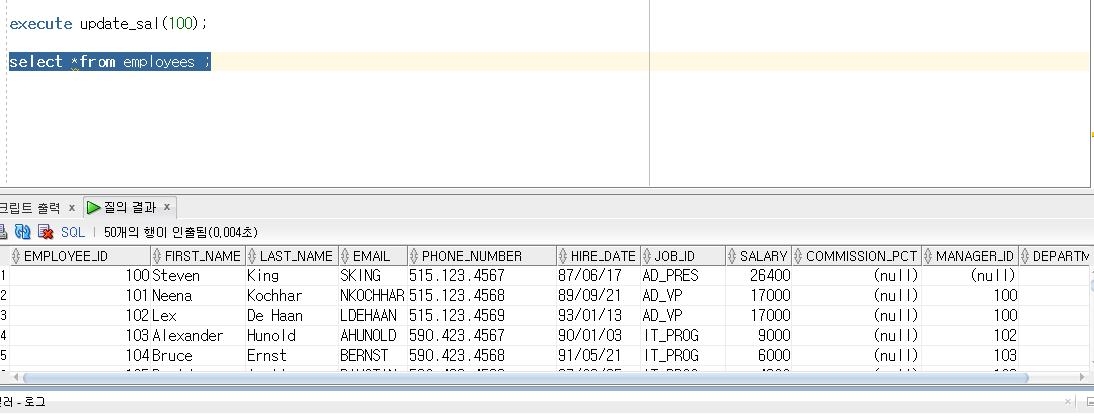
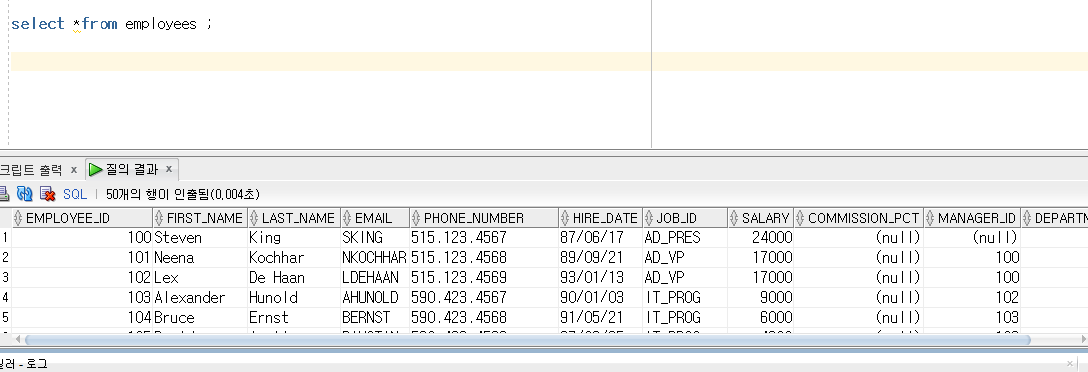
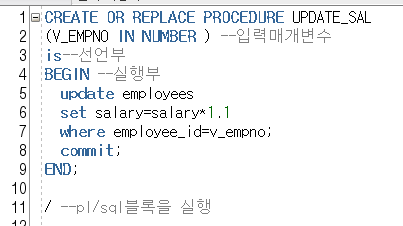
--execute 저장프로시저이름 (입력값)

execute update\_sal(100);

select \*from employees ;

===================================

--자바에서 프로시저 실행



나) 한 줄 메모장 저장 프로시저 실습

========

----한 줄 메모장 테이블

create table memo(

idx number primary key, --기본키

writer varchar2(50) not null,

memo varchar2(500) not nul,l

post\_date date default sysdate—기본값 현재시간

)

-----시퀀스 생성

create sequence memo\_seq

start with 1 --부터 시작

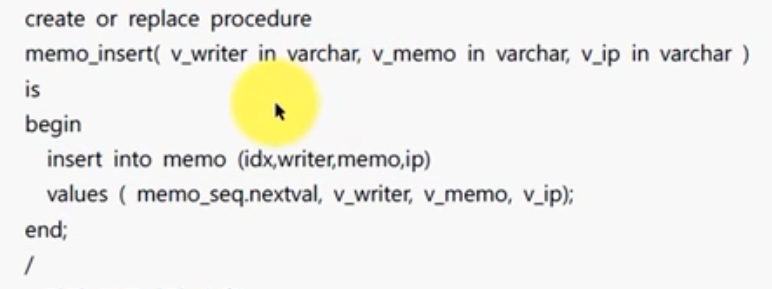
increment by 1 –씩 증가

nomaxvalue; --무제한 증가

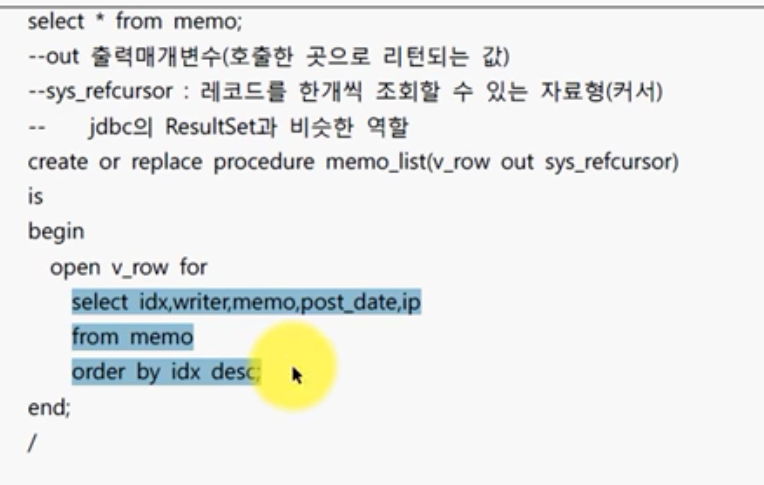
insert into memo

vaelues(memo\_seq.nextval,’kim’,’memo’)

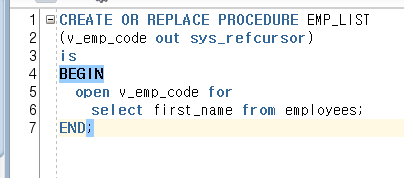
--저장 프로시저 작성



--저장 프로시저 호출



다) 커서를 이용해서 hr계정의 employees테이블에서 정보 뽑기 실습



CREATE OR REPLACE PROCEDURE EMP\_LIST

(v\_emp\_code out sys\_refcursor)

is

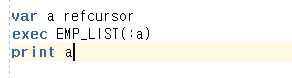
BEGIN

open v\_emp\_code for

select first\_name from employees;

END;

실행:



var a refcursor

exec EMP\_LIST(:a)

print a

==================================

마. 함수(Function)

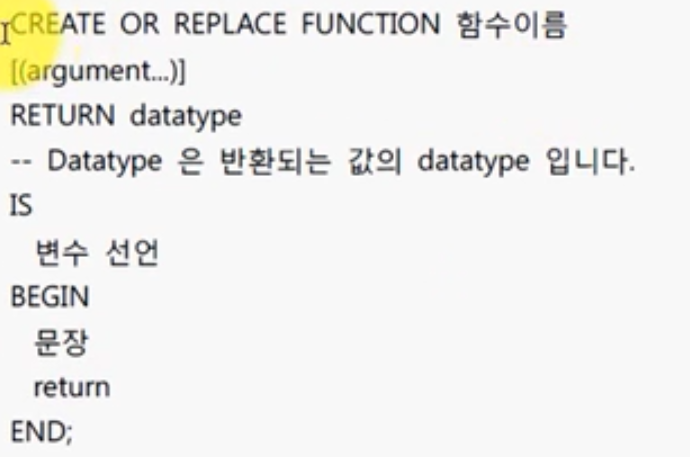
1) 값을 계산하고 결과값을 반환하기 위해서 사용

2) 저장 프로시저와의 차이점

입력 매개변수만 사용할 수 있고 리턴 타입을 반드시 지정해야 함

(return이 반드시 필요한데 한개 밖에 안됨)

3) 형식



create or replace functuon 함수이름

[(argument...)]

return datatype

--Datatype은 반환되는 값의 datatype 입니다

IS

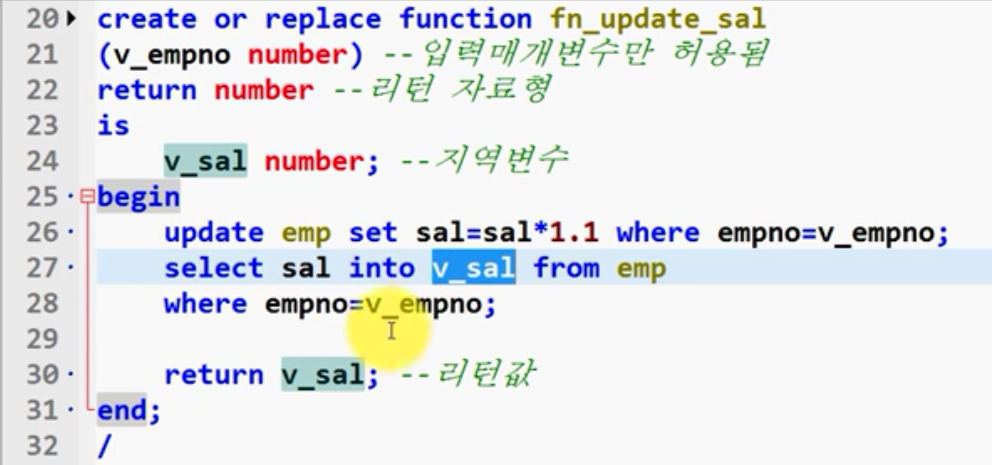
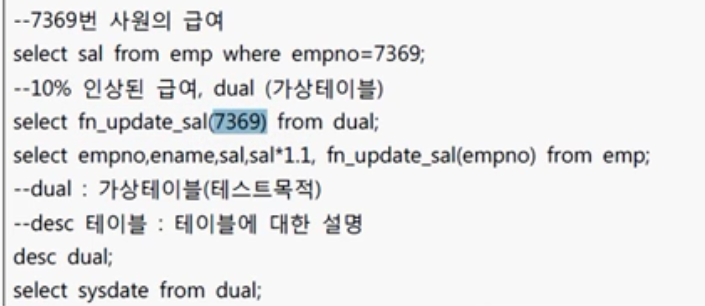
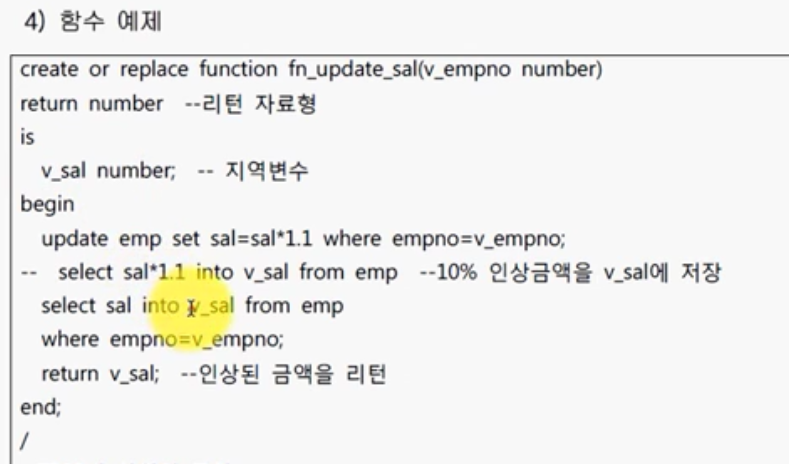
변수선언

Begin

문장

return

End;



create or replace function fn\_update\_sal(v\_empno number) --입력 매개변수만 허용됨

return number

is

v\_sal number;

begin

update employees set salary=salary\*1.1 where employee\_id=v\_empno;

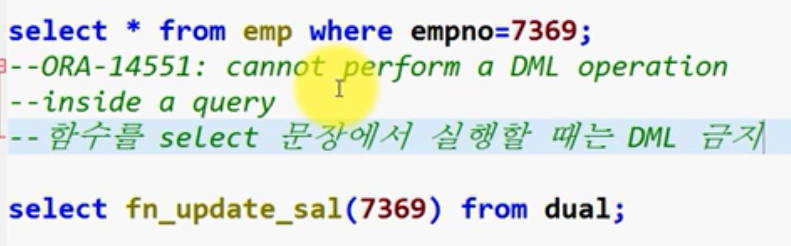
select salary into v\_sal from employees

where employee\_id=v\_empno;

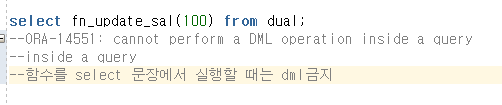
return v\_sal; --- 리턴값

end;

/



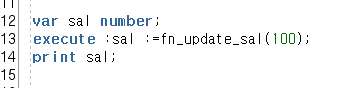
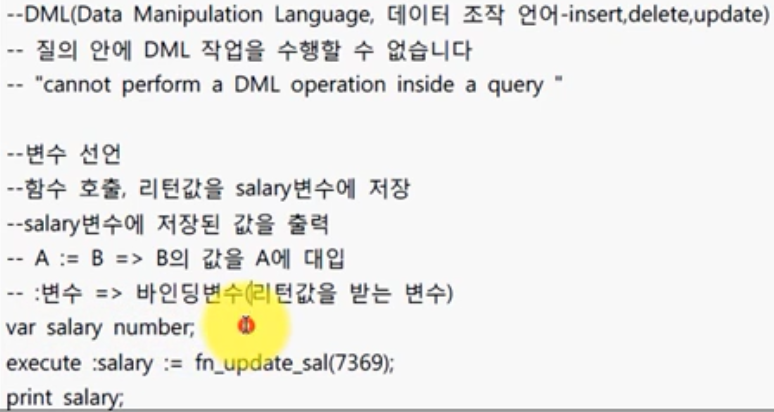
실행해보기:



select \*from employees;

select \*from employees where employee\_id=100;

select fn\_update\_sal(100) from dual;



var sal number;

execute :sal :=fn\_update\_sal(100);

print sal;

함수 안에서는 update delete 같은 문장이 사용가능 하지만 최대한 사용x 하는것이 좋다고 함

--> 저장 프로시저 사용

바. pl/sql 제어문

--> pl/sql에도 제어문이 있음 ex) if문 for문 while문 등등

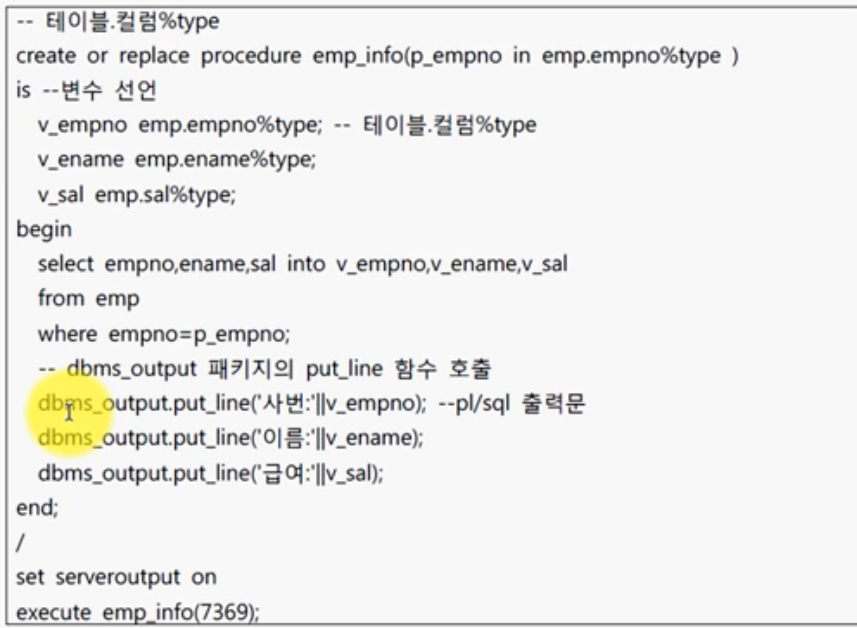
1) %type 테이터형

가) 테이블의 컬럼 데이터 타입을 모를 경우 사용

나) 테이블의 테이터 타입이 변경될 경우 다시 수정할 필요가 없음

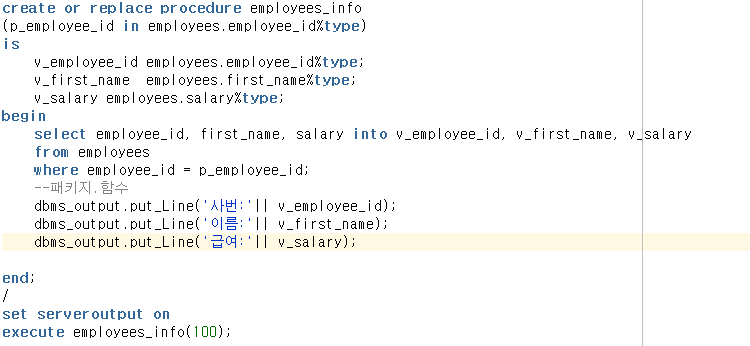
다) 예제

%type을 사용하게 되면 우리가 자료형을 구체적을 적지 않아도 그 테이블의 자료형으로 처리가 가능



자바에서 패키지란? 여러가지 클래스들을 묶은 것

오라클에서 패키지란? 여러가지 함수나 프로시저들을 묶은 것



create or replace procedure employees\_info

(p\_employee\_id in employees.employee\_id%type)

is

v\_employee\_id employees.employee\_id%type;

v\_first\_name employees.first\_name%type;

v\_salary employees.salary%type;

begin

select employee\_id, first\_name, salary into v\_employee\_id, v\_first\_name, v\_salary

from employees

where employee\_id = p\_employee\_id;

--패키지.함수

dbms\_output.put\_Line('사번:'|| v\_employee\_id);

dbms\_output.put\_Line('이름:'|| v\_first\_name);

dbms\_output.put\_Line('급여:'|| v\_salary);

end;

/

set serveroutput on

execute employees\_info(100);

2) if 문

가) 형식



나) if문 예제



pl/sql 에서 A=B 같다, A:=B 대입

3) for loop 문

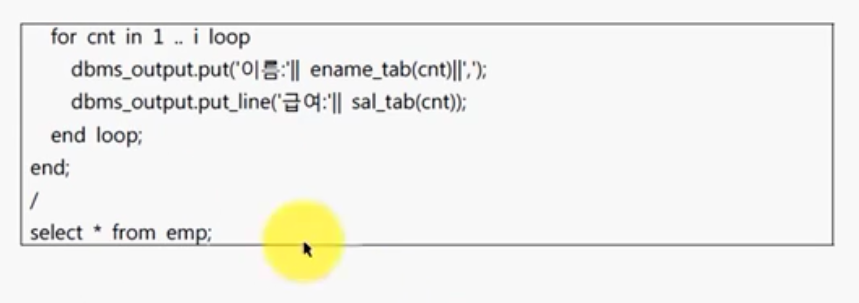
가) index: 자동 선언되는 binary\_integer 형 변수 1씩 증가

나) reverse 옵션이 사용되 경우 1씩 감소

다) in 다음에는 cursor나 select 문이 올수 있음



ex)



라) 예제

-- binary\_integer: pl/sql에서 사용하는 정수형 변수(java의 int)

-- type 사용자정의 자료형 is table of 테이블

-- index by 배열의 인덱스 지정

declare -- 선언부

-- 사용자 정의 자료형



--무병 블록

declare --선언부

--type 자료형 이름 is ...

type ename\_table

is table of employees.employee\_id%type index by binary\_integer;

type sal\_table

is table of employees.salary%type index by binary\_integer;

--변수명 자료형

ename\_tab ename\_table;

sal\_tab sal\_table;

i binary\_integer := 0;

begin

-- for 레코드 변수 in 집합

for emp\_row in (select employee\_id, salary from employees) loop

i := i+1;

ename\_tab(i) := emp\_row.employee\_id;

sal\_tab(i) := emp\_row.salary;

end loop;

--for 카운트 변수 in 시작... 끝

for cnt in 1 .. i loop

dbms\_output.put('이름:'|| ename\_tab(cnt)||',');

dbms\_output.put\_Line('급여:'|| sal\_tab(cnt));

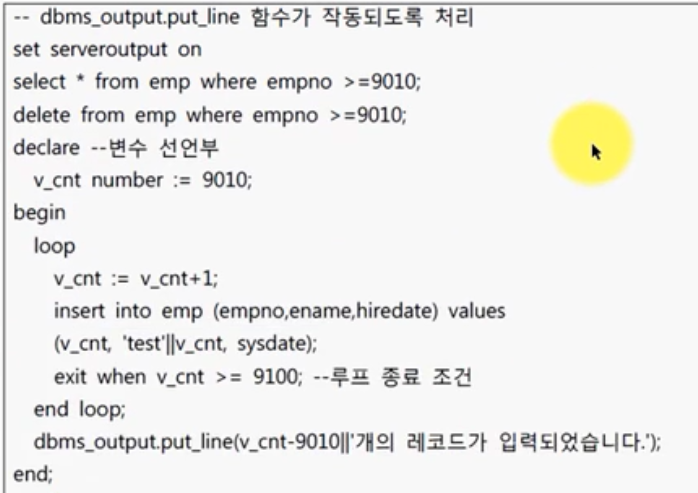
end loop;

end;

4) loop문

가)exit: 무조건 loop문 종료

나) exit when: loop문 빠져 나가는 조건을 제어



5) while loop문

가) for문과 비슷하며 조건이 true 일 경우만 반복되는 loop

나) 예제

