

요구사항번호	B-02	작성일	2023-04-05
요구사항명	기초정보 관리기능	작성자	조아현
DML			
1.과정명 관리 1.1 과정명 등록 create or replace procedure a_courseinsert (pcourse_seq in number, pcourse_name in varchar2) is vcourse_seq number; begin select nvl(max(course_seq),0) + 1 into vcourse_seq from course; insert into course values (vcourse_seq, pcourse_name);			

```
        dbms_output.put_line('등록 성공');  
end a_courseinsert;
```

--호출문

```
declare  
    vcourse_seq number;  
begin  
    a_courseinsert(vcourse_seq, '기초자바');  
end;
```

1.2 과정명 수정

```
create or replace procedure a_courseupdate (  
    pcourse_seq in number,  
    pcourse_name in varchar2
```

```
)  
  
is  
  
begin  
  
    update course  
  
    set  
  
        course_name = pcourse_name  
  
    where course_seq = pcourse_seq;  
  
exception  
  
    when others then  
  
        dbms_output.put_line('수정에 실패했습니다');  
  
end;  
  
  
--호출문  
  
begin  
  
    a_courseupdate(20,'테스트중');
```

```
end;
```

1.3 과정명 삭제

```
create or replace procedure a_coursedelete (
```

```
    pcourse_seq in number
```

```
)
```

```
is
```

```
begin
```

```
    delete from course where course_seq = pcourse_seq;
```

```
    dbms_output.put_line('삭제 성공');
```

```
exception
```

```
    when others then
```

```
dbms_output.put_line('삭제 실패');
```

```
end;
```

```
--과정명 삭제
```

```
begin
```

```
    a_coursedelete(20);
```

```
end;
```

```
1.4 과정 조회
```

```
select * from course;
```

--과목명 관리

--과목 조회

select * from subject;

-- 과목명 등록

drop procedure a_subjectinsert;

create or replace procedure a_subjectinsert(

 psubject_seq number,

 psubject_name varchar2

)

is

begin

 insert into subject values (psubject_seq,psubject_name);

```
    dbms_output.put_line('등록 성공');

end;

--호출문

begin

    a_subjectinsert(subject_seq.nextVal, '테스트');

end;


--과목명 수정

drop procedure a_subjectupdate;

create or replace procedure a_subjectupdate (
```

```
    psubject_seq number,  
  
    psubject_name varchar2  
)  
  
is  
  
begin  
  
    update subject set psubject_name = psubject_name where subject_seq = psubject_seq;  
  
    dbms_output.put_line('수정 성공');  
  
end;  
  
  
--호출문  
  
begin  
  
    a_subjectupdate(51,'수정내용');  
  
end;
```


--과목명 삭제

drop procedure a_subjectdelete;

create or replace procedure a_subjectdelete (

 psubject_seq number

)

is

begin

 delete from subject where subject_seq = psubject_seq;

 dbms_output.put_line('삭제 성공');

end;

--호출문

begin

 a_subjectdelete(51);

end;

3. 강의실명 관리

--강의실명 조회

select * from classroom;

--강의실명 등록

create or replace procedure a_classroom_in(

 pseq number,

```
    pcapacity number
)
is
begin
    insert into classroom values (pseq,pcapacity);

    dbms_output.put_line('등록 성공');

end;

--호출문

begin

    a_classroom_in(classroom_seq.nextVal,30);

end;
```

--강의실명 수정

create or replace procedure a_classroom_up (

 pseq number,

 pcapacity number

)

is

begin

 update classroom set capacity = pcapacity where classroom_seq = pseq;

 dbms_output.put_line('수정 성공');

end;

--호출문

begin

```
a_classroom_up(7,26);

end;

--강의실명 삭제

create or replace procedure a_classroom_de (

    pseq number

)

is

begin

    delete from classroom where classroom_seq = pseq;

    dbms_output.put_line('삭제 성공');

end;
```

--호출문

begin

 a_classroom_de(7);

end;

4.교재명 관리

-- 교재 조회

select * from book;

-- 교재명 등록

create or replace procedure a_bookinsert(

 pbook_seq number,

```
    pname varchar2,  
  
    pwriter varchar2,  
  
    ppublisher varchar2,  
  
    ppublishdate varchar2  
)  
  
is  
  
begin  
  
    insert into book values (pbook_seq,pname,pwriter,ppublisher,ppublishdate);  
  
    dbms_output.put_line('등록 성공');  
  
end;  
  
  
--호출문  
  
begin
```

```
a_bookinsert(book_seq.nextVal,'테스트','테스트','테스트',sysdate);
```

```
end;
```

```
--교재명 수정
```

```
drop procedure a_bookupdate;
```

```
create or replace procedure a_bookupdate (
```

```
    pbook_seq number,
```

```
    pname varchar2
```

```
)
```

```
is
```

```
begin
```

```
    update book set name = pname where book_seq = pbook_seq;
```



```
    dbms_output.put_line('수정 성공');

end;

--호출문

begin

    a_bookupdate(95,'수정내용');

end;

--교재명 삭제

drop procedure a_bookdelete;

create or replace procedure a_bookdelete (

    pbook_seq number

)
```

```
is  
  
begin  
  
    delete from book where book_seq = pbook_seq;  
  
    dbms_output.put_line('삭제 성공');  
  
end;  
  
  
--호출문  
  
begin  
  
    a_bookdelete(95);  
  
end;
```

요구사항번호	B-03	작성일	2023.04.05
--------	------	-----	------------

요구사항명	교사계정 관리기능	작성자	이민지
DML			
<pre>--1.등록 set serveroutput on; create or replace procedure procD_signupTeacher (--교사 등록 프로시저 pname in varchar2, -- 교수 이름 pssn in varchar2, -- 교수 주민번호 뒷자리 ptel in varchar2, -- 교수 전화번호 vnum out number) is begin select nvl(max(teacher_seq),0)+1 into vnum from teacher;</pre>			

--1. 교사 등록

insert into teacher values (vnum, pname, pssn, ptel);

end;

create or replace procedure procD_insertAbleSubject (--교사 강의 가능 과목 등록 프로시저

teacherNum in number,

subjectNum in number

)

is

begin

-- 교사 강의 가능 과목

insert into ableSubject values (teacherNum, subjectNum);

```
end;
```

■ 호출

```
declare
```

```
    vresult number;
```

```
begin
```

```
    procD_signupTeacher('이순신', '2105933', '010-4122-3860', vresult);
```

```
-- 2. 교사 가능 과목 등록
```

```
procD_insertAbleSubject(vresult, 1); -- 교사의 강의 가능 과목 선택적으로 추가, 공통적으로 가능한 과목 : 1,2,7,9,10,13,47
```

```
procD_insertAbleSubject(vresult, 2);
```

```
procD_insertAbleSubject(vresult, 7);
```

```
procD_insertAbleSubject(vresult, 9);
```

```
procD_insertAbleSubject(vresult, 10);
```

```
procD_insertAbleSubject(vresult, 13);  
  
procD_insertAbleSubject(vresult, 47);  
  
end;
```

--2. 조회

--2-1. 교사 정보 조회

- 교사 전체 명단
- 특정 교사를 선택한 경우 해당 교사의 강의 과목 리스트가 나오는 [VIEW vwTShowSubjectList](#)

create or replace view vwTShowSubjectList

as

select

t.teacher_seq as "교사번호",

t.name as "교사명",

s.subject_name as "배정된 과목",

```
os.startDate as "과목 시작 기간",

os.endDate as "과목 종료 기간",

c.course_name as "과정명",

oc.startDate as "과정 시작 기간",

oc.enddate as "과정 종료 기간",

b.name as "교재명",

oc.classroom_seq as "강의실",

case

    when oc.startDate > sysdate then '강의 예정'

    when sysdate between oc.startDate and oc.endDate then '강의중'

    when oc.endDate < sysdate then '강의 종료'

end as "강의 진행 여부"

from teacher t

inner join openSubject os on t.teacher_seq = os.teacher_seq
```

```
inner join subject s on s.subject_seq = os.subject_seq
```

```
inner join openCourse oc on oc.openCourse_seq = os.openCourse_seq
```

```
inner join course c on c.course_seq = oc.course_seq
```

```
inner join book b on b.book_seq = os.book_seq;
```

EX) VIEW 활용 : SELECT * FROM vwTShowSubjectList where "교사명" = '최성은'; --강의 목록을 보고 싶은 교사명을 입력

--특정 교사를 선택할 경우, 해당하는 강의목록만 출력되는 프로시저

```
create or replace procedure procD_ShowSubjectList (
```

tname in varchar2

)

is

```
cursor vcursor is select * from vwTShowSubjectList where "교사명" = tname;
```

begin

```
dbms_output.put_line('교사번호' || ' ' || '교사명' || ' ' || '배정된 과목' || ' ' || '과목 시작 기간' || ' ' || '과목 종료 기간' || ' ' || ' ');
```



```
과정명' || ' ' || '과정 시작 기간' || ' ' || '과정 종료 기간' || ' ' || '교재명' || ' ' || '강의실' || ' ' || '강의 진행 여부' );
```

```
for vrow in vcursor loop --vcursor를 결과 테이블=레코드집합으로 봄.
```

```
        dbms_output.put_line(vrow."교사번호" || ' ' || vrow."교사명" || ' ' || vrow."배정된 과목" || ' ' || vrow."과목 시작  
기간" || ' ' || vrow."과목 종료 기간" || ' ' || vrow."과정명" || ' ' || vrow."과정 시작 기간" || ' ' || vrow."과정 종료 기간"  
|| ' ' || vrow."교재명" || ' ' || vrow."강의실" || ' ' || vrow."강의 진행 여부" );
```

```
end loop;
```

```
end;
```

```
begin
```

```
-- 교사 전체 명단에서 보고싶은 교사명 입력
```

```
procD_ShowSubjectList('최성은');
```

```
end;
```

```
--3. 수정
```

```
create or replace procedure procD_signupTeacher (
```

```
tname in varchar2,  
  
tssn in number,  
  
ttel in varchar2)  
  
is  
  
vcount number;  
  
vseq number;  
  
begin  
  
  
  
--해당 교사가 있는지 확인  
  
select count(*) into vcount from teacher where name = tname;  
  
  
  
if vcount = 0 then  
  
    dbms_output.put_line('해당하는 교사가 없습니다.');  
    return;
```

```
else

    select teacher_seq into vseq from teacher where name = tname;

end if;


-- 해당 교사 업데이트

update teacher set name = tname, ssn = tssn, tel = ttel where teacher_seq = vseq;


exception

    when others then

        dbms_output.put_line('추가에 실패했습니다.');
```

end;

-- 실제 수정작업

begin

```
procD_signupTeacher('이순신', '2313232', '010-4343-2342');
```

```
end;
```

```
--4. 삭제
```

```
create or replace procedure procD_deleteTeacher (
```

```
    tname in varchar2
```

```
)
```

```
is
```

```
    tseq number;
```

```
begin
```

```
--1. 교사 이름이 테이블에 있는지 확인
```

```
select teacher_seq into tseq from teacher where name = tname;
```

--2. 교사 가능 과목에서 교사번호에 해당하는 레코드 삭제

```
delete from ableSubject where teacher_seq = tseq;
```

--3. 교사 테이블에서 해당 교사 삭제

```
delete from teacher where teacher_seq = tseq;
```

```
dbms_output.put_line('교사 ' || tname || '이/가 성공적으로 삭제되었습니다.');
```

exception

-- 4. 교사 이름이 존재하지 않을 경우 에러를 출력

```
when NO_DATA_FOUND then
```

```
    dbms_output.put_line('교사 ' || tname || '을 찾을 수 없습니다.');
```

```
end;
```

```
begin

    -- 삭제할 교사명 입력

    procD_deleteTeacher('이순신');

end;

--교사 삭제(퇴사) 트리거 > 교사 수업가능과목에서 해당교사의 내역 모두 삭제

create or replace trigger trgFiredTeacher

    before

    delete

    on teacher

    for each row

begin

    if deleting then

        delete from ablesubject where teacher_seq = :old.teacher_seq;
```

```
        dbms_output.put_line('퇴사처리 성공했습니다.');
```

```
    end if;
```

```
exception
```

```
    when others then
```

```
        dbms_output.put_line('삭제 실패');
```

```
end trgFiredTeacher;
```

요구사항번호	B-04	작성일	2023-04-05
요구사항명	개설과정 관리 기능	작성자	조아현
DML			

1. 개설 과정 등록

```
create or replace procedure a_opencourse_in (
```

```
    p_ocseq number,
```

```
    p_cseq number,
```

```
    p_clseq number,
```

```
    p_dseq number,
```

```
    p_sdate date,
```

```
    p_edate date,
```

```
    p_period varchar2
```

```
)
```

```
is
```

```
begin
```

```
    insert into opencourse values (p_ocseq,p_cseq,p_clseq,p_dseq,p_sdate,p_edate,p_period);
```

```
    dbms_output.put_line('등록 성공');
```

```
end;
```


--호출문

begin

 a_opencourse_in(opencourse_seq.nextVal,4,1,3,'22/01/01','22/03/02','6');

end;

2. 개설 과정 수정

create or replace procedure a_opencourse_up (

 p_ocseq number,

 p_cseq number,

 p_clseq number,

 p_dseq number,

 p_sdate date,

 p_edate date,

 p_period varchar2

```
)  
  
is  
  
begin  
    update opencourse set opencourse_seq = p_ocseq,  
                           course_seq = p_cseq,  
                           classroom_seq = p_clseq,  
                           director_seq = p_dseq,  
                           startdate = p_sdate,  
                           enddate = p_edate,  
                           period = p_period  
    where opencourse_seq = p_ocseq;  
  
    dbms_output.put_line('수정 성공');  
  
end;  
  
--호출문
```

```
begin
```

```
    a_opencourse_up(14,4,1,3,'22/03/01','22/06/02','6');
```

```
end;
```

3. 개설 과정 삭제

```
create or replace procedure a_opencourse_de (
```

```
    p_ocseq number
```

```
)
```

```
is
```

```
begin
```

```
    delete from opencourse where opencourse_seq = p_ocseq;
```

```
    dbms_output.put_line('삭제 성공');
```

```
end;
```

```
--호출문
```

```
begin
```

```
    a_opencourse_de(14);
```

```
end;
```

4. 개설 과정 정보 출력

```
create or replace procedure a_opencourse_r(
```

```
    pcursor out sys_refcursor
```

```
)
```

```
is
```

```
begin
```

```
    open pcursor
```

```
    for
```

```
        select * from vwAOpencourseread;
```

```
end;
```

```
--호출문
```

```
declare
```

```
vcursor sys_refcursor;
```

```
vrow vwAOpenCourseRead%rowtype;
```

```
begin
```

```
a_opencourse_r(vcursor);
```

```
loop
```

```
fetch vcursor into vrow;
```

```
exit when vcursor%notfound;
```

```
dbms_output.put_line(vrow.개설과정번호||'   ||vrow.과정명||'   ||vrow.과정시작날짜||'   ||vrow.강의실||'   ||vrow.개설과목등록여부||'   ||vrow."
교육생 등록 인원");
```

```
end loop;
```

```
end;
```

```
-- 특정 개설 과정 선택시 정보 출력
```

```
create or replace procedure a_ocstudent_r(
```

```
    pseq number,
```

```
    pcursor out sys_refcursor
```

```
)
```

```
is
```

```
begin
```

```
    open pcursor
```

```
for

    select * from vwACourseStudentInfo where 과정번호 = pseq;

end;

--호출문

declare

    vcursor sys_refcursor;

    vrow vwACourseStudentInfo%rowtype;

begin

    a_ocstudent_r('1',vcursor);

    loop

        fetch vcursor into vrow;

        exit when vcursor%notfound;
```

```
        dbms_output.put_line(vrow.과정번호||'   '||vrow.개설과목명||'   '||vrow.과목시작날짜||'   '||vrow.과목종료날짜||'   '||vrow.교재명||'   '||vrow.교사  
명|| vrow."주민번호 뒷자리" || vrow.전화번호 || vrow.등록일 || vrow.수료상태);  
  
    end loop;  
  
end;
```

5. 교육생 수료상태 지정

5.1 종료과정 내 교육생 수료상태 정보 확인

```
create or replace procedure a_studentstate_r(  
  
    pcursor out sys_refcursor  
  
)  
  
is  
  
begin
```



```
open pcursor

for

select * from vwACourseStudentState

      where 과정종료날짜 < sysdate and 수료상태 <> '중도탈락';

end;


--호출문

declare

vcursor sys_refcursor;

vrow vwACourseStudentState%rowtype;

begin


a_studentstate_r(vcursor);
```

```

loop

    fetch vcursor into vrow;

    exit when vcursor%notfound;

    dbms_output.put_line(vrow.개설과정번호||'   '||vrow.과정명||'   '||vrow.과정시작날짜||'   '||vrow.과정종료날짜||'   '||vrow.학생번호||'   '||vrow.학
생명||' || vrow."수료상태" );

end loop;

end;

```

5.2 수료상태 수정

```

create or replace procedure a_studentstate_up(

    p_oc_seq number

)

```

```

is

begin

    update signupcourse set completion = '수료' where opencourse_seq = p_oc_seq and completion <> '중도탈락';

end;

--호출문

begin

    a_studentstate_up(13);

end;

```

요구사항번호	B-05	작성일	2023-04-05
요구사항명	개설 과목 관리 기능	작성자	조아현
DML			

--개설과목 신규 등록

create or replace procedure a_opensubject_in(

 posseq number,

 pocseq number,

 psdate date,

 pedate date,

 pbnun number,

 psnum number,

 ptnum number

)

is

begin

 insert into opensubject

 values(posseq,pocseq,psdate,pedate,pbnun,psnum,ptnum);

end;

--호출문

begin

```
a_opensubject_in(opensubject_seq.nextVal,1,'22/01/01','22/01/01',2,2,2);
```

```
end;
```

```
--개설과목 정보 출력
```

```
create or replace procedure a_opensubject_r(
```

```
    pcursor out sys_refcursor
```

```
)
```

```
is
```

```
begin
```

```
    open pcursor
```

```
    for
```

```
    select * from vwAOpenSubject;
```

```
end;
```

```
--호출문
```

```
declare
```

```

vcursor sys_refcursor;

vrow vwACoursePlan%rowtype;

begin

    a_opensubject_r(3,vcursor);

    loop

        fetch vcursor into vrow;

        exit when vcursor%notfound;

        dbms_output.put_line(vrow.개설과정번호||'   '||vrow.과정명||'   '||vrow.계획번호||'   '||vrow.월||'   '||vrow.계획내용);

    end loop;

end;

```

--개설과목 수정

```

create or replace procedure a_opensubject_up(

    posseq number,

```

```
pocseq number,  
psdate date,  
pedate date,  
pbnun number,  
psnum number,  
ptnum number  
)  
is  
begin  
    update opensubject set opensubject_seq = posseq, opencourse_seq = pocseq,  
        startdate = psdate, enddate = pedate, book_seq = pbnun,  
        subject_seq = psnum, teacher_seq = ptnun  
    where opensubject_seq = posseq;  
end;  
  
--호출문  
begin
```

```
a_opensubject_up(1,2,'22/01/02','22/01/03',1,2,3);  
end;  
  
--개설과목 삭제  
create or replace procedure a_opensubject_de(  
    posseq number  
)  
is  
begin  
    delete from opensubject where opensubject_seq = posseq;  
end;  
  
--호출문  
begin  
    a_opensubject_de(1);  
end;
```


요구사항번호	B-06	작성일	2023-04-05
요구사항명	교육생 관리 기능	작성자	조아현
DML			
<pre>--교육생 등록 drop procedure a_student_in; create or replace procedure a_student_in(pname varchar2, pssn varchar2, ptel varchar2, pstudent_date varchar2) is</pre>			

```
begin

    insert into student values(student_seq.nextVal,pname,pssn,ptel,pstudent_date);
```

```
end;
```

```
--호출문
```

```
begin

    a_student_in('테스트','854545-4545454','010-8792-3080','23/08/20');
```

```
end;
```

```
--전체 교육생 정보 출력
```

```
create or replace procedure a_student_r(
```

```
    pcursor out sys_refcursor
```

```
)  
  
is  
  
begin  
  
    open pcursor  
  
    for  
  
        select * from VWAALLSTUDENTREAD;  
  
end;  
  
--호출문  
  
declare  
  
    vcursor sys_refcursor;  
  
    vrow vwAAllStudentRead%rowtype;  
  
begin
```

```

a_student_r(vcursor);

loop

    fetch vcursor into vrow;

    exit when vcursor%notfound;

    dbms_output.put_line(vrow.교육생명||'   '||vrow."주민번호 뒷자리"||'   '||vrow.전화번호||'   '||vrow.등록일||'   ');

end loop;

end;

--특정 교육생 정보 출력

create or replace procedure a_specificstudent_r(

    pname varchar2,

```

```
pcursor out sys_refcursor
)
is
begin
    open pcursor
    for
        select * from vwAStudentState where 교육생명 = pname;
    end;

--호출문

declare

    vcursor sys_refcursor;

    vrow vwAStudentState%rowtype;
```

```
begin
```

```
    a_specificstudent_r('배 린서',vcursor);
```

```
    loop
```

```
        fetch vcursor into vrow;
```

```
        exit when vcursor%notfound;
```

```
        dbms_output.put_line(vrow.교육생명||'   ||vrow.과정명||'   ||vrow.과정시작날짜||'   ||vrow.과정종료날짜||'   ||vrow.강의실||'   ||vrow.수료여부||'  
'||vrow."수료 및 중도탈락 날짜"||'   ');
```

```
    end loop;
```

```
end;
```

```
--교육생 수정
```

```
create or replace procedure a_student_up(
```

```
    pseq number,
```

```
    pname varchar2,

    pssn varchar2,

    ptel varchar2,

    pstudent_date varchar2

)

is

begin

    update student set student_seq=pseq,name = pname, ssn = pssn, tel = ptel,student_date = pstudent_date where student_seq = pseq;

end;


--호출문

begin

    a_student_up(367,'테스트2','854545-4545454','010-8792-3080','23/08/20');
```

```
end;
```

```
-- 교육생 삭제
```

```
create or replace procedure a_student_de(
```

```
    pname varchar2
```

```
)
```

```
is
```

```
begin
```

```
    delete from student where name = pname;
```

```
end;
```

```
--호출문
```



```

begin

    a_student_de('배린서');

end;

```

요구사항번호	B-07	작성일	2023.04.06
요구사항명	시험 관리 및 성적 조회	작성자	김경민
DML			
1. 시험 관리 --1.1 -- 특정 개설 과목 정보(과정 번호, 과목명, 과목기간, 교재명, 교사명, 교육생명, 주민번호 뒤자리, 전화번호, 등록일, 수료 및 중도탈락) 을 출력 set serveroutput on; create or replace procedure procAOpenSubject(vopencourse_seq in NUMBER) is cursor vcursor is (select			

```
oc.opencourse_seq as "과정번호",
os.opensubject_seq as "과목번호",
c.course_name as "과정명",
(oc.startdate || '~' || oc.enddate)as "과정기간",
oc.classroom_seq as "강의실", s.subject_name as "과목명",
(os.startdate || '~' || os.enddate)as "과목기간",
  b.name as "교재명",
t.name as "교사명",
case
  when score.wscore is not null
    and score.pscore is not null
    and score.ascore is not null then 'o'
else 'x'
end as "성적 등록여부",
case
when sysdate>os.enddate and tq.content is not null then 'o'
when sysdate>os.startdate and sysdate<os.enddate then '수업중'
when sysdate<os.startdate then '수업개시안함'
else 'x'
end as "시험문제등록"
from opensubject os
  inner join subject s
    on os.subject_seq = s.subject_seq
    inner join opencourse oc
      on oc.opencourse_seq = os.opencourse_seq
```

```

        inner join course c
        on oc.course_seq = c.course_seq
        inner join book b
        on os.book_seq = b.book_seq
        inner join teacher t
        on os.teacher_seq = t.teacher_seq
        left outer join score
        on os.opensubject_seq = score.opensubject_seq
        left join test
        on os.opensubject_seq = test.opensubject_seq
        inner join testquiz tq
        on test.test_seq = tq.test_seq
        where oc.opencourse_seq = vopencourse_seq);

begin
for vrow in vcursor
loop
dbms_output.put_line(vrow."과목번호"||' '||vrow."과정명"||' '||vrow."과정기간"||' '||vrow."강의실"||' '||vrow."과목명"||' '||vrow."과목기간"||' '||vrow."교재명"||' '||vrow."교사명" );

end loop;
end procAOpenSubject;
--예시-----
begin
    procAOpenSubject(3);

```

```
end;
```

```
drop procedure procAOpenSubject;
```

```
-----
```

```
--1.2
```

--과정 번호로 선택하면 해당 과정의 개설과목정보(과목번호, 과정명, 과정기간, 강의실, 과목명, 과목기간, 교재명,교사명)을 출력하고 해당 과목별 성적 등록여부와 시험문제 파일 등록 여부를 출력한다.

```
create or replace procedure procAOpenSubject(
```

```
    vopencourse_seq in NUMBER
```

```
)
```

```
is
```

```
cursor vcursor is
```

```
(select
```

```
oc.opencourse_seq as "과정번호",
```

```
os.opensubject_seq as "과목번호",
```

```
c.course_name as "과정명",
```

```
(oc.startdate || '~' || oc.enddate)as "과정기간",
```

```
oc.classroom_seq as "강의실", s.subject_name as"과목명",
```

```
(os.startdate || '~' || os.enddate)as "과목기간",
```

```
    b.name as "교재명",
```

```
t.name as "교사명",
```

```
case
```

```
    when score.wscore is not null
```

```
        and score.pscore is not null
```



```

on test.test_seq = tq.test_seq
    where oc.opencourse_seq = vopencourse_seq);

begin
for vrow in vcursor
loop
dbms_output.put_line(vrow."과목번호"||' '||vrow."과정명"||' '||vrow."과정기간"||' '||vrow."강의실"||' '||vrow."과목명"||' '||vrow."과목기간"||' '||vrow."교재명"||' '||vrow."교사명" );

end loop;
end procAOpenSubject;
--예시-----
begin
    procAOpenSubject(3);
end;
drop procedure procAOpenSubject;
-----

```

요구사항번호	B-08	작성일	2023.04.06
요구사항명	출결 관리 및 출결 조회	작성자	김경민

DML

--2.1

--관리자가 특정 과목번호로 개설과목을 선택할 경우 해당 선택과목을 수강한 교육생 이름, 주민번호 뒷자리, 필기 성적, 실기 성적을 출력한다

```
create or replace procedure procASubjectScore(
    vopensubject_seq in NUMBER
)
is
cursor vcursor is
select
os.opensubject_seq as "과목번호",
s.name "교육생이름",
substr(s.ssn,8) as "주민번호뒷자리",
score.wscore"필기성적",
score.pscore "실기성적"
from score
    inner join opensubject os
        on score.opensubject_seq = os.opensubject_seq
        inner join student s
            on score.student_seq = s.student_seq
        where os.opensubject_seq=vopensubject_seq
        order by os.opensubject_seq;

begin
for vrow in vcursor
loop
```

```

dbms_output.put_line(vrow."교육생이름"||' '||vrow."주민번호뒷자리"||' '||vrow."필기성적"||' '||vrow."실기성적");
end loop;
end procASubjectScore;

```

--예시-----

```

begin
procASubjectScore(1);
end;

```

--2.2

--관리자가 특정 교육생을 선택할 경우 해당 교육생이 수강한 모든 개설과목명, 개설과목기간, 교사명, 필기성적, 실기성적을 출력한다

```

create or replace procedure proctASStudentScore(
    vstudent_seq in NUMBER
)
is
cursor vcursor is
select s.name as "교육생명",
subject.subject_name as "개설과목명",
os.startdate || ' ~ ' || os.enddate as "개설과목기간",
t.name as "교사명",
score.wscore as "필기성적",
score.pscore as "실기성적"

```



```

from student s
  inner join score
    on s.student_seq = score.student_seq
    inner join opensubject os
      on score.opensubject_seq = os.opensubject_seq
      inner join subject
        on os.subject_seq = subject.subject_seq
        inner join teacher t
          on os.teacher_seq = t.teacher_seq
          where s.student_seq =vstudent_seq;

begin
  for vrow in vcursor
  loop

    dbms_output.put_line(vrow."개설과목명"||' '||vrow."개설과목기간"||' '||vrow."교사명"||' '||vrow."필기성적"||' '||vrow."실기성적");

  end loop;
end proctASStudentScore;
--예시|-----
begin
  proctASStudentScore(1);
end;
-----

```

--2.3

--관리자가 특정 근태 상황을 입력하면 그 정보에 해당하는 모든 교육생 출결 정보 조회를 출력한다.

```
create or replace procedure procAstatus(  
    status in varchar2
```

```
)
```

```
is
```

```
cursor vcursor is
```

```
select ATTENDANCE_SEQ as "seq",
```

```
STUDENT_SEQ as "학생번호" ,
```

```
ATTENDANCE_DATE as "DAY",
```

```
STARTTIME as "출근",
```

```
ENDTIME as "퇴근",
```

```
ETC as "근태현황" from attendance where etc = status;
```

```
cursor vcursor2 is
```

```
select ATTENDANCE_SEQ as "seq",
```

```
STUDENT_SEQ as "학생번호" ,
```

```
ATTENDANCE_DATE as "DAY",
```

```
STARTTIME as "출근",
```

```
ENDTIME as "퇴근",
```

```
"근태현황" from vwAAttendStudent;
```

```

cursor vcursor3 is
select ATTENDANCE_SEQ as "seq",
STUDENT_SEQ as "학생번호" ,
ATTENDANCE_DATE as "DAY",
STARTTIME as "출근",
ENDTIME as "퇴근",
"근태현황" from vwAlateStudent;
begin

case
when status in ('병가','외출','조퇴') then
for vrow in vcursor
loop
dbms_output.put_line(vrow."seq"||'   '||vrow."학생번호"||'   '||vrow."DAY"||'   '||
vrow."출근"||'   '||vrow."퇴근"||'   '||vrow."근태현황");
end loop;
when status = '정상' then
for vrow in vcursor2
loop
dbms_output.put_line(vrow."seq"||'   '||vrow."학생번호"||'   '||vrow."DAY"||'   '||
vrow."출근"||'   '||vrow."퇴근"||'   '||'정상');
end loop;

when status = '지각' then
for vrow in vcursor3

```

```

loop
dbms_output.put_line(vrow."seq"||'   '||vrow."학생번호"||'   '||vrow."DAY"||'   '||
vrow."출근"||'   '||vrow."퇴근"||'   '||'지각');
end loop;
end case;
end procAstatus;

begin
procAstatus('정상');
end;

```

요구사항번호	B-09	작성일	2023-04-05
요구사항명	수업계획서 관리	작성자	조아현
DML			

--모든 수업계획서 조회

```
select * from courseplan;
```

-- 수업계획서 등록

```
create or replace procedure a_plan_in(
```

```
    pseq number,
```

```
    pmonth number,
```

```
    pcontent varchar2,
```

```
    pnum number
```

```
)
```

```
is
```

```
begin
```

```
    insert into courseplan values(pseq,pmonth,pcontent,pnum);
```

```
end;
```

```
--호출문
```

```
begin
```

```
    a_plan_in(courseplan_seq.nextVal,1,'테스트',1);
```

```
end;
```

```
--개설과정 수업계획서 조회
```

```
--개설과정 조회 및 개설과정 번호 확인
```

```
select * from vwAOpenCourse;
```

```
--특정 개설과정 수업계획서 조회
```

```
create or replace procedure a_specificplan_r(  
    pseq number,  
    pcursor out sys_refcursor  
)  
is  
begin  
    open pcursor  
    for  
        select * from vwACoursePlan e where 개설과정번호 = pseq;  
end;  
  
--호출문  
  
declare
```

```
vcursor sys_refcursor;

vrow vwACoursePlan%rowtype;

begin

a_specificplan_r(3,vcursor);

loop

    fetch vcursor into vrow;

    exit when vcursor%notfound;

    dbms_output.put_line(vrow.개설과정번호||'   '||vrow.과정명||'   '||vrow.계획번호||'   '||vrow.월||'   '||vrow.계획내용);

end loop;

end;

--수업계획서 수정

-- 개설과정의 수업계획서 조회 및 계획번호 확인
```



```
create or replace procedure a_specificplan_r(  
    pseq number,  
    pcursor out sys_refcursor  
)  
is  
begin  
    open pcursor  
    for  
        select * from vwACoursePlan e where 개설과정번호 = pseq;  
end;  
  
--호출문  
begin
```

```
a_specificplan_r(2);
```

```
end;
```

```
--수업계획서 수정
```

```
create or replace procedure a_plan_up(
```

```
    pcseq number,
```

```
    pmonth number,
```

```
    pcontent varchar2,
```

```
    poseq number
```

```
)
```

```
is
```

```
begin
```

```
    update courseplan set courseplan_seq = pcseq, month = pmonth,
```

```
content = pcontent, opencourse_seq = poseq
```

```
where courseplan_seq = pcseq;
```

```
end;
```

```
--호출문
```

```
begin
```

```
    a_plan_up(84,2,'테스트 1 2 이라 1',5);
```

```
end;
```

4. 수업계획서 삭제

```
create or replace procedure a_plan_de(
```

```
    pcseq number
```

```
)
```

```

is

begin

    delete from courseplan where courseplan_seq = pcseq;

end;

--호출문

begin

    a_plan_de(84);

end;

```

요구사항번호	B-10	작성일	2023-04-05
요구사항명	사후관리(취업분야)	작성자	최재현
PROCEDURE			

```
-- 관리자 > 사후관리(취업현황) 등록
CREATE OR REPLACE PROCEDURE PROCAINSERTJOB(
    pstudent_seq number,
    pstate varchar2,
    pfield varchar2
)
is
begin
    if pfield not in ('풀스택', '프론트엔드', '백엔드') then
        raise_application_error(-20887, '취업 분야를 다시 작성해주세요, 유효값: 풀스택, 프론트엔드, 백엔드');
    end if;

    if pstate = '미취업' then
        INSERT INTO JOB (student_seq, state) VALUES(pstudent_seq, pstate);
    else
        INSERT INTO JOB VALUES(pstudent_seq, pstate, pfield);
        INSERT INTO JOBSTUDENT VALUES(pstudent_seq);
    end if;
end;
/
begin
```

```

    PROCAINSERTJOB(4, '미취업', null);
--    PROCAINSERTJOB(200, '취업', '백드'); -- 취업분야가 잘못됐음
--    PROCAINSERTJOB(200, '취업', '백엔드'); --정상삽입
end;
/

SELECT * FROM job WHERE student_seq = 200;

--관리자 > 사후관리(취업현황) 수정
CREATE OR REPLACE PROCEDURE PROCAUPDATEJOB(
    pfield varchar2,
    pstudent_seq number
)
is
    cursor vcursor is SELECT student_seq FROM job;
    vresult number;
begin
    for vrow in vcursor loop
        if vrow.student_seq = pstudent_seq then
            vresult := 1;

```

```

        end if;
    end loop;

    if vresult = 1 then  --1 성공
        if pfield not in ('풀스택', '프론트엔드', '백엔드') then
            raise_application_error(-20887, '취업 분야를 다시 작성해주세요, 유효값: 풀스택, 프론트엔드, 백엔드');
        else
            UPDATE JOB SET field = pfield, state = '취업' where student_seq = pstudent_seq;
            dbms_output.put_line('취업분야: ' || pfield || ' | ' || '학생번호: ' || pstudent_seq );
        end if;

    else

        if pfield not in ('풀스택', '프론트엔드', '백엔드') then
            raise_application_error(-20888, '존재하지 않는 학생번호입니다.   | 학생번호와, 취업 분야를 다시 작성해주세요, 유효값: 풀스택,
프론트엔드, 백엔드');
        else
            raise_application_error(-20777, '존재하지 않는 학생번호입니다.');
```

```
end;
/

begin
    PROCAUPDATEJOB('AAA', 300);
end;
/
--관리자 > 사후관리(취업현황) 삭제
CREATE OR REPLACE PROCEDURE PROCADELETEJOB(
    pstudent_seq number
)
is
    cursor vcursor is SELECT student_seq FROM job;
    vresult number;

begin
    for vrow in vcursor loop
        if vrow.student_seq = pstudent_seq then
            vresult := 1;
        end if;
    end loop;
end;
```



```
end loop;

if vresult = 1 then --학생이 있음
    DELETE FROM job where student_seq = pstudent_seq;
    DELETE FROM jobstudent where student_seq = pstudent_seq;
else
    raise_application_error(-20111, '존재하지 않는 값입니다.');
```

end if;

end;

/

--begin

-- PROCDELETEJOB(300);

--end;

/

요구사항번호	C-01	작성일	2023-04-05
--------	------	-----	------------

요구사항명	강의스케줄 조회	작성자	김경민
DML			
<pre>--강의예정 스케줄 조회 create or replace procedure procTTeacherSchedulePlan(vteacher_seq number) is cursor vcursor is select os.opensubject_seq as "개설과목번호", c.course_name as "과정명", oc.startdate as "과정시작날짜", oc.enddate as "과정종료날짜", cl.classroom_seq ' 강의실' as "강의실",</pre>			

```
s.subject_name as "과목명",

os.startdate as "과목시작날짜",

os.enddate as "과목종료날짜",

b.name as "교재명",

(select

    count(sc.student_seq)

from signupcourse sc

    where sc.opencourse_seq = oc.opencourse_seq) as "교육생 등록 인원"

from subject s

    inner join opensubject os on os.subject_seq = s.subject_seq

    inner join opencourse oc on oc.opencourse_seq = os.opencourse_seq

    inner join course c on c.course_seq = oc.course_seq

    inner join classroom cl on cl.classroom_seq = oc.classroom_seq
```

```
inner join book b on b.book_seq = os.book_seq

        where os.teacher_seq = vteacher_seq and os.startdate > sysdate;

begin

for vrow in vcursor

loop

dbms_output.put_line(vrow."개설과목번호"||'   '||vrow."과정명"||'   '||vrow."과정시작날짜"||'   '||vrow."과정종료날짜"||'   '||vrow."강의실"||'   '||vrow."과목
명"||'   '||vrow."과목시작날짜"||'   '||vrow."과목종료날짜"||'   '||vrow."교재명"||'   '||vrow."교육생 등록 인원");

end loop;

end procTTeacherSchedule;


begin

procTTeacherSchedule(1);

end;
```

--강의중 스케줄 조회

create or replace procedure procTTeacherScheduleInProgress(

 vteacher_seq varchar2

)

is

cursor vcursor is

(select

 os.opensubject_seq as "개설과목번호",

 c.course_name as "과정명",

 oc.startdate as "과정시작날짜",

```
oc.enddate as "과정종료날짜",

cl.classroom_seq || '강의실' as "강의실",

s.subject_name as "과목명",

os.startdate as "과목시작날짜",

os.enddate as "과목종료날짜",

b.name as "교재명",

(select

    count(sc.student_seq)

from signupcourse sc

    where sc.opencourse_seq = oc.opencourse_seq) as "교육생 등록 인원"

from subject s

    inner join opensubject os on os.subject_seq = s.subject_seq

    inner join opencourse oc on oc.opencourse_seq = os.opencourse_seq
```

```

inner join course c on c.course_seq = oc.course_seq

inner join classroom cl on cl.classroom_seq = oc.classroom_seq

inner join book b on b.book_seq = os.book_seq

        where os.teacher_seq = vteacher_seq and os.startdate < sysdate and os.enddate > sysdate);

begin

for vrow in vcursor

loop

dbms_output.put_line(vrow."개설과목번호"||'   '||vrow."과정명"||'   '||vrow."과정시작날짜"||'   '||vrow."과정종료날짜"||'   '||vrow."강의실"||'   '||vrow."과목
명"||'   '||vrow."과목시작날짜"||'   '||vrow."과목종료날짜"||'   '||vrow."교재명"||'   '||vrow."교육생 등록 인원");

end loop;

end procTCourseStudentInfo;

begin

procTCourseStudentInfo(1);

```

```
end;
```

```
--1.2.3 강의종료 스케줄 조회
```

```
create or replace procedure procTTeacherScheduleEnd(
```

```
    vteacher_seq number
```

```
)
```

```
is
```

```
cursor vcursor is
```

```
(select
```

```
    os.opensubject_seq as "개설과목번호",
```

```
    c.course_name as "과정명",
```

```
    oc.startdate as "과정시작날짜",
```

```
    oc.enddate as "과정종료날짜",
```



```
cl.classroom_seq || '강의실' as "강의실",

s.subject_name as "과목명",

os.startdate as "과목시작날짜",

os.enddate as "과목종료날짜",

b.name as "교재명",

(select

    count(sc.student_seq)

from signupcourse sc

    where sc.opencourse_seq = oc.opencourse_seq) as "교육생 등록 인원"

from subject s

    inner join opensubject os on os.subject_seq = s.subject_seq

    inner join opencourse oc on oc.opencourse_seq = os.opencourse_seq

    inner join course c on c.course_seq = oc.course_seq
```

```
inner join classroom cl on cl.classroom_seq = oc.classroom_seq

inner join book b on b.book_seq = os.book_seq

where os.teacher_seq = vteacher_seq and os.enddate < sysdate );

begin

for vrow in vcursor

loop

dbms_output.put_line(vrow."개설과목번호"||'   '||vrow."과정명"||'   '||vrow."과정시작날짜"||'   '||vrow."과정종료날짜"||'   '||vrow."강의실"||'   '||vrow."과목
명"||'   '||vrow."과목시작날짜"||'   '||vrow."과목종료날짜"||'   '||vrow."교재명");

end loop;


end procTCourseStudentInfo;


begin
```

```
procTCourseStudentInfo(1);
```

```
end;
```

--2.2 강의중인 개설과목의 개설과정 내 교육생 정보 확인

```
create or replace procedure procTCourseStudentInfo(
```

```
    vteacher_seq number ,
```

```
    voc_seq number
```

```
)
```

```
is
```

```
cursor vcursor is
```

```
(select
```

```
    oc.opencourse_seq as "개설과정번호",
```

```
    st.name as "교육생명",
```

st.tel as "전화번호",

st.student_date as "등록일",

sc.completion as "수료상태"

from teacher t

inner join opensubject os on os.teacher_seq = t.teacher_seq

inner join opencourse oc on oc.opencourse_seq = os.opencourse_seq

inner join signupcourse sc on sc.opencourse_seq = oc.opencourse_seq

inner join student st on st.student_seq = sc.student_seq

inner join subject s on s.subject_seq = os.subject_seq

where t.teacher_seq = vteacher_seq and oc.opencourse_seq = voc_seq and os.startdate < sysdate and os.enddate > sysdate);

begin

for vrow in vcursor

loop

```
dbms_output.put_line(vrow."개설과정번호"||'   '||vrow."교육생명"||'   '||vrow."전화번호"||'   '||vrow."등록일"||'   '||vrow."수료상태");

end loop;

end procTCourseStudentInfo;

begin

procTCourseStudentInfo(9,11);

end;
```

요구사항번호	C-02	작성일	2023-04-05
--------	------	-----	------------

요구사항명	배점 입출력	작성자	김경민
DML			
--2. 배점 확인 과목 목록 출력 create or replace procedure procTGradePercent(vtseq number) is cursor vcursor is (select s.subject_seq as "과목번호", c.course_name as "과정명", oc.startdate as "과정시작", oc.enddate as "과정끝", oc.classroom_seq as "강의실",			

s.subject_name as "과목명",

os.startdate as "과목시작",

os.enddate as "과목끝",

b.name as "교재명",

gp.attendance as "출결",

gp.wscore as "필기",

gp.psocre as "실기"

from subject s

inner join opensubject os on os.subject_seq = s.subject_seq

inner join opencourse oc on oc.opencourse_seq = os.opencourse_seq

inner join course c on c.course_seq = oc.course_seq

inner join teacher t on t.teacher_seq = os.teacher_seq

inner join book b on b.book_seq = os.book_Seq

```
inner join gradepercent gp on gp.opensubject_seq = os.opensubject_seq

    where t.teacher_Seq = vtseq);

begin

for vrow in vcursor

loop

dbms_output.put_line(vrow."과목번호"||'    '||vrow."과정명"||'    '||vrow."과정시작"||'    '||vrow."과정끝"||'    '||vrow."강의실"||'    '||vrow."과목명"||'    '||vrow."
과목시작"||'    '||vrow."과목끝"||'    '||vrow."교재명"||'    '||vrow."출결"||'    '||vrow."필기"||'    '||vrow."실기");

end loop;

end procTGradePercent;

begin

procTGradePercent(1);
```



```
end;
```

```
--3.1 강의종료된 개설과목 배점 확인
```

```
create or replace procedure procTEndCourseGradePercent(
```

```
vteacher_seq number,
```

```
venddate date)
```

```
is
```

```
cursor vcursor is
```

```
(select
```

```
os.opensubject_seq as "개설과목번호",
```

```
s.subject_name as "과목명",
```

```
gp.attendance as "출결배점",
```

```
gp.wscore as "필기배점",
```

```

gp.psocre as "실기배점"

from subject s

    inner join opensubject os on os.subject_seq = s.subject_seq

    inner join gradepercent gp on os.opensubject_seq = gp.opensubject_seq

    inner join teacher t on t.teacher_seq = os.teacher_seq

        where t.teacher_seq = vteacher_seq and os.enddate < sysdate);

begin

for vrow in vcursor

loop

dbms_output.put_line(vrow."개설과목번호"||'   '||vrow."과목명"||'   '||vrow."출결배점"||'   '||vrow."필기배점"||'   '||vrow."실기배점");

end loop;

end procTEndCourseGradePercent;

```

```
begin
```

```
procTEndCourseGradePercent(1,sysdate);
```

```
end;
```

```
--해당 개설과목의 배점 수정
```

```
--update gradepercent set <수정할 내용> where opensubject_seq = <개설과목번호>;
```

```
create or replace procedure procTupdataGrade(
```

```
vopensubject_seq number,
```

```
vattendance number,
```

```
vwscore number,
```

```
vpsocre number
```

```
)
```

```
is
```

```
begin

update gradepercent set attendance=vattendance,wscore=vwscore,psocre=vpsocre

where opensubject_seq = vopensubject_seq;

dbms_output.put_line('성공');

end procTupdateGrade;


--delete from gradepercent where opensubject_seq = <개설과목번호>;

create or replace procedure procTdeleteGrade(

vopensubject_seq number)

is

begin

delete from gradepercent where opensubject_seq = vopensubject_seq;

end procTdeleteGrade;
```

--

요구사항번호	C-03	작성일	2023.04.05
요구사항명	시험 성적 관리	작성자	이민지
DML			

-- 1. 성적 입출력

-- 1-1. 종료 과목 한정 과목 목록 출력

-- 1-2. 특정 선생님의 종료 과목 중 특정 과목 선택

- 특정 선생님의 종료 과목 목록을 가져오기 위한 [view vwTEndSubject](#)

create or replace view vwTEndSubject

as

select distinct

os.opensubject_seq as "과목번호",

c.course_name as "과정명",

oc.startDate as "과정 시작일",

oc.endDate as "과정 종료일",

oc.openCourse_seq as "개설 과정 번호",

oc.classroom_seq as "강의실",

```
s.subject_name as "과목명",  
  
os.startDate as "과목 시작일",  
  
os.endDate as "과목 종료일",  
  
b.name as "교재명",  
  
gp.attendance as "출석 배점",  
  
gp.wscore as "필기 배점",  
  
gp.psocre as "실기 배점",  
  
t.name as "담당 선생님",  
  
case  
  
    when s.opensubject_seq in (select opensubject_seq from score group by openSubject_seq) then 'O'  
  
    when s.opensubject_seq not in (select opensubject_seq from score group by openSubject_seq) then 'X'  
  
end as "성적 등록 여부"  
  
from opensubject os  
  
inner join teacher t on os.teacher_seq = t.teacher_seq
```

```
inner join subject s on s.subject_seq = os.subject_seq

inner join openCourse oc on oc.openCourse_seq = os.openCourse_seq

inner join course c on oc.course_seq = c.course_seq

inner join book b on b.book_seq = os.book_seq

inner join gradePercent gp on gp.openSubject_seq = os.openSubject_seq

inner join score s on s.opensubject_seq = s.opensubject_seq

where sysdate > os.enddate

order by oc.startDate;
```

EX) VIEW 활용 : select * from vwTEndSubject where "담당 선생님" = '조성아'; -- 강의 종료 과목을 보고싶은 선생님을 입력

--특정 교사를 선택할 경우, 해당하는 강의목록만 출력되는 프로시저

```
create or replace procedure procT_ShowEndSubjectList (

    tname in varchar2

)
```


is

```
cursor vcursor is select * from vwTEndSubject where "담당 교사명" = tname;
```

begin

```
dbms_output.put_line('과목번호' || ' ' || '과정명' || ' ' || '과정 시작일' || ' ' || '과정 종료일' || ' ' || '강의실' || ' ' || '과목명' || ' ' || '과목 시작일' || ' ' || '과목 종료일' || ' ' || '교재명' || ' ' || '출석 배점' || ' ' || '필기 배점' || ' ' || '실기 배점' || ' ' || '성적 등록 여부' );
```

```
for vrow in vcursor loop
```

```
dbms_output.put_line(vrow."과목번호" || ' ' || vrow."과정명" || ' ' || vrow."과정 시작일" || ' ' || vrow."과정 종료일" || ' ' || vrow."강의실" || ' ' || vrow."과목명" || ' ' || vrow."과목 시작일" || ' ' || vrow."과목 종료일" || ' ' || vrow."교재명" || ' ' || vrow."출석 배점" || ' ' || vrow."필기 배점" || ' ' || vrow."실기 배점" || ' ' || vrow."성적 등록 여부" );
```

```
end loop;
```

end;

-- 강의를 마친 과목을 출력할 시, 원하는 교사명 입력

begin

```
procT_ShowEndSubjectList('조성아');
```

```
end;
```

--1.3 종료 과목 번호 선택시, 교육생 정보 출력

- 종료과목을 수강했던 교육생 정보 출력을 위한 [view vwTShowSignupScoreInfo](#)

```
create or replace view vwTShowSignupScoreInfo
```

```
as
```

```
select
```

```
    "담당 선생님",
```

```
    "과목번호",
```

```
    s.student_seq as "교육생 번호",
```

```
    s.name as "이름",
```

```
    s.tel as "전화번호",
```

```
    sc.completion as "수료/중도탈락",
```

```
case
```

```
    when sc.completion = '수료' then 'null'
```

```
    when sc.completion = '중도탈락' then sc.signupCourse_date || "
```

```
end as "중도탈락 날짜",
```

```
sco.asscore as "출결 점수",
```

```
sco.wscore as "필기 점수",
```

```
sco.pscore as "실기 점수"
```

```
from vwTEndSubject ve
```

```
inner join openCourse oc on ve."개설 과정 번호" = oc.openCourse_seq
```

```
inner join signupCourse sc on oc.openCourse_seq = sc.openCourse_seq
```

```
inner join student s on s.student_seq = sc.student_seq
```

```
left outer join score sco on sco.opensubject_seq = ve."과목번호" and sco.student_seq = sc.student_seq;
```

EX) VIEW 활용 : select * from vwTShowSignupScoreInfo where "담당 선생님" = '조성아' and "과목번호" = 2; - 담당 선생님 입력, 교육생 정보를 보고 싶은 과목번호 입력

- 교육생 정보를 보고싶은 과목번호와 담당 교수명 입력하면 나타나는 교육생 성적 정보 출력 프로시저

```
create or replace procedure procT_ShowESL_StudentScore (
```

```
    tname in varchar2,
```

```
    tseq in number,
```

```
    outname out varchar2,
```

```
    outseq out number
```

```
)
```

```
is
```

```
    cursor vcursor is select * from vwTShowSignupScoreInfo where "담당 교수명" = tname and "과목번호" = tseq;
```

```
begin
```

```
    outname := tname;
```

```
    outseq := tseq;
```

```
    dbms_output.put_line('번호' || ' ' || '이름' || ' ' || '전화번호' || ' ' || '수료/중도탈락' || ' ' || '출결 점수' || ' ' || '필기 점수' || ' ' || '실기 점수' );
```

```
    for vrow in vcursor loop
```

```

        dbms_output.put_line(vrow."교육생 번호" || ' ' || vrow."이름" || ' ' || vrow."전화번호" || ' ' || vrow."수료/중도탈락" || ' ' || vrow."출결 점수" || ' ' || vrow."필기 점수" || ' ' || vrow."실기 점수" );

    end loop;

end;

--호출

declare

    teacherName varchar2(300);

    teacherSeq number;

begin

    -- 특정 선생님의 특정 과목 선택

    procT_ShowESL_StudentScore('조성아', 2);

end;

```

- 1.3 특정 교육생 선택시 (교육생 이름 : 박린희), 시험 점수 입력 가능하도록

- EX) VIEW 활용 :

```
update score set ascore=80, wscore=96, pscore=89 where openSubject_seq =
```

```
(select
```

```
    distinct "과목번호"
```

```
from vwTShowSignupScoreInfo
```

```
where "담당 선생님" = '조성아' and "과목번호" = 2) and
```

```
student_seq = (select
```

```
    "교육생 번호"
```

```
from vwTShowSignupScoreInfo
```

```
    where "담당 선생님" = '조성아' and "과목번호" = 2 and "이름" = '박린희'); --담당 선생님과 과목번호 지정 후, 성적을 입력 하고싶은 교육생  
명을 입력하고 성적 입력
```

- 프로시저

```
create or replace procedure procT_updateStudentScore (  
    tname in varchar2,  
    tseq in number,  
    sname in varchar2,  
    i_ascore in varchar2,  
    i_wscore in varchar2,  
    i_pscore in varchar2  
)  
is  
begin  
  
    update score set ascore=i_ascore, wscore=i_wscore, pscore=i_pscore  
    where openSubject_seq =  
        (select
```

```
distinct "과목번호"
```

```
from vwTShowSignupScoreInfo
```

```
where "담당 교사명" = tname and "과목번호" = tseq) and
```

```
student_seq = (select
```

```
"교육생 번호"
```

```
from vwTShowSignupScoreInfo
```

```
where "담당 교사명" = tname and "과목번호" = tseq and "이름" = sname);
```

```
dbms_output.put_line(sname || ' 학생의 성적이 성공적으로 입력되었습니다.');
```

```
end;
```

```
--호출
```

```
declare
```



```

teacherName varchar2(300);

teacherSeq number;

begin

    -- 1. 성적 보고싶은 교수명과 과목 번호 입력하고

    procT_ShowESL_StudentScore('조성아', 2, teacherName, teacherSeq);

    --2. 성적 입력하고 싶은 학생 번호와 출결/필기/실기 점수 넣기

    procT_updateStudentScore(teacherName, teacherSeq, '배린서', 70, 80, 88);

end;

```

요구사항번호	C-04	작성일	2023.04.05
요구사항명	출결 관리	작성자	이민지

DML

-- 2. 출결 관리 및 출결 조회

-- 출결은 특정 교사가 강의한 과정에 한해서 출결현황 열람 가능

-- 개설 과정 + 담당 선생님 리스트 출력을 위한 [view vwTshowCourseList](#)

create or replace view vwTshowCourseList

as

select

 distinct oc.opencourse_seq as "과정 번호",

 t.name as "선생님",

 c.course_name as "강의 과정명"

from opensubject os

inner join openCourse oc on os.openCourse_seq = oc.openCourse_seq

inner join teacher t on os.teacher_seq = t.teacher_seq

inner join course c on c.course_Seq = oc.course_seq

```
order by oc.opencourse_seq;
```

```
-- 2.1 특정 교사가 강의한 과정
```

```
--2.2 특정 교사가 강의한 과정의 모든 교육생의 출결 조회 (전체 과정 출결) --- 버퍼 용량 초과로 프로시저 생성 불가
```

```
-- 특정 교사가 강의한 전체 과정에서 모든 교육생의 출결을 위한 view vwTAlI CourseStudentAttendance
```

```
create or replace view vwTAlI CourseStudentAttendance
```

```
as
```

```
select
```

```
distinct sc.openCourse_seq as "과정번호",
```

```
    c.course_name as "과정명",
```

```
    t.name as "교수명",
```

```
    sc.student_seq as "학생번호",
```

```
    s.name as "학생명",
```

```
    att.attendance_date as "날짜",
```

```
TO_CHAR(att.starttime, 'hh24:mi:ss') as "출근 시간",
```

```
TO_CHAR(att.endtime, 'hh24:mi:ss') as "퇴근 시간",
```

```
att.etc as "기타"
```

```
from opensubject os
```

```
inner join openCourse oc on os.openCourse_seq = oc.openCourse_seq
```

```
inner join teacher t on os.teacher_seq = t.teacher_seq
```

```
inner join signupCourse sc on sc.opencourse_seq = oc.openCourse_seq
```

```
inner join attendance att on att.student_seq = sc.student_seq
```

```
inner join course c on oc.course_seq = c.course_seq
```

```
inner join student s on sc.student_seq = s.student_seq
```

```
order by openCourse_seq;
```

```
EX) select * from vwAllCourseStudentAttendance where "교수명" = '조성아'; -- 모든 과정의 교육생 출결을 보고싶은 교수명을 입력
```

--2.3 특정 교사가 강의한 특정 과정의 모든 교육생의 출결 조회(과정별 조회)

-- '조성아'라는 교수의 과정번호 '1'에 해당하는 출력 : view 활용

```
select * from vwTAlCourseStudentAttendance where "교수명" = '조성아' and "과정번호" = 1;
```

-- '조성아'라는 교수의 과정번호 '1'에 해당하는 출력 : 프로시저

```
create or replace procedure procT_CourseAttendance (
```

```
    tname in varchar2,
```

```
    tseq in number
```

```
)
```

```
is
```

```
    cursor vcursor is select * from vwTAlCourseStudentAttendance where "교수명" = tname and "과정번호" = tseq order by "과정번호", "학생번호", "날짜";
```

```
begin
```

```
    dbms_output.put_line('과정번호' || ' ' || '과정명' || ' ' || '교수명' || ' ' || '학생명' || ' ' || '날짜' || ' ' || '출근 시간' || ' ' || '퇴근 시간' || ' ' || '기타' );
```

```

for vrow in vcursor loop

    dbms_output.put_line(vrow."과정번호" || ' ' || vrow."과정명" || ' ' || vrow."교수명" || ' ' || vrow."학생명" || ' ' || vrow."
날짜" || ' ' || vrow."출근 시간" || ' ' || vrow."퇴근 시간" || ' ' || vrow."기타" );

end loop;

end;

-- 호출

begin

    procT_CourseAttendance('조성아', 1);

end;

-- 2.4 특정 교사가 강의한 과정에서 근태상황으로 출결 조회(인원별 조회)

--2.4.1 view

-- 1) 정상

```

```
select * from vwTAlI CourseStudentAttendance where "교수명" = '조성아' and ("기타" is null or "기타" = 'null');
```

```
-- 2) 조퇴, 외출, 병가
```

```
select * from vwTAlI CourseStudentAttendance where "기타" = [조퇴,외출,병가];
```

```
-- 3) 지각
```

```
select * from vwTAlI CourseStudentAttendance where "교수명" = '조성아' and "출근 시간" > to_char('09:10:00');
```

```
--2.4.2 프로시저
```

```
create or replace procedure procT_statusAttendance (
```

```
    tname in varchar2,
```

```
    status in varchar2
```

```
)
```

```
is
```

```
    cursor vcursor1 is select * from vwTAlI CourseStudentAttendance where "교수명" = tname and "기타" = status; -- 병가, 외출, 조퇴
```

```
    cursor vcursor2 is select * from vwTAlI CourseStudentAttendance where "교수명" = tname and "출근 시간" > to_char('09:10:00'); -- 지각
```

```
cursor vcursor3 is select * from vwTAlIcourseStudentAttendance where "교수명" = tname and ("기타" is null or "기타" = 'null'); --정상
```

```
begin
```

```
    dbms_output.put_line('과정번호' || ' ' || '과정명' || ' ' || '교수명' || ' ' || '학생명' || ' ' || '날짜' || ' ' || '출근 시간' || ' ' || '퇴근 시간' || ' ' || '기타' );
```

```
case
```

```
    when status = '병가' then
```

```
        for vrow1 in vcursor1 loop
```

```
            dbms_output.put_line(vrow1."과정번호" || ' ' || vrow1."과정명" || ' ' || vrow1."교수명" || ' ' || vrow1."학생명" || ' ' || vrow1."날짜" || ' ' || vrow1."출근 시간" || ' ' || vrow1."퇴근 시간" || ' ' || vrow1."기타" );
```

```
        end loop;
```

```
    when status = '외출' then
```

```
        for vrow1 in vcursor1 loop
```

```
            dbms_output.put_line(vrow1."과정번호" || ' ' || vrow1."과정명" || ' ' || vrow1."교수명" || ' ' || vrow1."학생명" || ' ' || vrow1."날짜" || ' ' || vrow1."출근 시간" || ' ' || vrow1."퇴근 시간" || ' ' || vrow1."기타" );
```



```

        end loop;

    when status = '조퇴' then

        for vrow1 in vcursor1 loop

            dbms_output.put_line(vrow1."과정번호" || ' ' || vrow1."과정명" || ' ' || vrow1."교수명" || ' ' || vrow1."학생명" || ' ' || vrow1."날짜" || ' ' || vrow1."출근 시간" || ' ' || vrow1."퇴근 시간" || ' ' || vrow1."기타" );

        end loop;

    when status = '지각' then

        for vrow2 in vcursor2 loop

            dbms_output.put_line(vrow2."과정번호" || ' ' || vrow2."과정명" || ' ' || vrow2."교수명" || ' ' || vrow2."학생명" || ' ' || vrow2."날짜" || ' ' || vrow2."출근 시간" || ' ' || vrow2."퇴근 시간" || ' ' || vrow2."기타" );

        end loop;

    when status = '정상' then

        for vrow3 in vcursor3 loop

            dbms_output.put_line(vrow3."과정번호" || ' ' || vrow3."과정명" || ' ' || vrow3."교수명" || ' ' || vrow3."학생명" || ' ' || vrow3."날짜" || ' ' || vrow3."출근 시간" || ' ' || vrow3."퇴근 시간" || ' ' || vrow3."기타" );

        end loop;

```

```
end case;
```

```
end;
```

```
set serveroutput on;
```

```
-- 호출
```

```
begin
```

```
    procT_statusAttendance('조성아', '지각');
```

```
end;
```

```
-- 2.5 특정 교사가 강의한 과정에서 기간별로 출결 조회(기간별 조회)
```

```
-- '조성아'라는 교수가 강의한 과정 중 2022-04-01 ~ 2022-04-10 사이의 출결 조회 : view 활용
```

```
select * from vwTAlI CourseStudentAttendance where "교수명" = '조성아' and "날짜" between '2022-04-01' and '2022-04-10';
```

```
-- '조성아'라는 교수가 강의한 과정 중 2022-04-01 ~ 2022-04-10 사이의 출결 조회 : 프로시저
```

```
create or replace procedure procT_periodAttendance (
```

```
    tname in varchar2,
```

```
    startdate in varchar2,
```

```
    enddate in varchar2
```

```
)
```

```
is
```

```
    cursor vcursor is select * from vwTAlI CourseStudentAttendance where "교수명" = tname and "날짜" between startdate and enddate;
```

```
begin
```

```
    dbms_output.put_line('과정번호' || ' ' || '과정명' || ' ' || '교수명' || ' ' || '학생명' || ' ' || '날짜' || ' ' || '출근 시간' || ' ' || '퇴근 시간' || ' ' || '기타' );
```

```
    for vrow in vcursor loop
```

```
        dbms_output.put_line(vrow."과정번호" || ' ' || vrow."과정명" || ' ' || vrow."교수명" || ' ' || vrow."학생명" || ' ' || vrow."
```

```

날짜" || '      ' || vrow."출근 시간" || '      ' || vrow."퇴근 시간" || '      ' || vrow."기타" );

        end loop;

end;

-- 호출

begin

        procT_periodAttendance('조성아', '2022-04-01', '2022-04-10');

end;

```

요구사항번호	C-05	작성일	2023-04-05
요구사항명	Q&A 게시판	작성자	조아현
DML			

--질문 목록 조회 & 교사 목록 조회

select * from vwTQuestionList;

select * from teacher;

--답변등록

create or replace procedure t_answer_in(

 paseq number,

 pcontent varchar2,

 pdate date,

 ptseq number,

 pqseq number

)

is

```
begin
```

```
    insert into answer values(paseq, pcontent, pdate, ptseq, pqseq);
```

```
end;
```

```
--호출문
```

```
begin
```

```
    t_answer_in(answer_seq.nextVal,'내용','22/01/01',2,3);
```

```
end;
```

```
--답변 수정
```

```
create or replace procedure t_answer_up(
```

```
    paseq number,
```

```
    pcontent varchar2,
```

```
    pdate date,
```

```
ptseq number,  
  
pqseq number  
  
)  
  
is  
  
begin  
  
    update answer set answer_seq = paseq, content = pcontent,  
  
        answer_date = pdate, teacher_seq = ptseq, question_seq = pqseq  
  
        where answer_seq = paseq;  
  
end;  
  
--호출문  
  
begin  
  
    t_answer_up(1,'내용','22/01/01',1,2);  
  
end;
```

--답변 삭제

create or replace procedure t_answer_de(

 paseq number

)

is

begin

 delete from answer where answer_seq = paseq;

end;

--호출문

begin


```
t_answer_de(2);

end;
```

요구사항번호	D-01	작성일	2023-04-05
요구사항명	교육생 성적조회	작성자	최재현
PROCEDURE			
--1번 학생이 수강중인 과정 기간, 과정명, 강의실 번호			
CREATE OR REPLACE PROCEDURE procSInfo(pnum number) is			

```
cursor vcursor is SELECT * FROM vwSInfo

WHERE "학생번호" = pnum;

begin

  for vrow in vcursor loop

    dbms_output.put_line(vrow.과정명 || ' | ' || vrow."시작 날짜" || ' | ' || vrow."종료 날짜" || ' | ' || vrow.강의실);

  end loop;

end procSInfo;

/

--실행

begin

  procSInfo(1);

end;

/
```

--1번학생의 성적조회

```
CREATE OR REPLACE PROCEDURE procSscore(
```

```
    pnum number
```

```
)
```

```
is
```

```
    cursor vcursor is SELECT * FROM vwSscore
```

```
    WHERE "학생번호" = pnum;
```

```
begin
```

```
    for vrow in vcursor loop
```

```
        dbms_output.put_line(vrow.과목 || ' ' || vrow."시험일자" || ' ' || vrow."출결점수" || ' ' || vrow.필기점수 || ' ' || vrow.실기점수);
```

```
    end loop;
```

```
end procSscore;
```

```
/
```

--실행

begin

 procSscore(1);

end;

/

--문제까지 같이 조회

CREATE OR REPLACE PROCEDURE procSscore_Quiz(

 pnum number

)

is

 cursor vcursor is SELECT * FROM vwSscore_Quiz

 WHERE "학생번호" = pnum;

begin

 for vrow in vcursor loop

```
dbms_output.put_line(vrow.과목 || ' ' || vrow."시험일자" || ' ' || vrow."출결점수" || ' ' || vrow.필기점수 || ' ' || vrow.실기점수 || ' ' || vrow.문제내용);
```

```
end loop;
```

```
end procSscore_Quiz;
```

```
/
```

```
--실행
```

```
begin
```

```
procSscore_Quiz(1);
```

```
end;
```

요구사항번호	D-02	작성일	2023-04-05
요구사항명	학생 출결 관리 및 출결 조회	작성자	최재현
PROCEDURE			
<pre>--학생 본인 전체 출결 조회 CREATE OR REPLACE PROCEDURE procSAttendance(pnumber number) is cursor vcursor is SELECT * FROM attendance where student_seq = pnumber order by attendance_date; vother varchar2(10);</pre>			

```
begin

  for vrow in vcursor loop

    case

      when vrow.starttime is null and vrow.endtime is null and vrow.etc is null then vother := '휴일';

      when to_char(vrow.starttime, 'hh24-mi-dd') < to_char('09:10:00') and vrow.endtime is not null and vrow.etc = 'null' then vother := '정상';

      when to_char(vrow.starttime, 'hh24-mi-dd') > to_char('09:10:00') and vrow.endtime is not null and vrow.etc = 'null' then vother := '지각';

      else vother := vrow.etc;

    end case;

    dbms_output.put_line('날짜: ' || vrow.attendance_date || ',      출근시간: ' || to_Char(vrow.starttime,'hh24:mi:ss') || ',      퇴근시간: ' ||
to_char(vrow.endtime,'hh24:mi:ss') || ',      기타: ' || vother);

  end loop;

end procSAttendance;
```

--학생 본인 한달 출결 조회

CREATE OR REPLACE PROCEDURE procSAAttendance_month(

 pnumber number,

 pstartdate date,

 penddate date

)

is

 cursor vcursor is

 SELECT *

 FROM attendance

 where student_seq = pnumber and attendance_date >= pstartdate and attendance_date <= penddate

 order by attendance_date;

 vother varchar2(10);

begin


```

for vrow in vcursor loop

    case

        when vrow.starttime is null and vrow.endtime is null and vrow.etc is null then vother := '휴일';

        when to_char(vrow.starttime, 'hh24-mi-dd') < to_char('09:10:00') and vrow.endtime is not null and vrow.etc = 'null' then vother := '정상';

        when to_char(vrow.starttime, 'hh24-mi-dd') > to_char('09:10:00') and vrow.endtime is not null and vrow.etc = 'null' then vother := '지각';

        else vother := vrow.etc;

    end case;

    dbms_output.put_line('날짜: ' || vrow.attendance_date || ',      출근시간: ' || to_Char(vrow.starttime,'hh24:mi:ss') || ',      퇴근시간: ' ||
to_char(vrow.endtime,'hh24:mi:ss') || ',      기타: ' || vother);

end loop;

end procSAttendance_month;

--학생 본인 하루 출결 조회

```

```
CREATE OR REPLACE PROCEDURE procSAttendance_day(  
    pnumber number,  
    pdate date  
)  
is  
    cursor vcursor is  
        SELECT *  
        FROM attendance  
        where student_seq = pnumber and attendance_date = pdate  
        order by attendance_date;  
  
    vother varchar2(10);  
begin  
    for vrow in vcursor loop  
        case
```

```

        when vrow.starttime is null and vrow.endtime is null and vrow.etc is null then vother := '휴일';

        when to_char(vrow.starttime, 'hh24-mi-dd') < to_char('09:10:00') and vrow.endtime is not null and vrow.etc = 'null' then vother := '정상';

        when to_char(vrow.starttime, 'hh24-mi-dd') > to_char('09:10:00') and vrow.endtime is not null and vrow.etc = 'null' then vother := '지각';

        else vother := vrow.etc;

    end case;

    dbms_output.put_line('날짜: ' || vrow.attendance_date || ',      출근시간: ' || to_Char(vrow.starttime,'hh24:mi:ss') || ',      퇴근시간: ' ||
to_char(vrow.endtime,'hh24:mi:ss') || ',      기타: ' || vother);

end loop;

end procSAttendance_day;

--프로시저 호출

begin

```

```
procsAttendance(1); 전체

--      procsAttendance_month(1, '2022-03-01', '2022-04-01'); 한달

--      procSAttendance_day(1, '2022-03-01'); 하루

end;
```

요구사항번호	D-03	작성일	2023-04-05
요구사항명	수업계획서 조회	작성자	최재현
PROCEDURE			
<pre>--수업 계획서 조회 CREATE OR REPLACE PROCEDURE procScourseplan(pnum number</pre>			

```
)  
  
is  
  
    cursor vcursor is SELECT * FROM vwScourseplan  
  
    WHERE "개설과정번호" = pnum;  
  
begin  
  
    for vrow in vcursor loop  
  
        dbms_output.put_line(vrow.과정명 || ' | ' || vrow."계획번호" || ' | ' || vrow."계획 월" || ' | ' || vrow.계획내용);  
  
    end loop;  
end procScourseplan;  
  
/  
  
--실행  
  
begin  
  
    procScourseplan(1);
```

end;

요구사항번호	D-04 교육생 > 멘토	작성일	2023-04-05
요구사항명	멘토 등록 및 커리큘럼 관리	작성자	최재현
PROCEDURE			
<p>--멘토 등록 > 초기에 커리큘럼 번호는 NULL로 설정.</p> <p>CREATE OR REPLACE procedure procSinsertMento(pstudent_seq number, pcurriculum_seq number) is</p>			

```
begin
```

```
    INSERT INTO MENTO VALUES(pstudent_seq, pcurriculum_seq);
```

```
end;
```

```
--실행
```

```
begin
```

```
    procSinsertMento(355, 5);
```

```
end;
```

```
--커리큘럼 등록
```

```
CREATE OR REPLACE procedure procSinsertCurriculum(
```

```
    ptitle varchar2,
```

```
pcontent varchar2,  
pcapacity number  
)  
is  
  
begin  
    INSERT INTO Curriculum VALUES(nvl(max(curriculum_seq, 0) + 1), ptitle, pcontent, pcapacity);  
end procSinsertCurriculum;  
  
/  
  
begin  
    procSinsertCurriculum('JAVA 개발자 스파르타 과정', '스파르타 300을 보고 영감받았습니다.', 300);  
end;  
  
--커리큘럼 제목 수정
```



```
CREATE OR REPLACE procedure procSupdateTitle(  
    pnum number,  
    ptitle varchar2  
)  
is  
  
begin  
    UPDATE curriculum set title = ptitle where curriculum_seq = pnum;  
end;  
  
/  
  
--begin  
--    procupdateTitle('수정할 번호', '수정할 제목');  
--end;  
  
--커리큘럼 내용 수정  
  
CREATE OR REPLACE procedure procSupdateContent(
```

```
pnum number,  
pcontent varchar2  
)  
is  
  
begin  
    UPDATE curriculum set content = pcontent where curriculum_seq = pnum;  
end;  
  
/  
  
--begin  
--    procupdateTitle('수정할 번호', '수정할 내용');  
--end;  
  
--커리큘럼 정원 수정  
  
CREATE OR REPLACE procedure procSupdateCapacity(
```

```
pnum number,  
pcapacity varchar2  
)  
is  
  
begin  
    UPDATE curriculum set capacity = pcapacity where curriculum_seq = pnum;  
end;  
  
/  
  
--begin  
  
--    procupdateCapacity('수정할 번호', 수정할 정원 수);  
  
--end;  
  
  
--커리큘럼 삭제  
  
CREATE OR REPLACE procedure procsDELETEDCurriculum(
```

```
pnum number
)
is

begin
    DELETE FROM curriculum where curriculum_seq = pnum;
end;

/

--begin

--    procsDELETEDCurriculum(삭제할 커리큘럼 번호);

--end;


--커리큘럼 조회

SELECT * FROM CURRICULUM;
```

요구사항번호	D-05 교육생 > 멘티	작성일	2023-04-05
요구사항명	멘티 등록 및 커리큘럼 참여	작성자	최재현
PROCEDURE			
<pre>--멘티 등록 > 초기에 커리큘럼 번호는 NULL로 설정. CREATE OR REPLACE procedure procsInsertMenti(pnum number, pcurriculum_seq number,) is begin INSERT INTO menti values(pnum, pcurriculum_seq); end;</pre>			

```
/
--begin
--    procsInsertMenti(학생번호, 커리큘럼번호);
--end;
--듣고 있는 커리큘럼 수정
CREATE OR REPLACE procedure procsUpdateMenti(
    pnum1 number,
    pnum2 number
)
is
begin
    UPDATE MENTI set CURRICULUM_SEQ = pnum1; where CURRICULUM_SEQ = pnum2;
end;
/
--begin
```

```
--      procsUpdateMenti(바꿀번호, 현재번호);

--end;


--멘티 탈퇴
CREATE OR REPLACE procedure procsDELETEMenti(
    pnum number
)
is
begin
    DELETE FROM menti WHERE STUDENT_SEQ = PNUM;
end;

/

--begin

--      procsDELETEMenti(학생번호);

--end;
```

--커리큘럼 조회

```
SELECT * FROM CURRICULUM;
```

-- 멘토가 지정해놓은 커리큘럼 정원이 초과되면 더 이상 신청할 수 없도록 제한을 둔다.

```
CREATE or replace procedure procInsertMenti(
```

```
    pstudent_seq number,
```

```
    pcurriculum_seq number
```

```
)
```

```
is
```

```
    cursor menti_cursor is SELECT * FROM vwSTest;
```

```
    cursor capacity_cursor is SELECT * FROM vwSTest2;
```

```
    vresult number;
```

```
    vnum number;
```

```
begin
```



```
vresult := 0;
```

```
for vcrow in capacity_cursor loop
```

```
    if vcrow.curriculum_seq = pcurriculum_seq then
```

```
        SELECT 인원수 into vnum FROM vwSTest where "커리큘럼번호" = pcurriculum_seq;
```

```
        if vnum < vcrow.capacity then
```

```
            dbms_output.put_line('멘티 등록이 완료되었습니다.');
```

```
        else
```

```
            vresult := 1;
```

```
        end if;
```

```
    end if;
```

```
end loop;
```

```
if vresult = 0 then
```

```
    raise_application_error(-20777, '존재하지 않는 과정입니다. 다시입력해주세요.');
```

```
else
```

```
    raise_application_error(-20777, '정원이 가득 찼습니다. 다시 입력해주세요.');
```

```
end if;
```

```
end;
```

```
/
```

```
CREATE or replace Trigger trgSInsertCurriculum
```

```
before
```

```
INSERT
```

```
on menti
```

```
for each row
```

```
declare
```

```
begin
```

```
    procSInsertMenti(:new.student_seq, :new.curriculum_seq);
```

end;

/

SELECT * FROM menti where STUDENT_SEQ = 366;

INSERT INTO menti VALUES(366, 2);

/

요구사항번호	D-06 교육생	작성일	2023-04-05
요구사항명	취업분야 및 취업률 조회	작성자	최재현
PROCEDURE			
--1. 종료된 과정번호, 과정명 검색			
SELECT * FROM vwSjob;			

--2. 과정선택 > 선택한 과정의 취업률, 진출분야 퍼센트 조회

```
CREATE or replace procedure procScourse(
```

```
    pnum number
```

```
)
```

```
is
```

```
    cursor vcursor is SELECT * FROM vwSjob WHERE 과정번호 = pnum;
```

```
begin
```

```
    for vrow in vcursor loop
```

```
        dbms_output.put_line(vrow."과정번호" || ' | ' || vrow."과정명");
```

```
        procSjobPercent(pnum);
```

```
    end loop;
```

```
end;
```

```
/
```

```
begin
```

```

    procScourse(4);

end;

/

--3 호출당하는 프로시저 > 취업률 보여줌.

SELECT * FROM vwSjobList;

/

CREATE or replace procedure procSjobPercent(

    pnum number

)

is

    cursor vcursor is SELECT * FROM vwSjobList WHERE openCourse_seq = pnum;

begin

    for vrow in vcursor loop

        dbms_output.put_line('취업률: ' || vrow."취업률" || ' |   백엔드:' || vrow."백엔드개발자" || ' |   풀스택:' || vrow.풀스택개발자 || ' |   프론트엔드:' || vrow.프

```

론트엔드개발자);

end loop;

end;

/

begin

procSjobPercent(1);

end;

/

요구사항번호	D-07 교육생	작성일	2023-04-05
요구사항명	Q&A 게시판	작성자	최재현
PROCEDURE			

-- 질문조회

```
SELECT * FROM QUESTION;
```

-- 질문 등록

```
CREATE OR REPLACE PROCEDURE PROCSINSERTQUESTION(
```

```
    pcontent varchar2,
```

```
    pstudent_seq number
```

```
)
```

```
is
```

```
    vseq number;
```

```
begin
```

```
    SELECT nvl(max(question_seq), 0) + 1 into vseq FROM QUESTION;
```

```
    INSERT INTO QUESTION VALUES(vseq, pcontent, default, pstudent_seq);
```

```
end;
```

```
/
```

```
--begin
```

```
--    PROCSINSERTQUESTION(내용, 학생번호)
```

```
--end;
```

```
--질문 조회 > 질문 수정
```

```
SELECT * FROM QUESTION;
```

```
CREATE OR REPLACE PROCEDURE PROCSUpdateQUESTION(
```

```
    pcontent varchar2,
```

```
    pquestion_seq number
```

```
)
```


is

begin

UPDATE QUESTION SET CONTNET = pcontent WHERE QUESTION_SEQ = pquestion_seq;

end;

/

--begin

-- PROCUPDATEQUESTION('수정할내용', '질문번호');

--end;

--질문조회 > 게시글 삭제

SELECT * FROM QUESTION;

CREATE OR REPLACE PROCEDURE PROCSDELETEQUESTION(

```
pquestion_seq number  
  
)  
  
is  
  
begin  
    DELETE FROM QUESTION WHERE QUESTION_SEQ = pquestion_seq;  
end;  
  
/  
  
--begin  
--    PROCSDELETEQUESTION('삭제할 질문번호');  
--end;
```

요구사항번호	D-08 교육생	작성일	2023-04-05
요구사항명	취뽀생 게시판	작성자	조아현
PROCEDURE			
--1. 취뽀생(교육생 중 취업에 성공한 사람) --1.1 취뽀생 게시글목록 조회 -- 게시글 번호, 게시글제목 을 볼 수 있다. create or replace procedure procS_ShowJobBoard is cursor vcursor is select successjob_seq as "게시글번호", title as "게시글제목"from successjob ; begin			

```

        dbms_output.put_line('게시글번호' || '          ' || '게시글제목' );

for vrow in vcursor loop

        dbms_output.put_line(vrow."게시글번호" || '          ' || vrow."게시글제목" );

end loop;

end;

begin

        procS_ShowJobBoard;

end;

```

--1.2 게시글 내용 조회

-- 게시글 번호, 게시글 제목, 게시글 내용, 교육생명(작성자명)

-- 댓글 번호, 댓글내용, 교육생명(작성자명)

```
drop view vwSSuccessJobRead;

create or replace view vwSSuccessJobRead

as

select

    sj.successJob_seq as "게시글번호",

    sj.title as "게시글제목",

    sj.content as "게시글내용",

    s.name as "게시글작성자명",

    jc.comment_seq as "댓글번호",

    jc.content as "댓글내용",

    st.name as "댓글작성자명"

from successJob sj

inner join student s on s.student_seq = sj.student_seq

inner join jobcomment jc on jc.successjob_seq = sj.successjob_seq

inner join student st on st.student_seq = jc.student_seq;
```

```
-- 전체 목록 조회 - 전체 게시글, 댓글까지
```

```
select * from vwSSuccessJobRead;
```

```
-- 특정 게시글 번호만 보고싶은 경우
```

```
create or replace procedure procS_ShowJobPostingNum (
```

```
    vseq in number
```

```
)
```

```
is
```

```
    cursor vcursor is select * from vwSSuccessJobRead where "게시글번호" = vseq;
```

```
begin
```

```
    dbms_output.put_line('게시글번호' || ' ' || '게시글제목' || ' ' || '게시글내용' || ' ' || '게시글작성자명' || ' ' || '댓글번호' || ' ' || '댓글내용' || ' ' || '댓글작성자명');
```

```
for vrow in vcursor loop

    dbms_output.put_line('게시글번호 : ' || vrow."게시글번호");

    dbms_output.put_line('게시글제목 : ' || vrow."게시글제목");

    dbms_output.put_line('게시글내용 : ' || vrow."게시글내용");

    dbms_output.put_line('게시글작성자명 : ' || vrow."게시글작성자명");

    dbms_output.put_line('댓글번호 : ' || vrow."댓글번호");

    dbms_output.put_line('댓글내용 : ' || vrow."댓글내용");

    dbms_output.put_line('댓글작성자명 : ' || vrow."댓글작성자명");

end loop;

end;

begin

    procS_ShowJobPostingNum(1);

end;
```

--1.2 취뽀생 게시글 등록

-- 게시글 번호, 게시글제목, 게시글내용, 교육생번호를 사용하여 게시글을 등록한다.

```
insert into successjob values (successjob_seq.nextVal,<게시글제목>,<게시글내용>,<교육생번호>);
```

```
create or replace procedure procS_InsertJobBoard (
```

```
    vtitle varchar2,
```

```
    vcontent varchar2,
```

```
    vstudent_seq number
```

```
)
```

```
is
```

```
    vnum number;
```

```
    vjobseq number;
```

```
begin
```



```
select nvl(max(successJob_seq),0)+1 into vnum from successJob;
```

```
select count(*) into vjobseq from jobStudent where student_seq = vstudent_seq;
```

```
if vjobseq = 0 then
```

```
    dbms_output.put_line('취뽀생만 게시글을 등록할 수 있습니다.');
```

```
end if;
```

```
insert into successjob values (vnum, vtitle, vcontent, vstudent_seq);
```

```
dbms_output.put_line('성공적으로 게시글을 등록하였습니다.');
```

```
end;
```

```
begin
```

```
procS_InsertJobBoard('풋팅', '다들 파이팅팅', 198);  
  
end;
```

--1.3 취뽀생 댓글 등록

-- 댓글번호, 게시글 번호, 댓글 내용, 교육생번호를 사용하여 댓글을 등록한다.

```
insert into jobcomment values(comment_seq.nextVal,<게시글번호>,<댓글내용>,<교육생번호>);
```

```
create or replace procedure procS_InsertJobComment (
```

```
    vpostnum varchar2,
```

```
    vcomment varchar2,
```

```
    vstudent_seq number
```

```
)  
  
is  
  
    vnum number;  
  
    valid number;  
  
begin  
  
    select nvl(max(comment_seq),0)+1 into vnum from jobcomment;  
  
    -- 유효한 게시글 번호인지 확인  
  
    select count(*) into valid from successJob where successjob_seq = vpostnum;  
  
    if valid = 0 then  
  
        dbms_output.put_line('게시글 번호가 유효하지 않습니다.');  
    end if;
```

```
insert into jobcomment values(vnum,vpostnum,vcomment,vstudent_seq);
```

```
dbms_output.put_line('성공적으로 게시글을 등록하였습니다.');
```

```
end;
```

```
select * from jobcomment;
```

```
begin
```

```
    procS_InsertJobComment(9, '감사합니다람쥐', 330);
```

```
end;
```

```
-- 취보게시글 삭제 => 게시글 지우기 전에 댓글 먼저 지우고 게시글 삭제
```

```
create or replace trigger trgDeleteJobPost
```

```
    before
```

```
delete

on successJob

for each row

begin

    delete from jobComment where successjob_seq = :OLD.successjob_seq;

    dbms_output.put_line('게시글을 성공적으로 삭제하였습니다.');
```

end;

```
delete from successJob where successjob_seq = 11;
```

--2. 취준생(모든 교육생)

--2.1 취뽀생 게시글목록 조회

-- 게시글 번호, 게시글제목 을 볼 수 있다.

```
select
```

```
    successjob_seq as "게시글번호",
```

```
    title as "게시글제목"
```

```
from successjob;
```

-- 2.2 게시글 내용 조회

- 게시글 번호, 게시글 제목, 게시글 내용, 교육생명(작성자명)

- 댓글 번호, 댓글내용, 교육생명(작성자명)

```
select
```

```
    sj.successJob_seq as "게시글번호",
```

```
    sj.title as "게시글제목",
```

```
    sj.content as "게시글내용",
```

```
    s.name as "게시글작성자명",
```

```
    jc.comment_seq as "댓글번호",
```

```
    jc.content as "댓글내용",  
    st.name as "댓글작성자명"  
from successJob sj  
inner join student s on s.student_seq = sj.student_seq  
inner join jobcomment jc on jc.successjob_seq = sj.successjob_seq  
inner join student st on st.student_seq = jc.student_seq;
```

-- 2.3 취뽀 게시판 댓글 등록

- 댓글번호, 게시글 번호, 댓글 내용, 교육생 내용을 사용하여 댓글을 등록한다.

```
insert into jobcomment values(comment_seq.nextVal,<게시글번호>,<댓글내용>,<교육생번호>);
```

--2.4 취뽀 게시판 댓글 수정

```
create or replace procedure s_jobcomment_up(  
    pcseq number,
```

```

pgseq number,

pcontent varchar2,

psseq number

)

is

begin

    update jobcomment set comment_seq=pcseq, successjob_seq =pgseq, content=pcontent, student_seq =psseq

        where comment_seq = pcseq;

end;

--호출문

begin

    a_jobcomment_up(2,1,'테스트',2);

end;

--취뽀 게시판 댓글 삭제
```



```
create or replace procedure s_jobcomment_de(  
    pcseq number  
)  
is  
begin  
    delete from jobcomment where comment_seq = pcseq;  
end;  
  
--호출문  
begin  
    a_jobcomment_up(2);  
end;
```