요구사항번호	B-02	작성일	2023-04-05
요구사항명	기초 정보 관리기능	작성자	조아현

DML

1. 과정명 관리 기능

```
select * from course;

insert into course values (course_seq.nextVal, '<과정명>’);

update course set course_name = '<과정명>' where course_seq = 10;

delete from course where course_seq = 19;
```

2. 과목명 관리 기능

```
select * from subject;

insert into subject values (subject_seq.nextVal, '<과목명>');

update subject set subject_name = '<과목명>' where subject_seq = 10;

delete from subject where subject_seq = 19;
```

3. 강의실명 관리 기능

```
select * from classroom;

insert into classroom values (classroom_seq.nextVal, 26);

update classroom set capacity = 30 where classroom_seq = 10;

delete from classroom where classroom_seq = 19;
```

4. 교재명 관리기능

```
select * from book;

insert into book values (book_seq.nextVal, '<교재명>','<저자명>','<출판사>','<발행일>');

update book set name = '<교재명>' where book_seq = 10;

delete from book where book_seq = 19;
```

요구사항번호	B-03	작성일	2023.04.04
요구사항명	교사계정 관리기능	작성자	이민지

DML

```
-- 관리자 : 교사 계정 관리


--1.등록

--1-1. 교사 계정 여러개 등록

select * from teacher;


insert into teacher values (11, '이윤정', '2793012', '010-4234-1121'); -- 1-1. 11번째 교사 등록

select * from subject; -- 1-2. 과목 테이블 확인 후,

insert into ableSuject values (11, 1); --1-3. 교사의 강의 가능 과목 선택적으로 추가, 공통적으로 가능한 과목 : 1,2,7,9,10,13,47
insert into ableSuject values (11, 2);

insert into ableSuject values (11, 3);

insert into ableSuject values (11, 4);

insert into ableSuject values (11, 7);

insert into ableSuject values (11, 9);

insert into ableSuject values (11, 10);

insert into ableSuject values (11, 11);

insert into ableSuject values (11, 13);

insert into ableSuject values (11, 19);

insert into ableSuject values (11, 20);

insert into ableSuject values (11, 23);

insert into ableSuject values (11, 29);

insert into ableSuject values (11, 30);
insert into ableSuject values (11, 33);
insert into ableSuject values (11, 34);
insert into ableSuject values (11, 38);
insert into ableSuject values (11, 40);
insert into ableSuject values (11, 41);
insert into ableSuject values (11, 42);
insert into ableSuject values (11, 47);



--2. 조회
--2-1. 교사 정보 조회
select t.name as "교사명", t.ssn as "주민번호 뒷자리", t.tel as "전화번호", s.subject_name as "강의 가능 과목"
from teacher t
```

```
inner join ableSubject a on t.teacher_seq = a.teacher_seq
inner join subject s on a.subject_seq = s.subject_seq;


--2-2. 특정 교사를 선택한 경우에 조회 => 조성아
select
    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
where t.name = '조성아';
```

```
--3. 수정

update teacher set tel = '010-2019-4818' where name = '이윤정';



--4. 삭제

delete from ableSubject where teacher_seq = 11; --교사 가능 과목 삭제

delete from teacher where name = '이윤정'; -- 교사 삭제
```

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

1. 개설 과정 등록

(1) 과정 등록

```
insert into course values (course_seq.nextVal, <과정명>)
```

(2) 과정 및 강의실 조회

```
select * from classroom;

select * from course;
```

(2) 개설 과정 등록

```
insert into opencourse valuse (opencourse_seq.nextVal, <과정번호>,<강의실번호>,<감독번호>,<과정시작날짜>,<과정종료날짜>, <과정기간>);
```

2. 개설 과정 수정

```
update opencourse set opencourse_name = <과정명> where opencourse_seq = 10;
```

3. 개설 과정 삭제

```
delete from opencourse where opencourse_seq = <개설과정번호>;
```

4. 개설 과정 조회

4.1 개설과정 정보 출력

```
select
    oc.opencourse_seq,
    c.course_name,
    oc.startdate,
    oc.enddate,
    cl.classroom_seq,
    (select
            case
                when count(os.opencourse_seq)>0 then 'O'
                when count(os.opencourse_seq)>0 then 'X'
            end
        from opencourse oc
```

```
            inner join opensubject os
                on os.opencourse_seq = oc.opencourse_seq) as "개설과목등록여부",
    (select
        count(sc.student_seq)
    from signupcourse sc
                where sc.opencourse_seq = oc.opencourse_seq) as "교육생 등록 인원"
from course c
    inner join opencourse oc on c.course_seq = oc.course_seq
    inner join classroom cl on oc.classroom_seq = cl.classroom_seq
            order by oc.opencourse_seq;
```

4.2 특정 개설 과정 선택시 정보 출력

```
select
    s.subject_name as "개설과목명",
    os.startdate as "과목시작날짜",
    os.enddate as "과목종료날짜",

    b.name as "교재명",

    t.name as "교사명",

    st.name as "교육생명",

    substr(st.ssn,8) as "주민번호 뒷자리",

    st.tel as "전화번호",

    st.student_date as "등록일",

    sc.completion as "수료상태"

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 book b on b.book_seq = os.opensubject_seq

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

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

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

```
        where oc.opencourse_seq = <개설과정번호>;
```

5. 교육생 수료상태 지정

5.1 종료과정 내 교육생 수료상태 정보 확인

```
select
    oc.opencourse_seq as "개설과정번호",
    c.course_name as "과정명",
    oc.startdate as "과정시작날짜",
    oc.enddate as "과정종료날짜",
    sc.student_seq as "학생번호",
    st.name as "학생명",
    sc.completion as "수료상태"
from signupcourse sc
    inner join opencourse oc on oc.opencourse_seq = sc.opencourse_seq
    inner join student st on st.student_seq = sc.student_seq
    inner join course c on c.course_seq = oc.course_Seq
        where oc.enddate < sysdate and sc.completion <> '중도탈락';
```

5.2 수료상태 수정

```
update signupcourse set completion = '수료' where opencourse_seq = <개설과정번호>;
```

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

DML

1. 개설 과목 신규 등록

(1) 개설과정,과목, 교재, 교사 조회

```
select * from subject;
select * from book;

select * from teacher;

select * from opencourse;
```

(2) 개설 과목 신규 등록

```
-insert into opensubject values (opensubject_seq.nextVal, <개설과정번호>,<시작날짜>,<종료날짜>,<교재번호>,<과목번호>,<교사번호>
```

2. 개설 과목 정보 출력

```
select

        s.subject_seq as "과목번호",

        c.course_name as "과정명",

        oc.startdate as "과정기간",

        oc.enddate as "과정기간",

        oc.classroom_seq as "강의실",

        s.subject_name as"과목명",
        os.startdate as "과목기간",
        os.enddate as "과목기간",
        b.name as "교재명",
        t.name as "교사명"
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;
```

3. 개설과목 수정

```
update opensubject set <수정대상> where opensubject_seq = <수정할 개설과목 번호>;
```

4. 개설 과목 삭제

```
delete from opensubject where opensubject_seq = <삭제할 개설과목 번호>;
```

요구사항번호	B-06	작성일	2023-04-05
요구사항명	교육생 관리	작성자	조아현

DML

1. 교육생 등록

```
insert into student values(student_seq.nextVal, <교육생명>,<주민번호>,<전화번호>,<등록일>);
```

2. 교육생 출력

2.1 전체 교육생 출력 교육생명, 주민번호 뒷자리, 전화번호, 등록일, 수강신청횟수

```
select
    st.name as "교육생명",
    substr(st.ssn,8) as "주민번호 뒷자리",
    st.tel as "전화번호",
    st.student_date as "등록일"
from student st
    inner join signupcourse sc on st.student_seq = sc.student_seq;
```

2.2 특정 교육생 정보 출력 특정 교육생의 과정명, 과정기간, 강의실, 수료여부, 수료 및 중도탈락 날짜

```
select
    st.name as "교육생명",
    c.course_name as "과정명",
    oc.startdate as "과정시작날짜",
    oc.enddate as "과정종료날짜",
    cl.classroom_seq || '강의실' as "강의실",
    sc.completion as "수료여부",
    case
```

```
        when sc.completion = '중도탈락' then to_char(sc.signupcourse_date)

        when sc.completion = '수료' then to_char(oc.enddate)

        when sc.completion = '수강중' then '수강중'

    end as "수료 및 중도탈락 날짜"

from student st

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

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

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

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

        where st.name = <교육생명>;
```

3. 교육생 수정

```
update student set <수정할 대상> where name = <교육생명>;
```

4. 교육생 삭제

```
delete from student where name = <교육생명>;
```

요구사항번호	B-07	작성일	2023.04.05
요구사항명	시험 관리 및 성적 조회	작성자	김경민
DML			

1. 시험 관리

- 특정 개설 과목 정보(과정 번호, 과목명, 과목기간, 교재명, 교사명, 교육생명, 주민번호 뒤자리, 전화번호, 등록일, 수료 및 중도탈락) 을 출력

```
select
oc.opencourse_seq as "과정번호",
subject.subject_name as "과목명",
(os.startdate || '~'|| os.enddate)as "과목기간",
b.name as "교재명",t.name as "교사명",
s.name as "교육생명", substr(s.ssn,8) as "주민번호뒷자리",
s.tel as "전화번호", s.student_date as "등록일",
sc.completion as "수료 및 중도탈락"
from course c
    inner join opencourse oc
        on c.course_seq = oc.course_seq
            inner join opensubject os
                on oc.opencourse_seq = os.opencourse_seq
                    inner join teacher t
                        on os.teacher_seq = t.teacher_seq
                        inner join subject
                            on os.subject_seq =subject.subject_seq
                                inner join book b
```

```
                                    on os.book_seq = b.book_seq
                                        inner join signupcourse sc
                                            on oc.opencourse_seq=sc.opencourse_seq
                                                inner join student s
                                                    on sc.student_seq = s.student_seq;
```

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.

```
select distinct
oc.opencourse_seq as "과정번호",
os.opensubject_seq as "과목 번호",
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 = <과정번호>
                 order by os.opensubject_seq;
```

2. 성적 조회

2.1 개설 과목별 출력

과정명, 개설 과정기간, 강의실명, 개설과목명, 교사명, 교재명을 출력한 후

```
select
distinct os.opensubject_seq as "과목번호",
c.course_name as "과정명",
oc.period ||'개월' as "과정기간",
oc.classroom_seq as "강의실",
s.subject_name as"개설과목명",
t.name as "교사명",
b.name as "교재명"
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
                                        order by os.opensubject_seq;
```

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

```
select
os.opensubject_seq as "과목번호",
s.name,substr(s.ssn,8) as "주민번호뒷자리",
score.wscore"필기 성적",
score.pscore "실기 성적"
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=<과정번호>
                        order by os.opensubject_seq;
```

2.2 교육생 개인별 출력

- 모든 교육생의 정보(교육생명, 주민번호 뒷자리, 개설과정명, 과정기간, 강의실명)을 출력한 후

```
select s.name as "교육생명",
substr(s.ssn,8) as "주민번호뒷자리",
c.course_name as "개설과정명",
oc.period ||'개월' as "과정기간",
oc.classroom_seq   as "강의실명"
from student s
    inner join SignupCourse sc
       on s.student_seq = sc.student_seq
        inner join opencourse oc
            on   sc.opencourse_seq = oc.opencourse_seq
```

```
                inner join course c
                    on oc.course_seq = c.course_seq
                        order by s.student_seq;
```

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

```
select s.name as "교육생명",
subject.subject_name as "개설 과목명",
os.startdate || ' ~ ' || os.enddate as"개설과목기간",
t.name as "교사명",
score.wscore as "필기점수",
score.pscore as "실기점수"
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 =<교육생번호>;
```

요구사항번호	B-08	작성일	2023.04.05
요구사항명	출결 관리 및 조회	작성자	김경민

DML

-과정 목록을 출력

```
select * from opencourse oc
          where opencourse_seq = <과정번호>;
```

- 선택한 과정 내 모든 교육생의 출결 정보를 조회한다.

```
select oc.opencourse_seq as "과정번호",
c.course_name as "과정명" ,
a.attendance_date as "날짜",
s.name as "교육생명",
case
when a.starttime is null and a.endtime is null and a.etc is null then '휴일'
when a.starttime is null and a.endtime is null and a.etc is not null then a.etc
when to_char(starttime, 'hh24-mi-dd') < to_char('09:10:00') and endtime is not null then '정상'
when to_char(starttime, 'hh24-mi-dd') > to_char('09:10:00') and endtime is not null then '지각'
when a.starttime is not null and a.endtime is null then a.etc
end as "근태상황"
from opencourse oc
    inner join course c
        on oc.course_seq = c.course_seq
            inner join signupcourse sc
                on oc.opencourse_seq = sc.opencourse_seq
                    inner join attendance a
                        on sc.student_seq = a.student_seq
                            inner join student s
                                on a.student_seq = s.student_seq
                                    where c.course_seq = <과정번호> ; --과정번호입력
```

- 특정 근태 상황(정상/지각/조퇴/외출/병가/기타)에 해당하는 모든 교육생 출결 정보를 조회한다.

```
select * from attendance where etc = <근태상황>; -- 병가,외출,조퇴 중 입력
-- 정상인 경우
select
attendance_seq ,
student_seq,
attendance_date,
starttime,
endtime,
case
when to_char(starttime, 'hh24-mi-dd') < to_char('09:10:00') and endtime is not null then '정상'
end as "근태현황"
from attendance
where starttime is not null and to_char(starttime, 'hh24-mi-dd') < to_char('09:10:00') and endtime is not null and etc is not null;
 -- 지각인 경우
select
attendance_seq ,
student_seq,
attendance_date,
starttime,
endtime,
case
when to_char(starttime, 'hh24-mi-dd') > to_char('09:10:00') and endtime is not null then '지각'
end as "근태현황"
from attendance
where starttime is not null and to_char(starttime, 'hh24-mi-dd') > to_char('09:10:00') and endtime is not null and etc is not
```

```
null;
```

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

```
select
attendance_seq ,
student_seq,
attendance_date,
starttime,
endtime,
case
when a.starttime is null and a.endtime is null and a.etc is null then '휴일'
when a.starttime is null and a.endtime is null and a.etc is not null then a.etc
when to_char(starttime, 'hh24-mi-dd') < to_char('09:10:00') and endtime is not null then '정상'
when to_char(starttime, 'hh24-mi-dd') > to_char('09:10:00') and endtime is not null then '지각'
when a.starttime is not null and a.endtime is null then a.etc
end as "근태상황"
from attendance a where attendance_date between <시작날짜> and <끝날짜>
order by attendance_seq;     --시작날짜 끝 날짜 입력
```

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

DML

1. 월별 개설과정 수업계획서 등록

```
insert into courseplan values (courseplan_seq.nextVal, <월>,<계획내용>,<개설과정번호>);
```

2. 월별 개설과정 수업계획서 조회

2.1 개설과정 조회 및 개설과정 번호 확인

```
select
    oc.opencourse_seq as "개설과정번호",
    c.course_name as "과정명"
from course c
    inner join opencourse oc
        on oc.course_seq = c.course_seq
    order by oc.opencourse_seq;
```

2.2 개설과정 수업계획서 조회

```
select
    oc.opencourse_seq as “개설과정번호”,
    c.course_name as “과정명”,
    cp.courseplan_seq as “계획번호”,
    cp.month as “월”,
    cp.content as “계획내용”
from course c
    inner join opencourse oc
        on oc.course_seq = c.course_seq
            inner join courseplan cp
                on cp.opencourse_seq = oc.opencourse_seq
```

```
          where oc.opencourse_seq = <개설과정번호>;
```

3. 수업계획서 수정

3.1 개설과정의 수업계획서 조회 및 계획번호 확인

```
select
     cp.*
from opencourse oc
     inner join courseplan cp
          on cp.opencourse_seq = oc.opencourse_seq
     where cp.opencourse_seq = <개설과정번호>;
```

3.2 개설과정의 수업계획서 수정

```
update courseplan set <수정할 내용> where opencourse_seq = <개설과정번호>;
```

4. 수업계획서 삭제

```
delete from courseplan where courseplan_seq = <계획번호>;
```

요구사항번호	B-10	작성일	2023-04-04
요구사항명	교육생 사후관리(취업현황)	작성자	최재현

DML

교육생 사후관리

1. 취업현황 입력

```
insert into job values (199, '취업', '프론트엔드'); -- 취업한 경우

insert into job values (200, '미취업', 'null'); --미취업한 경우
```

2. 취업현황 수정 (미취업에서 취업으로 넘어갈 경우)

```
update job set state = '취업', field = '백엔드' where student_seq = 200;
```

3. 취업현황 삭제

```
delete from job where student_seq = 199;

delete from job where student_seq = 200;
```

4. 취업 현황 조회

```
--1. 종료된 과정번호, 과정명 검색

SELECT op.opencourse_seq, c.course_name

FROM course c

Inner join opencourse op

on c.course_seq = op.course_seq

where op.enddate < sysdate

order by op.opencourse_seq;
```

```
--2. 과정선택

SELECT sc.opencourse_seq

FROM signupcourse sc

Inner join opencourse op

on sc.opencourse_seq = op.opencourse_seq

Inner join course c

on op.course_seq = c.course_seq

WHERE op.enddate < sysdate and op.opencourse_seq = <과정번호 선택>

GROUP BY sc.opencourse_seq;


--3.선택과정 내 교육생 취업현황(취업률, 분야, 분야, 분야)

SELECT

        (case

        when sc.opencourse_seq = “<선택된 과정번호>”
then round(100 / count(*) *
 (SELECT COUNT(*)
     FROM JOB j
     Inner join student st
     on st.student_seq = j.student_seq
     Inner join signupcourse sc
     on sc.student_seq = st.student_seq
     Inner join opencourse op
     on op.opencourse_seq = sc.opencourse_seq
     where state = '취업'
     group by op.opencourse_seq
     having op.opencourse_seq = “<선택된 과정번호>”))
 end) as "취업률",
(case

when sc.opencourse_seq = “<선택된 과정번호>”   then round(100 / count(*) * (SELECT COUNT(*)

                                     FROM JOB j

                                     Inner join student st

                                     on st.student_seq = j.student_seq

                                     Inner join signupcourse sc

                                     on sc.student_seq = st.student_seq

                                     Inner join opencourse op

                                     on op.opencourse_seq = sc.opencourse_seq

                                     where field = '백엔드'

                                     group by op.opencourse_seq

                                     having op.opencourse_seq = “<선택된 과정번호>”))
```

```
end) as "백엔드",


(case

when sc.opencourse_seq = “<선택된 과정번호>”

                                        then round(100 / count(*) * (SELECT COUNT(*)

                                        FROM JOB j

                                        Inner join student st

                                        on st.student_seq = j.student_seq

                                        Inner join signupcourse sc

                                        on sc.student_seq = st.student_seq

                                        Inner join opencourse op

                                        on op.opencourse_seq = sc.opencourse_seq

                                        where field = '풀스택'
                                        group by op.opencourse_seq
                                        having op.opencourse_seq = “<선택된 과정번호>”))
end) as "풀스택",


(case
when sc.opencourse_seq = “<선택된 과정번호>”   then round(100 / count(*) * (SELECT COUNT(*)
                                        FROM JOB j
                                        Inner join student st
                                        on st.student_seq = j.student_seq
                                        Inner join signupcourse sc
                                        on sc.student_seq = st.student_seq
                                        Inner join opencourse op
                                        on op.opencourse_seq = sc.opencourse_seq
                                        where field = '프론트엔드'
                                        group by op.opencourse_seq
                                        having op.opencourse_seq = “<선택된 과정번호>”))
      end) as "프론트엔드"
FROM signupcourse sc
Inner join opencourse op
on sc.opencourse_seq = op.opencourse_seq
Inner join course c
on op.course_seq = c.course_seq
WHERE op.enddate < sysdate and op.opencourse_seq = “<선택된 과정번호>”
GROUP BY sc.opencourse_seq;GROUP BY sc.opencourse_seq;
```

요구사항번호	C-01	작성일	2023-04-04
요구사항명	교사 강의스케줄 조회	작성자	조아현

DML

1. 강의 스케줄 정보 출력

1.1 교사 정보 조회 및 교사번호 확인

```
select * from teacher;
```

1.2 강의 상태별 강의 스케줄 조회

1.2.1 강의예정 스케줄 조회

```
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 = <교사번호> and os.startdate > sysdate;
```

1.2.2 강의중 스케줄 조회

```
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 = <교사번호> and os.startdate < sysdate and os.enddate > sysdate ;
```

1.2.3 강의종료 스케줄 조회

```
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 = <교사번호> and os.enddate < sysdate ;
```

2. 강의중인 개설과목의 개설과정 내 교육생 정보 확인

2.1 본인 교사번호 확인, 개설과정 번호 확인

2.1.1 교사번호 확인

```
select * from teacher;
```

2.1.2 본인이 강의중인 개설과목의 개설과정 번호 확인

```
select
     oc.opencourse_seq,
     s.subject_name
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 subject s on s.subject_seq = os.subject_seq
```

```
    inner join course c on c.course_seq = oc.course_seq
        where t.teacher_seq = 1 and os.startdate < sysdate and os.enddate > sysdate;
```

2.2 강의중인 개설과목의 개설과정 내 교육생 정보 확인

```
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 = <교사번호> and oc.opencourse_seq = <개설과정번호> and os.startdate < sysdate and os.enddate > sysdate;
```

요구사항번호	C-02	작성일	2023-04-04
요구사항명	배점 입출력	작성자	조아현

DML

1. 배점 정보 등록

1.1 교사 번호 확인

```
select * from teacehr;
```

1.2 해당 교사의 강의 종료 개설과목 확인

```
select * from opensubject where teacher_seq = <교사 번호> and enddate < sysdate;
```

1.3 배점 테이블에서 해당하는 개설과목 번호의 배점 정보 등록

```
insert into gradepercent values (<개설과목번호>, <출결배점>,<필기배점>,<실기배점>);
```

2. 배점 확인 과목 목록 출력

```
select
        s.subject_seq as "과목번호",
        c.course_name as "과정명",
        oc.startdate as "과정기간",
        oc.enddate as "과정기간",
        oc.classroom_seq as "강의실",
        s.subject_name as"과목명",
        os.startdate as "과목기간",
        os.enddate as "과목기간",
        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 = <교사 본인 번호> ;
```

3. 배점 수정

3.1 강의종료된 개설과목 배점 확인

```
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 = <교사번호> and os.enddate < sysdate;
```

3.2 해당 개설과목의 배점 수정

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

4. 배점 삭제

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

요구사항번호	C-03	작성일	2023.04.05
요구사항명	성적 입출력	작성자	이민지
DML			

```
-- 1. 성적 입출력

-- 1-1. 종료 과목 한정 과목 목록 출력

select distinct

    os.opensubject_seq as "과목번호",

    c.course_name as "과정명",

    oc.startDate as "과정 시작일",

    oc.endDate as "과정 종료일",

    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",

    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; --59


--교사 선택을 위한 view 만들기
drop view vwEndSubject;
create or replace view vwEndSubject
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; --59


-- 1.2 '조성아' 선생님의 종료 과목 중
```

```
select * from vwEndSubject where "담당 선생님" = '조성아';
-- 1.2 , 특정 과목 선택시(2), 교육생 정보 출력
select
    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.ascore as "출결 점수",
    sco.wscore as "필기 점수",
    sco.pscore as "실기 점수"
    from vwEndSubject 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
where "담당 선생님" = '조성아' and "과목번호" = 2;


-- 1.3 특정 교육생 선택시 (교육생 이름 : 박린희), 시험 점수 입력 가능하도록
update score set ascore=88, wscore=99, pscore=81   where openSubject_seq =
        (select
             distinct ve."과목번호" as "개설과목번호"
        from vwEndSubject 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

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

    student_seq = (select

            s.student_seq as "교육생번호"

        from vwEndSubject 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

        where "담당 선생님" = '조성아' and "과목번호" = 2 and s.name = '박린희');
```

요구사항번호	C-04	작성일	2023.04.05
요구사항명	출결 관리 및 출결 조회	작성자	이민지

DML

```
-- 2.1 '조성아' 교사가 강의한 과정 출력

select

	distinct oc.opencourse_seq as "과정 번호",

	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
```

```
where t.name = '조성아'

order by oc.opencourse_seq;



--2.2 '조성아' 교사가 강의한 과정의 모든 교육생의 출결 조회 (전체 과정 출결)

select

distinct sc.openCourse_seq as "과정번호",

      c.course_name as "과정명",

      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

where t.name = '조성아' order by openCourse_seq;



-- 2.3 '조성아' 교사가 강의한 1 과정(반응형 UI/UX 웹콘텐츠 개발자 양성과정) 의 모든 교육생의 출결 조회(과정별 조회)

select

distinct sc.openCourse_seq as "과정번호",

      c.course_name as "과정명",
    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
```

```
where t.name = '조성아' and sc.openCourse_seq = 1

order by "학생번호", "날짜";




-- 2.4 '조성아' 교사가 강의한 과정에서 근태상황으로 출결 조회(인원별 조회)

-- 1) 정상

select

distinct sc.openCourse_seq as "과정번호",

       c.course_name as "과정명",

       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",
        '정상' as "근태상황"
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
where t.name = '조성아' and (att.etc is null or att.etc = 'null')
order by "학생번호", "날짜";

-- 2) 조퇴
```

```
select

distinct sc.openCourse_seq as "과정번호",

      c.course_name as "과정명",

      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 "근태상황"

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
where t.name = '조성아' and att.etc = '조퇴'
order by "학생번호", "날짜";


-- 3) 외출
select
distinct sc.openCourse_seq as "과정번호",
        c.course_name as "과정명",
        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 "근태상황"
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
where t.name = '조성아' and att.etc = '외출'
order by "학생번호", "날짜";
```

```
-- 4)  병가
select
distinct sc.openCourse_seq as "과정번호",
    c.course_name as "과정명",
    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 "근태상황"
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
where t.name = '조성아' and att.etc = '병가'
order by "학생번호", "날짜";



-- 5)  지각
select
distinct sc.openCourse_seq as "과정번호",
       c.course_name as "과정명",
       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",

    '지각' as "근태 상황"

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

where t.name = '조성아' and TO_CHAR(att.starttime, 'hh24:mi:ss') > to_char('09:10:00')
```

```
order by "학생번호", "날짜";







-- 2.5 ' 조성아' 교사가 강의한 과정에서 기간별로 출결 조회(기간별 조회) ex) 2022-04-01 ~ 2022-04-10

select

distinct sc.openCourse_seq as "과정번호",

        c.course_name as "과정명",

        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
where t.name = '조성아' and att.attendance_date between '2022-04-01' and '2022-04-10'
order by "학생번호", "날짜";
```

요구사항번호	C-05	작성일	2023-04-04

요구사항명	Q&A 게시판	작성자	조아현

DML

1. 질문 목록 조회

```
select

     q.question_seq as "질문번호",

     q.content as "질문",

     st.name as "학생명",

     q.question_date as "날짜"

from student st

     inner join question q on q.student_seq = st.student_seq;
```

2. 답변 등록

```
insert into answer values (answer_seq.nextVal, <답변내용>,<답변날짜>,<교사번호>,<질문번호>
```

3. 답변 수정

```
update answer set <수정할 내용> where answer_seq = <답변번호>;
```

4. 답변 삭제

```
delete from answer where answer_seq = <답변번호>;
```

요구사항번호	D-02. 교육생	작성일	2023-04-05
요구사항명	학생 출결 관리 및 출결 조회	작성자	최재현

DML

```
--출석 시간입력

INSERT INTO STUDENT(starttime) VALUES(to_char('09:00:00', 'hh24:mi:ss');



--퇴근 시간입력
```

```
INSERT INTO STUDENT(endtime) VALUES(to_char(‘17:50:00’, ‘hh24:mi:ss’);



--학생 본인 전체 출결 조회

SELECT student_seq as "학생번호", attendance_date as "날짜", to_char(starttime, 'hh24:mi:ss') as "출근시간", to_char(endtime,
'hh24:mi:ss') as "퇴근시간", etc,

    (case

        when starttime is null and endtime is null and etc is null then '휴일'

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

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

        else etc

    end) as "분류"

FROM attendance WHERE student_seq = <학생번호> order by attendance_date;

--학생 본인 한달 출결 조회

SELECT student_seq as "학생번호", attendance_date as "날짜", to_char(starttime, 'hh24:mi:ss') as "출근시간, to_char(endtime,
'hh24:mi:ss') as "퇴근시간", etc,
    (case
        when starttime is null and endtime is null and etc is null then '휴일'
        when to_char(starttime, 'hh24-mi-dd') < to_char('09:10:00') and endtime is not null and etc = 'null' then '정상'
        when to_char(starttime, 'hh24-mi-dd') > to_char('09:10:00') and endtime is not null and etc = 'null' then '지각'
        else etc
     end) as "분류"
FROM attendance WHERE student_seq = <학생본인번호> and attendance_date >= “<언제부터>” and attendance_date <= “<언제까지>” order by attendance_date;


--학생 본인 하루 출결 조회
SELECT student_seq as "학생번호", attendance_date as "날짜", to_char(starttime, 'hh24:mi:ss') as "출근시간", to_char(endtime, 'hh24:mi:ss') as "퇴근시간", etc,
    (case
        when starttime is null and endtime is null and etc is null then '휴일'

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

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

        else etc

    end) as "분류"

FROM attendance WHERE student_seq = “<학생번호>” and attendance_date = “<조회할 날짜>”
```

요구사항번호	D-01. 교육생	작성일	2023-04-05
요구사항명	성적 조회	작성자	최재현
DML			

```
--1번 학생이 수강중인 과정 기간, 과정명, 강의실 번호
SELECT distinct oc.startdate as "시작 날짜", oc.enddate as "종료 날짜", course_name as "과정명", cr.classRoom_seq as "강의실"
FROM opencourse oc
Inner join signupCourse sc
on oc.opencourse_seq = sc.opencourse_seq
Inner join course c
on oc.course_seq = c.course_seq
Inner join classRoom cr
on oc.classRoom_seq = cr.classRoom_seq
Inner join opensubject os
on oc.opencourse_seq = os.opencourse_seq
Inner join student s
on sc.student_seq = s.student_seq
```

```
where s.student_seq = '<학생 본인 번호>';



--1번학생의 성적조회
SELECT distinct su.subject_name as "과목", t.test_date as "시험일자", s.ascore as "출결점수", s.wscore as "필기점수", s.pscore as "실기점수"
FROM opencourse op
Inner join opensubject os
on op.opencourse_seq = os.opencourse_seq
Inner join score s
on s.opensubject_seq = os.opensubject_seq
Inner join subject su
on su.subject_seq = os.subject_seq
```

```
Inner join test t
on t.opensubject_seq = os.opensubject_seq
where s.student_seq = '<학생 본인 번호>';



--문제까지 같이 조회
SELECT distinct su.subject_name as "과목", t.test_date as "시험일자", s.ascore as "출결점수", s.wscore as "필기점수", s.pscore as "실기점수" , tq.content
FROM opencourse op
Inner join opensubject os
on op.opencourse_seq = os.opencourse_seq
Inner join score s
on s.opensubject_seq = os.opensubject_seq
```

```
Inner join subject su

on su.subject_seq = os.subject_seq

Inner join test t

on t.opensubject_seq = os.opensubject_seq

Inner join testquiz tq

on t.test_seq = tq.test_seq

where s.student_seq = '<학생 본인 번호>'

order by t.test_date;
```

요구사항번호	D-03 교육생	작성일	2023-04-04
요구사항명	수업계획서 조회	작성자	조아현

DML

1. 개설과정 수업계획서 조회

```
select
    oc.opencourse_seq as “개설과정번호”,
    c.course_name as “과정명”,
    cp.courseplan_seq as “계획번호”,
    cp.month as “월”,
    cp.content as “계획내용”
from course c
    inner join opencourse oc
        on oc.course_seq = c.course_seq
            inner join courseplan cp
                on cp.opencourse_seq = oc.opencourse_seq
        where oc.opencourse_seq = <개설과정번호>;
```

요구사항번호	D-04	작성일	2023-04-04
요구사항명	멘토 등록 및 커리큘럼 관리	작성자	조아현

DML

1. 전공자(멘토) 기능

1.1 커리큘럼 등록

```
insert into curriculum values (curriculum_seq.nextVal, <커리큘럼제목>,<내용>,<정원>);
```

1.2 커리큘럼 조회

```
select
      c.curriculum_seq as "커리큘럼 번호",
	s.student_seq as "교육생번호",
	s.name as "멘토명",
	c.title as "제목",
	c.capacity as "정원",
	c.content as "내용"
from mento m
	inner join curriculum c on c.curriculum_seq = m.curriculum_seq
	inner join student s on s.student_seq = m.student_seq;
```

1.3 커리큘럼 수정

```
update curriculum set <수정할내용> where curriculum_seq = <커리큘럼번호>;
```

1.4 커리큘럼 삭제

```
delete from curriculum where curriculum_seq = <커리큘럼번호>;
```

요구사항번호	D-05	작성일	2023-04-04
요구사항명	멘티 등록 및 커리큘럼 관리	작성자	조아현

DML

```
--멘티 등록 > 초기에 커리큘럼 번호는 NULL 로 설정.

INSERT INTO MENTI(<학생번호>, <커리큘럼번호>);



--듣고 있는 커리큘럼 수정

UPDATE MENTI SET CURRICULUM_SEQ = <커리큘럼 번호> WHERE CURRICULUM_SEQ = <커리큘럼 번호>;



--멘티 탈퇴

DELETE FROM MENTI WHERE STUDENT_SEQ = “<학생 번호>”;



--커리큘럼 조회

SELECT * FROM CURRICULUM;
```

요구사항번호	D-06 교육생	작성일	2023-04-05
요구사항명	취업분야 및 취업률 조회	작성자	최재현

DML

```
--1. 종료된 과정번호, 과정명 검색

SELECT op.opencourse_seq, c.course_name

FROM course c

Inner join opencourse op

on c.course_seq = op.course_seq

where op.enddate < sysdate

order by op.opencourse_seq;
```

```
--2. 과정선택
SELECT sc.opencourse_seq
FROM signupcourse sc
Inner join opencourse op
on sc.opencourse_seq = op.opencourse_seq
Inner join course c
on op.course_seq = c.course_seq
WHERE op.enddate < sysdate and op.opencourse_seq = <과정번호 선택>
GROUP BY sc.opencourse_seq;


--3.선택과정 내 교육생 취업현황(취업률, 분야, 분야, 분야)
SELECT
```

```
 (case

 when sc.opencourse_seq = “<선택된 과정번호>“

then round(100 / count(*) *

 (SELECT COUNT(*)

     FROM JOB j

     Inner join student st

     on st.student_seq = j.student_seq

     Inner join signupcourse sc

     on sc.student_seq = st.student_seq

     Inner join opencourse op

     on op.opencourse_seq = sc.opencourse_seq

     where state = '취업'

     group by op.opencourse_seq
```

```
          having op.opencourse_seq = “<선택된 과정번호>”))

      end) as "취업률",


      (case

      when sc.opencourse_seq = “<선택된 과정번호>”   then round(100 / count(*) * (SELECT COUNT(*)

                                                   FROM JOB j

                                                   Inner join student st

                                                   on st.student_seq = j.student_seq

                                                   Inner join signupcourse sc

                                                   on sc.student_seq = st.student_seq

                                                   Inner join opencourse op

                                                   on op.opencourse_seq = sc.opencourse_seq

                                                   where field = '백엔드'
```

```
                              group by op.opencourse_seq
                              having op.opencourse_seq = “<선택된 과정번호>”))
end) as "백엔드",

(case
when sc.opencourse_seq = “<선택된 과정번호>”
                              then round(100 / count(*) * (SELECT COUNT(*)
                              FROM JOB j
                              Inner join student st
                              on st.student_seq = j.student_seq
                              Inner join signupcourse sc
                              on sc.student_seq = st.student_seq
                              Inner join opencourse op
```

```
                                        on op.opencourse_seq = sc.opencourse_seq
                                        where field = '풀스택'
                                        group by op.opencourse_seq
                                        having op.opencourse_seq = “<선택된 과정번호>”))
end) as "풀스택",

(case
when sc.opencourse_seq = “<선택된 과정번호>”   then round(100 / count(*) * (SELECT COUNT(*)
                                        FROM JOB j
                                        Inner join student st
                                        on st.student_seq = j.student_seq
                                        Inner join signupcourse sc
                                        on sc.student_seq = st.student_seq
```

```
                                                  Inner join opencourse op

                                                  on op.opencourse_seq = sc.opencourse_seq

                                                  where field = '프론트엔드'

                                                  group by op.opencourse_seq

                                                  having op.opencourse_seq = “<선택된 과정번호>”))

        end) as "프론트엔드"

FROM signupcourse sc

Inner join opencourse op

on sc.opencourse_seq = op.opencourse_seq

Inner join course c

on op.course_seq = c.course_seq

WHERE op.enddate < sysdate and op.opencourse_seq = “<선택된 과정번호>”

GROUP BY sc.opencourse_seq;GROUP BY sc.opencourse_seq;
```

요구사항번호	D-07 교육생	작성일	2023-04-04
요구사항명	Q&A 게시판	작성자	조아현

DML

1. 질문 답변 목록 조회

```
select
        q.question_seq as "질문번호",
        q.content as "질문",
        a.content as "답변",
        st.name as "학생명",
        t.name as "교사명",
        q.question_date as "날짜"
from student st
        inner join question q on q.student_seq = st.student_seq
        inner join answer a on a.question_seq = q.question_seq
        inner join teacher t on t.teacher_seq = a.teacher_Seq;
```

2. 질문 등록

2.1 학생 번호 확인

```
select * from student where name = <교육생명>;
```

2.2 질문 등록

```
insert into question values (question_seq.nextVal,<질문내용>,<질문날짜>,<교육생번호>;
```

3. 질문 수정

```
update question set <수정할 내용> where question_seq = <질문번호>;
```

요구사항번호	D-08 교육생	작성일	2023-04-04
요구사항명	취뽀생 게시판	작성자	조아현

DML

```
--1. 취뽀생(교육생 중 취업에 성공한 사람)

--1.1 취뽀생 게시글목록 조회

-- 게시글 번호, 게시글제목 을 볼 수 있다.

select

        successjob_seq as "게시글번호",

        title as "게시글제목"

from successjob;
```

```
--1.2 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;



--1.2 취뽀생 게시글 등록

-- 게시글번호, 게시글제목, 게시글내용, 교육생번호를 사용하여 게시글을 등록한다.

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



--1.3 취뽀생 댓글 등록

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

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