

DB Object 스크립트	문서번호	
	작성자	4조
	작성일자	24.03.08

요구사항번호	B-03(관리자)	작성일	2023-03-08
요구사항명	특정 교사 정보 출력	작성자	김희연
DML			
create or replace procedure procTeacherInfo ( pname in varchar2 )			

is

vtName tblTeacher.name%type;

vsName tblSubject.name%type;

vsStart tblopendsubjectlist.startdate%type;

vsEnd tblopendsubjectlist.enddate%type;

vcName tblcurriculum.name%type;

vcstart tblopendcurriculum.startdate%type;

vcend tblopendcurriculum.enddate%type;

vbname tblTextbook.name%type;

vrname tblroom.name%type;

vStat varchar2(15);

cursor vcursor is

select

s\_name as "개설 과목명",

osl\_startdate as 과목시작일,

```
osl_enddate as 과목종료일,
c_name as 과정명,
oc_startdate as 과정시작일,
oc_enddate as 과정종료일,
b.name as 교재명,
r_name as 강의실,
case
    when osl_startdate < sysdate and sysdate > osl_enddate then '강의종료'
    when osl_startdate > sysdate then '강의예정'
    when osl_startdate <= sysdate and sysdate <= osl_enddate then '강의중'
end as "강의상태"
from vwCurriculum v
inner join tblTextbook b
    on b.seq_textbook = v.seq_textbook
inner join tblcurriculumprogress g
    on v.seq_curriculumprogress = g.seq_curriculumprogress
```

```

        inner join tblteacher t

        on v.seq_teacher = t.seq_teacher

        where t.name = pname

        order by 과목시작일;

begin

    open vcursor;

    loop

        fetch vcursor into vsName, vsStart, vsEnd, vcname, vcstart, vchend, vbname, vrname, vStat;

        exit when vcursor%notfound;

        dbms_output.put_line('교사명: ' || pname || ' 과목명: ' || vsName || ' 과목시작일: ' || vsStart || ' 과목종료일: ' || vsEnd || ' 과정명: ' ||
vcname || ' 과정시작일: ' || vcstart || ' 과정종료일: ' || vchend || ' 교재명: ' || vbname || ' 강의실명: ' || vrname || ' 강의상태: ' || vStat);

        dbms_output.put_line('-----');

    end loop;

    close vcursor;

end procTeacherInfo;

/

```

```
begin
```

```
    procTeacherInfo('김희연');
```

```
end;
```

```
/
```

```
create or replace procedure procTeacherWhoCan(
```

```
    pname in varchar2
```

```
)
```

```
is
```

```
    vTname tblteacher.name%type;
```

```
    vSubName tblsubject.name%type;
```

```
    cursor vcursor is
```

```
    select
```

```
        distinct t.name, s.name
```

```
    from tblteacher t
```

```
        inner join tblAvailableSubjectList asl
            on t.seq_teacher = asl.seq_teacher

        inner join tblsubject s
            on s.seq_subject = asl.seq_subject

        where s.name = pname;

begin
    open vcursor;

    loop

        fetch vcursor into vTname, vSubName;

        exit when vcursor%notfound;

        dbms_output.put_line('이름: ' || vTname || ' | 과목명: ' || vSubName);

        dbms_output.put_line('-----');

    end loop;

    close vcursor;

end procTeacherWhoCan;

/
```

```
begin
  procTeacherWhoCan('자바');
end;
/
```

요구사항번호	B-04(관리자)	작성일	2023-03-08
요구사항명	개설 교육 과정 번호를 입력하여 개설 과목 정보 확인	작성자	김희연
DML			
/* 특정 개설 과정 선택 */			

--개설 교육 과정 번호를 입력하여 개설 과목 정보 확인

create or replace procedure procCurriDetail (

    pnum in number

)

is

    vSName tblSubject.name%type;

    vsStart tblOpensubjectlist.startdate%type;

    vsEnd tblOpensubjectlist.enddate%type;

    vtName tbltrainees.name%type;

    vTel varchar2(15);

    vRegdate date;

    vStat varchar2(30);

    cursor vcursor is

    select



```
v.s_name as "과목명",  
v.osl_startdate as "과목시작일",  
v.osl_enddate as "과목종료일",  
t.name as "학생명",  
t.tel as "전화번호",  
t.registrationDate as "등록일",  
(case  
    when tl.status is not null then tl.status  
    when v.seq_curriculumprogress = 1 then '강의 진행 예정'  
    when v.seq_curriculumprogress = 2 then '강의 진행 중'  
end) as "수료 여부"  
from vwcurriculum v  
left outer join tbltraineeelist tl  
    on tl.seq_opencurriculum = v.seq_opencurriculum  
    inner join tbltrainees t  
        on t.seq_trainee = tl.seq_trainee
```

```

        where v.seq_opencurriculum = pnum;

begin
    open vcursor;

    loop

        fetch vcursor into vSName, vsStart, vsEnd, vtName, vTel, vRegdate, vStat;

        exit when vcursor%notfound;

        dbms_output.put_line('과목명: ' || vSName || ' | 과목시작일: ' || vsStart || ' | 과목종료일: ' || vsEnd || ' | 학생명: ' || vtName || ' | 전화번호: '
|| vTel || ' | 등록일: ' || vRegdate || ' | 수료여부: ' || vStat);

        dbms_output.put_line('-----');

    end loop;

    close vcursor;

end procCurriDetail;

/

begin

    procCurriDetail(1); --개설 교육과정 번호

```

end;

/

요구사항번호	B-07(관리자)	작성일	2023-03-08
요구사항명	시험 관리 및 성적 조회	작성자	장원준
DML			
<pre>-- b-7  --과목별  CREATE OR REPLACE PROCEDURE procCurriculumBySubject (</pre>			

```
p_subject_name IN VARCHAR2
)
IS
    v_output VARCHAR2(4000);
BEGIN
    FOR cur IN (
        SELECT vc.c_name || ',' || cp.period || ',' || vc.r_name || ',' || vc.s_name || ',' || tea.name || ',' || tb.name || ',' || vt.t_name || ',' || vt.t_ssn
        || ',' || vg.writtengrade || ',' || vg.practicalgrade AS output_line
        FROM vwcurriculum vc
        INNER JOIN vwtrainees vt ON vt.seq_opencurriculum = vc.seq_opencurriculum
        INNER JOIN vwgrades vg ON vg.seq_traineeelist = vt.seq_traineeelist
        INNER JOIN tblteacher tea ON tea.seq_teacher = vc.seq_teacher
        INNER JOIN tbltextbook tb ON tb.seq_textbook = vc.seq_textbook
        INNER JOIN tblcourseperiod cp ON cp.seq_courseperiod = vc.seq_courseperiod
        WHERE vc.c_name = p_subject_name
    )
    LOOP
```

```
v_output := cur.output_line;

DBMS_OUTPUT.PUT_LINE(v_output);

END LOOP;

END procCurriculumBySubject;

/

/

/

BEGIN

    procCurriculumBySubject ('자바(Java)기반 Web과 스마트폰 Full-Stack 과정(A)');

END;

/

-- 특정 개설 과정

CREATE OR REPLACE PROCEDURE procGradesCourseInfo(
```

```
p_course IN vwcurriculum.c_name%TYPE
) IS
CURSOR c_grades IS
    SELECT vt.t_name, vc.c_name, vc.s_name, t.name AS teacher_name,
           vg.writtengrade, vg.practicalgrade,
           (CASE WHEN vg.writtengrade IS NOT NULL THEN '필기성적입력완료' END) AS written,
           (CASE WHEN vg.practicalgrade IS NOT NULL THEN '실기성적입력완료' END) AS practical
    FROM vwgrades vg
    INNER JOIN vwtrainees vt ON vt.seq_trainee = vg.seq_trainee
    INNER JOIN vwcurriculum vc ON vc.seq_subject = vg.seq_subject
    INNER JOIN tblteacher t ON t.seq_teacher = vc.seq_teacher
    WHERE vc.c_name = p_course
    GROUP BY vt.t_name, vc.c_name, vc.s_name, t.name, vg.writtengrade, vg.practicalgrade;

v_grades c_grades%ROWTYPE;

written VARCHAR2(50);
```

```
practical VARCHAR2(50);  
  
BEGIN  
  
  OPEN c_grades;  
  
  LOOP  
  
    FETCH c_grades INTO v_grades;  
  
    EXIT WHEN c_grades%NOTFOUND;  
  
  
    written := v_grades.written;  
  
    practical := v_grades.practical;  
  
  
    DBMS_OUTPUT.PUT_LINE('과정 이름: ' || v_grades.c_name || ', 주제: ' || v_grades.s_name ||  
      ', 강사 이름: ' || v_grades.teacher_name || ', 필기 성적: ' ||  
      v_grades.writtengrade || ', 실기 성적: ' || v_grades.practicalgrade ||  
      ', 필기성적입력 ' || written || ', 실기성적입력 ' || practical);
```

```
dbms_output.put_line('-----');
dbms_output.put_line('-----');

    END LOOP;

    CLOSE c_grades;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('해당 교육생의 성적 정보를 찾을 수 없습니다.');
```



--교육생 개인 별

```
CREATE OR REPLACE PROCEDURE procGradesInfoByTrainee(  
    p_trainee_name IN VARCHAR2  
) IS  
BEGIN  
    FOR rec IN (  
        SELECT vt.t_name AS trainee_name, vt.t_ssn, vc.c_name AS course_name, cp.period,  
            vc.r_name AS classroom_name, vc.s_name AS subject_name, vc.period AS subject_period,  
            tea.name AS teacher_name, ass.situation, vg.writtengrade, vg.practicalgrade  
        FROM vwgrades vg  
        INNER JOIN vwtrainees vt ON vt.seq_traineeelist = vg.seq_traineeelist  
        INNER JOIN vwcurriculum vc ON vc.seq_subject = vg.seq_subject  
        INNER JOIN tblcourseperiod cp ON cp.seq_courseperiod = vc.seq_courseperiod  
        INNER JOIN tblattendancestatus ass ON ass.seq_attendancestatus = vt.seq_attendancestatus  
        INNER JOIN tblteacher tea ON tea.seq_teacher = vc.seq_teacher  
        WHERE vt.t_name = p_trainee_name
```

```
GROUP BY vt.t_name, vt.t_ssn, vc.c_name, cp.period, vc.oc_startdate, vc.oc_enddate,
        vc.r_name, vc.c_name, vc.period, tea.name, ass.situation, vc.s_name, vg.writtengrade, vg.practicalgrade
)
LOOP
    DBMS_OUTPUT.PUT_LINE('교육생 이름: ' || rec.trainee_name || ', 주민등록번호: ' || rec.t_ssn ||
        ', 과정명: ' || rec.course_name || ', 교사: ' || rec.period ||
        ', 강의실: ' || rec.classroom_name || ', 주제: ' || rec.subject_name ||
        ', 기간: ' || rec.subject_period || ', 교사 이름: ' || rec.teacher_name ||
        ', 출결상태: ' || rec.situation || ', 필기성적: ' ||
        rec.writtengrade || ', 실기성적: ' || rec.practicalgrade);
END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('해당 교육생의 성적 정보를 찾을 수 없습니다.');
```

```
    WHEN OTHERS THEN
        ROLLBACK;
```

```
RAISE;  
  
END procGradesInfoByTrainee;  
  
/  
  
/  
  
BEGIN  
  
    procGradesInfoByTrainee('천유서');  
  
END;  
  
/
```

요구사항번호	B-08(관리자)	작성일	2023-03-08
요구사항명	개설 과정 특정 기간 출결 조회	작성자	장원준
DML			

```
CREATE OR REPLACE PROCEDURE procCourseInfo(  
    p_course IN tblCurriculum.name%TYPE,  
    p_start_date IN DATE,  
    p_end_date IN DATE  
  
) IS  
  
    CURSOR c_course IS  
  
        SELECT t.name AS trainee_name, a.day, c.name AS course_name, ad.situation  
  
        FROM tblAttendance a  
  
        INNER JOIN tblTraineeList tl ON a.seq_traineeList = tl.seq_traineeList  
  
        INNER JOIN tblopencurriculum oc ON oc.seq_opencurriculum = tl.seq_opencurriculum  
  
        INNER JOIN tblCurriculum c ON c.seq_curriculum = oc.seq_opencurriculum  
  
        INNER JOIN tblattendancestatus ad ON ad.seq_attendancestatus = a.seq_attendancestatus  
  
        INNER JOIN tblTrainees t ON t.seq_trainee = tl.seq_trainee  
  
        WHERE c.name = p_course  
  
        AND a.day BETWEEN p_start_date AND p_end_date
```

```
GROUP BY t.name, a.day, c.name, ad.situation;
```

```
v_course c_course%ROWTYPE;
```

```
BEGIN
```

```
OPEN c_course;
```

```
LOOP
```

```
FETCH c_course INTO v_course;
```

```
EXIT WHEN c_course%NOTFOUND;
```

```
DBMS_OUTPUT.PUT_LINE('교육생 이름: ' || v_course.trainee_name || ', 날짜: ' || v_course.day || ', 과정 이름: ' ||  
v_course.course_name || ', 출석 상황: ' || v_course.situation);
```

```
dbms_output.put_line('-----  
-----');
```

```
END LOOP;
```

```
CLOSE c_course;
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN

    DBMS_OUTPUT.PUT_LINE('해당 과정의 정보를 찾을 수 없습니다.');
```

WHEN OTHERS THEN

```
    ROLLBACK;

    RAISE;

END procCourseInfo;

/

BEGIN

    procCourseInfo('AWS 클라우드와 Elasticsearch를 활용한 Java Full-Stack 과정(B)', TO_DATE('2023-11-01', 'YYYY-MM-DD'),
TO_DATE('2023-12-30', 'YYYY-MM-DD'));

END;

/

-- 2. 특정 인원

CREATE OR REPLACE PROCEDURE procTraineeAttendanceInfo(

    p_name IN vwtrainees.t_name%TYPE
```

```
) IS
```

```
CURSOR c_attendance IS
```

```
    SELECT vt.t_name, vt.a_day, vc.c_name, vt.situation
```

```
    FROM vwtrainees vt
```

```
    INNER JOIN vwcurriculum vc ON vt.seq_opencurriculum = vc.seq_opencurriculum
```

```
    WHERE vt.t_name = p_name
```

```
    GROUP BY vt.t_name, vt.a_day, vc.c_name, vt.situation;
```

```
v_attendance c_attendance%ROWTYPE;
```

```
BEGIN
```

```
    OPEN c_attendance;
```

```
    LOOP
```

```
        FETCH c_attendance INTO v_attendance;
```

```
        EXIT WHEN c_attendance%NOTFOUND;
```

```
        DBMS_OUTPUT.PUT_LINE('교육생 이름: ' || v_attendance.t_name || ', 날짜: ' || v_attendance.a_day || ', 과정 이름: ' ||  
v_attendance.c_name || ', 상태: ' || v_attendance.situation);
```

```
dbms_output.put_line('-----  
-----');
```

```
END LOOP;

CLOSE c_attendance;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('해당 이름을 가진 교육생의 출석 정보를 찾을 수 없습니다.');
```

WHEN OTHERS THEN

```
        ROLLBACK;

        RAISE;

END procTraineeAttendanceInfo;

/

BEGIN

    procTraineeAttendanceInfo('제류혁');

END;

/
```



--

요구사항번호	B-13(관리자)	작성일	2023-03-08
요구사항명	출결 인정 서류 관리	작성자	정원혁
DML			
<pre>create or replace procedure procInsertattendancePapers (     p_seq_attendancePapers tblAttendancePapers.seq_attendancePapers%type,     p_document tblAttendancePapers.document%type ) is     vdocument varchar2(50); begin</pre>			

```
vdocument := p_document;

if (vdocument like '%코로나%' or
    vdocument like '%사망%' or
    vdocument like '%입원%' or
    vdocument like '%의사%' or
    vdocument like '%출생 신고서%' or
    vdocument like '%국가 자격증 시험%' or
    vdocument like '%예비군%' or
    vdocument like '%면접%' or
    vdocument like '%국가%' or
    vdocument like '%병가%') then

    update tblAttendancePapers

    set admitattendance = '출석 인정'

    where seq_attendancePapers = p_seq_attendancePapers;

    dbms_output.put_line('출석이 인정되었습니다.');
```

```
else

    update tblAttendancePapers

    set admitattendance = '출석 미인정'

    where seq_attendancePapers = p_seq_attendancePapers;

    dbms_output.put_line('출석으로 인정되지 않는 서류입니다.');
```

end if;

```
end procInsertattendancePapers;

/

begin

    procInsertattendancePapers(1, '경찰서');
```

end;

```
/
```

요구사항번호	B-14(관리자)	작성일	2023-03-08
요구사항명	기자재 & 사물함 관리	작성자	정원혁
DML			
<pre> create or replace procedure procManagelocker (     p_status tblTraineeList.status%type,     p_seq_locker tblLocker.seq_locker%type ) is begin     if p_status = '수료' then         update tblLocker         set seq_traineeList = null         where seq_locker = p_seq_locker;          dbms_output.put_line('교육생이 수료했습니다. 사물함이 비었습니다.');</pre>			
end if;			

```
end procManagelocker;
```

```
/
```

```
declare
```

```
    v_seq_locker tblLocker.seq_locker%type;
```

```
begin
```

```
    v_seq_locker := 111;
```

```
    procManagelocker('수료', v_seq_locker);
```

```
end;
```

```
/
```

```
-- 교육생 목록에 리스트가 들어갔을 때 사물함이 자동 배정
```

```
create or replace procedure procAutoInsertlocker (
```

```
    p_seq_traineeList tblTraineeList.seq_traineeList%type,
```

```
    p_seq_trainee tblTraineeList.seq_trainee%type,
```

```
    p_seq_openCurriculum tblTraineeList.seq_openCurriculum%type
```

```
)
```

is

v\_seq\_locker tblLocker.seq\_locker%type;

begin

select seq\_locker into v\_seq\_locker

from tblLocker

where seq\_traineeList is null

and seq\_locker between 21 and 178

and rownum = 1;

update tblLocker

set seq\_traineeList = p\_seq\_traineeList

where seq\_locker = v\_seq\_locker;

-- 배정 결과를 출력합니다.

dbms\_output.put\_line('교육생이 사물함에 자동으로 배정되었습니다.');

```
exception
```

```
    when no_data_found then
```

```
        dbms_output.put_line('비어있는 사물함이 없습니다.');
```

```
    when others then
```

```
        dbms_output.put_line('오류가 발생했습니다.');
```

```
end procAutoInsertlocker;
```

```
/
```

```
begin
```

```
    procAutoInsertlocker(71, 51, 3);
```

```
end;
```

```
/
```

요구사항번호	B-16(관리자)	작성일	2023-03-08
요구사항명	취업현황 조회 및 관리	작성자	정원혁
DML			
<pre> create or replace procedure procSelectemploymentStatus (     p_curriculum in number ) is     p_seq_employmentStatus tblEmploymentStatus.seq_employmentStatus%type;     p_seq_traineeList tblTraineeList.seq_traineeList%type;     p_seq_openCurriculum tblOpenCurriculum.seq_openCurriculum%type;     p_name tblCurriculum.name%type;     p_estatus tblEmploymentStatus.status%type;     p_city tblEmploymentStatus.city%type;     p_field tblEmploymentStatus.field%type;     p_salary tblEmploymentStatus.salary%type; </pre>			



```
cursor vcursor is
select
es.seq_employmentStatus as "번호",
tl.seq_traineeList as "교육생 목록 번호",
oc.seq_openCurriculum as "개설 교육과정 번호",
c.name as "교육 과정",
es.status as "취업 여부",
es.city as "지역",
es.field as "분야",
es.salary as "연봉"
from tblEmploymentStatus es
    left outer join tblTraineeList tl
        on tl.seq_traineeList = es.seq_traineeList
        inner join tblOpenCurriculum oc
            on oc.seq_openCurriculum = tl.seq_openCurriculum
            inner join tblCurriculum c
```

```

        on oc.seq_curriculum = c.seq_curriculum

        where oc.seq_openCurriculum = p_curriculum

        order by es.seq_employmentStatus;

begin
open vcursor;

loop
fetch vcursor into p_seq_employmentStatus, p_seq_traineeList, p_seq_openCurriculum, p_name, p_estatus, p_city, p_field, p_salary;
exit when vcursor%notfound;

dbms_output.put_line('번호: ' || p_seq_employmentStatus || ' | 교육생 목록 번호: ' || p_seq_traineeList || ' | 교육 과정: ' ||
p_seq_openCurriculum || ' | 취업 여부: ' || p_estatus || ' | 지역: ' || p_city || ' | 분야: ' || p_field || ' | 연봉: ' || p_salary);

dbms_output.put_line('-----');
-----');

end loop;

close vcursor;

end procSelectemploymentStatus;

/

```

```
begin
```

```
    procSelectemploymentStatus(1);
```

```
end;
```

```
/
```

요구사항번호	C-1(교사)	작성일	2023-03-08
요구사항명	교사의 강의 스케줄 조회	작성자	김민곤
DML			
CREATE OR REPLACE PROCEDURE proctschedule( pname VARCHAR2 ) IS vc_name VARCHAR2(200);			

```
vcp_status varchar2(30);
```

```
voc_startdate date;
```

```
voc_enddate date;
```

```
vs_seq_subject NUMBER;
```

```
vs_name varchar2(200);
```

```
vos_startdate DATE;
```

```
vos_enddate DATE;
```

```
vtb_name VARCHAR2(200);
```

```
vtl_seq_opencurriculum number;
```

```
CURSOR vcursor IS
```

```
    select distinct
```

```
c.name as 과정명,
```

```
cp.status as 과정상태,
```

```
oc.startDate as 시작일,
```

```
oc.endDate as 종료일,
```

```
s.seq_subject as 과목번호,  
s.name as 과목명,  
os.startdate as 과목시작일,  
os.enddate as 과목종료일,  
tb.name as 교재명,  
count(tl.seq_opencurriculum) as "교육생 등록 인원"  
from tblTeacher t  
    inner join tblOpenSubjectList os  
        on t.seq_teacher = os.seq_teacher  
        inner join tblOpenCurriculum oc  
            on oc.seq_opencurriculum = os.seq_opencurriculum  
            inner join tblCurriculumProgress cp  
                on cp.seq_curriculumProgress = oc.seq_curriculumProgress  
                inner join tblTraineeList tl  
                    on oc.seq_opencurriculum = tl.seq_opencurriculum  
                    inner join tblTrainees tr
```

```
on tr.seq_trainee = tl.seq_trainee

inner join tblsubjectlist sl

on sl.seq_subjectList = os.seq_subjectList

inner join tblsubject s

on s.seq_subject = sl.seq_subject

inner join tblRoom r

on r.seq_room = oc.seq_room

inner join tblCurriculum c

on c.seq_Curriculum = oc.seq_Curriculum

inner join tblCoursePeriod cop

on cop.seq_coursePeriod = c.seq_coursePeriod

inner join tblTextBook tb

on tb.seq_textbook = os.seq_textbook

where t.name = pname

group by c.name, cp.status, oc.startDate,oc.endDate,s.seq_subject,s.name,

os.startdate,os.enddate,tb.name;
```

```
BEGIN
```

```
OPEN vcursor;
```

```
LOOP
```

```
    FETCH vcursor INTO vc_name,vcp_status, voc_startdate,voc_enddate,vs_seq_subject, vs_name, vos_startdate, vos_enddate,  
    vtb_name, vtl_seq_opencurriculum;
```

```
    EXIT WHEN vcursor%NOTFOUND;
```

```
    DBMS_OUTPUT.PUT_LINE( '과정명: ' || vc_name || ' | 과정상태' || vcp_status|| ' | 과정시작일: ' || voc_startdate || ' | 과정종료일: ' ||  
    voc_enddate ||
```

```
                                ' | 과목번호'||vs_seq_subject|| ' | 과목명: ' ||vs_name ||' | 과목시작일: ' || vos_startdate || ' | 과목종료일: ' ||  
    vos_enddate ||
```

```
                                ' | 교재명'|| vtb_name||' | 등록인원' || vtl_seq_opencurriculum);
```

```
    dbms_output.put_line  
    ('-----');
```

```
END LOOP;
```

```
CLOSE vcursor;
```

```
END proctschedule;
```

```
/

begin
proctschedule('김민곤');
end;

/

--c-1.2교육중인 과정에 등록된 교육생 정보(교육생 이름, 전화번호, 등록일, 수료 또는 중도탈락)을 확인할 수 있어야 한다.
CREATE OR REPLACE PROCEDURE proctinfo(
    pname VARCHAR2
)
IS
    vc_name varchar2(200);
    vtr_name varchar2(10);
    vtr_tel varchar2(15);
    vtr_registrationDate date;
    vtl_status varchar2(15);
```



```
CURSOR vcursor IS
```

```
select distinct
```

```
c.name as 과정명,
```

```
tr.name as 교육생이름,
```

```
tr.tel as 교육생전화번호,
```

```
tr.registrationDate as 교육생등록일,
```

```
tl.status as 교육생상태
```

```
from tblTeacher t
```

```
    inner join tblOpenSubjectList os
```

```
        on t.seq_teacher = os.seq_teacher
```

```
        inner join tblOpenCurriculum oc
```

```
            on oc.seq_openCurriculum = os.seq_openCurriculum
```

```
            inner join tblCurriculumProgress cp
```

```
                on cp.seq_curriculumProgress = oc.seq_curriculumProgress
```

```
                inner join tblTraineeList tl
```

```
on oc.seq_openCurriculum = tl.seq_openCurriculum

inner join tblTrainees tr

    on tr.seq_trainee = tl.seq_trainee

    inner join tblsubjectlist sl

        on sl.seq_subjectList = os.seq_subjectList

        inner join tblsubject s

            on s.seq_subject = sl.seq_subject

            inner join tblRoom r

                on r.seq_room = oc.seq_room

                inner join tblCurriculum c

                    on c.seq_Curriculum = oc.seq_Curriculum

                    inner join tblCoursePeriod cp

                        on cp.seq_coursePeriod = c.seq_coursePeriod

                        inner join tblTextBook tb

                            on tb.seq_textbook = os.seq_textbook

                            where t.name = ptname;
```

```
BEGIN
```

```
    OPEN vcursor;
```

```
    LOOP
```

```
        FETCH vcursor INTO  vc_name, vtr_name,vtr_tel, vtr_registrationDate, vtl_status;
```

```
        EXIT WHEN vcursor%NOTFOUND;
```

```
        DBMS_OUTPUT.PUT_LINE( '과정명: ' || vc_name || ' | 교육생이름: ' || vtr_name|| ' | 전화번호: ' || vtr_tel || ' | 등록일: ' ||  
vtr_registrationDate ||
```

```
        ' | 상태: ' || vtl_status);
```

```
        dbms_output.put_line  
( '-----');
```

```
    END LOOP;
```

```
    CLOSE vcursor;
```

```
END proctinfo;
```

```
/
```

```
begin
```

```
proctinfo('김민곤');
```

```
end;
```

```
/
```

요구사항번호	C-2(교사)	작성일	2023-03-08
요구사항명	배점 입출력 및 시험 문제,날짜 등록	작성자	정혜정
DML			
create or replace procedure procTestInfoIn( pSeq_teacher in number, pSeq_testInfo in number, pSeq_openSubjectList in number, pWrittenpoints in number,			

```
pPracticalpoints in number,  
pAttendancepoints in number  
)  
is  
    vEnddate date;  
begin  
    -- tblopensubjectlist 테이블의 enddate 확인  
    select osl.enddate into vEnddate  
    from tblOpenSubjectList osl  
        inner join tblTestInfo ti  
            on ti.seq_openSubjectList = osl.seq_openSubjectList  
            where osl.seq_teacher = pSeq_teacher  
            and osl.seq_OpenSubjectList = pSeq_openSubjectList;  
  
    -- enddate가 오늘 이전인 경우에만 배점 입력  
    if vEnddate < sysdate then
```

```
-- 배점 입력

insert into tblTestInfo (seq_testInfo, seq_openSubjectList, writtenpoints, practicalpoints, attendancepoints)

values (pSeq_testInfo, pSeq_openSubjectList, pWrittenpoints, pPracticalpoints, pAttendancepoints);


if sql%rowcount = 0 then

    dbms_output.put_line('해당 시험 정보가 존재하지 않습니다.');
```

else

```
    dbms_output.put_line('배점이 등록되었습니다.');
```

end if;

else

```
    dbms_output.put_line('해당 과목은 아직 종료되지 않았습니다.');
```

end if;

```
end procTestInfoIn;
```

/

begin

```
    procTestInfoIn(1, 1, 1, 40, 40, 20);
```

```
end;
```

```
/
```

```
-- 1.2.배점 출력
```

```
create or replace procedure procTestInfoOut(
```

```
    pSeq_teacher in number,
```

```
    pSeq_openSubjectList in number
```

```
)
```

```
is
```

```
    cursor vcursor is
```

```
        select
```

```
            ti.seq_openSubjectList,
```

```
            ti.writtenpoints,
```

```
            ti.practicalpoints,
```

```
ti.attendancepoints
```

```
from tblTestInfo ti
```

```
inner join tblOpenSubjectList osl
```

```
on ti.seq_openSubjectList = osl.seq_openSubjectList
```

```
where osl.seq_teacher = pSeq_teacher
```

```
and osl.seq_OpenSubjectList = pSeq_openSubjectList
```

```
and osl.enddate < sysdate;
```

```
vSeq_opensubjectlist tblopensubjectlist.seq_opensubjectlist%type;
```

```
vWrittenpoints tbltestinfo.writtenpoints%type;
```

```
vPracticalpoint tbltestinfo.practicalpoints%type;
```

```
vAttendancepoints tbltestinfo.attendancepoints%type;
```

```
begin
```

```
-- 배점 출력
```

```
open vcursor;
```



```
loop

    fetch vcursor into vSeq_opensubjectlist, vWrittenpoints, vPracticalpoint, vAttendancepoints;

    exit when vcursor%notfound;

    -- 결과 출력

    dbms_output.put_line('개설 과목 목록 번호: ' || vSeq_opensubjectlist);

    dbms_output.put_line('필기 배점: ' || vWrittenpoints);

    dbms_output.put_line('실기 배점: ' || vPracticalpoint);

    dbms_output.put_line('출결 배점: ' || vAttendancepoints);

    dbms_output.put_line('-----');

end loop;

close vcursor;

end procTestInfoOut;

/
```

```
begin
    procTestInfoOut(1, 1);
end;
/
```

-- 2. 교사는 자신이 강의를 마친 과목의 목록 중에서 특정 과목을 선택하고 해당 과목의 배점 정보를 출결, 필기, 실기로 구분해서 등록할 수 있어야 한다.(위에서 구현 완료) 시험 날짜, 시험 문제를 추가할 수 있어야 한다.

-- 2.5. 시험 날짜 추가

```
create or replace procedure procTestDateIn (
    pSeq_teacher in number,
    pSeq_testInfo in number,
    pWrittenDate in date,
    pPracticalDate in date
)
is
    vEnddate date;
```

```
begin
  -- tblopendsubjectlist 테이블의 enddate 확인
  select osl.enddate into vEnddate
  from tblOpenSubjectList osl
    inner join tblTestInfo ti
      on ti.seq_openSubjectList = osl.seq_openSubjectList
      where osl.seq_teacher = pSeq_teacher
      and ti.seq_testinfo = pSeq_testInfo;

  -- enddate가 오늘 이전인 경우에만 시험 날짜 추가
  if vEnddate < sysdate then
    -- 시험 날짜 추가
    update tblTestInfo
      set writtenDate = TO_DATE(pWrittenDate, 'YYYY-MM-DD'),
        practicalDate = TO_DATE(pPracticalDate, 'YYYY-MM-DD')
      where seq_testInfo = pSeq_testInfo;
```

```
if sql%rowcount = 0 then

    dbms_output.put_line('해당 시험 정보가 존재하지 않습니다.');
```

else

```
    dbms_output.put_line('시험 날짜가 추가되었습니다.');
```

end if;

else

```
    dbms_output.put_line('해당 과목은 아직 종료되지 않았습니다.');
```

end if;

```
end procTestDateIn;
```

/

begin

```
    procTestDateIn(1, 1, TO_DATE('2023-10-03', 'YYYY-MM-DD'), TO_DATE('2023-10-04', 'YYYY-MM-DD'));
```

end;

/

-- 2.6. 시험 문제 추가

create or replace procedure procTestQuestionIn (

    pSeq\_teacher in number,

    pSeq\_question in number,

    pSeq\_openSubjectList in number,

    pSeq\_examPaper in number,

    pQuestion in varchar2,

    pAnswer in varchar2,

    pKind in varchar2

)

is

    vEnddate date;

begin

    -- tblopensubjectlist 테이블의 enddate 확인

```
select osl.enddate into vEnddate
from tblOpenSubjectList osl
    inner join tblTestInfo ti
        on ti.seq_openSubjectList = osl.seq_openSubjectList
    where osl.seq_teacher = pSeq_teacher
        and ti.seq_openSubjectList = pSeq_openSubjectList;

-- enddate가 오늘 이전인 경우에만 시험 문제 추가
if vEnddate < sysdate then
    -- 시험 문제 추가
    insert into tblQuestion (seq_question, question, answer)
    values (pSeq_question, pQuestion, pAnswer);

    insert into Tblexampaper (seq_examPaper, seq_question, seq_openSubjectList, kind)
    values (pSeq_examPaper, pSeq_question, pSeq_openSubjectList, pKind);
```

```
if sql%rowcount = 0 then
    dbms_output.put_line('해당 시험 정보가 존재하지 않습니다.');
```

else

```
    dbms_output.put_line('시험 문제가 추가되었습니다.');
```

end if;

else

```
    dbms_output.put_line('해당 과목은 아직 종료되지 않았습니다.');
```

end if;

```
end procTestQuestionIn;
```

/

begin

```
    procTestQuestionIn(1, 1, 1, 1, '자바 소스 확장자는?', '3', '필기');
```

end;

/

-- 교사 번호를 입력하면 본인이 수강 중인 과목 목록 다 출력

-- 4. 과목 목록 출력 시 과목번호, 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 출결, 필기, 실기 배점 등이 출력되고,

-- 특정 과목을 과목번호로 선택 시 출결 배점, 필기 배점, 실기 배점, 시험 날짜, 시험 문제를 입력할 수 있는 화면으로 연결되어야 한다.  
(자바에서 구현해야할 듯)

-- 5. 배점 등록이 안 된 과목인 경우는 과목 정보가 출력될 때 배점 부분은 null 값으로 출력한다.

```
create or replace procedure procSubjectListOut (
```

```
    pSeq_teacher in number
```

```
)
```

```
is
```

```
    cursor vcursor is
```

```
        select
```

```
        vc.seq_openSubjectList "과정 번호",
```

```
        vc.seq_subject "과목 번호",
```

```
        vc.c_name "과정명",
```

```
        vc.oc_startDate "과정 시작일",
```



```
vc.oc_enddate "과정 종료일",  
vc.r_name 강의실,  
vc.s_name 과목명,  
vc.osl_startdate "과목 시작일",  
vc.osl_enddate "과목 종료일",  
b.name 교재명,  
ti.attendancePoints "출결 배점",  
ti.writtenPoints "필기 배점",  
ti.practicalPoints "실기 배점"  
from vwCurriculum vc  
    inner join tblTextbook b  
        on b.seq_textbook = vc.seq_textbook  
    inner join tblTestInfo ti  
        on ti.seq_openSubjectList = vc.seq_openSubjectList  
    where vc.seq_teacher = pSeq_teacher  
    and vc.osl_enddate < sysdate;
```

```
vrecord vcursor%rowtype;
```

```
begin
```

```
-- 정보 출력
```

```
open vcursor;
```

```
loop
```

```
    fetch vcursor into vrecord;
```

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

```
    dbms_output.put_line('과정 번호: ' || vrecord."과정 번호");
```

```
    dbms_output.put_line('과목 번호: ' || vrecord."과목 번호");
```

```
    dbms_output.put_line('과정명: ' || vrecord."과정명");
```

```
    dbms_output.put_line('과정 시작일: ' || vrecord."과정 시작일");
```

```
    dbms_output.put_line('과정 종료일: ' || vrecord."과정 종료일");
```

```
    dbms_output.put_line('강의실: ' || vrecord.강의실);
```

```
    dbms_output.put_line('과목명: ' || vrecord.과목명);
```

```

    dbms_output.put_line('과목 시작일: ' || vrecord."과목 시작일");

    dbms_output.put_line('과목 종료일: ' || vrecord."과목 종료일");

    dbms_output.put_line('교재명: ' || vrecord.교재명);

    dbms_output.put_line('출결 배점: ' || vrecord."출결 배점");

    dbms_output.put_line('필기 배점: ' || vrecord."필기 배점");

    dbms_output.put_line('실기 배점: ' || vrecord."실기 배점");

    dbms_output.put_line('-----');

end loop;

close vcursor;

exception

when no_data_found then

    dbms_output.put_line('해당 과목이 존재하지 않습니다.');
```

when others then

```

    dbms_output.put_line('오류 발생: ' || SQLERRM);

end procSubjectListOut;

/
```

```
begin
  procSubjectListOut(1);
end;
/
```

요구사항번호	C-3(교사)	작성일	2023-03-08
요구사항명	성적 입 출력	작성자	정혜정
DML			
create or replace procedure procTraineeInfoOut ( pSeq_teacher in number,			

```
pSeq_openSubjectList in number
)
is
cursor vcursor is
    select
        distinct osl.seq_subjectList "과목 목록 번호",
        vt.seq_trainee "학생 번호",
        vt.t_name "이름",
        vt.t_id "아이디",
        vt.t_tel "전화번호"
    from vwTrainees vt
        inner join tblOpenSubjectList osl
            on osl.seq_openCurriculum = vt.seq_openCurriculum
        where osl.seq_teacher = pSeq_teacher
        and osl.seq_openSubjectList = pSeq_openSubjectList
        and osl.enddate < sysdate;
```

```
vrecord vcursor%rowtype;
```

```
begin
```

```
-- 정보 출력
```

```
open vcursor;
```

```
loop
```

```
    fetch vcursor into vrecord;
```

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

```
-- 정보 출력
```

```
    dbms_output.put_line('과목 목록 번호: ' || vrecord."과목 목록 번호");
```

```
    dbms_output.put_line('학생 번호: ' || vrecord."학생 번호");
```

```
    dbms_output.put_line('이름: ' || vrecord."이름");
```

```
    dbms_output.put_line('아이디: ' || vrecord."아이디");
```

```
    dbms_output.put_line('전화번호: ' || vrecord."전화번호");
```

```
        dbms_output.put_line('-----');
    end loop;

    close vcursor;
exception
    when no_data_found then
        dbms_output.put_line('해당 정보가 없습니다.');
```

when others then

```
        dbms_output.put_line('오류 발생: ' || SQLERRM);
    end procTraineeInfoOut;
/

begin
    procTraineeInfoOut (1, 1);
end;
/
```

-- 3.2. 성적 출력

```
create or replace procedure procTestGradesOut (
```

```
    pSeq_teacher in number,
```

```
    pSeq_openSubjectList in number
```

```
)
```

```
is
```

```
    cursor vcursor is
```

```
        select
```

```
            distinct tl.seq_traineeList "교육생 목록 번호",
```

```
            t.name 이름,
```

```
            t.tel 전화번호,
```

```
            tl.status 상태,
```

```
            tl.day "상태 완료 날짜",
```

```
            g.attendanceGrade "출결 점수",
```

```
            g.writtenGrade "필기 점수",
```



```
g.practicalGrade "실기 점수"
```

```
from tblTraineeList tl
```

```
left outer join tblOpenSubjectList osl
```

```
on osl.seq_openCurriculum = tl.seq_openCurriculum
```

```
left outer join tblGrades g
```

```
on g.seq_traineeList = tl.seq_traineeList
```

```
left outer join tblTrainees t
```

```
on t.seq_trainee = tl.seq_trainee
```

```
where osl.seq_teacher = pSeq_teacher
```

```
and osl.seq_openSubjectList = pSeq_openSubjectList
```

```
and osl.endDate < sysdate;
```

```
vrecord vcursor%rowtype;
```

```
vCompletion varchar2(10);
```

```
begin
```

```
-- 정보 출력
```

```
open vcursor;
```

```
loop
```

```
    fetch vcursor into vrecord;
```

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

```
-- 출결, 필기, 실기 점수가 모두 입력된 경우에만 "완료" 출력
```

```
if vrecord."출결 점수" is not null and vrecord."필기 점수" is not null and vrecord."실기 점수" is not null then
```

```
    vCompletion := '완료';
```

```
else
```

```
    vCompletion := '미완료';
```

```
end if;
```

```
-- 정보 출력
```

```
dbms_output.put_line('교육생 목록 번호: ' || vrecord."교육생 목록 번호");
```

```
dbms_output.put_line('이름: ' || vrecord."이름");
```

```

    dbms_output.put_line('전화번호: ' || vrecord."전화번호");

    dbms_output.put_line('상태: ' || vrecord."상태");

    dbms_output.put_line('상태 완료 날짜: ' || vrecord."상태 완료 날짜");

    dbms_output.put_line('출결 점수: ' || vrecord."출결 점수");

    dbms_output.put_line('필기 점수: ' || vrecord."필기 점수");

    dbms_output.put_line('실기 점수: ' || vrecord."실기 점수");

    dbms_output.put_line('성적 등록 여부: ' || vCompletion);

    dbms_output.put_line('-----');

end loop;

close vcursor;

exception

when no_data_found then

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

```

when others then

    dbms_output.put_line('오류 발생: ' || SQLERRM);
```

```
end procTestGradesOut;
```

```
/
```

```
begin
```

```
    procTestGradesOut (1, 1);
```

```
end;
```

```
/
```

요구사항번호	C-4(관리자)	작성일	2023-03-08
요구사항명	출결 현황 기간별로 구분 후 조회	작성자	정혜정
DML			
create or replace procedure procTraineeAttendanceOut (			

```
pSeq_teacher in number,  
pSeq_openSubjectList in number,  
pStartDate in date,  
pEndDate in date  
)  
is  
cursor vcursor is  
    select  
        vc.seq_openCurriculum "개설 교육과정 번호",  
        asl.seq_subject "과목 번호",  
        tl.seq_trainee "교육생 번호",  
        a.day 날짜,  
        to_char(a.inTime, 'HH24:MI:SS') "출근 시간",  
        to_char(a.outTime, 'HH24:MI:SS') "퇴근 시간",  
        tas.situation "상태"  
    from vwCurriculum vc
```

```
inner join tblAvailableSubjectList asl
```

```
on asl.seq_subject = vc.seq_subject
```

```
inner join tblTraineeList tl
```

```
on tl.seq_openCurriculum = vc.seq_openCurriculum
```

```
inner join tblAttendance a
```

```
on a.seq_traineeList = tl.seq_traineeList
```

```
inner join tblAttendanceStatus tas
```

```
on tas.seq_attendanceStatus = a.seq_attendanceStatus
```

```
where a.inTime between pStartDate and pEndDate
```

```
and vc.seq_teacher = pSeq_teacher
```

```
and vc.seq_openSubjectList = pSeq_openSubjectList
```

```
order by tl.seq_trainee;
```

```
vrecord vcursor%rowtype;
```

```
begin
```

```
-- 정보 출력
```

```
open vcursor;

loop

    fetch vcursor into vrecord;

    exit when vcursor%notfound;

    -- 정보 출력

    dbms_output.put_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");

    dbms_output.put_line('과목 번호: ' || vrecord."과목 번호");

    dbms_output.put_line('교육생 번호: ' || vrecord."교육생 번호");

    dbms_output.put_line('날 짜: ' || vrecord.날 짜);

    dbms_output.put_line('출근 시간: ' || vrecord."출근 시간");

    dbms_output.put_line('퇴근 시간: ' || vrecord."퇴근 시간");

    dbms_output.put_line('상 태: ' || vrecord.상 태);

    dbms_output.put_line('-----');

end loop;
```

```
close vcursor;

exception

    when no_data_found then

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

when others then

```
        dbms_output.put_line('오류 발생: ' || SQLERRM);
end procTraineeAttendanceOut;

/

begin

    procTraineeAttendanceOut (1, 1, to_date('2023-09-04', 'YYYY-MM-DD'), to_date('2023-10-04', 'YYYY-MM-DD'));
end;

/

-- 4.1.2. 특정(특정 과정, 특정 인원) 출결 현황을 조회할 수 있어야 한다.

-- 특정 과정
```



```
create or replace procedure procSpecificCurriculumAttendanceOut (
```

```
    pSeq_teacher in number,
```

```
    pSeq_openSubjectList in number,
```

```
    pSeq_openCurriculum in number
```

```
)
```

```
is
```

```
    cursor vcursor is
```

```
        select
```

```
            vc.seq_openCurriculum "개설 교육과정 번호",
```

```
            asl.seq_subject "과목 번호",
```

```
            tl.seq_trainee "교육생 번호",
```

```
            a.day 날짜,
```

```
            to_char(a.inTime, 'HH24:MI:SS') "출근 시간",
```

```
            to_char(a.outTime, 'HH24:MI:SS') "퇴근 시간"
```

```
        from vwCurriculum vc
```

```
            inner join tblAvailableSubjectList asl
```

```
        on asl.seq_subject = vc.seq_subject
        inner join tblTraineeList tl
            on tl.seq_openCurriculum = vc.seq_openCurriculum
            inner join tblAttendance a
                on a.seq_traineeList = tl.seq_traineeList
                where vc.seq_openCurriculum = pSeq_openCurriculum
                    and vc.seq_teacher = pSeq_teacher
                    and vc.seq_openSubjectList = pSeq_openSubjectList
                    order by tl.seq_trainee;

vrecord vcursor%rowtype;

begin
    -- 정보 출력
    open vcursor;
    loop
        fetch vcursor into vrecord;
```

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

```
-- 정보 출력
```

```
dbms_output.put_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");
```

```
dbms_output.put_line('과목 번호: ' || vrecord."과목 번호");
```

```
dbms_output.put_line('교육생 번호: ' || vrecord."교육생 번호");
```

```
dbms_output.put_line('날 짜: ' || vrecord.날 짜);
```

```
dbms_output.put_line('출근 시간: ' || vrecord."출근 시간");
```

```
dbms_output.put_line('퇴근 시간: ' || vrecord."퇴근 시간");
```

```
dbms_output.put_line('-----');
```

```
end loop;
```

```
close vcursor;
```

```
exception
```

```
when no_data_found then
```

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

```
when others then

    dbms_output.put_line('오류 발생: ' || SQLERRM);

end procSpecificCurriculumAttendanceOut;

/

begin

    procSpecificCurriculumAttendanceOut (1, 1, 7);

end;

/

-- 특정 인원

create or replace procedure procSpecificTraineeAttendanceOut (

    pSeq_teacher in number,

    pSeq_openSubjectList in number,

    pSeq_trainee in number
```

)

is

cursor vcursor is

select

vc.seq\_openCurriculum "개설 교육과정 번호",

asl.seq\_subject "과목 번호",

tl.seq\_trainee "교육생 번호",

a.day 날짜,

to\_char(a.inTime, 'HH24:MI:SS') "출근 시간",

to\_char(a.outTime, 'HH24:MI:SS') "퇴근 시간"

from vwCurriculum vc

inner join tblAvailableSubjectList asl

on asl.seq\_subject = vc.seq\_subject

inner join tblTraineeList tl

on tl.seq\_openCurriculum = vc.seq\_openCurriculum

inner join tblAttendance a

```
        on a.seq_traineeList = tl.seq_traineeList

        where vc.seq_openSubjectList = pSeq_openSubjectList

              and vc.seq_teacher = pSeq_teacher

              and tl.seq_trainee = pSeq_trainee

              order by tl.seq_trainee;

vrecord vcursor%rowtype;

begin
    -- 정보 출력

    open vcursor;

    loop

        fetch vcursor into vrecord;

        exit when vcursor%notfound;

        -- 정보 출력

        dbms_output.put_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");
```

```
dbms_output.put_line('과목 번호: ' || vrecord."과목 번호");

dbms_output.put_line('교육생 번호: ' || vrecord."교육생 번호");

dbms_output.put_line('날 짜: ' || vrecord.날 짜);

dbms_output.put_line('출근 시간: ' || vrecord."출근 시간");

dbms_output.put_line('퇴근 시간: ' || vrecord."퇴근 시간");

dbms_output.put_line('-----');

end loop;


close vcursor;

exception

when no_data_found then

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

when others then

```
    dbms_output.put_line('오류 발생: ' || SQLERRM);

end procSpecificTraineeAttendanceOut;

/
```

```
begin
  procSpecificTraineeAttendanceOut (1, 1, 1);
end;
/
```

요구사항번호	C-5(교사)	작성일	2023-03-08
요구사항명	강사가 자신의 이름 검색 하면 수료한 학생들이 작성한 평가 확인	작성자	김민곤
DML			
<pre>create or replace procedure procTce(   pname varchar2 )</pre>			



is

vt\_name varchar2(10);

vce\_grade number;

vce\_content varchar2(10000);

CURSOR vcursor IS

select distinct

t.name as 강사명,

ce.grade as 점수,

ce.content as 내용

from tblCurriculumEvaluation ce

inner join tblTraineeList tl

on tl.seq\_traineeList = ce.seq\_traineeList

inner join tblOpenCurriculum oc

on oc.seq\_openCurriculum = tl.seq\_openCurriculum

inner join tblOpenSubjectList sl

```
        on oc.seq_openCurriculum = sl.seq_openCurriculum

        inner join tblTeacher t

            on t.seq_teacher = sl.seq_teacher

            where t.name = ptname;

begin

    open vcursor;

    loop

        fetch vcursor into vt_name, vce_grade, vce_content;

        exit when vcursor%notfound;

        DBMS_OUTPUT.PUT_LINE('점수: ' || vce_grade || ' | 내용: ' || vce_content);

        dbms_output.put_line('-----');
        -----');

    end loop;

    close vcursor;

end procTce;

/
```

```
begin
proctce('김민곤');
end;
/
```

요구사항번호	D-1(교육생)	작성일	2023-03-08
요구사항명	강사명 검색시 과목별 정보와 배점,시험문제 조회	작성자	김민곤
DML			
CREATE OR REPLACE PROCEDURE proctga( pname VARCHAR2 )			

IS

vtc\_name VARCHAR2(15); -- 강사명

vg\_seq\_subject NUMBER; -- 과목 번호

vg\_s\_name VARCHAR2(100); -- 과목명

vb\_name VARCHAR2(100); -- 교재명

vosl\_startdate DATE; -- 과목 시작일

vosl\_enddate DATE; -- 과목 종료일

vg\_writtenDate DATE; -- 필기날짜

vg\_practicalDate DATE; -- 실기날짜

vg\_WRITTENPOINTS NUMBER; -- 필기배점

vg\_practicalPoints NUMBER; -- 실기배점

vg\_attendancePoints NUMBER; -- 출결배점

vq\_question VARCHAR2(5000); -- 시험 문제

CURSOR vcursor IS

SELECT DISTINCT

tc.name AS 강사명,  
g.seq\_subject AS 과목번호,  
g.s\_name AS 과목명,  
b.name AS 교재명,  
osl.startdate AS 과목시작일,  
osl.enddate AS 과목종료일,  
g.writtenPoints AS 필기배점,  
g.practicalPoints AS 실기배점,  
g.attendancePoints AS 출결배점,  
g.writtenDate AS 필기날짜,  
g.practicalDate AS 실기날짜,  
q.question AS 문제

FROM

vwgrades g

INNER JOIN tbltraineelist tl ON g.seq\_trainee = tl.seq\_trainee

INNER JOIN tbltrainees t ON tl.seq\_trainee = t.seq\_trainee

```
INNER JOIN tblteacher tc ON g.seq_teacher = tc.seq_teacher
```

```
INNER JOIN tblopensubjectlist osl ON g.seq_opensubjectlist = osl.seq_opensubjectlist
```

```
INNER JOIN tbltextbook b ON osl.seq_textbook = b.seq_textbook
```

```
INNER JOIN tblopencurriculum oc ON g.seq_opencurriculum = oc.seq_opencurriculum
```

```
INNER JOIN tblroom r ON oc.seq_room = r.seq_room
```

```
INNER JOIN tblexampaper ep ON osl.seq_openSubjectList = ep.seq_openSubjectList
```

```
INNER JOIN tblquestion q ON ep.seq_question = q.seq_question
```

```
WHERE
```

```
tc.name = ptname
```

```
ORDER BY
```

```
g.seq_subject asc;
```

```
BEGIN
```

```
OPEN vcursor;
```

```
LOOP
```

```
FETCH vcursor INTO vtc_name, vg_seq_subject, vg_s_name, vb_name, vosl_startdate, vosl_enddate, vg_writtenPoints,
```

```

vg_practicalPoints, vg_attendancePoints, vg_writtenDate, vg_practicalDate, vq_question;

EXIT WHEN vcursor%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('강사명: ' || vtc_name || ' | 과목번호: ' || vg_seq_subject || ' | 과목명: ' || vg_s_name || ' | 교재명: ' || vb_name
||
                        '| 과목시작일: ' || vosl_startdate || ' | 과목종료일: ' || vosl_enddate || ' | 필기배점: ' || vg_writtenPoints || ' | 실기배점: ' ||
vg_practicalPoints ||
                        '| 출결배점: ' || vg_attendancePoints || ' | 필기날짜: ' || vg_writtenDate || ' | 실기날짜: ' || vg_practicalDate || ' | 시험문제: ' ||
vq_question);

DBMS_OUTPUT.PUT_LINE('-----
-----');

END LOOP;

CLOSE vcursor;

END proctga;

/

begin

```

```
proctga('김민곤');
```

```
end;
```

```
/
```

```
--D-1.2교육생 이름으로 성적 검색 프로시저
```

```
CREATE OR REPLACE PROCEDURE proctg(
```

```
    pname VARCHAR2
```

```
)
```

```
IS
```

```
    vt_name VARCHAR2(500);
```

```
    vs_name VARCHAR2(500);
```

```
    vosl_startdate DATE;
```

```
    vosl_enddate DATE;
```

```
    vg_writtenDate DATE;
```

```
    vg_practicalDate DATE;
```



vg\_writtenGrade NUMBER;

vg\_practicalGrade NUMBER;

vg\_attendanceGrade NUMBER;

CURSOR vcursor IS

SELECT DISTINCT

t.name AS 교육생이름,

g.s\_name AS 과목명,

osl.startdate AS 과목시작일,

osl.enddate AS 과목종료일,

g.writtenDate AS 필기날짜,

g.practicalDate AS 실기날짜,

g.writtenGrade AS 필기점수,

g.practicalGrade AS 실기점수,

g.attendanceGrade AS 출결점수

FROM

```
vwgrades g
```

```
INNER JOIN tbltraineelist tl ON g.seq_traineeelist = tl.seq_traineeelist
```

```
INNER JOIN tbltrainees t ON tl.seq_trainee = t.seq_trainee
```

```
INNER JOIN tblteacher tc ON g.seq_teacher = tc.seq_teacher
```

```
INNER JOIN tblopensubjectlist osl ON g.seq_opensubjectlist = osl.seq_opensubjectlist
```

```
INNER JOIN tbltextbook b ON osl.seq_textbook = b.seq_textbook
```

```
INNER JOIN tblopencurriculum oc ON g.seq_opencurriculum = oc.seq_opencurriculum
```

```
INNER JOIN tblroom r ON oc.seq_room = r.seq_room
```

```
INNER JOIN tblexampaper ep ON osl.seq_opensubjectlist = ep.seq_opensubjectlist
```

```
INNER JOIN tblquestion q ON ep.seq_question = q.seq_question
```

```
WHERE
```

```
    t.name = ptname;
```

```
BEGIN
```

```
    OPEN vcursor;
```

LOOP

FETCH vcursor INTO vt\_name, vs\_name, vosl\_startdate, vosl\_enddate, vg\_writtenDate, vg\_practicalDate, vg\_writtenGrade,  
vg\_practicalGrade, vg\_attendanceGrade;

EXIT WHEN vcursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('수강생이름: ' || vt\_name || ' | 과목명: ' || vs\_name || ' | 과목시작일: ' || vosl\_startdate || ' | 과목종료일: ' ||  
vosl\_enddate ||

' | 필기날짜: ' || vg\_writtenDate || ' | 실기날짜: ' || vg\_practicalDate || ' | 필기점수: ' || vg\_writtenGrade || ' | 실기점수: ' ||  
vg\_practicalGrade ||

' | 출결점수: ' || vg\_attendanceGrade);

dbms\_output.put\_line('-----  
-----');

END LOOP;

CLOSE vcursor;

END proctg;

/

begin

```
proctg('곤백안');
```

```
end;
```

```
/
```

```
-- D-1
```

```
-- 성적조회
```

```
CREATE OR REPLACE PROCEDURE procTraineeInfo(
```

```
    p_name IN vwtrainees.t_name%TYPE
```

```
) IS
```

```
    CURSOR c_info IS
```

```
        SELECT DISTINCT vt.t_name, vt.t_id, vt.t_ssn, vt.t_tel, vc.s_name, vc.osl_startdate, vc.osl_enddate
```

```
        FROM vwtrainees vt
```

```
        INNER JOIN vwgrades vg ON vt.seq_opencurriculum = vg.seq_opencurriculum
```

```
        INNER JOIN vwCurriculum vc ON vc.seq_opensubjectlist = vg.seq_opensubjectlist
```

```
        WHERE vt.t_name = p_name
```

```
ORDER BY vt.t_name, vt.t_id, vt.t_ssn, vt.t_tel, vc.s_name, vc.osl_startdate, vc.osl_enddate;

v_info c_info%ROWTYPE;

BEGIN

OPEN c_info;

LOOP

    FETCH c_info INTO v_info;

    EXIT WHEN c_info%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE('교육생 이름: ' || v_info.t_name || ', 아이디: ' || v_info.t_id || ', 주민등록번호: ' || v_info.t_ssn || ', 전화번호: ' || v_info.t_tel || ', 과목 이름: ' || v_info.s_name || ', 시작 날짜: ' || v_info.osl_startdate || ', 종료 날짜: ' || v_info.osl_enddate);

END LOOP;

CLOSE c_info;

EXCEPTION

WHEN NO_DATA_FOUND THEN

    DBMS_OUTPUT.PUT_LINE('해당 이름을 가진 교육생의 정보를 찾을 수 없습니다.');
```

```
WHEN OTHERS THEN

    ROLLBACK;

    RAISE;
```

```
END procTraineeInfo;
```

```
/
```

```
BEGIN
```

```
    procTraineeInfo('모백양');
```

```
END;
```

```
/
```

요구사항번호	D-3(교육생)	작성일	2023-03-08
요구사항명	교육 과정 평가 (수료 학생만 가능)	작성자	장원준
DML			
CREATE OR REPLACE PROCEDURE procAddCurriculumEvaluation( p_seq_curriculum IN tblCurriculumEvaluation.seq_opencurriculum%TYPE,			

```
p_seq_opencurriculum IN tblCurriculumEvaluation.seq_opencurriculum%TYPE,  
p_seq_trainee IN tblCurriculumEvaluation.seq_traineeelist%TYPE,  
p_score IN tblCurriculumEvaluation.grade%TYPE,  
p_comment IN tblCurriculumEvaluation.content%TYPE
```

```
) IS
```

```
CURSOR c_trainee IS
```

```
    SELECT s.name, tl.status FROM tbltraineeelist tl
```

```
    INNER JOIN tbltrainees s ON s.seq_trainee = tl.seq_trainee;
```

```
v_trainee c_trainee%ROWTYPE;
```

```
BEGIN
```

```
    OPEN c_trainee;
```

```
    LOOP
```

```
        FETCH c_trainee INTO v_trainee;
```

```
        EXIT WHEN c_trainee%NOTFOUND;
```

```
        IF v_trainee.status = '수료' THEN
```

```
        INSERT INTO tblCurriculumEvaluation VALUES (p_seq_curriculum, p_seq_opencurriculum, p_seq_trainee, p_score, p_comment);

        DBMS_OUTPUT.PUT_LINE('교육생 이름: ' || v_trainee.name || ', 상태: ' || v_trainee.status || ', 교육 평가 정보가 추가되었습니다.');
```

END IF;

END LOOP;

CLOSE c\_trainee;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('교육 평가 정보 추가에 실패하였습니다.');

RAISE;

END procAddCurriculumEvaluation;

/

BEGIN

procAddCurriculumEvaluation(1, 1, 1, 5, '설명을 자세하게 해주신다.');

END;

/



요구사항번호	D-4(교육생)	작성일	2023-03-08
요구사항명	교사 추천 도서 조회	작성자	장원준
DML			
<pre>-- 교사 추천 도서 조회  create or replace procedure procTextbookInfo(     pSeq_trainee in varchar2 ) is     cursor vcursor is         select             distinct osl.seq_opencurriculum 교육과정명,             tea.name 교사명,             rb.grade 별점,</pre>			

```
b.name "책 제목",
```

```
b.publisher "출판사명"
```

```
from tblrecommendtextbook rb
```

```
inner join tblopensubjectlist osl
```

```
on rb.seq_teacher = osl.seq_teacher
```

```
inner join tbltraineelist tl
```

```
on tl.seq_opencurriculum = osl.seq_opencurriculum
```

```
inner join tblteacher tea
```

```
on tea.seq_teacher = osl.seq_teacher
```

```
inner join tbltextbook b
```

```
on b.seq_textbook = rb.seq_textbook
```

```
where osl.seq_teacher = rb.seq_teacher
```

```
and tl.seq_trainee = pSeq_trainee;
```

```
vrecord vcursor%rowtype;
```

```
begin
```

```
open vcursor;

loop

    fetch vcursor into vrecord;

    exit when vcursor%notfound;


    dbms_output.put_line('교육과정명: ' || vrecord.교육과정명 || ', 교사명: ' || vrecord.교사명 || ', 별점: ' || vrecord.별점 || ', 책 제목: ' ||
vrecord."책 제목" || ', 출판사명: ' || vrecord."출판사명");

dbms_output.put_line('-----');

end loop;

close vcursor;

exception

when no_data_found then

    dbms_output.put_line('교과서 정보를 찾을 수 없습니다.');
```

```
when others then

    rollback;

    raise;
```

```
end procTextbookInfo;
```

```
/
```

```
begin
```

```
    procTextbookInfo(1);
```

```
end;
```

```
/
```

요구사항번호	C-4(교사)	작성일	2023-03-08
요구사항명	출결 현황 기간별로 구분 후 조회	작성자	정혜정
DML			
create or replace procedure procTraineeAttendanceOut ( pSeq_teacher in number, pSeq_openSubjectList in number,			

```
pStartDate in date,  
pEndDate in date  
)  
is  
cursor vcursor is  
    select  
        vc.seq_openCurriculum "개설 교육과정 번호",  
        asl.seq_subject "과목 번호",  
        tl.seq_trainee "교육생 번호",  
        a.day 날짜,  
        to_char(a.inTime, 'HH24:MI:SS') "출근 시간",  
        to_char(a.outTime, 'HH24:MI:SS') "퇴근 시간",  
        tas.situation "상태"  
    from vwCurriculum vc  
        inner join tblAvailableSubjectList asl  
            on asl.seq_subject = vc.seq_subject
```

```
inner join tblTraineeList tl
```

```
on tl.seq_openCurriculum = vc.seq_openCurriculum
```

```
inner join tblAttendance a
```

```
on a.seq_traineeList = tl.seq_traineeList
```

```
inner join tblAttendanceStatus tas
```

```
on tas.seq_attendanceStatus = a.seq_attendanceStatus
```

```
where a.inTime between pStartDate and pEndDate
```

```
and vc.seq_teacher = pSeq_teacher
```

```
and vc.seq_openSubjectList = pSeq_openSubjectList
```

```
order by tl.seq_trainee;
```

```
vrecord vcursor%rowtype;
```

```
begin
```

```
-- 정보 출력
```

```
open vcursor;
```

```
loop
```

```
fetch vcursor into vrecord;
```

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

```
-- 정보 출력
```

```
dbms_output.put_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");
```

```
dbms_output.put_line('과목 번호: ' || vrecord."과목 번호");
```

```
dbms_output.put_line('교육생 번호: ' || vrecord."교육생 번호");
```

```
dbms_output.put_line('날짜: ' || vrecord.날짜);
```

```
dbms_output.put_line('출근 시간: ' || vrecord."출근 시간");
```

```
dbms_output.put_line('퇴근 시간: ' || vrecord."퇴근 시간");
```

```
dbms_output.put_line('상태: ' || vrecord.상태);
```

```
dbms_output.put_line('-----');
```

```
end loop;
```

```
close vcursor;
```

```
exception
```

```

when no_data_found then

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

when others then

```

    dbms_output.put_line('오류 발생: ' || SQLERRM);
end procTraineeAttendanceOut;

/

begin

    procTraineeAttendanceOut (1, 1, to_date('2023-09-04', 'YYYY-MM-DD'), to_date('2023-10-04', 'YYYY-MM-DD'));

end;

/

-- 4.1.2. 특정(특정 과정, 특정 인원) 출결 현황을 조회할 수 있어야 한다.

-- 특정 과정

create or replace procedure procSpecificCurriculumAttendanceOut (

    pSeq_teacher in number,
```



```
pSeq_openSubjectList in number,  
pSeq_openCurriculum in number  
)  
is  
cursor vcursor is  
    select  
        vc.seq_openCurriculum "개설 교육과정 번호",  
        asl.seq_subject "과목 번호",  
        tl.seq_trainee "교육생 번호",  
        a.day 날짜,  
        to_char(a.inTime, 'HH24:MI:SS') "출근 시간",  
        to_char(a.outTime, 'HH24:MI:SS') "퇴근 시간"  
    from vwCurriculum vc  
        inner join tblAvailableSubjectList asl  
            on asl.seq_subject = vc.seq_subject  
        inner join tblTraineeList tl
```

```
on tl.seq_openCurriculum = vc.seq_openCurriculum
```

```
inner join tblAttendance a
```

```
on a.seq_traineeList = tl.seq_traineeList
```

```
where vc.seq_openCurriculum = pSeq_openCurriculum
```

```
and vc.seq_teacher = pSeq_teacher
```

```
and vc.seq_openSubjectList = pSeq_openSubjectList
```

```
order by tl.seq_trainee;
```

```
vrecord vcursor%rowtype;
```

```
begin
```

```
-- 정보 출력
```

```
open vcursor;
```

```
loop
```

```
fetch vcursor into vrecord;
```

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

```

-- 정보 출력

dbms_output.put_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");

dbms_output.put_line('과목 번호: ' || vrecord."과목 번호");

dbms_output.put_line('교육생 번호: ' || vrecord."교육생 번호");

dbms_output.put_line('날 짜: ' || vrecord.날 짜);

dbms_output.put_line('출근 시간: ' || vrecord."출근 시간");

dbms_output.put_line('퇴근 시간: ' || vrecord."퇴근 시간");

dbms_output.put_line('-----');

end loop;


close vcursor;

exception

when no_data_found then

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

```

when others then

    dbms_output.put_line('오류 발생: ' || SQLERRM);
```

```
end procSpecificCurriculumAttendanceOut;
```

```
/
```

```
begin
```

```
    procSpecificCurriculumAttendanceOut (1, 1, 7);
```

```
end;
```

```
/
```

```
-- 특정 인원
```

```
create or replace procedure procSpecificTraineeAttendanceOut (
```

```
    pSeq_teacher in number,
```

```
    pSeq_openSubjectList in number,
```

```
    pSeq_trainee in number
```

```
)
```

```
is
```

```
cursor vcursor is
```

```
select
```

```
vc.seq_openCurriculum "개설 교육과정 번호",
```

```
asl.seq_subject "과목 번호",
```

```
tl.seq_trainee "교육생 번호",
```

```
a.day 날짜,
```

```
to_char(a.inTime, 'HH24:MI:SS') "출근 시간",
```

```
to_char(a.outTime, 'HH24:MI:SS') "퇴근 시간"
```

```
from vwCurriculum vc
```

```
inner join tblAvailableSubjectList asl
```

```
on asl.seq_subject = vc.seq_subject
```

```
inner join tblTraineeList tl
```

```
on tl.seq_openCurriculum = vc.seq_openCurriculum
```

```
inner join tblAttendance a
```

```
on a.seq_traineeList = tl.seq_traineeList
```

```
where vc.seq_openSubjectList = pSeq_openSubjectList
```

```
        and vc.seq_teacher = pSeq_teacher
        and tl.seq_trainee = pSeq_trainee
        order by tl.seq_trainee;

vrecord vcursor%rowtype;

begin
    -- 정보 출력

    open vcursor;

    loop

        fetch vcursor into vrecord;

        exit when vcursor%notfound;

        -- 정보 출력

        dbms_output.put_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");

        dbms_output.put_line('과목 번호: ' || vrecord."과목 번호");

        dbms_output.put_line('교육생 번호: ' || vrecord."교육생 번호");
```

```
    dbms_output.put_line('날 짜: ' || vrecord.날 짜);

    dbms_output.put_line('출근 시간: ' || vrecord."출근 시간");

    dbms_output.put_line('퇴근 시간: ' || vrecord."퇴근 시간");

    dbms_output.put_line('-----');

end loop;


close vcursor;

exception

when no_data_found then

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

when others then

```
    dbms_output.put_line('오류 발생: ' || SQLERRM);

end procSpecificTraineeAttendanceOut;

/

begin
```

```
procSpecificTraineeAttendanceOut (1, 1, 1);  
  
end;  
  
/
```

요구사항번호	B-3 (관리자)	작성일	2023-03-08
요구사항명	강의진행여부 자동 변경	작성자	
PROCEDURE			
<pre>create or replace procedure procUpdateCurriProgress  is      vSeq number;      vStartDate DATE;      vEndDate DATE;</pre>			



```
cursor vcursor is

    select seq_opencurriculum, startdate, enddate

    from tblopencurriculum;

begin

    open vcursor;

    loop

        fetch vcursor into vSeq, vStartdate, vEnddate;

        exit when vcursor%notfound;


        -- 개강일이 현재보다 미래일 때

        if vstartDate > sysdate

            then update tblopencurriculum set seq_curriculumprogress = 1 where seq_opencurriculum = vseq;


        -- 현재가 개강일과 종료일 사이일 때

        ELSIF vstartDate < sysdate and venddate > sysdate

            then update tblopencurriculum set seq_curriculumprogress = 2 where seq_opencurriculum = vseq;
```

```
-- 현재가 종료일보다 미래일 때

ELSIF venddate < sysdate

    then update tblopencurriculum set seq_curriculumprogress = 3 where seq_opencurriculum = vseq;

END IF;

end loop;

close vcursor;

end procUpdateCurriProgress;

/

begin

    procUpdateCurriProgress();

end;

/
```

요구사항번호	B-4 (관리자)	작성일	2023-03-08
요구사항명	강의 종료 시 교육생을 수료 상태로 변경하고 수료 날짜를 자동 업데이트	작성자	
PROCEDURE			
<pre>-- 테스트를 위한 null 만들기  update tbltraineeList  set status = null,     day = null  where seq_traineeList between 1 and 20 and seq_traineeList != 10;  rollback;  -- 과정 종료된 목록 불러오기  select tl.seq_traineeList, oc.endDate, tl.status, tl.day</pre>			

```
from tblTraineeList tl

    left outer join tblOpenCurriculum oc

        on tl.seq_openCurriculum = oc.seq_openCurriculum

where oc.endDate < sysdate;
```

-- 전 체

```
select tl.seq_traineeList, oc.seq_openCurriculum, oc.endDate, tl.status, tl.day
from tblTraineeList tl

    left outer join tblOpenCurriculum oc

        on tl.seq_openCurriculum = oc.seq_openCurriculum;
```

create or replace procedure procUpdateCurriCompletionDate

is

```
    vSeq_traineeList tblTraineeList.seq_traineeList%type;
```

```
vEndDate tblOpenCurriculum.endDate%type;
```

```
vStatus tblTraineeList.status%type;
```

```
cursor vcursor is
```

```
    select tl.seq_traineeList, oc.endDate, tl.status
```

```
    from tblTraineeList tl
```

```
        left outer join tblOpenCurriculum oc
```

```
            on tl.seq_openCurriculum = oc.seq_openCurriculum
```

```
    where oc.endDate < sysdate; --과정이 종료된 경우
```

```
begin
```

```
    open vcursor;
```

```
    loop
```

```
        fetch vcursor into vSeq_traineeList, vEndDate, vStatus;
```

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

```
-- 날짜가 비어있는 경우

if vStatus is null then

    update tblTraineeList

    set status = '수료',

        day = vEndDate

    where seq_traineeList = vSeq_traineeList;

end if;


end loop;

close vcursor;

end procUpdateCurriCompletionDate;

/


--프로시저 실행

begin

    procUpdateCurriCompletionDate;
```

end;

/

요구사항번호	B-8 (관리자)	작성일	2023-03-08
요구사항명	출결 테이블에 결석생 자동 추가	작성자	
PROCEDURE			
<pre>create or replace procedure procUpdateAbsent is     vHoliday date;     vtrue number;     vStart date;     vEnd date;     vStatus varchar2(30);</pre>			

vSeq number;

vcnt number;

vcnt1 number;

cursor vcursor1 is

select holiday

from tblPublicHoliday;

cursor vcursor2 is

select oc.startdate, oc.enddate, tl.status, tl.seq\_traineeList

from tblOpenCurriculum oc

inner join tblTraineeList tl

on oc.seq\_opencurriculum = tl.seq\_opencurriculum;

begin

vtrue := 0;

open vcursor1;



```
loop
```

```
    fetch vcursor1 into vHoliday;
```

```
    exit when vcursor1%notfound;
```

```
    -- 오늘이 휴일이나 주말이면 vtrue가 0 이상
```

```
    if vholiday = sysdate or to_char(sysdate, 'd') = '1' or to_char(sysdate, 'd') = '7'
```

```
        then vtrue := vtrue + 1;
```

```
    end if;
```

```
end loop;
```

```
close vcursor1;
```

```
-- 오늘이 평일이면 vtrue = 0
```

```
if vtrue = 0 then
```

```
    open vcursor2;
```

```
    loop
```

```
        fetch vcursor2 into vStart, vEnd, vStatus, vSeq;
```

```
exit when vcursor2%notfound;
```

```
select count(*) into vcnt from tblAttendance
```

```
where to_char(day, 'yyyy-mm-dd') = to_char(sysdate, 'yyyy-mm-dd') and seq_traineeList = vseq;
```

```
-- 오늘 아무기록이 없고 수강중일 때 결석 입력
```

```
if (sysdate between vstart and vend) and vcnt = 0 and vstatus is null
```

```
then insert into tblAttendance values ((select max(seq_attendance) from tblAttendance) + 1, vseq, 7, to_date(sysdate, 'yyyy-mm-dd'), null, null);
```

```
end if;
```

```
end loop;
```

```
close vcursor2;
```

```
end if;
```

```
end procUpdateAbsent;
```

```
/
```

```
begin
```

```
procUpdateAbsent();  
  
end;  
  
/
```

요구사항번호	B-8 (관리자)	작성일	2023-03-08
요구사항명	출결 상황을 자동 업데이트	작성자	
PROCEDURE			
<pre>create or replace procedure procUpdateaAttendanceStatus is    vSeq_attendance tblAttendance.seq_attendance%type;    vInTime tblAttendance.inTime%type;    vOutTime tblAttendance.outTime%type;    vSeq_attendanceStatus tblAttendance.seq_attendanceStatus%type;</pre>			

```
vCurrentTime varchar2(10); -- 현재 시간을 담아둘 변수
```

```
cursor vcursor is
```

```
    select
```

```
        seq_attendance,
```

```
        inTime,
```

```
        outTime,
```

```
        seq_attendanceStatus
```

```
    from tblAttendance;
```

```
begin
```

```
    -- 현재 시간의 시간 부분을 가져오기
```

```
    vCurrentTime := to_char(sysdate, 'HH24:MI');
```

```
    open vcursor;
```

loop

fetch vcursor into vSeq\_attendance, vInTime, vOutTime, vSeq\_attendanceStatus;

exit when vcursor%notfound;

-- seq\_attendanceStatus가 null이면 값을 넣음

if vSeq\_attendanceStatus is null then

-- 9시 전에 오고, 17시 50분 이후 퇴실 한 경우

if to\_char(vInTime, 'HH24:MI') < '09:00' and to\_char(vOutTime, 'HH24:MI') >= '17:50' then

update tblAttendance

set seq\_attendanceStatus = 1 -- 정상

where seq\_attendance = vSeq\_attendance;

-- inTime이 null이고 현재 시간이 9시 이후인 경우 or 9시 이후에 온 경우

elsif vInTime is null and vCurrentTime >= '09:00' or to\_char(vInTime, 'HH24:MI') >= '09:00' then

update tblAttendance

set seq\_attendanceStatus = 2 -- 지각

```

        where seq_attendance = vSeq_attendance;

        -- outTime이 null이고 현재 시간이 18시 이후인 경우 or 그 전에 간 경우

        elsif vOutTime is null and vCurrentTime >= '18:00' or to_char(vOutTime, 'HH24:MI') < '17:50' then

            update tblAttendance

            set seq_attendanceStatus = 3 -- 조퇴

            where seq_attendance = vSeq_attendance;

        end if;

    end if;

end loop;

close vcursor;

end;

/

begin

    procUpdateaAttendanceStatus;

```

end;

/

요구사항번호	B-4 (관리자)	작성일	2023-03-08
요구사항명	과목 기간들이 과점 기간보다 작거나 같도록 제한하는 트리거	작성자	
TRIGGER			
<pre>CREATE OR REPLACE TRIGGER trgCheckPeriodSum BEFORE INSERT OR UPDATE ON tblSubjectList FOR EACH ROW DECLARE     v_seq_curri number;     v_period_total NUMBER;</pre>			

```
v_period_now NUMBER;

v_period NUMBER;

BEGIN

v_seq_curri := :new.seq_curriculum;

-- 해당 과정의 총 period를 가져옵니다.

SELECT period INTO v_period_total

from tblcurriculum c

    inner join tblcourseperiod cp

        on c.seq_courseperiod = cp.seq_courseperiod

        where c.seq_curriculum = v_seq_curri;


--현재 period 합을 가져옵니다.

SELECT SUM(period) into v_period_now

from tblSubjectList sl

inner join tblSubject s

    on sl.seq_subject = s.seq_subject
```



```
where sl.seq_curriculum = v_seq_curri;
```

```
--추가한 과목의 기간을 가져옵니다.
```

```
select period into v_period from tblsubject where seq_subject = :NEW.seq_subject;
```

```
-- 새로운 코스 기간이나 업데이트된 코스 기간의 period 합이 tblsubject의 period 합과 같은지 확인합니다.
```

```
IF v_period_total < v_period_now+v_period THEN
```

```
    RAISE_APPLICATION_ERROR(-20001, '과정 총 기간을 확인해주세요.');
```

```
END IF;
```

```
END;
```

```
/
```

요구사항번호	C-3 (교사)	작성일	2023-03-08
--------	----------	-----	------------

요구사항명	필기, 실기, 출석 최대 점수를 제한하는 트리거	작성자	
TRIGGER			
<pre> CREATE OR REPLACE TRIGGER trgCheckPoints BEFORE INSERT ON tblTestInfo FOR EACH ROW DECLARE     v_total_points NUMBER; BEGIN     -- writtenpoints, practicalpoints, attendancepoints의 합을 계산합니다.     v_total_points := :NEW.writtenpoints + :NEW.practicalpoints + :NEW.attendancepoints;      IF :NEW.writtenpoints &gt;= 40 THEN         RAISE_APPLICATION_ERROR(-20002, '최대 40까지 입력 가능합니다.');</pre>			
END IF;			

```
IF :NEW.practicalpoints >= 40 THEN
    RAISE_APPLICATION_ERROR(-20003, '최대 40까지 입력 가능합니다.');
```

END IF;

  

```
IF :NEW.attendancepoints >= 30 THEN
    RAISE_APPLICATION_ERROR(-20004, '최대 30까지 입력 가능합니다.');
```

END IF;

  

-- writtenpoints, practicalpoints, attendancepoints의 합이 100을 초과하는 경우 오류를 발생시킵니다.

```
IF v_total_points > 100 THEN
    RAISE_APPLICATION_ERROR(-20005, '총점이 100을 초과할 수 없습니다.');
```

END IF;

END;

/

요구사항번호	C-2 (교사)	작성일	2023-03-08
요구사항명	교육생 성적이 100점이 넘지 않게 제한하는 트리거	작성자	
TRIGGER			
<pre> CREATE OR REPLACE TRIGGER trgTraineeCheckPoints BEFORE INSERT ON tblgrades FOR EACH ROW DECLARE     v_total_grade NUMBER;     v_seq_testinfo number;     v_writtenpoints number;     v_practicalpoints number;     v_attendancepoints number; BEGIN     -- writtengrade, practicalgrade, attendancegrade의 합을 계산합니다.     v_total_grade := :NEW.writtengrade + :NEW.practicalgrade + :NEW.attendancegrade; </pre>			

```
v_seq_testinfo := :new.seq_testinfo;
```

```
select writtenpoints, practicalpoints, v_attendancepoints into v_writtenpoints, v_practicalpoints, v_attendancepoints  
from tbltestinfo where seq_testinfo = v_seq_testinfo;
```

```
IF :NEW.writtengrade > v_writtenpoints THEN
```

```
    RAISE_APPLICATION_ERROR(-20006, '최대 배점까지만 입력 가능합니다.');
```

```
END IF;
```

```
IF :NEW.practicalgrade > v_practicalpoints THEN
```

```
    RAISE_APPLICATION_ERROR(-20007, '최대 배점까지만 입력 가능합니다.');
```

```
END IF;
```

```
IF :NEW.attendancegrade > v_attendancepoints THEN
```

```
    RAISE_APPLICATION_ERROR(-20008, '최대 배점까지만 입력 가능합니다.');
```

```
END IF;
```

```
-- writtengrade, practicalgrade, attendancegrade의 합이 100을 초과하는 경우 오류를 발생시킵니다.  
  
IF v_total_grade > 100 THEN  
  
    RAISE_APPLICATION_ERROR(-20009, '총점이 100을 초과할 수 없습니다.');  
END IF;  
  
END;  
  
/
```

요구사항번호	D-3 (교육생)	작성일	2023-03-08
요구사항명	교육생이 수료했을 때만 평가를 할 수 있게 제한하는 트리거	작성자	
TRIGGER			
CREATE OR REPLACE TRIGGER trgAllowEvaluation			

```
BEFORE INSERT ON tblcurriculumevaluation
FOR EACH ROW
DECLARE
    v_status VARCHAR2(20);
BEGIN
    -- tbltrainelist에서 해당 trainee의 status를 가져옵니다.
    SELECT status INTO v_status
    FROM tbltrainelist
    WHERE seq_trainelist = :NEW.seq_trainelist;

    -- status가 '수료'인 경우에만 데이터를 삽입할 수 있습니다.
    IF v_status != '수료' or v_status is null THEN
        RAISE_APPLICATION_ERROR(-20010, '수료 상태일 때만 평가를 입력할 수 있습니다.');
```

END IF;

```
END;
```

-- 5번

-- 성적은 종료 후 입력이 되니까 그 전에 입력 하지 못하도록 하는 트리거

```
CREATE OR REPLACE TRIGGER trgCheckEnddate
BEFORE INSERT ON tblGrades
FOR EACH ROW
DECLARE
    v_enddate DATE;
BEGIN
    -- tblOpenSubjectList에서 해당 과목의 종료일(enddate)을 가져옵니다.
    SELECT osl.enddate INTO v_enddate
    from tblTestInfo ti
        inner join tblOpenSubjectList osl
            on ti.seq_opensubjectlist = osl.seq_opensubjectlist
        where ti.seq_testinfo = :NEW.seq_testInfo;

    -- 종료일(enddate) 이전에 성적 입력 시도시 차단
    IF v_enddate > SYSDATE THEN
        RAISE_APPLICATION_ERROR(-20011, '과목 종료일 이후에만 시험 정보를 삽입할 수 있습니다.');
```



```

END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20012, '해당 과목 ID가 tblOpensubjectList에 존재하지 않습니다.');
```

END;

/

요구사항번호	C-2 (교사)	작성일	2023-03-08
요구사항명	과목 종료일을 가져오고 종료일 이전에 성적 입력 시 제한하는 트리거	작성자	
TRIGGER			
CREATE OR REPLACE TRIGGER trgCheckEnddate  BEFORE INSERT ON tblGrades			

```
FOR EACH ROW
DECLARE
    v_enddate DATE;
BEGIN
    -- tblOpensubjectList에서 해당 과목의 종료일(enddate)을 가져옵니다.
    SELECT osl.enddate INTO v_enddate
    from tblTestInfo ti
        inner join tblOpenSubjectList osl
            on ti.seq_opensubjectlist = osl.seq_opensubjectlist
        where ti.seq_testinfo = :NEW.seq_testInfo;

    -- 종료일(enddate) 이전에 성적 입력 시도시 차단
    IF v_enddate > SYSDATE THEN
        RAISE_APPLICATION_ERROR(-20011, '과목 종료일 이후에만 시험 정보를 삽입할 수 있습니다.');
```

```
    END IF;
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN

    RAISE_APPLICATION_ERROR(-20012, '해당 과목 ID가 tblOpensubjectList에 존재하지 않습니다.');
```

END;

/

요구사항번호	C-3 (교사)	작성일	2023-03-08
요구사항명	중도 탈락한 학생에 대한 성적 입력을 제한하는 트리거	작성자	
TRIGGER			
CREATE OR REPLACE TRIGGER trgCheckDropout			
BEFORE INSERT OR UPDATE ON tblGrades			
FOR EACH ROW			
DECLARE			

```
v_status VARCHAR2(20);

v_day DATE;

v_writtendate date;

v_practicaldate date;

BEGIN

-- tbltrainelist에서 해당 trainee의 status와 day 값을 가져옵니다.

SELECT status, day INTO v_status, v_day

FROM tbltrainelist

    WHERE seq_trainelist = :new.seq_trainelist;


--writtendate와 practicaldate 값을 가져와서 비교합니다.

select writtendate, practicaldate into v_writtendate, v_practicaldate from tblTestInfo where seq_testinfo = :new.seq_testinfo;


-- status가 '중도탈락'이고, 시험 날짜가 중도탈락 날짜 이후면 성적을 삽입 불가

IF v_status = '중도탈락' and (v_writtendate > v_day OR v_practicaldate > v_day)

    THEN RAISE_APPLICATION_ERROR(-20013, '중도탈락 상태일 때는 해당 날짜의 성적을 삽입할 수 없습니다.');
```

```

END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20014, '해당 trainee_id가 tbltraineeelist에 존재하지 않습니다.');
```

END;

/

요구사항번호	B-13 (관리자)	작성일	2023-03-08
요구사항명	출석 인정 상태로 변했을 때 출결 상태의 번호가 1로 변하는 트리거	작성자	정원혁
TRIGGER			
create or replace trigger trgUpdateAttendance after update of admitattendance on tblAttendancePapers			

```
for each row
begin
  if :new.admitattendance = '출석 인정' then
    update tblAttendance a
    set a.seq_attendanceStatus = 1
    where a.seq_traineeList = :new.seq_traineeList and a.day = :new.day;
  end if;
end;
/
```

요구사항번호	B-13 (관리자)	작성일	2023-03-08
--------	------------	-----	------------

요구사항명	교육과정 기간 수료 중에 면접 스케줄을 잡을 수 없게하는 트리거	작성자	장원준
TRIGGER			
<pre> CREATE OR REPLACE TRIGGER trgInterviewScheduleInsert BEFORE INSERT ON tblinterviewschedule FOR EACH ROW DECLARE     v_curriculum_progress_seq NUMBER;     v_trainee_exists NUMBER; BEGIN     -- Check if seq_trainee exists in tbltraineeelist table     SELECT COUNT(*) INTO v_trainee_exists     FROM tbltraineeelist     WHERE seq_trainee = :NEW.seq_trainee; </pre>			

```
-- If seq_trainee does not exist in tbltraineeelist table, allow insert

IF v_trainee_exists = 0 THEN

    RETURN;6

END IF;


-- Check if seq_curriculumprogress is not null in tblogencurriculum table

SELECT seq_curriculumprogress

INTO v_curriculum_progress_seq

FROM tblogencurriculum

WHERE seq_curriculumprogress = (SELECT seq_opencurriculum FROM tbltraineeelist WHERE seq_trainee = :NEW.seq_trainee);


IF v_curriculum_progress_seq IS NOT NULL THEN

    -- Raise an exception to prevent the insert

    RAISE_APPLICATION_ERROR(-20015, 'seq_curriculumprogress 컬럼이 NULL이 아닌 경우에는 tblinterviewschedule 테이블에 insert 할 수 없습니다.');
```

END IF;



END;

/

/

요구사항번호	B-13 (관리자)	작성일	2023-03-08
요구사항명	합격자만 교육생 목록으로 insert 할 수 있게 제한하는 트리거	작성자	장원준
TRIGGER			
<pre>CREATE OR REPLACE TRIGGER trgPerInsertTraineeList BEFORE INSERT ON tbltraineeList FOR EACH ROW DECLARE     v_status tblinterviewResults.Status%TYPE;     v_seq_schedule tblinterviewschedule.seq_schedule%TYPE;</pre>			

```
BEGIN
```

```
-- tblinterviewschedule 테이블에서 해당 seq_trainee의 seq_schedule 값을 가져옵니다.
```

```
SELECT seq_schedule INTO v_seq_schedule
```

```
FROM tblinterviewschedule
```

```
WHERE seq_trainee = :NEW.seq_trainee;
```

```
-- 가져온 seq_schedule 값이 tblinterviewResults 테이블에 있는지 확인합니다.
```

```
SELECT Status INTO v_status
```

```
FROM tblinterviewResults
```

```
WHERE seq_schedule = v_seq_schedule;
```

```
-- 가져온 seq_schedule 값이 없거나, 가져온 Status 값이 '불합격'인 경우에만 삽입을 막습니다.
```

```
IF v_status = '불합격' OR v_seq_schedule IS NULL THEN
```

```
    RAISE_APPLICATION_ERROR(-20016, '불합격 또는 아직 면접을 보지 않았습니다.');
```

```
END IF;
```

```
END;
```

/

요구사항번호	B-13 (관리자)	작성일	2023-03-08
요구사항명	수료자만 취업 현황 테이블에 insert 할수 있게하는 트리거	작성자	장원준
TRIGGER			
<pre>CREATE OR REPLACE TRIGGER trgEmploymentStatus BEFORE INSERT ON tblEmploymentStatus FOR EACH ROW DECLARE     v_status VARCHAR2(20); BEGIN     -- tbltrainee에서 해당 trainee의 status를 가져옵니다.     SELECT status INTO v_status</pre>			

```
FROM tbltrainelist  
  
WHERE seq_trainelist = :NEW.seq_trainelist;  
  
-- status가 '수료'인 경우에만 데이터를 삽입할 수 있습니다.  
  
IF v_status != '수료' or v_status is null THEN  
  
    RAISE_APPLICATION_ERROR(-20017, '수료 상태일 때만 평가를 입력할 수 있습니다.');
```

END IF;

END;

/