

## 1조 DB Object 스크립트(PL-SQL)

요구사항번호	B-01	작성일	2022-05-19
요구사항명	관리자 계정	작성자	김성태
DML			
<p>계정으로 로그인 후 관리자 기본정보 간단하게 출력</p> <pre>create or replace procedure admin_P(     p_admin_ssn in number,     p_result out sys_refcursor ) is begin     open p_result     for     select         admin.seq,         admin.name as "이름",         admin.tel as "연락처"     from admin</pre>			

```
where ssn = p_admin_ssn;  
end admin_P;
```

요구사항번호	B-02	작성일	2022-05-19
요구사항명	기초정보 관리기능	작성자	김성태
DML			
<pre>--과정 명 관리 select * from curriculum;  --과정 명 추가 create or replace procedure c_insert_p(p_seq in curriculum.seq%type, p_name in curriculum.name%type) is begin     insert into curriculum(seq,name) values(p_seq,p_name);      commit;</pre>			

```
    dbms_output.put_line('INSERT SUCCESS');
end c_insert_p;

--과정 명 수정
CREATE OR REPLACE PROCEDURE c_update_p ( cur_seq_p IN NUMBER, cur_name_p IN VARCHAR2)
IS
BEGIN
    UPDATE curriculum
    SET seq = cur_seq_p,
        name = cur_name_p
    WHERE seq = cur_seq_p;

    EXCEPTION WHEN OTHERS THEN

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

ROLLBACK;

```
END;

--과정 명 삭제
create or replace procedure c_delete_p(p_seq in curriculum.seq%type)
is
```

```
begin

    delete from curriculum

    where p_seq = seq;

    commit;

end c_delete_p;


--과목 명 관리
select * from subject;

--과목 명 추가
create or replace procedure s_insert_p(p_seq in subject.seq%type, p_subject_name IN subject.name%type)
is
begin

    insert into subject(seq, subject_name) values(p_seq,p_subject_name);

    commit;

    dbms_output.put_line('INSERT SUCCESS');

end s_insert_p;


--과목 명 수정
create or replace procedure s_update_p ( sub_seq_p in number, sub_subject_name_p in varchar2)
is
```

```
begin
    update subject
    set seq = sub_seq_p,
        subject_name = sub_subject_name_p
    where seq = sub_seq_p;
    exception when others then
        dbms_output.put_line('추가에 실패했습니다.');
```

rollback;

```
end;
```

--과목 명 삭제

```
create or replace procedure s_delete_p(p_seq in subject.seq%type)
```

```
is
```

```
begin
```

```
    delete from subject
```

```
    where p_seq = seq;
```

```
    commit;
```

```
end s_delete_p;
```

--강의실 명 관리

```
select * from lectureroom;
```

--강의실 명 추가

```
create or replace procedure l_insert_p(p_seq in lectureroom.seq%type, p_rname IN lectureroom.room_name%type, p_sl in lectureroom.student_limit%type)
```

```
is
```

```
begin
```

```
    insert into lectureroom(seq, room_name, student_limit) values(p_seq,p_rname, p_sl);
```

```
    commit;
```

```
    dbms_output.put_line('INSERT SUCCESS');
```

```
end l_insert_p;
```

--강의실 명 수정

```
create or replace procedure l_update_p ( lect_seq_p in number, lect_room_name_p in varchar2, lect_student_limit_p in number)
```

```
is
```

```
begin
```

```
    update lectureroom
```

```
    set seq = lect_seq_p,
```

```
        room_name = lect_room_name_p,
```

```
        student_limit = lect_student_limit_p
```

```
    where seq = lect_seq_p;
```

```
exception when others then

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

rollback;

end;

  

--강의실 명 삭제

create or replace procedure l\_delete\_p(p\_seq in lectureroom.seq%type)

is

begin

delete from lectureroom

where p\_seq = seq;

commit;

end l\_delete\_p;

  

--교재 명 관리

select \* from book;

  

--교재 명 추가

create or replace procedure b\_insert\_p(p\_seq in book.seq%type, p\_sub IN book.subject%type, p\_aut in book.author%type,

p\_pub in book.publisher%type, p\_puye in book.pub\_year%type)

```
is
begin
    insert into book(seq, subject, author, publisher, pub_year) values(p_seq, p_sub, p_aut, p_pub, p_puye);
    commit;
    dbms_output.put_line('INSERT SUCCESS');
end b_insert_p;
```

--교재 명 수정

```
create or replace procedure b_update_p ( b_seq_p in number, b_subject_p in varchar2, b_author_p in varchar2,
                                         b_publisher_p in varchar2, b_pub_year_p in date)
```

```
is
begin
    update book
    set seq = b_seq_p,
        subject = b_subject_p,
        author = b_author_p,
        publisher = b_publisher_p,
        pub_year = b_pub_year_p
```



```
        where seq = b_seq_p;

        exception when others then

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

rollback;

end;

--교재 명 삭제

create or replace procedure b\_delete\_p(p\_seq in book.seq%type)

is

begin

delete from book

where p\_seq = seq;

commit;

end b\_delete\_p;

요구사항번호	B-03	작성일	2022-05-19
요구사항명	교사계정 관리기능	작성자	김성태

#### DML

--교사정보 등록

```
create or replace procedure teacher_insert_p(p_seq in teacher.seq%type, p_name IN teacher.name%type, p_ssn in teacher.ssn%type,
                                             p_tel in teacher.tel%type)
```

is

begin

```
    insert into teacher(seq, name, ssn, tel) values(p_seq, p_name, p_ssn, p_tel);
```

```
    commit;
```

```
    dbms_output.put_line('INSERT SUCCESS');
```

```
end teacher_insert_p;
```

--교사정보 수정

```
create or replace procedure t_update_p ( t_seq_p in number, t_name_p in varchar2, t_ssn_p in number, t_tel_p in varchar2)
```

is

begin

```
    update teacher
```

```
    set seq = t_seq_p,
```

```
        name = t_name_p,
```

```
        ssn = t_ssn_p,
```

```
        tel = t_tel_p
```

```

        where seq = t_seq_p;
        exception when others then
        dbms_output.put_line('추가에 실패했습니다.');
```

rollback;

```

end;

--교사정보 삭제
create or replace procedure teacher_delete_p(p_seq in teacher.seq%type)
is
begin
    delete from teacher
    where p_seq = seq;
    commit;
end teacher_delete_p;

--교사의 강의과목에 대한 정보 조회
create or replace procedure open_subject_book_P (
    p_open_subject_seq in number,
    p_result out sys_refcursor
)
is
begin
    open p_result
    for
    select subject
```

```
from book b
  inner join subject_book s
    on b.seq = s.book_seq
    where s.subject_seq = (select subject_seq from open_subject where open_subject. p_open_subject_seq);
end open_subject_book_P;
```

```
set SERVEROUTPUT ON;
```

```
declare
```

```
  v_result sys_refcursor;
  v_open_subject_seq open_subject.seq%type;
  v_open_subject_state varchar2(15);
  v_curriculum_name curriculum.name%type;
  v_open_curs_begin open_curs.begin_date%type;
  v_open_curs_end open_curs.end_date%type;
  v_lectureroom_name lectureroom.room_name%type;
  v_open_subject_begin open_subject.begin_date%type;
  v_open_subject_end open_subject.end_date%type;
  v_book_name sys_refcursor;
  v_subject_name subject.subject_name%type;
  v_student_num number;
  v_seq number;
```

```
begin
```

```
  teacher_subject_P(1, v_result);
  v_seq :=1;
```

```

dbms_output.put_line('=====강의 스케줄 조회=====');

loop
    fetch v_result into v_open_subject_seq, v_open_subject_state, v_curriculum_name, v_open_curs_begin, v_open_curs_end, v_lectureroom_name,
        v_subject_name, v_open_subject_begin, v_open_subject_end, v_student_num;
    exit when v_result%notfound;

    dbms_output.put_line('-----');
    dbms_output.put_line(v_seq || '. ' || '[' || v_open_subject_state || '] ' || v_subject_name || ' (과목번호: ' || v_open_subject_seq || ')');
    dbms_output.put_line('        - 기간: ' || v_open_subject_begin || ' ~ ' || v_open_subject_end);
    dbms_output.put_line('        - 인원: ' || v_student_num || '명');
    dbms_output.put_line('        - 교재: ' || v_book_name);
    dbms_output.put_line('        - 과정: ' || v_curriculum_name || ' (' || v_open_curs_begin || ' ~ ' || v_open_curs_end || ')');
    dbms_output.put_line('        - 강의실: ' || v_lectureroom_name);

    v_seq := v_seq + 1;

end loop;
close v_result;
dbms_output.put_line('-----');

dbms_output.put_line('=====');
end;

```

요구사항번호	B-04	작성일	2022-05-20
요구사항명	과정 수료 처리	작성자	임종운
DML			
<pre> SELECT      s.seq, m.name, s.drop_out, s.drop_out_date  FROM open_curs oc      INNER JOIN student s ON oc.seq = s.open_curs_seq      INNER JOIN member m ON s.member_seq = m.seq  WHERE oc.seq = 12;  CREATE OR REPLACE PROCEDURE change_complete_date_P(      course_seq in  number, -- 수료 완료과정      complete_date  in date  )  AS  BEGIN      UPDATE student s SET s.drop_out = '수료', s.drop_out_date = complete_date </pre>			

WHERE

EXISTS(

SELECT \*

FROM member m

WHERE s.member\_seq = m.seq

AND s.open\_curs\_seq = course\_seq

)

AND

(s.drop\_out IS NULL

OR

s.drop\_out <> '중도탈락');

END;

/

--과정번호, 수료일자 입력

EXECUTE change\_complete\_date\_P(12, '2022-05-29');

요구사항번호	B-04	작성일	2022-05-20
요구사항명	과정 등록인원 출력	작성자	임종운
DML			
<pre> CREATE OR REPLACE TYPE curs_member_obj AS OBJECT(      name varchar2(50),      ssn varchar2(50),      tel varchar2(50),      regdate date,      drop_out varchar2(15)  );  /  CREATE OR REPLACE TYPE member_tables AS TABLE OF curs_member_obj;  /  CREATE OR REPLACE FUNCTION curs_member_list_F  ( </pre>			



```
input_curs_seq NUMBER

) RETURN member_tables

IS

v_rtn member_tables;

BEGIN

SELECT

    curs_member_obj(rs.name, rs.ssn, rs.tel, rs.regdate, rs.drop_out)

    BULK COLLECT INTO v_rtn

FROM (SELECT

    m.name, m.ssn, m.tel, s.regdate, s.drop_out

FROM (SELECT *

FROM open_curs oc

WHERE seq = input_curs_seq) rs    --seq 가 확실할 과정 번호

    INNER JOIN enrollment e ON rs.seq = e.open_curs_seq

    INNER JOIN member m ON e.member_seq = m.seq
```

```
INNER JOIN student s ON m.seq = s.member_seq) rs;
```

```
RETURN v_rtn;
```

```
END curs_member_list_F;
```

```
/
```

```
DROP TYPE member_tables;
```

```
DROP TYPE curs_member_obj;
```

```
DROP FUNCTION curs_member_list;
```

```
SELECT * FROM TABLE(curs_member_list_F(12));
```

요구사항번호	B-05	작성일	2022-05-20
요구사항명	개설 과정의 과목 정보 출력	작성자	임종운
DML			

```
CREATE OR REPLACE TYPE curs_info_obj AS OBJECT(  
    subject_name varchar2(50),  
    begin_date date,  
    end_date date,  
    subject varchar2(100),  
    name varchar2(15)  
);  
  
/  
  
CREATE OR REPLACE TYPE curs_info_tables AS TABLE OF curs_info_obj;  
  
/  
  
CREATE OR REPLACE FUNCTION search_curs_info_list_F  
  
(  
    input_curs_seq NUMBER  
  
) RETURN curs_info_tables
```

IS

v\_rtn curs\_info\_tables;

BEGIN

SELECT

curs\_info\_obj(info.subject\_name, info.begin\_date, info.end\_date, info.subject, info.name)

BULK COLLECT INTO v\_rtn

FROM (SELECT

s.subject\_name, os.begin\_date, os.end\_date, b.subject ,t.name

FROM (SELECT seq FROM open\_curs oc WHERE seq = 1) sr

INNER JOIN open\_subject os ON sr.seq = os.open\_curs\_seq

INNER JOIN subject s ON os.subject\_seq = s.seq

INNER JOIN subject\_book sb ON s.seq = sb.subject\_seq

INNER JOIN book b ON sb.book\_seq = b.seq

INNER JOIN teacher t ON os.teacher\_seq = t.seq) info;

```
RETURN v_rtn;
```

```
END search_curs_info_list_F;
```

```
/
```

```
DROP TYPE curs_info_obj;
```

```
DROP TYPE curs_info_tables;
```

```
DROP FUNCTION search_curs_info_list_F;
```

```
SELECT * FROM TABLE(search_curs_info_list_F(12));
```

요구사항번호	B-07	작성일	2022-05-19
요구사항명	시험 관리 및 성적 조회 기능	작성자	강지윤

DML

--개설 과목별(수강종료, 수강중인 개설과목만)로 시험 문제 파일 등록 여부(필기,실기), 성적 등록 여부(필기,실기,출결)확인

--프로시저 호출시 원하는 개설과목명을 입력

--프로시저명 : subject\_info\_P

--subject\_info\_P(<개설과목명>)으로 호출.

```

create or replace procedure subject_info_P (
    psubject_name varchar2 := ''
)
is
    jname subject.subject_name%type;
    osseq open_subject.seq%type;
    ossubject_seq open_subject.subject_seq%type;
    ocseq open_curs.seq%type;
    cname curriculum.name%type;
    a1 varchar2(1);
    a2 varchar2(1);
    b1 varchar2(1);
    b2 varchar2(1);
    b3 varchar2(1);
    cursor vcursor is

```

```
select
    DISTINCT j.subject_name as 과목명,
    os.seq as 개설과목번호,
    os.subject_seq as 과목번호,
    oc.seq as 개설과정번호,
    c.name as 개설과정명,
    (case when rq.written_test_question = '시험문제등록안함' then 'N' else 'Y' end) as 필기시험문제등록여부,
    (case when sq.skill_test_question = '시험문제등록안함' then 'N' else 'Y' end) as 실기시험문제등록여부,
    (case when ss.written_score is not null then 'Y' else 'N' end) as 필기성적등록여부,
    (case when ss.skill_score is not null then 'Y' else 'N' end) as 실기성적등록여부,
    (case when ss.attendance_score is not null then 'Y' else 'N' end) as 출결성적등록여부

from open_subject os
    inner join subject_score ss on ss.open_subject_seq = os.seq
    inner join subject j on j.seq = os.subject_seq
    inner join open_curs oc on oc.seq = os.open_curs_seq
    inner join curriculum c on c.seq = oc.curs_seq
    inner join written_test r on r.open_subject_seq = os.seq
    inner join skill_test s on s.open_subject_seq = os.seq
    inner join written_test_question rq on rq.written_test_seq = r.seq
    inner join skill_test_question sq on sq.skill_test_seq = s.seq

where j.subject_name = psubject_name
order by oc.seq;

begin
```

```

open vcursor;
loop
fetch vcursor into jname, osseq, ossubject_seq, ocseq, cname, a1, a2, b1, b2, b3;
exit when vcursor%notfound;

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

        dbms_output.put_line('과목명: ' || jname || ' | 개설과목번호 : ' || osseq || ' | 과목번호: ' || ossubject_seq || ' | 개설과정번호: ' || ocseq || ' |
개설과정명: ' || cname || ' | 필기시험문제 등록여부: ' || a1 ||
                                ' | 실기시험문제 등록여부: ' || a2 || ' | 필기성적등록여부 : ' || b1 || ' | 실기성적등록여부 : ' || b2 || ' |
출결성적등록여부 : ' || b3);
        dbms_output.put_line('-----');
-----');

        end loop;
close vcursor;

end;

--호출
--subject_info_P('개설 과목명')

--ex. 'java'의 개설과목정보와 시험문제파일등록여부, 성적등록여부를 알고싶을때
begin
    subject_info_P('java');
end;
```



--개설 과목별 성적 출력시 개설과목별 수강생의 성적정보 (교육생 이름, 주민번호 뒷자리, 필기점수, 실기점수,출결점수) 출력

--프로시저 호출시 원하는 개설과목명을 입력

--프로시저명 subject\_score\_check\_P

-- subject\_score\_check\_P(<개설과목명>)으로 호출.

create or replace procedure subject\_score\_check\_P (

    popen\_subject\_name varchar2 := ''

)

is

    stseq student.seq%type;

    mname member.name%type;

    mssn member.ssn%type;

    ssubject\_name subject.subject\_name%type;

    sswritten\_score subject\_score.written\_score%type;

    ssskill\_score subject\_score.skill\_score%type;

    ssattendance\_score subject\_score.attendance\_score%type;

    cursor vcursor is

select

    st.seq as "교육생 번호",

    m.name as "교육생 이름",

    m.ssn as "주민번호 뒷자리",

    s.subject\_name as "개설과목명",

    ss.written\_score as 필기점수,

```

        ss.skill_score as 실기점수,
        ss.attendance_score as 출결점수
from student st
    inner join member m on m.seq = st.member_seq
    inner join subject_score ss on ss.student_seq = st.seq
    inner join open_subject os on os.seq = ss.open_subject_seq
    inner join subject s on s.seq = os.subject_seq

where s.subject_name = popen_subject_name;

begin
    open vcursor;
    loop
        fetch vcursor into stseq, mname, msn, ssubject_name, sswritten_score, ssskill_score, ssattendance_score;

        exit when vcursor%notfound;

        dbms_output.put_line('-----');
        dbms_output.put_line('교육생번호: ' || stseq || ' | 교육생이름: ' || mname || ' | 개설과목명: ' || ssubject_name || ' | 필기점수 : ' || sswritten_score ||
'점'
                                || ' | 실기점수 : ' || ssskill_score || '점' || ' | 출결점수 : ' || ssattendance_score || '점');
        dbms_output.put_line('-----');

    end loop;
    close vcursor;
end;

```

```
--호출시 subject_score_check_P('개설과목명')
```

```
--ex. 'java'를 수강하는 모든 교육생의 성적정보를 알고싶을때
```

```
begin
```

```
    subject_score_check_P('java');
```

```
end;
```

#### **--4.교육생 개인별 출력시 교육생정보+교육생이 수강한 모든 개설과목에 대한 성적 정보**

--(교육생번호, 교육생 이름, 주민번호 뒷자리, 개설과정명, 개설과정시작, 개설과정종료, 강의실, 개설 과목명, 교사명, 개설과목시작, 개설과목종료날짜, 필기, 실기, 출결)

--프로시저 호출시 원하는 교육생이름을 입력

--프로시저명 : student\_score\_check\_P

-- student\_score\_check\_P (<교육생이름>)으로 호출.

```
create or replace procedure student_score_check_P (
```

```
    pstudent_name varchar2 := ''
```

```
)
```

is

ssubject\_name subject.subject\_name%type;

stseq student.seq%type;

mname member.name%type;

mssn member.ssn%type;

cname curriculum.name%type;

ocbegin\_date open\_curs.begin\_date%type;

ocend\_date open\_curs.end\_date%type;

lroom\_name lecturerroom.room\_name%type;

tname teacher.name%type;

osbegin\_date open\_subject.begin\_date%type;

osend\_date open\_subject.end\_date%type;

sswritten\_score subject\_score.written\_score%type;

ssskill\_score subject\_score.skill\_score%type;

ssattendance\_score subject\_score.attendance\_score%type;

```
cursor vcursor is
```

```
select
```

```
    distinct s.subject_name as "개설과목명",
```

```
    st.seq as "교육생 번호",
```

```
    m.name as "교육생 이름",
```

```
    m.ssn as "주민번호 뒷자리",
```

```
    c.name as 과정명,
```

```
    oc.begin_date as "과정시작날짜",
```

```
    oc.end_date as "과정종료날짜",
```

```
    l.room_name as 강의실명,
```

```
    t.name as 교사명,
```

```
    os.begin_date as "과목시작날짜",
```

```
    os.end_date as "과목종료날짜",
```

```
    ss.written_score as 필기점수,
```

ss.skill\_score as 실기점수,

ss.attendance\_score as 출결점수

from student st

inner join member m on m.seq = st.member\_seq

inner join subject\_score ss on ss.student\_seq = st.seq

inner join open\_subject os on os.seq = ss.open\_subject\_seq

inner join subject s on s.seq = os.subject\_seq

inner join open\_curs oc on oc.seq = os.open\_curs\_seq

inner join curriculum c on oc.curs\_seq = c.seq

inner join lectureroom l on oc.room\_seq = l.seq

inner join teacher t on os.teacher\_seq = t.seq

inner join subject\_book sb on sb.subject\_seq = s.seq

inner join book b on sb.book\_seq = b.seq

where m.name = pstudent\_name;

```

begin

    open vcursor;

    loop

        fetch vcursor into ssubject_name, stseq, mname, mssn, cname, ocbegin_date, ocend_date, lroom_name, tname, osbegin_date, osend_date,
        sswritten_score, ssskill_score, ssattendance_score;

        exit when vcursor%notfound;

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

        dbms_output.put_line('교육생 번호: ' || stseq || ' | 교육생 이름: ' || mname || ' | 주민번호 뒷자리: ' || mssn || ' | 개설과정명: ' || cname || ' |
        개설과정시작: ' || ocbegin_date || ' | 개설과정종료: ' || ocend_date ||

                                ' | 강의실명: ' || lroom_name || ' | 개설과목명: ' || ssubject_name || ' | 교사명: ' || tname || ' | 개설과목시작: ' ||
        osbegin_date || ' | 개설과목종료: ' || osend_date ||

                                ' | 필기점수: ' || sswritten_score || '점' || ' | 실기점수: ' || ssskill_score || '점' || ' | 출결점수: ' || ssattendance_score ||
        '점');

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

```

```

end loop;

close vcursor;

end;

--호출시 student_score_check_P('교육생이름');

--ex. 교육생 '홍영화'의 기본 정보와 '홍영화'가 수강한 모든 개설과목에 대한 성적 정보를 알고 싶을 때

begin

    student_score_check_P('홍영화');

end;

```

요구사항번호	B-08	작성일	2022-05-19
요구사항명	출결 관리 및 출결 조회	작성자	고민지
DML			



-- 1. 개설과정별 교육생 출결 조회

```
create or replace procedure admin_attendance_inquiry_P(
    p_open_curs_seq in number,
    p_result      out sys_refcursor
)
is
begin
    open p_result
    for
    select
        "학생번호",
        count(decode("출석", '출석', 1)) as "출석일수",
        count(decode("지각조퇴", '출석..', 1)) as "정상일수",
        count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",
        count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",
        count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",
        count(decode("지각조퇴", '외출', 1)) as "외출일수",
        count(decode("지각조퇴", '결석', 1)) as "결석일수",
        count(decode("출석", '병가', 1)) as "병가일수",
        count(decode("출석", '기타', 1)) as "기타일수"
    from
        (select i.*, 출석||지각||조퇴 as "지각조퇴" from attendance_info i
         where "학생번호" in (select seq from student where open_curs_seq = p_open_curs_seq))
        group by "학생번호";

end admin_attendance_inquiry_P;
```

-- 1. 개설과정별 교육생 출결 조회 > 실행문

```
declare
  v_student_name member.name%type;
  v_student_seq  number; --학생번호
  v_attendance   number; --출석일수
  v_normal       number; --정상일수
  v_tardy        number; --지각일수
  v_early_leaving number; --조퇴일수
  v_tardy_leaving number; --지각 및 조퇴일수
  v_out          number; --외출일수
  v_absence      number; --결석일수
  v_sickness     number; --병가일수
  v_warrant      number; --기타일수
  v_result sys_refcursor;
  v_seq          number;
  v_open_curs_seq number;
  v_curs_name     curriculum.name%type;
  v_curs_begin    open_curs.begin_date%type;
  v_curs_end      open_curs.end_date%type;
begin
  v_open_curs_seq := 12;

  select c.name, o.begin_date, o.end_date into v_curs_name, v_curs_begin, v_curs_end
  from open_curs o
  inner join curriculum c on o.curs_seq = c.seq
  where o.seq = v_open_curs_seq;
```

```

dbms_output.put_line('===== 과정 정보 =====');
dbms_output.put_line(' [과목] ' || v_curs_name || '(과정번호: ' || v_open_curs_seq || ')');
dbms_output.put_line('   - 기간: ' || v_curs_begin || ' ~ ' || v_curs_end);
--dbms_output.put_line('   - 인원: ' || subject_student_num(v_open_curs_seq) || '명');
dbms_output.put_line('');

dbms_output.put_line('===== 교육생 출결 조회 =====');

dbms_output.put_line('번호. 이름(학생번호)                               단위: 일');
dbms_output.put_line('          [출석] [정상] [지각] [조퇴] [지각+조퇴] [결석] [외출] [병가] [기타]');
dbms_output.put_line('-----');

admin_attendance_inquiry_P(v_open_curs_seq, v_result);
v_seq := 1;

loop
  fetch v_result into v_student_seq,
  v_attendance, v_normal, v_tardy, v_early_leaving, v_tardy_leaving,
  v_out, v_absence, v_sickness, v_warrant;
  exit when v_result%notfound;
  select name into v_student_name from member where seq = (select member_seq from student where seq = v_student_seq);

  dbms_output.put_line(v_seq || ' ' || v_student_name || ' (' || v_student_seq || ')   ' || v_attendance || '   ' || v_normal || '   ' || v_tardy || '   ' || v_early_leaving || '   ' ||
    v_tardy_leaving || '   ' || v_absence || '   ' || v_out || '   ' || v_sickness || '   ' || v_warrant);

  dbms_output.put_line('-----');
  v_seq := v_seq + 1;

end loop;

```

```

        dbms_output.put_line('=====');

        close v_result;
end;

```

### 3. 특정 교육생의 출결현황 조회

```

create or replace procedure student_attendance_inquiry_P(
    p_student_seq in number,
    p_result      out sys_refcursor
)
is
begin
    open p_result
    for
        select * from attendance_info
            where "학생번호" = p_student_seq
            order by "날짜";
end;

```

### 3. 특정 교육생의 출결현황 조회 > 실행문

```

declare
    v_student_seq    number;
    v_result          sys_refcursor;

```

```

v_attendance_seq attendance.seq%type;
v_attendance_date attendance.attendance_date%type;
v_attendance      varchar2(15);
v_tardy           varchar2(10);
v_early_leaving   varchar2(10);
v_out             varchar2(10);
v_temp            number;
v_member_info      member%rowtype;
v_curs_name        curriculum.name%type;
v_regdate          date;
v_seq             number;
begin
    v_student_seq := 3;

    student_attendance_inquiry_P(v_student_seq, v_result);

    dbms_output.put_line('===== 교육생 출결 조회 =====');

    select * into v_member_info from member where seq = (select member_seq from student where seq = v_student_seq);

    select c.name, s.regdate into v_curs_name, v_regdate from student s
        inner join open_curs o on s.open_curs_seq = o.seq
        inner join curriculum c on o.curs_seq = c.seq
        where s.seq = v_student_seq;

    dbms_output.put_line('이름: '||v_member_info.name||' (교육생 번호:'||v_student_seq||')');
    dbms_output.put_line('등록일: '||v_regdate);
    dbms_output.put_line('전화번호: '||v_member_info.tel);
    dbms_output.put_line('수강과정: '||v_curs_name);

```

```

dbms_output.put_line("");
dbms_output.put_line('===== 교육생 세부 출결 =====');

v_seq := 1;

loop
    fetch v_result into v_temp, v_attendance_seq, v_attendance_date, v_attendance, v_tardy, v_early_leaving, v_out;
    exit when v_result%notfound;

    v_tardy := replace(v_tardy, ',', '');
    v_early_leaving := replace(v_early_leaving, ',', '');
    v_out := replace(v_out, ',', '');

    if v_attendance <> '(주말)' and v_attendance <> '(공휴일)'
    then
        dbms_output.put_line('-----');
        dbms_output.put_line(v_seq||'. '||v_attendance_date||' '||v_attendance||' '||v_tardy||' '||v_early_leaving||' '||v_out);
        v_seq := v_seq + 1;
    end if;

end loop;

dbms_output.put_line('=====');

close v_result;

end;

```

--

요구사항번호	B-09	작성일	2022-05-19
요구사항명	취업 공고 관리 및 조회	작성자	강지윤
DML			
<p>▶ where 절 &lt; &gt; 위치에는 &lt;&gt;안에 있는 내용에 해당하는 데이터 입력</p> <p>--1. 전체 취업 공고조회(회사명, 지역, 분야, 계약형태, 모집인원, 모집시작날짜, 모집종료날짜)</p> <p>--뷰이름 : job_announce_V</p> <p>create or replace view job_announce_V</p> <p>as</p> <p>(</p>			

```
select

    a.company as 회사명,

    city.city as 지역,

    job_field.field_name as 분야,

    contract_type.contract_type as 계약형태,

    a.people as 모집인원,

    a.begin_date as 모집시작날짜,

    a.end_date as 모집종료날짜

from job_announce a

    inner join city on city.seq = a.city_seq

    inner join job_field on job_field.seq = a.field_seq

    inner join contract_type on contract_type.seq = a.contract_seq

)

order by 모집시작날짜;

--전체 취업 공고를 조회하고 싶을 때
```



```
select * from job_announce_V;
```

**--2. 기간별 조회(모집종료날짜가 'yyyy-mm-dd'부터 'yyyy-mm-dd' 사이인 기간 조회)**

--프로시저 호출시 원하는 기간(기간시작날짜, 기간종료날짜) 입력

--프로시저명 : period\_job\_check\_P

-- period\_job\_check\_P (<'기간시작날짜'>, <'기간종료날짜'>)으로 호출.

```
create or replace procedure period_job_check_P (
```

```
    start_period varchar2 := "",
```

```
    end_period varchar2 := "
```

```
)
```

```
is
```

```
    acompany job_announce.company%type;
```

```
    ccity city.city%type;
```

```
   ffield_name job_field.field_name%type;
```

```
    ctcontract_type contract_type.contract_type%type;
```

```
    apeople job_announce.people%type;
```

```
abegin_date job_announce.begin_date%type;
```

```
aend_date job_announce.end_date%type;
```

```
cursor vcursor is
```

```
select
```

```
a.company as 회사명,
```

```
c.city as 지역,
```

```
f.field_name as 분야,
```

```
ct.contract_type as 계약형태,
```

```
a.people as 모집인원,
```

```
a.begin_date as 모집시작날짜,
```

```
a.end_date as 모집종료날짜
```

```
from job_announce a
```

```
inner join city c on c.seq = a.city_seq
```

```
inner join job_field f on f.seq = a.field_seq
```

```
inner join contract_type ct on ct.seq = a.contract_seq
```

```

        where a.end_date between start_period and end_period order by 모집시작날짜;

begin

    open vcursor;

        loop

            fetch vcursor into acompany, ccity, ffield_name, ctcontract_type, apeople, abegin_date, aend_date;

            exit when vcursor%notfound;

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

            dbms_output.put_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield_name || ' | 계약형태: ' || ctcontract_type || ' | 모집인원: ' ||
apeople || '명' || ' | 모집시작날짜: ' ||

                                abegin_date || ' | 모집종료날짜: ' || aend_date);

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

        close vcursor;

end;

--모집종료날짜의 기간 입력

```

```
-- period_job_check_P('기간시작날짜', '기간종료날짜')
```

ex. 모집종료 날짜가 2022-05-07~2022-05-31 사이인 취업공고를 알고싶을때

```
begin
```

```
    period_job_check_P('2022-05-07', '2022-05-31');
```

```
end;
```

**--3.분야별 (분야명으로 검색 ex. 웹개발) 조회**

--프로시저 호출시 원하는 분야명을 입력

--프로시저명 : field\_job\_check\_P

-- field\_job\_check\_P (<'분야명'>)으로 호출.

```
create or replace procedure field_job_check_P (
```

```
    pjob_field varchar2 := ''
```

```
)
```

```
is
```

```
    acompany job_announce.company%type;
```

ccity city.city%type;

ffield\_name job\_field.field\_name%type;

ctcontract\_type contract\_type.contract\_type%type;

apeople job\_announce.people%type;

abegin\_date job\_announce.begin\_date%type;

aend\_date job\_announce.end\_date%type;

cursor vcursor is

select

a.company as 회사명,

c.city as 지역,

f.field\_name as 분야,

ct.contract\_type as 계약형태,

a.people as 모집인원,

a.begin\_date as 모집시작날짜,

a.end\_date as 모집종료날짜

```
from job_announce a

    inner join city c on c.seq = a.city_seq

    inner join job_field f on f.seq = a.field_seq

    inner join contract_type ct on ct.seq = a.contract_seq

where f.field_name = pjob_field order by 모집시작날짜;

begin

    open vcursor;

    loop

        fetch vcursor into acompany, ccity, ffield_name, ctcontract_type, apeople, abegin_date, aend_date;

        exit when vcursor%notfound;

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

        dbms_output.put_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield_name || ' | 계약형태: ' || ctcontract_type || ' | 모집인원: ' ||
apeople || '명' || ' | 모집시작날짜: ' ||
```

```
abegin_date || ' | 모집종료날짜: ' || aend_date);
```

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

```
end loop;
```

```
close vcursor;
```

```
end;
```

```
--호출시 field_job_check_P('분야명');
```

```
--ex. '웹 개발'분야의 취업공고를 확인하고싶을때
```

```
begin
```

```
field_job_check_P('웹 개발');
```

```
end;
```

#### --4. 지역별(지역명으로 검색 - ex. 서울) 조회

--프로시저 호출시 원하는 지역명을 입력

--프로시저명 : city\_job\_check\_P

-- city\_job\_check\_P (<'지역명'>)으로 호출.

```
create or replace procedure city_job_check_P (  
    pcity varchar2 := "  
)
```

is

```
    accompany job_announce.company%type;  
    ccity city.city%type;  
   ffield_name job_field.field_name%type;  
    ctcontract_type contract_type.contract_type%type;  
    apeople job_announce.people%type;  
    abegin_date job_announce.begin_date%type;  
    aend_date job_announce.end_date%type;  
    cursor vcursor is
```



```
select
```

```
    a.company as 회사명,  
    c.city as 지역,  
    f.field_name as 분야,  
    ct.contract_type as 계약형태,  
    a.people as 모집인원,  
    a.begin_date as 모집시작날짜,  
    a.end_date as 모집종료날짜
```

```
from job_announce a
```

```
    inner join city c on c.seq = a.city_seq  
    inner join job_field f on f.seq = a.field_seq  
    inner join contract_type ct on ct.seq = a.contract_seq
```

```
where c.city = pcity order by 모집시작날짜;
```

```
begin
```

```
    open vcursor;
```

```
    loop
```

```
        fetch vcursor into acompany, ccity, ffield_name, ctcontract_type, apeople, abegin_date, aend_date;
```

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

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

```

        dbms_output.put_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield_name || ' | 계약형태: ' || ctcontract_type || ' | 모집인원: ' ||
apeople || '명' || ' | 모집시작날짜: ' ||
                                abegin_date || ' | 모집종료날짜: ' || aend_date);

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

    end loop;
    close vcursor;

end;

--호출시 city_job_check_P(<지역명>)

--ex. '서울'지역의 취업공고를 확인하고싶을 때

begin

    city_job_check_P('서울');

end;
```

요구사항번호	C-02	작성일	2022-05-19
요구사항명	강의 스케줄 조회 기능	작성자	고민지
DML			
<pre> -- 교육생 등록인원 function create or replace function subject_student_num(     pseq open_subject.seq%type ) return number is     v_student_num number; begin     select case         when s.begin_date &gt; sysdate -- 강의예정             then (select count(*) from student where open_curs_seq = c.seq                 and (drop_out_date is null or (drop_out_date &gt;= s.begin_date)))         when s.end_date &lt; sysdate -- 강의종료             then (select count(*) from student where open_curs_seq = c.seq                 and (drop_out_date is null or (drop_out = '수료') or ((drop_out = '중도탈락') and (drop_out_date &gt; s.end_date))))         else (select count(*) from student where open_curs_seq = c.seq --'강의 중'             and (drop_out_date is null or (drop_out_date &gt; sysdate)))     end into v_student_num     from open_subject s     inner join open_curs c         on s.open_curs_seq = c.seq         where s.seq = pseq;      return v_student_num; end; </pre>			

-- 1. 로그인한 교사의 강의정보를 강의예정, 강의 중, 강의 종료로 구분하여 조회

```
create or replace procedure teacher_subject_inquiry_P(
  p_teacher_seq in number,
  p_result out sys_refcursor
)
is
begin
  open p_result
  for
  select
    s.seq as "과목 번호",
    case
      when s.begin_date > sysdate then '강의 예정'
      when s.end_date < sysdate then '강의 종료'
      else '강의 중'
    end as "진행 상태",
    (select name from curriculum where seq = c.curs_seq) as "과정명",
    c.begin_date as "과정 시작일",
    c.end_date as "과정 종료일",
    (select room_name from lectureroom where seq = c.room_seq) as "강의실",
    (select subject_name from subject where seq = s.subject_seq) as "과목명",
    s.begin_date as "과목 시작일",
    s.end_date as "과목 종료일",
    subject_student_num(s.seq) as "교육생 등록인원"
  from open_subject s
  inner join open_curs c
```

```
on s.open_curs_seq = c.seq
where teacher_seq = p_teacher_seq
order by s.begin_date;
```

```
end teacher_subject_inquiry_P;
```

#### -- 1. 과목별 교재명을 조회

```
create or replace procedure open_subject_book_P (
```

```
    p_open_subject_seq in number,
```

```
    p_result out sys_refcursor
```

```
)
```

```
is
```

```
begin
```

```
    open p_result
```

```
    for
```

```
    select subject
```

```
    from book b
```

```
        inner join subject_book s
```

```
        on b.seq = s.book_seq
```

```
        where s.subject_seq = (select subject_seq from open_subject where open_subject.seq = p_open_subject_seq);
```

```
end open_subject_book_P;
```

#### -- 1. 로그인한 교사의 강의정보를 강의예정, 강의 중, 강의 종료로 구분하여 조회 > 출력문

```
declare
```

```
    v_result sys_refcursor;
```

```
    v_open_subject_seq open_subject.seq%type;
```

```
    v_open_subject_state varchar2(15);
```

```
    v_curriculum_name curriculum.name%type;
```

```

v_open_curs_begin open_curs.begin_date%type;
v_open_curs_end open_curs.end_date%type;
v_lectureroom_name lectureroom.room_name%type;
v_open_subject_begin open_subject.begin_date%type;
v_open_subject_end open_subject.end_date%type;
v_subject_name subject.subject_name%type;
v_student_num number;
v_seq number;
v_book_name sys_refcursor;
v_temp book.subject%type;
v_book_seq number;
begin
teacher_subject_inquiry_P(1, v_result);
v_seq := 1;
dbms_output.put_line('=====강의 스케줄 조회=====');

loop
fetch v_result into v_open_subject_seq, v_open_subject_state, v_curriculum_name, v_open_curs_begin, v_open_curs_end, v_lectureroom_name,
v_subject_name, v_open_subject_begin, v_open_subject_end, v_student_num;
exit when v_result%notfound;

dbms_output.put_line('-----');
dbms_output.put_line(v_seq ||'. '||v_open_subject_state||' '||v_subject_name||' (과목번호: '||v_open_subject_seq||)');
dbms_output.put_line('          - 기간: '||v_open_subject_begin||' ~ '||v_open_subject_end||');
dbms_output.put_line('          - 인원: '||v_student_num||'명');

```

```

open_subject_book_P (v_open_subject_seq, v_book_name);

v_book_seq := 1;
loop
    fetch v_book_name into v_temp;
    exit when v_book_name%notfound;
    dbms_output.put_line('          - 교재'||v_book_seq||': '||v_temp);
    v_book_seq := v_book_seq + 1;
end loop;
close v_book_name;

dbms_output.put_line('          - 과정: '||v_curriculum_name || ' ('||v_open_curs_begin||' ~ '||v_open_curs_end||')');
dbms_output.put_line('          - 강의실: '||v_lectureroom_name);

v_seq := v_seq + 1;

end loop;
close v_result;
dbms_output.put_line('-----');
dbms_output.put_line('=====');
end;

```

## -- 2. 로그인한 교사의 특정 강의과목의 수강생 정보 조회

```

create or replace procedure teacher_student_inquiry_P(
    p_open_subject_seq in number,
    p_result out sys_refcursor
)
is

```

```

begin
  open p_result
  for
  select
    m.name as "이름",
    m.tel as "전화번호",
    t.regdate as "등록일",
    case
      when t.drop_out = '중도탈락' and t.drop_out_date < s.end_date then '중도탈락'
      when s.begin_date >= sysdate then '수강예정'
      when (s.begin_date < sysdate) and (sysdate < s.end_date) then '수강중'
      when s.end_date <= sysdate then '수료'
    end as "수강 상태"
  from open_subject s
    inner join open_curs c
      on s.open_curs_seq = c.seq
    inner join student t
      on t.open_curs_seq = c.seq
    inner join member m
      on t.member_seq = m.seq
  where s.seq = p_open_subject_seq;
end teacher_student_inquiry_P;

```

-- 2. 로그인한 교사의 특정 강의과목의 수강생 정보 조회 > 실행문

```

declare
  v_seq number;
  v_name member.name%type;
  v_tel member.tel%type;
  v_regdate student.regdate%type;

```



```

v_state varchar(15);
v_result sys_refcursor;
v_open_subject_seq open_subject.seq%type;
v_subject_name subject.subject_name%type;
v_subject_begin open_subject.begin_date%type;
v_subject_end open_subject.end_date%type;
v_book_name sys_refcursor;
v_book book.subject%type;
v_book_seq number;
begin
    v_open_subject_seq := 1;

    teacher_student_inquiry_P(v_open_subject_seq, v_result);

    select s.subject_name, o.begin_date, o.end_date into v_subject_name, v_subject_begin, v_subject_end
    from open_subject o
        inner join subject s on o.subject_seq = s.seq
        where o.seq = v_open_subject_seq;

    dbms_output.put_line('===== 과목 정보 =====');
    dbms_output.put_line(' [과목] ' || v_subject_name || '(과목 번호: ' || v_open_subject_seq || ')');
    dbms_output.put_line('   - 기간: ' || v_subject_begin || ' ~ ' || v_subject_end);
    dbms_output.put_line('   - 인원: ' || subject_student_num(v_open_subject_seq) || '명');

    open_subject_book_P (v_open_subject_seq, v_book_name);

    v_book_seq := 1;

```

```

loop
    fetch v_book_name into v_book;
    exit when v_book_name%notfound;
    dbms_output.put_line('    - 교재'||v_book_seq||': '||v_book);
    v_book_seq := v_book_seq + 1;
end loop;
close v_book_name;

dbms_output.put_line('');
dbms_output.put_line('===== 교육생 정보 조회 =====');

dbms_output.put_line('-----');
dbms_output.put_line('번호    이름        전화번호        등록일        수강상태');
v_seq := 1;
loop
    fetch v_result into v_name, v_tel, v_regdate, v_state;
    exit when v_result%notfound;
    dbms_output.put_line('-----');
    dbms_output.put_line(v_seq||'    '||v_name||'        '||v_tel||'        '||v_regdate||'        '||v_state);
    v_seq := v_seq + 1;
end loop;

if v_result%rowcount = 0
then dbms_output.put_line('-----');
    dbms_output.put_line('등록한 교육생이 존재하지 않습니다. ');
end if;

close v_result;

```

```

dbms_output.put_line('=====');

end;

```

요구사항번호	C-03	작성일	2022-05-19
요구사항명	평가 관리 및 조회	작성자	고민지
DML			
-- 1. 배점 테이블에 배점 수정, 삭제시 성적데이터와의 무결성 유지  create or replace trigger teacher_point_T before update or delete on point for each row declare v_score_num number; v_subject_seq number; begin  if updating then v_subject_seq := :new.open_subject_seq; elsif deleting then v_subject_seq := :old.open_subject_seq; end if;			

```
select count(*) into v_score_num from subject_score s where s.open_subject_seq = v_subject_seq;

if v_score_num > 0
then raise_application_error(-20100, '학생의 성적 데이터가 존재하여 배점 수정에 실패하였습니다.');
```

end if;

end;

## -- 2. 성적 테이블에 성적 입력, 수정시 배점데이터와의 무결성 유지

```
create or replace trigger teacher_score_T
before insert or update
on subject_score
for each row
declare
v_attend_point point.attend_point%type;
v_written_point point.written_point%type;
v_skill_point point.skill_point%type;
begin

select p.attend_point, p.written_point, p.skill_point
into v_attend_point, v_written_point, v_skill_point
from point p where p.open_subject_seq = :new.open_subject_seq;

if :new.attendance_score > v_attend_point
or :new.written_score > v_written_point
or :new.skill_score > v_skill_point
then raise_application_error(-20101, '배점이상의 성적은 입력하여 성적 등록 및 수정에 실패하였습니다.');
```

end if;

end;

요구사항번호	C-04	작성일	2022-05-19
요구사항명	시험 관리 및 성적 조회 기능	작성자	권민채
DML			
<pre>--과목 목록 출력 create or replace view subject_catalog_v as select     open_subject.seq as 과목번호,     curriculum.name as 과정명,     subject.subject_name as 개설과목,     open_subject.begin_date as "과목 시작일",     open_subject.end_date as "과목 종료일",     lectureroom.room_name as 강의실명,     teacher.name as 교사명 from open_curs     inner join open_subject on open_subject.open_curs_seq = open_curs.seq     inner join curriculum on open_curs.curs_seq = curriculum.seq     inner join subject on subject.seq = open_subject.subject_seq     inner join lectureroom on open_curs.room_seq = lectureroom.seq</pre>			

```

inner join teacher on open_subject.teacher_seq = teacher.seq
    order by open_subject.seq;

-- 특정 과목번호의 교육생 정보 및 성적 조회
create or replace procedure member_info_score_p(
    msubjectseq number := ''
)
is
    msubseq open_subject.seq%type;
    msubname subject.subject_name%type;
    mmemname member.name%type;
    mmemtel member.tel%type;
    mdrop student.drop_out%type;
    mwritten subject_score.written_score%type;
    mskill subject_score.skill_score%type;
    mattend subject_score.attendance_score%type;
    cursor vcursor is
select
    open_subject.seq as 과목번호,
    subject.subject_name as 과목명,
    member.name as 이름,
    member.tel as 전화번호,
    student.drop_out as "중도탈락 여부",
    subject_score.written_score as 필기점수,
    subject_score.skill_score as 실기점수,
    subject_score.attendance_score as 출결점수
from student
    inner join member on member.seq = student.member_seq

```

```

        inner join subject_score on subject_score.student_seq = student.seq
        inner join open_subject on open_subject.seq = subject_score.open_subject_seq
        inner join subject on subject.seq = open_subject.subject_seq
where open_subject.seq = msubjectseq;

begin
    open vcursor;
        loop
            fetch vcursor into msubseq,msubname,mmemname,mmemtel,mdrop,mwritten,mskill,mattend;
            exit when vcursor%notfound;
            dbms_output.put_line(' | 과목번호: ' || msubseq || ' | 과목명 : ' || msubname || ' | 이름: ' || mmemname || ' | 전화번호: ' || mmemtel || ' |
중도탈락여부: ' || mdrop || ' | 필기점수: ' || mwritten || '점' ||
                                ' | 실기점수: ' || mskill || '점' || ' | 출결점수 : ' || mattend || '점');

        end loop;
    close vcursor;
end;

--점수가 없는 학생들
create or replace view score_null_v as
select
    open_subject.seq as 과목번호,
    subject.subject_name as 과목명,
    member.name as 이름,
    student.seq as 교육생번호,
    student.drop_out as "중도탈락 여부",
    subject_score.written_score as 필기점수,
    subject_score.skill_score as 실기점수,
    subject_score.attendance_score as 출결점수
from student

```

```
inner join member on member.seq = student.member_seq
inner join subject_score on subject_score.student_seq = student.seq

inner join open_subject on open_subject.seq = subject_score.open_subject_seq

inner join subject on subject.seq = open_subject.subject_seq

where subject_score.written_score is null;

--시험점수 등록
create or replace procedure subject_score_p
(
    w_score in number
    s_score in number
    a_score in number
)
is
begin
    update subject_score
        set
            written_score = w_score
            skill_score = s_score
            attendance_score = a_score;
end;
```



요구사항명	출결관리 및 출결 조회	작성자	권민채
DML			
<pre>--전체 출결 조회 create or replace procedure attendance_check_p(      pseq in number,      popencurs in number,      pdatestart in date,      pdateend in date  is      vstudentnum student.seq%type;      출석일수 student.seq%type;      정상일수 student.seq%type;      지각일수 student.seq%type;      조퇴일수 student.seq%type;      지각조퇴 student.seq%type;      외출 student.seq%type;      결석 student.seq%type;      병가 student.seq%type;</pre>			

```

기타      student.seq%type;

cursor vcursor is

select

"학생번호",

count(decode("출석", '출석', 1)) as "출석일수",

count(decode("지각조퇴", '출석.', 1)) as "정상일수",

count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",

count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",

count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",

count(decode("지각조퇴", '외출', 1)) as "외출일수",

count(decode("지각조퇴", '결석', 1)) as "결석일수",

count(decode("출석", '병가', 1)) as "병가일수",

count(decode("출석", '기타', 1)) as "기타일수"

from

(select i.*,

출석||지각||조퇴 as "지각조퇴"

```

```
from attendance_info i where "학생번호" in (select seq from student where open_curs_seq = popencurs )
```

```
and "날짜" between pdatestart and pdateend)
```

```
where "학생번호"=pseq
```

```
group by "학생번호";
```

```
begin
```

```
open vcursor;
```

```
loop
```

```
fetch vcursor into vstudentnum, 출석일수, 정상일수, 지각일수, 조퇴일수, 지각조퇴,외출,결석,병가,기타;
```

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

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

```
dbms_output.put_line('교육생번호| ' ||vstudentnum|| ' 출석일수| ' ||출석일수|| ' 정상일수| ' ||정상일수|| ' 지각일수| ' ||지각일수 );
```

```
dbms_output.put_line('지각조퇴 | ' ||지각조퇴|| ' 외출일수| ' ||외출|| ' 결석일수| ' ||결석|| ' 조퇴일수| ' ||조퇴일수 );
```

```
dbms_output.put_line('병가일수 | ' ||병가|| ' 기타일수| ' ||기타 );
```

```

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

    end loop;

    close vcursor;

end;
```

요구사항번호	C-06	작성일	2022-05-19
요구사항명	상담일지 작성 및 조회	작성자	권민채
DML			
<pre> --상담일지 조회  create or replace view counsel_v  as  select      member.name as 교육생이름,      counsel_topic.counsel_topic as 상담주제,      counsel_date as 상담날짜,</pre>			

```
curriculum.name as 수강과정
```

```
from counsel
```

```
inner join counsel_topic on counsel_topic.seq = counsel_topic_seq
```

```
inner join student on student.seq = student_seq
```

```
inner join member on member.seq = student.member_seq
```

```
inner join teacher on teacher.seq = counsel.teacher_seq
```

```
inner join open_curs on open_curs.seq = student.open_curs_seq
```

```
inner join curriculum on curriculum.seq = open_curs.curs_seq
```

```
order by counsel_date;
```

```
-- 특정 교육생의 상담현황 조회
```

```
create or replace procedure counsel_member_p(
```

```
    cname varchar2:=",
```

```
)
```

```
is
```

```
    cmember member.name%type;
```

```
ctopic counsel_topic.counsel_topic%type;
```

```
cdate counsel.counsel_date%type;
```

```
cuname curriculum.name%type;
```

```
cursor vcursor is
```

```
select
```

```
member.name as 교육생이름,
```

```
counsel_topic.counsel_topic as 상담주제,
```

```
counsel_date as 상담날짜,
```

```
curriculum.name as 수강과정
```

```
from counsel
```

```
inner join counsel_topic on counsel_topic.seq = counsel_topic_seq
```

```
inner join student on student.seq = student_seq
```

```
inner join member on member.seq = student.member_seq
```

```
inner join teacher on teacher.seq = counsel.teacher_seq
```

```
inner join open_curs on open_curs.seq = student.open_curs_seq
```

```
inner join curriculum on curriculum.seq = open_curs.curs_seq
```

```
        where member.name = cname order by counsel_date;

begin

    open vcursor;

        loop

            fetch vcursor into cmember, ctopic, cdate, cuname;

            exit when vcursor%notfound;

            dbms_output.put_line('교육생이름: ' || cmember || ' | 상담주제: ' || ctopic || ' | 상담날짜: ' || cdate || ' | 상담주제: ' || cuname);

        end loop

    close vcursor;

end;

-- 특정 과정별 상담현황 조회

create or replace procedure counsel_curriculum_p(

    ccurri varchar2:=",

)

is

    cmember member.name%type;
```

```
ctopic counsel_topic.counsel_topic%type;
```

```
cdate counsel.counsel_date%type;
```

```
cuname curriculum.name%type;
```

```
cursor vcursor is
```

```
select
```

```
member.name as 교육생이름,
```

```
counsel_topic.counsel_topic as 상담주제,
```

```
counsel_date as 상담날짜,
```

```
curriculum.name as 수강과정
```

```
from counsel
```

```
inner join counsel_topic on counsel_topic.seq = counsel_topic_seq
```

```
inner join student on student.seq = student_seq
```

```
inner join member on member.seq = student.member_seq
```

```
inner join teacher on teacher.seq = counsel.teacher_seq
```

```
inner join open_curs on open_curs.seq = student.open_curs_seq
```

```
inner join curriculum on curriculum.seq = open_curs.curs_seq
```



```
where curriculum.name = ccurri order by counsel_date;

begin

    open vcursor;

        loop

            fetch vcursor into cmember, ctopic, cdate, cuname;

            exit when vcursor%notfound;

            dbms_output.put_line('교육생이름: ' || cmember || ' | 상담주제: ' || ctopic || ' | 상담날짜: ' || cdate || ' | 개설과정: ' || cuname);

        end loop

    close vcursor;

end;

--상담일지 등록

create or replace procedure insert_counsel_p(

    cseq in varchar2,

    ctopic in varchar2,

    cdate in date

)
)
```

```
is
```

```
begin
```

```
    insert into counsel values((select max(seq)+1 from counsel), cseq, ctopic, cdate);
```

```
end;
```

```
--상담일지 수정
```

```
create or replace procedure modify_counsel_p(
```

```
    cseq in varchar2,
```

```
    ctopic in varchar2,
```

```
    cdate in date
```

```
)
```

```
is
```

```
begin
```

```
    update counsel
```

```
    set
```

```
        student_seq = cseq
```

```
        counsel_topic_seq = ctopic
```

```

        counsel_date = cdate;

end;

--상담일지 삭제

create or replace procedure delete_counsel_p(

        ccseq in number

)

is

begin

        delete from counsel values where seq = ccseq;

end;

```

요구사항번호	C07	작성일	2022-05-19
요구사항명	과제 등록 및 제출 현황 조회	작성자	주혜원
DML			

## --과제 등록하기

create or replace view task\_list\_V(인서트확인용 뷰)

as

select\*from task\_list;

create or replace procedure add\_assgin\_P(

    psubject in varchar2,

    passgin in varchar2,

    deadline in date

)

is

begin

    insert into task\_list(seq,open\_curs\_seq,open\_subject\_seq,task\_description,summit\_limit) values

    ((select max(seq) + 1 from task\_list),12, psubject, passgin,deadline); --테이블에 원하는 값 넣기

    exception

    when others then

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

        dbms\_output.put\_line('                    잘못 입력하였습니다.');

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

end;

```

begin
  add_assgin_P(1, '자바 열심히해라', '2022-01-01'); --원하는 과목명,과제,제출기한 입력하기
end;

```

요구사항번호	D01	작성일	2022-05-19
요구사항명	교육생 계정	작성자	주혜원
DML			
<b>-로그인 된 교육생 계정 출력(기본정보)</b> <b>--교육생 계정으로 로그인 후 본인의 개인정보와 본인이 수강한 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실 출력</b> create or replace procedure student_info_P( ppw in member.ssn%type ) is vstudentNum student.seq%type; vstudentName member.name%type; vtel member.tel%type; vcursName curriculum.name%type; vcurTerm open_curs.current_term%type; vcursStart open_curs.begin_date%type; vcursFinish open_curs.end_date%type; vlectureRoom lectureroom.room_name%type;			

```

cursor vcursor is select
    s.seq as "교육생번호",
    m.name as "이름",
    m.tel as "전화번호",
    c.name as "과정이름",
    o.current_term as "과정기간",
    o.begin_date as "과정시작일",
    o.end_date as "과정종료일",
    l.room_name as "강의실"
from student s
inner join member m on m.seq = s.member_seq
inner join open_curs o on o.seq = s.open_curs_seq
inner join curriculum c on c.seq = o.curs_seq
inner join lectureroom l on l.seq = o.room_seq
where m.ssn = (select ssn from member where ssn = ppw);

```

begin

```
open vcursor;
```

```
loop
```

```
fetch vcursor into vstudentNum, vstudentName, vtel, vcursName, vcurTerm, vcursStart, vcursFinish, vlectureRoom;
```

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

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

```
dbms_output.put_line('교육생번호 : ' || vstudentNum|| '이름 : ' || vstudentName || '전화번호 : ' ||vtel|| '과정이름: ' ||
vcursName|| '강의실: ' ||vlectureRoom );
```

```
dbms_output.put_line('과정기간 : ' ||vcurTerm|| '과정시작일 : ' ||vcursStart|| '과정종료일 : ' ||vcursFinish );
```

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

exception --3 아이디와 비밀번호가 일치하지 않다면 예외처리
    when others then
        dbms_output.put_line('        일치하는 정보가 없습니다. 비밀번호를 다시 확인하세요!');
        dbms_output.put_line('');
        dbms_output.put_line('-----');
        -----');

end;

begin
    student_info_P('4256385');
end;
```

요구사항 명	D-02	작성일	2022-05-19
System	성적 조회	작성자	주혜원

## Query

### 1.제목출력(D-01) 쿼리와 같음

### 2.학생번호,과목번호, 과목명, 과목기간, 교재명, 교사명, 과목별 배점 정보(출결,필기,실기 배점), 과목별 성적정보(출결,필기,실기 점수)

```
select*from student_score_V--뷰
```

```
create or replace procedure student_score_P(  
    ppw in member.ssn%type  
)
```

```
is
```

```
vstudentNum student.seq%type;  
vsubjectNum  subject.seq%type;  
vjumin       member.ssn%type;  
vsubjectName subject.subject_name%type;  
vsubjectTerm subject.seq%type;  
vbookName    book.subject%type;  
vteacherName teacher.name%type;  
vattendPoint point.attend_point%type;  
vwrittenPoint point.written_point%type;  
vskillPoint  point.skill_point%type;  
vattendScore subject_score.written_score%type;  
vwrittenScore subject_score.skill_score%type;  
vskillScore  subject_score.attendance_score%type;  
cursor vcursor is (select*from student_score_V where 주민번호 = (select ssn from member where ssn =4256385 ));
```

```
begin
```



```

open vcursor;
loop
    fetch vcursor into vstudentNum, vsubjectNum , vjumin, vsubjectName, vsubjectTerm , vbookName , vteacherName , vattendPoint ,
vwrittenPoint ,vskillPoint , vattendScore, vwrittenScore ,vskillScore;
    exit when vcursor%notfound;
    dbms_output.put_line('-----');
    dbms_output.put_line('교육생번호 : ' || vstudentNum || ' 과목 이름 : ' || vsubjectName || ' 과목기간 : ' ||vsubjectTerm|| ' 교재이름: ' ||
vbookName || ' 교사명: ' || vteacherName );
    dbms_output.put_line('출석배점 : ' || vattendPoint|| '필기배점 : ' ||vwrittenPoint|| '실기배점 : ' ||vskillPoint );
    dbms_output.put_line('출석점수 : ' || vattendScore|| '필기점수 : ' || vwrittenScore|| '실기점수 : ' ||vskillScore );
    dbms_output.put_line('-----');
end loop;
close vcursor;

exception --3 아이디와 비밀번호가 일치하지 않다면 예외처리
when others then
    dbms_output.put_line(' 일치하는 정보가 없습니다. 비밀번호를 다시 확인하세요!');
    dbms_output.put_line('');
    dbms_output.put_line('-----');
end;

```

```
begin
    student_score_P('4256385');
end;
```

#### **--필기시험 목록 조회**

```
create or replace procedure written_test_P( ----원하는 과목명 입력
    psubject in written_test.open_subject_seq%type
)
is
    vsubjectNum    written_test.open_subject_seq%type; --과목번호
    vsubjectName   subject.subject_name%type; --과목이름
    vsubjectTerm   subject.seq%type; --과목기간
    vwrittenNum    written_test.seq%type; --필기시험번호
    vwrittenName   written_test.written_name%type; --필기시험이름
    vwrittenQuestion written_test_question.written_test_question%type; --필기시험 문제

    cursor vcursor is
    select
        wt.open_subject_seq as "과목번호",
        s.subject_name as "과목이름",
        os.end_date - os.begin_date as "과목기간",
        wt.seq as "필기시험번호",
        wt.written_name as "필기시험 이름",
```

```

q.written_test_question as "필기시험문제"
from written_test wt
inner join written_test_question q on wt.seq = q.written_test_seq
inner join open_subject os on os.seq = wt.open_subject_seq
inner join subject s on s.seq = os.subject_seq
where wt.open_subject_seq= psubject;

```

```

begin-----2

```

```

open vcursor;

```

```

LOOP

```

```

fetch vcursor into vsubjectNum, vsubjectName , vsubjectTerm, vwrittenNum, vwrittenName , vwrittenQuestion ;

```

```

exit when vcursor%notfound;

```

```

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

```

```

dbms_output.put_line('과목번호 : ' || vsubjectNum || ' 과목 이름 : ' || vsubjectName || ' 과목기간(일) : ' ||vsubjectTerm );

```

```

dbms_output.put_line('필기시험번호 : ' || vwrittenNum|| '필기시험 이름 : ' ||vwrittenName );

```

```

dbms_output.put_line('필기시험문제 : ' || vwrittenQuestion );

```

```

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

```

```

end loop;

```

```

close vcursor;

```

```

exception -----3
  when others then
    dbms_output.put_line('          권한이 없습니다. 로그인 정보를 다시 확인하세요');
    dbms_output.put_line('');
    dbms_output.put_line('          -----');

```

```

end; -----4

```

```

begin
  written_test_P(57); --필기시험 과목 번호 입력
end;

```

#### -- 실기시험 목록 조회

```

create or replace procedure skill_test_P( ----원하는 과목명 입력
  psubject in skill_test.open_subject_seq%type
)
is
  vsubjectNum skill_test.open_subject_seq%type; --과목번호
  vsubjectName subject.subject_name%type; --과목이름
  vsubjectTerm subject.seq%type; --과목기간

```

```
vskillNum      skill_test.seq%type; --실기시험번호
vskillName     skill_test.skill_name%type; --실기시험이름
vskillQuestion skill_test_question.skill_test_question%type; --실기시험 문제
```

```
cursor vcursor is
select
  st.open_subject_seq as "과목번호",
  s.subject_name as "과목이름",
  os.end_date - os.begin_date as "과목기간",
  st.seq as "실기시험번호",
  st.skill_name as "실기시험 이름",
  q.skill_test_question as "실기시험문제"
from skill_test st
  inner join skill_test_question q on st.seq = q.skill_test_seq
  inner join open_subject os      on os.seq = st.open_subject_seq
  inner join subject s           on s.seq = os.subject_seq
  where st.open_subject_seq= psubject;
```

```
begin-----2
```

```
  open vcursor;
```

```
  LOOP
```

```
    fetch vcursor into  vsubjectNum, vsubjectName , vsubjectTerm, vskillNum, vskillName , vskillQuestion ;
```

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

```
      dbms_output.put_line('-----
```

```
-----');
```

```
dbms_output.put_line('과목번호 : ' || vsubjectNum || ' 과목 이름 : ' || vsubjectName || ' 과목기간(일) : ' || vsubjectTerm );
dbms_output.put_line('필기시험번호 : ' || vskillNum || ' 필기시험 이름 : ' || vskillName );
dbms_output.put_line('필기시험문제 : ' || vskillQuestion );
dbms_output.put_line('-----');
-----');

end loop;
close vcursor;

exception -----3
when others then
dbms_output.put_line(' 권한이 없습니다. 로그인 정보를 다시 확인하세요');
dbms_output.put_line("");
dbms_output.put_line(' -----');
end;

begin
skill_test_P(57); --실기시험 과목 번호 입력
end;
```

요구사항 명	D-03	작성일	2022-05-19
System	출결 관리 및 출결 조회	작성자	주혜원
Query			
--출결사항 조회하기			

```

create or replace procedure attendance_check_P(

    pseq in number, --학생번호
    popenCurs in number, --특정개설과정번호
    pdateStart in Date, --과정시작날짜
    pdateEnd in Date --끝날짜
)
is
    vstudentNum student.seq%type; --학생번호
    출석일수 student.seq%type; --출석일수
    정상일수 student.seq%type; --정상일수
    지각일수 student.seq%type; --지각일수
    조퇴일수 student.seq%type; --조퇴일수
    지각조퇴 student.seq%type; --지각 및 조퇴일수
    외출 student.seq%type; --외출일수
    결석 student.seq%type; --결석일수
    병가 student.seq%type; --병가일수
    기타 student.seq%type; --기타일수
    cursor vcursor is
    select
        "학생번호",
        count(decode("출석", '출석', 1)) as "출석일수",
        count(decode("지각조퇴", '출석.', 1)) as "정상일수",
        count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",
        count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",
        count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",
        count(decode("지각조퇴", '외출', 1)) as "외출일수",

```

```

count(decode("지각조퇴", '결석', 1)) as "결석일수",
count(decode("출석", '병가', 1)) as "병가일수",
count(decode("출석", '기타', 1)) as "기타일수"
from
  (select i.*,
   출석||지각||조퇴 as "지각조퇴"
  from attendance_info i where "학생번호" in (select seq from student where open_curs_seq = popenCurs )
   and "날짜" between pdateStart and pdateEnd)
  where "학생번호"=pseq
  group by "학생번호";

begin
  open vcursor;
  loop
    fetch vcursor into vstudentNum, 출석일수, 정상일수, 지각일수, 조퇴일수, 지각조퇴 ,외출,결석,병가,기타;
    exit when vcursor%notfound;
    dbms_output.put_line('-----');
-----');
    dbms_output.put_line('교육생번호| ' ||vstudentNum|| '   출석일수| ' ||출석일수|| '   정상일수| ' ||정상일수|| '   지각일수| ' ||지각일수 );
    dbms_output.put_line('지각조퇴 | '   ||지각조퇴|| '   외출일수| ' ||외출|| '   결석일수| ' ||결석|| '   조퇴일수| ' ||조퇴일수 );
    dbms_output.put_line('병가일수 | ' ||병가|| '   기타일수| ' ||기타 );

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

```



```
end loop;
close vcursor;

exception --3 아이디와 비밀번호가 일치하지 않다면 예외처리
when others then
    dbms_output.put_line('          일치하는 정보가 없습니다. 비밀번호를 다시 확인하세요!');
    dbms_output.put_line('');
    dbms_output.put_line('-----');
    -----');

end;--실행

begin
    attendance_check_P(11, 11, '2022-02-02','2022-03-03');
end;
```

요구사항 명	D-04	작성일	2022-05-19
System	상담일지조회	작성자	주혜원
Query			
-상담일지조회			
-학생이름,상담날짜,상담주제,교사명			

```

create or replace procedure counsel_check_P(
    ppw in member.ssn%type
)
is
    vstudentName member.name%type; --학생이름
    vcounselDate  counsel.counsel_date%type; --상담날짜
    vcounselTopic counsel_topic.counsel_topic%type; --상담 주제
    vteacherName teacher.name%type; --교사명
    cursor vcursor is
    select
        m.name as 이름,
        c.counsel_date as 상담날짜,
        t.counsel_topic as 상담주제,
        t.name as 교사명
    from counsel c
        inner join counsel_topic t on t.seq = c.counsel_topic_seq
        inner join student s on s.seq = c.student_seq
        inner join member m on m.seq = s.member_seq
        inner join teacher t on t.seq = c.teacher_seq
    where m.ssn = (select ssn from member where ssn = ppw);

begin
    open vcursor;
    loop
        fetch vcursor into vstudentName, vcounselDate, vcounselTopic , vteacherName;
        exit when vcursor%notfound;
    end loop;
end;

```

```

        dbms_output.put_line('-----');
        -----');
        dbms_output.put_line('교육생이름 : ' || vstudentName|| ' 상담날짜 : ' || vcounselDate || ' 상담주제 : ' ||vcounselTopic|| ' 교사명: ' ||
vteacherName );
        dbms_output.put_line('-----');
        -----');

    end loop;
close vcursor;

exception --3 아이디와 비밀번호가 일치하지 않다면 예외처리
    when others then
        dbms_output.put_line('          일치하는 정보가 없습니다. 비밀번호를 다시 확인하세요!');
        dbms_output.put_line('');
        dbms_output.put_line('-----');
        -----');

end;

begin
    counsel_check_P(2660833);
end;

```

요구사항번호	D-05	작성일	2022-05-19
요구사항명	과제 조회 및 제출	작성자	김성태
DML			
<pre>--자신이 제출한 과제 목록 조회 create or replace procedure submit_P(   p_member_ssn in number,   p_result out sys_refcursor ) is begin   open p_result   for   select     s.seq,     t.task_description as 과제설명,     m.name as 제출자,     s.submit_date as 제출날짜,     t.submit_limit as 제출기한   from submit s</pre>			

```

inner join task_list t
  on s.task_seq = t.seq
  inner join student stu
    on s.student_seq = stu.seq
    inner join member m
      on stu.member_seq = m.seq
      where m.ssn = p_member_ssn;
end submit_P;

```

요구사항번호	D-06	작성일	2022-05-20
요구사항명	주요제트 제목 검색기능	작성자	임종운
DML			
CREATE OR REPLACE TYPE study_obj AS OBJECT( topic varchar2(50), study_name varchar2(50), study_description varchar2(150), name varchar(20), email varchar2(50),			

```
tel varchar2(15),

begin_date date,

end_date date,

complement varchar2(5)

);

/

CREATE OR REPLACE TYPE study_tables AS TABLE OF study_obj;

/

CREATE OR REPLACE FUNCTION study_list_F

(

    input_topic_or_description varchar2

) RETURN study_tables

IS

    v_rtn study_tables;
```

```
BEGIN
```

```
SELECT
```

```
study_obj(rs.topic, rs.study_name, rs.study_description, rs.name, rs.email, rs.tel, rs.begin_date, rs.end_date, rs.completement)
```

```
BULK COLLECT INTO v_rtn
```

```
FROM (SELECT
```

```
st.topic, sr.study_name, sr.study_description, m.name, sr.email, m.tel, sr.begin_date, sr.end_date, sr.completement
```

```
FROM study_recruit sr
```

```
INNER JOIN study_topic st ON sr.study_topic_seq = st.seq
```

```
INNER JOIN student s ON s.seq = sr.student_seq
```

```
INNER JOIN member m ON s.member_seq = m.seq
```

```
WHERE
```

```
sr.study_name LIKE '%'||input_topic_or_description||'%' ) rs;
```

```
RETURN v_rtn;
```

```

END study_list_F;

/

DROP TYPE study_tables;

DROP TYPE study_obj;

DROP FUNCTION study_list_F;


--스터디 제목으로 검색 -- 공백입력시 전체 스터디 모집내역 출력됨.

SELECT * FROM TABLE(study_list_F('웹'));

```

요구사항번호	D-07	작성일	2022-05-19
요구사항명	취업 공고 조회	작성자	권민채
DML			
<pre> --전체 취업공고 조회 create or replace view job_list_v as( select     city.city as "지역",     job_announce.company as "회사명",     job_field.field_name as "분야", </pre>			



```

        contract_type.contract_type as "직위",
        job_announce.people as "모집인원",
        job_announce.begin_date as "모집 시작일",
        job_announce.end_date as "모집 종료일"
from job_announce
inner join city on job_announce.city_seq = city.seq
inner join job_field on job_announce.field_seq= job_field.seq
inner join contract_type on job_announce.contract_seq = contract_type.seq);

```

--기간별 취업공고 조회

```

create or replace procedure job_period_p (
    start_period varchar2 := "",
    end_period varchar2 := ""
)
is
    acompany job_announce.company%type;
    ccity city.city%type;
   ffield_name job_field.field_name%type;
    ctcontract_type contract_type.contract_type%type;
    apeople job_announce.people%type;
    abegin_date job_announce.begin_date%type;
    aend_date job_announce.end_date%type;
    cursor vcursor is
select
    a.company as 회사명,
    c.city as 지역,
    f.field_name as 분야,
    ct.contract_type as 계약형태,

```

```

a.people as 모집인원,
a.begin_date as 모집시작날짜,
a.end_date as 모집종료날짜
from job_announce a
inner join city c on c.seq = a.city_seq
inner join job_field f on f.seq = a.field_seq
inner join contract_type ct on ct.seq = a.contract_seq
where a.end_date between start_period and end_period order by 모집시작날짜;
begin
open vcursor;
loop
fetch vcursor into acompany, ccity, ffield_name, ctcontract_type, apeople, abegin_date, aend_date;
exit when vcursor%notfound;
dbms_output.put_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield_name || ' | 계약형태: ' || ctcontract_type || ' | 모집인원: ' || apeople || '명' || ' |
모집시작날짜: ' ||
abegin_date || ' | 모집종료날짜: ' || aend_date);

end loop;
close vcursor;
end;

--지역별 취업공고 조회

create or replace procedure city_job_check_P (

pcity varchar2 := ''

)

```

is

acompany job\_announce.company%type;

ccity city.city%type;

ffield\_name job\_field.field\_name%type;

ctcontract\_type contract\_type.contract\_type%type;

apeople job\_announce.people%type;

abegin\_date job\_announce.begin\_date%type;

aend\_date job\_announce.end\_date%type;

cursor vcursor is

select

a.company as 회사명,

c.city as 지역,

f.field\_name as 분야,

ct.contract\_type as 계약형태,

a.people as 모집인원,

a.begin\_date as 모집시작날짜,

```
a.end_date as 모집종료날짜
```

```
from job_announce a
```

```
inner join city c on c.seq = a.city_seq
```

```
inner join job_field f on f.seq = a.field_seq
```

```
inner join contract_type ct on ct.seq = a.contract_seq
```

```
where c.city = pcity order by 모집시작날짜;
```

```
begin
```

```
open vcursor;
```

```
loop
```

```
fetch vcursor into acompany, ccity, ffield_name, ctcontract_type, apeople, abegin_date, aend_date;
```

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

```
        dbms_output.put_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield_name || ' | 계약형태: ' || ctcontract_type || ' | 모집인원: ' || apeople || '명' ||
' | 모집시작날짜: ' ||

                                abegin_date || ' | 모집종료날짜: ' || aend_date);

    end loop;

    close vcursor;

end;

--분야별 취업공고 조회

create or replace procedure field_job_check_P (

    pjob_field varchar2 := ''

)

is

    acompany job_announce.company%type;

    ccity city.city%type;
```

```
ffield_name job_field.field_name%type;
```

```
ctcontract_type contract_type.contract_type%type;
```

```
apeople job_announce.people%type;
```

```
abegin_date job_announce.begin_date%type;
```

```
aend_date job_announce.end_date%type;
```

```
cursor vcursor is
```

```
select
```

```
a.company as 회사명,
```

```
c.city as 지역,
```

```
f.field_name as 분야,
```

```
ct.contract_type as 계약형태,
```

```
a.people as 모집인원,
```

```
a.begin_date as 모집시작날짜,
```

```
a.end_date as 모집종료날짜
```

```
from job_announce a
```

```
inner join city c on c.seq = a.city_seq
```

```
inner join job_field f on f.seq = a.field_seq

inner join contract_type ct on ct.seq = a.contract_seq

where f.field_name = pjob_field order by 모집시작날짜;

begin

open vcursor;

loop

    fetch vcursor into acompany, ccity, ffield_name, ctcontract_type, apeople, abegin_date, aend_date;

    exit when vcursor%notfound;

    dbms_output.put_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield_name || ' | 계약형태: ' || ctcontract_type || ' | 모집인원: ' || apeople || '명' ||
' | 모집시작날짜: ' ||

                                abegin_date || ' | 모집종료날짜: ' || aend_date);

end loop;

close vcursor;

end;
```