**DB Object 스크립트(PL-SQL)**

| 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  --프로지서명 : 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  --프로지서명 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, mssn, 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;  **--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  --프로지서명 : 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 lectureroom.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('교육생번호: ' || stseq || ' | 교육생이름: ' || 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-09 | 작성일 | 2022-05-19 |
| --- | --- | --- | --- |
| 요구사항명 | 취업 공고 관리 및 조회 | 작성자 | 강지윤 |
| DML | | | |
| ▶ where절 < > 위치에는 <>안에 있는 내용에 해당하는 데이터 입력  **--1. 전체 취업 공고조회(회사명, 지역, 분야, 계약형태, 모집인원, 모집시작날짜, 모집종료날짜)**  --뷰이름 : job\_announce\_V  create or replace view job\_announce\_V  as  (  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  --프로지서명 : 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  --프로지서명 : 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  --프로지서명 : city\_job\_check\_P  -- city\_job\_check\_P (<’지역명’’>)으로 호출.  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('-------------------------------------------------------------------------------------------------------------');    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; | | | |