**업무 스크립트**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-1. 기초 정보 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-1.1) 과정 정보 관리** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 과정 정보 조회**  -- 과정 정보 저장 프로시저  create or replace procedure procSetCourse(  presult out sys\_refcursor  )  is  begin  dbms\_output.put\_line(chr(10) || '[과정 정보 조회]');  dbms\_output.put\_line('--------------------------------------------------');  dbms\_output.put\_line('|No.|' || lpad('과정명', 43) || lpad('|', 39) || ' 기간 |');  dbms\_output.put\_line('--------------------------------------------------');  open presult  for select \* from tblCourse order by course\_seq;  end procSetCourse;  -- 과정 정보 조회 프로시저  create or replace procedure procGetCourse  is  vresult sys\_refcursor;  vrow tblCourse%rowtype;  begin  procSetCourse(vresult);  loop  fetch vresult into vrow;  exit when vresult%notfound;  dbms\_output.put\_line('| ' || to\_char(vrow.course\_seq, '00') || '|' || vrow.course\_name || chr(9)  || '|' || to\_char(vrow.course\_period, '0.0') || '개월|');  dbms\_output.put\_line('--------------------------------------------------');  end loop;  end procGetCourse;  -- 과정 정보 조회  begin  procGetCourse;  end;  **-- 2. 과정 정보 등록**  -- 과정 기간 확인 함수  create or replace function fnIsValidPeriod (  pperiod number  ) return varchar2  is  vcheck varchar2(1);  begin  if pperiod in (5.5, 6, 7) then  vcheck := 'Y';  else  vcheck := 'N';  end if;  return vcheck;  end fnIsValidPeriod;  -- 과정 등록 프로시저  create or replace procedure procAddCourse (  pname varchar2,  pperiod number  )  is  begin  dbms\_output.put\_line(chr(10) || '[과정 정보 등록]' || chr(10)  || 'No.' || course\_seq.nextVal || ' ' || pname  || '(' || to\_char(pperiod, '0.0') || '개월)' || chr(10));  if fnIsValidPeriod(pperiod) = 'N' then  dbms\_output.put\_line('☞실패; 기간 부적합');  else  insert into tblCourse (course\_seq, course\_name, course\_period)  values (course\_seq.currVal, pname, pperiod);  dbms\_output.put\_line('☞성공!');  end if;  exception  when others then  dbms\_output.put\_line('☞실패; ' || sqlerrm);  end procAddCourse;  -- 과정 정보 등록  begin  procAddCourse(과정명, 기간);  end;  **-- 3. 과정 정보 수정**  -- 과정 수정 트리거  create or replace trigger trgUpdateCourse  after  update on tblCourse  for each row  begin  dbms\_output.put\_line('수정 전: No.' || :old.course\_seq || ' '  || :old.course\_name || '('  || to\_char(:old.course\_period, '0.0') || '개월)');  dbms\_output.put\_line('수정 후: No.' || :new.course\_seq || ' '  || :new.course\_name || '('  || to\_char(:new.course\_period, '0.0') || '개월)');  end;  -- 과정 수정 프로시저  create or replace procedure procUpdateCourse (  pseq number,  pname varchar2,  pperiod number  )  is  vinfo varchar2(1000);  begin  dbms\_output.put\_line(chr(10) || '[과정 정보 수정]');    vinfo := 'No.' || pseq || ' ' || pname  || '(' || to\_char(pperiod, '0.0') || '개월)' || chr(10) || chr(10);  if fnIsValidPeriod(pperiod) = 'N' then  dbms\_output.put\_line(vinfo || '☞실패; 기간 부적합');  else  dbms\_output.put\_line(vinfo || '☞성공!');  update tblCourse set course\_name = pname,  course\_period = pperiod  where course\_seq = pseq;  end if;    exception  when others then  dbms\_output.put\_line(vinfo || '☞실패; ' || sqlerrm);  end procUpdateCourse;  -- 과정 수정  begin  procUpdateCourse(번호, 과정명, 기간);  end;  **-- 4. 과정 정보 삭제**  -- 과정 삭제 프로시저  create or replace procedure procDeleteCourse (  pseq number  )  is  vc tblCourse%rowtype;  begin  select \*  into vc from tblCourse  where course\_seq = pseq;    dbms\_output.put\_line(chr(10) || '[과정 정보 삭제]' || chr(10)  || 'No.' || pseq || ' ' || vc.course\_name  || '(' || vc.course\_period || '개월)' || chr(10));    delete from tblCourse  where course\_seq = pseq;    dbms\_output.put\_line('☞성공!');  exception  when others then  dbms\_output.put\_line('☞실패; ' || sqlerrm);  end procDeleteCourse;  -- 과정 삭제  begin  procDeleteCourse(번호);  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-1. 기초 정보 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-1.2) 과목 정보 관리** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 과목 정보 조회**  select  subject\_seq as "번호",  subject\_name as "과정명",  to\_char(subject\_period, '0.0') || '개월' as "기간"  from tblSubject  order by subject\_seq;  **-- 2. 과목 정보 등록**  begin  procAddSubject(과목명, 기간);  end;  -- 과목 등록 프로시저  create or replace procedure procAddSubject (  pname varchar2,  pperiod number  )  is  begin  insert into tblSubject (subject\_seq, subject\_name, subject\_period)  values (subject\_seq.nextVal, pname, pperiod);    dbms\_output.put\_line('과목 등록 성공: No.' || subject\_seq.currVal || ' '  || pname || '(' || to\_char(pperiod, '0.0') || '개월)');  exception  when others then  dbms\_output.put\_line('과목 등록 실패: No.' || subject\_seq.currVal || ' '  || pname || '(' || to\_char(pperiod, '0.0') || '개월)');  dbms\_output.put\_line(sqlerrm);  end procAddSubject;  **-- 3. 과목 정보 수정**  begin  procUpdateSubject(번호, 과목명, 기간);  end;  -- 과목 수정 프로시저  create or replace procedure procUpdateSubject (  pseq number,  pname varchar2,  pperiod number  )  is  begin  update tblSubject set subject\_name = pname,  subject\_period = pperiod  where subject\_seq = pseq;    dbms\_output.put\_line('과목 변경 성공: No.' || pseq || ' '  || pname || '(' || to\_char(pperiod, '0.0') || '개월)');  exception  when others then  dbms\_output.put\_line('과목 변경 실패: No.' || pseq || ' '  || pname || '(' || to\_char(pperiod, '0.0') || '개월)');  dbms\_output.put\_line(sqlerrm);  end procUpdateSubject;  **-- 4. 과목 정보 삭제**  declare  vname tblSubject.subject\_name%type;  vperiod number;  begin  procDeleteSubject(번호, vname, vperiod);  end;  -- 과목 삭제 프로시저  create or replace procedure procDeleteSubject (  pseq number,  pname out varchar2,  pperiod out number  )  is  begin  select subject\_name, subject\_period  into pname, pperiod from tblSubject  where subject\_seq = pseq;  delete from tblSubject  where subject\_seq = pseq;    dbms\_output.put\_line('과목 삭제 성공: No.' || pseq || ' '  || pname || '(' || to\_char(pperiod, '0.0') || '개월)');  exception  when others then  dbms\_output.put\_line('과목 삭제 실패: No.' || pseq || ' '  || pname || '(' ||to\_char(pperiod, '0.0') || '개월)');  dbms\_output.put\_line(sqlerrm);  end procDeleteSubject; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-1. 기초 정보 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-1.3) 강의실 정보 관리** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 강의실 정보 조회**  select  room\_seq as "번호",  room\_name as "이름",  room\_capacity || '명' as "수용 인원"  from tblRoom  order by room\_seq;  **-- 2. 강의실 정보 등록**  begin  procAddRoom(강의실명, 인원);  end;  -- 강의실 등록 프로시저  create or replace procedure procAddRoom (  pname varchar2,  pcapacity number  )  is  begin  insert into tblRoom (room\_seq, room\_name, room\_capacity)  values (room\_seq.nextval , pname, pcapacity);    dbms\_output.put\_line('강의실 등록 성공: No.' || room\_seq.currVal || ' '  || pname || '(수용: ' || pcapacity || '명)');  exception  when others then  dbms\_output.put\_line('강의실 등록 실패: No.' || room\_seq.currVal || ' '  || pname || '(수용: ' || pcapacity || '명)');  dbms\_output.put\_line(sqlerrm);  end procAddRoom;  **-- 3. 강의실 정보 수정**  begin  procUpdateRoom(번호, 강의실명, 인원);  end;  -- 강의실 수정 프로시저  create or replace procedure procUpdateRoom (  pseq number,  pname varchar2,  pcapacity number  )  is  begin  update tblRoom set room\_name = pname,  room\_capacity = pcapacity  where room\_seq = pseq;    dbms\_output.put\_line('강의실 변경 성공: No.' || pseq || ' '  || pname || '(수용: ' || pcapacity || '명)');  exception  when others then  dbms\_output.put\_line('강의실 변경 실패: No.' || pseq || ' '  || pname || '(수용: ' || pcapacity || '명)');  dbms\_output.put\_line(sqlerrm);  end procUpdateRoom;  **-- 4. 강의실 정보 삭제**  declare  vname tblRoom.room\_name%type;  vcapacity number;  begin  procDeleteRoom(번호, vname, vcapacity);  end;  -- 강의실 삭제 프로시저  create or replace procedure procDeleteRoom (  pseq number,  pname out varchar2,  pcapacity out number  )  is  begin  select room\_name, room\_capacity  into pname, pcapacity from tblRoom  where room\_seq = pseq;  delete from tblRoom  where room\_seq = pseq;    dbms\_output.put\_line('강의실 삭제 성공: No.' || pseq || ' '  || pname || '(수용: ' || pcapacity || '명)');  exception  when others then  dbms\_output.put\_line('강의실 삭제 실패: No.' || pseq || ' '  || pname || '(수용: ' || pcapacity || '명)');  dbms\_output.put\_line(sqlerrm);  end procDeleteRoom; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-1. 기초 정보 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-1.4) 교재 정보 관리** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 교재 정보 조회**  select  b.book\_seq as "번호",  b.book\_name as "이름",  b.book\_publisher as "출판사",  b.book\_price || '원 'as "가격",  s.subject\_name as "과목명"  from tblBook b inner join tblSubject s  on (b.subject\_seq = s.subject\_seq)  order by b.book\_seq;  **-- 2. 교재 정보 등록**  begin  procAddBook(교재명, 출판사, 가격, 과목\_번호);  end;  -- 교재 등록 프로시저  create or replace procedure procAddBook (  pname varchar2,  ppublisher varchar2,  pprice number,  psseq number  )  is  begin  insert into tblBook (book\_seq, book\_name, book\_publisher, book\_price, subject\_seq)  values (book\_seq.nextVal, pname, ppublisher, pprice, psseq);    dbms\_output.put\_line('교재 등록 성공: No.' || book\_seq.currVal || ' ' || pname  || '(출판사: ' || ppublisher  || ', 가격: ' || pprice || '원)');  exception  when others then  dbms\_output.put\_line('교재 등록 성공: No.' || book\_seq.currVal || ' ' || pname  || '(출판사: ' || ppublisher  || ', 가격: ' || pprice || '원)');  dbms\_output.put\_line(sqlerrm);  end procAddBook;  -- 3. 교재 정보 수정  begin  procUpdateBook(번호, 교재명, 출판사, 가격, 과목\_번호);  end;  -- 교재 수정 프로시저  create or replace procedure procUpdateBook (  pseq number,  pname varchar2,  ppublisher varchar2,  pprice number,  psseq number  )  is  begin  update tblBook set book\_name = pname,  book\_publisher = ppublisher,  book\_price = pprice,  subject\_seq = psseq  where book\_seq = pseq;    dbms\_output.put\_line('교재 변경 성공: No.' || pseq || ' ' || pname  || '(출판사: ' || ppublisher  || ', 가격: ' || pprice || '원)');  exception  when others then  dbms\_output.put\_line('교재 변경 실패: No.' || pseq || ' ' || pname  || '(출판사: ' || ppublisher  || ', 가격: ' || pprice || '원)');  dbms\_output.put\_line(sqlerrm);  end procUpdateBook;  **-- 4. 교재 정보 삭제**  declare  vname tblBook.book\_name%type;  vpublisher tblBook.book\_publisher%type;  vprice number;  begin  procDeleteBook(번호, vname, vpublisher, vprice);  end;  -- 교재 삭제 프로시저  create or replace procedure procDeleteBook (  pseq number,  pname out varchar2,  ppublisher out varchar2,  pprice out number  )  is  begin  select book\_name, book\_publisher, book\_price  into pname, ppublisher, pprice from tblBook  where book\_seq = pseq;  delete from tblBook  where book\_seq = pseq;    dbms\_output.put\_line('교재 삭제 성공: No.' || pseq || ' ' || pname  || '(출판사: ' || ppublisher  || ', 가격: ' || pprice || '원)');  exception  when others then  dbms\_output.put\_line('교재 삭제 실패: No.' || pseq || ' ' || pname  || '(출판사: ' || ppublisher  || ', 가격: ' || pprice || '원)');  dbms\_output.put\_line(sqlerrm);  end procDeleteBook; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-2. 교사 계정 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-2.1) 교사 정보 관리** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 교사 정보 조회**  select  teacher\_seq as "번호",  teacher\_name as "이름",  teacher\_ssn as "주민등록번호",  teacher\_tel as "연락처"  from tblTeacher  order by teacher\_seq;  **-- 2. 교사 정보 등록**  begin  procAddTeacher(이름, 주민등록번호, 연락처);  end;  -- 교사 등록 프로시저  create or replace procedure procAddTeacher (  pname varchar2,  pssn varchar2,  ptel varchar2  )  is  begin  insert into tblTeacher (teacher\_seq, teacher\_name, teacher\_ssn, teacher\_tel)  values (teacher\_seq.nextVal, pname, pssn, ptel);    dbms\_output.put\_line('교사 등록 성공: No.' || teacher\_seq.currtVal || ' ' || pname  || '(주민등록번호: ' || substr(pssn, 1, 8) || '\*\*\*\*\*\*'  || ', tel: ' || ptel || ')');  exception  when others then  dbms\_output.put\_line('교사 등록 실패: No.' || teacher\_seq.currVal || ' ' || pname  || '(주민등록번호: ' || substr(pssn, 1, 8) || '\*\*\*\*\*\*'  || ', tel: ' || ptel || ')');  dbms\_output.put\_line(sqlerrm);  end procAddTeacher;  **-- 3. 교사 정보 수정**  begin  procUpdateTeacher(번호, 이름, 주민등록번호, 연락처);  end;  -- 교사 수정 프로시저  create or replace procedure procUpdateTeacher (  pseq number,  pname varchar2,  pssn varchar2,  ptel varchar2  )  is  begin  update tblTeacher set teacher\_name = pname,  teacher\_ssn = pssn,  teacher\_tel = ptel  where teacher\_seq = pseq;    dbms\_output.put\_line('교사 수정 성공: No.' || pseq || ' ' || pname  || '(주민등록번호: ' || substr(pssn, 1, 8) || '\*\*\*\*\*\*'  || ', tel: ' || ptel || ')');  exception  when others then  dbms\_output.put\_line('교사 수정 실패: No.' || pseq || ' ' || pname  || '(주민등록번호: ' || substr(pssn, 1, 8) || '\*\*\*\*\*\*'  || ', tel: ' || ptel || ')');  dbms\_output.put\_line(sqlerrm);  end procUpdateTeacher;  end;  **-- 4. 교사 정보 삭제**  declare  vname tblTeacher.teacher\_name%type;  vssn tblTeacher.teacher\_ssn%type;  vtel tblTeacher.teacher\_tel%type;  begin  procDeleteTeacher(번호, vname, vssn, vtel);  end;  -- 교사 삭제 프로시저  create or replace procedure procDeleteTeacher (  pseq number,  pname out varchar2,  pssn out varchar2,  ptel out varchar2  )  is  begin  select teacher\_name, teacher\_ssn, teacher\_tel  into pname, pssn, ptel from tblTeacher  where teacher\_seq = pseq;  delete from tblTeacher  where teacher\_seq = pseq;    dbms\_output.put\_line('교사 삭제 성공: No.' || pseq || ' ' || pname  || '(주민등록번호: ' || substr(pssn, 1, 8) || '\*\*\*\*\*\*'  || ', tel: ' || ptel || ')');  exception  when others then  dbms\_output.put\_line('교사 삭제 실패: No.' || pseq || ' ' || pname  || '(주민등록번호: ' || substr(pssn, 1, 8) || '\*\*\*\*\*\*'  || ', tel: ' || ptel || ')');  dbms\_output.put\_line(sqlerrm);  end procDeleteTeacher; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-2. 교사 계정 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-2.2) 교사별 강의 가능 과목 관리** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 가능 과목 조회**  select  st.st\_seq as "번호",  t.teacher\_name as "교사명",  s.subject\_name as "과목명"  from tblSubjectTeacher st inner join tblTeacher t  on (st.teacher\_seq = t.teacher\_seq) inner join tblSubject s  on (st.subject\_seq = s.subject\_seq);    **-- 2. 가능 과목 등록**  declare  vsname tblSubject.subject\_name%type;  vtname tblTeacher.teacher\_name%type;  begin  procAddSubjectTeacher(과목\_번호, 강사\_번호, vsname, vtname);  end;  -- 교사 가능 과목 등록 프로시저  create or replace procedure procAddSubjectTeacher (  psseq number,  ptseq number,  psname out varchar2,  ptname out varchar2  )  is  begin  select subject\_name  into psname from tblSubject  where subject\_seq = psseq;  select teacher\_name  into ptname from tblTeacher  where teacher\_seq = ptseq;  insert into tblSubjectTeacher (st\_seq, subject\_seq, teacher\_seq)  values (st\_seq.nextVal, psseq, ptseq);    dbms\_output.put\_line('강의 가능 교사 등록 성공: No.' || st\_seq.currVal || ' '  || ptname || ' - ' || psname);  exception  when others then  dbms\_output.put\_line('강의 가능 교사 등록 실패: No.' || st\_seq.currVal || ' '  || ptname || ' - ' || psname);  dbms\_output.put\_line(sqlerrm);  end procAddSubjectTeacher;  **-- 3. 가능 과목 수정**  declare  vsname tblSubject.subject\_name%type;  vtname tblTeacher.teacher\_name%type;  begin  procUpdateSubjectTeacher(번호, 과목\_번호, 강사\_번호, vsname, vtname);  end;  -- 교사 가능 과목 수정 프로시저  create or replace procedure procUpdateSubjectTeacher (  pseq number,  psseq number,  ptseq number,  psname out varchar2,  ptname out varchar2  )  is  begin  select subject\_name  into psname from tblSubject  where subject\_seq = psseq;  select teacher\_name  into ptname from tblTeacher  where teacher\_seq = ptseq;  update tblSubjectTeacher set subject\_seq = psseq,  teacher\_seq = ptseq  where st\_seq = pseq;    dbms\_output.put\_line('강의 가능 교사 수정 성공: No.' || pseq || ' '  || ptname || ' - ' || psname);  exception  when others then  dbms\_output.put\_line('강의 가능 교사 수정 실패: No.' || pseq || ' '  || ptname || ' - ' || psname);  dbms\_output.put\_line(sqlerrm);  end procUpdateSubjectTeacher;  **-- 4. 가능 과목 삭제**  declare  vsname tblSubject.subject\_name%type;  vtname tblTeacher.teacher\_name%type;  begin  procDeleteSubjectTeacher(번호, 과목\_번호, 강사\_번호, vsname, vtname);  end;  -- 교사 가능 과목 삭제 프로시저  create or replace procedure procDeleteSubjectTeacher (  pseq number,  psseq number,  ptseq number,  psname out varchar2,  ptname out varchar2  )  is  begin  select subject\_name  into psname from tblSubject  where subject\_seq = psseq;  select teacher\_name  into ptname from tblTeacher  where teacher\_seq = ptseq;  delete from tblSubjectTeacher  where st\_seq = pseq;    dbms\_output.put\_line('강의 가능 교사 삭제 성공: No.' || pseq || ' '  || ptname || ' - ' || psname);  exception  when others then  dbms\_output.put\_line('강의 가능 교사 삭제 실패: No.' || pseq || ' '  || ptname || ' - ' || psname);  dbms\_output.put\_line(sqlerrm);  end procDeleteSubjectTeacher; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-2. 교사 계정 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-2.3) 특정 교사 선택시 상세 정보 조회** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 상세 정보 조회**  select  teacher\_name as "교사명",  course\_seq as "과정명",  oc\_startdate as "과정 시작일",  oc\_enddate as "과정 종료일",  subject\_name as "과목명",  os\_startdate as "과목 시작일",  os\_enddate as "과목 종료일",  book\_name as "교재",  room\_name as "강의실"  from vwOpenCourseInfo  where teacher\_seq = 교사\_번호; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-3. 개설 과정 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-3.1) 개설 과정 정보 관리** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 개설 과정 조회**  -- 개설 과정 정보 저장 프로시저  create or replace procedure procSetOpenCourse(  presult out sys\_refcursor  )  is  begin  dbms\_output.put\_line(chr(10) || '[개설 과정 조회]');  dbms\_output.put\_line('--------------------------------------------------------------------');  dbms\_output.put\_line('|No.|' || lpad('과정명', 43) || lpad('|', 39)  || lpad('기간', 17) || lpad('|', 13)  || ' 강의실|수용|');  dbms\_output.put\_line('--------------------------------------------------------------------');  open presult  for select \* from vwOpenCourse  order by oc\_seq;  end procSetOpenCourse;  -- 개설 과정 조회 프로시저  create or replace procedure procGetOpenCourse  is  vresult sys\_refcursor;  vrow vwOpenCourse%rowtype;  begin  procSetOpenCourse(vresult);  loop  fetch vresult into vrow;  exit when vresult%notfound;  dbms\_output.put\_line('| ' || to\_char(vrow.oc\_seq, '00') || '|' || vrow.course\_name || chr(9)  || '|' || vrow.oc\_startdate || '~' || vrow.oc\_enddate  || '|' || vrow.room\_name || '|' || vrow.room\_capacity || '명|');  dbms\_output.put\_line('--------------------------------------------------------------------');  end loop;  end procGetOpenCourse;  -- 개설 과정 조회  begin  procGetOpenCourse;  end;  **-- 2. 개설 과정 등록**  -- 개설 가능한 날짜 함수  create or replace function fnIsValidDate (  pdate date  ) return varchar2  is  vcnt number;  vcheck varchar2(1);  begin  select count(\*)  into vcnt  from tblHoliday  where holiday\_date = pdate;    if to\_char(pdate, 'd') in ('1', '7')  or pdate < sysdate  or vcnt > 0 then  vcheck := 'N';  else  vcheck := 'Y';  end if;    return vcheck;  end fnIsValidDate;  -- 개설 가능한 강의실 함수  create or replace function fnIsValidRoom (  pseq number,  pstartdate date  ) return varchar2  is  vdate date;  vcheck varchar2(1);  begin  select max(oc\_enddate)  into vdate  from vwOpenCourse  where room\_seq = pseq;    if vdate > pstartdate then  vcheck := 'N';  else  vcheck := 'Y';  end if;    return vcheck;  end fnIsValidRoom;  -- 종료일 계산 함수  create or replace function fnGetEnddate (  pdate date,  pseq number  ) return date  is  vperiod number;  begin  select course\_period  into vperiod from tblCourse  where course\_seq = pseq;  return add\_months(pdate, vperiod);  end fnGetEnddate;  -- 개설 과정 등록 프로시저  create or replace procedure procAddOpenCourse (  pseq number,  prseq number,  pstartdate date  )  is  venddate date;  begin  venddate := fnIsEnddate(pstartdate, pseq);    dbms\_output.put\_line(chr(10) || '[개설 과정 등록]' || chr(10)  || 'No.' || oc\_seq.nextVal || ' ' || fnGetCourseName(pseq)  || '(' || pstartdate || ' ~ ' || venddate  || ', ' || fnGetRoomName(prseq) || ')' || chr(10));  if fnIsValidDate(pstartdate) = 'N' then  dbms\_output.put\_line('☞실패; 시작일 부적합');  elsif fnIsValidDate(venddate) = 'N' then  dbms\_output.put\_line('☞실패; 종료일 부적합');  elsif fnIsValidRoom(prseq, pstartdate) = 'N' then  dbms\_output.put\_line('☞실패; 강의실 부적합');  else  insert into tblOpenCourse (oc\_seq, course\_seq, oc\_startdate, oc\_enddate, room\_seq)  values(oc\_seq.currval, pseq, pstartdate, venddate, prseq);    dbms\_output.put\_line('☞성공!');  end if;    exception  when others then  dbms\_output.put\_line('☞실패; ' || sqlerrm);  end procAddOpenCourse;  -- 개설 과정 등록  begin  procAddOpenCourse(과정\_번호, 강의실\_번호, 시작일);  end;  **-- 3. 개설 과정 수정**  -- 개설 과정 수정 트리거  create or replace trigger trgUpdateOpenCourse  after  update on tblOpenCourse  for each row  begin  dbms\_output.put\_line('수정 전: No.' || :old.oc\_seq || ' '  || fnGetCourseName(:old.course\_seq)  || '(' || :old.oc\_startdate || ' ~ ' || :old.oc\_enddate  || ', ' || fnGetRoomName(:old.room\_seq) || ')');  dbms\_output.put\_line('수정 후: No.' || :new.oc\_seq || ' '  || fnGetCourseName(:new.course\_seq)  || '(' || :new.oc\_startdate || ' ~ ' || :new.oc\_enddate  || ', ' || fnGetRoomName(:new.room\_seq) || ')');  end;  -- 개설 과정 수정 프로시저  create or replace procedure procUpdateOpenCourse (  pseq number,  pcseq number,  pstartdate date,  penddate date,  prseq number  )  is  vinfo varchar2(1000);  begin  dbms\_output.put\_line(chr(10) || '[개설 과정 수정]');    vinfo := 'No.' || pseq || ' ' || fnGetCourseName(pcseq)  || '(' || pstartdate || ' ~ ' || penddate  || ', ' || fnGetRoomName(prseq) || ')' || chr(10) || chr(10);    if fnIsValidDate(pstartdate) = 'N' then  dbms\_output.put\_line(vinfo || '☞실패; 시작일 부적합');  elsif fnIsValidDate(penddate) = 'N'  or penddate < pstartdate then  dbms\_output.put\_line(vinfo || '☞실패; 종료일 부적합');  elsif fnIsValidRoom(prseq, pstartdate) = 'N' then  dbms\_output.put\_line(vinfo || '☞실패; 강의실 부적합');  else  dbms\_output.put\_line(vinfo || '☞성공!');    update tblOpenCourse set course\_seq = pcseq,  oc\_startdate = pstartdate,  oc\_enddate = penddate,  room\_seq = prseq  where oc\_seq = pseq;  end if;    exception  when others then  dbms\_output.put\_line(vinfo || '실패; ' || sqlerrm);  end procUpdateOpenCourse;  -- 개설 과정 수정  begin  procUpdateOpenCourse(번호, 과정\_번호, 시작일, 종료일, 강의실\_번호);  end;  **-- 4. 개설 과정 삭제**  -- 개설 과정 삭제 프로시저  create or replace procedure procDeleteOpenCourse (  pseq number  )  is  voc tblOpenCourse%rowtype;  begin  select \* into voc from tblOpenCourse  where oc\_seq = pseq;    dbms\_output.put\_line(chr(10) || '[개설 과정 삭제]' || chr(10)  || 'No.' || pseq || ' ' || fnGetCourseName(voc.course\_seq)  || '(' || voc.oc\_startdate || ' ~ ' || voc.oc\_enddate  || ', ' || fnGetRoomName(voc.room\_seq) || ')' || chr(10));    delete from tblOpenCourse  where oc\_seq = pseq;    dbms\_output.put\_line('☞성공!');  exception  when others then  dbms\_output.put\_line('☞실패; ' || sqlerrm);  end procDeleteOpenCourse;  -- 개설 과정 삭제  begin  -- procDeleteOpenCourse(번호);  procDeleteOpenCourse(3);  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-3. 개설 과정 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-3.2) 과정 상세 정보 조회** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 과정 상세 정보 조회**  -- 개설 과정 상세 정보 저장 프로시저  create or replace procedure procSetOpenCourseInfo(  presult out sys\_refcursor  )  is  begin  dbms\_output.put\_line(chr(10) || '[개설 과정 상세 조회]');  dbms\_output.put\_line('----------------------------------------------------------------------------');  dbms\_output.put\_line('|No.|' || lpad('과정명', 43) || lpad('|', 39)  || lpad('기간', 17) || lpad('|', 13)  || ' 강의실| 등록|과목 개설|');  dbms\_output.put\_line('----------------------------------------------------------------------------');  open presult  for select \* from vwOpenCourse  order by oc\_seq;  end procSetOpenCourseInfo;  -- 개설 과정 상세 정보 조회 프로시저  create or replace procedure procGetOpenCourseInfo  is  vresult sys\_refcursor;  vrow vwOpenCourse%rowtype;  vname vwOpenCourse.course\_name%type;  begin  procSetOpenCourseInfo(vresult);    loop  fetch vresult into vrow;  exit when vresult%notfound;  dbms\_output.put\_line('| ' || to\_char(vrow.oc\_seq, '00') || '|' || vrow.course\_name || chr(9)  || '|' || vrow.oc\_startdate || '~' || vrow.oc\_enddate  || '|' || vrow.room\_name || '|'  || to\_char(vrow.num, '00') || '명| '  || vrow.regsub || ' |');  dbms\_output.put\_line('----------------------------------------------------------------------------');  end loop;  end procGetOpenCourseInfo;  -- 개설 과정 상세 조회  begin  procGetOpenCourseInfo;  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-3. 개설 과정 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-3.3) 개설 과목 정보 조회** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 개설 과목 정보 조회**  select  distinct subject\_name as "개설 과목명",  os\_startdate as "시작일",  os\_enddate as "종료일"  from vwOpenCourse  --where oc\_seq = 개설\_과정\_번호  where oc\_seq = 1  order by os\_startdate desc; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-3. 개설 과정 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-3.4) 교육생 정보 조회** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 과정에 등록한 교육생 정보 조회**  select  name as "이름",  ssn as "주민등록번호",  tel as "연락처"  from vwStudentInfo  --where oc\_seq = 개설\_과정\_번호  where oc\_seq = 1  order by name; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-4. 개설 과목 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-4.1) 개설 과목 입력/수정/삭제** | 작성자 | 서민종 |
| DML | | | |
| --------------------------------------------------------------------------------  -- A-4. 개설 과목 관리  -- A-4.1) 개설과목 입력/수정/삭제  --------------------------------------------------------------------------------  insert into tblOpenSubject (os\_seq, subject\_seq, oc\_seq, os\_startdate, os\_enddate)  values ((select max(os\_seq)+1 from tblopensubject), 52, 13, '20-10-24', '20-12-02');  update tblopensubject set subject\_seq = 56 where os\_seq = 0;  update tblopensubject set oc\_seq = 5 where os\_seq = 0;  delete from tblopensubject where os\_seq = 0;  --------------------------------------------------------------------------------  ------------------------------개설 과목 입력---------------------------------  begin  procNewSubject(60,5,'20-10-24','20-12-02');  end; -- 기초과목번호 subject\_seq, 소속개설과정번호 oc\_seq  --------------------------------------------------------------------------------  create or replace procedure procNewSubject(  psub1 number,  poc1 number,  posdate1 date,  posdate2 date  --select \* from tblcourse;  --select \* from tblsubject;  --select \* from tblopencourse;  --select \* from tblopensubject;  )  is  psub1check number;  poc1check number;  posdate1check date;  posdate2check date;  begin  select subject\_seq into psub1check from tblsubject where subject\_seq = psub1;  select oc\_seq into poc1check from tblopencourse where oc\_seq = poc1;  select oc\_startdate into posdate1check from tblopencourse where oc\_seq = poc1;  select oc\_enddate into posdate2check from tblopencourse where oc\_seq = poc1;  if psub1 = psub1check and  poc1 = poc1check and  posdate1 >= posdate1check and  posdate2 <= posdate2check then  insert into tblOpenSubject (os\_seq, subject\_seq, oc\_seq, os\_startdate, os\_enddate)  values ((select max(os\_seq)+1 from tblopensubject), psub1, poc1, posdate1, posdate2);  dbms\_output.put\_line('해당개설과정이 성공적으로 등록되었습니다.');  else  dbms\_output.put\_line('해당개설과정의 날짜범위를 벗어났습니다.');  end if;  exception  when NO\_DATA\_FOUND then  psub1check := 0;  poc1check := 0;  dbms\_output.put\_line('유효하지않는값존재');  end procNewsubject; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-4. 개설 과목 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-4.2) 개설 과목 정보 출력** | 작성자 | 서민종 |
| DML | | | |
| --------------------------------------------------------------------------------  -- A-4.2) 개설 과목 정보 출력  --------------------------------------------------------------------------------  select  distinct s.subject\_name as "과목명",  os.os\_startdate || '~' || os.os\_enddate as "기간",  b.book\_name as "교재명",  t.teacher\_name as "교사명"  from tblsubject s  inner join tblopensubject os  on s.subject\_seq = os.subject\_seq  inner join tblbook b  on s.subject\_seq = b.subject\_seq  inner join tblopencourse oc  on oc.oc\_seq = os.oc\_seq  inner join tblteachermanagement tm  on oc.oc\_seq = tm.oc\_seq  inner join tblteacher t  on tm.teacher\_seq = t.teacher\_seq  order by s.subject\_name;  ------------------------------개설 과목 출력--------------------------------  begin  procOpenSubjectOutput(10);  end; -- 개설과목번호 os\_seq(29...33해당os\_seq미존재:기초과목테이블에는있음)  -------------------------------------------------------------------------------  -- 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 교사명  -------------------------------------------------------------------------------  create or replace procedure procOpenSubjectOutput(  pos1 number  )  is  vrow tblOpenSubject%rowtype;  psub\_seq number;  psub\_name varchar2(60);  pteacher\_name varchar2(30);  pbook\_name varchar2(100);  begin  select \* into vrow from tblopensubject where os\_seq = pos1;  select subject\_seq into psub\_seq from tblopensubject where os\_seq = pos1;  select subject\_name into psub\_name from tblsubject where subject\_seq = psub\_seq;  select t.teacher\_name into pteacher\_name from tblteacher t where teacher\_seq  = (select tm.teacher\_seq from tblteachermanagement tm where oc\_seq  = (select os.oc\_seq from tblopensubject os where os\_seq = pos1));  select book\_name into pbook\_name from tblbook where subject\_seq = pos1;  dbms\_output.put\_line('개설과목명: ' || psub\_name);  dbms\_output.put\_line('시작년월일: ' || vrow.os\_startdate);  dbms\_output.put\_line('종료년월일: ' || vrow.os\_enddate);  dbms\_output.put\_line('교사명: ' || pteacher\_name);  dbms\_output.put\_line('교재명: ' || pbook\_name);  exception  when NO\_DATA\_FOUND then  dbms\_output.put\_line('해당os\_seq가 존재하지않습니다');  end procOpenSubjectOutput; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-4. 개설 과목 관리** | 작성일 | 2021-12-02 |
| 상세 | **A-4.3) 교재 정보 입력/출력/수정/삭제** | 작성자 | 서민종 |
| DML | | | |
| --------------------------------------------------------------------------------  -- A-4.3) 교재 정보 입력/출력/수정/삭제  --------------------------------------------------------------------------------  insert into tblbook (book\_seq, book\_name, book\_publisher, book\_price, subject\_seq)  values ((select max(book\_seq)+1 from tblbook), '함께하는 오라클', 쌍용북스, 15000, 3);  update tblbook set book\_name where subject\_seq = 3;  update tblbook set book\_publisher where subject\_seq = 3;  update tblbook set book\_price where subject\_seq = 3;  delete from tblbook where subject\_seq = 3;  -------------------------------------------------------------------------------- | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-6. 시험 관리 및 성적 조회** | 작성일 | 2021-12-02 |
| 상세 | **A-6.1) 개설 과목 별로 성적 등록 여부, 시험 문제 파일 등록 여부** | 작성자 | 서민종 |
| DML | | | |
| --------------------------------------------------------------------------------  -- A-6. 시험 관리 및 성적 조회  -- A-6.1) 개설 과목 별로 성적 등록 여부, 시험 문제 파일 등록 여부  --------------------------------------------------------------------------------  select  os.os\_seq as "개설과목번호",  s.score\_exam as "필기성적",  s.score\_skill as "실기성적",  s.score\_attendance as "출결성적",  t.test\_file as "파일경로"  from tblopensubject os  inner join tblscore s  on os.oc\_seq = s.os\_seq  inner join tbltest t  on os.oc\_seq = t.os\_seq  order by os.os\_seq; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-6. 시험 관리 및 성적 조회** | 작성일 | 2021-12-02 |
| 상세 | **A-6.2) 교육생 개인이 수강한 모든 개설 과목에 대한 성적 정보(개설 과목명, 개설 과목 기간, 교사명, 출결, 필기, 실기)** | 작성자 | 서민종 |
| DML | | | |
| select  distinct sj.subject\_name as "개설과목명",  os.os\_startdate as "시작일",  os.os\_enddate as "종료일",  sc.score\_exam as "필기성적",  sc.score\_skill as "실기성적",  sc.score\_attendance as "출결성적",  t.teacher\_name "교사명"  from tblstudent s  inner join tblenrollment e  on s.student\_seq = e.student\_seq  inner join tblopencourse oc  on e.oc\_seq = oc.oc\_seq  inner join tblopensubject os  on oc.oc\_seq = os.oc\_seq  inner join tblsubject sj  on sj.subject\_seq = os.subject\_seq  inner join tblteachermanagement tm  on oc.oc\_seq = tm.oc\_seq  inner join tblscore sc  on os.os\_seq = sc.os\_seq  inner join tblteacher t  on tm.teacher\_seq = t.teacher\_seq  where e.student\_seq = 3  order by os.os\_startdate;  -------------------------------------------------------------------------------  begin  procStudentScoreInfo(3);  end;  -- 수강신청번호  -------------------------------------------------------------------------------  create or replace procedure procStudentScoreInfo(  pes1 number  )  is  cursor vcursor  is select  "개설과목명",  "시작일",  "종료일",  "필기성적",  "실기성적",  "출결성적",  "교사명",  "a",  "b",  "c"  from vwStudentScoreInfo where "수강번호" = pes1;    vexam\_result number;  vskill\_result number;  vattendance\_result number;  begin  for vrow in vcursor loop    vexam\_result := vrow."a" \* (vrow."필기성적"/50);  vskill\_result := vrow."b" \* (vrow."실기성적"/50);  vattendance\_result := vrow."c" \* (vrow."출결성적"/50);  dbms\_output.put\_line(  '['||vrow."개설과목명"||']'||' '||  '['||vrow."시작일"||']'||' '||  '['||vrow."종료일"||']'||' '||  '['||vexam\_result||']'||' '||  '['||vskill\_result||']'||' '||  '['||vattendance\_result||']'||' '||  '['||vrow."교사명"||']');    end loop;  end;  -------------------------------------------------------------------------------  -------------------------------------------------------------------------------  create or replace view vwStudentScoreInfo  as select  distinct sj.subject\_name as "개설과목명",  os.os\_startdate as "시작일",  os.os\_enddate as "종료일",  sc.score\_exam as "필기성적",  sc.score\_skill as "실기성적",  sc.score\_attendance as "출결성적",  t.teacher\_name as "교사명",  e.enrollment\_seq as "수강번호",  sp.exam\_per as "a",  sp.skill\_per as "b",  sp.attendance\_per as "c",  os.os\_seq  from tblstudent s  inner join tblenrollment e  on s.student\_seq = e.student\_seq  inner join tblopencourse oc  on e.oc\_seq = oc.oc\_seq  inner join tblopensubject os  on oc.oc\_seq = os.oc\_seq  inner join tblsubject sj  on sj.subject\_seq = os.subject\_seq  inner join tblteachermanagement tm  on oc.oc\_seq = tm.oc\_seq  inner join tblscore sc  on os.os\_seq = sc.os\_seq  inner join tblteacher t  on tm.teacher\_seq = t.teacher\_seq  inner join tblscorepercent sp  on sp.os\_seq = os.os\_seq  order by os.os\_startdate  ------------------------------------------------------------------------------- | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-6. 시험 관리 및 성적 조회** | 작성일 | 2021-12-02 |
| 상세 | **A-6.3) 개설 과목별 성적 출력(과정명, 개설과정기간, 강의실명, 개설과목명, 교사명, 교육생 이름, 교육생주민등록번호, 필기, 실기, 출결)** | 작성자 | 서민종 |
| DML | | | |
| select  distinct c.course\_name as "과정명",  c.course\_period as "개설과정기간",  r.room\_name as "강의실명",  s.subject\_name as "개설과목명",  t.teacher\_name as "교사명",  st.student\_name as "교육생이름",  st.student\_ssn as "교육생주민번호",  sc.score\_exam as "필기점수",  sc.score\_skill as "실기점수"  from tblopencourse oc  inner join tblcourse c  on oc.course\_seq = c.course\_seq  inner join tblroom r  on oc.room\_seq = r.room\_seq  inner join tblopensubject os  on os.oc\_seq = oc.oc\_seq  inner join tblsubject s  on os.subject\_seq = s.subject\_seq  inner join tblteachermanagement tm  on tm.oc\_seq = oc.oc\_seq  inner join tblteacher t  on tm.teacher\_seq = t.teacher\_seq  inner join tblenrollment e  on e.oc\_seq = oc.oc\_seq  inner join tblstudent st  on st.student\_seq = e.student\_seq  inner join tblscore sc  on sc.enrollment\_seq = e.enrollment\_seq; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-8. 면접 관리** | 작성일 | 2021-12-02 |
| 상세 | 1. 비회원의 면접 신청 정보를 조회한다. | 작성자 | 심연경 |
| DML | | | |
| insert into tblInterview(interviewer\_seq, interview\_date, interview\_result)  values(803,  --'2019-01-06', --임의 입력  (select interviewer\_date from tblInterviewer where interviewer\_seq = 803),--희망날짜 = 면접날짜  null); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-8. 면접 관리** | 작성일 | 2021-12-02 |
| 상세 | 2.면접 신청 일자와 가능일자를 비교하여 면접 확정일을 입력한다. | 작성자 | 심연경 |
| DML | | | |
| --면접 일자 없는 것만 뽑아오기  select  ir.interviewer\_seq as "면접번호",  ir.interviewer\_name as "이름",  ir.interviewer\_tel as "전화번호",  ir.oc\_seq as "희망과정번호",  ir.interviewer\_date as "희망 면접일",  i.interview\_date as "확정 면접일"  from from tblInterviewer ir  left join tblInterview i  on ir.interviewer\_seq = i.interviewer\_seq  where i.interview\_date is null;  --이번 달 희망 면접일//면접 가능일자 확인하기  --vwThisMonthHopeDate로 확인  select v.regdate, v.interviewer\_name,v.interviewer\_seq from vwThisMonthHopeDate v left outer join tblinterview i on v.interviewer\_seq = i.interviewer\_seq;      --희망면접일이 공휴일이 아니라 면접이 가능한 학생 번호  select tm.regdate, n."면접번호" from vwthismonthhopedate tm left join vwinterviewnull n on tm.interviewer\_seq = n."면접번호"  where n."확정 면접일" is null and instr(tm.interviewer\_name,'\*')=0;  --면접일이 공휴일+주말이라 안되는 신청자  select tm.regdate, n."면접번호" from vwthismonthhopedate tm inner join vwinterviewnull n on tm.interviewer\_seq = n."면접번호"  where n."확정 면접일" is null and instr(tm.interviewer\_name,'\*')>0;  --면접 확정일자 입력  insert into tblInterview(interviewer\_seq, interview\_date, interview\_result)  values(803,  --'2019-01-06', --임의 입력  (select interviewer\_date from tblInterviewer where interviewer\_seq = 803),--희망날짜 = 면접날짜  null);  --평일인 경우 확정일자 반복해서 입력  --프로시저 loop 이용  DECLARE  V\_CNT NUMBER(2) := 0; --초기값  vcount number :=0;  BEGIN  select count(\*) into vcount from vwthismonthhopedate tm inner join vwinterviewnull n on tm.interviewer\_seq = n."면접번호"  where n."확정 면접일" is null and instr(tm.interviewer\_name,'\*')=0;  LOOP  EXIT WHEN V\_CNT = vcount;  V\_CNT := V\_CNT + 1;  insert into tblinterview(interviewer\_seq, interview\_date, interview\_result)  values((select tm.interviewer\_seq from vwthismonthhopedate tm inner join vwinterviewnull n on tm.interviewer\_seq = n."면접번호" where n."확정 면접일" is null  and instr(tm.interviewer\_name,'\*')=0 and rownum =1),  (select tm.regdate from vwthismonthhopedate tm inner join vwinterviewnull n on tm.interviewer\_seq = n."면접번호" where n."확정 면접일" is null  and instr(tm.interviewer\_name,'\*')=0 and rownum =1),  null);  END LOOP;  END;  –vwinterviewnull  --면접일자 없는 사람  create or replace view vwInterviewNull  as  select  ir.interviewer\_seq as "면접번호",  ir.interviewer\_name as "이름",  ir.interviewer\_tel as "전화번호",  ir.oc\_seq as "희망과정번호",  ir.interviewer\_date as "희망 면접일",  i.interview\_date as "확정 면접일"  from tblInterviewer ir  left join tblInterview i  on ir.interviewer\_seq = i.interviewer\_seq  where i.interview\_date is null;    commit; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-8. 면접 관리** | 작성일 | 2021-12-02 |
| 상세 | **3.면접 응시 완료자들은 면접 합격 여부를 입력한다.** | 작성자 | 심연경 |
| DML | | | |
| UPDATE tblInterview  SET interview\_result = 'Y'  WHERE to\_char(interview\_date) =to\_char(sysdate, 'yy/mm/dd')--오늘 날짜  and interviewer\_seq = 802;--면접자 번호 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-8. 면접 관리** | 작성일 | 2021-12-02 |
| 상세 | **4.면접 합격자의 수강 과정은 희망 과정을 참고하며, 변경할 수 있다.** | 작성자 | 심연경 |
| DML | | | |
| insert into tblEnrollment (enrollment\_seq, oc\_seq, student\_seq)  values (enrollment\_seq.nextVal,과정번호,학생번호); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-9. 사후관리** | 작성일 | 2021-12-02 |
| 상세 | **1.** 교육생 정보와 취업 여부, 취업지원 희망 여부를 입력 및 조회한다. | 작성자 | 심연경 |
| DML | | | |
| --사후 처리 기재된 전체 교육생  select  p.pm\_seq as "사후관리 여부",  v.sname as " 학생명",  v.stel as "전화번호",  p.pm\_employee as "취업여부",  p.pm\_jobseeker as "취업지원 희망 여부"  from vwstudentinfo v  inner join tblpostmanagement p  on v.eseq = p.enrollment\_seq  inner join tbljobseeker j  on p.pm\_seq = j.pm\_seq;  --그 중 취업한 사람  select  p.pm\_seq as "사후관리 여부",  v.sname as " 학생명",  v.stel as "전화번호",  p.pm\_employee as "취업여부",  p.pm\_jobseeker as "취업지원 희망 여부"  from vwstudentinfo v  inner join tblpostmanagement p  on v.eseq = p.enrollment\_seq  inner join tbljobseeker j  on p.pm\_seq = j.pm\_seq  where p.pm\_employee = 'Y'  ; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-9. 사후관리** | 작성일 | 2021-12-02 |
| 상세 | **2. 취업자의 경우 취업분야, 취업일, 고용보험 가입 여부 확인 가능** | 작성자 | 심연경 |
| DML | | | |
| select  v.sname as " 학생명",  v.stel as "전화번호",  --p.pm\_employee as "취업여부",  e.employee\_hiredate as "취업일",  e.employee\_insurance as"고용 보험 가입 여부",  rc.rc\_name as "취업 분야"  from vwstudentinfo v  inner join tblpostmanagement p  on v.eseq = p.enrollment\_seq  inner join tblemployee e  on p.pm\_seq = e.pm\_seq  inner join tblrecruitcategory rc  on rc.rc\_seq = e.rc\_seq  where p.pm\_employee = 'Y'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-9. 사후관리** | 작성일 | 2021-12-02 |
| 상세 | 취업 지원 희망자의 경우 희망 연봉, 근무지, 희망 분야를 확인 | 작성자 | 심연경 |
| DML | | | |
| select  p.pm\_seq as "사후관리 여부",  s.student\_name as " 학생명",  s.student\_tel as "전화번호",  j.js\_salary as "희망연봉",  a.area\_name as "희망근무지",  rc.rc\_name as "희망취업분야",  --p.pm\_jobseeker as "취업지원 희망 여부",  p.pm\_employee as "취업여부"    from from tblenrollment e  inner join tblstudent s  on e.student\_seq = s.student\_seq  inner join tblopencourse o  on e.oc\_seq = o.oc\_seq  inner join tblcourse c  on o.course\_seq = c.course\_seq  inner join tblpostmanagement p  on e.enrollment\_seq = p.enrollment\_seq  inner join tblemployee e  on p.pm\_seq = e.pm\_seq  inner join tblrecruitcategory rc  on rc.rc\_seq = e.rc\_seq  inner join tbljobseeker j  on p.pm\_seq = j.pm\_seq  inner join tblarea a  on j.area\_seq = a.area\_seq  where p.pm\_jobseeker ='Y'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **A-9. 사후관리** | 작성일 | 2021-12-02 |
| 상세 | 알선하는 기업의 희망 근무자와 일치하는 교육생에게 연결한다. | 작성자 | 심연경 |
| DML | | | |
| select  s.sname as "학생명",  s.stel as "학생번호",  a.area\_name as "지역명",  c.company\_name as "회사명",  c.company\_tel as "회사번호"  from tblRecruitInfo r  inner join tblJobSeeker j  on j.rc\_seq = r.rc\_seq  and r.ri\_salary = j.js\_salary --연봉  inner join tblcompany c  on r.company\_seq = c.company\_seq  inner join tblarea a  on c.area\_seq = a.area\_seq  inner join tblpostmanagement p  on p.pm\_seq = j.pm\_seq  inner join vwstudentinfo s  on p.enrollment\_seq =s.eseq; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-1. 강의 스케줄 조회** | 작성일 | 2021-12-02 |
| 상세 | **B-1. 1) 강의 스케줄 조회** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 현재 강의 스케줄 확인**  -- 강의 스케줄 저장 프로시저  create or replace procedure procSetTeacherSchedule(  presult out sys\_refcursor,  pseq number  )  is  vc tblTeacher%rowtype;  begin  select \*  into vc  from tblTeacher  where teacher\_seq = pseq;    dbms\_output.put\_line(chr(10) || '[ ' || vc.teacher\_name || ' 선생님 강의 스케줄 조회 ]');  dbms\_output.put\_line('------------------------------------------------------------');  dbms\_output.put\_line('|' || lpad('과정명', 43) || lpad('|', 41)  || lpad('기간', 17) || lpad('|', 13)  || '상태|');  dbms\_output.put\_line('------------------------------------------------------------');    open presult  for select \* from vwTeacherSchedule  where teacher\_seq = pseq  order by oc\_startdate desc;  end procSetTeacherSchedule;  -- 강의 스케줄 조회 프로시저  create or replace procedure procGetTeacherSchedule(  pseq number  )  is  vresult sys\_refcursor;  vrow vwTeacherSchedule%rowtype;  vname vwTeacherSchedule.course\_name%type;  begin  procSetTeacherSchedule(vresult, pseq);    loop  fetch vresult into vrow;  exit when vresult%notfound;  dbms\_output.put\_line('|' || chr(9) || vrow.course\_name || chr(9)  || '|' || vrow.oc\_startdate || '~' || vrow.oc\_enddate  || '|' || vrow.state || '|');  dbms\_output.put\_line('------------------------------------------------------------');  end loop;  end procGetTeacherSchedule;  -- 해당 교사의 강의 스케줄 조회  begin  procGetTeacherSchedule(2);  end;  -- 2. 강의 스케줄 상세 조회  select  course\_name as "과정명",  oc\_startdate as "과정 시작일",  oc\_enddate as "과정 종료일",  subject\_name as "과목명",  os\_startdate as "과목 시작일",  os\_enddate as "과목 종료일",  room\_name as "강의실",  book\_name as "교재",  (select count(\*) from tblEnrollment  group by oc\_seq having oc\_seq = v.oc\_seq) as "등록 인원"  from vwOpenCourse v  --where teacher\_seq = 교사\_번호  -- and oc\_seq = 개설\_과정\_번호;  where teacher\_seq = 1  and oc\_seq = 20;  -- 3. 해당 과정의 교육생 정보 조회  select  name as "이름",  tel as "연락처",  case  when (select count(\*) from tblAbandonment  where enrollment\_seq = v.enrollment\_seq) = 1 then '중도 탈락'  when oc\_startdate > sysdate then '예정'  when oc\_enddate >= sysdate then '진행 중'  when oc\_enddate < sysdate then '수료'  end as "상태"  from vwStudentInfo v  where oc.seq = 개설\_과정\_번호  order by name; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-6. 배점 입출력** | 작성일 | 2021-12-02 |
| 상세 | **B-2.1) 배점 정보를 출결, 필기, 실기로 구분해서 등록** | 작성자 | 서민종 |
| DML | | | |
| insert into tblScorePercent (sp\_seq, os\_seq, exam\_per, skill\_per, attendance\_per)  values ((select max(sp\_seq)+1 from tblscorepercent), 50, 46, 16, 38); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-6. 배점 입출력** | 작성일 | 2021-12-02 |
| 상세 | **B-2.2) 시험날짜, 시험 문제 추가** | 작성자 | 서민종 |
| DML | | | |
| insert into tblTest (test\_seq, test\_file, os\_seq, test\_type, test\_date)  values (test\_seq.nextVal, 'https://www.sycenter.net/document/00446', 20, '실기', '01-08-04'); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-6. 배점 입출력** | 작성일 | 2021-12-02 |
| 상세 | **B-2.3) 과목 목록 출력시 개설과목명, 개설과목기간, 개설과정명, 개설과정기간, 강의실, 교재명, 배점정보, 성적정보** | 작성자 | 서민종 |
| DML | | | |
| select  distinct s.subject\_name as "개설과목명",  s.subject\_period as "개설과목기간",  c.course\_name as "개설과정명",  c.course\_period as "개설과정기간",  oc.room\_seq as "강의실",  b.book\_name as "교재명",  sp.attendance\_per as "출결비중",  sp.skill\_per as "실기비중",  sp.exam\_per "필기비중",  sc.score\_attendance as "출결성적",  sc.score\_skill as "실기성적",  sc.score\_exam as "필기성적",  te.test\_file as "파일등록"  from tblopensubject os  inner join tblsubject s  on s.subject\_seq = os.subject\_seq  inner join tblopencourse oc  on oc.oc\_seq = os.oc\_seq  inner join tblbook b  on b.subject\_seq = s.subject\_seq  inner join tblteachermanagement tm  on oc.oc\_seq = tm.oc\_seq  inner join tblteacher t  on tm.teacher\_seq = t.teacher\_seq  inner join tblscore sc  on os.os\_seq = sc.os\_seq  inner join tblScorePercent sp  on sp.os\_seq = os.os\_seq  inner join tbltest te  on te.os\_seq = os.os\_seq  inner join tblcourse c  on oc.course\_seq = c.course\_seq | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-3. 성적 입출력** | 작성일 | 2021-12-02 |
| 상세 | **B-3.1) 특정 과목 선택시 교육생 이름, 교육생 전화번호, 중도탈락 및 출결, 필기, 실기 성적 출력** | 작성자 | 서민종 |
| DML | | | |
| select  os.os\_seq,  st.student\_name as "교육생이름",  st.student\_tel as "교육생전화번호",  a.abandonment\_date as "중도탈락일",  sc.score\_attendance as "출결성적",  sc.score\_exam as "필기성적",  sc.score\_skill as "실기성적"  from tblopensubject os  inner join tblopencourse oc  on os.oc\_seq = oc.oc\_seq  inner join tblenrollment e  on oc.oc\_seq = e.oc\_seq  inner join tblstudent st  on e.student\_seq = st.student\_seq  inner join tblabandonment a  on e.enrollment\_seq = a.enrollment\_seq  inner join tblscore sc  on os.os\_seq = sc.os\_seq  where os.os\_seq = 3  order by os.os\_seq; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-3. 성적 입출력** | 작성일 | 2021-12-02 |
| 상세 | **B-3.1) 특정 교육생 정보 선택 시 해당 교육생의 시험 점수를 입력** | 작성자 | 서민종 |
| DML | | | |
| insert into tblScore (score\_seq, enrollment\_seq, os\_seq, sp\_seq, score\_exam, score\_skill, score\_attendance)  values  ((select max(score\_seq) from tblscore)+1,  (select max(enrollment\_seq)-- 더미데이터 존재로 max를 사용했으나 교육생이름과 수강신청번호는 1:1매칭이기때문에 실제데이터 구현시 max는 삭제  from tblenrollment e  inner join tblstudent s  on e.student\_seq = s.student\_seq  where s.student\_name = '서민종'),  16,  3,  22,  16,  19); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-3. 성적 입출력** | 작성일 | 2021-12-02 |
| 상세 | **B-3.2) 교육생 전체에 대해서 성적을 등록했는지의 여부 출력** | 작성자 | 서민종 |
| DML | | | |
| select  distinct s.student\_name as "교육생이름",  case  when sc.score\_exam is not null then '등록'  when sc.score\_exam is null then '미등록'  end as "필기등록여부",  case  when sc.score\_skill is not null then '등록'  when sc.score\_skill is null then '미등록'  end as "실기등록여부",  case  when sc.score\_attendance is not null then '등록'  when sc.score\_attendance is null then '미등록'  end as "출결등록여부"  from tblenrollment e  left outer join tblstudent s  on e.student\_seq = s.student\_seq  left outer join tblscore sc  on sc.enrollment\_seq = e.enrollment\_seq; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | B-6. 상담일지 작성 및 조회 | 작성일 | 2021-12-02 |
| System | 상담일지 작성/ 수정/ 삭제 | 작성자 | 송성현 |
| DML | | | |
| -- 상담 작성  **INSERT** **INTO** tblCounselLog(cl\_seq, enrollment\_seq, cl\_date, cc\_seq, cl\_content) **values**(11,1,**to\_date**('21-12-01', 'yy-mm-dd'),3,'취업연계과정에 대한 번반적인 설명');  -- 상담 수정  **UPDATE** tblCounselLog **SET** cl\_content = '취업연계과정에 대한 전반적인 설명' **WHERE** cl\_seq = 11;  -- 상담 삭제  **delete** **from** tblCounselLog **where** cl\_seq = 11; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | B-6. 상담일지 작성 및 조회 | 작성일 | 2021-12-02 |
| System | 상담일지 조회 | 작성자 | 송성현 |
| DML | | | |
| **SELECT**  cl.cl\_date **AS** cl\_date,  c.course\_name **AS** course\_name,  s.student\_name **AS** student\_name  **FROM** tblCourse c  **INNER** **JOIN** tblOpenCourse oc **ON** c.course\_seq = oc.course\_seq  **INNER** **JOIN** tblEnrollment e **ON** oc.OC\_SEQ = e.OC\_SEQ  **INNER** **JOIN** tblStudent s **ON** e.student\_seq = s.student\_seq  **INNER** **JOIN** tblQuestion q **ON** s.student\_seq = q.student\_seq  **INNER** **JOIN** tblAnswer a **ON** q.question\_seq = a.question\_seq  **INNER** **JOIN** tblTeacher t **ON** a.teacher\_seq = t.teacher\_seq  **INNER** **JOIN** tblCounselLog cl **ON** cl.enrollment\_seq = e.enrollment\_seq; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | B-6. 상담일지 작성 및 조회 | 작성일 | 2021-12-02 |
| System | 카테고리별 보기 | 작성자 | 송성현 |
| DML | | | |
| **SELECT** cl\_date, cl\_content **FROM** tblCounselLog cl **INNER** **JOIN** TBLCOUNSELCATEGORY cc **ON** cl.cc\_seq = cc.cc\_seq ; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-7. 사후 처리 조회** | 작성일 | 2021-12-02 |
| 상세 | **1.** 교육생 정보와 취업 여부, 취업지원 희망 여부를 입력 및 조회한다. | 작성자 | 심연경 |
| DML | | | |
| --사후 처리 기재된 전체 교육생  select  p.pm\_seq as "사후관리 여부",  v.sname as " 학생명",  v.stel as "전화번호",  p.pm\_employee as "취업여부",  p.pm\_jobseeker as "취업지원 희망 여부"  from vwstudentinfo v  inner join tblpostmanagement p  on v.eseq = p.enrollment\_seq  inner join tbljobseeker j  on p.pm\_seq = j.pm\_seq;  --그 중 취업한 사람  select  p.pm\_seq as "사후관리 여부",  v.sname as " 학생명",  v.stel as "전화번호",  p.pm\_employee as "취업여부",  p.pm\_jobseeker as "취업지원 희망 여부"  from vwstudentinfo v  inner join tblpostmanagement p  on v.eseq = p.enrollment\_seq  inner join tbljobseeker j  on p.pm\_seq = j.pm\_seq  where p.pm\_employee = 'Y'  ; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-7. 사후 처리 조회** | 작성일 | 2021-12-02 |
| 상세 | **2. 취업자의 경우 취업분야, 취업일, 고용보험 가입 여부 확인 가능** | 작성자 | 심연경 |
| DML | | | |
| select  v.sname as " 학생명",  v.stel as "전화번호",  --p.pm\_employee as "취업여부",  e.employee\_hiredate as "취업일",  e.employee\_insurance as"고용 보험 가입 여부",  rc.rc\_name as "취업 분야"  from vwstudentinfo v  inner join tblpostmanagement p  on v.eseq = p.enrollment\_seq  inner join tblemployee e  on p.pm\_seq = e.pm\_seq  inner join tblrecruitcategory rc  on rc.rc\_seq = e.rc\_seq  where p.pm\_employee = 'Y'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-7. 사후 처리 조회** | 작성일 | 2021-12-02 |
| 상세 | 취업 지원 희망자의 경우 희망 연봉, 근무지, 희망 분야를 확인 | 작성자 | 심연경 |
| DML | | | |
| select  p.pm\_seq as "사후관리 여부",  v.sname as " 학생명",  v.stel as "전화번호",  j.js\_salary as "희망연봉",  a.area\_name as "희망근무지",  rc.rc\_name as "희망취업분야",  --p.pm\_jobseeker as "취업지원 희망 여부",  p.pm\_employee as "취업여부"    from vwstudentinfo v  inner join tblpostmanagement p  on v.eseq = p.enrollment\_seq  inner join tblemployee e  on p.pm\_seq = e.pm\_seq  inner join tblrecruitcategory rc  on rc.rc\_seq = e.rc\_seq  inner join tbljobseeker j  on p.pm\_seq = j.pm\_seq  inner join tblarea a  on j.area\_seq = a.area\_seq  where p.pm\_jobseeker ='Y'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **B-7. 사후 처리 조회** | 작성일 | 2021-12-02 |
| 상세 | 알선하는 기업의 희망 근무자와 일치하는 교육생에게 연결한다. | 작성자 | 심연경 |
| DML | | | |
| select  s.sname as "학생명",  s.stel as "학생번호",  a.area\_name as "지역명",  c.company\_name as "회사명",  c.company\_tel as "회사번호"  from tblRecruitInfo r  inner join tblJobSeeker j  on j.rc\_seq = r.rc\_seq  and r.ri\_salary = j.js\_salary --연봉  inner join tblcompany c  on r.company\_seq = c.company\_seq  inner join tblarea a  on c.area\_seq = a.area\_seq  inner join tblpostmanagement p  on p.pm\_seq = j.pm\_seq  inner join vwstudentinfo s  on p.enrollment\_seq =s.eseq; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | B-8. 질의응답 답변 작성 및 조회 | 작성일 | 2021-12-02 |
| System | 질의응답 답변 작성/ 수정/ 삭제 | 작성자 | 송성현 |
| DML | | | |
| -- 답변  **insert** **into** tblAnswer (question\_seq, teacher\_seq, answer\_regdate, answer\_content) **values** (11,1,**to\_date**('21-12-01', 'yy-mm-dd'),'3열에 오타가 남네요.');  -- 답변 수정  **UPDATE** tblAnswer **SET** answer\_content = '작성한 쿼리의 열에 오타가 났네요.' **WHERE** cl\_seq = 11;  -- 답변 삭제  **delete** **from** tblAnswer **where** cl\_seq = 11; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | B-8. 질의응답 답변 작성 및 조회 | 작성일 | 2021-12-02 |
| System | 질의응답 답변 조회 | 작성자 | 송성현 |
| DML | | | |
| **SELECT**  c.course\_name,  s.student\_name,  q.question\_date,  q.question\_content  **FROM** tblCourse c  **INNER** **JOIN** tblOpenCourse oc **ON** c.course\_seq = oc.course\_seq  **INNER** **JOIN** tblEnrollment e **ON** oc.OC\_SEQ = e.OC\_SEQ  **INNER** **JOIN** tblStudent s **ON** e.student\_seq = s.student\_seq  **INNER** **JOIN** tblQuestion q **ON** s.student\_seq = q.student\_seq  **INNER** **JOIN** tblAnswer a **ON** q.question\_seq = a.question\_seq  **INNER** **JOIN** tblTeacher t **ON** a.teacher\_seq = t.teacher\_seq  **ORDER** **BY** q.question\_seq ; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | B-8. 질의응답 답변 작성 및 조회 | 작성일 | 2021-12-02 |
| System | 질의응답 답변완료 조회 | 작성자 | 송성현 |
| DML | | | |
| **SELECT**  s.student\_name **AS** questioner,  q.question\_content **AS** question,  a.answer\_content **AS** answer  **FROM** tblStudent s  **LEFT** **OUTER** **JOIN** tblQuestion q **ON** s.student\_seq = q.student\_seq  **LEFT** **OUTER** **JOIN** tblAnswer a **ON** q.question\_seq = a.question\_seq  **LEFT** **OUTER** **JOIN** tblTeacher t **ON** a.teacher\_seq = t.teacher\_seq  **WHERE** answer\_content **IS** **NOT** **NULL**; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **C-1. 교육생이 수강한 과목에 대한 성적 조회** | 작성일 | 2021-12-02 |
| 상세 | **C-1.1) 과목번호, 과목명, 과목기간, 교재명, 교사명, 배점정보, 성적정보, 과목별 시험날짜, 시험문제 출력** | 작성자 | 서민종 |
| DML | | | |
| select  distinct s.subject\_name as "개설과목명",  s.subject\_period as "개설과목기간",  c.course\_name as "개설과정명",  c.course\_period as "개설과정기간",  oc.room\_seq as "강의실",  b.book\_name as "교재명",  sp.attendance\_per as "출결비중",  sp.skill\_per as "실기비중",  sp.exam\_per "필기비중",  sc.score\_attendance as "출결성적",  sc.score\_skill as "실기성적",  sc.score\_exam as "필기성적",  te.test\_file as "파일등록"  from tblopensubject os  inner join tblsubject s  on s.subject\_seq = os.subject\_seq  inner join tblopencourse oc  on oc.oc\_seq = os.oc\_seq  inner join tblbook b  on b.subject\_seq = s.subject\_seq  inner join tblteachermanagement tm  on oc.oc\_seq = tm.oc\_seq  inner join tblteacher t  on tm.teacher\_seq = t.teacher\_seq  inner join tblscore sc  on os.os\_seq = sc.os\_seq  inner join tblScorePercent sp  on sp.os\_seq = os.oc\_seq  inner join tbltest te  on te.os\_seq = os.os\_seq  inner join tblcourse c  on oc.course\_seq = c.course\_seq | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | **C-5 사후 처리 입력** | 작성일 | 2021-12-02 |
| System | 1 취업 상황(분야/취업일/고용보험 여부)을 기재  2 취업 알선 희망 여부 등록 | 작성자 | 심연경 |
| DML | | | |
| --사후관리 수기입력  insert into tblPostManagement (pm\_seq,enrollment\_seq,pm\_employee,pm\_jobseeker)  values (  pm\_seq.nextVal, --사후관리번호  1,--수강신청번호  'N',--취업여부  'Y');--취업 알선 희망 여부  --취업 입력  insert into tblEmployee (pm\_seq,rc\_seq,employee\_hiredate,employee\_insurance,employee\_salary)  values (  pm\_seq,--사후번호  7,--취업분야  to\_date('19/06/16', --취업일  'rr/mm/dd'),  'Y',--고용보험여부  2700);--연봉 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | **C-5 사후 처리 입력** | 작성일 | 2021-12-02 |
| System | 3 알선 희망시 연봉, 근무지, 분야 기재 | 작성자 | 심연경 |
| DML | | | |
| insert into tblJobSeeker (pm\_seq,js\_salary,area\_seq,rc\_seq)  values (11, --사후관리번호  3200,--희망 연봉  (select area\_seq from tblArea where instr(area\_name, '종로구') >0),--근무지  (select rc\_seq from tblrecruitcategory where instr(rc\_name, 'game\_designer') >0));--분야 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | **D-1. 교육 과정 조회** | 작성일 | 2021-12-02 |
| 상세 | **D-1.1) 개설 과정 상세 조회** | 작성자 | 이유미 |
| DML | | | |
| **-- 1. 개설 과정 상세 조회**  select  oc\_seq as "번호",  teacher\_name as "교사명",  course\_name as "과정명",  oc\_startdate as "과정 시작일",  oc\_enddate as "과정 종료일",  subject\_name as "과목명",  os\_startdate as "과목 시작일",  os\_enddate as "과목 종료일"  from vwOpenCourseInfo  order by oc\_startdate desc; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | **D-2. 면접 신청 및 일자 조회** | 작성일 | 2021-12-02 |
| System | 1. 희망하는 면접 일자를 선택하여 면접 신청을 한다. | 작성자 | 심연경 |
| DML | | | |
| insert into tblInterviewer(interviewer\_seq, interviewer\_name, interviewer\_tel, oc\_seq, interviewer\_date)  values (interviewer\_seq.nextval,  '희망', --이름  '010-1111-2222', --번호  22, --희망과정  to\_date('2021-12-21', --면접 희망날짜 <개설과정 시작일  'YY-MM-DD')  ); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | **D-2. 면접 신청 및 일자 조회** | 작성일 | 2021-12-02 |
| System | 2. 면접 신청한 뒤 관리자의 승인을 받아 확정 면접일자가 결정된다.  3. 면접일자가 확정되면 확정 면접 일자 조회가 가능하다. | 작성자 | 심연경 |
| DML | | | |
| select i.interview\_date from tblinterviewer ir  inner join tblinterview i  on ir.interviewer\_seq = i.interviewer\_seq  --where seq = 1;--면접번호로 면접일자 조회  --where name ='탁경아';--면접자명로 면접일자 조회  where substr(tel, -4)= '2161';--전화번호 뒤 4자리로 면접일자 조회 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | **D-2. 면접 신청 및 일자 조회** | 작성일 | 2021-12-02 |
| System | 4.추후 면접 실시 후 면접 통과 여부 조회가 가능하다. | 작성자 | 심연경 |
| DML | | | |
| select i.interview\_result from tblinterviewer ir  inner join tblinterview i  on ir.interviewer\_seq = i.interviewer\_seq  --where seq = 1;  --where name ='탁경아';  where substr(tel, -4)= '2161'; | | | |