**DB Object 스크립트(PL-SQL)**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-12 | 작성일 | 2023-04-06 |
| 요구사항명 | 교사별 평점 조회 | 작성자 | 장기성 |
| DML | | | |
| -- 교사별 평점 조회 프로시저  create or replace procedure procAReview  is  cursor vcursor is  select \* from vwReview;  begin  dbms\_output.put\_line('<' ||' 교사별 평점 ' || '>');  for vrow in vcursor loop  dbms\_output.put\_line('|등수: '|| vrow.등수 || ' |교사: ' || vrow.교사 || ' |평점 : ' || vrow.총평점 || '|');  end loop;  end procAReview;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-12 | 작성일 | 2023-04-06 |
| 요구사항명 | 선택 교사의 과목별 평점 조회 | 작성자 | 장기성 |
| DML | | | |
| -- 선택 교사 과목별 평점 프로시저  create or replace procedure procTReview (  pname varchar2  )  is  cursor vcursor is  select  과목번호,  과목명,  (select  round(avg(score) ,2) as 강의평점  from tblReview r  inner join tblTeacher t  on r.t\_seq = t.t\_seq  where t.t\_seq = (select t\_seq from tblTeacher where name = pname) -- 선택한 교사 명 입력  and r.os\_seq = vos.과목번호) as 평점,  과목기간  from vwOpenSubject vos  where 교사명 = pname -- 선택한 교사 이름 입력  and substr(과정기간,12,8) <= sysdate -- 종료된 과정의 과목들만 출력  order by 과목번호;  begin  dbms\_output.put\_line('<' || pname || ' 강사 과목별 평점' || '>');  for vrow in vcursor loop  dbms\_output.put\_line('|과목번호: '|| vrow.과목번호 || ' |과목명: ' || vrow.과목명 || ' |평점 : ' || vrow.평점 || '|');  end loop;  end procTReview;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-03 | 작성일 | 2023-04-06 |
| 요구사항명 | 선택학생의 출결시각 기록 | 작성자 | 장기성 |
| DML | | | |
| -- 교육생 출결체크  create or replace procedure procEnterTag(  pid varchar2  )  is  cursor vcursor is  select  (select name from tblStudent where id = pid) as 이름,  to\_char(attenddate, 'yyyy-mm-dd') as 날짜,  to\_char(entertime, 'hh24:mi') as 등원,  to\_char(outertime, 'hh24:mi') as 하원,  state as 상태  from tblS\_attend  where s\_seq = (select s\_seq from tblStudent where id = pid)  and attenddate = to\_char(sysdate, 'yyyy-mm-dd');  begin  insert into tblS\_attend (seq, attenddate, s\_seq, entertime)  values ((select max(seq) + 1 from tblS\_attend), to\_char(sysdate, 'yyyy-mm-dd'),  (select s\_seq from tblStudent where id = pid), sysdate);    for vrow in vcursor loop  dbms\_output.put\_line('<'|| vrow.날짜 || '>');  dbms\_output.put\_line('이름: '||vrow.이름);  dbms\_output.put\_line('등원: '|| vrow.등원);  dbms\_output.put\_line('하원: '|| vrow.하원);  dbms\_output.put\_line('상태: '|| vrow.상태);  end loop;    exception  when others then  dbms\_output.put\_line('예외처리.');  end procEnterTag;  /  create or replace procedure procOuterTag(  pid varchar2  )  is  cursor vcursor is  select  (select name from tblStudent where id = pid) as 이름, to\_char(attenddate, 'yyyy-mm-dd') as 날짜,  to\_char(entertime, 'hh24:mi') as 등원, to\_char(outertime, 'hh24:mi') as 하원, state as 상태  from tblS\_attend  where s\_seq = (select s\_seq from tblStudent where id = pid)  and attenddate = to\_char(sysdate, 'yyyy-mm-dd');  vs\_seq number;  begin  select s\_seq into vs\_Seq from tblStudent where id = pid;    update tblS\_attend  set outertime = sysdate  where seq = (select max(seq) from tblS\_attend where s\_seq =  (select s\_seq from tblStudent where id = pid)); -- 로그인한 아이디  update tblS\_attend set state =  case  when to\_char(entertime, 'hh24:mi') <= '09:10' and to\_char(outertime, 'hh24:mi') >= '17:50' then '정상'  when to\_char(entertime, 'hh24:mi') > '09:10' and to\_char(outertime, 'hh24:mi') < '17:50' then '조퇴'  when to\_char(entertime, 'hh24:mi') > '09:10' and to\_char(outertime, 'hh24:mi') >= '17:50' then '지각'  when to\_char(entertime, 'hh24:mi') <= '09:10' and to\_char(outertime, 'hh24:mi') < '17:50' then '조퇴'  end  where attenddate = to\_char(sysdate, 'yyyy-mm-dd')  and s\_seq = vs\_seq;    for vrow in vcursor loop  dbms\_output.put\_line('<'|| vrow.날짜 || '>');  dbms\_output.put\_line('이름: '||vrow.이름);  dbms\_output.put\_line('등원: '|| vrow.등원);  dbms\_output.put\_line('하원: '|| vrow.하원);  dbms\_output.put\_line('상태: '|| vrow.상태);    end loop;  exception  when others then  dbms\_output.put\_line('예외처리.');  end procOuterTag;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-08 | 작성일 | 2023-04-06 |
| 요구사항명 | 선택 교육생 출결현황 조회 | 작성자 | 장기성 |
| DML | | | |
| create or replace procedure procS\_attend(  pos\_seq number    )  is  cursor vcursor is  select  distinct  s.s\_seq as "교육생번호",  s.name as "이름",  (select  count(  case  when sa2.state = '정상' then 1  end  )  from tblS\_attend sa2 where sa2.s\_seq = sa.s\_seq) as "정상출석",  (select  count(  case  when sa2.state = '지각' then 1  end  )  from tblS\_attend sa2 where sa2.s\_seq = sa.s\_seq) as "지각",  (select  count(  case  when sa2.state = '조퇴' then 1  end  )  from tblS\_attend sa2 where sa2.s\_seq = sa.s\_seq) as "조퇴",  (select  count(  case  when sa2.state = '결석' then 1  end  )  from tblS\_attend sa2 where sa2.s\_seq = sa.s\_seq) as "결석"  from tblStudent s  inner join tblOpenClass oc  on s.oc\_seq = oc.oc\_seq  inner join tblS\_attend sa  on sa.s\_seq = s.s\_seq  where s.oc\_seq = pos\_seq -- 선택한 과정 번호 입력  order by s.s\_seq;  begin  for vrow in vcursor loop  dbms\_output.put\_line('|교육생번호: ' || vrow.교육생번호 || ' | 이름: ' || vrow.이름 || ' | 정상출석: ' || vrow.정상출석 || ' | 지각: ' || vrow.지각 || ' | 조퇴: ' || vrow.조퇴 || ' | 결석: '|| vrow.결석 || '|');  end loop;    exception  when others then  dbms\_output.put\_line('예외처리.');    end procS\_attend;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-03 | 작성일 | 2023-04-03 |
| 요구사항명 | 교사계정 관리 | 작성자 | 양진영 |
| DML | | | |
| **<View>**  **- view 생성**  - 관리자가 교사계정 관리 시 전체 교사 정보 조회  create or replace view vwTeacher  as  select  oc.oc\_seq as 과정번호,  os.os\_seq as 과목번호,  t.name as 교사명,  bs.name as 과목명,  os.beginDate as 과목시작일,  os.endDate as 과목종료일,  bc.name as 과정명,  oc.beginDate 과정시작일,  oc.endDate as 과정종료일,  b.title as 교재명,  br.name as 강의실,  br.suyong as 등록인원,  case  when sysdate > os.endDate then '강의종료'  when sysdate >= os.beginDate and sysdate <= os.endDate then '강의중'  when sysdate < os.begindate then '강의예정'  end as 강의진행여부,  st.name as 학생명,  st.tel as 전화번호,  oc.beginDate as 등록일,  st.complete as 수료상태,  q.s\_seq as 중도탈락자  from tblTeacher t  inner join tblOpenSubject os  on t.t\_seq = os.t\_seq  inner join tblBasicSubject bs  on os.bs\_seq = bs.bs\_seq  inner join tblOpenClass oc  on oc.oc\_seq = os.oc\_seq  inner join tblBook b  on os.book\_seq = b.book\_seq  inner join tblBasicRoom br  on oc.br\_seq = br.br\_seq  inner join tblBasicClass bc  on oc.bc\_seq = bc.bc\_seq  inner join tblStudent st  on oc.oc\_seq = st.oc\_seq  left outer join tblquit q  on st.s\_seq = q.s\_seq  order by os.os\_seq;  **- view 호출**  select \* from tblTeacher; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-04 | 작성일 | 2023-04-03 |
| 요구사항명 | 개설과정 관리기능 | 작성자 | 양진영 |
| DML | | | |
| **<View>**  **- view 생성**  - 개설과정 조회  create or replace view vwOpenClass  as  select  oc.oc\_seq as 과정번호,  bc.name as 과정명,  oc.beginDate || ' ~ ' || oc.endDate as 과정기간,  br.name as 강의실명  from tblOpenClass oc  inner join tblBasicClass bc  on oc.bc\_seq = bc.bc\_seq  inner join tblBasicRoom br  on oc.br\_seq = br.br\_seq  order by oc\_seq;  **- view 호출**  select \* from vwOpenClass; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-05 | 작성일 | 2023-04-03 |
| 요구사항명 | 개설과목 관리기능 | 작성자 | 양진영 |
| DML | | | |
| **<View>**  **- view 생성**  - 개설과목 조회 기능  create or replace view vwOpenSubject  as  select  oc.oc\_seq as 과정번호,  os.os\_seq as 과목번호,  bc.name as 과정명,  oc.beginDate || ' ~ ' || oc.endDate as 과정기간,  br.name as 강의실명,  bs.name as 과목명,  os.beginDate || ' ~ ' || os.endDate as 과목기간,  b.title as 교재명,  t.name as 교사명  from tblOpenSubject os  inner join tblOpenClass oc  on oc.oc\_seq = os.oc\_seq  inner join tblbasicsubject bs  on os.bs\_seq = bs.bs\_seq  inner join tblBasicClass bc  on oc.bc\_seq = bc.bc\_seq  inner join tblTeacher t  on os.t\_seq = t.t\_seq  inner join tblBook b  on os.book\_seq = b.book\_seq  inner join tblBasicRoom br  on oc.br\_seq = br.br\_seq;    **- view 호출**  select \* from vwOpenSubject;  **<Procedure>**  **- select procedure 생성**  - 특정 개설 과정번호를 선택시 개설 과목 정보(과정명, 과정기간, 강의실명, 과목명, 과목기간, 교재명, 교사명) 출력  create or replace procedure procOpenSubject(  pocseq number  )  is  cursor vcursor is  select  oc.oc\_seq as 과정번호,  os.os\_seq as 과목번호,  bc.name as 과정명,  oc.beginDate || ' ~ ' || oc.endDate as 과정기간,  br.name as 강의실명,  bs.name as 과목명,  os.beginDate || ' ~ ' || os.endDate as 과목기간,  b.title as 교재명,  t.name as 교사명  from tblOpenSubject os  inner join tblOpenClass oc  on oc.oc\_seq = os.oc\_seq  inner join tblbasicsubject bs  on os.bs\_seq = bs.bs\_seq  inner join tblBasicClass bc  on oc.bc\_seq = bc.bc\_seq  inner join tblTeacher t  on os.t\_seq = t.t\_seq  inner join tblBook b  on os.book\_seq = b.book\_seq  inner join tblBasicRoom br  on oc.br\_seq = br.br\_seq  where pocseq = oc.oc\_seq;    begin  for vrow in vcursor loop  dbms\_output.put\_line('과정명: ' || vrow.과정명 || ' | 과정기간: ' || vrow.과정기간 || ' | 강의실명: ' || vrow.강의실명  || ' | 과목명: ' || vrow.과목명 || ' | 과목기간: ' || vrow.과목기간 || ' | 교재명: ' || vrow.교재명 || ' | 교사명: ' || vrow.교사명);  end loop;  end procOpenSubject;    **- procedure 호출**  exec procOpenSubject(개설과정 번호); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-03 | 작성일 | 2023-04-03 |
| 요구사항명 | 배점 입출력 | 작성자 | 양진영 |
| DML | | | |
| **<Procedure>**  **- 1. select procedure 생성**  - 특정 과목을 과목번호로 선택 시 출결 배점, 필기 배점, 실기 배점을 출력  create or replace procedure procScoreRatio(  posseq number  )  is  cursor vcursor is  select  t.name as 교사명,  bs.name as 과목명,  sr.pilgiRatio as 필기배점,  sr.silgiRatio as 실기배점,  sr.attendRatio as 출결배점,  os.beginDate as 과목시작일,  os.endDate as 과목종료일  from tblOpenSubject os  left outer join tblBasicSubject bs  on os.bs\_seq = bs.bs\_seq  left outer join tblScoreRatio sr  on os.os\_seq = sr.os\_seq  left outer join tblTeacher t  on os.t\_seq = t.t\_seq  where posseq = os.os\_seq;  begin  for vrow in vcursor  loop  dbms\_output.put\_line('교사명: ' || vrow.교사명 || ' | 과목명: ' || vrow.과목명 || ' | 필기배점: ' || nvl(to\_char(vrow.필기배점, '0.0'), ‘null’)  || ' | 실기배점: ' || nvl(to\_char(vrow.실기배점, '0.0'), 'null') || ' | 출결배점: ' || nvl(to\_char(vrow.출결배점, '0.0'), 'null')  || ' | 과목시작일: ' || vrow.과목시작일 || ' | 과목종료일: ' || vrow.과목종료일);  end loop;  Exception  when others then  dbms\_output.put\_line('검색되지 않습니다.');  end procScoreRatio;    **- procedure 호출**  begin  procScoreRatio(과목번호);  end;    **- 2. insert procedure 생성**  --프로시저 호출 후 배점을 입력할 개설과목번호, 필기배점, 실기배점, 출결배점을 입력 시 tblScoreRatio 테이블에 insert된다.  --만일 입력한 배점의 합이 1일 경우에는 insert되지만, 1이 되지 않을 시 예외처리된다.  create or replace procedure procRatio(  pos\_seq number,  ppilgi number,  psilgi number,  pattend number  )  is  vtotal number;  begin  vtotal := ppilgi + psilgi + pattend;    if vtotal = 1 then  insert into tblScoreRatio values((select max(seq) +1 from tblScoreRatio), pos\_seq, ppilgi, psilgi, pattend);  dbms\_output.put\_line('배점입력이 완료되었습니다.');  else  dbms\_output.put\_line('배점 합이 1이 되지 않습니다.');  end if;  end procRatio;    **- procedure 호출**  begin  procRatio(87, 0.3, 0.3, 0.4);  end;  **<Trigger>**  **- trigger 생성**  - 교사가 배점 수정/삭제 trigger 성적이 있는 테이블에는 배점을 수정/삭제할 수 없음(강제 오류 발생)  create or replace trigger trgRatio  before update or delete  on tblScoreRatio  for each row  declare  vscoreNum number := 0;  vos\_seq number;  begin  if updating then  vos\_seq := :new.os\_seq;  elsif deleting then  vos\_seq := :old.os\_seq;  end if;    select count(\*) into vscoreNum from tblScore s where s.os\_seq = vos\_seq;    if vscoreNum > 0 then  raise\_application\_error(-20001, '학생의 성적 데이터가 존재하여 배점 수정/삭제에 실패하였습니다.');  end if;  end;    **- trigger 호출**  --교사가 과목의 배점을 수정/삭제 시 성적이 등록되어 있는 경우 고의로 오류를 발생시킴  update tblScoreRatio set pilgiRatio = 0.5 where seq = 배점 번호; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-06 | 작성일 | 2023-04-03 |
| 요구사항명 | 자료실 조회 | 작성자 | 양진영 |
| DML | | | |
| **<Procedure>**  **- select procedure 생성**  - 특정학생의 이름을 입력 시 해당 학생의 개설과목 담당 교사가 올린 자료실 자료 출력  create or replace procedure procJaryosil(  pname varchar2  )  is  cursor vcursor is  select  jrs\_seq as 자료번호,  t.name as 작성자,  bs.name as 과목명,  j.title as 제목,  j.contents as 내용,  j.fileName as 첨부파일  from tblJaryosil j  inner join tblOpenSubject os  on os.os\_seq = j.os\_seq  inner join tblOpenClass oc  on oc.oc\_seq = os.oc\_seq  inner join tblTeacher t  on os.t\_seq = t.t\_seq  inner join tblBasicSubject bs  on os.bs\_seq = bs.bs\_seq  inner join tblStudent st  on oc.oc\_seq = st.oc\_seq  where pname = st.name  order by j.jrs\_seq;    begin  for vrow in vcursor loop  dbms\_output.put\_line('자료번호: ' || vrow.자료번호 || ' | 작성자: '  || vrow.작성자 || ' | 과목명: ' || vrow.과목명  || ' | 제목: ' || vrow.제목 || ' | 내용: ' || vrow.내용  || ' | 첨부파일: ' || nvl(vrow.첨부파일, '(null)'));  end loop;  Exception  when others then  dbms\_output.put\_line('게시물이 없습니다.');  end procJaryosil;      **- procedure 호출**  begin  procJaryosil('교육생 이름');  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-11 | 작성일 | 2022-04-10 |
| 요구사항명 | 교육생 교재수령여부 출력 | 작성자 | 정상우 |
| DML | | | |
| create or replace procedure procTakeBook (  pos\_seq number  )  is  cursor vcursor is  select  st.s\_Seq as 교육생번호,  st.name as 이름,  tb.take as 수령여부,  bs.name as 과목  from tblStudent st  inner join tblOpenSubject os  on st.oc\_Seq = os.oc\_seq  inner join tblTakeBook tb  on st.s\_seq = tb.s\_Seq  inner join tblBook b  on os.book\_Seq = b.book\_Seq  inner join tblBasicSubject bs  on bs.bs\_seq = os.bs\_seq  where tb.book\_seq = b.book\_Seq  and os.os\_SEq = pos\_seq;  begin  dbms\_output.put\_line('<' ||' 교재 수령여부 ' || '>');  for vrow in vcursor loop  dbms\_output.put\_line('|교육생번호: '|| vrow.교육생번호 || ' |이름: ' || vrow.이름 || ' |수령여부 : ' || vrow.수령여부 || '|');  end loop; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-04 | 작성일 | 2022-04-10 |
| 요구사항명 | 교사의 수강과목 학생 성적출력 | 작성자 | 정상우 |
| DML | | | |
| create or replace procedure procstudentscore (  pos\_SEQ number  )  is  cursor vcursor is  select  st.name as 이름,st.tel as 전화번호,st.complete as 수료여부,  case  when q.quitdate is null then 'X'  else '중도탈락'  end as 중도탈락여부  ,sc.attendscore as 출결점수,sc.pilgiscore as 필기점수,sc.silgiscore as 실기점수  from tblStudent st  inner join tblOpenSubject os  on st.oc\_seq = os.oc\_seq  left outer join tblScore sc  on os.os\_Seq = sc.os\_Seq and sc.s\_Seq = st.s\_Seq  left outer join tblQuit q  on st.s\_seq = q.s\_seq  where os.os\_seq = pos\_SEQ  order by st.s\_seq;    vosname varchar2(300);  begin  select name into vosname from tblBasicSubject bs  inner join tblOpenSubject os  on os.bs\_seq = bs.bs\_seq  where os.os\_Seq = pos\_seq;  dbms\_output.put\_line('< ' || vosname || ' 과목 수강 교육생 성적' || ' >');  for vrow in vcursor loop  dbms\_output.put\_line('|이름: '|| vrow.이름 || ' |전화번호: ' ||vrow.전화번호 || ' |수요여부: '  ||vrow.수료여부 || ' |중도탈락여부: ' || vrow.중도탈락여부||' |출결점수: ' ||vrow.출결점수 ||' |필기점수: '|| vrow.필기점수 || ' |실기점수: ' ||vrow.실기점수||'|' );  end loop;  end procstudentscore; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-10 | 작성일 | 2022-04-07 |
| 요구사항명 | 관리자의 취업현황 등록 및 조회 | 작성자 | 김수림 |
| DML | | | |
| --취업현황 프로시저  create or replace procedure proc\_seeGetJob(  pOpenClass number  )  is  cursor vcursor is  select  s.s\_seq as "교육생번호",  s.name as "교육생명",  s.tel as "전화번호",  s.complete as "수료여부",  gj.company as "회사명"  from tblStudent s  left outer join tblGetjob gj  on s.s\_seq = gj.s\_Seq  where s.oc\_seq = pOpenClass;    vrow2 vwopenClass%rowtype;    begin  select 과정명, 과정기간, 강의실명 into vrow2.과정명, vrow2.과정기간, vrow2.강의실명 from vwopensubject where 과목번호 = pOpenClass;  dbms\_output.put\_line('개설과정: ' || vrow2.과정명 || ' | 과정기간:' || vrow2.과정기간 || ' | 강의실명:' || vrow2.강의실명);  dbms\_output.put\_line('');  for vrow in vcursor loop  dbms\_output.put\_line('교육생번호: ' || vrow.교육생번호 ||' | 교육생 명:'|| vrow.교육생명 || ' | 전화번호' || vrow.전화번호 || ' | 수료여부:' ||vrow.수료여부 || ' | 회사명:' || nvl(vrow.회사명, '미취업자'));  end loop;  exception  when others then  dbms\_output.put\_line('등록된 정보가 없습니다.');  end proc\_seeGetJob;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-07 | 작성일 | 2022-04-07 |
| 요구사항명 | 관리자의 시험 관리 및 성적 조회 | 작성자 | 김수림 |
| DML | | | |
| --성적 프로시저  --과목별 출력시 개설 과정명, 개설 과정기간, 강의실명, 개설 과목명, 교사명, 교재명 등을 출력  select 과정명, 과정기간, 강의실명, 과목명, 교사명, 교재명 from vwopensubject where 과목번호 = 2;  -- 해당 개설 과목을 수강한 모든 교육생들의 성적 정보(교육생 이름, 주민번호 뒷자리, 필기, 실기)를 같이 출력  select bs.name as "개설과목",  st.name as "교육생 이름",  st.ssn as "주민번호 뒷자리",  sc.pilgiScore as "필기점수",  sc.silgiScore as "실기점수"  from tblStudent st  inner join tblScore sc  on sc.s\_seq = st.s\_seq  inner join tblopenSubject os  on os.os\_seq = sc.os\_seq  inner join tblbasicSubject bs  on bs.bs\_seq = os.bs\_seq  where os.os\_seq = 2; --과목 선택  -- 해당 개설 과정을 수강한 든 교육생들의 성적 정보(교육생 이름, 주민번호 뒷자리, 필기, 실기)와 과정정보를 출력  create or replace procedure proc\_seeScore(  pSubjnum number  )  is  cursor vcursor is  select bs.name as "개설과목",  st.name as "교육생명",  sc.pilgiScore as "필기점수",  sc.silgiScore as "실기점수",  sc.attendScore as "출석점수"  from tblStudent st  inner join tblScore sc  on sc.s\_seq = st.s\_seq  inner join tblopenSubject os  on os.os\_seq = sc.os\_seq  inner join tblbasicSubject bs  on bs.bs\_seq = os.bs\_seq  where os.os\_seq = pSubjnum;  vrow2 vwopensubject%rowtype;    begin  select 과정명, 과정기간, 강의실명, 과목명, 교사명, 교재명 into vrow2.과정명, vrow2.과정기간, vrow2.강의실명, vrow2.과목명, vrow2.교사명, vrow2.교재명 from vwopensubject where 과목번호 = pSubjnum;  dbms\_output.put\_line('개설과정: ' || vrow2.과정명 || ' | 과정기간:' || vrow2.과정기간 || ' | 강의실명:' || vrow2.강의실명);  dbms\_output.put\_line('과목명:' || vrow2.과목명 || ' | 교사명:' || vrow2.교사명 || ' | 교재명:' || vrow2.교재명);  dbms\_output.put\_line('');  for vrow in vcursor loop  dbms\_output.put\_line('개설과목: ' || vrow.개설과목 ||' | 교육생 이름:'|| vrow.교육생명 || ' | 필기점수:' || nvl(vrow.필기점수, -1) || ' | 실기점수:' || nvl(vrow.실기점수, -1) || ' | 출석점수:' || nvl(vrow.출석점수, -1));  end loop;  exception  when others then  dbms\_output.put\_line('등록된 점수가 없습니다.');  end proc\_seeScore;  / | | | |