**업무 SQL**

**A. 관리자**

| 요구사항명 | 기초 정보의 등록 및 관리 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 기초 정보관리 | 작성자 | 3조 |
| DML | | | |
| -- ① 기초 정보 추가  -- 기초 과정 정보 추가  EXECUTE procCourseIstAct\_M('테스트과정명2', '테스트 과정목표2', '테스트 과정설명2');  CREATE OR REPLACE PROCEDURE procCourseIst\_M(  pName VARCHAR2,  pGoal VARCHAR2,  pDetail VARCHAR2,  pResult out number  )  IS  pSeq VARCHAR2(10);  BEGIN  SELECT CONCAT('L', lpad(MAX(to\_number(substr(course\_seq, 2)))+1, 3, '0')) INTO pSeq FROM tblCourse;  INSERT INTO tblCourse (course\_seq, name, goal, detail)  VALUES (pSeq, pName, pGoal, pDetail);  pResult := 1;  exception  when others then  pResult := 0;  END procCourseIst\_M;  -- 기초 과목 정보 추가  EXECUTE procSubjectIstAct\_M('테스트과목명', '공통');  CREATE OR REPLACE PROCEDURE procSubjectIst\_M(  pName VARCHAR2,  pType VARCHAR2,  pResult out number  )  IS  pSeq VARCHAR2(10);  BEGIN  SELECT CONCAT('SUB', lpad(MAX(to\_number(substr(subject\_seq, 4)))+1, 3, '0')) INTO pSeq FROM tblSubject;  INSERT INTO tblSubject (subject\_seq, name, type) VALUES (pSeq, pName, pType);  pResult := 1;  exception  when others then  pResult := 0;  END procSubjectIst\_M;  -- 기초 교재 정보 추가  EXECUTE procTextbookIstAct\_M('테스트교재명', '테스트출판사');  CREATE OR REPLACE PROCEDURE procTextbookIst\_M(  pName VARCHAR2,  pPublisher VARCHAR2,  pResult out number  )  IS  pSeq VARCHAR2(10);  BEGIN  SELECT concat('B', lpad(max(to\_number(substr(textbook\_seq, 2)))+1, 3, '0')) INTO pSeq FROM tblTextbook;  INSERT INTO tblTextbook (textbook\_seq, name, publisher) VALUES (pSeq, pName, pPublisher);  pResult := 1;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  END procTextbookIst\_M;  -- 기초 강의실 정보 추가  EXECUTE procClassroomIstAct\_M('10강의실', 20);  CREATE OR REPLACE PROCEDURE procClassroomIst\_M(  pName VARCHAR2,  pCapacity VARCHAR2,  pResult out number  )  IS  pSeq VARCHAR2(10);  BEGIN  SELECT CONCAT('R', lpad(MAX(to\_number(substr(classroom\_seq, 2)))+1, 3, '0')) INTO pSeq FROM tblClassroom;  INSERT INTO tblClassroom (classroom\_seq, name, capacity) VALUES (pSeq, pName, pCapacity);  pResult := 1;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  END procClassroomIst\_M;  -- ② 기초 정보 출력  -- 기초 과정 정보 조회  select course\_seq as "과정코드", name as "과정명", goal as "강의목표", detail as "강의설명" from tblCourse;  -- 기초 과목 정보 조회  select subject\_seq as "과목코드", name as "과목명", type as "과목분류" from tblSubject;  -- 기초 교재 정보 조회  select textbook\_seq as "교재코드", name as "교재명", publisher as "출판사" from tbltextbook;  -- 기초 강의실 정보 조회  select classroom\_seq as "강의실코드", name as "강의실명", capacity as "강의실 정원" from tblClassroom;  -- ③ 기초 정보 수정  -- 3-1)기초 과정 정보 수정  -- 타겟 1: 과정이름 수정 > '과정명 수정'  EXECUTE procCourseSltAct\_M('L018', '1', '과정명 수정');  -- 타겟 2: 과정목표 수정 > '과정목표 수정'  EXECUTE procCourseSltAct\_M('L018', '2', '과정목표 수정');  -- 타겟 3: 과정이름 수정 > '과정설명 수정'  EXECUTE procCourseSltAct\_M('L018', '3', '과정설명 수정');  -- 프로시저  CREATE OR REPLACE PROCEDURE procUpdateCourse\_M(  pTargetSeq VARCHAR2,  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  UPDATE tblCourse SET name=pValue WHERE course\_seq=pTargetSeq;  pResult := 1;  elsif pTargetColumn = '2' THEN  UPDATE tblCourse SET goal=pValue WHERE course\_seq=pTargetSeq;  pResult := 2;  elsif pTargetColumn = '3' THEN  UPDATE tblCourse SET detail=pValue WHERE course\_seq=pTargetSeq;  pResult := 3;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  END procUpdateCourse\_M;  -- 3-2)기초 과목 정보 수정  -- 타겟 1: 과정이름 수정 > '과목명 수정'  EXECUTE procUpdateSubjectAct\_M('SUB041', '1', '수정과목명');  -- 타겟 1: 과정이름 수정 > '과목분류 수정'  EXECUTE procUpdateSubjectAct\_M('SUB041', '2', '수정과목분류');  -- 프로시저  CREATE OR REPLACE PROCEDURE procUpdateSubject\_M(  pTargetSeq VARCHAR2,  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  UPDATE tblSubject SET name=pValue WHERE subject\_seq=pTargetSeq;  pResult := 1;  elsif pTargetColumn = '2' THEN  IF pValue IN ('공통','선택') THEN  UPDATE tblSubject SET type=pValue WHERE subject\_seq=pTargetSeq;  pResult := 2;  ELSE  pResult := -2;  end if;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 과목명 수정  -- 2:타겟 > 과목타입 수정 (공통, 선택)  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procUpdateSubject\_M;  -- 3-3)기초 교재 정보 수정  -- 1:타겟 > 교재명 수정  EXECUTE procUpdateTextbookAct\_M('B122', '1', '수정 교재명');  -- 2:타겟 > 교재 출판사 수정  EXECUTE procUpdateTextbookAct\_M('B122', '2', '수정 출판사');  -- 프로시저  CREATE OR REPLACE PROCEDURE procUpdateTextbook\_M(  pTargetSeq VARCHAR2,  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  UPDATE tblTextbook SET name=pValue WHERE textbook\_seq=pTargetSeq;  pResult := 1;  elsif pTargetColumn = '2' THEN  UPDATE tblTextbook SET publisher=pValue WHERE textbook\_seq=pTargetSeq;  pResult := 2;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 교재명 수정  -- 2:타겟 > 출판사 수정  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procUpdateTextbook\_M;  -- 3-4)기초 강의실 정보 수정  -- 1:타겟 > 강의실 명 수정  EXECUTE procUpdateClassroomAct\_M('R007', '1', '강의실10');  -- 2:타겟 > 강의실 정원 수정  EXECUTE procUpdateClassroomAct\_M('R007', '2', '21');  -- 프로시저  -- 1. 타겟인자 2. 컬럼인자 3. 값인자  CREATE OR REPLACE PROCEDURE procUpdateClassroom\_M(  pTargetSeq VARCHAR2,  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  UPDATE tblClassroom SET name=pValue WHERE classroom\_seq=pTargetSeq;  pResult := 1;  elsif pTargetColumn = '2' THEN  UPDATE tblClassroom SET capacity=pValue WHERE classroom\_seq=pTargetSeq;  pResult := 2;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 강의실 명 수정  -- 2:타겟 > 강의실 정원 수정  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procUpdateClassroom\_M;  -- ④ 기초 정보 삭제  -- 4-1)기초 과정 정보 삭제  -- 1:타겟 > 과정코드로 찾아서 삭제  EXECUTE procCourseDltAct\_M('1', 'L018');  -- 2:타겟 > 과정명으로 찾아서 삭제  EXECUTE procCourseDltAct\_M('2', '테스트과정명2');  -- 프로시저  -- 1. 타겟컬럼 2. 타겟컬럼의 값  CREATE OR REPLACE PROCEDURE procCourseDlt\_M(  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  DELETE FROM tblCourse WHERE course\_seq=pValue;  pResult := 1;  elsif pTargetColumn = '2' THEN  DELETE FROM tblCourse WHERE name=pValue;  pResult := 2;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 과정코드로 찾아서 삭제  -- 2:타겟 > 과정명으로 찾아서 삭제  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procCourseDlt\_M;  -- 4-2)기초 과목 정보 삭제  -- 1:타겟 > 과목코드로 찾아서 삭제  EXECUTE procSubjectDltAct\_M('1', 'SUB041');  -- 2:타겟 > 과목명으로 찾아서 삭제  EXECUTE procSubjectDltAct\_M('2', '테스트과목명');  -- 프로시저  CREATE OR REPLACE PROCEDURE procSubjectDlt\_M(  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  DELETE FROM tblSubject WHERE subject\_seq=pValue;  pResult := 1;  elsif pTargetColumn = '2' THEN  DELETE FROM tblSubject WHERE name=pValue;  pResult := 2;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 과목코드로 찾아서 삭제  -- 2:타겟 > 과목명으로 찾아서 삭제  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procSubjectDlt\_M;  -- 4-3)기초 교재 정보 삭제  -- 1:타겟 > 교재코드로 찾아서 삭제  EXECUTE procTextbookDltAct\_M('1', 'B122');  -- 2:타겟 > 교재명으로 찾아서 삭제  EXECUTE procTextbookDltAct\_M('2', '교재명');  -- 프로시저  CREATE OR REPLACE PROCEDURE procTextbookDlt\_M(  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  DELETE FROM tblTextbook WHERE textbook\_seq=pValue;  pResult := 1;  elsif pTargetColumn = '2' THEN  DELETE FROM tblTextbook WHERE name=pValue;  pResult := 2;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 교재코드로 찾아서 삭제  -- 2:타겟 > 교재명으로 찾아서 삭제  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procTextbookDlt\_M;  -- 4-4)기초 강의실 정보 삭제  -- 1:타겟 > 강의실코드로 찾아서 삭제  EXECUTE procClassroomDltAct\_M('1', 'R007');  -- 2:타겟 > 강의실명으로 찾아서 삭제  EXECUTE procClassroomDltAct\_M('2', '강의실9');  -- 프로시저  CREATE OR REPLACE PROCEDURE procClassroomDlt\_M(  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  DELETE FROM tblClassroom WHERE classroom\_seq=pValue;  pResult := 1;  elsif pTargetColumn = '2' THEN  DELETE FROM tblClassroom WHERE name=pValue;  pResult := 2;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 강의실코드로 찾아서 삭제  -- 2:타겟 > 강의실명으로 찾아서 삭제  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procClassroomDlt\_M; | | | |

| 요구사항명 | 교사 정보의 등록 및 관리 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 교사 계정관리 | 작성자 | 3조 |
| DML | | | |
| -- ① 로그인 기능 프로시저  EXECUTE procLogin\_All('아이디(이름)','비밀번호','교사or학생or관리자');  -- ex) 교사 로그인하기  EXECUTE procLogin\_All('이윤후','2482932','교사');  -- ex) 학생 로그인하기  EXECUTE procLogin\_All('정하하','2654210','학생');  -- ex) 관리자 로그인하기  EXECUTE procLogin\_All('qkcu5302','adlj8683','관리자');  -- 프로시저  CREATE OR REPLACE PROCEDURE procLogin\_All(  pId VARCHAR2,  pPw VARCHAR2,  pType VARCHAR2  )  IS  loginValid NUMBER;  name VARCHAR(30);  BEGIN  loginValid:=0;  IF pType='학생' THEN  SELECT COUNT(\*) INTO loginValid FROM tblstudent WHERE name=pId AND idcard\_number=pPw;  IF loginValid=1 THEN  dbms\_output.put\_line(pType||' 로그인에 성공하였습니다.');  dbms\_output.put\_line('안녕하세요. '||pId||'님');  ELSE  dbms\_output.put\_line('로그인에 실패하였습니다.');  dbms\_output.put\_line('아이디와 비밀번호를 확인하세요.');  END IF;  ELSIF pType='교사' THEN  SELECT COUNT(\*) INTO loginValid FROM tblTeacher WHERE name=pId AND idcard\_number=pPw;  IF loginValid=1 THEN  dbms\_output.put\_line(pType||' 로그인에 성공하였습니다.');  dbms\_output.put\_line('안녕하세요. '||pId||'님');  ELSE  dbms\_output.put\_line('로그인에 실패하였습니다.');  dbms\_output.put\_line('아이디와 비밀번호를 확인하세요.');  END IF;  ELSIF pType='관리자' THEN  SELECT COUNT(\*) INTO loginValid FROM tblmanager WHERE id=pId AND pw=pPw;  IF loginValid=1 THEN  dbms\_output.put\_line(pType||' 로그인에 성공하였습니다.');  dbms\_output.put\_line('안녕하세요. '||pId||'님');  ELSE  dbms\_output.put\_line('로그인에 실패하였습니다.');  dbms\_output.put\_line('아이디와 비밀번호를 확인하세요.');  END IF;  ELSE  dbms\_output.put\_line('로그인 유형 "학생", "교사", "관리자" 중 하나를 골라 세 번째 값에 입력하시오.');  END IF;  END procLogin\_All;  -- ② 교사 등록하기  EXECUTE procTeacherIstAct\_M('이름', '주민번호 뒷자리', '연락처');  -- ex  EXECUTE procTeacherIstAct\_M('김연중', '1111111', '010-1234-5678');  -- 프로시저  CREATE OR REPLACE PROCEDURE procTeacherIst\_M(  pName VARCHAR2,  pIdcard\_number VARCHAR2,  pTel VARCHAR2,  pResult out number  )  IS  pSeq VARCHAR2(10);  BEGIN  SELECT CONCAT('T', lpad(MAX(to\_number(substr(teacher\_seq, 2)))+1, 3, '0')) INTO pSeq FROM tblTeacher;  INSERT INTO tblteacher (teacher\_seq, name, idcard\_number, tel)  VALUES (pSeq, pName, pIdcard\_number, pTel);  pResult := 1;  exception  when others then  pResult := 0;  END procTeacherIst\_M;  -- ③ 교사 정보 조회하기  SELECT teacher\_seq as "강사코드", name as "이름", idcard\_number as "주민번호 뒷자리", tel as "연락처" FROM tblTeacher;  -- ④ 교사 정보 수정하기  EXECUTE procUpdateTeacherAct\_M('바꿀 타겟 시퀀스', '1 or 2 or 3', '값');  -- ex) 이름 바꾸기  EXECUTE procUpdateTeacherAct\_M('T011', '1', '이연중');  -- ex) 주민번호 뒷자리 바꾸기  EXECUTE procUpdateTeacherAct\_M('T011', '2', '1234567');  -- ex) 연락처 바꾸기  EXECUTE procUpdateTeacherAct\_M('T011', '3', '010-0000-0000');  -- 프로시저  CREATE OR REPLACE PROCEDURE procUpdateTeacher\_M(  pTargetSeq VARCHAR2,  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  IF pTargetColumn='1' THEN  UPDATE tblTeacher SET name=pValue WHERE teacher\_seq=pTargetSeq;  pResult:=1;  elsif pTargetColumn = '2' THEN  UPDATE tblTeacher SET idcard\_number=pValue WHERE teacher\_seq=pTargetSeq;  pResult:=2;  elsif pTargetColumn = '3' THEN  UPDATE tblTeacher SET tel=pValue WHERE teacher\_seq=pTargetSeq;  pResult:=3;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 이름 수정  -- 2:타겟 > 주민번호 뒷자리 수정  -- 3:타겟 > 연락처 수정  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procUpdateTeacher\_M;  -- ⑤ 교사 정보 삭제하기  EXECUTE procTeacherDltAct\_M('1 or 2', '목표값');  -- ex) 시퀀스로 검색 후 삭제하기  EXECUTE procTeacherDltAct\_M('1', 'T011');  -- ex) 이름으로 검색 후 삭제하기  EXECUTE procTeacherDltAct\_M('2', '김연중');  -- 프로시저  CREATE OR REPLACE PROCEDURE procTeacherDlt\_M(  pTargetColumn VARCHAR2,  pValue VARCHAR2,  pResult out number  )  IS  BEGIN  if pTargetColumn = '1' THEN  DELETE FROM tblTeacher WHERE teacher\_seq=pValue;  pResult := 1;  elsif pTargetColumn = '2' THEN  DELETE FROM tblTeacher WHERE name=pValue;  pResult := 2;  else  pResult := -1;  end if;  EXCEPTION  WHEN OTHERS THEN  pResult := 0;  -- 1:타겟 > 교사코드로 찾아서 삭제  -- 2:타겟 > 이름으로 찾아서 삭제  -- -1: 1~3 외에 다른 값을 입력 (오류)  -- 0: -1오류를 제외한 알 수 없는 오류  END procTeacherDlt\_M;  -- ⑥ 전체 교사 목록 출력하기 (강의 가능 과목까지 전부)  SELECT \* FROM vwAllTeacher;  -- ⑦ 단일 교사 정보 출력하기  SELECT  b.name AS "교사명",  c.name AS "과목명",  a.subject\_start\_date AS "과목시작날짜",  a.subject\_end\_date AS "과목종료날짜",  f.name AS "과정명",  d.course\_start\_date AS "과정시작일",  d.course\_end\_date AS "과정종료일",  g.name AS "교재명", e.name AS "강의실",  d.course\_progress AS "강의진행여부"  FROM tblOpenSubject a  INNER JOIN tblTeacher b ON a.teacher\_seq=b.teacher\_seq  INNER JOIN tblSubject c ON a.subject\_seq=c.subject\_seq  INNER JOIN tblOpenCourse d ON a.open\_course\_seq=d.open\_course\_seq  INNER JOIN tblClassroom e ON d.classroom\_seq=e.classroom\_seq  INNER JOIN tblcourse f ON d.course\_seq=f.course\_seq  INNER JOIN tblTextbook g ON a.textbook\_seq=g.textbook\_seq  WHERE b.teacher\_seq='T001';--교사코드 입력 | | | |

| 요구사항명 | 개설 과정의 등록 및 관리 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 개설 과정 관리 | 작성자 | 3조 |
| DML | | | |
| --① <개설 과정 입력>  EXECUTE procNewCourseIst\_M('자바 기술자 과정', '자바 전문가가 된다.', '자바 전문가가 되는 과정입니다.', '과정시작일', '과정종료일', '과목등록여부', '강의진행여부', '강의실Seq');  -- ex  EXECUTE procNewCourseIst\_M('테스트 과정명', '테스트 과정 목표', '테스트 과정 설명', '2021-12-05', '2022-03-20', 'Y', '강의예정', 'R006');  -- 프로시저  CREATE OR REPLACE PROCEDURE procNewCourseIst\_M(  pName VARCHAR2,  pGoal VARCHAR2,  pDetail VARCHAR2,  pCourse\_start\_date VARCHAR2,  pCourse\_end\_date VARCHAR2,  pReg\_open\_subject CHAR,  pCourse\_progress VARCHAR2,  pClassroom\_seq VARCHAR2  )  IS  pCourse\_seq VARCHAR(30);  pOpen\_course\_seq VARCHAR(30);  BEGIN  select concat('L', lpad(max(to\_number(substr(course\_seq, 2)))+1, 3, '0')) INTO pCourse\_seq from tblCourse;  select concat('OL', lpad(max(to\_number(substr(open\_course\_seq, 3)))+1, 3, '0')) INTO pOpen\_course\_seq from tblOpenCourse;  INSERT INTO tblcourse (course\_seq, name, goal, detail)  VALUES (pCourse\_seq, pName, pGoal, pDetail);  INSERT INTO tblOpenCourse (  open\_course\_seq,  course\_start\_date,  course\_end\_date,  reg\_open\_subject,  course\_progress,  classroom\_seq,  course\_seq)  VALUES (  pOpen\_course\_seq,  TO\_DATE(pCourse\_start\_date, 'yyyy-mm-dd'),  TO\_DATE(pCourse\_end\_date, 'yyyy-mm-dd'),  pReg\_open\_subject,  pCourse\_progress,  pClassroom\_seq,  pCourse\_seq);  END procNewCourseIst\_M;  --② 모든 개설 과정 정보 출력 ★  select \* from vwAllCourse;  --③ 특정 개설 과정 정보 출력  --4-1) 등록된 개설 과목 정보 ★  SELECT b.name AS "과목명", a.subject\_start\_date AS "과목시작일", a.subject\_end\_date AS "과목종료일", d.name AS "교재명", c.name AS "교사명"  FROM tblOpenSubject a  INNER JOIN tblSubject b ON a.subject\_seq=b.subject\_seq  INNER JOIN tblTeacher c ON a.teacher\_seq=c.teacher\_seq  INNER JOIN tblTextbook d ON a.textbook\_seq=d.textbook\_seq  WHERE a.open\_course\_seq='OL007';--과정코드 입력  --4-2) 등록된 교육생 정보 ★  SELECT b.name AS "교육생 이름", b.idcard\_number AS "주민번호 뒷자리", b.tel AS "전화번호", b.registration\_date AS "등록일", a.completion\_status AS "수료여부", a.completion\_date AS "수료날짜", a.dropout\_date AS "중도탈락날짜"  FROM tblRegister a  INNER JOIN tblStudent b ON a.student\_seq=b.student\_seq  WHERE a.open\_course\_seq='OL007';--과정코드 입력 | | | |

| 요구사항명 | 개설과목의 등록 및 관리 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 개설 과목 관리 | 작성자 | 3조 |
| DML | | | |
| --특정과정(OL006)에 새로운 과목을 개설 할 수 있다.  --프로시저 정의  CREATE OR REPLACE PROCEDURE *procAddOpenSubject\_M*(  P\_SUBJECT\_SEQ VARCHAR2,  P\_TEACHER\_SEQ VARCHAR2,  P\_TEXTBOOK\_SEQ VARCHAR2,  P\_SUBJECT\_START\_DATE VARCHAR2,  P\_SUBJECT\_END\_DATE VARCHAR2,  P\_SUBJECT\_PROGRESS VARCHAR2,  P\_SCORE\_CHECK CHAR,  P\_TESTFILE\_CHECK CHAR,  P\_ATTENDANCE\_DISTRIBUTION NUMBER,  P\_WRITTEN\_DISTRIBUTION NUMBER,  P\_PRACTICAL\_DISTRIBUTION NUMBER,  P\_OPEN\_CUORSE\_SEQ VARCHAR2,  P\_RESULT OUT NUMBER --성공(1) or 실패(0)  )  IS  -- pseq 변수 하나 만들어서 SELECT 문에 INTO 사용해야함  PSEQ VARCHAR2(10);  BEGIN  --1.  SELECT CONCAT('OS', LPAD(MAX(SUBSTR(OPEN\_SUBJECT\_SEQ, 4, 2)) + 1, 3, '0')) INTO PSEQ FROM TBLOPENSUBJECT;  INSERT INTO TBLOPENSUBJECT (OPEN\_SUBJECT\_SEQ, SUBJECT\_SEQ, TEACHER\_SEQ, TEXTBOOK\_SEQ, SUBJECT\_START\_DATE,  SUBJECT\_END\_DATE, SUBJECT\_PROGRESS, SCORE\_CHECK, TESTFILE\_CHECK,  ATTENDANCE\_DISTRIBUTION,  WRITTEN\_DISTRIBUTION, PRACTICAL\_DISTRIBUTION, OPEN\_COURSE\_SEQ)  VALUES (PSEQ,  P\_SUBJECT\_SEQ,  P\_TEACHER\_SEQ,  P\_TEXTBOOK\_SEQ,  P\_SUBJECT\_START\_DATE,  P\_SUBJECT\_END\_DATE,  P\_SUBJECT\_PROGRESS,  P\_SCORE\_CHECK,  P\_TESTFILE\_CHECK,  P\_ATTENDANCE\_DISTRIBUTION,  P\_WRITTEN\_DISTRIBUTION,  P\_PRACTICAL\_DISTRIBUTION,  P\_OPEN\_CUORSE\_SEQ);  P\_RESULT := 1;  EXCEPTION  WHEN OTHERS THEN  P\_RESULT := 0;  END *procAddOpenSubject\_M*;  --\*\*\*\*\*\*\*\*\*\* 추가한 개설과목번호 삭제프로시저 \*\*\*\*\*\*\*\*\*\*--  DECLARE  V\_RESULT NUMBER;  BEGIN  *ProcDeleteOpenSubject\_M*('OS082', V\_RESULT);  IF V\_RESULT = 1 THEN  DBMS\_OUTPUT.*PUT\_LINE*('삭제완료');  else  DBMS\_OUTPUT.*PUT\_LINE*('삭제실패');  end if;  end;  --\*\*\*\*\*\*\*\*\*\* 교사 변경 프로시저 \*\*\*\*\*\*\*\*\*\*--  DECLARE  V\_RESULT NUMBER;  BEGIN  *ProcUpdateOpenSubject\_M*('T009', 'OS081', V\_RESULT);  IF V\_RESULT = 1 THEN  DBMS\_OUTPUT.*PUT\_LINE*('수정성공');  ELSE  DBMS\_OUTPUT.*PUT\_LINE*('수정실패');  end if;  end; | | | |

| 요구사항명 | 교육생 등록 및 관리 | 작성일 | 2021-21-07 |
| --- | --- | --- | --- |
| System | 교육생관리 | 작성자 | 3조 |
| DML | | | |
| --\*\*\*\*\*\*\*\*\*\* 교육생 추가 정의 프로시저 \*\*\*\*\*\*\*\*\*\*--  CREATE OR REPLACE PROCEDURE *ProcAddStudent\_M*(  P\_NAME VARCHAR2,  P\_IDCARD\_NUMBER VARCHAR2,  P\_TEL VARCHAR2,  P\_RESULT OUT NUMBER  )  IS  PSEQ VARCHAR2(10);  BEGIN  SELECT CONCAT('S', LPAD(MAX(SUBSTR(STUDENT\_SEQ, 3, 3)) + 1, 4, '0')) INTO PSEQ FROM TBLSTUDENT;  INSERT INTO TBLSTUDENT (STUDENT\_SEQ, NAME, IDCARD\_NUMBER, TEL, REGISTRATION\_DATE)  VALUES (PSEQ,  P\_NAME,  P\_IDCARD\_NUMBER,  P\_TEL,  TO\_CHAR(SYSDATE, 'RRRR-MM-DD'));  P\_RESULT := 1;  EXCEPTION  WHEN OTHERS THEN  P\_RESULT := 0;  END *ProcAddStudent\_M*;  --\*\*\*\*\*\*\*\*\*\* 중도탈락 프로시저 \*\*\*\*\*\*\*\*\*\*--  CREATE OR REPLACE PROCEDURE *ProcUpdateStudentDropoutDate\_M*(  P\_STUDENT\_SEQ VARCHAR2,  P\_OPEN\_COURSE\_SEQ VARCHAR2,  P\_COMPLETION\_STATUS CHAR,  P\_DROPOUT\_DATE VARCHAR2,  P\_RESULT OUT NUMBER  )  IS  BEGIN  UPDATE TBLREGISTER  SET COMPLETION\_STATUS= P\_COMPLETION\_STATUS  WHERE STUDENT\_SEQ = P\_STUDENT\_SEQ AND OPEN\_COURSE\_SEQ = P\_OPEN\_COURSE\_SEQ;  UPDATE TBLREGISTER  SET DROPOUT\_DATE = P\_DROPOUT\_DATE  WHERE STUDENT\_SEQ = P\_STUDENT\_SEQ AND OPEN\_COURSE\_SEQ = P\_OPEN\_COURSE\_SEQ;  P\_RESULT := 1;  EXCEPTION  WHEN OTHERS THEN  P\_RESULT := 0;  END *ProcUpdateStudentDropoutDate\_M*;  --개설 과정 정보(과정명, 과정기간(시작,끝),강의실, 수료 및 중도탈락 여부, 수료 및 중도탈락 날짜) 출력  SELECT COUR.NAME AS 과정명,  OPENCOUR.COURSE\_START\_DATE AS 과정시작날짜,  OPENCOUR.COURSE\_END\_DATE AS 과정종료날짜,  ROOM.NAME AS 강의실,  REG.COMPLETION\_STATUS AS 수료여부,  REG.COMPLETION\_DATE AS 수료날짜,  REG.DROPOUT\_DATE AS 중도탈락날짜  FROM TBLSTUDENT STU  INNER JOIN TBLREGISTER REG  on STU.STUDENT\_SEQ = REG.STUDENT\_SEQ  INNER JOIN TBLOPENCOURSE OPENCOUR  on REG.OPEN\_COURSE\_SEQ = OPENCOUR.OPEN\_COURSE\_SEQ  INNER JOIN TBLCOURSE COUR  on OPENCOUR.COURSE\_SEQ = COUR.COURSE\_SEQ  INNER JOIN TBLCLASSROOM ROOM  on OPENCOUR.CLASSROOM\_SEQ = ROOM.CLASSROOM\_SEQ  WHERE STU.STUDENT\_SEQ = 'S0106'; | | | |

| 요구사항명 | 시험 및 성적 관리 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 시험 관리 및 성적 조회 | 작성자 | 3조 |
| DML | | | |
| -- 특정 개설 과정을 선택  SELECT OPENSUB.OPEN\_SUBJECT\_SEQ AS 개설과목번호,  SUB.NAME AS 과목명,  OPENSUB.SCORE\_CHECK AS 성적등록여부,  OPENSUB.TESTFILE\_CHECK AS 시험문제파일등록여부  FROM TBLOPENCOURSE OPENCOUR  INNER JOIN TBLOPENSUBJECT OPENSUB  on OPENCOUR.OPEN\_COURSE\_SEQ = OPENSUB.OPEN\_COURSE\_SEQ  INNER JOIN TBLSUBJECT SUB  on OPENSUB.SUBJECT\_SEQ = SUB.SUBJECT\_SEQ  INNER JOIN TBLCOURSE COUR on OPENCOUR.COURSE\_SEQ = COUR.COURSE\_SEQ  WHERE OPENCOUR.OPEN\_COURSE\_SEQ = 'OL001';  --과목별출력시  SELECT DISTINCT COUR.NAME AS 개설과정명,  OPENCOUR.COURSE\_START\_DATE AS 과정시작날짜,  OPENCOUR.COURSE\_END\_DATE AS 과정종료날짜,  ROOM.NAME AS 강의실명,  SUB.NAME AS 개설과목명,  TC.NAME AS 교사명,  TXT.NAME AS 교재명,  STU.NAME AS 교육생이름,  STU.IDCARD\_NUMBER AS 주민번호뒷자리,  SC.ATTENDANCE\_SCORE AS 출결점수,  ROUND(SC.WRITTEN\_SCORE, 1) AS 필기점수,  ROUND(SC.PRACTICAL\_SCORE, 1) AS 실기점수,  ROUND(SC.SUBJECT\_SCORE, 1) AS 과목점수  FROM TBLSCOREINFO SC  INNER JOIN TBLOPENSUBJECT OPENSUB  on SC.OPENSUBJECT\_SEQ = OPENSUB.OPEN\_SUBJECT\_SEQ  INNER JOIN TBLREGISTER REG  on SC.STUDENT\_SEQ = REG.STUDENT\_SEQ  INNER JOIN TBLOPENCOURSE OPENCOUR  on OPENSUB.OPEN\_COURSE\_SEQ = OPENCOUR.OPEN\_COURSE\_SEQ  INNER JOIN TBLSTUDENT STU  on REG.STUDENT\_SEQ = STU.STUDENT\_SEQ  INNER JOIN TBLCOURSE COUR  on OPENCOUR.COURSE\_SEQ = COUR.COURSE\_SEQ  INNER JOIN TBLSUBJECT SUB  on OPENSUB.SUBJECT\_SEQ = SUB.SUBJECT\_SEQ  INNER JOIN TBLCLASSROOM ROOM  on OPENCOUR.CLASSROOM\_SEQ = ROOM.CLASSROOM\_SEQ  INNER JOIN TBLTEXTBOOK TXT  on OPENSUB.TEXTBOOK\_SEQ = TXT.TEXTBOOK\_SEQ  INNER JOIN TBLTEACHER TC  on OPENSUB.TEACHER\_SEQ = TC.TEACHER\_SEQ  WHERE SC.OPENSUBJECT\_SEQ = 'OS001';  --교육생개인별  SELECT DISTINCT COUR.NAME AS 개설과정명,  OPENCOUR.COURSE\_START\_DATE AS 과정시작날짜,  OPENCOUR.COURSE\_END\_DATE AS 과정종료날짜,  ROOM.NAME AS 강의실명,  SUB.NAME AS 개설과목명,  TC.NAME AS 교사명,  TXT.NAME AS 교재명,  STU.NAME AS 교육생이름,  STU.IDCARD\_NUMBER AS 주민번호뒷자리,  SC.ATTENDANCE\_SCORE AS 출결점수,  ROUND(SC.WRITTEN\_SCORE, 1) AS 필기점수,  ROUND(SC.PRACTICAL\_SCORE, 1) AS 실기점수,  ROUND(SC.SUBJECT\_SCORE, 1) AS 과목점수  FROM TBLSCOREINFO SC  INNER JOIN TBLOPENSUBJECT OPENSUB  on SC.OPENSUBJECT\_SEQ = OPENSUB.OPEN\_SUBJECT\_SEQ  INNER JOIN TBLREGISTER REG on SC.STUDENT\_SEQ = REG.STUDENT\_SEQ  INNER JOIN TBLOPENCOURSE OPENCOUR on OPENSUB.OPEN\_COURSE\_SEQ = OPENCOUR.OPEN\_COURSE\_SEQ  INNER JOIN TBLSTUDENT STU on REG.STUDENT\_SEQ = STU.STUDENT\_SEQ  INNER JOIN TBLCOURSE COUR on OPENCOUR.COURSE\_SEQ = COUR.COURSE\_SEQ  INNER JOIN TBLSUBJECT SUB on OPENSUB.SUBJECT\_SEQ = SUB.SUBJECT\_SEQ  INNER JOIN TBLCLASSROOM ROOM on OPENCOUR.CLASSROOM\_SEQ = ROOM.CLASSROOM\_SEQ  INNER JOIN TBLTEACHER TC on OPENSUB.TEACHER\_SEQ = TC.TEACHER\_SEQ  INNER JOIN TBLTEXTBOOK TXT on OPENSUB.TEXTBOOK\_SEQ = TXT.TEXTBOOK\_SEQ  WHERE SC.STUDENT\_SEQ = 'S0381'; | | | |

| 요구사항명 | 교육생 출결 관리 및 출결 조회 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 출결 관리 및 출결 조회 | 작성자 | 3조 |
| DML | | | |
| -- 출결관리 및 출결조회  -- procAttendanceManagementAct('과정코드', '0', '0') > 전체검색  -- procAttendanceManagementAct('과정코드', '이름', '0') > 이름검색  -- procAttendanceManagementAct('과정코드', '시작날짜', '종료날짜') > 날짜검색  -- ex 전체검색  EXECUTE procAttendanceManagementAct('OL007', '0', '0');  -- ex 이름검색  EXECUTE procAttendanceManagementAct('OL007', '이시조', '0');  -- ex 날짜검색  EXECUTE procAttendanceManagementAct('OL007', '2021-01-01', '2021-01-31');  -- 프로시저  create or replace procedure procAttendanceManagement(  pOpen\_course\_seq varchar2,  pValue1 VARCHAR2,  pValue2 VARCHAR2,  presult out sys\_refcursor  )  is  begin  IF pValue1='0' AND pValue2='0' THEN  open presult for  SELECT DISTINCT d.attendance\_date AS "날짜", c.name AS "교육생 이름", e.attendance\_type AS "근태 상황"  FROM tblOpenSubject a  INNER JOIN tblRegister b ON a.open\_course\_seq = b.open\_course\_seq  INNER JOIN tblStudent c ON b.student\_seq = c.student\_seq  INNER JOIN tblAttendance d ON a.open\_course\_seq = d.open\_course\_seq AND b.student\_seq = d.student\_seq  INNER JOIN tblAttendanceType e ON d.attendance\_type\_seq = e.attendance\_type\_seq  WHERE a.open\_course\_seq=pOpen\_course\_seq -- 과정코드만 입력  ORDER BY d.attendance\_date;  ELSIF pValue2='0' THEN  open presult for  SELECT DISTINCT d.attendance\_date AS "날짜", c.name AS "교육생 이름", e.attendance\_type AS "근태 상황"  FROM tblOpenSubject a  INNER JOIN tblRegister b ON a.open\_course\_seq = b.open\_course\_seq  INNER JOIN tblStudent c ON b.student\_seq = c.student\_seq  INNER JOIN tblAttendance d ON a.open\_course\_seq = d.open\_course\_seq AND b.student\_seq = d.student\_seq  INNER JOIN tblAttendanceType e ON d.attendance\_type\_seq = e.attendance\_type\_seq  WHERE a.open\_course\_seq=pOpen\_course\_seq AND c.name=pValue1 --과정코드, 이름 입력  ORDER BY d.attendance\_date;  ELSE  open presult for  SELECT DISTINCT d.attendance\_date AS "날짜", c.name AS "교육생 이름", e.attendance\_type AS "근태 상황"  FROM tblOpenSubject a  INNER JOIN tblRegister b ON a.open\_course\_seq = b.open\_course\_seq  INNER JOIN tblStudent c ON b.student\_seq = c.student\_seq  INNER JOIN tblAttendance d ON a.open\_course\_seq = d.open\_course\_seq AND b.student\_seq = d.student\_seq  INNER JOIN tblAttendanceType e ON d.attendance\_type\_seq = e.attendance\_type\_seq  WHERE a.open\_course\_seq=pOpen\_course\_seq AND d.attendance\_date BETWEEN TO\_DATE(pValue1, 'YYYY-MM-DD') AND TO\_DATE(pValue2, 'YYYY-MM-DD')--과정코드, 기간 입력  ORDER BY d.attendance\_date; --과정코드, 날짜 입력  END IF;  end procAttendanceManagement; | | | |

**B. 교사**

| 테이블명 | 교사 강의 스케줄 조회 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 강의스케줄 조회 | 작성자 | 3조 |
| DML | | | |
| **-- 1.1) 교사별 > 강의스케줄 출력**  select  os.open\_subject\_seq as 개설과목번호,  course.name as 과정명,  ol.course\_start\_date as 과정시작일,  ol.course\_end\_date as 과정종료일,  room.name as 강의실명,  sub.name as 과목명,  os.subject\_start\_date as 과목시작일,  os.subject\_end\_date as 과목종료일,  os.subject\_progress as 과목강의상태,  book.name as 교재명,  (select count(\*) from tblregister where open\_course\_seq = os.open\_course\_seq and dropout\_date is null) || '명' as "교육생 등록 인원"  from tblopensubject os  inner join tblsubject sub on os.subject\_seq = sub.subject\_seq  inner join tblopencourse ol on ol.open\_course\_seq = os.open\_course\_seq  inner join tblcourse course on ol.course\_seq = course.course\_seq  inner join tblclassroom room on ol.classroom\_seq = room.classroom\_seq  inner join tbltextbook book on os.textbook\_seq = book.textbook\_seq  where os.teacher\_seq = 'T001'  order by ol.course\_start\_date, os.subject\_start\_date;  **-- 1.2) 특정 과목번호 선택 > 해당 과정 교육생 정보 출력**  select  os.open\_subject\_seq as 개설과목번호,  student.name as "교육생 이름",  student.tel as 전화번호,  student.registration\_date as 등록일,  case  when reg.completion\_status = 'Y' then '수료'  when reg.completion\_status = 'N' then '중도탈락'  when current\_date < ol.course\_start\_date then '과정미시작'  else '과정진행중'  end as 수료여부  from tblopensubject os  inner join tblregister reg on os.open\_course\_seq = reg.open\_course\_seq  inner join tblopencourse ol on reg.open\_course\_seq = ol.open\_course\_seq  inner join tblstudent student on reg.student\_seq = student.student\_seq  where os.open\_subject\_seq = 'OS032'  order by student.student\_seq; | | | |

| 테이블명 | 교사 배점 입출력 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 배점 입출력 | 작성자 | 3조 |
| DML | | | |
| **-- 2.1 ) 교사별 > '강의종료' 과목 출력**  select  os.open\_subject\_seq as 개설과목번호,  course.name as 과정명,  ol.course\_start\_date as 과정시작일,  ol.course\_end\_date as 과정종료일,  room.name as 강의실명,  sub.name as 과목명,  os.subject\_start\_date as 과목시작일,  os.subject\_end\_date as 과목종료일,  os.subject\_progress as 과목강의상태,  book.name as 교재명,  os.attendance\_distribution as 출결배점,  os.written\_distribution as 필기배점,  os.practical\_distribution as 실기배점,  os.testfile\_check as 시험문제등록여부  from tblopensubject os  inner join tblsubject sub on os.subject\_seq = sub.subject\_seq  inner join tblopencourse ol on ol.open\_course\_seq = os.open\_course\_seq  inner join tblcourse course on ol.course\_seq = course.course\_seq  inner join tblclassroom room on ol.classroom\_seq = room.classroom\_seq  inner join tbltextbook book on os.textbook\_seq = book.textbook\_seq  where os.teacher\_seq = 'T001'  and os.subject\_progress = '강의종료'  order by ol.course\_start\_date, os.subject\_start\_date;  **-- 2.2) 특정 과목번호 선택 > 배점 및 시험날짜/문제 입력 화면**  select  os.open\_subject\_seq as 개설과목번호,  os.attendance\_distribution as 출결배점,  os.written\_distribution as 필기배점,  os.practical\_distribution as 실기배점,  os.testfile\_check as 시험문제등록여부,  test.test\_seq as 시험번호,  test.test\_date as 시험날짜,  test.type as 시험유형  from tblopensubject os  inner join tblTest test on os.open\_subject\_seq = test.open\_subject\_seq  where os.open\_subject\_seq = 'OS001';  **----------- 2.3.1) 출결배점 수정(입력)\_프로시저**  create or replace procedure procUpdateAtPoint\_T(  popen\_subject\_seq varchar2,  pattendance\_distribution number,  presult out number  )  is  vtotal number;  begin    -- 1) Check the total distribution  select sum(pattendance\_distribution + written\_distribution + practical\_distribution)  into vtotal from tblopensubject  where open\_subject\_seq = popen\_subject\_seq;    -- 2) Update at\_distribution  if vtotal <= 100 then  update tblopensubject  set attendance\_distribution = pattendance\_distribution  where open\_subject\_seq = popen\_subject\_seq;    presult := 1;  else  presult := 0;  dbms\_output.put\_line('배점의 총점은 100을 넘을 수 없습니다.');  end if;    exception  when others then  presult := 0;    end procUpdateAtPoint\_T;  **----------- 2.3.2) 필기배점 수정(입력)\_프로시저**  create or replace procedure procUpdateWtPoint\_T(  popen\_subject\_seq varchar2,  pwritten\_distribution number,  presult out number  )  is  vtotal number;  begin    -- 1) Check the total distribution  select sum(attendance\_distribution + pwritten\_distribution + practical\_distribution)  into vtotal from tblopensubject  where open\_subject\_seq = popen\_subject\_seq;    -- 2) Update wt\_distribution  if vtotal <= 100 then  update tblopensubject  set written\_distribution = pwritten\_distribution  where open\_subject\_seq = popen\_subject\_seq;    presult := 1;  else  presult := 0;  dbms\_output.put\_line('배점의 총점은 100을 넘을 수 없습니다.');  end if;  exception  when others then  presult := 0;  end procUpdateWtPoint\_T;  **----------- 2.3.3) 실기배점 수정(입력)**  create or replace procedure procUpdatePtPoint\_T(  popen\_subject\_seq varchar2,  ppractical\_distribution number,  presult out number  )  is  vtotal number;  begin    -- Check the total distribution  select sum(attendance\_distribution + written\_distribution + ppractical\_distribution)  into vtotal from tblopensubject  where open\_subject\_seq = popen\_subject\_seq;    -- Update pt\_distribution  if vtotal <= 100 then  update tblopensubject  set practical\_distribution = ppractical\_distribution  where open\_subject\_seq = popen\_subject\_seq;    presult := 1;  else  presult := 0;  dbms\_output.put\_line('배점의 총점은 100을 넘을 수 없습니다.');  end if;  exception  when others then  presult := 0;    end procUpdatePtPoint\_T;  **----------- 2.3.4.1) 시험 등록 (기존에 시험 등록을 안한 경우)**  create or replace PROCEDURE procAddTest\_T(  ptest\_date date,  ptest\_type varchar2,  popen\_subject\_seq varchar2,  presult out number  )  is  vcount number;  begin  -- 1) Counting test records  select count(\*) into vcount  from tbltest  where open\_subject\_seq = popen\_subject\_seq and type = ptest\_type;      -- 2) if there's no test record > insert query  if vcount = 0 then  insert into tbltest (test\_seq, test\_date, type, open\_subject\_seq) values (  (select concat('TEST', lpad(max(to\_number(substr(test\_seq, 5)))+1, 4, '0')) from tbltest),  ptest\_date,  ptest\_type,  popen\_subject\_seq  );    -- Update testfile\_check column  procUpdateTestFileCheck\_T(popen\_subject\_seq);  presult := 1;    else  presult := 0;  dbms\_output.put\_line('이미 시험이 등록 되어있습니다.');  end if;  exception  when others then  presult := 0;  end procAddTest\_T;  **-- 2.3.4.2) '시험문제파일등록여부' 컬럼 변경 (해당 과목의 시험이 필기/실기 둘 다 등록 됐을 경우)**  create or replace PROCEDURE procUpdateTestFileCheck\_T (  popen\_subject\_seq varchar2  )  is  vcount number;  begin  -- 1) Counting test records  select count(\*) into vcount from tblTest where open\_subject\_seq = popen\_subject\_seq;      -- 2) if there are both test records(writting/practical) then update testfile\_check column  if vcount = 2 then  update tblopensubject  set testfile\_check = 'Y'  where open\_subject\_seq = popen\_subject\_seq;  end if;  end procUpdateTestFileCheck\_T;  **----------- 2.3.5) 시험날짜 수정(입력)**  create or replace procedure procUpdateTestDate\_T(  popen\_subject\_seq varchar2,  ptest\_type varchar2,  pupdate\_date date,  presult out number  )  is  begin  update tbltest  set test\_date = pupdate\_date  where open\_subject\_seq = popen\_subject\_seq and type = ptest\_type;    presult := 1;    exception  when others then  presult := 0;    end procUpdateTestDate\_T; | | | |

| 테이블명 | 교사 성적 입출력 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 성적 입출력 | 작성자 | 3조 |
| DML | | | |
| **-- 3.1) 교사별 > 강의종료 과목 목록 출력**  select  os.open\_subject\_seq as 개설과목번호,  course.name as 과정명,  ol.course\_start\_date as 과정시작일,  ol.course\_end\_date as 과정종료일,  room.name as 강의실명,  sub.name as 과목명,  os.subject\_start\_date as 과목시작일,  os.subject\_end\_date as 과목종료일,  os.subject\_progress as 과목강의상태,  book.name as 교재명,  os.attendance\_distribution as 출결배점,  os.written\_distribution as 필기배점,  os.practical\_distribution as 실기배점,  os.score\_check as 성적등록여부  from tblopensubject os  inner join tblsubject sub on os.subject\_seq = sub.subject\_seq  inner join tblopencourse ol on ol.open\_course\_seq = os.open\_course\_seq  inner join tblcourse course on ol.course\_seq = course.course\_seq  inner join tblclassroom room on ol.classroom\_seq = room.classroom\_seq  inner join tbltextbook book on os.textbook\_seq = book.textbook\_seq  where os.teacher\_seq = 'T001'  and os.subject\_progress = '강의종료'  order by ol.course\_start\_date, os.subject\_start\_date;  **-- 3.2) 특정 과목번호 선택 > 교육생 정보 및 성적 출력 (중도탈락생, 중도탈락 날짜 포함)**  select  os.open\_subject\_seq as 수강과목번호,  std.student\_seq as 학생번호,  std.name as 이름,  std.tel as 전화번호,  case  when reg.completion\_status = 'Y'  or (reg.completion\_status is null and reg.dropout\_date is null) then '수료'  when reg.completion\_status = 'N' then '중도탈락'  end as 수료여부,  score.attendance\_score as 출결점수,  round(score.written\_score, 1) as 필기점수,  round(score.practical\_score, 1) as 실기점수,  round(score.subject\_score, 1) as 총점,  reg.dropout\_date as 중도탈락날짜  from tblregister reg  inner join tblopencourse course on reg.open\_course\_seq = course.open\_course\_seq  inner join tblopensubject os on course.open\_course\_seq = os.open\_course\_seq  left outer join tblscoreinfo score on os.open\_subject\_seq = score.opensubject\_seq  and reg.student\_seq = score.student\_seq  and reg.open\_course\_seq = score.open\_course\_seq  inner join tblstudent std on reg.student\_seq = std.student\_seq  where os.open\_subject\_seq = 'OS001'  order by reg.student\_seq;  **----------- 3.3.1) 출결점수 입력(수정)\_프로시저**  create or replace PROCEDURE procUpdateAtScore\_T (  popen\_subject\_seq varchar2,  pstudent\_seq varchar2,  pattendance\_score number,  presult out number  )  is  begin  -- 1) Update attendance score  update tblscoreinfo  set attendance\_score = pattendance\_score  where opensubject\_seq = popen\_subject\_seq  and student\_seq = pstudent\_seq;    -- 2) Update subject\_score tblScoreInfo  procUpdateSubScore\_T(popen\_subject\_seq, pstudent\_seq);  presult := 1;  Exception  when others then  presult := 0;    end procUpdateAtScore\_T;  **----------- 3.3.2) 필기점수 입력(수정)**  -- if there is no record in testresult > insert  -- if a record exists in testresult > update  create or replace PROCEDURE procAddWtScore\_T(  popen\_subject\_seq varchar2, -- 개설과목번호  ptest\_type varchar2, -- 시험타입  pstudent\_seq varchar2, -- 학생번호  pwritten\_score number, -- 점수  presult out number  )  is  vcount number;  vtest\_seq tbltest.test\_seq%type;  vopen\_course\_seq tblregister.open\_course\_seq%type;  begin    -- Found the 'test\_seq'  select test\_seq into vtest\_seq  from tbltest where open\_subject\_seq = popen\_subject\_seq and type = ptest\_type;      -------- 1) Counting testResult record for 'open\_subject\_seq' and 'type' and 'student\_seq'  select count(\*) into vcount  from tbltestresult  where test\_seq = vtest\_seq and student\_seq = pstudent\_seq;      -------- 2) Insert or Update Score  -- 2.1) if there is no record > insert  if vcount = 0 then    -- Found the 'open\_course\_seq'  select open\_course\_seq into vopen\_course\_seq  from tblopensubject where open\_subject\_seq = popen\_subject\_seq;    -- insert score  insert into tbltestresult (student\_seq, test\_seq, score, open\_course\_seq)  values(pstudent\_seq, vtest\_seq, pwritten\_score, vopen\_course\_seq);      -- 2.2) if there is record > update  elsif vcount = 1 then  update tbltestresult  set score = pwritten\_score  where test\_seq = vtest\_seq and student\_seq = pstudent\_seq;    end if;      -------- 3) Update written\_score in tblscoreinfo  procUpdateWtScore\_T(popen\_subject\_seq, pstudent\_seq, vtest\_seq);      -------- 4) Update subject\_score in tblscoreinfo  procUpdateSubScore\_T(popen\_subject\_seq, pstudent\_seq);    presult := 1;    Exception  when others then  presult := 0;    end procAddWtScore\_T;  **----------- 3.3.2.2) 성적정보테이블 > 필기점수 환산 입력(수정)**  create or replace PROCEDURE procUpdateWtScore\_T(  popen\_subject\_seq varchar2, -- 개설과목번호  pstudent\_seq varchar2, -- 학생번호  ptest\_seq varchar2 -- 시험번호  )  is  begin  update tblscoreinfo  set written\_score =  -- 점수 \* 배점  (select score from tbltestresult  where test\_seq = ptest\_seq and student\_seq = pstudent\_seq)  \*  (select written\_distribution / 100 from tblopensubject  where open\_subject\_seq = popen\_subject\_seq)  where opensubject\_seq = popen\_subject\_seq and student\_seq = pstudent\_seq;  end procUpdateWtScore\_T;  **----------- 3.3.3) 실기점수 입력(수정)**  -- if there is no record in testresult > insert  -- if a record exists in testresult > update  create or replace PROCEDURE procAddPtScore\_T(  popen\_subject\_seq varchar2, -- 개설과목번호  ptest\_type varchar2, -- 시험타입  pstudent\_seq varchar2, -- 학생번호  ppractical\_score number, -- 점수  presult out number  )  is  vcount number;  vtest\_seq tbltest.test\_seq%type;  vopen\_course\_seq tblregister.open\_course\_seq%type;  begin    -- Found the 'test\_seq'  select test\_seq into vtest\_seq  from tbltest where open\_subject\_seq = popen\_subject\_seq and type = ptest\_type;      -------- 1) Counting testResult record for 'open\_subject\_seq' and 'type' and 'student\_seq'  select count(\*) into vcount  from tbltestresult  where test\_seq = vtest\_seq and student\_seq = pstudent\_seq;      -------- 2) Insert or Update Score  -- 2.1) if there is no record > insert  if vcount = 0 then    -- Found the 'open\_course\_seq'  select open\_course\_seq into vopen\_course\_seq  from tblopensubject where open\_subject\_seq = popen\_subject\_seq;    -- insert score  insert into tbltestresult (student\_seq, test\_seq, score, open\_course\_seq)  values(pstudent\_seq, vtest\_seq, ppractical\_score, vopen\_course\_seq);      -- 2.2) if there is record > update  elsif vcount = 1 then  update tbltestresult  set score = ppractical\_score  where test\_seq = vtest\_seq and student\_seq = pstudent\_seq;    end if;      -------- 3) Update practical\_score in tblscoreinfo  procUpdatePtScore\_T(popen\_subject\_seq, pstudent\_seq, vtest\_seq);    -------- 4) Update subject\_score in tblscoreinfo  procUpdateSubScore\_T(popen\_subject\_seq, pstudent\_seq);    presult := 1;    Exception  when others then  presult := 0;    end procAddPtScore\_T;  **----------- 3.3.3.2) 성적정보테이블 > 실기점수 환산 입력(수정)**  create or replace PROCEDURE procUpdatePtScore\_T(  popen\_subject\_seq varchar2, -- 개설과목번호  pstudent\_seq varchar2, -- 학생번호  ptest\_seq varchar2 -- 시험번호  )  is  begin  update tblscoreinfo  set practical\_score =  -- 점수 \* 배점  (select score from tbltestresult  where test\_seq = ptest\_seq and student\_seq = pstudent\_seq)  \*  (select practical\_distribution / 100 from tblopensubject  where open\_subject\_seq = popen\_subject\_seq)  where opensubject\_seq = popen\_subject\_seq and student\_seq = pstudent\_seq;  end procUpdatePtScore\_T;  **----------- 3.3.4) 총점 재계산**  create or replace PROCEDURE procUpdateSubScore\_T(  popen\_subject\_seq varchar2,  pstudent\_seq varchar2  )  is  begin  update tblscoreinfo  set subject\_score = attendance\_score + written\_score + practical\_score  where opensubject\_seq = popen\_subject\_seq  and student\_seq = pstudent\_seq;  end procUpdateSubScore\_T; | | | |

| 테이블명 | 교사 출결 관리 및 출결 조회 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 출결 관리 및 출결 조회 | 작성자 | 3조 |
| DML | | | |
| **-- 4.1) 교사별 > 강의 과목 목록 출력(강의 스케줄)**  select  os.open\_subject\_seq as 개설과목번호,  sub.subject\_seq as 과목번호,  course.name as 과정명,  ol.course\_start\_date as 과정시작일,  ol.course\_end\_date as 과정종료일,  room.name as 강의실명,  sub.name as 과목명,  os.subject\_start\_date as 과목시작일,  os.subject\_end\_date as 과목종료일,  os.subject\_progress as 과목강의상태,  book.name as 교재명,  (select count(\*) from tblregister where open\_course\_seq = os.open\_course\_seq and dropout\_date is null) || '명' as "교육생 등록 인원"  from tblopensubject os  inner join tblsubject sub on os.subject\_seq = sub.subject\_seq  inner join tblopencourse ol on ol.open\_course\_seq = os.open\_course\_seq  inner join tblcourse course on ol.course\_seq = course.course\_seq  inner join tblclassroom room on ol.classroom\_seq = room.classroom\_seq  inner join tbltextbook book on os.textbook\_seq = book.textbook\_seq  where os.teacher\_seq = 'T001'  order by ol.course\_start\_date, os.subject\_start\_date;  **-- 4.2.1) 특정 과목번호 선택 > 교육생 출결 출력(과목별)**  select  at.attendance\_date as 날짜,  reg.open\_course\_seq as 과정번호,  os.open\_subject\_seq as 과목번호,  std.name as 학생이름,  attype.attendance\_type as 근태    from tblopensubject os  inner join tblopencourse ol on os.open\_course\_seq = ol.open\_course\_seq  inner join tblregister reg on ol.open\_course\_seq = reg.open\_course\_seq  inner join tblattendance at on reg.student\_seq = at.student\_seq and reg.open\_course\_seq = at.open\_course\_seq  inner join tblattendancetype attype on at.attendance\_type\_seq = attype.attendance\_type\_seq  inner join tblstudent std on reg.student\_seq = std.student\_seq  where  os.open\_subject\_seq = 'OS001'  and at.attendance\_date >= (select subject\_start\_date from tblopensubject where open\_subject\_seq = 'OS001')  and at.attendance\_date <= (select subject\_end\_date from tblopensubject where open\_subject\_seq = 'OS001')    order by at.attendance\_date, at.student\_seq;  **-- 4.2.2) 교육생 출결 출력(연도별)**  select  at.attendance\_date as 날짜,  reg.open\_course\_seq as 과정번호,  os.open\_subject\_seq as 과목번호,  std.name as 학생이름,  attype.attendance\_type as 근태    from tblopensubject os  inner join tblopencourse ol on os.open\_course\_seq = ol.open\_course\_seq  inner join tblregister reg on ol.open\_course\_seq = reg.open\_course\_seq  inner join tblattendance at on reg.student\_seq = at.student\_seq and reg.open\_course\_seq = at.open\_course\_seq  inner join tblattendancetype attype on at.attendance\_type\_seq = attype.attendance\_type\_seq  inner join tblstudent std on reg.student\_seq = std.student\_seq  where  os.open\_subject\_seq = 'OS001'  and at.attendance\_date >= (select subject\_start\_date from tblopensubject where open\_subject\_seq = 'OS001')  and at.attendance\_date <= (select subject\_end\_date from tblopensubject where open\_subject\_seq = 'OS001')  and to\_char(at.attendance\_date,'YYYY') like '2021'    order by at.attendance\_date, at.student\_seq;  **-- 4.2.3) 교육생 출결 출력(월별)**  select  at.attendance\_date as 날짜,  reg.open\_course\_seq as 과정번호,  os.open\_subject\_seq as 과목번호,  std.name as 학생이름,  attype.attendance\_type as 근태    from tblopensubject os  inner join tblopencourse ol on os.open\_course\_seq = ol.open\_course\_seq  inner join tblregister reg on ol.open\_course\_seq = reg.open\_course\_seq  inner join tblattendance at on reg.student\_seq = at.student\_seq and reg.open\_course\_seq = at.open\_course\_seq  inner join tblattendancetype attype on at.attendance\_type\_seq = attype.attendance\_type\_seq  inner join tblstudent std on reg.student\_seq = std.student\_seq  where  os.open\_subject\_seq = 'OS003'  and at.attendance\_date >= (select subject\_start\_date from tblopensubject where open\_subject\_seq = 'OS003')  and at.attendance\_date <= (select subject\_end\_date from tblopensubject where open\_subject\_seq = 'OS003')  and to\_char(at.attendance\_date,'YYYY-MM') like '2021-11'    order by at.attendance\_date, at.student\_seq;  **-- 4.2.4) 교육생 출결 출력(일별)**  select  to\_char(at.attendance\_date, 'yyyy-mm-dd') as 날짜,  reg.open\_course\_seq as 과정번호,  os.open\_subject\_seq as 과목번호,  std.name as 학생이름,  attype.attendance\_type as 근태    from tblopensubject os  inner join tblopencourse ol on os.open\_course\_seq = ol.open\_course\_seq  inner join tblregister reg on ol.open\_course\_seq = reg.open\_course\_seq  inner join tblattendance at on reg.student\_seq = at.student\_seq and reg.open\_course\_seq = at.open\_course\_seq  inner join tblattendancetype attype on at.attendance\_type\_seq = attype.attendance\_type\_seq  inner join tblstudent std on reg.student\_seq = std.student\_seq  where  os.open\_subject\_seq = 'OS001'  and at.attendance\_date >= (select subject\_start\_date from tblopensubject where open\_subject\_seq = 'OS001')  and at.attendance\_date <= (select subject\_end\_date from tblopensubject where open\_subject\_seq = 'OS001')  and at.attendance\_date = '2021-09-08'  order by at.attendance\_date, at.student\_seq; | | | |

**C. 교육생**

| 요구사항명 | 교육생 성적조회 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 성적조회 | 작성자 | 3조 |
| DML | | | |
| --[ 교육생 개인의 정보와 교육생이 수강한 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실이 타이틀로 출력된다. ]  SELECT  st.student\_seq AS "교육생 번호",  st.NAME AS "교육생 이름",  st.tel AS "전화번호",  st.registration\_date AS "등록일자",  C.NAME AS "수강한 과정 이름",  oc.course\_start\_date AS "과정 시작일",  oc.course\_end\_date AS "과정 종료일",  cr.NAME AS "강의실 이름"  FROM tblregister R  INNER JOIN tblstudent st  ON R.student\_seq = st.student\_seq  INNER JOIN tblopencourse oc  ON R.open\_course\_seq = oc.open\_course\_seq  INNER JOIN tblclassroom cr  ON oc.classroom\_seq = cr.classroom\_seq  INNER JOIN tblcourse C  ON oc.course\_seq = C.course\_seq  WHERE st.student\_seq = 'S0038'; --'S0038'에 원하는 교육생 번호 입력  --[ 과목번호, 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 교사명, 과목별 배점정보 (출결, 필기, 실기 배점), 과목별 성적 정보( 출결, 필기, 실기 점수), 과목별 시험날짜가 목록형태로 출력된다. ]  --[ 성적이 등록되지 않은 과목이 있는 경우 과목 정보는 출력되고 점수는 null값으로 출력되도록 한다. ]  SELECT  st.NAME AS "교육생 이름",  os.open\_subject\_seq AS "개설과목번호",  os.subject\_seq AS "과목번호",  S.NAME AS "과목명",  os.subject\_start\_date AS "과목 시작일",  os.subject\_end\_date AS "과목 종료일",  tb.NAME AS "교재명",  T.NAME AS "교사명",  os.attendance\_distribution AS "출결 배점",  os.written\_distribution AS "필기 배점",  os.practical\_distribution AS "실기 배점",  sc.attendance\_score AS "출결 점수",  round(sc.written\_score, 1) AS "필기 점수",  round(sc.practical\_score, 1) AS "실기 점수",  ts.test\_date AS "시험 날짜"  FROM tblopensubject os  INNER JOIN tblteacher T  ON os.teacher\_seq = T.teacher\_seq  INNER JOIN tbltextbook tb  ON os.textbook\_seq = tb.textbook\_seq  INNER JOIN tblsubject S  ON os.subject\_seq = S.subject\_seq  INNER JOIN tbltest ts  ON os.open\_subject\_seq = ts.open\_subject\_seq  INNER JOIN tblscoreinfo sc  ON os.open\_subject\_seq = sc.opensubject\_seq  INNER JOIN tblregister R  ON R.student\_seq = sc.student\_seq AND R.open\_course\_seq = sc.open\_course\_seq  INNER JOIN tblstudent st  ON st.student\_seq = R.student\_seq  WHERE st.student\_seq = 'S0038'; --'S0038'에 원하는 교육생 번호 입력  --[ 과정기간이 끝나지 않은 교육생 또는 중도 탈락 처리된 교육생인 경우 일부 과목만 수강한다. ]  -- \*\*\* **과정 기간이 끝나지 않은 또는 중도 탈락 처리된 교육생의 목록 뷰** \*\*\*  CREATE OR REPLACE VIEW vwfailed\_st  AS  SELECT \* FROM tblregister  WHERE completion\_status = 'N' OR dropout\_date IS NOT NULL;  SELECT  st.student\_seq AS "학생 번호",  st.NAME AS "학생 이름",  R.completion\_status AS "수료여부",  R.completion\_date AS "수료날짜",  R.dropout\_date AS "중도탈락날짜",  S.NAME AS "과목 이름",  os.subject\_start\_date AS "과목 시작일",  os.subject\_end\_date AS "과목 종료일",  A.attendance\_date AS "출결 날짜",  AT.attendance\_type AS "근태 유형",  oc.open\_course\_seq AS "개설과정 번호",  C.NAME AS "과정 이름"  FROM tblregister R  INNER JOIN tblattendance A  ON R.student\_seq = A.student\_seq AND R.open\_course\_seq = A.open\_course\_seq  INNER JOIN tblattendancetype AT  ON A.attendance\_type\_seq = AT.attendance\_type\_seq  INNER JOIN tblstudent st  ON st.student\_seq = R.student\_seq  INNER JOIN tblopencourse oc  ON oc.open\_course\_seq = R.open\_course\_seq  INNER JOIN tblcourse C  ON oc.course\_seq = C.course\_seq  INNER JOIN tblopensubject os  ON oc.open\_course\_seq = os.open\_course\_seq  INNER JOIN tblsubject S  ON os.subject\_seq = S.subject\_seq  WHERE (R.completion\_status = 'N' OR R.dropout\_date IS NOT NULL)  AND R.student\_seq = (SELECT student\_seq FROM **vwfailed\_st** WHERE open\_course\_seq = 'OL001') --’OL001’에 원하는 개설 과정 번호 입력  AND R.dropout\_date BETWEEN os.subject\_start\_date AND os.subject\_end\_date  ORDER BY os.subject\_start\_date, A.attendance\_date; | | | |

| 요구사항명 | 교육생 출결관리 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 출결관리 | 작성자 | 3조 |
| DML | | | |
| --[ 매일의 근태를(출근 1회, 퇴근1회)를 기록할 수 있다. ]  -- \*\*\* **학생번호를 넣으면 현재 이 학생이 듣고 있는 과정번호를 반환해주는 함수** \*\*\*  CREATE OR REPLACE FUNCTION fngetcourseseq\_st(  pseq VARCHAR2  ) RETURN VARCHAR2  IS  vseq tblregister.open\_course\_seq%TYPE;  BEGIN    SELECT R.open\_course\_seq INTO vseq  FROM tblregister R INNER JOIN tblstudent S ON R.student\_seq = S.student\_seq  INNER JOIN tblopencourse oc ON R.open\_course\_seq = oc.open\_course\_seq  WHERE R.student\_seq = pseq  AND sysdate BETWEEN oc.course\_start\_date AND oc.course\_end\_date;    RETURN vseq;  END fngetcourseseq\_st;  -- \*\*\* **오늘자 출근 시간 입력 + 지각 근태유형 입력 저장 프로시저** \*\*\*  CREATE OR REPLACE PROCEDURE procadddayattendancestart\_st(  ptime VARCHAR2,  pstudent\_seq VARCHAR2,  presult OUT NUMBER --성공(1) or 실패(0)  )  IS  vcntda NUMBER;  BEGIN    SELECT COUNT(\*) INTO vcntda FROM tbldayattendance --일일출결 테이블  WHERE to\_char(day\_attendance\_date, 'yyyy-mm-dd') = to\_char(current\_date, 'yyyy-mm-dd')  AND student\_seq = pstudent\_seq  AND open\_course\_seq = fngetcourseseq\_st(pstudent\_seq)  AND is\_attendance = '출근'; --오늘자 출결 '출근' 행이 있는지 확인하기 위함 (중복 레코드 방지)    IF vcntda = 0 AND TO\_DATE(ptime, 'hh24:mi:ss') < TO\_DATE('09:05:01', 'hh24:mi:ss') THEN  INSERT INTO tbldayattendance (day\_attendance\_date, TIME, student\_seq, open\_course\_seq, is\_attendance) VALUES (  current\_date,  TO\_DATE(ptime, 'hh24:mi:ss'),  pstudent\_seq,  fngetcourseseq\_st(pstudent\_seq),  '출근'  );  presult := 1;    ELSIF vcntda = 0  AND TO\_DATE(ptime, 'hh24:mi:ss') BETWEEN TO\_DATE('09:05:01', 'hh24:mi:ss') AND TO\_DATE('12:50:59', 'hh24:mi:ss') THEN  --점심시간 전까지 오면 지각 처리    INSERT INTO tbldayattendance (day\_attendance\_date, TIME, student\_seq, open\_course\_seq, is\_attendance) VALUES (  current\_date,  TO\_DATE(ptime, 'hh24:mi:ss'),  pstudent\_seq,  fngetcourseseq\_st(pstudent\_seq),  '출근'  );  INSERT INTO tblattendance (attendance\_date, student\_seq, open\_course\_seq, attendance\_type\_seq) VALUES (  current\_date,  pstudent\_seq,  fngetcourseseq\_st(pstudent\_seq),  'TA02' --지각  );  presult := 2;  END IF;  EXCEPTION  WHEN OTHERS THEN  presult := 0;  END procadddayattendancestart\_st;  -- \*\*\* **오늘자 퇴근 시간 입력 + 정상 근태유형 입력 저장 프로시저** \*\*\*  CREATE OR REPLACE PROCEDURE procadddayattendanceend\_st(  ptime VARCHAR2,  pstudent\_seq VARCHAR2,  presult OUT NUMBER --성공(1) or 실패(0)  )  IS  vcntda NUMBER;  vcnta NUMBER;  BEGIN  SELECT COUNT(\*) INTO vcntda FROM tbldayattendance --일일출결 테이블  WHERE to\_char(day\_attendance\_date, 'yyyy-mm-dd') = to\_char(current\_date, 'yyyy-mm-dd')  AND student\_seq = pstudent\_seq  AND open\_course\_seq = fngetcourseseq\_st(pstudent\_seq)  AND is\_attendance = '퇴근'; --오늘자 출결 '퇴근'행이 있는지 확인하기 위함 (중복 레코드 방지)    SELECT COUNT(\*) INTO vcnta FROM tblattendance --근태 테이블  WHERE to\_char(attendance\_date, 'yyyy-mm-dd') = to\_char(current\_date, 'yyyy-mm-dd')  AND student\_seq = pstudent\_seq  AND open\_course\_seq = fngetcourseseq\_st(pstudent\_seq); --근태 유형이 지각(TA02)으로 찍힌 행이 있는지 확인하기 위함    IF vcntda = 0 AND vcnta <> 0 AND TO\_DATE(ptime, 'hh24:mi:ss') > TO\_DATE('09:50:59', 'hh24:mi:ss') THEN  --1교시 끝나고부터 외출/조퇴/병가 퇴근 가능  INSERT INTO tbldayattendance (day\_attendance\_date, TIME, student\_seq, open\_course\_seq, is\_attendance) VALUES (  current\_date,  TO\_DATE(ptime, 'hh24:mi:ss'),  pstudent\_seq,  fngetcourseseq\_st(pstudent\_seq),  '퇴근'  );  presult := 1;    ELSIF vcntda = 0 AND vcnta = 0 AND TO\_DATE(ptime, 'hh24:mi:ss') > TO\_DATE('17:45:00', 'hh24:mi:ss') THEN  --수업 끝나기 5분 전부터 정상 처리 허용  INSERT INTO tbldayattendance (day\_attendance\_date, TIME, student\_seq, open\_course\_seq, is\_attendance) VALUES (  current\_date,  TO\_DATE(ptime, 'hh24:mi:ss'),  pstudent\_seq,  fngetcourseseq\_st(pstudent\_seq),  '퇴근'  );  INSERT INTO tblattendance (attendance\_date, student\_seq, open\_course\_seq, attendance\_type\_seq) VALUES (  current\_date,  pstudent\_seq,  fngetcourseseq\_st(pstudent\_seq),  'TA01' --정상  );  presult := 2;    END IF;  EXCEPTION  WHEN OTHERS THEN  presult := 0;  END procadddayattendanceend\_st;  -- \*\*\* **오늘자 나머지 근태유형(조퇴, 외출, 병가, 기타) 입력/수정 저장 프로시저** \*\*\*  CREATE OR REPLACE PROCEDURE procaddattendance\_st (  pstudent\_seq VARCHAR2,  ptype VARCHAR2,  presult OUT NUMBER --성공(1) or 실패(0)  )  IS  vcnta NUMBER;  BEGIN  SELECT COUNT(\*) INTO vcnta FROM tblattendance --근태 테이블  WHERE to\_char(attendance\_date, 'yyyy-mm-dd') = to\_char(current\_date, 'yyyy-mm-dd')  AND student\_seq = pstudent\_seq  AND open\_course\_seq = fngetcourseseq\_st(pstudent\_seq); --근태 유형 행이 있는지 확인하기 위함 (중복 레코드 방지)    IF vcnta = 0 THEN  INSERT INTO tblattendance (attendance\_date, student\_seq, open\_course\_seq, attendance\_type\_seq) VALUES (  current\_date,  pstudent\_seq,  fngetcourseseq\_st(pstudent\_seq),  ptype  );  presult := 1;    ELSE  UPDATE tblattendance SET  attendance\_type\_seq = ptype  WHERE student\_seq = pstudent\_seq  AND open\_course\_seq = fngetcourseseq\_st(pstudent\_seq)  AND to\_char(attendance\_date, 'yyyy-mm-dd') = to\_char(current\_date, 'yyyy-mm-dd');    presult := 2;    END IF;  EXCEPTION  WHEN OTHERS THEN  presult := 0;  END procaddattendance\_st;  -- \*\*\* **원하는날짜 나머지 근태유형(조퇴, 외출, 병가, 기타) 수정 저장 프로시저** \*\*\*  CREATE OR REPLACE PROCEDURE procwantupdateattendance\_st (  pdate VARCHAR2,  pstudent\_seq VARCHAR2,  ptype VARCHAR2,  presult OUT NUMBER --성공(1) or 실패(0)  )  IS  vcnta NUMBER;  BEGIN  SELECT COUNT(\*) INTO vcnta FROM tblattendance  WHERE to\_char(attendance\_date, 'yyyy-mm-dd') = TO\_DATE(pdate, 'yyyy-mm-dd')  AND student\_seq = pstudent\_seq  AND open\_course\_seq = fngetcourseseq\_st(pstudent\_seq); --바꾸고 싶은 날짜의 근태 행이 존재하는지 확인하기 위함    IF vcnta = 1 THEN  UPDATE tblattendance SET  attendance\_type\_seq = ptype  WHERE student\_seq = pstudent\_seq  AND open\_course\_seq = fngetcourseseq\_st(pstudent\_seq)  AND to\_char(attendance\_date, 'yyyy-mm-dd') = TO\_DATE(pdate, 'yyyy-mm-dd');    presult := 1;    END IF;  EXCEPTION  WHEN OTHERS THEN  presult := 0;  END procwantupdateattendance\_st;  -- \*\*\* **(날짜, 학생번호, 근태유형번호) -> 원하는 날짜 근태유형 수정 실행 저장 프로시저** \*\*\*  CREATE OR REPLACE PROCEDURE procwantupdateattendanceact\_st(  pdate VARCHAR2,  pstudent\_seq VARCHAR2,  ptype VARCHAR2  )  IS  vresult NUMBER;  BEGIN  procwantupdateattendance\_st(pdate, pstudent\_seq, ptype, vresult);    IF vresult = 1 THEN  dbms\_output.put\_line(pdate || '일자 근태유형 ' || ptype || '으로 수정 완료.');  ELSE  dbms\_output.put\_line('다시 입력해주세요.');  END IF;  END procwantupdateattendanceact\_st; | | | |

| 요구사항명 | 교육생 출결 조회 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 출결조회 | 작성자 | 3조 |
| DML | | | |
| --[ 모든 출결 조회는 근태 상황을 구분할 수 있다.(정상, 지각, 조퇴, 외출, 병가, 기타) ]  --[ 다른 교육생의 현황은 조회할 수 없다. ]  --[ 본인의 출결 현황을 기간별(전체, 월, 일)로 조회할 수 있다. ]  SELECT  DISTINCT  S.student\_seq AS "교육생 번호",  S.NAME AS "교육생 이름",  A.attendance\_date AS "날짜",  att.attendance\_type AS "근태 상황",  to\_char(da.TIME, 'hh24:mi:ss') AS "시간",  da.is\_attendance AS "출/퇴근"  FROM tblattendance A  INNER JOIN tblattendancetype att  ON A.attendance\_type\_seq = att.attendance\_type\_seq  INNER JOIN tbldayattendance da  ON A.student\_seq = da.student\_seq  AND A.open\_course\_seq = da.open\_course\_seq  AND A.attendance\_date = da.day\_attendance\_date  INNER JOIN tblstudent S  ON S.student\_seq = A.student\_seq  WHERE S.student\_seq = ‘S0038’  ORDER BY A.attendance\_date DESC;  --전체 기간 조회  SELECT  DISTINCT  S.student\_seq AS "교육생 번호",  S.NAME AS "교육생 이름",  A.attendance\_date AS "날짜",  att.attendance\_type AS "근태 상황",  to\_char(da.TIME, 'hh24:mi:ss') AS "시간",  da.is\_attendance AS "출/퇴근"  FROM tblattendance A  INNER JOIN tblattendancetype att  ON A.attendance\_type\_seq = att.attendance\_type\_seq  INNER JOIN tbldayattendance da  ON A.student\_seq = da.student\_seq  AND A.open\_course\_seq = da.open\_course\_seq  AND A.attendance\_date = da.day\_attendance\_date  INNER JOIN tblstudent S  ON S.student\_seq = A.student\_seq  WHERE S.student\_seq = ‘S0038’  AND A.attendance\_date BETWEEN TO\_DATE('2021-09-01', 'YYYY-MM-DD') AND TO\_DATE('2021-09-30', 'YYYY-MM-DD')  ORDER BY A.attendance\_date DESC;  --원하는 기간별로 조회가 가능하다  AND to\_char(A.attendance\_date, 'mm') = '09'  --WHERE문의 두번째 조건을 이렇게 바꾸면 원하는 '월'의 출결 조회가 가능하다  SELECT  DISTINCT  S.student\_seq AS "교육생 번호",  S.NAME AS "교육생 이름",  A.attendance\_date AS "날짜",  att.attendance\_type AS "근태 상황",  to\_char(da.TIME, 'hh24:mi:ss') AS "시간",  da.is\_attendance AS "출/퇴근"  FROM tblattendance A  INNER JOIN tblattendancetype att  ON A.attendance\_type\_seq = att.attendance\_type\_seq  INNER JOIN tbldayattendance da  ON A.student\_seq = da.student\_seq  AND A.open\_course\_seq = da.open\_course\_seq  AND A.attendance\_date = da.day\_attendance\_date  INNER JOIN tblstudent S  ON S.student\_seq = A.student\_seq  WHERE S.student\_seq = fnlogin\_st('김조하', '1770114')  AND to\_char(A.attendance\_date, 'YYYY-MM-DD') = '2021-10-05'  -- and a.attendance\_date = to\_date('2021-10-05', 'YYYY-MM-DD')  ORDER BY A.attendance\_date;  --원하는 날짜의 출퇴근 현황을 조회할 수 있다  AND to\_char(day\_attendance\_date, 'YYYY-MM-DD') = to\_char(current\_date, 'YYYY-MM-DD')  --WHERE문의 두번째 조건을 이렇게 바꾸면 오늘자의 출퇴근 현황을 조회할 수 있다 | | | |

**D. 추가**

| 요구사항명 | 국가 지원 제도 이용 여부 확인 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 관리자 - 교육생 정보 등록 | 작성자 | 3조 |
| DML | | | |
| **--지원제도 입력**  INSERT INTO TBLSUPPORTSYSTEM (STUDENT\_SEQ, OPEN\_COURSE\_SEQ, TRAINING\_INSENTIVE, EMPLOYEE\_SUPPORT)  VALUES ('S0496', 'OL015', 'Y', 'Y');  **--지원여부수정**  UPDATE TBLSUPPORTSYSTEM SET EMPLOYEE\_SUPPORT = 'N'  WHERE STUDENT\_SEQ = 'S0496' AND open\_course\_seq = 'OL015';  **--내일배움 훈련장려금 수령교육생 조회**  select DISTINCT  STUDENT\_SEQ as "내일배움 지원교육생번호"  from TBLSUPPORTSYSTEM  where TRAINING\_INSENTIVE = 'Y';  **--국민취업 지원제도 국민취업제도 수령 교육생조회**  select DISTINCT  STUDENT\_SEQ as "국민취업지원제도 지원교육생번호"  from TBLSUPPORTSYSTEM  where EMPLOYEE\_SUPPORT = 'Y'; | | | |

| 요구사항명 | 전체 기수 수료 및 연계기업 취업 통계를 출력한다. | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 관리자 - 통계 | 작성자 | 3조 |
| DML | | | |
| **-- 취업연계된 구직자**  select  TBLJOBSEEKER.student\_seq as "교육생번호",  TBLJOBSEEKER.OPEN\_COURSE\_SEQ as "개강과정번호",  TBLJOBSEEKER.employment\_date as "취업날짜",  tblCompany.name as "취업기업명"  from TBLJOBSEEKER join tblJobPosting  on(TBLJOBSEEKER.recruiter\_seq = tblJobPosting.recruiter\_seq)  join tblCompany  on(tblJobPosting.company\_seq = tblCompany.company\_seq)  WHERE TBLJOBSEEKER.employment\_state = 'Y';  **-- 취업자수**  select count(\*) as 취업자수  from TBLJOBSEEKER  WHERE employment\_state = 'Y';  **--취업자 평균급여**  select STUDENT\_SEQ as "취업자번호",  OPEN\_COURSE\_SEQ as "과정번호",  avg(SALARY) as "평균급여"  from TBLJOBSEEKER join TBLJOBPOSTING  on(TBLJOBSEEKER.RECRUITER\_SEQ = TBLJOBPOSTING.RECRUITER\_SEQ)  WHERE employment\_state = 'Y'  GROUP by rollup(OPEN\_COURSE\_SEQ, STUDENT\_SEQ);  **--과정별 취업기업**  select tblJobSeeker.OPEN\_COURSE\_SEQ as "과정번호",  REGEXP\_REPLACE(LISTAGG(tblCompany.name, ',') within group (order by tblCompany.name), '([^,]+)(,\1)\*(,|$)', '\1\3') as "취업기업"  from TBLJOBSEEKER join TBLJOBPOSTING  on(TBLJOBSEEKER.RECRUITER\_SEQ = TBLJOBPOSTING.RECRUITER\_SEQ)  join tblCompany on(TBLJOBPOSTING.company\_seq = tblCompany.company\_seq)  WHERE employment\_state = 'Y'  group by tblJobSeeker.OPEN\_COURSE\_SEQ;  **--취업률**  CREATE OR REPLACE PROCEDURE procEmploymentRate\_A  IS  v\_jobseeker number;  v\_getjob number;  BEGIN  select count(\*)into v\_getjob  from TBLJOBSEEKER  WHERE employment\_state = 'Y';  select count(\*) into v\_jobseeker  from TBLJOBSEEKER;    dbms\_output.put\_line('구직자 취업률 :' || round(v\_getjob/v\_jobseeker\*100,1) || '%');  END;  **--수료율**  CREATE OR REPLACE PROCEDURE procCompletionRate\_A  IS  v\_Num number;  v\_name varchar2(300);  type varray\_type is varray(99) of varchar(20);  type varray\_type\_n is varray(99) of number;  v\_Course varray\_type := varray\_type();  v\_Completion varray\_type\_n := varray\_type\_n();  v\_Register varray\_type\_n := varray\_type\_n();  CURSOR OpenCourse IS  SELECT distinct open\_course\_seq FROM tblOpenCourse;  BEGIN  v\_Num := 1;  for getOpenCourse in OpenCourse  loop  v\_Course.EXTEND;  v\_Course(v\_Num) := getOpenCourse.open\_course\_seq;  v\_Num := v\_Num+1;  end loop;  for i in 1..v\_Course.count  loop  v\_Register.EXTEND;  select count(\*) into v\_Register(i) from tblRegister  where open\_course\_seq = v\_Course(i);  v\_Completion.EXTEND;  select count(\*) into v\_Completion(i) from tblRegister  where open\_course\_seq = v\_Course(i)  and tblRegister.completion\_status = 'Y';    select tblCourse.name into v\_name  from tblOpenCourse join tblCourse  on(tblOpenCourse.course\_seq = tblCourse.course\_seq)  where tblOpenCourse.open\_course\_seq = v\_Course(i);    if v\_Register(i) = 0  then dbms\_output.put\_line(v\_name);  dbms\_output.put\_line('수료율 : '|| 0);  else  dbms\_output.put\_line(v\_name);  dbms\_output.put\_line('수료율 : '|| round(v\_Completion(i)/v\_Register(i),3) \* 100 || '%');  end if;  end loop;    END; | | | |

| 요구사항명 | 관리자 - 교육생 기념품 정보 관리 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 기념품 정보 확인 | 작성자 | 3조 |
| DML | | | |
| **--기념품정보**  INSERT INTO TBLGIFT (GIFT\_SEQ, NAME, PRICE)  VALUES ('GIFT11', '신세계 백화점 상품권', 10000); --기념품 추가  UPDATE TBLGIFT SET NAME = '롯데 백화점 상품권'  WHERE GIFT\_SEQ = 'GIFT11'; -- 기념품수정  DELETE FROM TBLGIFT  WHERE GIFT\_SEQ = 'GIFT11'; --기념품삭제  **--교육 과정 별지급한 기념품을 확인**  select OPEN\_COURSE\_SEQ as "과정번호",  REGEXP\_REPLACE(LISTAGG(tblgift.name, ',') within group (order by tblgift.name), '([^,]+)(,\1)\*(,|$)', '\1\3') as "과정별 증정 기념품"  from tblGetgift join tblgift  on(tblGetgift.gift\_seq = tblgift.gift\_seq)  group by tblgetgift.open\_course\_seq;  **--수강별 지급받은 기념품 확인**  select OPEN\_COURSE\_SEQ as "과정번호",  STUDENT\_SEQ as "교육생번호",  REGEXP\_REPLACE(LISTAGG(tblgift.name, ',') within group (order by tblgift.name), '([^,]+)(,\1)\*(,|$)', '\1\3') as "교육생별 증정 기념품"  from tblGetgift join tblgift  on(tblGetgift.gift\_seq = tblgift.gift\_seq)  group by tblgetgift.OPEN\_COURSE\_SEQ, tblgetgift.STUDENT\_SEQ;  **--기념품증정**  CREATE OR REPLACE PROCEDURE procGetGift\_A (  v\_GET\_GIFT\_SEQ VARCHAR2,  v\_GIFT\_SEQ VARCHAR2,  v\_STUDENT\_SEQ VARCHAR2,  v\_OPEN\_COURSE\_SEQ VARCHAR2  )  IS  BEGIN  INSERT INTO TBLGETGIFT (GET\_GIFT\_SEQ, GET\_DATE, GIFT\_SEQ, STUDENT\_SEQ, OPEN\_COURSE\_SEQ)  VALUES (v\_GET\_GIFT\_SEQ, to\_date(sysdate, 'YYYY-MM-DD'), v\_GIFT\_SEQ, v\_STUDENT\_SEQ, v\_OPEN\_COURSE\_SEQ);  dbms\_output.put\_line('기념품을 증정했습니다.');  END; | | | |

| 요구사항명 | 학생 성적 통계 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 관리자 - 시험 관리 및 성적 조회 | 작성자 | 3조 |
| DML | | | |
| **--각과목별 평균점수**  select opensubject\_seq as "과목번호",  round(avg(SUBJECT\_SCORE),1) as "과목평균점수"  from tblScoreInfo  where SUBJECT\_SCORE is not null  group by opensubject\_seq  order by avg(SUBJECT\_SCORE) desc;  **--수료생의 과정 최종평점**  select tblScoreInfo.student\_seq as "교육생번호",  tblScoreInfo.open\_course\_seq as "과정번호",  round(avg(SUBJECT\_SCORE),0) as "과정평점"  from tblScoreInfo join tblRegister  on(tblScoreInfo.student\_seq = tblRegister.student\_seq and tblScoreInfo.open\_course\_seq = tblRegister.open\_course\_seq)  where tblRegister.completion\_status = 'Y'  group by tblScoreInfo.student\_seq, tblScoreInfo.open\_course\_seq;  **--각 과정 교육생들의 평균 점수, 최고 점수, 최저 점수를 조회**  select open\_course\_seq as "과정번호",  round(avg(SUBJECT\_SCORE),0) as "과정평균점수",  max(SUBJECT\_SCORE) as "최고점수",  min(SUBJECT\_SCORE) as "최저점수"  from tblScoreInfo  group by open\_course\_seq;  **--평가점수입력**  CREATE OR REPLACE PROCEDURE procUpdateEvalScore\_A  IS  v\_Num number;  v\_score number;  type varray\_type is varray(999) of varchar(20);  type varray\_type\_n is varray(99) of number;  v\_Course varray\_type := varray\_type();  v\_Student varray\_type := varray\_type();  CURSOR CourseStudent IS  select open\_course\_seq, student\_seq  from tblRegister;  BEGIN  v\_Num := 1;  for getOpenCourse in CourseStudent  loop  v\_Course.EXTEND;  v\_Student.EXTEND;  v\_Course(v\_num) := getOpenCourse.open\_course\_seq;  v\_Student(v\_num):= getOpenCourse.student\_seq;  v\_Num := v\_Num+1;  end loop;  select count(\*) into v\_num from tblRegister;  for i in 1..v\_num  loop  v\_score := 0;  select round(avg(subject\_score),1) into v\_score  from tblScoreInfo  where student\_seq = v\_Student(i) and open\_course\_seq = v\_Course(i);  if not(v\_score = 0) then  update tblRegister  set evalution\_score = v\_score  where student\_seq = v\_Student(i) and open\_course\_seq = v\_Course(i);  end if;  end loop;    END; | | | |

| 테이블명 | 교육생과 기업간 취업 연계 | 작성일 | 2021-12-07 |
| --- | --- | --- | --- |
| System | 관리자 - 취업연계 | 작성자 | 3조 |
| DML | | | |
| **-- 교육생의 취업지원**  INSERT INTO TBLJOBSEEKER (STUDENT\_SEQ, OPEN\_COURSE\_SEQ, DESIRED\_PLACE, MIN\_SALARY, MAX\_SALARY, ACADEMIC, EMPLOYMENT\_STATE, EMPLOYMENT\_DATE, RECRUITER\_SEQ, TASK\_SEQ)  VALUES ((SELECT tblregister.student\_seq FROM tblRegister WHERE STUDENT\_SEQ = 'S0381' AND open\_course\_seq = 'OL001'),  (SELECT tblregister.open\_course\_seq FROM tblRegister WHERE STUDENT\_SEQ = 'S0381' AND open\_course\_seq = 'OL001'),  '부산광역시', 2000, 3000, '고졸', 'N', NULL,  (select tbljobposting.RECRUITER\_SEQ from TBLJOBPOSTING WHERE tbljobposting.salary >= 2000 AND tbljobposting.EDUCATION\_LEVEL LIKE '고졸' AND work\_place LIKE '%부산광역시%' AND rownum = 1),  (select tbltask.task\_seq from tblTask where tbltask.name LIKE '웹개발'));  **–자격증정보**  INSERT INTO TBLLICENSE (LICENSE\_SEQ, NAME)  VALUES ('SP111', 'ITQ'); --자격증 등록  UPDATE TBLLICENSE SET NAME = 'ITQA'  WHERE LICENSE\_SEQ = 'SP111'; --자격증 정보수정  DELETE FROM TBLLICENSE  WHERE LICENSE\_SEQ = 'SP111'; --자격증삭제  **--교육생자격증입력**  INSERT INTO TBLSTUDENTLICENSE (STUDENT\_SEQ, LICENSE\_SEQ, REGISTRATION\_DATE)  VALUES ((SELECT tblregister.student\_seq FROM tblRegister WHERE STUDENT\_SEQ = 'S0381' AND open\_course\_seq = 'OL001'),  (SELECT LICENSE\_SEQ from tblLicense where name = '네트워크관리사'), to\_date('12/01/2019', 'MM/DD/YYYY'));  **--기업등록**  INSERT INTO TBLCOMPANY (COMPANY\_SEQ, NAME, BUSINESS\_NUMBER, POSTCODE, ADDRESS, TEL)  VALUES ('COM0051', '네오위즈', '229-12-76696', 32624, '경기도 남양주시 오장로1길 6', '031-428-3246');  **--기업의 구인의뢰 등록**  INSERT INTO TBLJOBPOSTING (RECRUITER\_SEQ, COMPANY\_SEQ, WORK\_PLACE, SALARY, EDUCATION\_LEVEL, EMPLOYEE\_NUMBER, START\_DATE, END\_DATE, STATE, TASK\_SEQ) VALUES ('RE0061',  (select tblcompany.company\_seq from tblCompany where tblcompany.name LIKE '네오위즈'),  (select regexp\_substr((select tblcompany.address from tblcompany where tblcompany.name like '네오위즈'),'[^ ]+ \w+',1) from dual),  2600, '무관', 8, to\_date('2022-03-01', 'YYYY-MM-DD'), to\_date('2022-11-26', 'YYYY-MM-DD'), '채용예정',  (select tbltask.task\_seq from tblTask where tbltask.name LIKE '웹개발'));    **--기업의 채용담당자 수정**  INSERT INTO TBLRECRUITER (RECRUITER\_SEQ, NAME, POSITION, TEL, EMAIL)  VALUES ('RE0061', '김현규', '대리', '010-3389-2084', 'srhkras@gmail.com');  UPDATE TBLRECRUITER SET POSITION = '과장'  WHERE RECRUITER\_SEQ = 'RE0061';  **--수료생- 구직공고 찾기**  select  TBLJOBPOSTING.recruiter\_seq as "모집번호",  tblcompany.name as "기업명",  TBLJOBPOSTING.work\_place as "근무지",  TBLJOBPOSTING.salary as "급여",  TBLJOBPOSTING.education\_level as "필요학력",  TBLJOBPOSTING.employee\_number as "채용인원",  tbltask.name as "직무명"  from tbltask join TBLJOBPOSTING  on(tbltask.TASK\_SEQ = TBLJOBPOSTING.TASK\_SEQ)  join tblcompany  on(TBLJOBPOSTING.company\_seq = tblcompany.company\_seq)  where WORK\_PLACE like concat(concat('%',(select desired\_place from TBLJOBSEEKER where student\_seq = 'S0250' and open\_course\_seq = 'OL007')),'%')  and SALARY >= (select min\_salary from TBLJOBSEEKER where student\_seq = 'S0250' and open\_course\_seq = 'OL007')  and TO\_CHAR(SYSDATE, 'YYYY-MM-DD') >= TO\_CHAR(START\_DATE, 'YYYY-MM-DD')  and TO\_CHAR(SYSDATE, 'YYYY-MM-DD') <= TO\_CHAR(END\_DATE, 'YYYY-MM-DD')  and TBLTASK.name = '웹개발';  **--기업 - 요건에 맞는 수료생찾기**  select TBLJOBSEEKER.student\_seq as "교육생번호",  TBLJOBSEEKER.open\_course\_seq as "개강과정번호",  TBLJOBSEEKER.desired\_place as "희망근무지",  TBLJOBSEEKER.min\_salary as "희망최저연봉",  tbltask.name as "희망직무명"  from TBLJOBSEEKER join tbltask  on(TBLJOBSEEKER.task\_seq = tbltask.task\_seq)  where TBLJOBSEEKER.desired\_place like concat(concat('%',REGEXP\_SUBSTR((select work\_place from TBLJOBPOSTING where recruiter\_seq= 'RE0059'),'\w+',1,1)),'%')  and min\_salary <= (select salary from TBLJOBPOSTING where recruiter\_seq= 'RE0059')  and TBLJOBSEEKER.TASK\_SEQ = (select TASK\_SEQ from TBLJOBPOSTING where recruiter\_seq= 'RE0059'); | | | |