**업무 쿼리문(ANSI-SQL)**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | A-02(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 기초 정보 관리 요구사항 | 작성자 | 4조(조인제, 정다음) |
| DML | | | |
| --교사  --1. 교사 정보 등록(교사번호, 교사 이름, 민증 뒷자리, 연락처)  insert into mTeacher values (11, '조성제', '1822111', '010-6822-3727');  --2. 교사 정보 수정  update mTeacher set name = '조인제' where Teacher\_id = 11;  --3. 교사 정보 삭제  delete from mTeacher where Teacher\_id = 11;  inner join mAvailableSubject a  on t.teacher\_id = a.teacher\_id  inner join mSubject s  on a.subject\_id = s.subject\_id;  --강의실  --1. 강의실 정보 등록(강의실 번호, 인원수)  INSERT INTO mRoom VALUES (7, 30);  --2. 강의실 정보 수정  update mRoom set capacity = 26 where room\_id = 7;  --3. 강의실 정보 삭제  delete from mRoom where room\_id = 7;  --4. 강의실 정보 조회(강의실 번호, 인원수)을 출력한다.  SELECT \* FROM mRoom; | | | |
| -- 교재  --1. 교재 정보 등록(교재 번호, 제목, 출판사)  INSERT INTO mBook VALUES (51, '조인제의 정석', '도우미출판');  --2. 교재 정보 수정  update mBook set title = '나의 정석' where book\_id = 51;  --3. 교재 정보 삭제  delete from mBook where book\_id = 51;  --4. 교재 정보 조회(교재 번호, 제목, 출판사)을 출력한다.  SELECT \* FROM mBook;  select count(\*) from mCurriculum;  select \* from mCurriculum;  -- 과정 생성  insert into mCurriculum values ((select count(\*) from mCurriculum) +1, 'Chat GPT를 활용한 웹 개발 과정');  -- 과정명 수정  update mCurriculum SET title = 'Chat GPT를 활용한 앱 개발 과정' where curriculum\_id = 51;  -- 과정 삭제  delete from mCurriculum where curriculum\_id = 51; | | | |
| select count(\*) from mSubject;  select \* from mSubject;  -- 과목 생성  insert into msubject values ((select count(\*) from msubject) + 1, 'LARAVEL');  -- 과목명 수정  update msubject set title = 'golang' where subject\_id = 51;  -- 과목 삭제  delete from msubject where subject\_id = 51;  select \* from mHoliday;  --공휴일 생성  insert into mHoliday values(to\_date('2027-06-06', 'yyyy-mm-dd'), '공휴일');  --공휴일 수정  update mHoliday set totaldate = '2027-10-4', dateClassfication = '평일' where totaldate = '2027-06-06';  --공휴일 삭제  delete from mHoliday where totaldate = '2027-10-4'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | A-03(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 모든 교사 정보 등록 및 관리 | 작성자 | 4조(김예진) |
| DML | | | |
| /\*교사 정보 출력 시>교사의 기본 정보 전부 출력(이름, 주민번호 뒷자리, 전화번호, 강의 가능 과목) 출력\*/  CREATE OR REPLACE VIEW teacher\_subject\_info AS  SELECT  t.name AS 교사명,  t.jumin AS 주민번호,  t.tel AS 전화번호,  s.title AS 강의\_가능\_과목  FROM  mTeacher  JOIN  mAvailableSubject asub ON t.teacher\_id = asub.teacher\_id  JOIN  mSubject s ON asub.subject\_id = s.subject\_id;  SELECT \* FROM teacher\_subject\_info; | | | |
| /\* 특정 교사 선택 시>배정된 개설 과목명, 개설 과목기간(시작 년원일, 종료 년월일), 과정 명, 개설 과정기간(시작 년월일, 종료 년월일), 교재 명, 강의실, 강의 진행 여부(강의 예정, 강의 중, 강의 종료) 출력 \*/  CREATE OR REPLACE VIEW teacher\_class\_schedule AS  SELECT  os.title AS 개설과목명,  os.startdate AS 개설과목시작일,  os.enddate AS 개설과목종료일,  c.title AS 과정명,  oc.startdate AS 개설과정시작일,  oc.enddate AS 개설과정종료일,  b.title AS 교재명,  r.room\_id AS 강의실번호, -- 강의실 번호  t.name AS 교사명,  CASE  WHEN SYSDATE < os.startdate THEN '강의예정'  WHEN SYSDATE BETWEEN os.startdate AND os.enddate THEN '강의중'  WHEN SYSDATE > os.enddate THEN '강의종료'  END AS 강의진행여부  FROM  mTeacher t  INNER JOIN mOpenCurriculum oc ON t.teacher\_id = oc.teacher\_id  INNER JOIN mCurriculum c ON oc.curriculum\_id = c.curriculum\_id  INNER JOIN mOpenCurriSubject ocs ON oc.open\_curriculum\_id = ocs.open\_curriculum\_id  INNER JOIN mOpenSubject os ON ocs.open\_subject\_id = os.open\_subject\_id  LEFT JOIN mOpensubjectBook osb ON os.open\_subject\_id = osb.open\_subject\_id  LEFT JOIN mBook b ON osb.book\_id = b.book\_id  LEFT JOIN mRoom r ON oc.room\_id = r.room\_id;  SELECT \*  FROM teacher\_class\_schedule  where 교사명 = '김미숙'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | A-04(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 개설 과정 관리 요구사항 | 작성자 | 4조(김예진) |
| DML | | | |
| /\* 개설 과정 정보 출력 시> 개설 과정 명, 개설 과정 기간(시작 년월일, 종료 년월일), 강의실 명, 개설 과목 등록 여부 등 출력 \*/  SELECT  open\_curriculum\_id AS "개설 과정 ID",  title AS "과정명",  startdate AS "시작일",  enddate AS "종료일",  room\_id AS "강의실 ID",  CASE WHEN EXISTS (  SELECT 1  FROM mOpenCurriSubject  WHERE open\_curriculum\_id = mOpenCurriculum.open\_curriculum\_id)  THEN 'O' ELSE 'X' END AS "개설 과목 등록 여부",  (SELECT COUNT(\*)  FROM mOpenCurriStudent  WHERE open\_curriculum\_id = mOpenCurriculum.open\_curriculum\_id) AS "교육생 등록 인원"  FROM mOpenCurriculum  ORDER BY open\_curriculum\_id; | | | |
| /\* 특정 개설 과정 정보 출력 시 > 개설 과목 정보(과목명, 과목기간(시작 년월일, 종료 넌월일), 교재 명, 교사명) 및 등록된 교육생 정보(교육생 이름, 주민번호 뒷자리, 전화번호, 등록일, 수료 및 중도탈락)을 출력 \*/  CREATE OR REPLACE VIEW curriculum\_subject\_and\_students AS  SELECT  os.title AS 과목명,  os.startdate AS 과목시작일,  os.enddate AS 과목종료일,  b.title AS 교재명,  t.name AS 교사명,  s.name AS 교육생이름,  s.jumin AS 주민번호,  s.tel AS 전화번호,  s.registration\_date AS 등록일,  s.completion\_date AS 수료일,  s.dropout\_date AS 중도탈락일  FROM  mOpenCurriculum oc  INNER JOIN mOpenCurriSubject ocs ON oc.open\_curriculum\_id = ocs.open\_curriculum\_id  INNER JOIN mOpenSubject os ON ocs.open\_subject\_id = os.open\_subject\_id  LEFT JOIN mOpensubjectBook osb ON os.open\_subject\_id = osb.open\_subject\_id  LEFT JOIN mBook b ON osb.book\_id = b.book\_id  LEFT JOIN mTeacher t ON oc.teacher\_id = t.teacher\_id  LEFT JOIN mOpenCurriStudent cs ON oc.open\_curriculum\_id = cs.open\_curriculum\_id  LEFT JOIN mStudent s ON cs.student\_id = s.student\_id  where oc.title = 'Python과 Django를 활용한 웹 개발자 양성 과정';  SELECT \*  FROM curriculum\_subject\_and\_students; | | | |
| /\* 과정 정보 입력 시 > (시작 년원일, 종료 년월일), 강의실 정보 입력, 강의실 정보는 기초 정보 강의실명에서 선택적으로 추가 필요  \*/  -- 1. 새로운 개설 과정 추가  insert into mopencurriculum values ((select count(\*) from mopencurriculum) + 1, 'AI 모델 학습 및 딥러닝을 활용한 AI개발자 양성 과정', '2025-04-01', '2025-10-01', '6', '강의 예정',48, 3, 1);  -- 2. 전체 개설 과정 조회  SELECT \* FROM mOpenCurriculum ORDER BY open\_curriculum\_id;  -- 3. 특정 개설 과정 정보 수정 (ID: 13)  update mopencurriculum set title = 'AI 모델 학습 및 머신러닝을 활용한 AI개발자 양성 과정', startdate = '2025-05-01', enddate = '2025-11-01' where open\_curriculum\_id = 13;  -- 4. 특정 개설 과정 삭제 (ID: 13)  delete from mopencurriculum where open\_curriculum\_id = 13;  /\* 특정 개설 과정은 수료된 경우 등록된 교육생 전체에 대해 수료 날짜 지정 > 중도 탈락자 제외 \*/  -- 5. 특정 개설 과정 수료 처리 (ID: 3)  UPDATE mStudent st  SET completion\_date = SYSDATE  WHERE st.student\_id IN (  SELECT ocs.student\_id  FROM mOpenCurriStudent ocs  JOIN mStudent s ON ocs.student\_id = s.student\_id  WHERE ocs.open\_curriculum\_id = 3  AND s.dropout\_date IS NULL  ); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | A-05(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 개설 과목 관리 요구사항 | 작성자 | 4조(김예진) |
| DML | | | |
| /\* 개설 과목 출력 시 > 개설 과정 정보(과정 명, 과정 기간(시작 년월일, 종료 년월일), 강의실)와 과목명, 과목 기간(시작 년월일, 종료 년월일), 교재 명, 교사 명 출력 \*/  CREATE OR REPLACE VIEW curriculum\_subject\_info AS  SELECT  oc.title AS 과정명, -- 개설 과정명  oc.startdate AS 과정시작일, -- 개설 과정 시작일  oc.enddate AS 과정종료일, -- 개설 과정 종료일  r.room\_id AS 강의실번호, -- 강의실 번호  os.title AS 과목명, -- 과목명  os.startdate AS 과목시작일, -- 과목 시작일  os.enddate AS 과목종료일, -- 과목 종료일  b.title AS 교재명, -- 교재명  t.name AS 교사명 -- 교사명  FROM  mOpenCurriculum oc  INNER JOIN mCurriculum c ON oc.curriculum\_id = c.curriculum\_id -- 개설 과정과 연결  LEFT JOIN mRoom r ON oc.room\_id = r.room\_id -- 강의실과 연결  INNER JOIN mOpenCurriSubject ocs ON oc.open\_curriculum\_id = ocs.open\_curriculum\_id -- 개설 과정과 개설 과목 연결  INNER JOIN mOpenSubject os ON ocs.open\_subject\_id = os.open\_subject\_id -- 개설 과목과 연결  LEFT JOIN mOpensubjectBook osb ON os.open\_subject\_id = osb.open\_subject\_id -- 교재 정보와 연결  LEFT JOIN mBook b ON osb.book\_id = b.book\_id -- 교재명  INNER JOIN mTeacher t ON oc.teacher\_id = t.teacher\_id; -- 교사 정보와 연결  SELECT \*  FROM curriculum\_subject\_info  where 과목명 = 'AWS'; | | | |
| /\* 특정 개설 과정 선택 시 > 개설 과목 정보 출력 및 개설 과목 신규 등록으로 나눔 \*/  CREATE OR REPLACE VIEW open\_curriculum\_info AS  SELECT  oc.title AS 개설과정명,  oc.startdate AS 개설과정시작일,  oc.enddate AS 개설과정종료일,  r.room\_id AS 강의실번호,  CASE  WHEN os.open\_subject\_id IS NOT NULL THEN '등록됨'  ELSE '미등록'  END AS 개설과목등록여부,  COUNT(DISTINCT cs.student\_id) AS 등록교육생수 -- DISTINCT를 사용하여 중복 교육생을 제거  FROM  mOpenCurriculum oc  LEFT JOIN mRoom r ON oc.room\_id = r.room\_id -- 강의실 정보  LEFT JOIN mOpenCurriSubject ocs ON oc.open\_curriculum\_id = ocs.open\_curriculum\_id -- 개설 과목  LEFT JOIN mOpenSubject os ON ocs.open\_subject\_id = os.open\_subject\_id -- 과목명  LEFT JOIN mOpenCurriStudent cs ON oc.open\_curriculum\_id = cs.open\_curriculum\_id -- 교육생 정보  GROUP BY  oc.title, oc.startdate, oc.enddate, r.room\_id, os.open\_subject\_id; -- 중복을 제거할 때 GROUP BY 사용  SELECT distinct \* from open\_curriculum\_info  WHERE 개설과정명 = '빅데이터 분석과 데이터구조를 활용한 데이터 분석 전문가 과정'; | | | |
| /\* 교사 명단은 현재 과목과 강의 가능 과목이 일치 하는 교사 명단만 표시 \*/  CREATE OR REPLACE VIEW available\_teachers\_for\_current\_subject AS  SELECT  t.name AS 교사명, -- 교사명  os.title AS 현재과목, -- 개설 과목명  s.title AS 강의가능과목 -- 강의 가능한 과목  FROM  mTeacher t  INNER JOIN mAvailableSubject asub ON t.teacher\_id = asub.teacher\_id -- 교사와 강의 가능한 과목 연결  INNER JOIN mOpenCurriculum oc ON oc.teacher\_id = t.teacher\_id -- 개설 과정과 교사 연결  INNER JOIN mOpenCurriSubject ocs ON oc.open\_curriculum\_id = ocs.open\_curriculum\_id -- 개설 과정과 개설 과목 연결  INNER JOIN mOpenSubject os ON ocs.open\_subject\_id = os.open\_subject\_id -- 개설 과목과 과목 연결  INNER JOIN mSubject s ON asub.subject\_id = s.subject\_id -- 강의 가능한 과목  WHERE  asub.subject\_id = os.subject\_id; -- 강의 가능한 과목과 개설 과목이 일치하는 교사만  SELECT \* FROM available\_teachers\_for\_current\_subject; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | A-06(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 교육생 관리 요구사항 | 작성자 | 4조(정다음) |
| DML | | | |
| /\* 교육생 정보 출력 시 > 교육생 이름, 주민번호 뒷자리, 전화번호, 등록일, 수강(신청) 횟수 출력 \*/  WITH student\_info AS (  SELECT s.student\_id, s.name, s.jumin, s.tel, s.registration\_date  FROM mstudent s  WHERE s.name = '김희재'  )  select  si.name AS 교육생명,  si.jumin AS "주민번호 뒷자리",  si.tel AS 전화번호,  si.registration\_date AS 등록일,  (SELECT COUNT(\*) FROM mopencurristudent c WHERE c.student\_id = si.student\_id) AS "수강(신청) 횟수"  from student\_info si; | | | |
| /\* 특정 교육생 정보 선택 시 > 수강 신청한 강의 또는 수강 중, 수강했던 개설 과정 정보(과정 명, 과정 기간(시작 년월일, 종료 년월일), 강의실, 수료 및 중도탈락 여부, 수료 및 중도탈락 날짜) 출력 \*/  select  교육생명,  i.title as 과정명,  i.startdate as "과정 시작일",  i.enddate as "과정 종료일",  i.room\_id as 강의실명,  case  when t.completion\_date is not null then '수료'  else '중도 탈락'  end as "수료(중도 탈락) 여부",  case  when t.completion\_date is not null then t.completion\_date  else null  end  from vwSubjectTest t  inner join vwcurrinfo i  on t.student\_id = i.student\_id  where 교육생명 = '박동현'; | | | |
| /\* 교육생 정보 CRUD 필요 \*/  select \* from mstudent  where name = '김희재';  -- CRUD  -- 교육생 정보 수정  update mStudent set name = '조인제' where Student\_id = 337;  -- 교육생 정보 삭제  delete from mStudent where Student\_id = 337;  -- 교육생 정보 조회  SELECT \* FROM mStudent; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | A-07(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 시험 관리 및 성적 조회 요구사항 | 작성자 | 4조(정다음) |
| DML | | | |
| /\* 특정 개설 과정 선택 시> 개설 과목 정보와 개설 과목 별로 성적 등록 여부, 시험 문제 파일 등록 여부 확인\*/  -- 특정 과정 선택시, 과목 확인(1-1)  select c.title as "과정", j.title as "과목" from mopencurrisubject cj  inner join mopensubject j  on cj.open\_subject\_id = j.open\_subject\_id  inner join mopencurriculum c  on cj.open\_curriculum\_id = c.open\_curriculum\_id  where c.title = 'AWS와 Client 와 SPRING을 활용한 클라우드 개발자 양성과정';  -- 개설 과목별로 성적 등록 여부, 시험 문제 파일 등록 여부를 확인 (1-2)  select  j.title, x.title, t.exam\_type,  case  when x.exam\_date is null then '미등록'  when x.exam\_date is not null then '등록'  end as "성적 등록 여부",  case  when x.exam\_id is null then '미등록'  when x.exam\_id is not null then '등록'  end as "시험 문제 파일 등록 여부"  from mopencurrisubject c  inner join mexam x  on c.open\_curriculum\_subject\_id = x.open\_curriculum\_subject\_id  inner join mexamtype t  on x.exam\_type\_id = t.exam\_type\_id  inner join mopensubject j  on c.open\_subject\_id = j.open\_subject\_id; | | | |
| /\* 성적 정보 출력 시 > 개설 과목 별, 교육생 개인 별로 출력 기능 필요\*/  create or replace view vwSubjectTest  as  select  s.name as 교육생명, j.title as 과목, e.title as 시험명,  case  when e.exam\_type\_id = 1 then '필기'  when e.exam\_type\_id = 2 then '실기'  end as 시험유형, r.score 성적, s.student\_id, s.jumin, s.completion\_date, j.startdate as "과목 시작일", j.enddate as "과목 종료일"  from mexamresult r  inner join mstudent s  on r.student\_id = s.student\_id  inner join mexam e  on r.exam\_id = e.exam\_id  inner join mopencurrisubject cj  on cj.open\_curriculum\_subject\_id = e.open\_curriculum\_subject\_id  inner join mopensubject j  on cj.open\_subject\_id = j.open\_subject\_id;  -- 개설 과목별 성적 출력  select 교육생명, 과목, 시험명, 시험유형, 성적 from vwsubjecttest  where 과목 = 'AWS';  -- 교육생별 성적 출력  select 교육생명, 과목, 시험명, 시험유형, 성적 from vwsubjecttest  where 교육생명 = '민유리'; | | | |
| -- 개설 현황 뷰 생성  create or replace view vwopenCurrStatus  as  select  oc.title as "개설 과정명",  oc.startdate as "개설 시작일",  oc.enddate as "개설 종료일",  oc.status as "개설 현황",  oc.room\_id as "강의실명",  os.title as "개설과목명",  oc.teacher\_id,  cs.open\_subject\_id,  os.startdate as "과목 시작일",  os.enddate as "과목 종료일"  from mopencurrisubject cs  inner join mopencurriculum oc  on cs.open\_curriculum\_id = oc.open\_curriculum\_id  inner join mopensubject os  on cs.open\_subject\_id = os.open\_subject\_id; | | | |
| /\* 과목 별 출력 시 > 개설 과정 명, 개설 과정 기간, 강의실 명, 개설 과목 명, 교사 명, 교재 명, 등을 출력하고, 해당 개설 과목을 수강한 모든 교육생들의 성적 정보(교육생 이름, 주민번호 뒷자리, 필기, 실기)를 같이 출력\*/  --과목별 출력 시 개설 과정명, 개설 과정 기간, 강의실명, 개설 과목명, 교사명, 교재명 등을 출력(2-1)  --개설 과정명, 개설 시작일, 개설 종료일, 개설 현황, 강의실명, 개설과목명  create or replace view vwCurrInfo  as  select 개설과목명, "개설 과정명", "개설 시작일", "개설 종료일", "개설 현황", 강의실명, t.name as 교사명, b.title 교재명 from mopensubjectbook sb  inner join vwopenCurrStatus c  on sb.open\_subject\_id = c.open\_subject\_id  inner join mbook b  on sb.book\_id = b.book\_id  inner join mteacher t  on c.teacher\_id = t.teacher\_id;  --해당 개설 과목을 수강한 모든 교육생들의 성적 정보(교육생 이름, 주민번호 뒷자리, 필기, 실기)를 같이 출력한다.(2-2)  select 과목, 교육생명, jumin as "주민번호 뒷자리", 시험명, 시험유형, 성적  from vwSubjectTest  where 과목 = '자바'; | | | |
| /\* 교육생 개인 별 출력 시 > 교육생 이름, 주민번호 뒷자리, 개설 과정 명, 개설 과정 기간, 강의실 명, 교육생 개인이 수강한 모든 개설 과목에 대한 성적 정보(개설 과목 명, 개설 과목 기간, 교사 명, 출결, 필기, 실기)를 같이 출력한다.\*/  -- 과정명, 기간, 강의실명  create or replace view vwcurrInfo  as  select c.title, c.startdate, c.enddate, c.status, c.room\_id, c.open\_curriculum\_id, t.name as 교사명, os.student\_id from mopencurristudent os  inner join mopencurriculum c  on os.open\_curriculum\_id = c.open\_curriculum\_id  inner join mteacher t  on t.teacher\_id = c.teacher\_id;  -- 교육생 이름, 민증, 과목명, 시험명, 유형, 성적  create or replace view vwStudentAlldate  as  select 교육생명, jumin as "주민번호 뒷자리",  i.title as "개설 과정명", i.startdate as "과정 시작일", i.enddate as "과정 종료일", i.status as "개설 현황",  i.room\_id as 강의실,  과목, 과목 시작일, 과목 종료일, 시험명, 시험유형, 성적, 교사명  from vwSubjectTest t  inner join vwcurrinfo i  on t.student\_id = i.student\_id;  select \* from vwStudentAlldate  where 교육생명 = '김기민'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | A-08(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 출결 관리 및 출결 조회 요구사항 | 작성자 | 4조(정다음) |
| DML | | | |
| /\* 특정 개설 과정 선택 시 > 해당 과정 모든 교육생의 출결 조회, 출결 현황을 기간 별(년, 월, 일)로 조회 필요  특정 과정, 특정 인원 출결 현황 조회 필요, 출결 조회 시 근태상황 구분 필요 > 정상, 지각, 조퇴, 외출, 병가, 기타 \*/  select \* from mopencurristudent os  inner join mstudent s  on os.student\_id = s.student\_id;  CREATE OR REPLACE VIEW vwcurriStudent  as  select c.title, s.name, s.student\_id from mopencurristudent os  inner join mopencurriculum c  on os.open\_curriculum\_id = c.open\_curriculum\_id  inner join mstudent s  on os.student\_id = s.student\_id  where c.status = '강의 진행중';  create or replace view vwStudentTime  as  select s.name, s.student\_id, t.clock\_out, t.clock\_in, a.name as status from mStudent s  left outer join mTimekeeping t  on s.student\_id = t.student\_id  left outer join mAttendanceType a  on t.attendance\_type\_id = a.attendance\_type\_id  order by t.clock\_in;  select \* from vwStudentTime; | | | |
| create or replace view vwCurriAttance  as  select s.title as "과목명", s.name as "교육생명", t.clock\_in as "입실", t.clock\_out as "퇴실", t.status as "근태 상황" from vwcurriStudent s  right outer join vwStudentTime t  on s.student\_id = t.student\_id  order by "입실";  select \* from vwCurriAttance;  -- 특정 개설 과정 선택 시 -> 해당 과정 모든 교육생의 출결 조회 > 출결 현황을 기간 별(년, 월, 일)로 조회  --특정 과정  select \* from vwCurriAttance  where 과목명 = 'Firebase와 React를 활용한 모바일 앱 개발 과정';  -- 특정 인원 출결 현황 조회 필요 > 출결 조회 시 근태상황 구분 필요 > 정상, 지각, 조퇴, 외출, 병가, 기타  --특정 인원  select \* from vwCurriAttance  where 교육생명 = '박호진'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-01(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 기초 정보 관리 요구사항 | 작성자 | 4조(조인제) |
| DML | | | |
| --관리자  --1. 관리자 정보 등록(관리자번호, 관리자 이름)  insert into mManager values (6, '조민제', '66666');  --2. 관리자 정보 수정  update mManager set name = '조인제' where Manager\_id = 6;  --3. 관리자 정보 삭제  delete from mManager where Manager\_id = 6;  --4. 관리자 정보 조회(관리자번호, 관리자 이름)을 출력한다.  SELECT \* FROM mManager | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-07 | 작성일 | 2025-02-06 |
| 요구사항명 | 과목별 성적 조회기능 | 작성자 | 4조(조인제) |
| DML | | | |
| – 과목번호, 교육생이름, 전화번호, 수료일, 수료여부, 출결점수, 필기점수, 실기점수를 출력  CREATE OR REPLACE VIEW vScoresBySubject AS  SELECT  ocs.open\_curriculum\_subject\_id as 과목번호,  s.name AS 교육생이름,  s.tel AS 전화번호,  case when s.dropout\_date IS NOT NULL THEN s.dropout\_date  WHEN s.dropout\_date IS NULL THEN s.completion\_date END AS 수료일,  CASE WHEN s.dropout\_date IS NOT NULL THEN '중도탈락'  WHEN s.completion\_date <= SYSDATE THEN '수료'  ELSE '수강중'  END AS 수료여부,  CASE  WHEN s.dropout\_date IS NOT NULL THEN null -- 중도 탈락자는 출결 0  ELSE  COALESCE(MAX(do.attendance), 0)  - SUM(CASE WHEN at.name = '기타' THEN 1 ELSE 0 END)  - FLOOR((SUM(CASE WHEN at.name = '지각' THEN 1 ELSE 0 END)  + SUM(CASE WHEN at.name = '조퇴' THEN 1 ELSE 0 END)) \* 0.5)  END AS 출결,  MAX(CASE WHEN s.dropout\_date IS NOT NULL THEN null WHEN er.exam\_type\_id = 1 THEN er.score END) AS 필기,  MAX(CASE WHEN s.dropout\_date IS NOT NULL THEN null WHEN er.exam\_type\_id = 2 THEN er.score END) AS 실기  FROM  mStudent s  LEFT JOIN mTimeKeeping tk ON s.student\_id = tk.student\_id  LEFT JOIN mAttendanceType at ON tk.attendance\_Type\_id = at.attendance\_Type\_id  LEFT JOIN mExamResult er ON er.student\_id = s.student\_id  LEFT JOIN mExamType et ON er.exam\_type\_id = et.exam\_type\_id  LEFT JOIN mExam e ON er.exam\_id = e.exam\_id  LEFT JOIN distributionOfScores do ON e.exam\_id = do.exam\_id  LEFT JOIN mOpenCurriSubject ocs ON e.open\_curriculum\_subject\_id = ocs.open\_curriculum\_subject\_id  LEFT JOIN mOpenSubject os ON ocs.open\_subject\_id = os.open\_subject\_id  GROUP BY ocs.open\_curriculum\_subject\_id, os.open\_subject\_id, s.name, s.tel, s.completion\_date, s.dropout\_date  ORDER BY ocs.open\_curriculum\_subject\_id;  select 교육생이름, 전화번호, 수료일, 수료여부, 출결, 필기, 실기 from vScoresBySubject; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-05 | 작성일 | 2025-02-06 |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 4조(조인제) |
| DML | | | |
| 과목 목록 출력시 과목번호, 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 출결, 필기, 실기 배점, 성적 등록 여부 등이 출력되고, 특정 과목을 과목번호로 선택시 교육생 정보(이름, 전화번호, 수료 또는 중도탈락) 및 성적이 출결, 필기, 실기 점수로 구분되어서 출력되어야 한다.  CREATE OR REPLACE VIEW vOpenSubjectsByCurriculum AS  SELECT  t.teacher\_id AS 교사번호,  oc.open\_curriculum\_id AS 개설과정번호,  oc.title AS 개설과정명,  oc.startdate AS 과정시작일,  oc.enddate AS 과정종료일,  r.room\_id AS 강의실,  os.title AS 개설과목명,  os.startdate AS 개설과목시작일,  os.enddate AS 개설과목종료일,  b.title AS 교재명,  os.open\_subject\_id AS 개설과목번호,  CASE  WHEN er.score IS NOT NULL THEN '등록' ELSE '미등록' END AS 성적등록여부,  CASE  WHEN er.score IS NULL THEN NULL  ELSE do.attendance  END AS 출결배점,  CASE  WHEN er.score IS NULL THEN NULL  ELSE do.writen  END AS 필기배점,  CASE  WHEN er.score IS NULL THEN NULL  ELSE do.practical  END AS 실기배점  FROM  mOpenCurriculum oc  INNER JOIN mRoom r ON oc.room\_id = r.room\_id  INNER JOIN mTeacher t ON oc.teacher\_id = t.teacher\_id  INNER JOIN mOpenCurriSubject ocs ON oc.open\_curriculum\_id = ocs.open\_curriculum\_id  INNER JOIN mOpenSubject os ON ocs.open\_subject\_id = os.open\_subject\_id  INNER JOIN mOpensubjectBook ob ON os.open\_subject\_id = ob.open\_subject\_id  INNER JOIN mBook b ON ob.book\_id = b.book\_id  INNER JOIN mExam e ON ocs.open\_curriculum\_subject\_id = e.open\_curriculum\_subject\_id  INNER JOIN mExamResult er ON e.exam\_id = er.exam\_id  INNER JOIN distributionOfScores do ON er.exam\_id = do.exam\_id  INNER JOIN mTeacher t ON oc.teacher\_id = t.teacher\_id;  select 교사번호, 개설과정번호, 개설과정명, 과정시작일, 과정종료일, 강의실, 개설과목명, 개설과목시작일, 개설과목종료일, 교재명, 개설과목번호, 성적등록여부, 출결배점, 필기배점, 실기배점 from vOpenSubjectsByCurriculum;  -- 자신이 강의한 과목에 한해 선택하는 경우 모든 교육생의 출결을 조회  CREATE OR REPLACE VIEW vAttendanceDetails AS  SELECT  EXTRACT(YEAR FROM tk.clock\_in) AS 출결년도,  EXTRACT(MONTH FROM tk.clock\_in) AS 출결월,  EXTRACT(DAY FROM tk.clock\_in) AS 출결일,  T.teacher\_id AS 교사번호,  oc.open\_curriculum\_id AS 과정번호,  oc.title AS 과정명,  st.student\_id AS 교육생번호,  st.name AS 교육생이름,  tk.clock\_in AS 입실,  tk.clock\_out AS 퇴실,  at.name AS 근태유형  FROM mTimeKeeping tk  RIGHT JOIN mStudent st ON tk.student\_id = st.student\_id  RIGHT JOIN mOpenCurriStudent ocs ON st.student\_id = ocs.student\_id  RIGHT JOIN mOpenCurriculum oc ON ocs.open\_curriculum\_id = oc.open\_curriculum\_id  RIGHT JOIN mAttendanceType at ON tk.attendance\_type\_id = at.attendance\_type\_id  RIGHT JOIN mTeacher t ON oc.teacher\_id = T.teacher\_id;  -- 2025년 전체 출결 조회  SELECT 교사번호, 과정번호, 과정명, 출결년도, 출결월, 출결일, 교육생번호, 교육생이름, 입실, 퇴실, 근태유형  FROM vAttendanceDetails  WHERE 교사번호 = 7;  -- 2025년 1월 출결 조회  SELECT 교사번호, 과정번호, 과정명, 출결년도, 출결월, 출결일, 교육생번호, 교육생이름, 입실, 퇴실, 근태유형  FROM vAttendanceDetails  WHERE 교사번호 = 7 AND 출결년도 = 2025  AND 출결월 = 1;  -- 특정 개설 과정(개설과정번호: 12)의 모든 교육생 출결 조회  SELECT 교사번호, 과정번호, 과정명, 출결년도, 출결월, 출결일, 교육생번호, 교육생이름, 입실, 퇴실, 근태유형  FROM vAttendanceDetails  WHERE 교사번호 = 7  AND 과정번호 = 11;  -- 특정 개설 과정(12) + 특정 교육생(교육생이름: 정수하) 출결 조회  SELECT 교사번호, 과정번호, 과정명, 출결년도, 출결월, 출결일, 교육생번호, 교육생이름, 입실, 퇴실, 근태유형  FROM vAttendanceDetails  WHERE 교사번호 = 7  AND 개설과정번호 = 12  AND 교육생이름 = '이태영';  -- 특정 근태 유형(지각)만 조회 (2025년 2월)  SELECT 교사번호, 과정번호, 과정명, 출결년도, 출결월, 출결일, 교육생번호, 교육생이름, 입실, 퇴실, 근태유형  FROM vAttendanceDetails  WHERE 교사번호 = 7  AND 근태유형 = '지각'  AND 출결년도 = 2025  AND 출결월 = 1; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-05 | 작성일 | 2025-02-06 |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 조인제 |
| DML | | | |
| -- 중도탈락된 수강생 기준 필기, 실기 점수 실데이터를 null 로 변경하는 프로시저 작성하기  CREATE OR REPLACE PROCEDURE DropoutStudentScoreUpdate  IS  BEGIN  -- 중도탈락된 교육생이 dropout\_date 이후에 본 시험 점수를 NULL로 업데이트  UPDATE mExamResult er  SET er.score = NULL  WHERE EXISTS (  SELECT 1  FROM mStudent s  JOIN mExam e ON er.exam\_id = e.exam\_id  WHERE er.student\_id = s.student\_id  AND s.dropout\_date IS NOT NULL  AND e.exam\_date > s.dropout\_date -- dropout\_date 이후의 시험만 반영  );  COMMIT;  END;  /  begin  DropoutStudentScoreUpdate;  end;  /  -- 중도탈락생 시험 점수 조회  SELECT er.student\_id, er.exam\_id, er.score, s.dropout\_date, e.exam\_date  FROM mExamResult er  JOIN mStudent s ON er.student\_id = s.student\_id  JOIN mExam e ON er.exam\_id = e.exam\_id  WHERE s.dropout\_date IS NOT NULL  AND e.exam\_date > s.dropout\_date; inner join tblapplicant app  on app.applicant\_seq = int.applicant\_seq  inner join tblcurriculum cul  on cul.curriculum\_seq = app.curriculum\_seq  inner join tbllecture lec  on lec.curriculum\_seq = cul.curriculum\_seq  where substr(to\_char(app.applicantssn),8) = '1448626'; -- 교육생 본인 주민등록번호 뒷자리 7글자 입력 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-01 | 작성일 | 2025-02-06 |
| 요구사항명 | 교사 계정 | 작성자 | 4조(김형년) |
| DML | | | |
| - 로그인 과정 후 일부 기능 사용  - 교사 계정은 관리자에 의해 데이터베이스에 등록  - 교사가 이용 할 수 있는 기능  - 강의 스케줄 조회, 배점 입출력, 성적 입출력 기능, 출결 조회 기능 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-02 | 작성일 | 2025-02-06 |
| 요구사항명 | 강의 스케줄 조회 | 작성자 | 4조(김형년) |
| DML | | | |
| - 강의 스케줄 조회  -- 교재명까지 VIEW  -- 강의 예정, 강의 중, 강의 종료로 구분  -- 날짜기준으로 확인  - 필요했던 VIEW  CREATE OR REPLACE VIEW vSubAndCurriAndBook  AS  SELECT s.subject\_id 과목번호,  c.title 과정명,  oc.startdate 과정시작일,  oc.enddate 과정종료일,  oc.room\_id 강의실번호,  s.title 과목명,  os.startdate 과목시작일,  os.enddate 과목종료일,  b.title 교재명,  t.jumin 교사비밀번호,  t.name 교사이름,  CASE  WHEN sysdate >= oc.enddate THEN '강의종료'  WHEN sysdate < oc.enddate THEN '강의중'  WHEN sysdate < oc.startdate THEN '강의예정'  END 강의여부,  ocs.open\_curriculum\_subject\_id 개설과정중간번호,  oc.open\_curriculum\_id 개설과정번호  FROM mSubject s  INNER JOIN mOpenSubject os  ON s.subject\_id = os.subject\_id  INNER JOIN mOpenCurriSubject ocs  ON os.open\_subject\_id = ocs.open\_subject\_id  INNER JOIN mOpenCurriculum oc  ON ocs.open\_curriculum\_id = oc.open\_curriculum\_id  INNER JOIN mCurriculum c  ON oc.curriculum\_id = c.curriculum\_id  INNER JOIN mOpensubjectBook obs  ON os.open\_subject\_id = obs.open\_subject\_id  INNER JOIN mBook b  ON obs.book\_id = b.book\_id  RIGHT JOIN mTeacher t  ON t.teacher\_id = oc.teacher\_id;  -- 교육생 정보 VIEW  CREATE OR REPLACE VIEW vStudent  AS  SELECT st.name 교육생명,  st.tel 연락처,  st.registration\_date 등록일,  COUNT(st.student\_id) OVER(PARTITION BY c.title) AS 교육생수,  case  WHEN st.dropout\_date IS NOT NULL THEN '중도탈락'  WHEN st.completion\_date IS NOT NULL THEN '수료'  ELSE '수료 중'  END 수료여부,  ocst.open\_curriculum\_id 개설과정번호  FROM mCurriculum c  INNER JOIN mOpenCurriculum oc  ON c.curriculum\_id = oc.curriculum\_id  INNER JOIN mOpenCurriStudent ocst  ON oc.open\_curriculum\_id = ocst.open\_curriculum\_id  INNER JOIN mStudent st  ON ocst.student\_id = st.student\_id;  --교재명+교육생정보 = 강의 스케줄 정보  CREATE OR REPLACE VIEW vClassSchedule  AS  SELECT b.교사비밀번호,  b.과목번호,  b.강의여부,  b.과정명,  b.과정시작일,  b.과정종료일,  b.강의실번호,  b.과목명,  b.과목시작일,  b.과목종료일,  b.교재명,  s.교육생명,  s.교육생수,  s.등록일,  s.수료여부  FROM vSubAndCurriAndBook b  INNER Join vStudent s  ON b.개설과정번호 = s.개설과정번호;  -- 자신의 강의 스케줄 확인  SELECT 과목번호,  강의여부,  과정명,  과정시작일,  과정종료일,  강의실번호,  과목명,  과목시작일,  과목종료일,  교재명,  교육생수,  교육생명,  등록일,  수료여부  FROM vClassSchedule  WHERE 교사비밀번호 = 1345678; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-03 | 작성일 | 2025-02-06 |
| 요구사항명 | 배점 입출력 | 작성자 | 4조(김형년) |
| DML | | | |
| - 필요했던 VIEW  CREATE OR REPLACE VIEW vSubAndCurriAndBook  AS  SELECT s.subject\_id 과목번호,  c.title 과정명,  oc.startdate 과정시작일,  oc.enddate 과정종료일,  oc.room\_id 강의실번호,  s.title 과목명,  os.startdate 과목시작일,  os.enddate 과목종료일,  b.title 교재명,  t.jumin 교사비밀번호,  t.name 교사이름,  CASE  WHEN sysdate >= oc.enddate THEN '강의종료'  WHEN sysdate < oc.enddate THEN '강의중'  WHEN sysdate < oc.startdate THEN '강의예정'  END 강의여부,  ocs.open\_curriculum\_subject\_id 개설과정중간번호,  oc.open\_curriculum\_id 개설과정번호  FROM mSubject s  INNER JOIN mOpenSubject os  ON s.subject\_id = os.subject\_id  INNER JOIN mOpenCurriSubject ocs  ON os.open\_subject\_id = ocs.open\_subject\_id  INNER JOIN mOpenCurriculum oc  ON ocs.open\_curriculum\_id = oc.open\_curriculum\_id  INNER JOIN mCurriculum c  ON oc.curriculum\_id = c.curriculum\_id  INNER JOIN mOpensubjectBook obs  ON os.open\_subject\_id = obs.open\_subject\_id  INNER JOIN mBook b  ON obs.book\_id = b.book\_id  RIGHT JOIN mTeacher t  ON t.teacher\_id = oc.teacher\_id;  -- 배점  CREATE OR REPLACE VIEW vDistribution  AS  SELECT d.attendance 출결배점,  d.writen 필기배점,  d.practical 실기배점,  d.distribution\_id 배점번호,  e.open\_curriculum\_subject\_id 개설과정중간번호,  d.exam\_id 시험번호  FROM distributionofscores d  INNER JOIN mExam e  ON d.exam\_id = e.exam\_id  INNER JOIN mOpenCurriSubject ocs  ON e.open\_curriculum\_subject\_id = ocs.open\_curriculum\_subject\_id  INNER JOIN mOpenCurriculum oc  ON ocs.open\_curriculum\_id = oc.open\_curriculum\_id  INNER JOIN mCurriculum c  ON oc.curriculum\_id = c.curriculum\_id;  select \* from vDistribution;  --시험 날짜, 시험 문제  CREATE OR REPLACE VIEW vExamDate  AS  SELECT e.exam\_date 시험날짜,  e.questions 문제,  e.exam\_id 시험번호  FROM mExam e  INNER JOIN mOpenCurriSubject ocs  ON ocs.open\_curriculum\_subject\_id = e.open\_curriculum\_subject\_id  INNER JOIN mOpenCurriculum oc  ON ocs.open\_curriculum\_id = oc.open\_curriculum\_id  INNER JOIN mCurriculum c  ON c.curriculum\_id = oc.curriculum\_id;  **- 출결 최소 20점, 출결, 필기, 실기의 합이 100이 되도록 한다.**  -- 배점 테이블  CREATE TABLE distributionOfScores (  distribution\_id number primary key,  attendance number null CHECK (attendance BETWEEN 20 AND 100),  written number null CHECK (written BETWEEN 10 AND 80),  practical number null CHECK (practical BETWEEN 10 AND 80),  CONSTRAINT check\_total\_score CHECK (practical + written + attendance = 100),  -- 시험별로 넣기  exam\_id NUMBER,  CONSTRAINT FK\_distributionOfScores\_mExam FOREIGN KEY (exam\_id) REFERENCES mExam(exam\_id)  );  **- 과목 목록 출력 시**  **-> 과목번호, 과정 명, 과정 기간(시작 년월일, 종료 년월일), 강의실, 과목 명 , 과목 기간(시작 년월일, 종료 년월일), 교재 명, 출결, 필기, 실기 배점 등이 출력**  -- 과목 목록 출력 – 배점  CREATE OR REPLACE VIEW vDistributionPrint  AS  SELECT b.과목번호,  b.과정명,  b.과정시작일,  b.과정종료일,  b.강의실번호,  b.과목명,  b.과목시작일,  b.과목종료일,  b.교재명,  d.출결배점,  d.필기배점,  d.실기배점,  b.교사비밀번호,  b.교사이름,  e.시험날짜,  e.문제,  e.시험번호  FROM vSubAndCurriAndBook b  INNER JOIN vDistribution d  ON b.개설과정중간번호 = d.개설과정중간번호  INNER JOIN vExamDate e  ON e.시험번호 = d.시험번호;  -- 배점- 과목 목록 출력  SELECT 과목번호,  과정명,  과정시작일,  과정종료일,  강의실번호,  과목명,  과목시작일,  과목종료일,  교재명,  출결배점,  필기배점,  실기배점  FROM vDistributionPrint  WHERE 교사비밀번호 = 1345678;  **- 특정 과목번호 선택 시**  **-> 출결 배점, 필기 배점, 실기 배점, 시험 날짜, 시험 문제 입력 가능 화면 이동**  **- 교사가 자신이 강의를 마친 과목의 목록 중 특정 과목 선택 가능**  CREATE OR REPLACE VIEW vDistributionExamPrint  AS  SELECT d.출결배점,  d.필기배점,  d.실기배점,  ed.시험날짜,  ed.문제  FROM vExamDate ed  INNER JOIN vDistribution d  ON ed.과정명 = d.과정명;  -- 과목번호 입력  -- 출결 배점, 필기 배점, 시험 날짜, 문제, 시험 번호 출력 입력가능 화면  SELECT DISTINCT  출결배점,  필기배점,  실기배점,  시험날짜,  문제,  시험번호  FROM vDistributionPrint  WHERE 교사비밀번호 = 1345678 AND 과목번호 = 27;  **- 선택 시 해당 과목의 배점 정보를 출결, 필기, 실기로 구분 후 등록**  **- 출결, 필기, 실기의 배점 비중 -> 담당 교사가 과목별로 결정**  -- 배점 입력  INSERT INTO distributionOfScores  VALUES((select count(\*) from distributionOfScores) +1, 20, 40, 40, 21);  -- 배점 수정  UPDATE distributionOfScores  SET attendance = 25, writte = 35, practical = 40  WHERE exam\_id = 21;  **- 시험 날짜, 시험 문제 추가 기능 필요**  -- 시험 추가  INSERT INTO mExam  VALUES((select count(\*) from mExam) +1, 'ORACLE 필기', 1, 35, 21, '2025-02-05');  -- 시험 수정  UPDATE mExam  SET questions = 50, exam\_date = '2025-02-05'  WHERE exam\_id = 21; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | E-02(교육생) | 작성일 | 2025-02-06 |
| 요구사항명 | 성적 조회 | 작성자 | 4조(김두현) |
| DML | | | |
| 1. 로그인 과정 통과 시 > 교육생 개인 정보, 과정 명, 과정 기간(시작 년월일, 종료 년월일), 강의실 타이틀 출력  select s.student\_id as 교육생번호,  s.name as 교육생이름,  s.jumin as "주민번호 뒷자리",  s.tel as 전화번호,  oc.title as 과정명 ,  oc.startdate as "과정기간 시작",  oc.enddate as "과정기간 종료",  r.room\_id as "강의실 이름"  from mStudent s  left outer join mOpenCurriStudent o  on s.student\_id = o.student\_id  inner join mOpenCurriculum oc  on o.open\_curriculum\_id = oc.open\_curriculum\_id  inner join mroom r  on oc.room\_id = r.room\_id  where s.student\_id = 285;  2. 성적 정보는 과목별 목록 형태로 출력 > 성적이 등록되지 않은 과목 존재 시 과목 정보는 출력, 점수는 null로 출력  select  s.student\_id as 교육생번호,  s.name as 교육생이름,  ms.subject\_id as 과목번호,  ms.title as 과목명,  os.startdate as "과목기간 시작",  os.enddate as "과목기간 종료",  b.title as 교재명,  t.name as 교사이름,  dos.attendance as 출결배점,  dos.written as 필기배점,  dos.practical as 실기배점,  xt.exam\_type as 시험유형,  e.score as 점수,  x.exam\_date as 시험날짜,  x.title as 시험문제  from mStudent s  left join mExamResult e  on s.student\_id = e.student\_id  left join mExamType xt  on e.exam\_type\_id = xt.exam\_type\_id  left join mExam x  on e.exam\_id = x.exam\_id  left join mOpenCurriSubject ocs  on x.open\_curriculum\_subject\_id = ocs.open\_curriculum\_subject\_id  left join mOpenSubject os  on ocs.open\_subject\_id = os.open\_subject\_id  left join mSubject ms  on os.subject\_id = ms.subject\_id  left join mOpensubjectBook osb  on os.open\_subject\_id = osb.open\_subject\_id  left join mBook b  on osb.book\_id = b.book\_id  left join mAvailableSubject avs  on os.open\_subject\_id = avs.available\_subject\_id  left join mTeacher t  on avs.teacher\_id = t.teacher\_id  left join distributionOfScores dos  on x.exam\_id = dos.exam\_id  where s.student\_id = 295; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | E-03(교육생) | 작성일 | 2025-02-06 |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 4조(김두현) |
| DML | | | |
| --1. 전체 기간 조회  select  s.student\_id as 교육생번호,  s.name as 교육생이름,  tk.clock\_in as 입실시간,  clock\_out as 퇴실시간,  at.name as 근태유형명  from mStudent s  inner join mTimeKeeping tk  on s.student\_id = tk.student\_id  inner join mAttendanceType at  on tk.attendance\_type\_id = at.attendance\_type\_id  where s.student\_id = 285;  ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD HH24:MI:SS';  ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD';  2. 월별 조회  select  s.student\_id as 교육생번호,  s.name as 교육생이름,  tk.clock\_in as 입실시간,  clock\_out as 퇴실시간,  at.name as 근태유형명  from mStudent s  inner join mTimeKeeping tk  on s.student\_id = tk.student\_id  inner join mAttendanceType at  on tk.attendance\_type\_id = at.attendance\_type\_id  where s.student\_id = 285 and to\_char(tk.clock\_in, 'MM') = '01';  ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD HH24:MI:SS';  ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD';  --3. 일별 조회  select  s.student\_id as 교육생번호,  s.name as 교육생이름,  tk.clock\_in as 입실시간,  clock\_out as 퇴실시간,  at.name as 근태유형명  from mStudent s  inner join mTimeKeeping tk  on s.student\_id = tk.student\_id  inner join mAttendanceType at  on tk.attendance\_type\_id = at.attendance\_type\_id  where s.student\_id = 285 and to\_char(tk.clock\_in, 'dd') = '20';  ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD HH24:MI:SS';  ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | F-01(교육생) | 작성일 | 2025-02-06 |
| 요구사항명 | 수료생 사후처리 | 작성자 | 4조(김두현) |
| DML | | | |
| - 수료생만 관리.  - 취업정보는 회사명, 연봉, 주소를 입력함.  - 수료일, 입사일은 날짜 형식으로 입력함. ex) 2025-02-06  - 연봉 0보다 큰 값으로 입력(천단위) ex) 연봉 4000만원 > 4000 입력  - 지역은 2000자 이하 ex) 경기 성남시 중원구  1.수료생 추가  INSERT INTO mAfterService((SELECT NVL(MAX(afterService\_id), 0) + 1 FROM mAfterService\_id), gradStudent\_id, finishDate, enterDate, companyName, annualSalary, region);  2. 수료생 검색  SELECT \* FROM mAfterService WHERE gradStudent\_id = 1;  3. 수료생 수정  UPDATE mAfterService SET finishDate = TO\_DATE('2025-05-01', 'YYYY-MM-DD'), enterDate = TO\_DATE('2025-06-01', 'YYYY-MM-DD'), companyName = 'Updated Company', annualSalary = 6000, region = 'Busan' WHERE gradStudent\_id = 1;  4.수료생 삭제  DELETE FROM mAfterService WHERE gradStudent\_id = 1; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | F-02(관리자) | 작성일 | 2025-02-06 |
| 요구사항명 | 관리자 강사 평가 | 작성자 | 4조(김예진) |
| DML | | | |
| -- 관리자 평가 조회  SELECT  a.adminEval\_id AS "관리자 평가 번호",  m.name AS "관리자 이름",  t.name AS "강사 이름",  a.evalScore AS "점수",  a.evalDate AS "작성일",  a.evalContent AS "평가 내용"  FROM  mAdminTeacherEval a  JOIN  mManager m ON a.manager\_id = m.manager\_id  JOIN  mTeacher t ON a.teacher\_id = t.teacher\_id;  -- 관리자 평가 추가  INSERT INTO mAdminTeacherEval (adminEval\_id, evalScore, evalDate, evalContent, manager\_id, teacher\_id)  VALUES (51, 4, TO\_DATE('2025-02-07', 'YYYY-MM-DD'), '강사님은 수업 자료 준비가 잘 되어 있었고, 교육생들이 매우 만족스러워 했습니다.', 1, 10);  -- 관리자 평가 수정  UPDATE mAdminTeacherEval  SET evalScore = 5,  evalContent = '강사님은 수업을 매우 체계적으로 진행하였고, 교육생들에게 많은 도움이 되었습니다.'  WHERE adminEval\_id = 51;  -- 관리자 평가 삭제  DELETE FROM mAdminTeacherEval WHERE adminEval\_id = 51; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | F-03(교사) | 작성일 | 2025-02-06 |
| 요구사항명 | 자료 관리 및 자료 조회 요구사항 | 작성자 | 4조(정다음) |
| DML | | | |
| -- 자료 전체 뷰  create or replace view vwMaterial  as  SELECT m.title as "자료명", m.mcontent as "내용", m.memo as "메모", t.name as 교사명, c.title as 강의명, c.open\_curriculum\_id, t.teacher\_id FROM mmaterial m  inner join mteacher t  on m.teacher\_id = t.teacher\_id  inner join mopencurriculum c  on m.open\_curriculum\_id = c.open\_curriculum\_id;  create or replace view vwMaterial  as  SELECT m.title as "자료명", m.mcontent as "내용", m.memo as "메모", t.name as 교사명, c.title as 강의명, c.open\_curriculum\_id, t.teacher\_id FROM mmaterial m  inner join mteacher t  on m.teacher\_id = t.teacher\_id  inner join mopencurriculum c  on m.open\_curriculum\_id = c.open\_curriculum\_id;  --자료명으로 관련 자료 모두 조회  select 자료명, 내용, 메모, 교사명, 강의명 from vwmaterial  where 자료명 like '%AWS%';  --강의명으로 관련 자료 모두 조회  select 자료명, 내용, 메모, 교사명, 강의명 from vwmaterial  where 강의명 like '%AWS와 Client%';  --선생님별 조회  select 자료명, 내용, 메모, 교사명, 강의명 from vwmaterial  where 교사명 like '김미숙';  --생성  select \* from mmaterial;  insert into mmaterial values ((select count(\*) from mmaterial) +1, '하하', '호호', '히히', 1, 1, sysdate);  rollback;  --삭제  delete from mmaterial  where title = 'AWS 공식 문서'; | | | |