# 관리자\_기초 정보 관리

-- \*\*\*\*\*기초 정보 입력\*\*\*\*\*

-- 기초 과정 정보 추가하기

INSERT INTO tblCourse (course\_seq, name, goal, detail) VALUES ('L018', '과정명', '과정목표', '과정설명');

-- 기초 과목 정보 추가하기

INSERT INTO tblSubject (subject\_seq, name, type) VALUES ('SUB041', '과목명', '과목분류');

-- 기초 교재 정보 추가하기

INSERT INTO tblTextbook (textbook\_seq, name, publisher) VALUES ('B122', '교재명', '출판사');

-- 기초 강의실 정보 추가하기

INSERT INTO tblClassroom (classroom\_seq, name, capacity) VALUES ('R007', '강의실 명', '강의실 정원');

-- \*\*\*\*\*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;

-- \*\*\*\*\*기초 정보 수정\*\*\*\*\*

-- 기초 과정 정보 수정하기

UPDATE tblCourse SET name='과정명 수정' WHERE course\_seq='과정코드';

UPDATE tblCourse SET goal='과정목표 수정' WHERE course\_seq='과정코드';

UPDATE tblCourse SET detail='과정설명 수정' WHERE course\_seq='과정코드';

-- 기초 과목 정보 수정하기

UPDATE tblSubject SET name='과목명 수정' WHERE subject\_seq='과목코드';

UPDATE tblSubject SET type='과목분류 수정' WHERE subject\_seq='과목코드';

-- 기초 교재 정보 수정하기

select \* from tblClassroom;

UPDATE tblTextbook SET name='교재명 수정' WHERE textbook\_seq='교재코드';

UPDATE tblTextbook SET publisher='출판사 수정' WHERE textbook\_seq='교재코드';

-- 기초 강의실 정보 수정하기

UPDATE tblClassroom SET name='강의실명 수정' WHERE classroom\_seq='강의실코드';

UPDATE tblClassroom SET capacity='강의실정원 수정' WHERE classroom\_seq='강의실코드';

-- \*\*\*\*\*기초 정보 삭제\*\*\*\*\*

-- 기초 과정 정보 삭제

DELETE FROM tblCourse WHERE course\_seq='과정코드';

DELETE FROM tblCourse WHERE name='과정명';

-- 기초 과목 정보 삭제

DELETE FROM tblSubject WHERE subject\_seq='과목코드';

DELETE FROM tblSubject WHERE name='과목명';

-- 기초 교재 정보 삭제

DELETE FROM tblTextbook WHERE textbook\_seq='교재코드';

DELETE FROM tblTextbook WHERE name='교재명';

-- 기초 강의실 정보 삭제

DELETE FROM tblClassroom WHERE classroom\_seq='강의실코드';

DELETE FROM tblClassroom WHERE name='강의실명';

# 관리자\_교사 계정 관리

-- \*\*\*\*\*교사 계정 관리\*\*\*\*\*

SELECT \* FROM tblteacher;

Select \* from tblStudent;

**-- 교사 로그인하기**

SELECT COUNT(\*) FROM tblTeacher WHERE name='이윤후' AND idcard\_number='2482932';

**-- 학생 로그인하기**

SELECT COUNT(\*) FROM tblstudent WHERE name='정하하' AND idcard\_number='2654210';

-- 교사 정보 입력하기

INSERT INTO tblteacher (teacher\_seq, name, idcard\_number, tel) VALUES ('T011', '이름', '주민번호 뒷자리', '연락처');

-- 교사 정보 조회하기

SELECT teacher\_seq as "강사코드", name as "이름", idcard\_number as "주민번호 뒷자리", tel as "연락처" FROM tblTeacher;

-- 교사 정보 수정하기

UPDATE tblTeacher SET name='이름 수정' WHERE teacher\_seq='강사코드';

UPDATE tblTeacher SET idcard\_number='주민번호 뒷자리 수정' WHERE teacher\_seq='강사코드';

UPDATE tblTeacher SET tel='연락처 수정' WHERE teacher\_seq='강사코드';

-- 교사 정보 삭제하기

DELETE FROM tblTeacher WHERE teacher\_seq='강사코드';

DELETE FROM tblTeacher WHERE name='이름';

-- 전체 교사 목록 출력하기

SELECT a.teacher\_seq AS "강사코드", a.name AS "이름", a.idcard\_number AS "주민번호 뒷자리", a.tel AS "연락처",

LISTAGG(c.name, ',') AS "강의 가능 과목"

FROM tblteacher a

INNER JOIN tblteachersubject b

ON b.teacher\_seq = a.teacher\_seq

INNER JOIN tblSubject c

ON b.subject\_seq = c.subject\_seq

GROUP BY a.teacher\_seq, a.name, a.idcard\_number, a.tel;

-- 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

# 관리자\_개설 과정 관리

/\*

<개설 과정 입력>

- 과정 정보는 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실 정보를 입력한다.

- 강의실 정보는 기초 정보 강의실명에서 선택적으로 추가할 수 있어야 한다.

\*/

-- 기초 과정 정보 입력

INSERT INTO tblcourse (course\_seq, name, goal, detail) VALUES ('L018', '과정명', '과정목표', '과정설명');

-- **동적 과정 정보 입력**

INSERT INTO tblOpenCourse (open\_course\_seq, course\_start\_date, course\_end\_date, reg\_open\_subject, course\_progress, classroom\_seq, course\_seq)

VALUES ('OL017', TO\_DATE('2021-12-01', 'yyyy-mm-dd'), TO\_DATE('2021-12-01', 'yyyy-mm-dd'), 'Y', '과정진행여부', '강의실코드', '과정코드');

/\*

<개설 과정 출력>

- 개설 과정 정보 출력시 개설 과정명, 개설 과정기간(시작 년월일, 끝 년월일), 강의 설명, 개설 과목 등록 여부, 교육생 등록 인원을 출력한다.

- 특정 개설 과정 선택시 개설 과정에 등록된 개설 과목 정보(과목명, 과목시간(시작 년월일, 끝 년월일), 교재명, 교사명) 및 등록된 교육생 정보(교육생 이름, 주민번호 뒷자리, 전화번호, 등록일, 수료 및 중도탈락)을 확인할 수 있어야 한다.

\*/

-- 모든 개설 과정 정보 출력

SELECT b.name AS "개설 과정명", a.course\_start\_date AS "과정 시작일", a.course\_end\_date AS "과정 종료일", b.detail AS "강의설명", a.reg\_open\_subject AS "개설 과목 등록 여부",

(SELECT COUNT(d.student\_seq) FROM tblRegister d WHERE d.open\_course\_seq=a.open\_course\_seq) AS "교육생 등록 인원"

FROM tblOpenCourse a

INNER JOIN tblCourse b ON a.course\_seq=b.course\_seq

INNER JOIN tblClassroom c ON a.classroom\_seq=c.classroom\_seq;

-- 특정 개설 과정 정보 출력

-- 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

-- 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

# 관리자\_개설과목관리

--2.특정 개설 과정 선택시 개설 과목 정보 출력

SELECT OPEN\_COURSE\_SEQ

FROM TBLOPENCOURSE

WHERE SYSDATE BETWEEN COURSE\_START\_DATE AND COURSE\_END\_DATE;

-- 현재 개설 과정 OL001 ~ OL006 까지 있음

SELECT OPEN\_COURSE\_SEQ AS 개설과정번호,

OPEN\_SUBJECT\_SEQ AS 개설과목번호,

SUBJECT\_SEQ AS 과목번호,

TEACHER\_SEQ AS 교사번호,

TEXTBOOK\_SEQ AS 교재번호,

SUBJECT\_START\_DATE AS 과목시작일,

SUBJECT\_END\_DATE AS 과목종료일,

SUBJECT\_PROGRESS AS 강의진행여부,

SCORE\_CHECK AS 성적등록여부,

TESTFILE\_CHECK AS 시험문제파일등록여부,

ATTENDANCE\_DISTRIBUTION AS 출결배점,

WRITTEN\_DISTRIBUTION AS 필기배점,

PRACTICAL\_DISTRIBUTION AS 실기배점

FROM TBLOPENSUBJECT

WHERE OPEN\_COURSE\_SEQ = 'OL006';

--OL006 과정선택시 출력되는 정보들

------------------------------------------------------------------------------------------------------------------------

--2-1.특정 개설 과정 선택시 개설 과목 신규 등록

-- 특정과정 OL006 에 새로운 과목 정보 입력시

-- 과목명, 과목기간(시작, 끝), 교재명, 교사명 입력 가능해야함

-- 교재명은 기초 정보 교재명에서 선택적으로 추가할 수 있다.

-- 추가사항

-- 개설과목번호[OS081], 과목명[UI디자인], 교사명[정성훈], 교재명[모바일 UI/UX 디자인 실무], 과목시작날짜[2022-07-04], 과목종료날짜[2022-08-03]

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 ('OS081',

--(SELECT CONCAT('OS',LPAD(MAX(SUBSTR(OPEN\_SUBJECT\_SEQ,4,2))+1,3,'0')) FROM TBLOPENSUBJECT),

(SELECT SUBJECT\_SEQ FROM TBLSUBJECT WHERE NAME = 'UI디자인'),

(SELECT TEACHER\_SEQ FROM TBLTEACHER WHERE NAME = '정성훈'),

(SELECT TEXTBOOK\_SEQ FROM TBLTEXTBOOK WHERE NAME = '모바일 UI/UX 디자인 실무'),

'2022-07-04',

'2022-08-03',

CASE

WHEN (SYSDATE BETWEEN TO\_DATE('2022-07-04', 'YYYY-MM-DD') AND TO\_DATE('2022-08-03', 'YYYY-MM-DD'))

THEN '강의중'

WHEN (SYSDATE < TO\_DATE('2022-07-04', 'YYYY-MM-DD')) THEN '강의예정'

WHEN (SYSDATE > TO\_DATE('2022-08-03', 'YYYY-MM-DD')) THEN '강의종료'

ELSE '폐강'

END,

'N',

'N',

'', '', '',

'OL006');

------------------------------------------------------------------------------------------------------------------------

-- 2-2. 교사 명단은 현재 과목과 강의 가능 과목이 일치하는 교사 명단만 보여야 한다.

-- 교재명은 기초 정보 교재명에서 선택적으로 추가할 수 있어야 한다

-- 교사명은 교사 명단에서 선택적으로 추가할 수 있어야한다.

-- 교사 명단은 현재 과목과 강의 가능 과목이 일치하는 교사 명단만 보여야한다

SELECT TC.TEACHER\_SEQ AS 강사번호,

TC.NAME AS 강사이름,

SUB.SUBJECT\_SEQ AS 강의가능과목번호,

SUB.NAME AS 강의가능과목이름

FROM TBLTEACHERSUBJECT

INNER JOIN TBLTEACHER TC on TBLTEACHERSUBJECT.TEACHER\_SEQ = TC.TEACHER\_SEQ

INNER JOIN TBLSUBJECT SUB on SUB.SUBJECT\_SEQ = TBLTEACHERSUBJECT.SUBJECT\_SEQ;

------------------------------------------------------------------------------------------------------------------------

-- 2-3. 개설과목 출력시 개설 과정 정보(과정명, 과정기간(시작,끝) 강의실), 과목명, 과목기간(시작,끝),교재명,교사명

SELECT COUR.NAME AS 과정명,

OPENCOUR.COURSE\_START\_DATE AS 과정시작기간,

OPENCOUR.COURSE\_END\_DATE AS 과정종료기간,

ROOM.NAME AS 강의실명,

SUB.NAME AS 과목명,

OPENSUB.SUBJECT\_START\_DATE AS 과목시작기간,

OPENSUB.SUBJECT\_END\_DATE AS 과목종료기간,

TXT.NAME AS 교재명,

TC.NAME AS 교사명

FROM TBLOPENSUBJECT OPENSUB

INNER JOIN TBLOPENCOURSE OPENCOUR

on OPENCOUR.OPEN\_COURSE\_SEQ = OPENSUB.OPEN\_COURSE\_SEQ

INNER JOIN TBLCOURSE COUR

on COUR.COURSE\_SEQ = OPENCOUR.COURSE\_SEQ

INNER JOIN TBLCLASSROOM ROOM

on ROOM.CLASSROOM\_SEQ = OPENCOUR.CLASSROOM\_SEQ

INNER JOIN TBLSUBJECT SUB

on SUB.SUBJECT\_SEQ = OPENSUB.SUBJECT\_SEQ

INNER JOIN TBLTEXTBOOK TXT

on OPENSUB.TEXTBOOK\_SEQ = TXT.TEXTBOOK\_SEQ

INNER JOIN TBLTEACHER TC on OPENSUB.TEACHER\_SEQ = TC.TEACHER\_SEQ;

------------------------------------------------------------------------------------------------------------------------

--1. 개설 과목 정보 출력

-- (1.1)개설과목 출력

SELECT OPEN\_COURSE\_SEQ AS 개설과정번호,

OPEN\_SUBJECT\_SEQ AS 개설과목번호,

SUBJECT\_SEQ AS 과목번호,

TEACHER\_SEQ AS 교사번호,

TEXTBOOK\_SEQ AS 력과목시작일,

SUBJECT\_END\_DATE AS 과목종료일,

SUBJECT\_PROGRESS AS 강의진행여부,

SCORE\_CHECK AS 성적등록여부,

TESTFILE\_CHECK AS 시험문제파일등록여부

FROM TBLOPENSUBJECT;

-- (1.2)개설과목 입력

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 ('OS081',

--(SELECT **CONCAT('OS',LPAD(MAX(SUBSTR(OPEN\_SUBJECT\_SEQ,4,2))+1,3,'0')) FROM TBLOPENSUBJECT),**

(SELECT SUBJECT\_SEQ FROM TBLSUBJECT WHERE NAME = 'UI디자인'),

(SELECT TEACHER\_SEQ FROM TBLTEACHER WHERE NAME = '정성훈'),

(SELECT TEXTBOOK\_SEQ FROM TBLTEXTBOOK WHERE NAME = '모바일 UI/UX 디자인 실무'), '2022-07-04', '2022-08-03', '강의예정', 'N',

'N', '', '', '', 'OL016');

-- (1.3)개설과목 삭제

-- 삭제하고자 하는 행의 개설과목번호(OPEN\_SUBJECT\_SEQ)를 선택하면 된다.

-- \*\*\* WHERE문 꼭 사용

DELETE

FROM TBLOPENSUBJECT

WHERE OPEN\_SUBJECT\_SEQ = 'OS081';

-- (1.4)개설과목 수정

-- 수정하고자 하는 행의 컬럼을 선택하고 수정값을 넣고, 수정하고자하는 PK키를 넣은 뒤 실행하면 수정이 진행된다.

-- \*\*\* WHERE문 꼭 사용

UPDATE TBLOPENSUBJECT

SET TEACHER\_SEQ='T009'

WHERE OPEN\_SUBJECT\_SEQ = 'OS081';

# 관리자\_교육생관리

------------------------------------------------------------------------------------------------------------------------

-- 교육생 정보 입력시 교육생이름, 주민번호 뒷자리, 전화번호를 기본으로 등록하고

-- 주민번호 뒷자리는 교육생 패스워드로 사용, 등록일은 자동으로 입력되도록 한다

INSERT INTO TBLSTUDENT (STUDENT\_SEQ, NAME, IDCARD\_NUMBER, TEL, REGISTRATION\_DATE)

VALUES ('S0501',

--(SELECT CONCAT('S',LPAD(MAX(SUBSTR(STUDENT\_SEQ,3,3))+1,4,'0')) FROM TBLSTUDENT),

'강규준',

'1234567',

'010-5671-0790',

TO\_CHAR(SYSDATE,'RRRR-MM-DD'));

--(SELECT CONCAT('S0',MAX(SUBSTR(STUDENT\_SEQ,3,3))+1) FROM TBLSTUDENT)

--년도수정

------------------------------------------------------------------------------------------------------------------------

DELETE

FROM TBLSTUDENT

WHERE NAME = '강규준';

------------------------------------------------------------------------------------------------------------------------

-- 교육생 정보 출력시 교육생 이름, 주민번호 뒷자리, 전화번호, 등록일, 수강(신청)횟수를 출력한다

SELECT DISTINCT STU.NAME AS 교육생이름,

STU.IDCARD\_NUMBER AS 주민번호뒷자리,

STU.TEL AS 전화번호,

STU.REGISTRATION\_DATE AS 등록일,

(SELECT COUNT(COMPLETION\_STATUS) FROM TBLREGISTER WHERE STUDENT\_SEQ = STU.STUDENT\_SEQ) AS 수강신청횟수

FROM TBLSTUDENT STU

INNER JOIN TBLREGISTER REG on STU.STUDENT\_SEQ = REG.STUDENT\_SEQ;

------------------------------------------------------------------------------------------------------------------------

--특정 교육생 선택시 교육생이 수강 신청한 또는 수강중인, 수강했던

--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';

--김주혁, 주혁시

------------------------------------------------------------------------------------------------------------------------

-- 교육생 정보를 쉽게 확인하기 위한 검색 기능을 사용할 수 있어야 한다

SELECT DISTINCT STU.NAME AS 교육생이름,

STU.IDCARD\_NUMBER AS 주민번호뒷자리,

STU.TEL AS 전화번호,

STU.REGISTRATION\_DATE AS 등록일,

(SELECT COUNT(COMPLETION\_STATUS) FROM TBLREGISTER WHERE STUDENT\_SEQ = STU.STUDENT\_SEQ) AS 수강신청횟수

FROM TBLSTUDENT STU

INNER JOIN TBLREGISTER REG on STU.STUDENT\_SEQ = REG.STUDENT\_SEQ

WHERE STU.STUDENT\_SEQ = 'S0373';

--김주혁,

------------------------------------------------------------------------------------------------------------------------

--교육생에 대한 수료 및 중도탈락처리를 할 수 있어야 한다. 수료 또는 중도탈락 날짜를 입력할 수 있어야한다.

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,

REG.STUDENT\_SEQ AS 교육생번호

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 = 'S0126';

--초기화용

UPDATE TBLREGISTER

SET COMPLETION\_STATUS='',

COMPLETION\_DATE='',

DROPOUT\_DATE=''

WHERE STUDENT\_SEQ = 'S0126'

AND OPEN\_COURSE\_SEQ = 'OL003';

--중도탈락시

UPDATE TBLREGISTER

SET COMPLETION\_STATUS='N',

DROPOUT\_DATE=TO\_DATE(SYSDATE, 'RRRR-MM-DD')

WHERE STUDENT\_SEQ = 'S0126'

AND OPEN\_COURSE\_SEQ = 'OL003';

--수료시

UPDATE TBLREGISTER

SET COMPLETION\_STATUS='Y',

COMPLETION\_DATE=TO\_DATE(SYSDATE, 'RRRR-MM-DD')

WHERE STUDENT\_SEQ = 'S0126'

AND OPEN\_COURSE\_SEQ = 'OL003';

# 관리자\_시험관리및성적조회

-- 특정 개설 과정을 선택하는 경우 등록된 개설 과목 정보를 출력하고,

-- 개설 과목 별로 성적 등록여부, 시험 문제 파일 등록 여부를 확인할 수 있어야 한다.

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';

--성적 정보 출력시 개설 과목별, 교육생 개인별로 출력할 수 있어야 한다.

--과목별출력시

--개설과정명, 개설과정기간, 강의실명, 개설과목명, 교사명, 교재명

--개설 과목을 수강한 모든 교육생의 성적정보(교육생이름, 주민번호뒷자리,필기,실기)

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

--개설과정명, 개설과정기간, 강의실명, 개설과목명, 교사명, 교재명

--개설 과목을 수강한 모든 교육생의 성적정보(교육생이름, 주민번호뒷자리,필기,실기)

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';

# 관리자\_출결 관리 및 출결 조회

-- 출결관리 및 출결조회

/\*

- 특정 개설 과정을 선택하는 경우 모든 교육생의 출결을 조회할 수 있어야 한다.

- 출결 현황을 기간별(년, 월, 일)로 조회할 수 있어야 한다.

- 특정(과정, 인원) 출결 현황을 조회할 수 있어야 한다.

- 모든 출결 조회는 근태 상황을 구분할 수 있어야 한다. (정상, 지각, 조퇴, 외출, 병가, 기타)

\*/

-- 특정 개설 과정 선택

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='OL007'--과정코드 입력

ORDER BY d.attendance\_date;

-- 출결 현황을 기간별(년, 월, 일)로 조회

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='OL007' AND d.attendance\_date BETWEEN TO\_DATE('2021-01-01', 'YYYY-MM-DD') AND TO\_DATE('2021-02-01', 'YYYY-MM-DD')--과정코드, 기간 입력

ORDER BY d.attendance\_date;

-- 특정(과정, 인원) 출결 현황 조회

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='OL007' AND c.name='박현문' --과정코드, 이름 입력

ORDER BY d.attendance\_date;

# 교사\_강의스케줄 조회

/\* 1. 강의 스케줄 조회 \*/

-- 교사는 자신의 강의 스케줄을 확인할 수 있어야 한다.

-- 강의 스케줄은 (강의예정, 강의중, 강의 종료)로 구분해서 확인할 수 있어야 한다.

-- 강의 진행 상태는 날짜를 기준으로 확인한다.

-- \* 강의 스케줄 정보 출력 시 : (과목번호, 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실명, 과목명, 과목 기간(시작 년월일, 끝 년월일), 교재명, 교육생 등록 인원) 확인할 수 있다.

-- \* 특정 과목을 과목 번호로 선택 시 : 해당 과정에 등록된 교육생 정보(교육생 이름, 전화번호, 등록일, 수료 또는 중도탈락)을 확인할 수 있다.

-- 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'

else '과정진행중'

end as 수료여부

from tblopensubject os

inner join tblregister reg on os.open\_course\_seq = reg.open\_course\_seq

inner join tblstudent student on reg.student\_seq = student.student\_seq

where os.open\_subject\_seq = 'OS001';

# 교사\_배점 입출력

/\* 2. 배점 입출력 \*/

-- 교사가 강의를 마친 과목에 대한 성적 처리를 위해서 배점 '입출력'을 할 수 있어야 한다.

-- 교사는 자신이 강의를 마친 과목의 목록 중에서 특정 과목을 선택하고 해당 과목의 배점 정보를 출결, 필기, 실기로 구분해서 등록할 수 있어야 한다.

-- 출결, 필기, 실기의 배점 비중은 담당 교사가 과목별로 결정한다. (단, 출결은 최소 20점 이상이어야 하고, 출결, 필기, 실기의 합은 100점이 되도록 한다.)

-- \* 과목 목록 출력시 :(과목번호, 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 출결, 필기, 실기 배점 등)

-- \* 특정 과목을 과목 번호로 선택 시 : (출결 배점, 필기 배점, 실기 배점, 시험 날짜, 시험 문제) 를 입력할 수 있는 화면으로 연결된다.

-- \* 배점 등록이 안된 과목인 경우 : 과목 정보 출력될 때 배점 부분은 null값으로 출력

-- 시험 날짜, 시험 문제를 추가할 수 있어야 한다.

-- 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) 배점 입력 / 시험날짜 / 시험문제 입력

-- 2.3.1) 출결배점 수정(입력)

update tblopensubject

set attendance\_distribution = 30

where open\_subject\_seq = 'OS001';

-- 2.3.2) 필기배점 수정(입력)

update tblopensubject

set written\_distribution = 30

where open\_subject\_seq = 'OS001';

-- 2.3.3) 실기배점 수정(입력)

update tblopensubject

set practical\_distribution = 40

where open\_subject\_seq = 'OS001';

-- 2.3.4.1) 시험 등록 (기존에 시험 등록을 안한 경우)

insert into tbltest values ('TEST151', current\_date, '필기', 'OS001');

-- 2.3.4.2) '시험문제파일등록여부' 컬럼 변경 (해당 과목의 시험이 필기/실기 둘 다 등록 됐을 경우)

update tblopensubject

set testfile\_check = 'Y'

where open\_subject\_seq = 'OS001'

and (select count(\*) from tblTest where open\_subject\_seq = 'OS001') >= 2;

-- 2.3.5) 시험날짜 수정(입력)

update tbltest

set test\_date = current\_date

where open\_subject\_seq = 'OS001' and type = '필기';

# 교사\_성적 입출력

/\* 3. 성적 입출력 \*/

-- 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) 특정 교육생 선택 > 해당 교육생의 시험 점수 입력(출결, 필기, 실기 점수 구분 입력)

-- 수정.ver

-- 3.3.1) 출결점수 입력(수정)

update tblscoreinfo

set attendance\_score = 15

where opensubject\_seq = 'OS001' and student\_seq = 'SOO4';

-- 3.3.2) 필기점수 입력(수정)

-- 3.3.2.1) 시험테이블 > 필기점수 입력

update tbltestresult

set score = 100

where test\_seq = (select test\_seq from tbltest where open\_subject\_seq = 'OS001' and type = '필기')

and student\_seq = 'S0004';

-- 3.3.2.2) 성적정보테이블 > 필기점수 환산 입력(수정)

update tblscoreinfo

set written\_score =

-- 점수 \* 배점

(select score from tbltestresult

where test\_seq = (select test\_seq from tbltest where open\_subject\_seq = 'OS001' and type = '필기') and student\_seq = 'S0004')

\*

(select written\_distribution / 100 from tblopensubject

where open\_subject\_seq = 'OS001')

where opensubject\_seq = 'OS001' and student\_seq = 'S0004';

-- 3.3.3) 실기점수 입력(수정)

-- 3.3.3.1) 시험테이블 > 실기점수 입력

update tbltestresult

set score = 100

where test\_seq = (select test\_seq from tbltest where open\_subject\_seq = 'OS001' and type = '실기')

and student\_seq = 'S0004';

-- 3.3.3.2) 성적정보테이블 > 실기점수 환산 입력(수정)

update tblscoreinfo

set practical\_score =

-- 점수 \* 배점

(select score from tbltestresult

where test\_seq = (select test\_seq from tbltest where open\_subject\_seq = 'OS001' and type = '실기') and student\_seq = 'S0004')

\*

(select practical\_distribution / 100 from tblopensubject

where open\_subject\_seq = 'OS001')

where opensubject\_seq = 'OS001' and student\_seq = 'S0004';

-- 3.3.4) 총점 재계산

update tblscoreinfo

set subject\_score = attendance\_score + written\_score + practical\_score

where opensubject\_seq = 'OS001' and student\_seq = 'S0004';

# 교사\_출결 관리 및 출결 조회

/\* 4. 출결 관리 및 출결 조회 \*/

-- 교사가 강의한 과정에 한해 선택하는 경우 모든 교육생의 출결을 조회할 수 있어야 한다.

-- 출결 현황을 기간별(년, 월, 일) 조회할 수 있어야 한다.

-- 특정(특정 과정, 특정 인원) 출결 현황을 조회할 수 있어야 한다.

-- 모든 출결 조회는 근태 상황을 구분할 수 있어야 한다.(정상, 지각, 조퇴, 외출, 병가, 기타)

-- 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

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 at.attendance\_date = '2021-10-01'

order by at.attendance\_date, at.student\_seq;

# 

# 

# 

# 

# 교육생\_성적 조회

--[ 교육생이 로그인에 성공하면 ]

select \* from tblstudent where student\_seq = 'S0038';

select

case

when count(\*) > 0 then '로그인 성공'

else '로그인 실패'

end as "교육생 로그인"

from tblstudent where student\_seq = 'S0038' and idcard\_number = '1770114';

--S0038번 교육생으로 로그인한다.

--[ 교육생 개인의 정보와 교육생이 수강한 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실이 타이틀로 출력된다. ]

--[ 교육생은 한개의 과정만을 등록해서 수강한다. ]

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';

--로그인할 경우 자동으로 로그인한 교육생의 '교육생번호'를 통해 해당하는 정보들을 보여준다.

--교육생은 '겹치지 않는' 한 개의 과정만을 등록해서 수강한다.

select

s.student\_seq,

count(\*)

from tblstudent s

inner join tblregister r

on s.student\_seq = r.student\_seq

inner join tblopencourse oc

on oc.open\_course\_seq = r.open\_course\_seq

group by s.student\_seq

order by s.student\_seq;

--교육생은 수료 후에도 다른 과정을 수강할 수 있다. (교육생 별 수강 횟수 출력)

--[ 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';

--로그인할 경우 자동으로 로그인한 교육생의 '교육생번호'를 통해 해당하는 정보들을 보여준다.

--[ 한 개의 과정 내에는 여러 개의 과목을 수강한다 ]

select

os.open\_course\_seq as "개설과정 번호",

c.name as "개설과정 이름",

os.open\_subject\_seq as "개설과목 번호",

s.name as "개설과목 이름"

from tblopensubject os

inner join tblsubject s

on s.subject\_seq = os.subject\_seq

inner join tblopencourse oc

on os.open\_course\_seq = oc.open\_course\_seq

inner join tblcourse c

on oc.course\_seq = c.course\_seq;

--[ 과정기간이 끝나지 않은 교육생 또는 중도 탈락 처리된 교육생인 경우 일부 과목만 수강한다. ]

select \* from tblregister

where completion\_status = 'N' or dropout\_date is not null;

--1) 과정 기간이 끝나지 않은 (수료하지 않은) 또는 중도 탈락 처리된 교육생의 목록 확인

select

student\_seq

from tblregister

where (completion\_status = 'N'

or dropout\_date is not null)

and open\_course\_seq = 'OL001';

--2) 그 교육생들 중 현재 강의 중인 과정번호가 OL001인 개설과정의 학생 한명 (의 학생번호) : S0106

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 tblregister

where (completion\_status = 'N' or dropout\_date is not null) and open\_course\_seq = 'OL001')

and r.dropout\_date between os.subject\_start\_date and os.subject\_end\_date

order by os.subject\_start\_date, a.attendance\_date;

--3) 결과셋 : 과정 기간이 끝나지 않은 (수료하지 않은) 또는 중도 탈락 처리된 교육생들 중 개설과정번호가 OL001인 과정을 들은 학생의

-- 중도탈락 이전 날짜별 근태상황과 해당 과정의 일부 개설과목의 과목 시작일, 종료일을 보여준다

교육\_출결 관리 및 출결 조회

--[ 매일의 근태를(출근 1회, 퇴근1회)를 기록할 수 있다. ]

insert into tblDayAttendance (day\_attendance\_date, time, student\_seq, open\_course\_seq, is\_attendance) values (

current\_date,

to\_date('09:05:55', 'hh24:mi:ss'),

'S0038',

'OL001',

'출근'

);

--현재 날짜를 자동입력 하고싶을 경우 current\_date를 인서트한다.

--근태 시간은 교육생이 직접 입력할 수 있도록 한다.

insert into tblDayAttendance (day\_attendance\_date, time, student\_seq, open\_course\_seq, is\_attendance) values (

to\_date('2021-12-02', 'YYYY-MM-DD'),

to\_date('18:03:33', 'hh24:mi:ss'),

'S0038',

'OL001',

'퇴근'

);

--날짜를 지정하여 입력하고싶을 경우.

select \* from tbldayattendance where student\_seq = 'S0038'; --잘 입력됐나 확인

-- + 관리자 요구사항에 추가해야할 필요 있어 보이는 것 : 근태 추가하기 => trigger 이용

-- 관리자 혹은 교육생이 근태를 추가 입력해야 함

select

r.open\_course\_seq

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 = 'S0038'

and sysdate between oc.course\_start\_date and oc.course\_end\_date;

--1) 교육생이 수강 중인 개설과정 번호

insert into tblattendance (attendance\_date, student\_seq, open\_course\_seq, attendance\_type\_seq) values (

current\_date,

'S0038',

(select r.open\_course\_seq

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 = 'S0038'

and sysdate between oc.course\_start\_date and oc.course\_end\_date),

'TA02'

);

--날짜를 지정하여 입력하고싶을 경우에는 to\_date('2021-12-02', 'YYYY-MM-DD')

select \* from tblattendance where student\_seq = 'S0038'; --잘 입력됐나 확인

--[ 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",

at.attendance\_type as "근태 상황",

to\_char(da.time, 'hh24:mi:ss') as "시간",

da.is\_attendance as "출/퇴근",

da.open\_course\_seq as "개설과정 번호"

from tblopensubject os

inner join tblregister r

on os.open\_course\_seq = r.open\_course\_seq

inner join tblstudent s

on r.student\_seq = s.student\_seq

inner join tblattendance a

on os.open\_course\_seq = a.open\_course\_seq and s.student\_seq = a.student\_seq

inner join tblattendancetype at

on a.attendance\_type\_seq = at.attendance\_type\_seq

inner join tbldayattendance da

on r.student\_seq = da.student\_seq and r.open\_course\_seq = da.open\_course\_seq

where s.student\_seq='S0038'

order by a.attendance\_date;

--교육생 로그인 시 자동으로 where 조건에 학생 번호가 붙어 해당 교육생의 전체 출결이 조회된다.

select

distinct

s.student\_seq as "교육생 번호",

s.name as "교육생 이름",

a.attendance\_date as "날짜",

at.attendance\_type as "근태 상황",

to\_char(da.time, 'hh24:mi:ss') as "시간",

da.is\_attendance as "출/퇴근",

da.open\_course\_seq as "개설과정 번호"

from tblopensubject os

inner join tblregister r

on os.open\_course\_seq = r.open\_course\_seq

inner join tblstudent s

on r.student\_seq = s.student\_seq

inner join tblattendance a

on os.open\_course\_seq = a.open\_course\_seq and s.student\_seq = a.student\_seq

inner join tblattendancetype at

on a.attendance\_type\_seq = at.attendance\_type\_seq

inner join tbldayattendance da

on r.student\_seq = da.student\_seq and r.open\_course\_seq = da.open\_course\_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;

--원하는 기간별로 조회가 가능하다

and to\_char(a.attendance\_date, 'mm') = '09'

--원하는 '월'의 출결 조회가 가능하다

select

distinct

s.student\_seq as "교육생 번호",

s.name as "교육생 이름",

a.attendance\_date as "날짜",

at.attendance\_type as "근태 상황",

to\_char(da.time, 'hh24:mi:ss') as "시간",

da.is\_attendance as "출/퇴근",

da.open\_course\_seq as "개설과정 번호"

from tblopensubject os

inner join tblregister r

on os.open\_course\_seq = r.open\_course\_seq

inner join tblstudent s

on r.student\_seq = s.student\_seq

inner join tblattendance a

on os.open\_course\_seq = a.open\_course\_seq and s.student\_seq = a.student\_seq

inner join tblattendancetype at

on a.attendance\_type\_seq = at.attendance\_type\_seq

inner join tbldayattendance da

on r.student\_seq = da.student\_seq and r.open\_course\_seq = da.open\_course\_seq

where s.student\_seq='S0381'

and to\_char(a.attendance\_date, 'YYYY-MM-DD') = '2021-10-10'

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')

--오늘자의 출퇴근 현황을 조회할 수 있다

--\*\*\*\*\*\*\*\*\*\*\*이게 최종 SELECT

SELECT DISTINCT to\_char(a.attendance\_date, 'yyyy-mm-dd') AS "날짜",

d.name AS "교육생 이름",

b.attendance\_type AS "근태 상황",

to\_char(c.time, 'hh24:mi:ss') AS "시간",

c.is\_attendance AS "출퇴근"

FROM tblAttendance a

INNER JOIN tblAttendanceType b ON a.attendance\_type\_seq=b.attendance\_type\_seq

INNER JOIN tblDayAttendance c ON a.student\_seq=c.student\_seq AND a.open\_course\_seq=c.open\_course\_seq AND a.attendance\_date=c.day\_attendance\_date

INNER JOIN tblStudent d ON d.student\_seq=a.student\_seq

WHERE d.student\_seq='S0038'

ORDER BY a.attendance\_date desc;

select

distinct

s.student\_seq as "교육생 번호",

s.name as "교육생 이름",

to\_char(a.attendance\_date, 'yyyy-mm-dd') 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;

# 추가\_기념품

--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

-- 과정별 기념품 목록

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; --교육 과정 별지급한 기념품을 확인

--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;

# 

# 추가\_시험성적통계

--시험점수

--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;

**--교육생 점수입력**

UPDATE tblRegister SET evalution\_score =

(select round(avg(SUBJECT\_SCORE),1)

from tblScoreInfo

WHERE STUDENT\_SEQ = 'S0354'

and OPEN\_COURSE\_SEQ = 'OL007')

where tblRegister.STUDENT\_SEQ = 'S0354' and tblRegister.OPEN\_COURSE\_SEQ = 'OL007';

# 추가\_교육생지원제도

--지원여부수정

DELETE FROM TBLSUPPORTSYSTEM

WHERE STUDENT\_SEQ = 'S0496' AND open\_course\_seq = 'OL015'; --지원제도삭제

INSERT INTO TBLSUPPORTSYSTEM (STUDENT\_SEQ, OPEN\_COURSE\_SEQ, TRAINING\_INSENTIVE, EMPLOYEE\_SUPPORT)

VALUES ('S0496', 'OL015', 'Y', 'Y'); --지원제도 입력

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';

# 

# 추가\_취업연계

--관리자 - 취업연계

-- 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'));

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

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')); --교육생자격증입력

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'); --기업등록

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 구인의뢰 등록

INSERT INTO TBLRECRUITER (RECRUITER\_SEQ, NAME, POSITION, TEL, EMAIL)

VALUES ('RE0061', '김현규', '대리', '010-3389-2084', 'srhkras@gmail.com'); --기업의 채용담당자 등록

UPDATE TBLRECRUITER SET POSITION = '과장'

WHERE RECRUITER\_SEQ = 'RE0061'; --기업의 채용담당자 수정

--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');

# 추가\_취업자통계

--취업자 통계

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'; -- 취업연계된 구직자

select count(\*) as 취업자수

from TBLJOBSEEKER

WHERE employment\_state = 'Y'; -- 취업자수

--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;

procAttendance\_T 교사

procAttendance\_ST 학생

procAttendance\_M 관리자