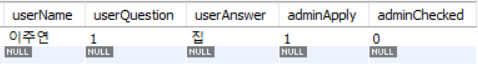
1. 회원가입 시

* 회원정보 입력 sql

INSERT INTO user VALUES (회원ID,비밀번호,이메일,학년,이수학기,학과,이름,질문,답,관리자권한 지원 여부, 관리자권한);





1. 로그인 시

* 회원ID에 따른 password를 불러오는 sql

SELECT userPassword FROM user WHERE userID=?



1. 비밀번호 분실 시

* 회원정보(회원ID, 질문, 답)가 일치할 시 회원ID를 불러오는 sql

SELECT userID FROM user WHERE userID=’201731022’ and userQuestion=’1’ and userAnswer=’집’



* 비밀번호 수정 sql

UPDATE user SET userPassword=’1111’ WHERE userID=’201731022’



1. 개인정보 수정 시

* 회원ID에 따른 회원정보 불러오는 sql

SELECT \* FROM user WHERE userID=201731022



* 회원 정보 수정 sql

UPDATE user SET userPassword=’1234’, userEmail=’wndus8834@naver’, userDepart=’소프트웨어공학과’, userName=’원준식’, userQuestion=’0’, userAnswer=’0’ WHERE userID=’201731022’



userID를 제외한 나머지 정보들이 바뀐 것을 확인할 수 있다.

구현 시에는 회원 정보 수정 화면에 이미 기존에 들어 있던 정보를 넣어준 후, 수정하지 않은 부분은 원래 들어 있던 정보 그대로 UPDATE 해준다.

1. 게시판

* 게시판 목록 조회 sql

SELECT \* FROM board WHERE boardGroup > (SELECT MAX(boardGroup) FROM board) - ? AND boardGroup <= (SELECT MAX(boardGroup) FROM board) - ? ORDER BY boardGroup DESC,boardSequence ASC



* 게시글 삽입

"INSERT INTO board SELECT ?, IFNULL((SELECT MAX(boardID) + 1 FROM board), 1), ?, ?, now(), 0, ?, ?, IFNULL((SELECT MAX(boardGroup) + 1 FROM board), 0), 0, 0, 1"

* 특정 게시글 조회

"SELECT \* FROM board WHERE boardID = ?"

* 게시글 조회수 상승

UPDATE board SET boardHit = boardHIt + 1 WHERE boardID =?

* 파일 다운로드를 위한 파일명 select

SELECT boardFile FROM board WHERE boardID =?

* 업로드한 파일의 실제 물리적 경로 select

SELECT boardRealFile FROM board WHERE boardID =?

* 게시글 수정

UPDATE board SET boardTitle =?, boardContent=?, boardFile=?, boardRealFile=? WHERE boardID=?

* 게시글 삭제

UPDATE board SET boardAvailable = 0 WHERE boardID =?

* 게시글 답변 작성

INSERT INTO board SELECT ?, IFNULL((SELECT MAX(boardID) + 1 FROM board), 1), ?, ?, now(), 0, ?, ?, ?, ?, ?, 1

* 게시글에 답변 달릴 시 답변글의 들여쓰기 구분을 위한 쿼리

UPDATE board SET boardSequence = boardSequence + 1 WHERE boardGroup=? AND boardSequence > ?

- 게시글 목록 다음 페이지가 존재하는지 조회

SELECT \* FROM board WHERE boardGroup >= ?

* 게시글 페이징을 위한 페이지 개수

SELECT COUNT(boardGroup) FROM board WHERE boardGroup > ?

1. 수강

* 수강과목 조회 ( 페이징, 검색, 과목 테이블에서 과목명, 분류 가져오기)

SELECT R.subjYear, R.subjSemester , R.subjCode, R.subjName, R.subjMain,R.score, R.userID, S.subjScore, S.bigDepartName, S.userDepart, S.coreCapaName, S.detailAreaName FROM class R INNER JOIN subj2 S ON R.subjCode = S.subjCode AND R.subjYear=S.subjYear AND R.subjSemester=S.subjSemester WHERE R.subjYear LIKE ? AND R.subjSemester LIKE ? AND R.subjMain LIKE ? AND CONCAT(R.subjCode) LIKE ? ORDER BY R.subjYear ASC LIMIT " + pageNumber\*5+ ", " +5

* 사용자에 따른 수강 리스트 조회

SELECT R.subjYear, R.subjSemester , R.subjCode, R.subjName, R.subjMain,R.score, R.userID, S.subjScore, S.bigDepartName, S.userDepart, S.coreCapaName, S.detailAreaName FROM class R INNER JOIN subj2 S ON R.subjCode = S.subjCode AND R.subjYear = S.subjYear AND R.subjSemester = S.subjSemester WHERE userID=? ORDER BY R.subjYear ASC

* 자신이 수강한 과목 삭제

DELETE FROM class WHERE subjYear=? AND subjSemester=? AND subjCode=? AND userID=?

* 자신이 수강한 과목 등록

INSERT INTO class (subjYear, subjSemester, subjCode, userID, subjName, subjMain, score, grade, changeMain) VALUES (?,?,?,?,?,?,?,?,false)

* 수강한 과목 수정

UPDATE class SET subjMain=replace(subjMain, '교선', '전선'), changeMain=true WHERE subjCode=? AND userID=?

* 이수 구분에 따른 자신이 수강한 과목 조회 (전선, 교선, 등등)

SELECT C.subjCode, C.subjName, S.userDepart FROM class C INNER JOIN subj2 S ON C.subjCode = S.subjCode AND C.subjYear = S.subjYear AND C.subjSemester = S.subjSemester AND C.grade=S.subjScore WHERE C.userID = ? AND C.subjMain = '전선'

* 특정 과목을 자신이 수강했는지 조회

SELECT \* FROM class WHERE subjCode = ? AND userID = ?

* 대체 과목(재수강) 트랜잭션

UPDATE class SET score = ? WHERE subjCode=? AND userID = ?

DELETE FROM class WHERE subjCode=? AND userID = ?

* 대체 과목(초수강) 트랜잭션

INSERT INTO class (subjYear, subjSemester, subjCode, userID, subjName, subjMain, score

VALUES (?, ?, ?, ?, ?, ?, ?)

DELETE FROM class WHERE subjCode=? AND userID = ?

* 특정 대체될 과목에 대한 대체과목 리스트 조회

SELECT subjYear, subjSemester, score FROM class WHERE userID =? AND subjCode=?

* 자신이 수강한 과목 && 폐지된 과목 && C+ 이하인 과목을 조회하기 위한 쿼리문

SELECT R.subjCode, I.beInsteadSubjName, R.score FROM joined.class R INNER JOIN joined.insteadtable I ON R.subjCode = I.beInsteadSubjCode WHERE userID=? ORDER BY R.subjYear ASC

1. 졸업요건 확인을 위한 회원 점수 저장 테이블

* 회원 점수 insert 쿼리

INSERT INTO userscore

(userID,refinementScore,culturalReqScore,majorSelectScore,majorReqScore,capaReinforceScore,majorFind)

VALUES (?,?,?,?,?,?,?)

* 회원 점수 UPDATE 쿼리

UPDATE userscore SET

refinementScore=?,culturalReqScore=?,majorSelectScore=?,

majorReqScore=?,capaReinforceScore=?,majorFind=?

WHERE userID=?

* 회원 점수 조회

SELECT \* FROM userscore WHERE userID = ?

1. 과목

* 폐지된 과목 & 과목 JOIN 쿼리 (대체과목 관리)

SELECT bigDepartName, userDepart , subjCode, subjName, subjInsteadCode, subjInsteadName, subjYear, subjSemester FROM subj WHERE userDepart LIKE ? AND CONCAT(subjCode) LIKE ? AND subjInsteadCode IS NOT NULL AND subjInsteadName IS NOT NULL ORDER BY userDepart ASC LIMIT " + pageNumber\*5+ ", " +5

* 전체 과목 리스트 조회

SELECT bigDepartName, userDepart , subjCode, subjName, subjYear, subjSemester FROM subj2 ORDER BY userDepart ASC

* 과목 수정 (과목 관리)

UPDATE subj SET subjName=?, subjInsteadCode=?, subjInsteadName=? WHERE subjCode=?

* 특별 과정 과목 조회 (특별과정 관리)

SELECT subjYear, subjSemester, subjCode, coreCapaName , detailAreaName, subjName, subjScore FROM subj2 WHERE coreCapaName LIKE ? AND CONCAT(subjName) LIKE ? AND coreCapaName IS NOT NULL AND detailAreaName IS NOT NULL ORDER BY userDepart ASC LIMIT " + pageNumber\*5+ ", " +5

* 특별 과정 과목 수정 (특별과정 관리)

UPDATE subj2 SET coreCapaName=?, detailAreaName=?, subjName=? WHERE subjYear='2018' AND subjCode=?

* 특별 과정 과목 삭제 (특별과정 관리)

UPDATE subj2 SET coreCapaName=?, detailAreaName=?, subjName=? WHERE subjYear='2018' AND subjCode=?

1. 학생 상담(메모) 관리

* 학생 상담내역이 존재하는지 SELECT

SELECT memoContent, memoTime FROM userStudentMemo WHERE userID=? AND adminID=?

* 학생 상담내용 삽입

INSERT INTO userStudentMemo VALUES (?,?,?,?)

* 학생 상담내용 수정

UPDATE userStudentMemo SET memoContent=?, memoTime=? WHERE userID = ? AND adminID =?

1. 대체과목 관리

* 대체과목 리스트 조회

SELECT bigDepartName, beInsteadDepart, beInsteadSubjCode, beInsteadSubjName, InsteadSubjCode, InsteadSubjName FROM insteadtable WHERE beInsteadDepart LIKE ? AND CONCAT(beInsteadSubjCode) LIKE ? AND InsteadSubjCode IS NOT NULL AND InsteadSubjName IS NOT NULL ORDER BY beInsteadDepart ASC LIMIT " + pageNumber\*5+ ", " +5

* 대체과목 수정

UPDATE insteadtable SET beInsteadSubjName=?, insteadSubjCode=?, InsteadSubjName=? WHERE beInsteadSubjCode=?

* 대체과목 삭제

DELETE FROM insteadtable WHERE beInsteadSubjCode=?

* 대체과목 등록
* INSERT INTO insteadtable (bigDepartName, beInsteadDepart, beInsteadSubjCode, beInsteadSubjName, subjMain, subjScore, insteadSubjCode, insteadSubjName
* VALUES (?, ?, ?, ?, ?, ?, ?, ?)
* 사용자 && 사용자의 학과 && 이수구분에 따른 대체과목 리스트 조회(전선, 교양 ….)

SELECT C.subjCode, C.subjName, S.userDepart FROM class C INNER JOIN subj S ON C.subjCode = S.subjCode WHERE C.userID = ? AND C.subjMain = '전선'

1. 필수과목 관리

* 필수과목 등록

INSERT INTO departr VALUES (?,?,?,?,?,?)

* 학과 && 과목코드를 통한 필수과목 조회

SELECT R.userDepart, S.subjName , R.userAdmission, R.subjCode, R.subjGrade, R.subjSemester, R.subjDivision FROM departr R INNER JOIN subj2 S ON R.subjCode = S.subjCode AND R.userAdmission=S.subjYear AND R.subjSemester=S.subjSemester WHERE R.userDepart LIKE ? AND CONCAT(R.subjCode) LIKE ? ORDER BY R.userDepart ASC LIMIT " + pageNumber\*5+ ", " +5

* 특정 필수과목에 대한 정보 조회

SELECT \* FROM departr WHERE subjCode=?

* 필수 과목 수정

UPDATE departr SET userDepart=?, subjGrade=?, subjSemester=?, subjDivision=? WHERE subjCode=? AND userAdmission=?

* 필수 과목 삭제

DELETE FROM departr WHERE subjCode=? AND userAdmission=?

* 사용자가 수강하지 못한 자신의 학과 필수과목 조회

SELECT DISTINCT I.beInsteadSubjCode, I.beInsteadSubjName FROM joined.insteadtable I WHERE I.beInsteadSubjCode IN (SELECT D.subjCode FROM joined.departr D LEFT JOIN joined.class C ON D.subjCode = C.subjCode AND C.userID = ? WHERE D.userAdmission = ? AND D.userDepart =? AND C.subjName Is null)

* 사용자에 따른 대체될 과목의 이수구분, 과목명 조회

SELECT C.subjMain, S.subjName FROM class C INNER JOIN subj2 S ON C.subjCode = S.subjCode AND C.subjYear=S.subjYear AND C.subjSemester=S.subjSemester WHERE C.subjCode=? AND userID=?

1. 관리자 점수 관리 (졸업요건 관리)

* 관리자가 졸업요건 점수 등록

INSERT INTO adminscore VALUES (?,?,?,?,?,?,?,?,?,?)

* 졸업요건 점수 조회

SELECT \* FROM adminscore ORDER BY userDepart ASC

* 특별과정 && 특정 과 && 학기 에 따른 졸업요건 점수 조회

SELECT \* FROM adminscore WHERE userProcess=? AND userDepart=? AND userAdmission=?

* 졸업요건 점수 수정

UPDATE adminscore SET totalCulture=?, totalCultureNecessary=?, totalMajor=?, totalMajorNecessary=?, totalGraduate=?, totalAbility=?, totalFindMajor=? WHERE userDepart=? AND userAdmission=? AND userProcess=?

* 졸업요건 점수 삭제 (관리자)

DELETE FROM adminscore WHERE userDepart=? AND userAdmission=? AND userProcess=?