*-총 화면 수 : 9개, -총 sql 수:*

**Exception**

**– 시스템 오류로 반환이 안된경우**

[alert] 시스템오류로 접속이 불가합니다 다시 시도해주세요

### [로그인]



>관리자가 비밀번호를 사번으로 설정해두고

>직원은 로그인(마이페이지)에서 비밀번호만 수정가능

***--SQL쿼리(1)***

**--사번, 비밀번호 입력 후 로그인 버튼을 눌렀을 때**

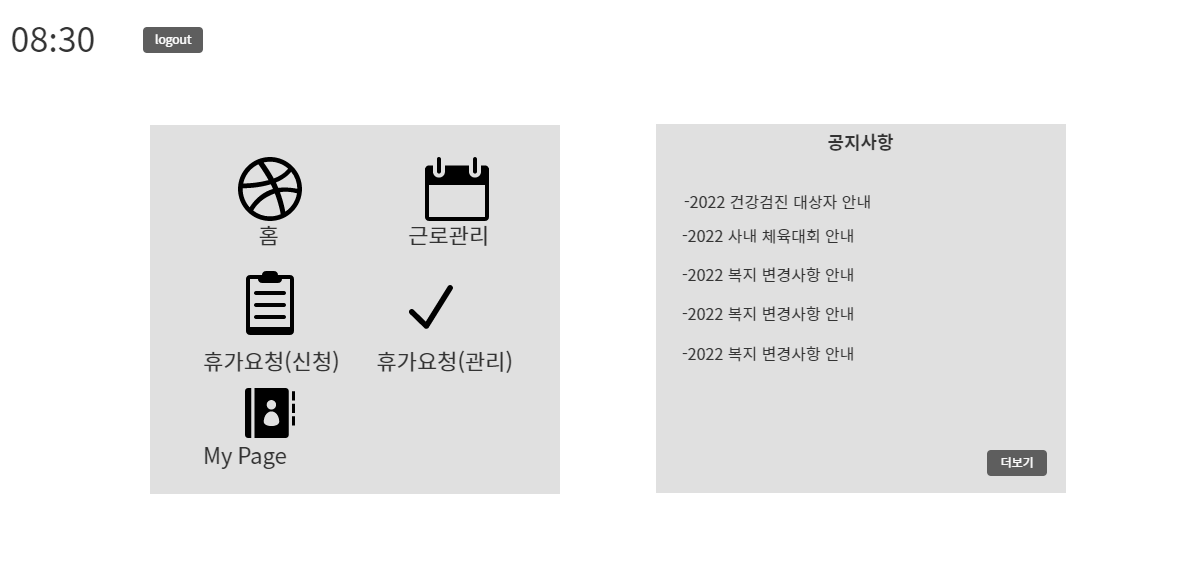
select \* from employee where employee\_id= [사번 입력값] and employee\_password= [비밀번호 입력값];

**Exception**

**–해당사원이 없거나 사번,비밀번호를 잘못입력한 경우**

[alert] 사번 혹은 비밀번호가 일치하지 않습니다

[메인페이지]



>메뉴가 아이콘과 함께 보이며 클릭시 해당 url로 넘어감

>공지사항에 최신글이 5줄 출력

***--SQL쿼리(1)***

**--공지사항란에 공지사항 최신 5개의 글 출력**

//select notice\_title, notice\_id from notice where rownum between <startRow> AND <endRow>; // 삭제예정

select \* from notice n

join employee e on (e.employee\_id = n.employee\_id)

where rownum between <startRow> AND <endRow>;

// Front에서 필요한 값만 받아오기

(메인공지사항: notice\_id, notice\_title 값만 필요)

### [공지사항 - 리스트]



>메인페이지에서 공지사항 더보기버튼 클릭시 나오는 화면

>제목으로 검색해주세요 클릭시 동작.’/

>한 페이지에 10개씩 출력

>제목 클릭시 상세페이지로 이동

**– 공지사항에 없는 검색어를 입력한 경우**

[결과가 뜨는 화면에] 검색어를 포함한 제목의 공지사항이 없습니다

**--SQL쿼리(추가)**

**--공지사항페이지에 최신 10개의 글 출력**

select \* from notice n

join employee e on (e.employee\_id = n.employee\_id)

where rownum between <startRow> AND <endRow>;

***--SQL쿼리(2)***

**--공지사항의 번호, 제목, 작성자, 작성날짜 최신순으로 반환**

SELECT \*

FROM (

SELECT rownum as rn, a.\*

FROM(

SELECT n.notice\_id, n.notice\_title, e.employee\_id, n.notice\_time

FROM notice n JOIN employee e ON (n.employee\_id=e.employee\_id)

ORDER BY notice\_time DESC

) a

)

WHERE rn BETWEEN 1 AND 10;

SELECT \*

FROM (

SELECT rownum as rn, a.\*

FROM(

SELECT n.notice\_id, n.notice\_title, e.employee\_id, n.notice\_time

FROM notice n JOIN employee e ON (n.employee\_id=e.employee\_id)

ORDER BY notice\_time DESC

) a

)

WHERE rn BETWEEN <startRow>AND <endRow>;

**--검색된 제목([검색된 값])과 같은 공지사항 가져오기 //페이징처리!!!!1**

SELECT \* FROM(

SELECT rownum as rn, n.notice\_id, n.notice\_title,n.notice\_time,e.employee\_name\_kr

FROM notice n join employee e on(n.employee\_id=e.employee\_id)

WHERE notice\_title like '%[검색된 값]%'

ORDER BY notice\_id DESC

)

WHERE rn BETWEEN <출력할행START> AND <출력할행 END>;

—-------------------------------------------------------------------------------------

SELECT \* FROM(

SELECT rownum as rn, n.notice\_id, n.notice\_title,n.notice\_time,n.notice\_cnt,e.employee\_name\_kr, e.employee\_id

FROM notice n join employee e on(n.employee\_id=e.employee\_id)\r\n"

+ " WHERE notice\_title LIKE '%' || ? || '%' \r\n"

+ " ORDER BY notice\_id DESC\r\n"

+ " )\r\n"

+ " WHERE rn BETWEEN ? AND ?

—----------------------------------------------------------------------------------------

--흠..이곤가

SELECT \*

FROM(

SELECT rownum as rn, a\*

FROM(

SELECT n.notice\_id, n.notice\_title, n.notice\_time

e.employee\_name\_kr

FROM notice n JOIN employee e ON (n.employee\_id=e.employee\_id)

WHERE notice\_title LIKE '%[검색된 값]%'

ORDER BY notice\_id DESC

) a

)

WHERE rn BETWEEN <출력할행START> AND <출력할행 END>;

### [공지사항-상세페이지]



***--SQL쿼리(1)***

--클릭한 공지사항의 상세정보(제목, 작성자, 작성시간, 내용) 가져오기

SELECT notice\_title, employee\_name\_kr, notice\_time, notice\_cnt

FROM notice n JOIN employee e ON (n.employee\_id = e.employee\_id)

WHERE notice\_id = [클릭한 공지번호];

SELECT \*

FROM notice n JOIN employee e ON (n.employee\_id = e.employee\_id)

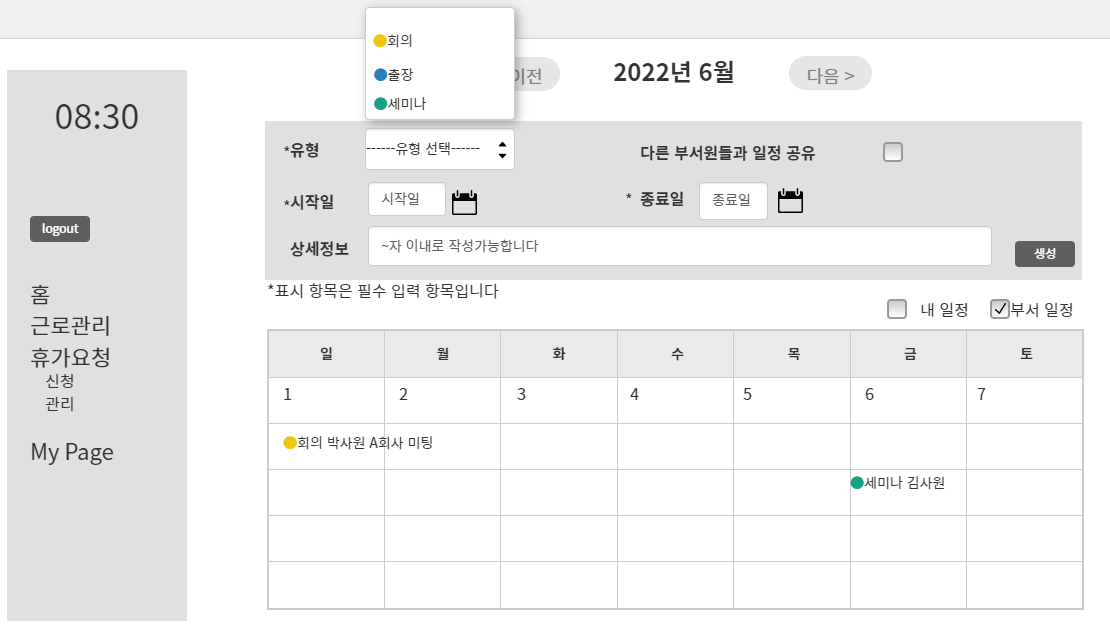
WHERE notice\_id = ? ;

SELECT notice\_id,notice\_cnt,notice\_title,notice\_time, notice\_cnt, employee

FROM notice n JOIN employee e ON (n.employee\_id = e.employee\_id)

WHERE notice\_id = ? ;

### [근로관리] CalendarRepository.java



>내 일정과 부서일정 둘 중 체크박스로 클릭 가능

>부서일정 클릭시 부서 모든 인원의 일정이 출력(같은부서번호로 불러오기)

>시작일/ 종료일 미니달력 선택

>달력 API

>상세정보 라이브러리

>상세정보에 제한된 글자수 이상 입력하면 입력이 안되도록 HTML에서 처리

>필수입력내용을 모두 입력하지 않으면 생성버튼이 생기지 않는다

>생성자 입력 ‘~자 이내로 작성가능합니다’

>공개 -> ‘다른 부서원들과 일정 공유’로 수정

>공개 체크 안된것이 default, 내일정 체크된것이 default

>월 표시된 곳에 연도도 같이 표시(위치는 만들때 정하자)

> 필수 입력 항목은 \*로 표시

> 일자 입력시 시작일 선택하면 종료일은 자동으로 같은 값이 들어감.

시작일보다 앞의 날짜는 선택할 수 없음

>revealcheck 0이 비공개, 1이 공개

***--SQL쿼리(3)***

--유형드롭다운하면 유형리스트 반환

SELECT cal\_type, cal\_type\_name FROM cal\_type;

메서드 insert(Calendar calendar)

-- 생성버튼 눌렀을때 캘린더에 입력

INSERT INTO calendar(cal\_no, employee\_id, cal\_type, cal\_revealed, cal\_cnt, cal\_start, cal\_end)

VALUES (seq\_cal\_no.NEXTVAL ,<로그인한 사용자 사번>, <입력한 유형번호>, <입력한 공개여부>, <입력한 상세정보>, <입력한 시작일>, <입력한 종료일>);

-- 내 일정 체크했을 때 출력 (revealed check 상관x) - 0712완료

-- 일정 출발일이나 종료일이 이번달과 같은 달일때만 출력

SELECT c.employee\_id, c.cal\_type, t.cal\_type\_name, c.cal\_cnt, c.cal\_start, c.cal\_end

FROM calendar c JOIN employee e ON(c.employee\_id = e.employee\_id)

JOIN cal\_type t ON(c.cal\_type = t.cal\_type)

WHERE c.employee\_id = <로그인된 사번>

AND(TO\_CHAR(<화면출력날짜>, 'MM') = TO\_CHAR(cal\_start, 'MM')

OR TO\_CHAR(<화면출력날짜>, 'MM') = TO\_CHAR(cal\_end, 'MM'));

--내일정과 부서일정 눌렀을때 출력(내 일정 + 부서(revealcheck된) 일정) -0712완료

SELECT c.employee\_id, e.employee\_name\_kr, c.cal\_revealed, c.cal\_type, t.cal\_type\_name, c.cal\_cnt, c.cal\_start, c.cal\_end

FROM calendar c JOIN employee e ON(c.employee\_id = e.employee\_id)

JOIN cal\_type t ON(c.cal\_type = t.cal\_type)

WHERE c.cal\_revealed=1 AND e.department\_id = (SELECT e.department\_id FROM employee e WHERE e.employee\_id = <로그인된 사번>)

OR (c.cal\_revealed=0 AND c.employee\_id = <로그인된 사번>)

AND(TO\_CHAR(<화면출력날짜>, 'MM') = TO\_CHAR(cal\_start, 'MM')

OR TO\_CHAR(<화면출력날짜>, 'MM') = TO\_CHAR(cal\_end, 'MM'));

***--String타입으로 화면출력날짜를 받을 경우***

SELECT c.employee\_id, e.employee\_name\_kr, c.cal\_revealed, c.cal\_type, t.cal\_type\_name, c.cal\_cnt, c.cal\_start, c.cal\_end

FROM calendar c JOIN employee e ON(c.employee\_id = e.employee\_id)

JOIN cal\_type t ON(c.cal\_type = t.cal\_type)

WHERE c.cal\_revealed=1 AND e.department\_id = (SELECT e.department\_id FROM employee e WHERE e.employee\_id = <로그인된 사번>)

OR (c.cal\_revealed=0 AND c.employee\_id =<로그인된 사번>)

AND(<화면출력날짜> = TO\_CHAR(cal\_start, 'MM')

OR <화면출력날짜> = TO\_CHAR(cal\_end, 'MM'))

-- 내일정 체크안하고 부서일정 체크했을때 출력 -완료

M employee WHERE employee\_id = <로그인된 사번>)

AND e.employee\_id != <로그인된 사번>

AND(TO\_CHAR(<화면출력날짜>, 'MM') = TO\_CHAR(cal\_start, 'MM')

OR TO\_CHAR(<화면출력날짜>, 'MM') = TO\_CHAR(cal\_end, 'MM'));

[휴가요청- 신청]

>현황을 자세하게 볼 수 있도록

>현황란은 입력x, 시스템에서 자동으로 보여주는 것

>유형, 시작일을 선택해야 기안버튼이 나옴

>결재자 설정(1차, 2차, 3차)

> 필수 입력 구간은 \*로 표시(유형, 일시, )

> 일자 입력시 시작일 선택하면 종료일은 자동으로 같은 값이 들어감.

시작일보다 앞의 날짜는 선택할 수 없음

***--SQL쿼리(5)***

– 유형 가져오기

SELECT holiday\_type\_name, holiday\_type

FROM holiday\_type;

--결재자 가져오기

SELECT e.\*

FROM approval\_level a JOIN employee e ON a.employee\_id = e.employee\_id;

--휴가 현황(총 휴가일/ 사용 휴가일 가져오기)

SELECT holiday\_total, holiday\_using

FROM holiday\_info WHERE employee\_id = <입력받은 로그인된 ID>;

**--기안하기 버튼 클릭**

**--휴가요청정보 DB에 저장(휴가신청, 휴가결재 두 테이블에 INSERT)**

**–INSERT 따로 따로 하면 됨**

--휴가신청INSERT

INSERT INTO holiday\_request

(holiday\_number, employee\_id, holiday\_type,

holiday\_start, holiday\_end, holiday\_desc,

holiday\_approval\_first,

holiday\_approval\_second,

holiday\_approval\_third,

holiday\_approval\_status)

VALUES(seq\_holiday\_no.NEXTVAL,

<로그인된 사번>, <선택된holiday\_type>,

<선택된holiday\_start>, <선택된holiday\_end>,

<입력된holiday\_desc>,<선택된 holiday\_approval\_first>,

<선택된holiday\_approval\_second>,

<선택된 holiday\_approval\_third>, 3);

[휴가요청 - 신청확인창]



[휴가요청-관리]



>요청취소 조건 : 1차결재자가 승인대기중일 때

>페이지당 10개

***--SQL쿼리(2)***

--휴가요청 리스트 출력

--강사님의 빛나는 코드

SELECT \*

FROM (

SELECT rownum as rn, a.\*

FROM(

SELECT hr.holiday\_number, hr.holiday\_type, hr.holiday\_desc, hr.holiday\_start, hr.holiday\_end,

ha.holiday\_approval\_status,

ht.holiday\_type\_name

FROM holiday\_request hr JOIN holiday\_approval ha ON (hr.holiday\_number=ha.holiday\_number)

JOIN HOLIDAY\_TYPE ht ON (hr.holiday\_type=ht.holiday\_type)

where EMPLOYEE\_ID = 220702

ORDER BY holiday\_number DESC

) a

)

WHERE rn BETWEEN <출력할행START> AND <출력할행 END>;

SELECT \*

FROM (

SELECT rownum as rn, a.\*

FROM(

SELECT hr.holiday\_number, hr.holiday\_type, hr.holiday\_desc, hr.holiday\_start, hr.holiday\_end,

hr.holiday\_request\_status,

ht.holiday\_type\_name

FROM holiday\_request hr JOIN HOLIDAY\_TYPE ht ON (hr.holiday\_type=ht.holiday\_type)

where EMPLOYEE\_ID = 220702

ORDER BY holiday\_number DESC

) a

)

WHERE rn BETWEEN <출력할행START> AND <출력할행 END>;

--요청취소 클릭시

UPDATE holiday\_request SET holiday\_request\_status = -2

WHERE holiday\_number = <요청취소할 휴가신청번호>;

--상태가 승인대기중에서 승인완료로 바뀌면 사원별휴가현황의 사용휴가수가 바뀜

UPDATE holiday\_using FROM holiday\_info

WHERE employee\_id=<로그인한 사원의 사번>;

[결재 상세보기]



>모달창(확인버튼 클릭시 이전화면으로 이동) 첨부파일여부 수정

***--SQL쿼리(4) –***

--한번에 가져오기

SELECT r.holiday\_start, r.holiday\_end, t.holiday\_type\_name,

(SELECT e.employee\_name\_kr

FROM approval\_level l

JOIN employee e ON(l.employee\_id = e.employee\_id)

JOIN holiday\_request r ON (l.employee\_id = r.holiday\_approval\_first)

WHERE (r.holiday\_approval\_first = e.employee\_id) and r.holiday\_number = ?),

(SELECT e.employee\_name\_kr

FROM approval\_level l

JOIN employee e ON(l.employee\_id = e.employee\_id)

JOIN holiday\_request r ON (l.employee\_id = r.holiday\_approval\_second)

WHERE (r.holiday\_approval\_second = e.employee\_id) and r.holiday\_number = ?),

(SELECT e.employee\_name\_kr

FROM approval\_level l

JOIN employee e ON(l.employee\_id = e.employee\_id)

JOIN holiday\_request r ON (l.employee\_id = r.holiday\_approval\_third)

WHERE (r.holiday\_approval\_third = e.employee\_id) and r.holiday\_number = ?),

r.holiday\_request\_status

FROM holiday\_request r

JOIN holiday\_type t ON(r.holiday\_type = t.holiday\_type)

JOIN employee e ON(r.employee\_id = e.employee\_id)

WHERE r.holiday\_number = ?;

--휴가신청이 승인완료되면 승인된휴가일수update

UPDATE holiday\_info SET holiday\_using = holiday\_usingr + 승인된 휴가신청 날짜???

WHERE employee\_id = <로그인된 사용자사번>

—----------------------------------------------------------------------------

[마이페이지]



> 자기정보확인용

> 비밀번호만 수정가능

***--SQL쿼리(2)***

--로그인된 사번의 정보(이름, 영문명, 생년월일, 성별, 주소, 사번, 비밀번호, 부서명, 핸드폰번호, 입사일, 이메일주소) 가져오기

--SELECT e.employee\_name\_kr, e.employee\_name\_eng, e.employee\_birthday, e.employee\_gender, e.employee\_address, e.employee\_id, e.employee\_password, d.department\_name, e.employee\_phonenumber, e.employee\_hiredate, e.employee\_email

FROM employee e JOIN department d ON e.department\_id=d.department\_id

WHERE e.employee\_id = <로그인된 사용자 사번>;

--respository 에 바꿨습니다

SELECT \*

FROM employee e JOIN department d ON e.department\_id=d.department\_id

WHERE e.employee\_id = 220702;

--수정버튼 클릭 시 입력된 값으로 비밀번호 변경됨

update employee set employee\_password = <비밀번호 입력값>

WHERE employee\_id = <로그인된 사용자 사번>;

**Exception**

**–비밀번호 변경시**

비밀번호는 최소 6자리 이상이어야합니다