# 

# **데이터베이스 프로그래밍**

# **󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲󰠲**

# **텍스트, 스크린샷, 폰트, 로고이(가) 표시된 사진 AI가 생성한 콘텐츠는 부정확할 수 있습니다.**

|  |  |
| --- | --- |
| **수강과목** | **: 데이터베이스 프로그래밍** |
| **담당교수** | **: 이승환 교수님** |
| **학과** | **: 컴퓨터공학과** |
| **제출일** | **: 2025.12.11** |
| **작성자** | **: 001분반 7조****텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.** |

# 

# 

# 

# 

# 

# 

# **목 차**

# **1.팀원 소개 및 사업 개요 ……………………………………………………………3P**

# **1.1 팀원소개**

# **1.2사업 개요**

* 사업명
* 사업배경

# 

# **2.요구사항 분석 ………………………………………………………………………. 4~16P**

**2.1 요구사항 정의서**

* 기능 요구사항
* 비기능 요구사항
* 데이터베이스 테이블 구조

# 

# 

# **3.화면구성 및 쿼리 …………………………………………………………………….17~67P**

# **학생페이지**

* **사서페이지**
* **공용페이지**

# 

# 

# 

# 

# 

# 

# 

# 

**1.1 팀원 소개**

|  |  |  |
| --- | --- | --- |
| 이름 | 학번 | 맡은 역할 |
| 조기강(팀장) | 20233849 | 화면(UI) 구현 , DML 작성 보조  전체적인 쿼리 및 문서 검토 + 문서 작업 |
| 서성준 | 20212925 | ddl 설계 DML 작성 |
| 정성민 | 20203091 | ppt 제작 및 발표, 발표 화면의 DML 피드백 |
| 김영국 | 20211985 | 요구사항 분석 및 초안 문서 작성 |
| 장창훈 | 20212871 | dml 과 화면 간의 불일치 검토 및 피드백 작성 |

**1.2 사업개요**

## 

## **1.2.1 사업명**

**동의대학교 중앙도서관 차세대 운영 시스템 구축 (30년 주기 개편 프로젝트)**

## **1.2.2 사업 배경**

동의대학교는 도서관 관리 시스템의 품질 향상과 소프트웨어융합과 **학생들의 실무 역량 강화**를 위해 30년 주기로 도서관 통합 프로그램을 전면 개편하는 “**현장 프로젝트**” 제도를 운영하고 있습니다. 2026년은 차세대 시스템 개편 주기의 시작으로, 우리팀은 소프트웨어융합부 컴퓨터공학과 대표로 프로젝트 참가권한을 부여받았으며. 학교는 프로젝트 수행을 인정받은 팀을 온누리 장학금 대상자로 선정할 계획이며, **팀원들은 졸업 전 실제 대형 프로젝트를 수행함으로써 실전 경험을 축적**하여 더 경쟁력있는  
신입으로 발굴 하고자 해당 사업이 진행되었습니다.

# **2. 요구사항 정의서**

# **2.1.1 기능 요구사항**

### **사용자(학생) 기능**

• 학생의 정보는 학과 시스템 에서 참고(학번,학과,연락처,성별)  
• 로그인 없이 도서 검색 가능하며 대출/예약/신청 기능은 로그인 후 이용  
• 개인 대시보드: 대출 가능 횟수, 현재 대출 중 도서, 연체 여부, 페널티 상태 조회,대출 기록들 조회  
• 대출 정지 상태인 경우 대출 및 예약 기능 제한

• 책을 검색하면 책이름 책 이미지,출판사,상세보기,출판년도 등 조회

• 책 상세 보기는 복권 대출 정보 + 예약 근황 +책위치

• 상호대차 ,신청 자료 구매 요청 가능

• 열람식 좌석예약기능(입실 시간은 하루2시간 제한)

### **도서 관리 기능**

• 도서 기본 정보 관리: 제목, 저자, 출판사, 연도, 장르, 등록 수량 입력 (장르는 장르 테이블 참조)  
• 도서 기본 정보는 사서 만 수정/삭제 가능  
• 개별 도서 상태(바코드 단위) 자동 생성: 등록 수량만큼 복권 생성  
• 도서 상태: 복건 번호 ,대출가능, 대출 중, 분실, 파손, 서고 배치위치  
• 개별 도서별 서고 위치(층, 구역) 관리  
• 도서 상태 변경은 사서 만 가능

### **대출 및 반납**

• 사용자는 대출 가능 횟수 내에서 대출 가능  
• 대출은 개별 도서(바코드) 기준으로 처리  
• 반납 시 상태 자동 변경: "대출가능"  
• 반납일 초과 시 자동 연체 처리  
• 연체 페널티:연체되면 연체권수 x 연체일수 x 2 로 페널티 기간을 적용, 파손은 30일   
사서가 권한으로 정지 할수 있음 90 일 이상 연체 → 블랙리스트 등재  
• 반납 전 연장 기능 제공(최대 1회)

### **패널티 관리**

• 연체된 대출 기록은 자동 패널티 대상  
• 패널티 테이블 기록: 패널티 유형(책 파손 ,연체, 블랙리스트, 도서 미반납 등), 시작일/종료일, 유효 여부  
• 대출 정지 상태 사용자는 대출/예약 불가 , 좌석 예악은 가능  
• 블랙리스트 사용자는 모든 도서관 기능 출입 까지 제한

### **예약 관리**

• 모든 개별 도서 (복권)가 대출 중일 때 예약 가능  
• 예약은 도서 복권 종류 단위로 신청  
• 예약 정보: 신청일, 예약 순번, 예약 상태('대기중', '수령가능', '취소됨', '수령완료)  
• 도서 반납 시 예약 1순위 사용자에게 자동 알림 및 우선 대출

### **신청자료(신규 희망 도서)**

• 학생은 도서명, 저자, 요청 사유(선택), 신청일로 신청 가능  
• 사서는 처리 상태 변경: ‘대기', '승인', '반려', '구매완료  
• 승인 시 실제 도서 목록에 추가 가능

### **사서 기능**

•사서 계정 로그인 후 내부 기능 접근  
•권한: 도서 등록/수정/삭제, 개별 도서 상태 관리, 서고 관리, 대출 승인/반납 처리  
•페널티 부여/해제, 신청자료 처리  
•사서 정보: 사서 ID, 비밀번호, 이름, 연락처,직책,자격증  
•사서는 무조건 자격증이 있어 야함 (정보란에 자격증 없으면 입력 불가)  
•어떤 책들이 대출 되었는지 통계 ,어떤 학생이 대출을 많이 했는지 출,입 로그 기록 , 통계 확인  
•상호대차,도서 구매 신청 을 받아서 처리   
•봉사활동 하는 학생 관리 하는 기능  
•사서 이상(admin)만 공지사항을 작성 가능

## **2.2 비기능 요구사항**

### **2.2.1 성능**

• 도서 상태 조회는 1초 이내 응답  
 • 대출/반납 처리 시 트랜잭션을 통한 데이터 무결성 보장

### **2.2.2 보안**

• 사서 비밀번호 해시 암호화(예: BCrypt)  
 • 모든 사용자 개인정보 암호화 저장  
 • 관리자 기능은 역할 기반 접근 제어(RBAC)로 권한 체크 필수

### **2.2.3 유지보수**

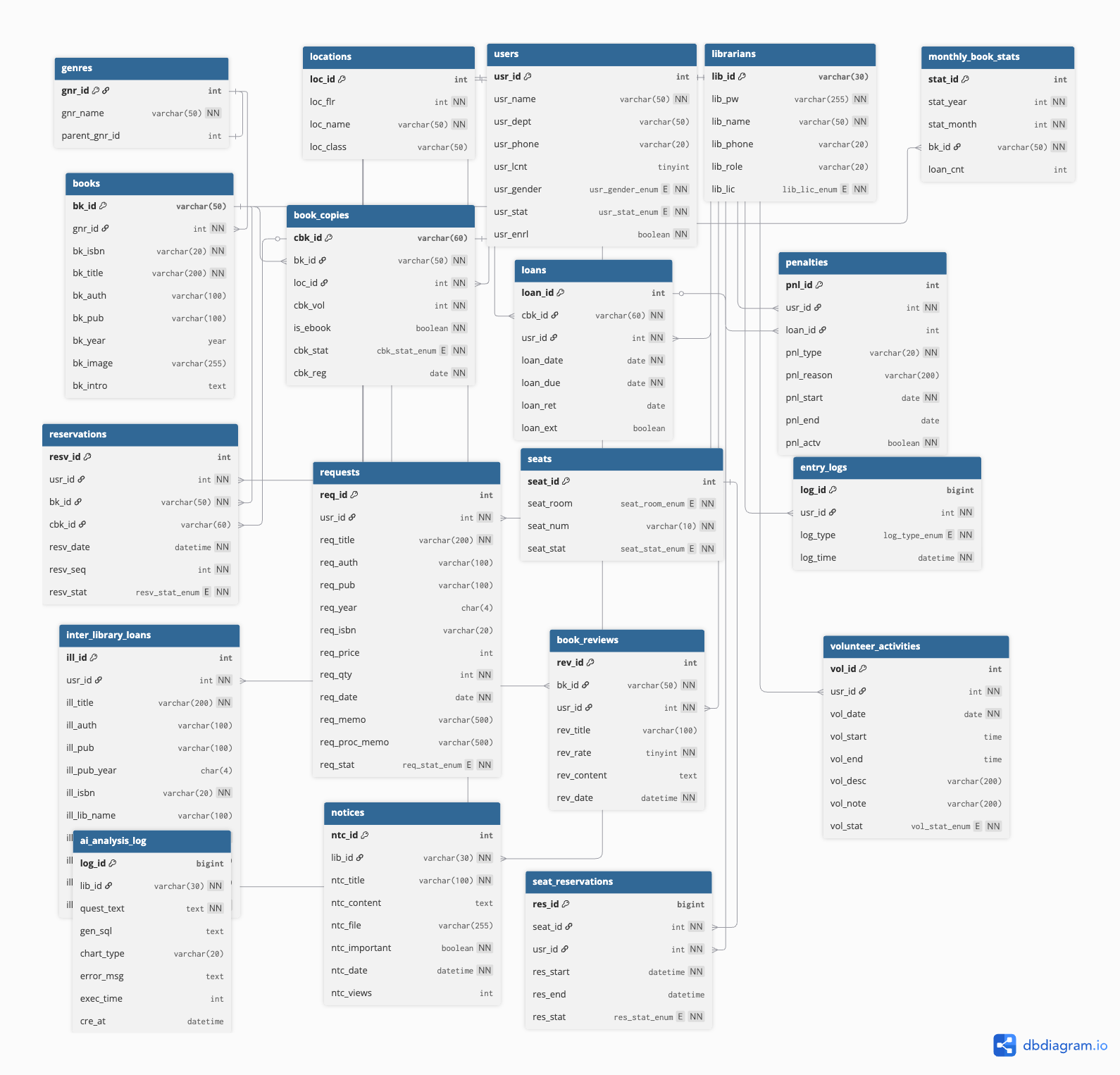
• MySQL 기반  
 • MySQL 8.0 이상 사용  
 • 30년 주기 차기 개편 시 이전 시스템과 데이터 호환 가능하도록 버전/마이그레이션 전략 포함

## **2.3.1 데이터베이스 테이블 구조**

|  |  |  |
| --- | --- | --- |
| No | 테이블명 | 설명 |
| 1 | users | 도서관 이용자(학생/교직원) |
| 2 | librarians | 도서관 관리자 (사서) 계정 |
| 3 | genres | 도서 장르 및 분류 체계 |
| 4 | locations | 자료실 및 서가 위치 정보 |
| 5 | books | 책 정보(Master) |
| 6 | book\_copies | 실물 도서 복권 상태 관리 |
| 7 | loans | 대출 및 반납 이력 |
| 8 | penalties | 이용자 제재 및 패널티 내역 |
| 9 | reservations | 대출 중 도서 예약 관리 |
| 10 | requests | 희망 도서 구매 신청 |
| 11 | inter\_library\_loans | 상호대차 신청 (ILL) |
| 12 | seats | 열람실 좌석 정보 |
| 13 | seat\_reservations | 좌석 예약 및 이용 이력 |
| 14 | entry\_logs | 도서관 출입 기록 |
| 15 | notices | 도서관 공지사항 |
| 16 | book\_reviews | 도서 리뷰 및 평점 |
| 17 | volunteer\_activities | 봉사활동 신청 및 내역 |
| 18 | monthly\_book\_stats | 월별 도서 대출 통계 (집계용) |
| 19 | ai\_analysis\_log | AI 챗봇 통계 분석 질의 로그 |

ER 다이어 그램 주소 <https://dbdiagram.io/d/6921be2b228c5bbc1a06c224>

# **ER 다이어그램**



# **DCL**

권한 설정으로는 사서 와 팀장 들은 도서 시스탬상 관한은 차이가 없습니다.

### **사서**

DROP(테이블 삭제) , ALTER(구조 변경) 등은 못하게 설정 되었습니다.

### **학생**

조회는 자유롭지만 , 수정/삭제는 본인의 신청 내역 등 극히 일부만 가능합니다.

DROP DATABASE IF EXISTS library\_system;

CREATE DATABASE library\_system DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

USE library\_system;

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

-- 1. 기초 정보 (장르, 위치, 사용자, 사서)

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

-- 1.1 장르 (카테고리)

CREATE TABLE genres (

gnr\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '장르 고유 ID',

gnr\_name VARCHAR(50) NOT NULL UNIQUE COMMENT '장르명',

parent\_gnr\_id INT COMMENT '상위 장르 ID (NULL이면 최상위)',

INDEX idx\_parent\_gnr\_id (parent\_gnr\_id),

CONSTRAINT fk\_genres\_parent FOREIGN KEY (parent\_gnr\_id)

REFERENCES genres (gnr\_id) ON DELETE SET NULL

) COMMENT '도서 장르 및 분류 체계';

-- 1.2 자료실 위치

CREATE TABLE locations (

loc\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '위치 ID',

loc\_flr INT NOT NULL COMMENT '층수',

loc\_name VARCHAR(50) NOT NULL UNIQUE COMMENT '자료실명',

loc\_class VARCHAR(50) COMMENT '청구기호 분류 (문자열)'

) COMMENT '도서관 내 자료실 위치 정보';

-- 1.3 사용자 (학생/교직원)

CREATE TABLE users (

usr\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '사용자 ID (학번 등)',

usr\_name VARCHAR(50) NOT NULL COMMENT '이름',

usr\_dept VARCHAR(50) COMMENT '학과 (전공)',

usr\_phone VARCHAR(20) UNIQUE COMMENT '연락처',

usr\_lcnt TINYINT DEFAULT 0 COMMENT '현재 대출 권수',

usr\_gender ENUM('남', '여') NOT NULL COMMENT '성별',

-- 상태 (정상, 정지, 졸업)

usr\_stat ENUM('정상', '정지', '졸업') NOT NULL DEFAULT '정상' COMMENT '상태',

usr\_enrl BOOLEAN NOT NULL COMMENT '재학 여부',

INDEX idx\_usr\_name (usr\_name),

INDEX idx\_usr\_stat (usr\_stat)

) COMMENT '도서관 이용자 정보';

-- 1.4 사서 (관리자)

CREATE TABLE librarians (

lib\_id VARCHAR(30) NOT NULL PRIMARY KEY COMMENT '사서 ID',

lib\_pw VARCHAR(255) NOT NULL COMMENT '비밀번호',

lib\_name VARCHAR(50) NOT NULL COMMENT '이름',

lib\_phone VARCHAR(20) COMMENT '연락처',

lib\_role VARCHAR(20) COMMENT '직책',

-- 자격 등급

lib\_lic ENUM( '준사서', '2급정사서', '1급정사서') NOT NULL COMMENT '자격 등급'

) COMMENT '도서관 관리자 정보';

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

-- 2. 도서 및 실물 관리

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

-- 2.1 도서 서지 정보

CREATE TABLE books (

bk\_id VARCHAR(50) NOT NULL PRIMARY KEY COMMENT '도서 청구기호 (ID)',

gnr\_id INT NOT NULL COMMENT '장르 ID',

bk\_isbn VARCHAR(20) NOT NULL UNIQUE COMMENT 'ISBN',

bk\_title VARCHAR(200) NOT NULL COMMENT '도서명',

bk\_auth VARCHAR(100) COMMENT '저자',

bk\_pub VARCHAR(100) COMMENT '출판사',

bk\_year YEAR COMMENT '출판 연도',

bk\_image VARCHAR(255) COMMENT '표지 이미지',

bk\_intro TEXT COMMENT '책 소개',

INDEX idx\_bk\_title (bk\_title),

INDEX idx\_bk\_auth (bk\_auth),

CONSTRAINT fk\_books\_genres FOREIGN KEY (gnr\_id)

REFERENCES genres (gnr\_id) ON DELETE RESTRICT

) COMMENT '도서 서지 정보 (메타데이터)';

-- 2.2 실물 도서 (복본)

CREATE TABLE book\_copies (

cbk\_id VARCHAR(60) NOT NULL PRIMARY KEY COMMENT '도서 등록번호 (바코드)',

bk\_id VARCHAR(50) NOT NULL COMMENT '청구기호 (FK)',

loc\_id INT NOT NULL COMMENT '소장 위치 (FK)',

cbk\_vol INT NOT NULL COMMENT '복본 번호',

is\_ebook BOOLEAN NOT NULL DEFAULT FALSE COMMENT '전자책 여부',

-- 상태 (대출가능, 대출중, 분실, 파손)

cbk\_stat ENUM('대출가능', '대출중', '분실', '파손') NOT NULL DEFAULT '대출가능' COMMENT '상태',

cbk\_reg DATE NOT NULL COMMENT '등록일',

UNIQUE KEY uk\_book\_vol (bk\_id, cbk\_vol),

INDEX idx\_cbk\_stat (cbk\_stat),

CONSTRAINT fk\_book\_copies\_books FOREIGN KEY (bk\_id)

REFERENCES books (bk\_id) ON DELETE CASCADE,

CONSTRAINT fk\_book\_copies\_locations FOREIGN KEY (loc\_id)

REFERENCES locations (loc\_id) ON DELETE RESTRICT

) COMMENT '실제 비치된 도서 및 전자책 정보';

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

-- 3. 대출/반납 및 제재

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

-- 3.1 대출 기록

CREATE TABLE loans (

loan\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '대출 ID',

cbk\_id VARCHAR(60) NOT NULL COMMENT '도서 등록번호',

usr\_id INT NOT NULL COMMENT '사용자 ID',

loan\_date DATE NOT NULL COMMENT '대출일',

loan\_due DATE NOT NULL COMMENT '반납 예정일',

loan\_ret DATE COMMENT '실제 반납일',

loan\_ext BOOLEAN DEFAULT FALSE COMMENT '연장 여부',

INDEX idx\_loan\_date (loan\_date),

INDEX idx\_loan\_due (loan\_due),

CONSTRAINT fk\_loans\_book\_copies FOREIGN KEY (cbk\_id)

REFERENCES book\_copies (cbk\_id) ON DELETE RESTRICT,

CONSTRAINT fk\_loans\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE RESTRICT

) COMMENT '도서 대출 및 반납 이력';

-- 3.2 패널티 (연체/제재)

CREATE TABLE penalties (

pnl\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '제재 ID',

usr\_id INT NOT NULL COMMENT '사용자 ID',

loan\_id INT COMMENT '관련 대출 ID (옵션)',

pnl\_type VARCHAR(20) NOT NULL COMMENT '제재 사유 (키워드)',

-- [팀원 피드백 반영] 상세 사유 추가

pnl\_reason VARCHAR(200) COMMENT '제재 상세 사유/비고',

pnl\_start DATE NOT NULL COMMENT '시작일',

pnl\_end DATE COMMENT '종료일',

pnl\_actv BOOLEAN NOT NULL DEFAULT TRUE COMMENT '유효 여부',

CONSTRAINT fk\_penalties\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE,

CONSTRAINT fk\_penalties\_loans FOREIGN KEY (loan\_id)

REFERENCES loans (loan\_id) ON DELETE SET NULL

) COMMENT '이용자 제재 내역';

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

-- 4. 신청 및 예약

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

-- 4.1 도서 예약

CREATE TABLE reservations (

resv\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '예약 ID',

usr\_id INT NOT NULL COMMENT '사용자 ID',

bk\_id VARCHAR(50) NOT NULL COMMENT '도서 청구기호',

-- [팀원 피드백 반영] 등록번호(복본) 추가 (특정 도서 지정 예약용)

cbk\_id VARCHAR(60) COMMENT '지정된 도서 등록번호 (옵션)',

resv\_date DATETIME NOT NULL COMMENT '예약 일시',

resv\_seq INT NOT NULL COMMENT '대기 순번',

-- 상태

resv\_stat ENUM('대기중', '수령가능', '취소됨', '수령완료') NOT NULL DEFAULT '대기중' COMMENT '상태',

UNIQUE KEY uk\_resv\_book\_user (usr\_id, bk\_id),

CONSTRAINT fk\_reservations\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE,

CONSTRAINT fk\_reservations\_books FOREIGN KEY (bk\_id)

REFERENCES books (bk\_id) ON DELETE CASCADE,

CONSTRAINT fk\_reservations\_copies FOREIGN KEY (cbk\_id)

REFERENCES book\_copies (cbk\_id) ON DELETE SET NULL

) COMMENT '대출 중 도서 예약';

-- 4.2 희망 도서 신청

CREATE TABLE requests (

req\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '신청 ID',

usr\_id INT NOT NULL COMMENT '신청자 ID',

req\_title VARCHAR(200) NOT NULL COMMENT '도서명',

req\_auth VARCHAR(100) COMMENT '저자',

req\_pub VARCHAR(100) COMMENT '출판사',

req\_year CHAR(4) COMMENT '출판연도 (YYYY)',

req\_isbn VARCHAR(20) COMMENT 'ISBN',

req\_price INT COMMENT '권당 가격',

req\_qty INT NOT NULL DEFAULT 1 COMMENT '신청 권수',

req\_date DATE NOT NULL COMMENT '신청일',

req\_memo VARCHAR(500) NULL COMMENT '신청 사유/비고',

req\_proc\_memo VARCHAR(500) NULL COMMENT '반려 사유/비고',

-- 상태

req\_stat ENUM('대기', '승인', '반려', '구매완료') NOT NULL DEFAULT '대기' COMMENT '처리 상태',

INDEX idx\_req\_stat (req\_stat),

INDEX idx\_req\_date (req\_date),

CONSTRAINT fk\_requests\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '희망 도서 구매 신청';

-- 4.3 상호대차

CREATE TABLE inter\_library\_loans (

ill\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '상호대차 ID',

usr\_id INT NOT NULL COMMENT '신청자 ID',

-- [도서 정보]

ill\_title VARCHAR(200) NOT NULL COMMENT '도서명',

ill\_auth VARCHAR(100) COMMENT '저자',

ill\_pub VARCHAR(100) COMMENT '출판사',

ill\_pub\_year CHAR(4) COMMENT '출판연도',

ill\_isbn VARCHAR(20) NOT NULL COMMENT 'ISBN',

ill\_lib\_name VARCHAR(100) DEFAULT '동의대학교' COMMENT '신청 도서관',

ill\_req\_date DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '신청 일시',

ill\_note VARCHAR(255) COMMENT '참고사항 (사용자 입력)',

-- [팀원 피드백 반영] 관리자 처리 비고 추가

ill\_ans\_note VARCHAR(255) COMMENT '처리 내용/관리자 비고',

-- [상태]

ill\_stat ENUM('신청중', '배송중', '도착', '반납완료', '취소됨') NOT NULL DEFAULT '신청중' COMMENT '상태',

CONSTRAINT fk\_ill\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '타 대학 도서관 상호대차';

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

-- 5. 시설 및 로그

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

-- 5.1 열람실 좌석

CREATE TABLE seats (

seat\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '좌석 고유 ID',

seat\_room ENUM('제1열람실', '제2열람실', '노트북실') NOT NULL COMMENT '열람실 구분',

seat\_num VARCHAR(10) NOT NULL COMMENT '좌석 번호',

seat\_stat ENUM('사용가능', '사용중', '수리중') NOT NULL DEFAULT '사용가능' COMMENT '현재 상태',

UNIQUE KEY uk\_seat\_room\_num (seat\_room, seat\_num)

) COMMENT '열람실 좌석 정보';

-- 5.2 좌석 이용 로그

CREATE TABLE seat\_reservations (

res\_id BIGINT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '좌석예약 ID',

seat\_id INT NOT NULL COMMENT '좌석 ID (FK)',

usr\_id INT NOT NULL COMMENT '사용자 ID (FK)',

res\_start DATETIME NOT NULL COMMENT '입실 시간',

res\_end DATETIME DEFAULT NULL COMMENT '퇴실 시간',

res\_stat ENUM('이용중', '반납완료', '강제퇴실') NOT NULL DEFAULT '이용중' COMMENT '이용 상태',

CONSTRAINT fk\_res\_seats FOREIGN KEY (seat\_id)

REFERENCES seats (seat\_id) ON DELETE CASCADE,

CONSTRAINT fk\_res\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '좌석 이용 및 예약 이력';

-- 5.3 출입 로그

CREATE TABLE entry\_logs (

log\_id BIGINT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '로그 ID',

usr\_id INT NOT NULL COMMENT '사용자 ID',

log\_type ENUM('입실', '퇴실') NOT NULL COMMENT '유형',

log\_time DATETIME NOT NULL COMMENT '시간',

INDEX idx\_log\_time (log\_time),

CONSTRAINT fk\_entry\_logs\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '도서관 출입 기록';

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

-- 6. 커뮤니티 및 활동

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

-- 6.1 공지사항

CREATE TABLE notices (

ntc\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '공지 ID',

lib\_id VARCHAR(30) NOT NULL COMMENT '작성자(사서) ID',

ntc\_title VARCHAR(100) NOT NULL COMMENT '제목',

ntc\_content TEXT COMMENT '내용',

ntc\_file VARCHAR(255) COMMENT '첨부파일',

ntc\_important BOOLEAN NOT NULL DEFAULT FALSE COMMENT '필독 여부',

ntc\_date DATETIME NOT NULL COMMENT '작성일',

ntc\_views INT DEFAULT 0 COMMENT '조회수',

INDEX idx\_ntc\_date (ntc\_date),

INDEX idx\_ntc\_important (ntc\_important),

CONSTRAINT fk\_notices\_librarians FOREIGN KEY (lib\_id)

REFERENCES librarians (lib\_id) ON DELETE RESTRICT

) COMMENT '도서관 공지사항';

-- 6.2 도서 리뷰

CREATE TABLE book\_reviews (

rev\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '리뷰 ID',

bk\_id VARCHAR(50) NOT NULL COMMENT '도서 ID',

usr\_id INT NOT NULL COMMENT '작성자 ID',

rev\_title VARCHAR(100) COMMENT '제목',

rev\_rate TINYINT NOT NULL COMMENT '평점',

rev\_content TEXT COMMENT '내용',

rev\_date DATETIME NOT NULL COMMENT '작성일',

UNIQUE KEY uk\_review\_user\_book (usr\_id, bk\_id),

CONSTRAINT fk\_book\_reviews\_books FOREIGN KEY (bk\_id)

REFERENCES books (bk\_id) ON DELETE CASCADE,

CONSTRAINT fk\_book\_reviews\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '도서 리뷰 및 평점';

-- 6.3 봉사활동

CREATE TABLE volunteer\_activities (

vol\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '활동 ID',

usr\_id INT NOT NULL COMMENT '신청자 ID',

vol\_date DATE NOT NULL COMMENT '활동 날짜',

vol\_start TIME COMMENT '시작 시간',

vol\_end TIME COMMENT '종료 시간',

vol\_desc VARCHAR(200) COMMENT '활동 내용',

-- [팀원 피드백 반영] 처리 비고 추가

vol\_note VARCHAR(200) COMMENT '관리자 처리/반려 사유',

vol\_stat ENUM('대기', '승인', '반려', '완료') NOT NULL DEFAULT '대기' COMMENT '상태',

INDEX idx\_vol\_stat (vol\_stat),

CONSTRAINT fk\_volunteer\_activities\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '봉사활동 신청 및 내역';

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

-- 7. 통계 및 로그

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

-- 7.1 월간 도서 대출 통계

CREATE TABLE monthly\_book\_stats (

stat\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '통계 ID',

stat\_year INT NOT NULL COMMENT '연도',

stat\_month INT NOT NULL COMMENT '월',

bk\_id VARCHAR(50) NOT NULL COMMENT '도서 ID',

loan\_cnt INT DEFAULT 0 COMMENT '대출 횟수',

UNIQUE KEY uk\_monthly\_stats (stat\_year, stat\_month, bk\_id),

CONSTRAINT fk\_monthly\_book\_stats\_books FOREIGN KEY (bk\_id)

REFERENCES books (bk\_id) ON DELETE CASCADE

) COMMENT '월별 인기도서 분석 통계';

-- 7.2 AI 스마트 분석 로그

CREATE TABLE ai\_analysis\_log (

log\_id BIGINT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '로그 고유 ID',

lib\_id VARCHAR(30) NOT NULL COMMENT '질문한 사서 ID (FK)',

quest\_text TEXT NOT NULL COMMENT '사용자 자연어 질문',

gen\_sql TEXT COMMENT 'AI가 생성한 SQL 쿼리',

chart\_type VARCHAR(20) COMMENT '시각화 타입',

error\_msg TEXT COMMENT '에러 메시지',

exec\_time INT COMMENT '소요 시간',

cre\_at DATETIME DEFAULT CURRENT\_TIMESTAMP COMMENT '기록 일시',

INDEX idx\_ai\_log\_lib (lib\_id),

INDEX idx\_ai\_log\_cre (cre\_at),

CONSTRAINT fk\_ai\_log\_librarians FOREIGN KEY (lib\_id)

REFERENCES librarians (lib\_id) ON DELETE CASCADE ON UPDATE CASCADE

) COMMENT 'AI 챗봇 통계 분석 질의 로그';

# 

# **DDL 쿼리문**

DROP DATABASE IF EXISTS library\_system;

CREATE DATABASE library\_system DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

USE library\_system;

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

-- 1. 기초 정보 (장르, 위치, 사용자, 사서)

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

-- 1.1 장르 (카테고리)

CREATE TABLE genres (

gnr\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '장르 고유 ID',

gnr\_name VARCHAR(50) NOT NULL UNIQUE COMMENT '장르명',

parent\_gnr\_id INT COMMENT '상위 장르 ID (NULL이면 최상위)',

INDEX idx\_parent\_gnr\_id (parent\_gnr\_id),

CONSTRAINT fk\_genres\_parent FOREIGN KEY (parent\_gnr\_id)

REFERENCES genres (gnr\_id) ON DELETE SET NULL

) COMMENT '도서 장르 및 분류 체계';

-- 1.2 자료실 위치

CREATE TABLE locations (

loc\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '위치 ID',

loc\_flr INT NOT NULL COMMENT '층수',

loc\_name VARCHAR(50) NOT NULL UNIQUE COMMENT '자료실명',

loc\_class VARCHAR(50) COMMENT '청구기호 분류 (문자열)'

) COMMENT '도서관 내 자료실 위치 정보';

-- 1.3 사용자 (학생/교직원)

CREATE TABLE users (

usr\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '사용자 ID (학번 등)',

usr\_name VARCHAR(50) NOT NULL COMMENT '이름',

usr\_dept VARCHAR(50) COMMENT '학과 (전공)',

usr\_phone VARCHAR(20) UNIQUE COMMENT '연락처',

usr\_lcnt TINYINT DEFAULT 0 COMMENT '현재 대출 권수',

usr\_gender ENUM('남', '여') NOT NULL COMMENT '성별',

-- 상태 (정상, 정지, 졸업)

usr\_stat ENUM('정상', '정지', '졸업') NOT NULL DEFAULT '정상' COMMENT '상태',

usr\_enrl BOOLEAN NOT NULL COMMENT '재학 여부',

INDEX idx\_usr\_name (usr\_name),

INDEX idx\_usr\_stat (usr\_stat)

) COMMENT '도서관 이용자 정보';

-- 1.4 사서 (관리자)

CREATE TABLE librarians (

lib\_id VARCHAR(30) NOT NULL PRIMARY KEY COMMENT '사서 ID',

lib\_pw VARCHAR(255) NOT NULL COMMENT '비밀번호',

lib\_name VARCHAR(50) NOT NULL COMMENT '이름',

lib\_phone VARCHAR(20) COMMENT '연락처',

lib\_role VARCHAR(20) COMMENT '직책',

-- 자격 등급

lib\_lic ENUM( '준사서', '2급정사서', '1급정사서') NOT NULL COMMENT '자격 등급'

) COMMENT '도서관 관리자 정보';

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

-- 2. 도서 및 실물 관리

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

-- 2.1 도서 서지 정보

CREATE TABLE books (

bk\_id VARCHAR(50) NOT NULL PRIMARY KEY COMMENT '도서 청구기호 (ID)',

gnr\_id INT NOT NULL COMMENT '장르 ID',

bk\_isbn VARCHAR(20) NOT NULL UNIQUE COMMENT 'ISBN',

bk\_title VARCHAR(200) NOT NULL COMMENT '도서명',

bk\_auth VARCHAR(100) COMMENT '저자',

bk\_pub VARCHAR(100) COMMENT '출판사',

bk\_year YEAR COMMENT '출판 연도',

bk\_image VARCHAR(255) COMMENT '표지 이미지',

bk\_intro TEXT COMMENT '책 소개',

INDEX idx\_bk\_title (bk\_title),

INDEX idx\_bk\_auth (bk\_auth),

CONSTRAINT fk\_books\_genres FOREIGN KEY (gnr\_id)

REFERENCES genres (gnr\_id) ON DELETE RESTRICT

) COMMENT '도서 서지 정보 (메타데이터)';

-- 2.2 실물 도서 (복본)

CREATE TABLE book\_copies (

cbk\_id VARCHAR(60) NOT NULL PRIMARY KEY COMMENT '도서 등록번호 (바코드)',

bk\_id VARCHAR(50) NOT NULL COMMENT '청구기호 (FK)',

loc\_id INT NOT NULL COMMENT '소장 위치 (FK)',

cbk\_vol INT NOT NULL COMMENT '복본 번호',

is\_ebook BOOLEAN NOT NULL DEFAULT FALSE COMMENT '전자책 여부',

-- 상태 (대출가능, 대출중, 분실, 파손)

cbk\_stat ENUM('대출가능', '대출중', '분실', '파손') NOT NULL DEFAULT '대출가능' COMMENT '상태',

cbk\_reg DATE NOT NULL COMMENT '등록일',

UNIQUE KEY uk\_book\_vol (bk\_id, cbk\_vol),

INDEX idx\_cbk\_stat (cbk\_stat),

CONSTRAINT fk\_book\_copies\_books FOREIGN KEY (bk\_id)

REFERENCES books (bk\_id) ON DELETE CASCADE,

CONSTRAINT fk\_book\_copies\_locations FOREIGN KEY (loc\_id)

REFERENCES locations (loc\_id) ON DELETE RESTRICT

) COMMENT '실제 비치된 도서 및 전자책 정보';

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

-- 3. 대출/반납 및 제재

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

-- 3.1 대출 기록

CREATE TABLE loans (

loan\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '대출 ID',

cbk\_id VARCHAR(60) NOT NULL COMMENT '도서 등록번호',

usr\_id INT NOT NULL COMMENT '사용자 ID',

loan\_date DATE NOT NULL COMMENT '대출일',

loan\_due DATE NOT NULL COMMENT '반납 예정일',

loan\_ret DATE COMMENT '실제 반납일',

loan\_ext BOOLEAN DEFAULT FALSE COMMENT '연장 여부',

INDEX idx\_loan\_date (loan\_date),

INDEX idx\_loan\_due (loan\_due),

CONSTRAINT fk\_loans\_book\_copies FOREIGN KEY (cbk\_id)

REFERENCES book\_copies (cbk\_id) ON DELETE RESTRICT,

CONSTRAINT fk\_loans\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE RESTRICT

) COMMENT '도서 대출 및 반납 이력';

-- 3.2 패널티 (연체/제재)

CREATE TABLE penalties (

pnl\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '제재 ID',

usr\_id INT NOT NULL COMMENT '사용자 ID',

loan\_id INT COMMENT '관련 대출 ID (옵션)',

pnl\_type VARCHAR(20) NOT NULL COMMENT '제재 사유 (키워드)',

-- [팀원 피드백 반영] 상세 사유 추가

pnl\_reason VARCHAR(200) COMMENT '제재 상세 사유/비고',

pnl\_start DATE NOT NULL COMMENT '시작일',

pnl\_end DATE COMMENT '종료일',

pnl\_actv BOOLEAN NOT NULL DEFAULT TRUE COMMENT '유효 여부',

CONSTRAINT fk\_penalties\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE,

CONSTRAINT fk\_penalties\_loans FOREIGN KEY (loan\_id)

REFERENCES loans (loan\_id) ON DELETE SET NULL

) COMMENT '이용자 제재 내역';

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

-- 4. 신청 및 예약

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

-- 4.1 도서 예약

CREATE TABLE reservations (

resv\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '예약 ID',

usr\_id INT NOT NULL COMMENT '사용자 ID',

bk\_id VARCHAR(50) NOT NULL COMMENT '도서 청구기호',

-- [팀원 피드백 반영] 등록번호(복본) 추가 (특정 도서 지정 예약용)

cbk\_id VARCHAR(60) COMMENT '지정된 도서 등록번호 (옵션)',

resv\_date DATETIME NOT NULL COMMENT '예약 일시',

resv\_seq INT NOT NULL COMMENT '대기 순번',

-- 상태

resv\_stat ENUM('대기중', '수령가능', '취소됨', '수령완료') NOT NULL DEFAULT '대기중' COMMENT '상태',

UNIQUE KEY uk\_resv\_book\_user (usr\_id, bk\_id),

CONSTRAINT fk\_reservations\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE,

CONSTRAINT fk\_reservations\_books FOREIGN KEY (bk\_id)

REFERENCES books (bk\_id) ON DELETE CASCADE,

CONSTRAINT fk\_reservations\_copies FOREIGN KEY (cbk\_id)

REFERENCES book\_copies (cbk\_id) ON DELETE SET NULL

) COMMENT '대출 중 도서 예약';

-- 4.2 희망 도서 신청

CREATE TABLE requests (

req\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '신청 ID',

usr\_id INT NOT NULL COMMENT '신청자 ID',

req\_title VARCHAR(200) NOT NULL COMMENT '도서명',

req\_auth VARCHAR(100) COMMENT '저자',

req\_pub VARCHAR(100) COMMENT '출판사',

req\_year CHAR(4) COMMENT '출판연도 (YYYY)',

req\_isbn VARCHAR(20) COMMENT 'ISBN',

req\_price INT COMMENT '권당 가격',

req\_qty INT NOT NULL DEFAULT 1 COMMENT '신청 권수',

req\_date DATE NOT NULL COMMENT '신청일',

req\_memo VARCHAR(500) NULL COMMENT '신청 사유/비고',

req\_proc\_memo VARCHAR(500) NULL COMMENT '반려 사유/비고',

-- 상태

req\_stat ENUM('대기', '승인', '반려', '구매완료') NOT NULL DEFAULT '대기' COMMENT '처리 상태',

INDEX idx\_req\_stat (req\_stat),

INDEX idx\_req\_date (req\_date),

CONSTRAINT fk\_requests\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '희망 도서 구매 신청';

-- 4.3 상호대차

CREATE TABLE inter\_library\_loans (

ill\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '상호대차 ID',

usr\_id INT NOT NULL COMMENT '신청자 ID',

-- [도서 정보]

ill\_title VARCHAR(200) NOT NULL COMMENT '도서명',

ill\_auth VARCHAR(100) COMMENT '저자',

ill\_pub VARCHAR(100) COMMENT '출판사',

ill\_pub\_year CHAR(4) COMMENT '출판연도',

ill\_isbn VARCHAR(20) NOT NULL COMMENT 'ISBN',

ill\_lib\_name VARCHAR(100) DEFAULT '동의대학교' COMMENT '신청 도서관',

ill\_req\_date DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '신청 일시',

ill\_note VARCHAR(255) COMMENT '참고사항 (사용자 입력)',

-- [팀원 피드백 반영] 관리자 처리 비고 추가

ill\_ans\_note VARCHAR(255) COMMENT '처리 내용/관리자 비고',

-- [상태]

ill\_stat ENUM('신청중', '배송중', '도착', '반납완료', '취소됨') NOT NULL DEFAULT '신청중' COMMENT '상태',

CONSTRAINT fk\_ill\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '타 대학 도서관 상호대차';

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

-- 5. 시설 및 로그

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

-- 5.1 열람실 좌석

CREATE TABLE seats (

seat\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '좌석 고유 ID',

seat\_room ENUM('제1열람실', '제2열람실', '노트북실') NOT NULL COMMENT '열람실 구분',

seat\_num VARCHAR(10) NOT NULL COMMENT '좌석 번호',

seat\_stat ENUM('사용가능', '사용중', '수리중') NOT NULL DEFAULT '사용가능' COMMENT '현재 상태',

UNIQUE KEY uk\_seat\_room\_num (seat\_room, seat\_num)

) COMMENT '열람실 좌석 정보';

-- 5.2 좌석 이용 로그

CREATE TABLE seat\_reservations (

res\_id BIGINT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '좌석예약 ID',

seat\_id INT NOT NULL COMMENT '좌석 ID (FK)',

usr\_id INT NOT NULL COMMENT '사용자 ID (FK)',

res\_start DATETIME NOT NULL COMMENT '입실 시간',

res\_end DATETIME DEFAULT NULL COMMENT '퇴실 시간',

res\_stat ENUM('이용중', '반납완료', '강제퇴실') NOT NULL DEFAULT '이용중' COMMENT '이용 상태',

CONSTRAINT fk\_res\_seats FOREIGN KEY (seat\_id)

REFERENCES seats (seat\_id) ON DELETE CASCADE,

CONSTRAINT fk\_res\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '좌석 이용 및 예약 이력';

-- 5.3 출입 로그

CREATE TABLE entry\_logs (

log\_id BIGINT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '로그 ID',

usr\_id INT NOT NULL COMMENT '사용자 ID',

log\_type ENUM('입실', '퇴실') NOT NULL COMMENT '유형',

log\_time DATETIME NOT NULL COMMENT '시간',

INDEX idx\_log\_time (log\_time),

CONSTRAINT fk\_entry\_logs\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '도서관 출입 기록';

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

-- 6. 커뮤니티 및 활동

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

-- 6.1 공지사항

CREATE TABLE notices (

ntc\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '공지 ID',

lib\_id VARCHAR(30) NOT NULL COMMENT '작성자(사서) ID',

ntc\_title VARCHAR(100) NOT NULL COMMENT '제목',

ntc\_content TEXT COMMENT '내용',

ntc\_file VARCHAR(255) COMMENT '첨부파일',

ntc\_important BOOLEAN NOT NULL DEFAULT FALSE COMMENT '필독 여부',

ntc\_date DATETIME NOT NULL COMMENT '작성일',

ntc\_views INT DEFAULT 0 COMMENT '조회수',

INDEX idx\_ntc\_date (ntc\_date),

INDEX idx\_ntc\_important (ntc\_important),

CONSTRAINT fk\_notices\_librarians FOREIGN KEY (lib\_id)

REFERENCES librarians (lib\_id) ON DELETE RESTRICT

) COMMENT '도서관 공지사항';

-- 6.2 도서 리뷰

CREATE TABLE book\_reviews (

rev\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '리뷰 ID',

bk\_id VARCHAR(50) NOT NULL COMMENT '도서 ID',

usr\_id INT NOT NULL COMMENT '작성자 ID',

rev\_title VARCHAR(100) COMMENT '제목',

rev\_rate TINYINT NOT NULL COMMENT '평점',

rev\_content TEXT COMMENT '내용',

rev\_date DATETIME NOT NULL COMMENT '작성일',

UNIQUE KEY uk\_review\_user\_book (usr\_id, bk\_id),

CONSTRAINT fk\_book\_reviews\_books FOREIGN KEY (bk\_id)

REFERENCES books (bk\_id) ON DELETE CASCADE,

CONSTRAINT fk\_book\_reviews\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '도서 리뷰 및 평점';

-- 6.3 봉사활동

CREATE TABLE volunteer\_activities (

vol\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '활동 ID',

usr\_id INT NOT NULL COMMENT '신청자 ID',

vol\_date DATE NOT NULL COMMENT '활동 날짜',

vol\_start TIME COMMENT '시작 시간',

vol\_end TIME COMMENT '종료 시간',

vol\_desc VARCHAR(200) COMMENT '활동 내용',

-- [팀원 피드백 반영] 처리 비고 추가

vol\_note VARCHAR(200) COMMENT '관리자 처리/반려 사유',

vol\_stat ENUM('대기', '승인', '반려', '완료') NOT NULL DEFAULT '대기' COMMENT '상태',

INDEX idx\_vol\_stat (vol\_stat),

CONSTRAINT fk\_volunteer\_activities\_users FOREIGN KEY (usr\_id)

REFERENCES users (usr\_id) ON DELETE CASCADE

) COMMENT '봉사활동 신청 및 내역';

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

-- 7. 통계 및 로그

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

-- 7.1 월간 도서 대출 통계

CREATE TABLE monthly\_book\_stats (

stat\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '통계 ID',

stat\_year INT NOT NULL COMMENT '연도',

stat\_month INT NOT NULL COMMENT '월',

bk\_id VARCHAR(50) NOT NULL COMMENT '도서 ID',

loan\_cnt INT DEFAULT 0 COMMENT '대출 횟수',

UNIQUE KEY uk\_monthly\_stats (stat\_year, stat\_month, bk\_id),

CONSTRAINT fk\_monthly\_book\_stats\_books FOREIGN KEY (bk\_id)

REFERENCES books (bk\_id) ON DELETE CASCADE

) COMMENT '월별 인기도서 분석 통계';

-- 7.2 AI 스마트 분석 로그

CREATE TABLE ai\_analysis\_log (

log\_id BIGINT NOT NULL AUTO\_INCREMENT PRIMARY KEY COMMENT '로그 고유 ID',

lib\_id VARCHAR(30) NOT NULL COMMENT '질문한 사서 ID (FK)',

quest\_text TEXT NOT NULL COMMENT '사용자 자연어 질문',

gen\_sql TEXT COMMENT 'AI가 생성한 SQL 쿼리',

chart\_type VARCHAR(20) COMMENT '시각화 타입',

error\_msg TEXT COMMENT '에러 메시지',

exec\_time INT COMMENT '소요 시간',

cre\_at DATETIME DEFAULT CURRENT\_TIMESTAMP COMMENT '기록 일시',

INDEX idx\_ai\_log\_lib (lib\_id),

INDEX idx\_ai\_log\_cre (cre\_at),

CONSTRAINT fk\_ai\_log\_librarians FOREIGN KEY (lib\_id)

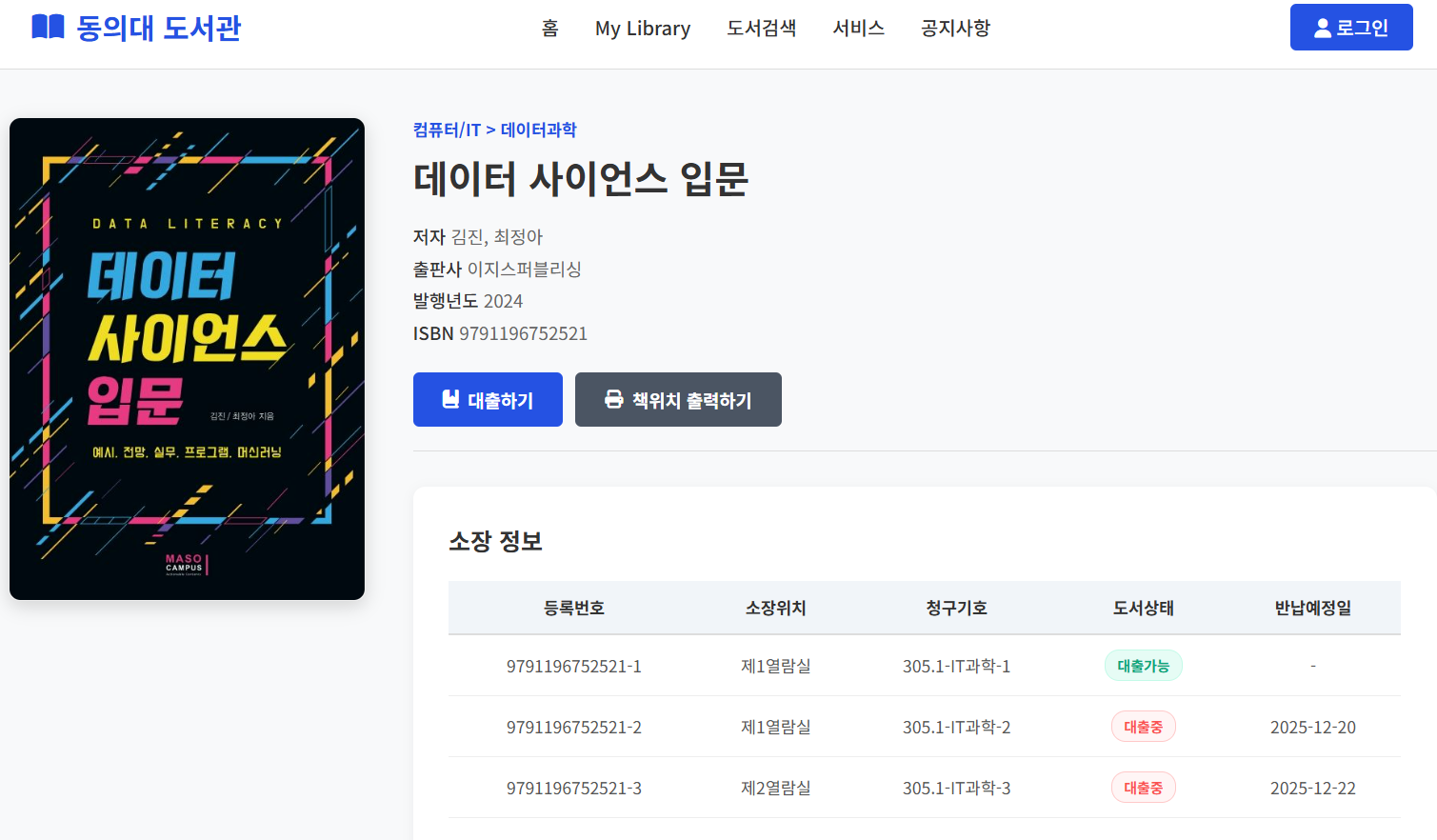
REFERENCES librarians (lib\_id) ON DELETE CASCADE ON UPDATE CASCADE

) COMMENT 'AI 챗봇 통계 분석 질의 로그';

# **3. 기능구현화면 및 쿼리매칭**

# **3.1 학생화면페이지**

### **도서상서 페이지화면**

****

**도서상서 페이지 쿼리**

USE library\_system;

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

-- 1-1. 도서 상세 메타데이터 조회

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

SELECT

T1.bk\_title AS 서명,

T1.bk\_auth AS 저자,

T1.bk\_pub AS 출판사,

T1.bk\_year AS 발행연도,

T1.bk\_isbn AS ISBN,

T1.bk\_image AS 표지이미지,

T1.bk\_intro AS 책소개,

T2.gnr\_name AS 장르

FROM

books T1

LEFT JOIN

genres T2 ON T1.gnr\_id = T2.gnr\_id

WHERE

T1.bk\_id = 1; -- [변수] 선택된 도서 청구기호(ID)

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

-- 1-2. 복본별 소장 현황 및 대출 상태 조회

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

SELECT

T1.cbk\_id AS 등록번호, -- (예: EM001234)

T2.loc\_name AS 소장처, -- (예: 제1열람실)

-- [수정] 청구기호는 위치정보(loc\_class)가 아니라 도서ID(bk\_id)를 사용

CONCAT(T1.bk\_id, ' c.', T1.cbk\_vol) AS 청구기호,

T1.cbk\_stat AS 복본상태, -- ENUM ('대출가능', '대출중', '분실', '파손')

DATE\_FORMAT(T3.loan\_due, '%Y-%m-%d') AS 반납예정일 -- 대출중일 경우 날짜 표시

FROM

book\_copies T1

JOIN

locations T2 ON T1.loc\_id = T2.loc\_id

LEFT JOIN

loans T3 ON T1.cbk\_id = T3.cbk\_id AND T3.loan\_ret IS NULL -- 현재 대출중인 기록만 조인

WHERE

T1.bk\_id = 1 -- [변수] 선택된 도서 ID

ORDER BY

T1.cbk\_vol ASC; -- 복본 번호순 정렬

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

-- 1-3. 리뷰 목록 조회 (최신순)

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

SELECT

T1.rev\_title AS 제목,

T1.rev\_rate AS 평점,

T1.rev\_content AS 내용,

T2.usr\_name AS 작성자,

DATE\_FORMAT(T1.rev\_date, '%Y-%m-%d') AS 작성일

FROM

book\_reviews T1

JOIN

users T2 ON T1.usr\_id = T2.usr\_id

WHERE

T1.bk\_id = 1 -- [변수] 선택된 도서 ID

ORDER BY

T1.rev\_date DESC;

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

-- 1-4. '대출하기' 버튼 활성화 여부 확인

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

-- 대출 가능한 복본이 1개 이상이면 버튼 활성화

SELECT COUNT(\*) AS avail\_count

FROM book\_copies

WHERE bk\_id = 1

AND cbk\_stat = '대출가능'; -- [ENUM 체크] DDL 값 일치

INSERT INTO genres (gnr\_id, gnr\_name)

VALUES (

1, -

'컴퓨터/IT' -- 장르 이름 (예시)

) ON DUPLICATE KEY UPDATE gnr\_name = VALUES(gnr\_name);

INSERT INTO locations (loc\_id, loc\_name, loc\_flr)

VALUES (

1,

'일반열람실',

1

) ON DUPLICATE KEY UPDATE loc\_name = VALUES(loc\_name);

INSERT INTO users (usr\_id, usr\_name, usr\_dept, usr\_stat, usr\_enrl)

VALUES (

'20233849',

'조기강',

'컴퓨터공학과',

'정상',

TRUE

) ON DUPLICATE KEY UPDATE usr\_name = VALUES(usr\_name);

INSERT INTO books (bk\_id, bk\_title, bk\_auth, bk\_pub, bk\_year, bk\_isbn, gnr\_id)

VALUES (

1, -- bk\_id: SELECT 쿼리에서 사용된 값

'[파이썬] 챗지피티와 함께 배우는 딥러닝 입문',

'김태영',

'위키북스',

'2024',

'9791196752521', -- HTML의 ISBN 값

1

) ON DUPLICATE KEY UPDATE bk\_title = VALUES(bk\_title);

INSERT INTO book\_copies (cbk\_id, bk\_id, loc\_id, cbk\_vol, cbk\_stat, cbk\_reg)

VALUES (

'9791196752521-1',

1,

1,

1,

'대출가능',

CURDATE()

) ON DUPLICATE KEY UPDATE cbk\_stat = VALUES(cbk\_stat);

-- =====================================================

-- 2. 기능 동작 (INSERT / UPDATE)

-- =====================================================

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

-- 2-1. 도서 대출 처리 (트랜잭션 필수)

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

START TRANSACTION;

-- [Step 1] 대출 기록 생성

INSERT INTO loans (

cbk\_id,

usr\_id,

loan\_date,

loan\_due,

loan\_ret,

loan\_ext

) VALUES (

'9791196752521-1', -- [변수] 선택된 복본 ID (등록번호)

'20233849', -- [변수] 로그인한 사용자 ID

CURDATE(), -- 대출일: 오늘

DATE\_ADD(CURDATE(), INTERVAL 7 DAY), -- 반납예정일: 7일 후

NULL, -- 반납일: 아직 없음

FALSE -- 연장여부: 초기값 False

);

-- [Step 2] 복본 상태 변경 ('대출가능' -> '대출중')

UPDATE book\_copies

SET cbk\_stat = '대출중' -- [ENUM 체크] DDL 값 일치

WHERE cbk\_id = '9791196752521-1'; -- [변수] 위와 동일한 복본 ID

COMMIT;

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

-- 2-2. 도서 예약 등록

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

-- 해당 도서에 대한 나의 대기 순번을 자동 계산하여 삽입

INSERT INTO reservations (

bk\_id,

usr\_id,

resv\_date,

resv\_seq,

resv\_stat

)

SELECT

1, -- [변수] 도서 ID

20233849, -- [변수] 사용자 ID

NOW(), -- 예약 일시

IFNULL(MAX(resv\_seq), 0) + 1, -- 현재 최대 순번 + 1

'대기중' -- [ENUM 체크] 초기 상태

FROM

reservations

WHERE

bk\_id = 1; -- [변수] 도서 ID (그룹핑용)

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

-- 2-3. 리뷰 등록

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

INSERT INTO book\_reviews (

bk\_id,

usr\_id,

rev\_title,

rev\_rate,

rev\_content,

rev\_date

) VALUES (

1, -- [변수] 도서 ID

'20233849', -- [변수] 사용자 ID

'배송도 빠르고 책 상태도 좋습니다.', -- [변수] 리뷰 제목

5, -- [변수] 평점 (1~5)

'학교 도서관에 희망도서 신청해서 들어오자마자 빌려봤어요. 깨끗해서 좋네요. 내용은 그럭저럭 평이합니다.', -- [변수] 내용

NOW()

);

### **도서관 메인화면**



### **도서관 메인화면 쿼리**

USE library\_system;

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

-- 1. [헤더] 로그인 사용자 정보 조회

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

SELECT

usr\_name,

usr\_stat -- (정상/정지/졸업)

FROM

users

WHERE

usr\_id = 20233849; -- [변수] 로그인한 사용자 ID

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

-- 2. [대시보드] 도서관 현황 통계 (한 번에 조회)

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

-- 용도: 메인 화면의 숫자 데이터 바인딩 (장서수, 이용자수, 좌석현황)

SELECT

-- (1) 총 장서 수 (실물 + 전자책 포함 모든 복본)

(SELECT COUNT(\*) FROM book\_copies) AS val\_total\_books,

-- (2) 전자책 수 (is\_ebook = TRUE)

(SELECT COUNT(\*) FROM book\_copies WHERE is\_ebook = 1) AS val\_ebooks,

-- (3) 현재 도서관 이용자 수 (추정치: 최근 3시간 내 '입실' 기록 수)

-- \* DDL log\_type ENUM: '입실', '퇴실'

(SELECT COUNT(DISTINCT usr\_id)

FROM entry\_logs

WHERE log\_time >= DATE\_SUB(NOW(), INTERVAL 3 HOUR)

AND log\_type = '입실') AS val\_current\_users,

-- (4) 열람실 전체 좌석 수

(SELECT COUNT(\*) FROM seats) AS val\_total\_seats,

-- (5) 사용 중인 좌석 수

-- \* DDL seat\_stat ENUM: '사용가능', '사용중', '수리중'

(SELECT COUNT(\*) FROM seats WHERE seat\_stat = '사용중') AS val\_occupied\_seats;

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

-- 3. [공지사항] 최신 공지 목록 (Top 3)

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

SELECT

ntc\_id,

ntc\_title,

DATE\_FORMAT(ntc\_date, '%m.%d') AS ntc\_date\_fmt -- 예: 12.09

FROM

notices

ORDER BY

ntc\_important DESC, -- 필독 공지 우선

ntc\_date DESC,

ntc\_id DESC

LIMIT 3;

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

-- 4. [인기 도서] 이달의 베스트셀러 Top 5

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

-- 조건: monthly\_book\_stats 테이블 활용 (성능 최적화)

SELECT

T1.bk\_id,

T1.bk\_title, -- 책 제목

T1.bk\_auth, -- 저자

T1.bk\_image -- 표지 이미지

FROM

books T1

JOIN

monthly\_book\_stats T2 ON T1.bk\_id = T2.bk\_id

WHERE

T2.stat\_year = YEAR(CURDATE())

AND T2.stat\_month = MONTH(CURDATE())

ORDER BY

T2.loan\_cnt DESC

LIMIT 5;

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

-- 5. [신착 도서] 새로 들어온 책 Top 5

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

-- 복본(book\_copies)의 등록일(cbk\_reg) 기준 최신순

SELECT

B.bk\_id,

B.bk\_title,

B.bk\_auth,

B.bk\_image,

MAX(C.cbk\_reg) AS recent\_reg\_date

FROM

books B

JOIN

book\_copies C ON B.bk\_id = C.bk\_id

GROUP BY

B.bk\_id

ORDER BY

recent\_reg\_date DESC

LIMIT 5;

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

-- 6. [검색] 도서 통합 검색 (메인 배너)

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

SELECT

bk\_id,

bk\_title,

bk\_auth,

bk\_pub,

bk\_year,

bk\_image

FROM

books

WHERE

bk\_title LIKE CONCAT('%', '데이터 사이언스 입문', '%') -- [변수] 검색어

OR bk\_auth LIKE CONCAT('%', '김진,최정아', '%') -- [변수] 검색어

ORDER BY

bk\_title ASC

LIMIT 10;

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

-- 7. [로그] 접속 기록 (방문자 집계용)

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

INSERT INTO entry\_logs (

usr\_id,

log\_type,

log\_time

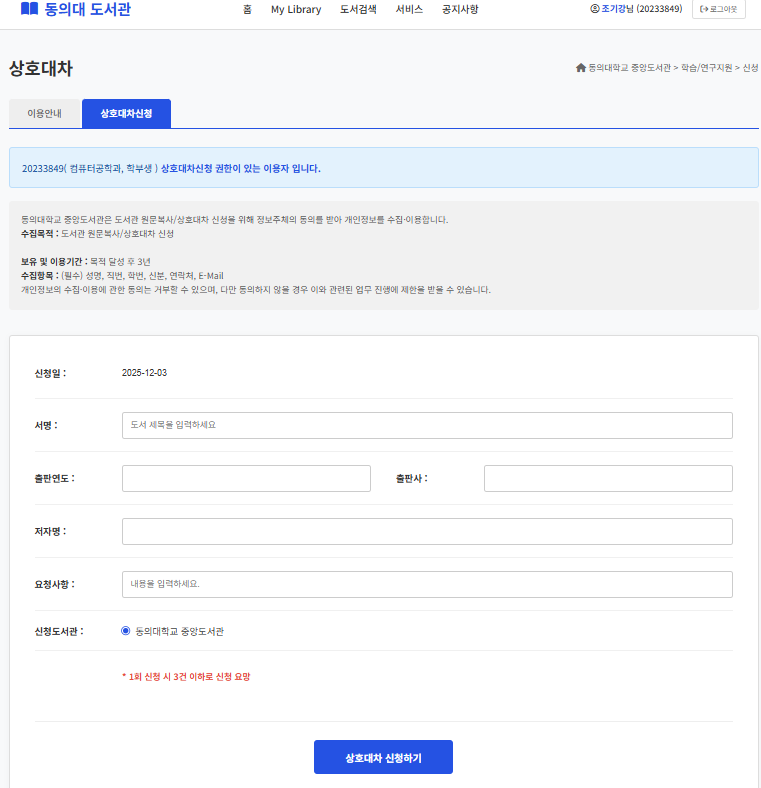
) VALUES (

20233849, -- [변수] 사용자 ID

'입실', -- DDL ENUM 값 준수

NOW()

);**상호대차 신청 페이지 화면**



### 

### **상호대차 신청 쿼리**

USE library\_system;

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

-- 1. [헤더/폼 로딩] 사용자 기본 정보 및 오늘 날짜 조회

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

SELECT

usr\_name,

usr\_stat, -- (정상/정지 여부 확인용)

CURDATE() AS today\_date -- 신청일 자동 표시용

FROM

users

WHERE

usr\_id = 20233849; -- [변수] 로그인한 사용자 ID

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

-- 2. [신청 버튼] 상호대차 신청 완료 (INSERT)

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

-- 화면의 입력 폼(서명, 저자, 출판사, 연도, ISBN,메모)을 저장.

-- 신청 도서관 ( Default '동의대학교')

INSERT INTO inter\_library\_loans (

usr\_id, -- 신청자 학번

ill\_title, -- [필수] 도서명

ill\_auth, -- [옵션] 저자

ill\_pub, -- [옵션] 출판사

ill\_pub\_year, -- [옵션] 출판연도 (CHAR 4)

ill\_isbn, -- [필수] ISBN (식별자)

ill\_note, -- [옵션] 참고사항 (사용자 요청 메시지)

ill\_req\_date, -- 신청 일시

ill\_stat -- 상태 (초기값)

)

VALUES (

20233849, -- [변수] usr\_id

'조기강', -- [변수] ill\_title (서명)

'김수정, 이민호', -- [변수] ill\_auth (저자)

'김수정, 이민호', -- [변수] ill\_pub (출판사)

'2025', -- [변수] ill\_pub\_year (출판연도)

'9791199551008', -- [변수] ill\_isbn (ISBN)

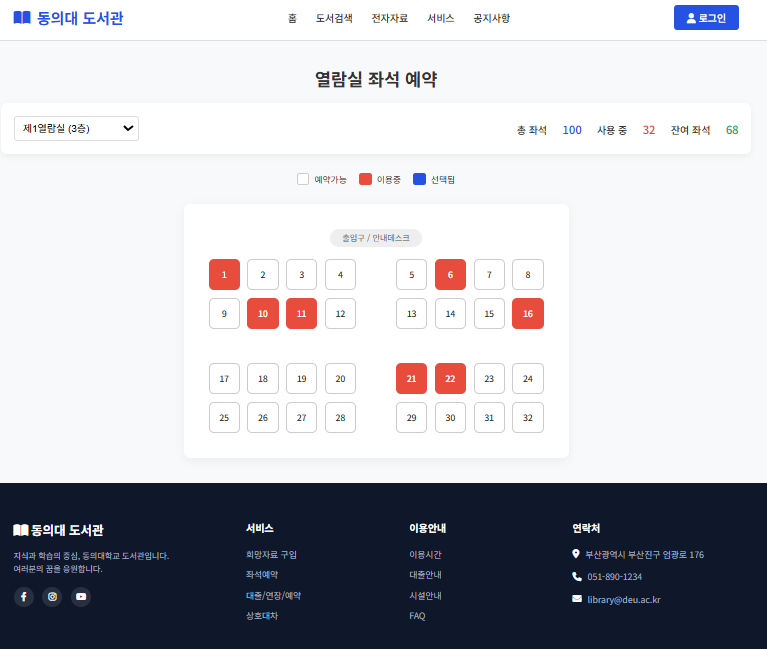
'타 학교 자료입니다. 빠른 처리 부탁드립니다.', -- [변수] ill\_note (참고사항)

NOW(), -- 현재 시간

'신청중' -- [ENUM 체크] DDL 값 일치 (초기 상태)

);

### **좌석배치 예약 화면**



### 

### **좌석배치 예약 쿼리**

USE library\_system;

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

-- 1-1. 특정 열람실의 좌석 배치도 및 상태 조회

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

SELECT

s.seat\_id,

s.seat\_num, -- 좌석 번호 (A-01, B-05 등)

s.seat\_stat, -- 상태 ('사용가능', '사용중', '수리중')

-- 현재 이 좌석을 이용 중인 사용자 이름 (관리자용, 학생에겐 마스킹 처리 필요)

(SELECT u.usr\_name

FROM seat\_reservations sr

JOIN users u ON sr.usr\_id = u.usr\_id

WHERE sr.seat\_id = s.seat\_id AND sr.res\_stat = '이용중'

LIMIT 1) AS current\_user\_name

FROM

seats s

WHERE

s.seat\_room = '제1열람실'; -- [변수] 열람실 이름

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

-- 1-2. 열람실 현황 통계 (잔여 좌석 확인)

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

SELECT

COUNT(\*) AS 총\_좌석수,

-- [수정] ENUM 값 한글화 ('occupied' -> '사용중')

SUM(CASE WHEN seat\_stat = '사용중' THEN 1 ELSE 0 END) AS 사용중\_좌석,

-- [수정] ENUM 값 한글화 ('available' -> '사용가능')

SUM(CASE WHEN seat\_stat = '사용가능' THEN 1 ELSE 0 END) AS 잔여\_좌석,

-- 잔여석 비율 계산

ROUND(SUM(CASE WHEN seat\_stat = '사용가능' THEN 1 ELSE 0 END) \* 100.0 / COUNT(\*), 1) AS 잔여율

FROM

seats

WHERE

seat\_room = '제1열람실'; -- [변수] 열람실 이름

-- =====================================================

-- 2. [입실] 좌석 예약 확정 (트랜잭션 필수)

-- =====================================================

START TRANSACTION;

-- [Step 1] 물리적 좌석 선점 (상태 변경)

-- [수정] ENUM 값 한글화 ('available' -> '사용가능', 'occupied' -> '사용중')

UPDATE seats

SET seat\_stat = '사용중'

WHERE

seat\_id = 101 -- [변수] 선택한 좌석 ID

AND seat\_stat = '사용가능'; -- [중요] 동시성 제어 조건

-- [Application Check]

-- 위 UPDATE 문이 영향을 준 행(Row)이 0개라면 -> "이미 예약된 좌석입니다" 에러 & ROLLBACK

-- [Step 2] 예약 이력(Log) 생성

-- [수정] ENUM 값 한글화 (res\_stat: '이용중')

INSERT INTO seats (seat\_id, seat\_room, seat\_num, seat\_stat)

VALUES (

101, -- seat\_id와 일치시켜야 함

'제1열람실', -- 열람실 이름 (예시)

'A-01', -- 좌석 번호 (예시)

'사용가능' -- 초기 상태

);

INSERT INTO users (usr\_id, usr\_name, usr\_dept, usr\_stat, usr\_enrl)

VALUES (

'20233849', -- usr\_id와 일치시켜야 함

'조기강', -- 이름 (예시)

'컴퓨터공학과', -- 학과 (예시)

'정상', -- 상태 (예시)

TRUE

);

COMMIT;

-- =====================================================

-- 3. [퇴실] 좌석 반납 처리 (트랜잭션 필수)

-- =====================================================

START TRANSACTION;

-- [Step 1] 예약 이력 종료

-- [수정] ENUM 값 한글화 ('이용중' -> '반납완료')

UPDATE seat\_reservations

SET

res\_end = NOW(),

res\_stat = '반납완료'

WHERE

usr\_id = 20233849 -- [변수] 로그인한 사용자 ID

AND res\_stat = '이용중';

-- [Step 2] 물리적 좌석 상태 원상복구

-- [수정] ENUM 값 한글화 ('사용중' -> '사용가능')

UPDATE seats

SET seat\_stat = '사용가능'

WHERE

seat\_id = (

-- 방금 종료시킨 예약 건의 좌석 ID를 조회

SELECT seat\_id

FROM seat\_reservations

WHERE usr\_id = 20233849 -- [변수] 사용자 ID

AND res\_stat = '반납완료'

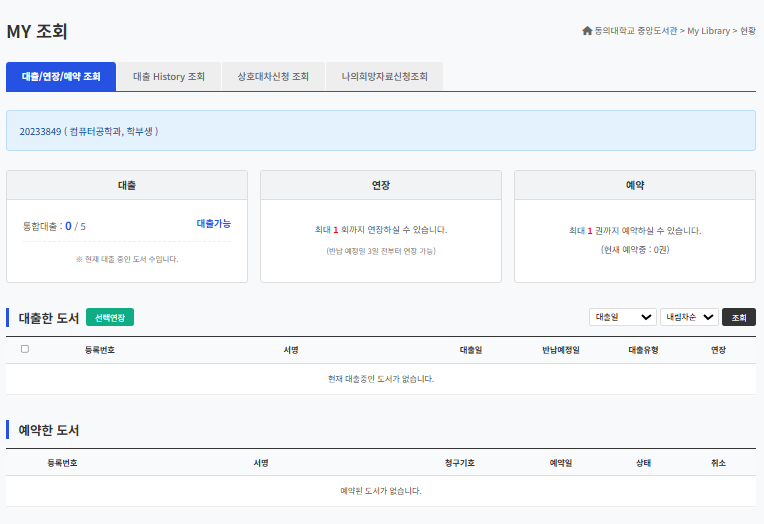
ORDER BY res\_end DESC

LIMIT 1

);

COMMIT;

### **학생 개인 조회(대출,연장,예약 조회) 화면**



### **학생 개인 조회(대출,연장,예약 조회) 쿼리**

USE library\_system;

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

-- 1. [헤더/대시보드] 사용자 요약 정보 조회

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

-- 1-1. 사용자 기본 정보 (이름, 학과, 상태)

SELECT

usr\_id,

usr\_name,

usr\_dept, -- 학과

usr\_stat -- 상태 (정상/정지/졸업)

FROM

users

WHERE

usr\_id = 20233849; -- [변수] 로그인한 학번

-- 1-2. 현재 대출 중인 권수

SELECT COUNT(\*) AS val\_loans

FROM loans

WHERE usr\_id = 20233849

AND loan\_ret IS NULL; -- 반납일이 없는 것 = 대출 중

-- 1-3. 현재 예약 중인 권수

SELECT COUNT(\*) AS val\_reservations

FROM reservations

WHERE usr\_id = 20233849

AND resv\_stat IN ('대기중', '수령가능'); -- [ENUM 체크] DDL 값 일치

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

-- 2. [리스트] 대출 현황 조회 (연장 기능 포함)

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

SELECT

L.loan\_id, -- (Hidden) 연장 처리용 ID

C.cbk\_id AS 등록번호, -- 도서 등록번호

B.bk\_title AS 서명, -- 책 제목

L.loan\_date AS 대출일,

L.loan\_due AS 반납예정일,

-- 연장 가능 여부 판단 (DB의 loan\_ext가 0/False여야 가능)

CASE

WHEN L.loan\_ext = 1 THEN '연장불가(횟수초과)'

WHEN L.loan\_due < CURDATE() THEN '연장불가(연체중)'

ELSE '연장가능'

END AS 연장상태,

L.loan\_ext -- (Hidden) 버튼 활성화/비활성화 로직용 True/False

FROM loans L

JOIN book\_copies C ON L.cbk\_id = C.cbk\_id

JOIN books B ON C.bk\_id = B.bk\_id

WHERE

L.usr\_id = 20233849 -- [변수] 학번

AND L.loan\_ret IS NULL -- 현재 대출 중인 도서만

ORDER BY

L.loan\_due ASC; -- 반납 급한 순 정렬

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

-- 3. [리스트] 예약 현황 조회 (취소 기능 포함)

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

SELECT

R.resv\_id, -- (Hidden) 취소 처리용 ID

-- 예약된 실물 도서 번호 (지정 예약인 경우 표시, 아니면 '-')

IFNULL(R.cbk\_id, '-') AS 예약도서번호,

B.bk\_title AS 서명,

B.bk\_id AS 청구기호,

R.resv\_date AS 예약일,

R.resv\_seq AS 대기순번,

R.resv\_stat AS 상태 -- '대기중', '수령가능'

FROM reservations R

JOIN books B ON R.bk\_id = B.bk\_id

WHERE

R.usr\_id = 20233849 -- [변수] 학번

AND R.resv\_stat IN ('대기중', '수령가능') -- [ENUM 체크] 진행 중인 건만 조회

ORDER BY

R.resv\_date DESC;

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

-- 4. [기능] 대출 연장 (UPDATE)

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

-- 조건: 본인 확인 + 연체 안 됨 + 연장 횟수 초과 안 함

UPDATE loans

SET

loan\_due = DATE\_ADD(loan\_due, INTERVAL 7 DAY), -- 7일 연장

loan\_ext = TRUE -- 연장 횟수 차감(1회 한정이면 True로 변경)

WHERE

loan\_id = 12345678 -- [변수] 대출 ID

AND usr\_id = 20233849 -- [변수] 학번 (본인확인)

AND loan\_due >= CURDATE() -- 연체된 책은 연장 불가

AND loan\_ext = FALSE; -- 이미 연장한 책은 불가 (1회 제한 정책)

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

-- 5. [기능] 예약 취소 (UPDATE)

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

-- 조건: '대기중'이나 '수령가능' 상태일 때만 취소 가능

UPDATE reservations

SET

resv\_stat = '취소됨' -- [ENUM 체크] '취소' 아님 -> '취소됨'

WHERE

resv\_id = 12345678 -- [변수] 예약 ID

AND usr\_id = 20233849 -- [변수] 학번

AND resv\_stat IN ('대기중', '수령가능');

### **학생 개인 조회(상호대차신청 조회) 화면**



### 

### **학생 개인 조회(상호대차신청 조회) 쿼리**

USE library\_system;

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

-- 1. [헤더] 사용자 기본 정보 조회

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

SELECT

usr\_name AS 이름,

usr\_id AS 학번,

usr\_dept AS 학과, -- [DDL] 학과 정보 표시

usr\_stat AS 상태 -- (정상/정지/졸업)

FROM

users

WHERE

usr\_id = 20233849; -- [변수] 로그인한 사용자 ID

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

-- 2. [리스트] 상호대차 신청 목록 조회

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

SELECT

ill\_id AS No,

ill\_title AS 서명,

ill\_auth AS 저자,

ill\_lib\_name AS 신청도서관, -- [DDL] 컬럼명 확인 (ill\_target\_lib 아님)

DATE\_FORMAT(ill\_req\_date, '%Y-%m-%d') AS 신청일,

ill\_stat AS 처리상태 -- (신청중/배송중/도착/반납완료/취소됨)

FROM

inter\_library\_loans

WHERE

usr\_id = 20233849 -- [변수] 로그인한 사용자 ID

ORDER BY

ill\_req\_date DESC

LIMIT 10 OFFSET 0; -- [페이징] 1페이지: 0, 2페이지: 10 ...

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

-- 3. [리스트] 총 신청 건수 조회 (페이징용)

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

SELECT

COUNT(\*) AS total\_count

FROM

inter\_library\_loans

WHERE

usr\_id = 20233849; -- [변수] 로그인한 사용자 ID

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

-- 4. [기능] 상호대차 신청 취소 (버튼 클릭 시)

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

-- 조건: 본인 신청 건이어야 하며, '신청중' 상태일 때만 취소 가능

UPDATE

inter\_library\_loans

SET

ill\_stat = '취소됨' -- [ENUM 체크] DDL 값 일치

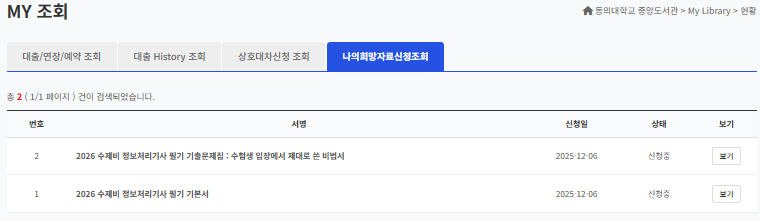
WHERE

ill\_id = 12345678 -- [변수] 취소할 신청 번호

AND usr\_id = 20233849 -- [변수] 로그인한 사용자 ID (보안)

AND ill\_stat = '신청중'; -- [중요] 배송 시작 전 단계에서만 취소 가능

### **학생 개인 조회(신청 자료 조회) 화면**



### **학생 개인 조회(신청 자료 조회) 쿼리**

USE library\_system;

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

-- 1. [헤더] 사용자 기본 정보 조회

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

SELECT

usr\_name,

usr\_dept, -- [DDL] 학과 정보

usr\_stat -- [DDL] 상태 (정상/정지)

FROM

users

WHERE

usr\_id = 20233849; -- [변수] 로그인한 사용자 ID

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

-- 2. [리스트] 희망도서 신청 목록 조회

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

SELECT

req\_id AS No,

req\_title AS 서명,

req\_auth AS 저자,

req\_pub AS 출판사, -- 화면에 출판사 정보도 주로 같이 표시됨

DATE\_FORMAT(req\_date, '%Y-%m-%d') AS 신청일,

req\_stat AS 처리상태 -- ('대기', '승인', '반려', '구매완료')

FROM

requests

WHERE

usr\_id = 20233849 -- [변수] 로그인한 사용자 ID

-- [선택] 검색 조건 (서명)

-- AND req\_title LIKE CONCAT('%', ?, '%')

ORDER BY

req\_date DESC

LIMIT 10 OFFSET 0; -- [페이징] 1페이지

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

-- 3. [리스트] 신청 총 건수 조회 (페이징용)

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

SELECT

COUNT(\*) AS total\_count

FROM

requests

WHERE

usr\_id = 20233849; -- [변수] 로그인한 사용자 ID

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

-- 4. [기능] 희망도서 신청 취소 (DELETE)

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

-- DDL에 '취소' 상태가 없으므로, 아직 '대기' 중인 건은 아예 삭제하는 것이 깔끔합니다.

DELETE FROM

requests

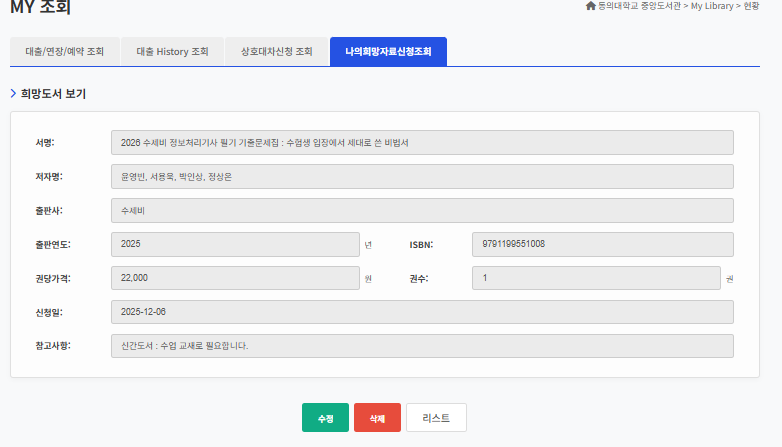
WHERE

req\_id = 12345678 -- [변수] 취소할 신청 번호

AND usr\_id = 20233849 -- [변수] 로그인한 사용자 ID (보안)

AND req\_stat = '대기'; -- [ENUM 체크] '접수대기' -> '대기'로 수정됨

### **학생 개인 조회(신청 자료 상세 페이지) 화면**



### **학생 개인 조회(신청 자료 상세 페이지) 쿼리**

USE library\_system;

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

-- 1. [헤더] 로그인 사용자 정보 조회

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

SELECT

usr\_name AS 이름,

usr\_id AS 학번

FROM

users

WHERE

usr\_id = 20233849; -- [변수] 로그인한 사용자 ID

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

-- 2. [상세화면] 신청 자료 상세 내용 조회 (READ)

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

SELECT

req\_title AS 서명,

req\_auth AS 저자,

req\_pub AS 출판사,

req\_year AS 출판연도,

req\_isbn AS ISBN,

req\_price AS 권당가격, -- [DDL] req\_price 컬럼 존재 확인

req\_qty AS 권수, -- [DDL] req\_qty 컬럼 존재 확인

DATE\_FORMAT(req\_date, '%Y-%m-%d') AS 신청일,

req\_memo AS 참고사항, -- [수정] req\_note -> req\_memo

req\_stat AS 처리상태 -- (화면 출력 X, 로직용)

FROM

requests

WHERE

req\_id = 12345678 -- [변수] 신청번호

AND usr\_id = 20233849; -- [변수] 학번

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

-- 3. [기능] 신청 정보 수정 (UPDATE)

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

-- 화면 하단 '수정' 버튼: '대기' 상태일 때만 비고(참고사항) 수정 가능

UPDATE

requests

SET

req\_memo = '오전 중에 도서관 방문하겠습니다.'

WHERE

req\_id = 12345678 -- [변수] 신청번호

AND usr\_id = 20233849 -- [변수] 학번

AND req\_stat = '대기'; -- [ENUM 체크] '접수대기' -> '대기'

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

-- 4. [기능] 신청 자료 삭제 (DELETE)

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

-- 화면 하단 '삭제' 버튼: 데이터를 완전히 삭제함

-- [주의] 이미 처리가 시작된(승인/구매완료) 건은 삭제되지 않도록 조건 유지

DELETE FROM

requests

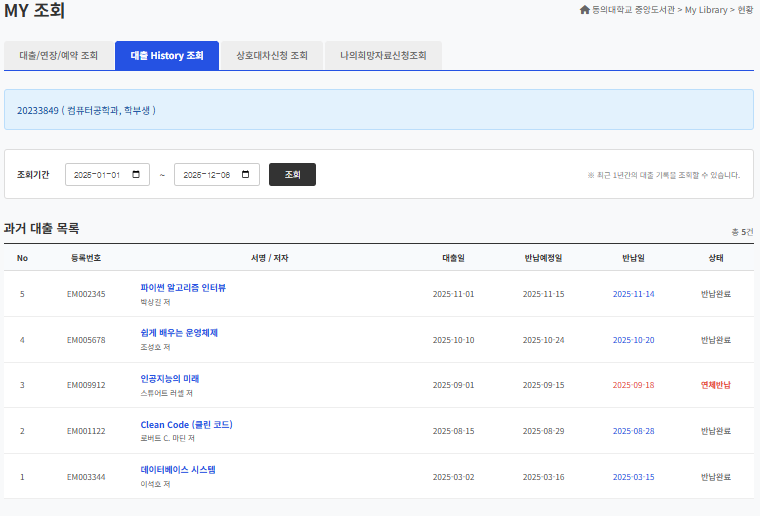
WHERE

req\_id = 12345678 -- [변수] 신청번호

AND usr\_id = 20233849 -- [변수] 학번 (본인 것만 삭제 가능)

AND req\_stat = '대기'; -- [ENUM 체크] 심사 전('대기')인 경우만 삭제 가능 [cite: 217, 218]

### **학생 개인 조회(대출 히스토리 조회) 화면**



### **학생 개인 조회(대출 히스토리 조회) 쿼리**

USE library\_system;

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

-- 0. [사전 작업] 필수 인덱스 생성

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

-- 대출 이력 조회 시 '특정 사용자'의 기록을 '날짜 역순'으로 빠르게 가져오기 위함

CREATE INDEX idx\_loans\_usr\_date

ON loans (usr\_id, loan\_date);

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

-- 1. [헤더] 사용자 기본 정보 조회

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

-- 화면 상단: "20233849 ( 컴퓨터공학과, 정상 )" 표시용

SELECT

usr\_name AS 이름,

usr\_dept AS 학과,

usr\_stat AS 상태

FROM

users

WHERE

usr\_id = '20233849'; -- [변수] 로그인한 사용자 ID

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

-- 2. [리스트] 대출 History 목록 조회

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

-- 조건: 로그인한 사용자의 '반납이 완료된(loan\_ret IS NOT NULL)' 기록만 조회

SELECT

T1.loan\_id AS No,

T2.cbk\_id AS 등록번호,

T3.bk\_title AS 서명,

T3.bk\_auth AS 저자,

DATE\_FORMAT(T1.loan\_date, '%Y-%m-%d') AS 대출일,

DATE\_FORMAT(T1.loan\_due, '%Y-%m-%d') AS 반납예정일,

DATE\_FORMAT(T1.loan\_ret, '%Y-%m-%d') AS 반납일,

-- [상태 계산] 반납일이 예정일보다 늦으면 '연체반납', 아니면 '반납완료'

CASE

WHEN T1.loan\_ret > T1.loan\_due THEN '연체반납'

ELSE '반납완료'

END AS 상태

FROM

loans T1

JOIN book\_copies T2 ON T1.cbk\_id = T2.cbk\_id

JOIN books T3 ON T2.bk\_id = T3.bk\_id

WHERE

T1.usr\_id = '20233849' -- [필수] 사용자 ID

AND T1.loan\_date BETWEEN '2025-01-01 00:00:00' AND '2025-12-06 23:59:59' -- [필터] 조회 기간

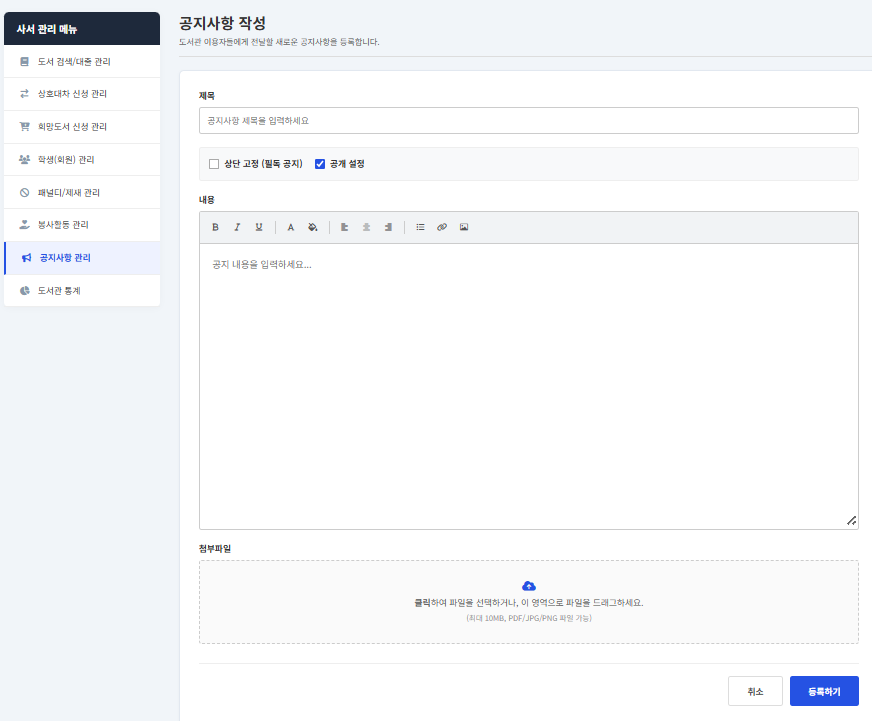
AND T1.loan\_ret IS NOT NULL -- [필터] 반납 완료된 건만 (History)

ORDER BY

T1.loan\_date DESC; -- [정렬] 최신순

# **3.2 사서 페이지**

### **사서 공지사항 작성 페이지 화면**



### 

### **사서 공지사항 작성 쿼리**

USE library\_system;

-- 작성자 정보 조회

SELECT

lib\_name

FROM

librarians

WHERE

lib\_id = '로그인한\_사서\_ID';

-- 작성일 자동 채움을 위한 현재 시각 조회

SELECT

NOW();

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

-- 공지사항 등록하기 버튼 클릭시

-- 새로운 공지사항을 저장하기 위한 INSERT문 필요

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

INSERT INTO librarians (lib\_id, lib\_pw, lib\_name, lib\_role)

VALUES (

'L1001',

'admin1234',

'관리자 김사서', -- 이름

'총무과' -- 부서

) ON DUPLICATE KEY UPDATE lib\_name = VALUES(lib\_name);

INSERT INTO notices (

lib\_id,

ntc\_title,

ntc\_content,

ntc\_date,

ntc\_important,

ntc\_views

)

VALUES (

'L1001', -- 유효한 사서 ID 사용 (예: L1001)

'입력된\_공지\_제목',

'입력된\_공지\_내용',

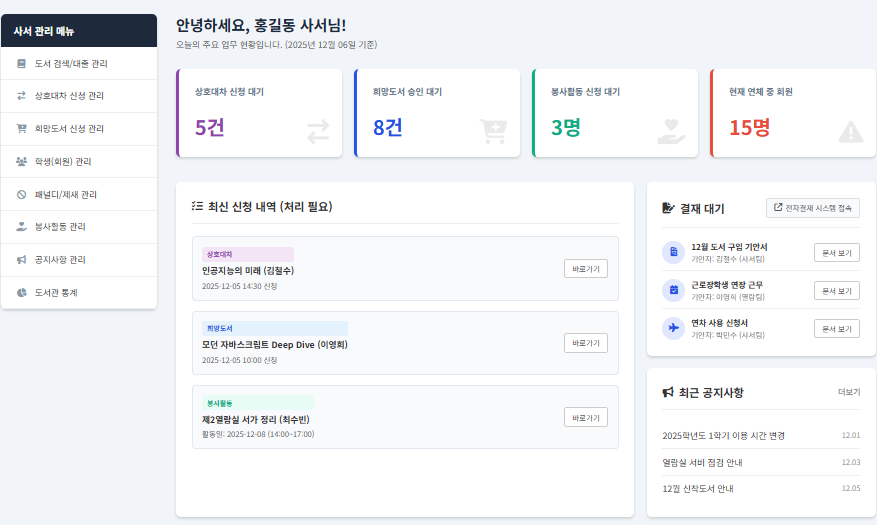
NOW(),

TRUE, -- TRUE/FALSE 또는 1/0 사용

0

);

### **사서 메인화면 페이지 화면**



### **사서 메인화면 쿼리**

USE library\_system;

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

-- 1. [헤더] 사서 이름 조회

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

SELECT lib\_name FROM librarians WHERE lib\_id = 'admin01'; -- [변수] 로그인 ID

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

-- 2. [상단 카드] 업무 현황 카운트 (ENUM 값 DDL 일치)

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

-- 2-1. 상호대차 신청 대기 (ILL)

SELECT COUNT(\*) AS cnt\_ill

FROM inter\_library\_loans WHERE ill\_stat = '신청중';

-- 2-2. 희망도서 승인 대기 (Request)

SELECT COUNT(\*) AS cnt\_req

FROM requests WHERE req\_stat = '대기';

-- 2-3. 봉사활동 신청 대기 (Volunteer)

SELECT COUNT(\*) AS cnt\_vol

FROM volunteer\_activities WHERE vol\_stat = '대기';

-- 2-4. 현재 연체 중인 '회원' 수

SELECT COUNT(DISTINCT usr\_id) AS cnt\_overdue\_users

FROM loans

WHERE loan\_ret IS NULL AND loan\_due < CURDATE();

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

-- 3. [메인] 미처리 업무 통합 리스트 (최신 신청 내역)

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

-- 서로 다른 3개 테이블(상호대차, 희망도서, 봉사)을 합쳐서

-- 날짜순(오래된 것부터)으로 정렬하여 '처리해야 할 일' 목록을 보여줌

SELECT

T.work\_type AS 업무구분, -- (상호대차/희망도서/봉사활동)

T.title AS 내용, -- (책 제목 or 활동 내용)

U.usr\_name AS 신청자명,

-- 화면 표시용 날짜 포맷 (예: 2025-12-05 14:30 신청)

DATE\_FORMAT(T.reg\_date, '%Y-%m-%d %H:%i') AS 신청일시,

-- [추가] 화면의 뱃지 색상 매핑용 코드 (bg-ill, bg-req, bg-vol)

CASE

WHEN T.work\_type = '상호대차' THEN 'bg-ill'

WHEN T.work\_type = '희망도서' THEN 'bg-req'

ELSE 'bg-vol'

END AS 뱃지\_클래스

FROM

(

-- 1. 상호대차 (신청중)

SELECT

'상호대차' AS work\_type,

ill\_title AS title,

ill\_req\_date AS reg\_date, -- 이미 DATETIME

usr\_id

FROM inter\_library\_loans

WHERE ill\_stat = '신청중'

UNION ALL

-- 2. 희망도서 (대기)

SELECT

'희망도서' AS work\_type,

req\_title AS title,

-- [수정] DATE 타입을 DATETIME으로 변환 (시간은 00:00:00으로)

CAST(CONCAT(req\_date, ' 00:00:00') AS DATETIME) AS reg\_date,

usr\_id

FROM requests

WHERE req\_stat = '대기'

UNION ALL

-- 3. 봉사활동 (대기)

SELECT

'봉사활동' AS work\_type,

vol\_desc AS title,

-- [수정] DATE 타입을 DATETIME으로 변환

CAST(CONCAT(vol\_date, ' 00:00:00') AS DATETIME) AS reg\_date,

usr\_id

FROM volunteer\_activities

WHERE vol\_stat = '대기'

) AS T

JOIN

users U ON T.usr\_id = U.usr\_id

ORDER BY

T.reg\_date ASC -- 오래된 요청부터 처리하도록 오름차순 정렬

LIMIT 5;

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

-- 4. [우측 하단] 최근 공지사항 (미니 리스트)

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

SELECT

ntc\_title,

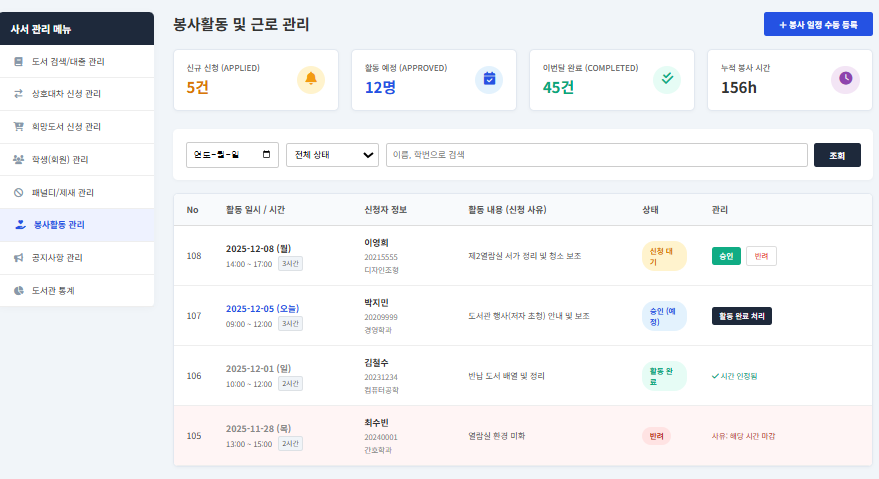
DATE\_FORMAT(ntc\_date, '%m.%d') AS ntc\_date\_fmt

FROM notices

ORDER BY ntc\_date DESC

LIMIT 3;

### **사서 봉사활동 관리 페이지 화면**



### **사서 봉사활동 관리 쿼리**

USE library\_system;

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

-- 1. [상단 통계] 신청 상태별 건수 조회 (Dashboard)

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

SELECT

-- 1) 신규 신청 (PENDING -> 대기)

(SELECT COUNT(\*) FROM requests WHERE req\_stat = '대기') AS 신규\_신청,

-- 2) 구매 진행중 (APPROVED -> 승인 or 구매완료)

(SELECT COUNT(\*) FROM requests WHERE req\_stat IN ('승인', '구매완료')) AS 구매\_진행중,

-- 3) 반려 (REJECTED -> 반려)

(SELECT COUNT(\*) FROM requests WHERE req\_stat = '반려') AS 반려\_건수,

-- 4) 전체 누적 신청

(SELECT COUNT(\*) FROM requests) AS 전체\_누적\_신청;

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

-- 2. [리스트] 희망도서 신청 목록 조회 (검색/필터)

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

SELECT

T1.req\_id AS No,

DATE\_FORMAT(T1.req\_date, '%Y-%m-%d') AS 신청일,

-- 신청자 정보

T2.usr\_name AS 신청자명,

T2.usr\_id AS 학번,

T2.usr\_dept AS 학과,

-- 도서 정보

T1.req\_title AS 서명,

T1.req\_auth AS 저자,

T1.req\_pub AS 출판사,

T1.req\_isbn AS ISBN,

-- 상태 및 관리 정보

T1.req\_stat AS 상태, -- (대기/승인/반려/구매완료)

T1.req\_memo AS 신청사유, -- (학생이 쓴 글)

T1.req\_proc\_memo AS 반려사유 -- (관리자가 쓴 글 - 화면 표시용)

FROM

requests T1

JOIN

users T2 ON T1.usr\_id = T2.usr\_id

WHERE

1=1

-- [필터 1] 상태 필터 (전체/대기/승인/반려)

AND T1.req\_stat = '대기' -- [변수] 선택된 값이 있을 때만 적용

-- [필터 2] 통합 검색 (도서명, 저자, 학번)

AND (

T1.req\_title LIKE CONCAT('%', '데이터 사이언스 입문', '%')

OR T1.req\_auth LIKE CONCAT('%', '김진, 최정아', '%')

OR T2.usr\_id LIKE CONCAT('%', '20233849', '%')

)

ORDER BY

-- 대기중인 건을 최상단으로, 그 다음 최신순 정렬

CASE WHEN T1.req\_stat = '대기' THEN 1 ELSE 2 END ASC,

T1.req\_date DESC;

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

-- 3. [액션] 관리자 승인/반려 처리 (UPDATE)

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

-- [버튼: 승인] 검토 대기 -> 승인 (구매 절차 시작)

UPDATE requests

SET

req\_stat = '승인',

req\_proc\_memo = NULL -- 반려 사유 초기화

WHERE

req\_id = 1; -- [변수] 신청 번호

-- [버튼: 반려] 검토 대기 -> 반려 (사유 입력 필수)

UPDATE requests

SET

req\_stat = '반려',

req\_proc\_memo = '예산 초과' -- [수정됨] 예: '학습 무관 도서', '예산 초과'

WHERE

req\_id = 1; -- [변수] 신청 번호

-- [버튼: 입고 완료/구매 완료] 승인 -> 구매완료 (최종 처리)

UPDATE requests

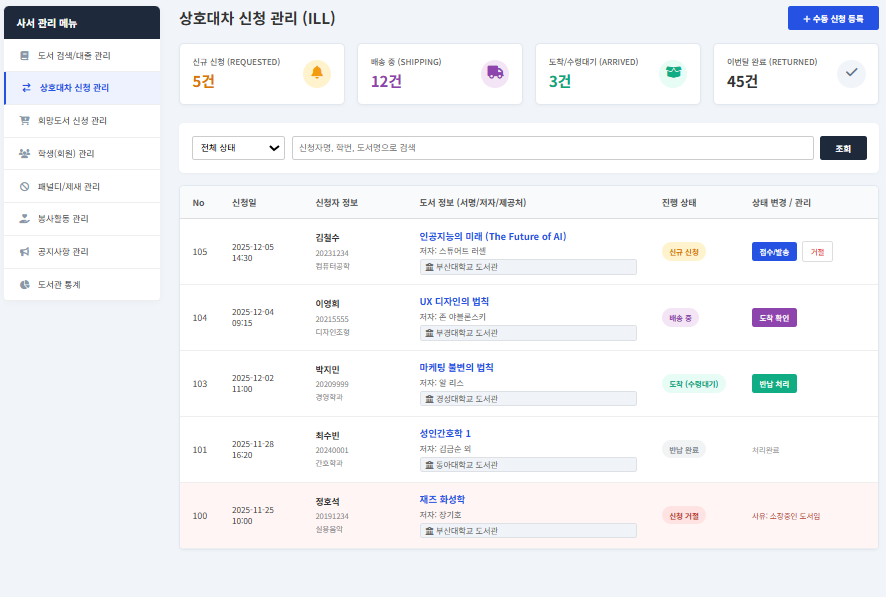
SET

req\_stat = '구매완료'

WHERE

req\_id = 1 AND req\_stat = '승인';

**사서 상호대차 관리 페이지 화면**



### **사서 상호대차 관리 쿼리**

USE library\_system;

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

-- 1. [상단 통계] 상태별 신청 건수 조회

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

SELECT

(SELECT COUNT(\*) FROM inter\_library\_loans WHERE ill\_stat = '신청중') AS 신규신청\_건수,

(SELECT COUNT(\*) FROM inter\_library\_loans WHERE ill\_stat = '배송중') AS 배송중\_건수,

(SELECT COUNT(\*) FROM inter\_library\_loans WHERE ill\_stat = '도착') AS 도착\_수령대기\_건수,

-- 이번달 완료 건수 (반납완료 상태이면서, 신청일이 이번 달인 경우로 근사 계산)

(SELECT COUNT(\*) FROM inter\_library\_loans

WHERE ill\_stat = '반납완료'

AND ill\_req\_date BETWEEN DATE\_FORMAT(NOW(), '%Y-%m-01') AND LAST\_DAY(NOW())

) AS 이번달\_완료\_건수;

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

-- 2. [리스트] 상호대차 신청 목록 조회 (검색/필터)

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

SELECT

T1.ill\_id AS No,

DATE\_FORMAT(T1.ill\_req\_date, '%Y-%m-%d %H:%i') AS 신청일시,

-- 신청자 정보

T2.usr\_name AS 신청자명,

T2.usr\_id AS 학번,

T2.usr\_dept AS 학과,

-- 도서 정보

T1.ill\_title AS 서명,

T1.ill\_auth AS 저자,

T1.ill\_lib\_name AS 제공\_도서관, -- [수정] 컬럼명 일치 (ill\_target\_lib -> ill\_lib\_name)

-- 상태 및 관리 정보

T1.ill\_stat AS 상태, -- (신청중/배송중/도착/반납완료/취소됨)

T1.ill\_note AS 신청자메모, -- 학생이 쓴 비고

T1.ill\_ans\_note AS 관리자메모 -- 사서가 쓴 비고/거절사유 [추가됨]

FROM

inter\_library\_loans T1

JOIN

users T2 ON T1.usr\_id = T2.usr\_id

WHERE

1=1

-- [필터 1] 상태 필터 (전체/신청중/배송중/...)

AND T1.ill\_stat = '신청중' -- [변수] 선택된 값이 있을 때만 적용

-- [필터 2] 통합 검색 (신청자명, 학번, 도서명)

AND (

T2.usr\_name LIKE CONCAT('%', '조기강', '%')

OR T2.usr\_id LIKE CONCAT('%', 20233849, '%')

OR T1.ill\_title LIKE CONCAT('%', '데이터 사이언스 입문', '%')

)

ORDER BY

-- '신청중'인 건을 최우선으로 보여주고, 그 외는 최신순

CASE WHEN T1.ill\_stat = '신청중' THEN 1 ELSE 2 END ASC,

T1.ill\_req\_date DESC;

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

-- 3. [액션] 상태 변경 워크플로우 (UPDATE)

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

-- [버튼: 접수/발송] 신청중 -> 배송중

-- 학생의 신청을 확인하고 타 대학 도서관에 요청을 보냈을 때

UPDATE inter\_library\_loans

SET

ill\_stat = '배송중',

ill\_ans\_note = NULL -- 기존 거절사유 등 초기화

WHERE

ill\_id = 1; -- [변수] 신청 번호

-- [버튼: 도착 확인] 배송중 -> 도착

-- 타 대학에서 책이 도착하여 학생에게 "수령하러 오세요" 알림을 보낼 때

UPDATE inter\_library\_loans

SET

ill\_stat = '도착'

WHERE

ill\_id = 1;

-- [버튼: 반납 처리] 도착 -> 반납완료

-- 학생이 책을 다 보고 반납하여, 타 대학으로 반송 처리했을 때

UPDATE inter\_library\_loans

SET

ill\_stat = '반납완료'

WHERE

ill\_id = 1;

-- [버튼: 거절] 신청중 -> 취소됨 (사유 입력 필수)

-- 우리 학교에 이미 있거나 대출 불가능한 경우

UPDATE inter\_library\_loans

SET

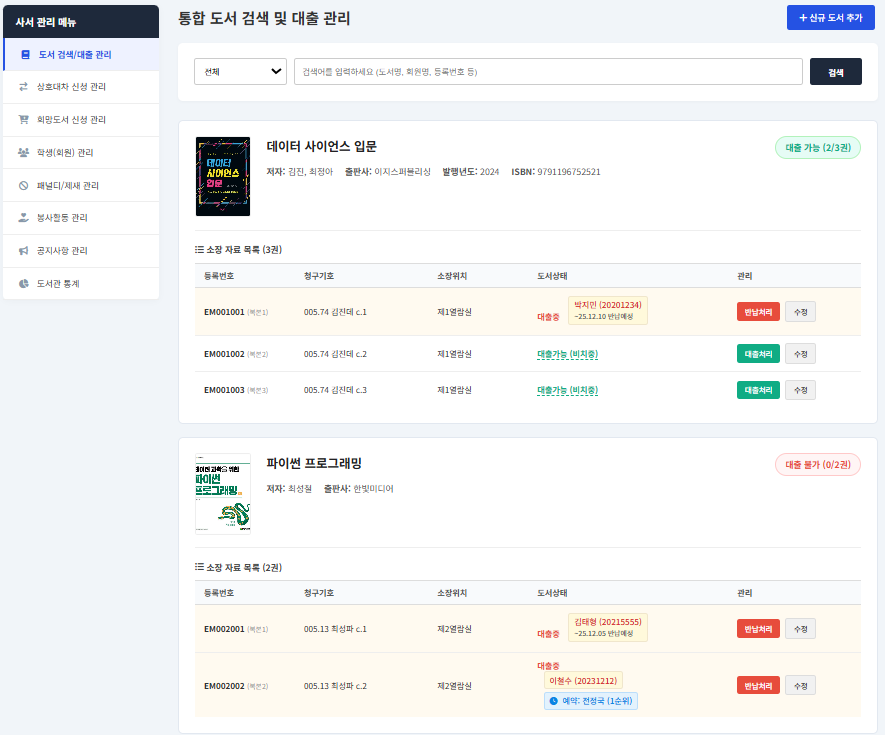
ill\_stat = '취소됨',

ill\_ans\_note = '본교 소장 도서임' -- [변수] 거절 사유 (예: "본교 소장 도서임")

WHERE

ill\_id = 1;

### **사서 책 검색 페이지 화면**



### 

### **사서 책 검색 쿼리**

USE library\_system;

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

-- 1. 통합 도서 검색 조회 (복본 단위 조회)

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

SELECT

-- [도서 마스터 정보]

T2.bk\_id AS 청구기호\_ID, -- 그룹핑 기준 키

T2.bk\_title AS 서명,

T2.bk\_auth AS 저자,

T2.bk\_pub AS 출판사,

T2.bk\_year AS 발행연도,

T2.bk\_isbn AS ISBN,

T2.bk\_image AS 표지이미지, -- [화면 표시용]

-- [도서 상태 뱃지용 집계 (서브쿼리)]

(SELECT COUNT(\*) FROM book\_copies WHERE bk\_id = T2.bk\_id) AS 총\_소장권수,

(SELECT COUNT(\*) FROM book\_copies WHERE bk\_id = T2.bk\_id AND cbk\_stat = '대출가능') AS 대출가능권수,

-- [복본 상세 정보]

T1.cbk\_id AS 등록번호,

CONCAT(T2.bk\_id, ' c.', T1.cbk\_vol) AS 청구기호\_조합, -- 예: 005.74 김진데 c.1

T1.cbk\_vol AS 복본번호,

T3.loc\_name AS 소장위치,

T1.cbk\_stat AS 도서상태, -- (대출가능, 대출중, 분실, 파손)

-- [대출 정보 (현재 대출중인 경우만)]

T5.usr\_name AS 대출자\_이름,

T5.usr\_id AS 대출자\_학번,

DATE\_FORMAT(T4.loan\_due, '%Y-%m-%d') AS 반납예정일,

-- [예약 정보 (1순위 예약자 표시)]

(SELECT CONCAT(U.usr\_name, ' (', U.usr\_id, ')')

FROM reservations R

JOIN users U ON R.usr\_id = U.usr\_id

WHERE R.cbk\_id = T1.cbk\_id

AND R.resv\_stat = '대기중'

ORDER BY R.resv\_seq ASC

LIMIT 1

) AS 예약자\_정보

FROM

book\_copies T1

JOIN

books T2 ON T1.bk\_id = T2.bk\_id

LEFT JOIN

locations T3 ON T1.loc\_id = T3.loc\_id

LEFT JOIN

loans T4 ON T1.cbk\_id = T4.cbk\_id AND T4.loan\_ret IS NULL -- 현재 대출중인 건 조인

LEFT JOIN

users T5 ON T4.usr\_id = T5.usr\_id

WHERE

1=1

/\* [검색 조건: UI Select Box에 따라 동적 적용] \*/

-- [전체 검색인 경우]

AND (

T2.bk\_title LIKE CONCAT('%', '데이터 사이언스 입문', '%') -- 서명

OR T2.bk\_auth LIKE CONCAT('%', '김진, 최정아', '%') -- 저자

OR T1.cbk\_id LIKE CONCAT('%', '9791196752521', '%') -- 등록번호

OR T5.usr\_name LIKE CONCAT('%', '박지민', '%') -- 대출자명

)

/\* [개별 필터링 예시]

-- AND T2.bk\_title LIKE ...

-- AND T1.cbk\_id = ...

\*/

ORDER BY

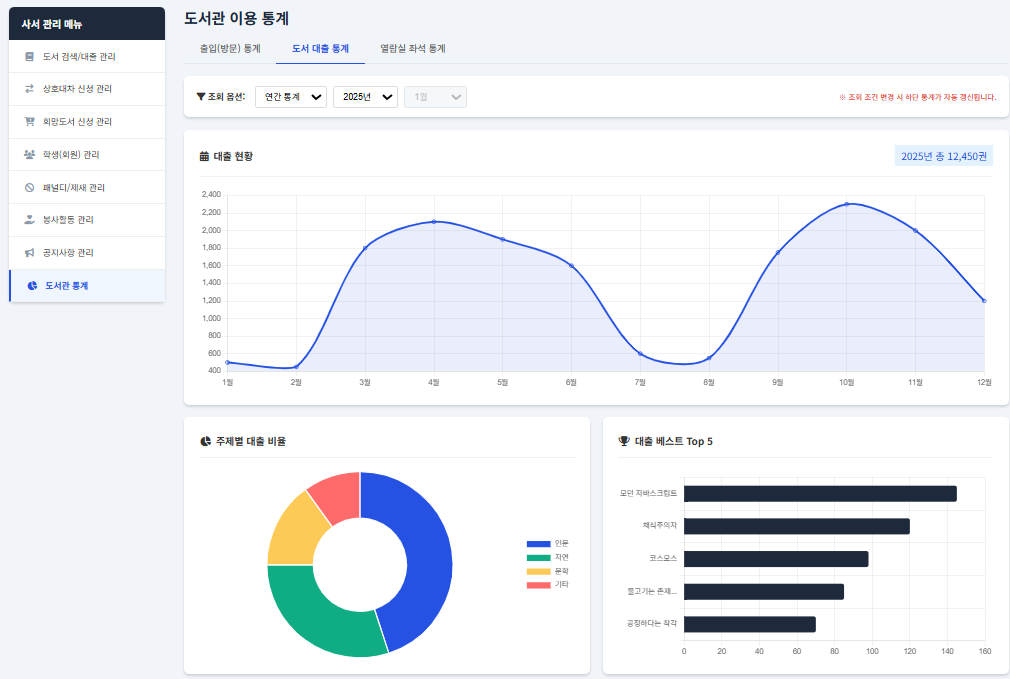
T2.bk\_title ASC, -- 서명순 정렬

T1.cbk\_vol ASC -- 복본번호순 정렬 (c.1, c.2 ...)

LIMIT 50;

### 

### **사서 통계(도서 대출 통계) 화면**



### **사서 통계(도서 대출 통계) 쿼리**

USE library\_system;

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

-- 1. [메인 차트] 전체 도서 월별 대출 추이 (Line Chart)

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

-- 화면 상단 '대출 현황' 그래프 및 '총 대출 뱃지' 데이터

SELECT

stat\_month AS 월,

SUM(loan\_cnt) AS 전체\_대출\_권수

FROM

monthly\_book\_stats

WHERE

stat\_year = 2025 -- [변수] UI에서 선택한 연도

GROUP BY

stat\_month

ORDER BY

stat\_month ASC;

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

-- 2. [서브 차트] 주제(장르)별 대출 비율 (Doughnut Chart)

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

-- 화면 중단 좌측 '주제별 대출 비율' 그래프

SELECT

G.gnr\_name AS 장르명,

SUM(S.loan\_cnt) AS 대출\_권수,

-- (옵션) 비율 계산은 보통 프론트엔드(Chart.js)에서 처리하지만, 필요시 SQL로 계산 가능

ROUND(SUM(S.loan\_cnt) \* 100.0 / (SELECT SUM(loan\_cnt) FROM monthly\_book\_stats WHERE stat\_year = 2025), 1) AS 비율\_퍼센트

FROM

monthly\_book\_stats S

JOIN

books B ON S.bk\_id = B.bk\_id

JOIN

genres G ON B.gnr\_id = G.gnr\_id

WHERE

S.stat\_year = 2025 -- [변수] UI에서 선택한 연도

GROUP BY

G.gnr\_name

ORDER BY

대출\_권수 DESC;

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

-- 3. [서브 차트] 대출 베스트 Top 5 (Bar Chart)

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

-- 화면 중단 우측 '대출 베스트 Top 5' 그래프

SELECT

B.bk\_title AS 서명,

SUM(S.loan\_cnt) AS 총\_대출수

FROM

monthly\_book\_stats S

JOIN

books B ON S.bk\_id = B.bk\_id

WHERE

S.stat\_year = 2025 -- [변수] UI에서 선택한 연도

GROUP BY

S.bk\_id, B.bk\_title

ORDER BY

총\_대출수 DESC

LIMIT 5;

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

-- 4. [검색] 도서 기본 정보 및 총 누적 통계 조회

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

-- 화면 하단 검색창 입력 후 '검색' 버튼 클릭 시 실행 (카드 좌측 정보)

SELECT

B.bk\_title AS 서명,

B.bk\_auth AS 저자,

B.bk\_pub AS 출판사,

-- [수정됨] 전체 누적이 아닌, '선택한 연도(2025)'의 대출 수만 합산

(SELECT IFNULL(SUM(loan\_cnt), 0)

FROM monthly\_book\_stats

WHERE bk\_id = B.bk\_id

AND stat\_year = 2025 -- [변수] UI 선택 연도와 동일하게 필터링

) AS 선택연도\_총\_대출수

FROM

books B

WHERE

B.bk\_title LIKE CONCAT('%', '검색어', '%') -- [변수] 예: '코스모스'

LIMIT 1;

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

-- 5. [검색] 개별 도서 월별 대출 추이 (Mini Bar Chart)

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

-- 화면 하단 검색 결과 카드 우측의 '미니 차트' 데이터

SELECT

S.stat\_month AS 월,

S.loan\_cnt AS 대출\_횟수

FROM

monthly\_book\_stats S

JOIN

books B ON S.bk\_id = B.bk\_id

WHERE

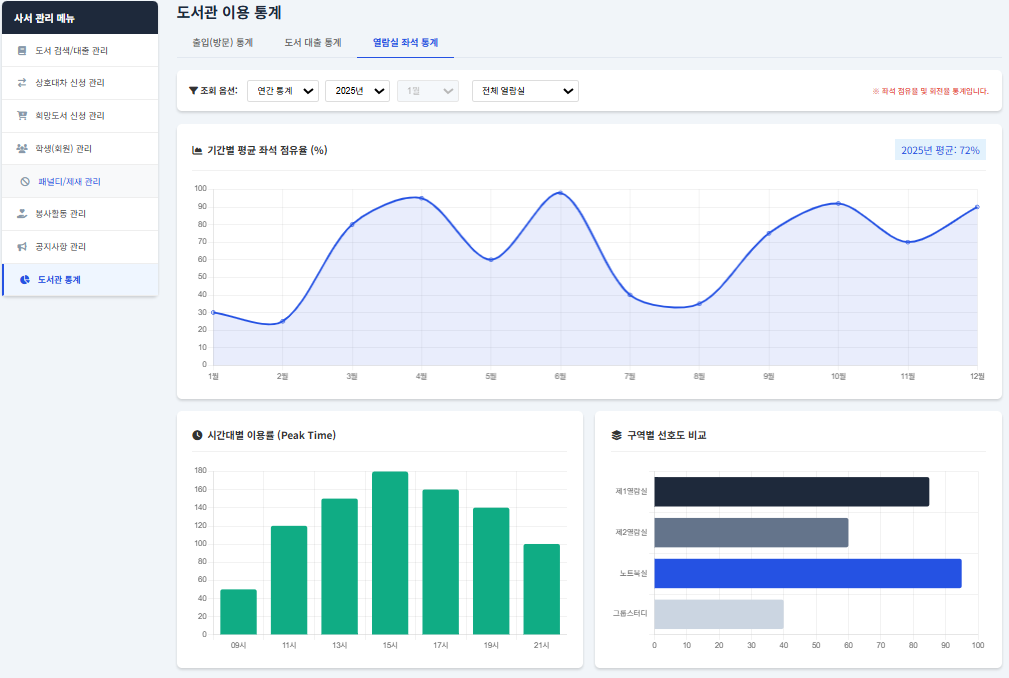
B.bk\_title LIKE CONCAT('%', '검색어', '%') -- [변수] 위와 동일한 검색어

AND S.stat\_year = 2025 -- [변수] 선택한 연도

ORDER BY

S.stat\_month ASC;

### **사서 통계(좌석 통계) 화면**



### **사서 통계(좌석 통계) 쿼리**

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

-- 1. [차트] 기간별 방문자 추이 (Line Chart)

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

-- UI 옵션(연간/월간)에 따라 쿼리 분기 필요

-- [Case A] 연간 통계 (월별 집계)

SELECT

MONTH(log\_time) AS 월,

COUNT(\*) AS 방문자수

FROM

entry\_logs

WHERE

log\_type = '입실' -- [수정] DDL에 맞춘 한글 값

AND YEAR(log\_time) = '2025' -- [변수] 선택한 연도 (예: 2025)

GROUP BY

MONTH(log\_time)

ORDER BY

월 ASC;

-- [Case B] 월간 통계 (일별 집계)

SELECT

DAY(log\_time) AS 일,

COUNT(\*) AS 방문자수

FROM

entry\_logs

WHERE

log\_type = '입실'

AND YEAR(log\_time) = '2025' -- [변수] 연도

AND MONTH(log\_time) = '10' -- [변수] 월

GROUP BY

DAY(log\_time)

ORDER BY

일 ASC;

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

-- 2. [차트] 방문왕 랭킹 Top 10 (Bar Chart)

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

SELECT

U.usr\_name AS 이름,

U.usr\_dept AS 학과,

COUNT(E.log\_id) AS 방문횟수

FROM

entry\_logs E

JOIN

users U ON E.usr\_id = U.usr\_id

WHERE

E.log\_type = '입실'

AND YEAR(E.log\_time) = '2025' -- [변수] 선택한 연도 (랭킹 범위 제한)

GROUP BY

U.usr\_id, U.usr\_name, U.usr\_dept

ORDER BY

방문횟수 DESC

LIMIT 10;

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

-- 3. [검색] 개별 학생 방문 기록 조회 (우측 카드)

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

-- 화면 표시 항목: 이름, 학번, 학과, [선택 기간 방문 횟수], [총 누적 시간]

SELECT

U.usr\_name AS 이름,

U.usr\_id AS 학번,

U.usr\_dept AS 학과,

-- 1) 선택 기간 방문 횟수

(SELECT COUNT(\*) FROM entry\_logs

WHERE usr\_id = U.usr\_id AND log\_type = '입실' AND YEAR(log\_time) = 2025) AS 기간\_방문횟수,

-- 2) 총 누적 체류 시간 (입실/퇴실 짝이 맞는 경우만 계산)

-- (주의: 대용량 로그에서 이 계산은 부하가 클 수 있으므로, 별도 통계 테이블 권장.

-- 여기서는 실시간 계산 로직 예시입니다.)

(

SELECT

ROUND(SUM(TIMESTAMPDIFF(MINUTE, E\_IN.log\_time, E\_OUT.log\_time)) / 60, 1)

FROM

entry\_logs E\_IN

JOIN

entry\_logs E\_OUT ON E\_IN.usr\_id = E\_OUT.usr\_id

AND E\_OUT.log\_time > E\_IN.log\_time

-- 입실 바로 다음의 퇴실 기록을 찾음 (간단화된 로직)

WHERE

E\_IN.usr\_id = U.usr\_id

AND E\_IN.log\_type = '입실'

AND E\_OUT.log\_type = '퇴실'

AND DATE(E\_IN.log\_time) = DATE(E\_OUT.log\_time) -- 같은 날짜 가정

) AS 총\_누적시간\_Hour

FROM

users U

WHERE

U.usr\_id = 20233849 OR U.usr\_name = '조기강'; -- [변수] 검색어 (학번 또는 이름)

### **사서 통계(챗봇 통계) 화면**



### **사서 통계(챗봇 통계) 쿼리**

INSERT INTO ai\_analysis\_log (

lib\_id,

quest\_text,

gen\_sql,

chart\_type,

error\_msg,

exec\_time

) VALUES (

'admin01', -- 로그인한 사서 ID

'컴퓨터공학과에서 가장 많이 빌린 책 Top 5', -- 자연어 질문

'SELECT

T4.bk\_title AS 서명,

T4.bk\_auth AS 저자,

COUNT(T1.loan\_id) AS 대출횟수

FROM loans T1

JOIN users T2 ON T1.usr\_id = T2.usr\_id

JOIN book\_copies T3 ON T1.cbk\_id = T3.cbk\_id

JOIN books T4 ON T3.bk\_id = T4.bk\_id

WHERE T2.usr\_dept = "컴퓨터공학과"

GROUP BY T4.bk\_id, T4.bk\_title, T4.bk\_auth

ORDER BY 대출횟수 DESC

LIMIT 5;',

'bar', -- 차트 타입

NULL, -- 에러 메시지 없음

120 -- 실행 시간 (ms)

);

/\*

챗봇 화면 만든 의도

기존 통계 화면은 개발자가 만들어준 툴 안에서 통계를 보는대 그치거나 직접 db쿼리문을 치면서 확인해야한다는 불편함이 있어서 추가했습니다

모든 사서가 DB 쿼리문에 능하지 않고 전산 직원과 연락을 주고 받으면서 불편하게 자신이 원하는 통계를 뽑아내는대 불편함이 있다고 판단하여

간단하게 도서관 사서 쪽에서 챗봇이 만들어준 통계를 뽑아서 팀장,관장 에게 결재를 받아 좋은 아이디어다 싶으면 전산직원에게 쿼리문이랑 질의를 전달하여 빠르게 개발하도록 의도를 가졌습니다.

ex) 입구 앞 키오스크 화면에 홍보에 활용

ex) 문과생들이 많이 대출하는도서 혹은 문이과들 생들의 대출 도서 차이

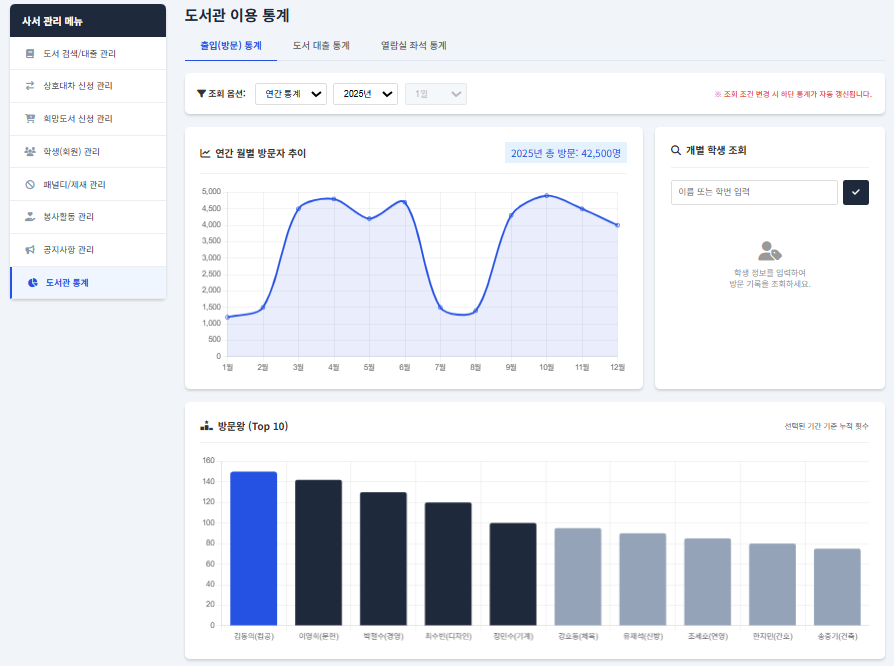
등등 사서가 아이디어가 떠오르면 바로 데이터 시각화 된 화면을 보고

쿼리문을 저장할수 있는게 핵심입니다.

\*/

### 

### **사서 통계(출입 방문) 화면**



### **사서 통계(출입 방문) 쿼리**

USE library\_system;

-- 기간별 총 방문자 수 조회

-- (최근 6개월 동안의 월별 총 방문자 수)

SELECT

YEAR(log\_time) AS log\_year,

MONTH(log\_time) AS log\_month,

COUNT(log\_id) AS total\_visits

FROM

entry\_logs

WHERE

log\_type = '입실' -- 도서관 입실 기록만 필터링

AND log\_time >= DATE\_SUB(CURDATE(), INTERVAL 6 MONTH)

GROUP BY

log\_year, log\_month

ORDER BY

log\_year ASC, log\_month ASC;

-- 시간대별 방문자 수 현황 조회

SELECT

HOUR(log\_time) AS log\_hour,

COUNT(\*) AS total\_entries

FROM

entry\_logs

WHERE

log\_type = '입실' -- 도서관 입실 기록만 필터링

AND log\_time >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH) -- 예시: 최근 1주간 데이터

GROUP BY

log\_hour

ORDER BY

log\_hour ASC;

-- 방문자 랭킹 조회

SELECT

T2.usr\_name AS 사용자\_이름,

COUNT(T1.log\_id) AS 방문\_횟수,

T2.usr\_dept

FROM

entry\_logs T1

JOIN

users T2 ON T1.usr\_id = T2.usr\_id

WHERE

T1.log\_type = '입실' -- 도서관 입실 기록만 필터링

AND T1.log\_time >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH) -- 예시: 최근 1개월 데이터 기준

GROUP BY

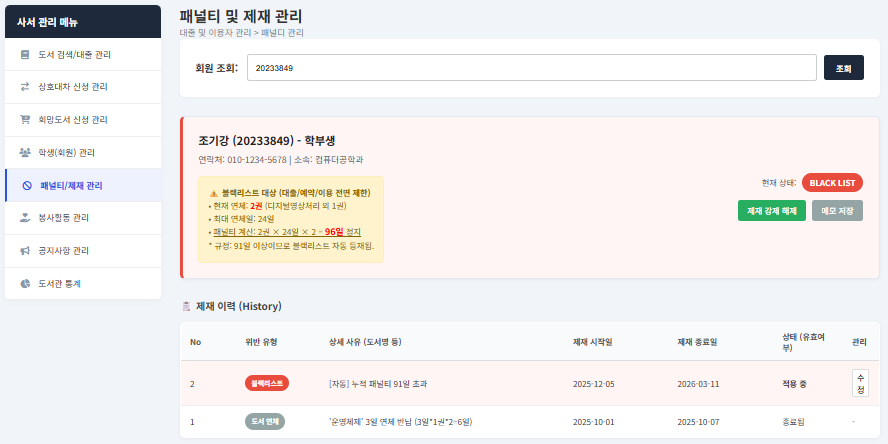
T1.usr\_id, T2.usr\_name

ORDER BY

방문\_횟수 DESC

LIMIT 10; -- 예시: Top 10 방문자

**사서 패널티 관리 페이지 화면**



### **사서 패널티 관리 쿼리**

USE library\_system;

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

-- 1. [회원 조회] 기본 정보 및 현재 상태 (블랙리스트 여부)

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

-- 화면 상단 '조기강 (20233849)' 카드 정보

SELECT

T1.usr\_id AS 학번,

T1.usr\_name AS 이름,

T1.usr\_dept AS 학과,

T1.usr\_phone AS 연락처,

T1.usr\_stat AS 회원상태, -- (정상/정지/졸업)

-- 현재 적용 중인 제재가 있는지 확인 (화면의 'BLACK LIST' 뱃지용)

(SELECT EXISTS(

SELECT 1 FROM penalties

WHERE usr\_id = T1.usr\_id AND pnl\_actv = 1 AND pnl\_end >= CURDATE()

)) AS 제재\_적용\_여부,

-- 현재 연체 중인 도서 권수 (화면: '2권')

(SELECT COUNT(\*) FROM loans

WHERE usr\_id = T1.usr\_id AND loan\_ret IS NULL AND loan\_due < CURDATE()) AS 연체\_도서\_수

FROM

users T1

WHERE

T1.usr\_id = 20233849; -- [변수] 검색창 입력값 (학번)

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

-- 2. [계산] 실시간 패널티 시뮬레이션 (Calc Box)

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

-- 화면 중간 '패널티 계산' 박스에 들어갈 데이터

SELECT

T3.bk\_title AS 연체\_도서명,

T1.loan\_due AS 반납\_예정일,

DATEDIFF(CURDATE(), T1.loan\_due) AS 연체일수, -- (오늘 - 예정일)

-- 예상 정지 일수 계산 (규정: 연체일수 \* 1일 정지 라고 가정 시)

-- 화면 로직(2배)에 맞춘다면 \* 2

(DATEDIFF(CURDATE(), T1.loan\_due) \* 1) AS 예상\_정지일

FROM

loans T1

JOIN

book\_copies T2 ON T1.cbk\_id = T2.cbk\_id

JOIN

books T3 ON T2.bk\_id = T3.bk\_id

WHERE

T1.usr\_id = 20233849 -- [변수] 학번

AND T1.loan\_ret IS NULL -- 미반납

AND T1.loan\_due < CURDATE(); -- 연체됨

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

-- 3. [리스트] 제재 이력 조회 (History Table)

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

-- 화면 하단 테이블 데이터

SELECT

pnl\_id AS No,

pnl\_type AS 위반\_유형, -- (연체, 분실, 파손 등)

pnl\_reason AS 상세\_사유, -- (도서명 등)

DATE\_FORMAT(pnl\_start, '%Y-%m-%d') AS 시작일,

DATE\_FORMAT(pnl\_end, '%Y-%m-%d') AS 종료일,

-- 상태 표시 로직

CASE

WHEN pnl\_actv = 1 AND pnl\_end >= CURDATE() THEN '적용 중'

ELSE '종료됨'

END AS 상태

FROM

penalties

WHERE

usr\_id = 20233849 -- [변수] 학번

ORDER BY

pnl\_start DESC;

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

-- 4. [액션] 제재 강제 해제 (버튼 클릭)

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

-- 화면의 '제재 강제 해제' 버튼 클릭 시 실행

-- 1) 패널티 테이블 비활성화 처리

UPDATE penalties

SET

pnl\_actv = 0, -- 비활성화

pnl\_reason = CONCAT(pnl\_reason, ' / (사서 강제 해제)') -- 사유 기록

WHERE

usr\_id = 20233849 -- [변수] 학번

AND pnl\_actv = 1

AND pnl\_end >= CURDATE();

-- 2) 유저 상태 '정상'으로 복구

UPDATE users

SET usr\_stat = '정상'

WHERE usr\_id = 20233849;

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

-- 5. [액션] 제재 이력 수정 (테이블 내 수정 버튼)

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

-- 하단 리스트의 작은 '수정' 버튼 클릭 시 (종료일 변경 등)

UPDATE penalties

SET

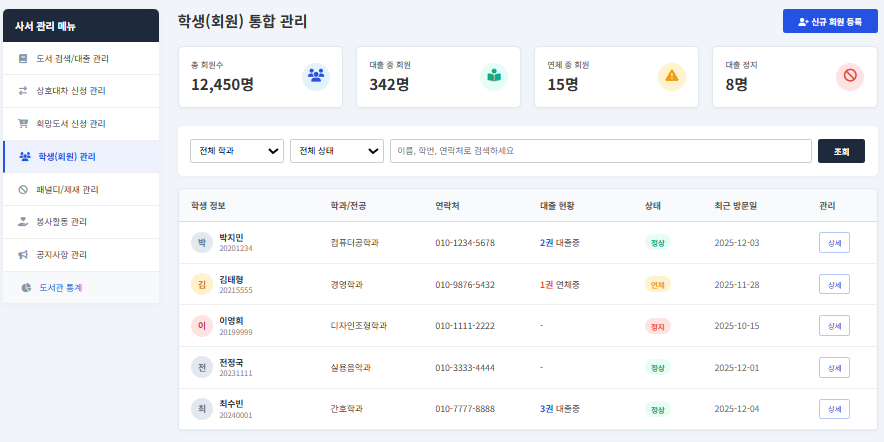
pnl\_end = '2026-03-31', -- [변수] 변경할 종료일

pnl\_reason = '장기간 도서 연체로 인한 패널티 기간 연장' -- [변수] 변경할 사유

WHERE

pnl\_id = 10; -- [변수] 패널티 ID (PK)

### **사서 학생 관리 페이지 화면**



### **사서 학생 관리 쿼리**

USE library\_system;

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

-- 1. [회원 조회] 기본 정보 및 현재 상태 (블랙리스트 여부)

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

-- 화면 상단 '조기강 (20233849)' 카드 정보

SELECT

T1.usr\_id AS 학번,

T1.usr\_name AS 이름,

T1.usr\_dept AS 학과,

T1.usr\_phone AS 연락처,

T1.usr\_stat AS 회원상태, -- (정상/정지/졸업)

-- 현재 적용 중인 제재가 있는지 확인 (화면의 'BLACK LIST' 뱃지용)

(SELECT EXISTS(

SELECT 1 FROM penalties

WHERE usr\_id = T1.usr\_id AND pnl\_actv = 1 AND pnl\_end >= CURDATE()

)) AS 제재\_적용\_여부,

-- 현재 연체 중인 도서 권수 (화면: '2권')

(SELECT COUNT(\*) FROM loans

WHERE usr\_id = T1.usr\_id AND loan\_ret IS NULL AND loan\_due < CURDATE()) AS 연체\_도서\_수

FROM

users T1

WHERE

T1.usr\_id = 20233849; -- [변수] 검색창 입력값 (학번)

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

-- 2. [계산] 실시간 패널티 시뮬레이션 (Calc Box)

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

-- 화면 중간 '패널티 계산' 박스에 들어갈 데이터

SELECT

T3.bk\_title AS 연체\_도서명,

T1.loan\_due AS 반납\_예정일,

DATEDIFF(CURDATE(), T1.loan\_due) AS 연체일수, -- (오늘 - 예정일)

-- 예상 정지 일수 계산 (규정: 연체일수 \* 1일 정지 라고 가정 시)

-- 화면 로직(2배)에 맞춘다면 \* 2

(DATEDIFF(CURDATE(), T1.loan\_due) \* 1) AS 예상\_정지일

FROM

loans T1

JOIN

book\_copies T2 ON T1.cbk\_id = T2.cbk\_id

JOIN

books T3 ON T2.bk\_id = T3.bk\_id

WHERE

T1.usr\_id = 20233849 -- [변수] 학번

AND T1.loan\_ret IS NULL -- 미반납

AND T1.loan\_due < CURDATE(); -- 연체됨

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

-- 3. [리스트] 제재 이력 조회 (History Table)

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

-- 화면 하단 테이블 데이터

SELECT

pnl\_id AS No,

pnl\_type AS 위반\_유형, -- (연체, 분실, 파손 등)

pnl\_reason AS 상세\_사유, -- (도서명 등)

DATE\_FORMAT(pnl\_start, '%Y-%m-%d') AS 시작일,

DATE\_FORMAT(pnl\_end, '%Y-%m-%d') AS 종료일,

-- 상태 표시 로직

CASE

WHEN pnl\_actv = 1 AND pnl\_end >= CURDATE() THEN '적용 중'

ELSE '종료됨'

END AS 상태

FROM

penalties

WHERE

usr\_id = 20233849 -- [변수] 학번

ORDER BY

pnl\_start DESC;

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

-- 4. [액션] 제재 강제 해제 (버튼 클릭)

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

-- 화면의 '제재 강제 해제' 버튼 클릭 시 실행

-- 1) 패널티 테이블 비활성화 처리

UPDATE penalties

SET

pnl\_actv = 0, -- 비활성화

pnl\_reason = CONCAT(pnl\_reason, ' / (사서 강제 해제)') -- 사유 기록

WHERE

usr\_id = 20233849 -- [변수] 학번

AND pnl\_actv = 1

AND pnl\_end >= CURDATE();

-- 2) 유저 상태 '정상'으로 복구

UPDATE users

SET usr\_stat = '정상'

WHERE usr\_id = 20233849;

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

-- 5. [액션] 제재 이력 수정 (테이블 내 수정 버튼)

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

-- 하단 리스트의 작은 '수정' 버튼 클릭 시 (종료일 변경 등)

UPDATE penalties

SET

pnl\_end = '2026-03-31', -- [변수] 변경할 종료일

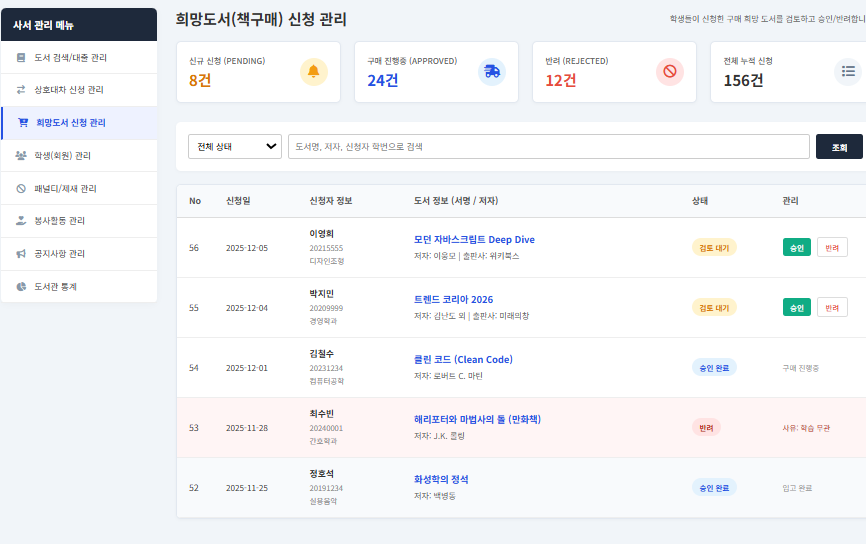
pnl\_reason = '장기간 도서 연체로 인한 패널티 기간 연장' -- [변수] 변경할 사유

WHERE

pnl\_id = 10; -- [변수] 패널티 ID (PK)

### 

### **사서 희망도서 구매 신청 화면**



### **사서 희망도서 구매 신청 쿼리**

USE library\_system;

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

-- 1. [상단 통계] 신청 상태별 건수 조회 (Dashboard)

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

SELECT

-- 1) 신규 신청 (PENDING -> 대기)

(SELECT COUNT(\*) FROM requests WHERE req\_stat = '대기') AS 신규\_신청,

-- 2) 구매 진행중 (APPROVED -> 승인 or 구매완료)

(SELECT COUNT(\*) FROM requests WHERE req\_stat IN ('승인', '구매완료')) AS 구매\_진행중,

-- 3) 반려 (REJECTED -> 반려)

(SELECT COUNT(\*) FROM requests WHERE req\_stat = '반려') AS 반려\_건수,

-- 4) 전체 누적 신청

(SELECT COUNT(\*) FROM requests) AS 전체\_누적\_신청;

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

-- 2. [리스트] 희망도서 신청 목록 조회 (검색/필터)

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

SELECT

T1.req\_id AS No,

DATE\_FORMAT(T1.req\_date, '%Y-%m-%d') AS 신청일,

-- 신청자 정보

T2.usr\_name AS 신청자명,

T2.usr\_id AS 학번,

T2.usr\_dept AS 학과,

-- 도서 정보

T1.req\_title AS 서명,

T1.req\_auth AS 저자,

T1.req\_pub AS 출판사,

T1.req\_isbn AS ISBN,

-- 상태 및 관리 정보

T1.req\_stat AS 상태, -- (대기/승인/반려/구매완료)

T1.req\_memo AS 신청사유, -- (학생이 쓴 글)

-- [수정됨] DDL 컬럼명 반영

T1.req\_proc\_memo AS 반려사유 -- (관리자가 쓴 글 - 화면 표시용)

FROM

requests T1

JOIN

users T2 ON T1.usr\_id = T2.usr\_id

WHERE

1=1

-- [필터 1] 상태 필터 (전체/대기/승인/반려)

AND T1.req\_stat = '대기' -- [변수] 선택된 값이 있을 때만 적용

-- [필터 2] 통합 검색 (도서명, 저자, 학번)

AND (

T1.req\_title LIKE CONCAT('%', '모던 자바스크립트 Deep Dive', '%')

OR T1.req\_auth LIKE CONCAT('%', '이웅모', '%')

OR T2.usr\_id LIKE CONCAT('%', '20233849', '%')

)

ORDER BY

-- 대기중인 건을 최상단으로, 그 다음 최신순 정렬

CASE WHEN T1.req\_stat = '대기' THEN 1 ELSE 2 END ASC,

T1.req\_date DESC;

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

-- 3. [액션] 관리자 승인/반려 처리 (UPDATE)

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

-- [버튼: 승인] 검토 대기 -> 승인 (구매 절차 시작)

UPDATE requests

SET

req\_stat = '승인',

req\_proc\_memo = NULL -- [수정됨] 반려 사유 초기화

WHERE

req\_id = 56; -- [변수] 신청 번호

-- [버튼: 반려] 검토 대기 -> 반려 (사유 입력 필수)

UPDATE requests

SET

req\_stat = '반려',

req\_proc\_memo = '학습 무관 도서' -- [수정됨] 예: '학습 무관 도서', '예산 초과'

WHERE

req\_id = 55; -- [변수] 신청 번호

-- [버튼: 입고 완료/구매 완료] 승인 -> 구매완료 (최종 처리)

UPDATE requests

SET

req\_stat = '구매완료'

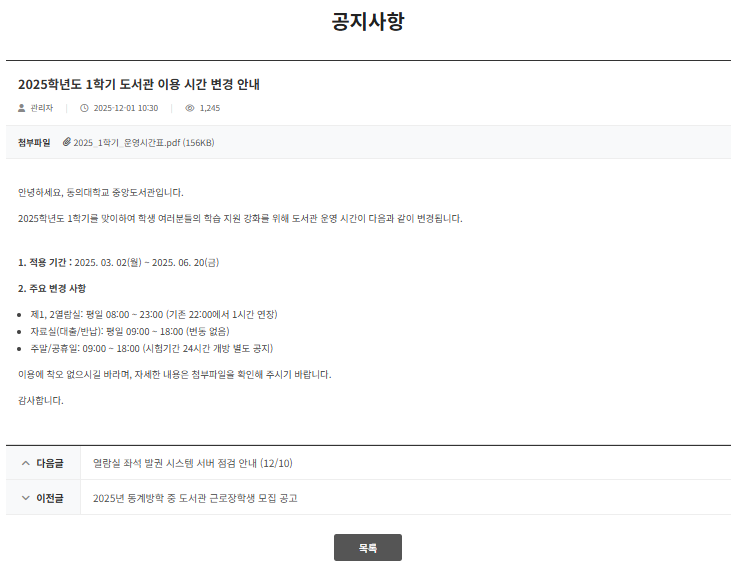
WHERE

req\_id = 54 AND req\_stat = '승인';

# **3.3 공용 페이지**

### 

### **공지사항 상세보기 페이지 화면**



### **공지사항 상세보기 쿼리**

/\*공지사항 ID로 상세 내용을 조회하고,

해당 공지사항의 조회수(ntc\_views)를 1 증가\*/

USE library\_system;

-- 조회수 증가

UPDATE notices

SET ntc\_views = ntc\_views + 1 -- 조회수 1 증가

WHERE ntc\_id = 1; -- 공지사항 ID

-- 상세 내용 조회

SELECT

n.ntc\_title,

n.ntc\_content,

n.ntc\_file,

DATE\_FORMAT(n.ntc\_date, '%Y.%m.%d %H:%i') AS '등록일시',

n.ntc\_views,

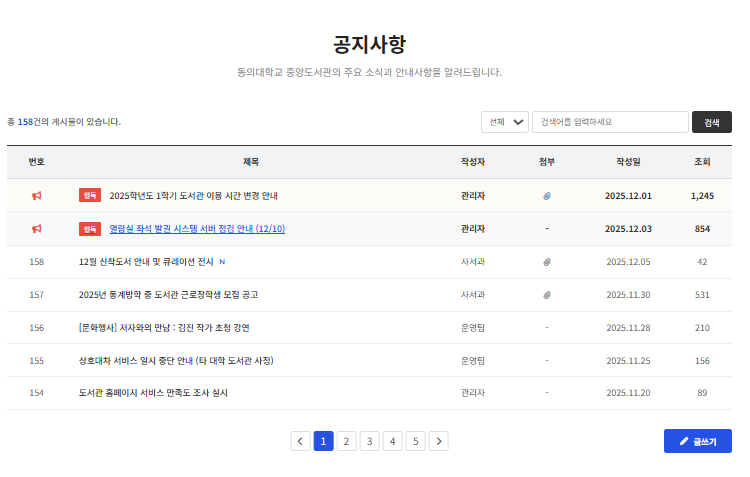
l.lib\_name AS '작성자'

FROM notices n

JOIN librarians l ON n.lib\_id = l.lib\_id

WHERE ntc\_id = 1;

### **공지사항 리스트 페이지 화면**



### **공지사항 리스트 쿼리**

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

USE library\_system;

-- 상단 필독 출력문

SELECT

n.ntc\_id,

n.ntc\_title,

-- [화면 로직] 값이 있으면 클립 아이콘 표시, 없으면 숨김

n.ntc\_file,

-- [화면 로직] 날짜 포맷 (예: 2025.12.01)

DATE\_FORMAT(n.ntc\_date, '%Y.%m.%d') AS ntc\_date\_fmt,

n.ntc\_views,

l.lib\_name AS writer\_name

FROM

notices n

JOIN

librarians l ON n.lib\_id = l.lib\_id

WHERE

n.ntc\_important = 1 -- 필독 갯수 조회

ORDER BY

n.ntc\_date DESC; -- 최신 필독 공지가 가장 위에

-- 2. 일반 공지사항 조회

SELECT

n.ntc\_id,

n.ntc\_title,

n.ntc\_file,

DATE\_FORMAT(n.ntc\_date, '%Y.%m.%d') AS ntc\_date\_fmt,

n.ntc\_views,

l.lib\_name AS writer\_name,

-- (백엔드에서 '오늘 날짜 - 작성일 < 3일'이면 New 배지 표시)

n.ntc\_date

FROM

notices n

JOIN

librarians l ON n.lib\_id = l.lib\_id

WHERE

n.ntc\_important = 0 -- ★ 핵심: 필독이 아닌 것만 조회

-- [검색 조건] 사용자가 검색어를 입력했을 경우 (AND 조건 추가)

-- AND n.ntc\_title LIKE CONCAT('%', ?, '%') -- 제목 검색 시

-- AND n.ntc\_content LIKE CONCAT('%', ?, '%') -- 내용 검색 시

ORDER BY

n.ntc\_date DESC, -- 최신순 정렬

n.ntc\_id DESC -- (날짜 같으면) 최신 등록순

LIMIT 10 OFFSET 0; -- 1페이지: OFFSET 0, 2페이지: OFFSET 10 ...

SELECT

COUNT(\*) AS total\_count

FROM

notices

WHERE

ntc\_important = 0;

-- 검색

SELECT

n.ntc\_id, n.ntc\_title, n.ntc\_file, n.ntc\_important,

DATE\_FORMAT(n.ntc\_date, '%Y.%m.%d') AS ntc\_date\_fmt,

n.ntc\_views,

l.lib\_name AS writer\_name

FROM notices n

JOIN librarians l ON n.lib\_id = l.lib\_id

WHERE

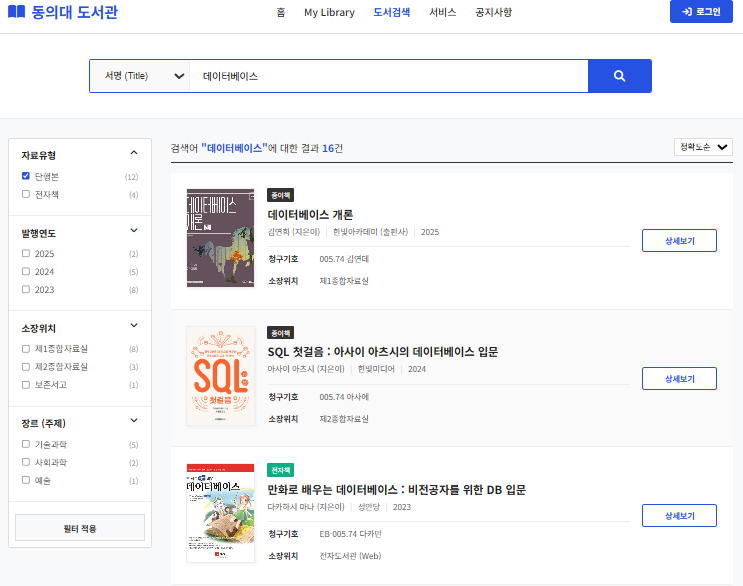
n.ntc\_important = 0 -- 필독 제외 (리스트 영역)

AND n.ntc\_title LIKE CONCAT('%', '이용시간', '%') -- ★ 제목 검색

ORDER BY n.ntc\_date DESC, n.ntc\_id DESC

LIMIT 10 OFFSET 0;

### **책 검색 페이지 화면**



### 

### **책 검색 쿼리**

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

-- 책 검색

USE library\_system;

SELECT

b.bk\_id, -- 청구기호

b.bk\_title, -- 서명

b.bk\_auth, -- 저자

b.bk\_pub, -- 출판사

b.bk\_year, -- [필터 연동] 발행연도

b.bk\_image, -- 표지 이미지

g.gnr\_name, -- [필터 연동] 장르(주제)

-- [필터 연동: 자료유형]

-- 실물(종이책) 존재 여부 (1이면 true)

MAX(CASE WHEN c.is\_ebook = FALSE THEN 1 ELSE 0 END) AS has\_paper,

-- 전자책 존재 여부 (1이면 true)

MAX(CASE WHEN c.is\_ebook = TRUE THEN 1 ELSE 0 END) AS has\_ebook,

-- [필터 연동: 소장위치]

-- 책이 소장된 '모든' 위치를 쉼표로 묶어서 반환 (예: "제1종합자료실, 제2종합자료실")

-- 서브쿼리 방식보다 빠르고, 특정 열람실 필터링 시에도 정보 누락이 없음

GROUP\_CONCAT(DISTINCT l.loc\_name ORDER BY l.loc\_name SEPARATOR ', ') AS all\_locations,

-- (참고) 현재 대출 가능한 실물 책 권수

COUNT(CASE WHEN c.is\_ebook = FALSE AND c.cbk\_stat = '사용가능' THEN 1 END) AS avail\_paper\_cnt

FROM

books b

JOIN

genres g ON b.gnr\_id = g.gnr\_id

LEFT JOIN

book\_copies c ON b.bk\_id = c.bk\_id

LEFT JOIN

locations l ON c.loc\_id = l.loc\_id -- 위치 명칭 조회를 위해 추가

WHERE

-- [백엔드에서 처리할 검색 조건 영역]

b.bk\_title LIKE CONCAT('%', '데이터 사이언스 입문', '%')

GROUP BY

b.bk\_id

ORDER BY

b.bk\_year DESC;

### 