

데이터베이스 기초 HW#2-2 (due date: 11월 29일)

수업 시간에 배운 데이터베이스 설계 절차에 따라 "도서관 무인대출반납시스템" 데이터베이스에 대한 물리적 설계를 마친 문서를 제공해 드립니다. 설계에 맞춰 SQL을 이용해 데이터베이스 구축을 해 주십시오. (클래스룸에 팀당 1개의 SQL Script 제출)

목차는 다음과 같습니다.

1.요구사항 분석

- 1.1. Swim Lane Chart
- 1.2. 시나리오
- 1.3. Business Rules
- 1.4. 요구사항 명세서

2.개념적 설계

- 2.1. 개체/관계/속성 정의
- 2.2. 개체 변환
- 2.3. 관계 변환
- 2.4. E-R 다이어그램

3. 논리적 설계

- 3.1. 개체 변환
- 3.2. 관계 변환

4. 물리적 설계

- 4.1. 영문 변환

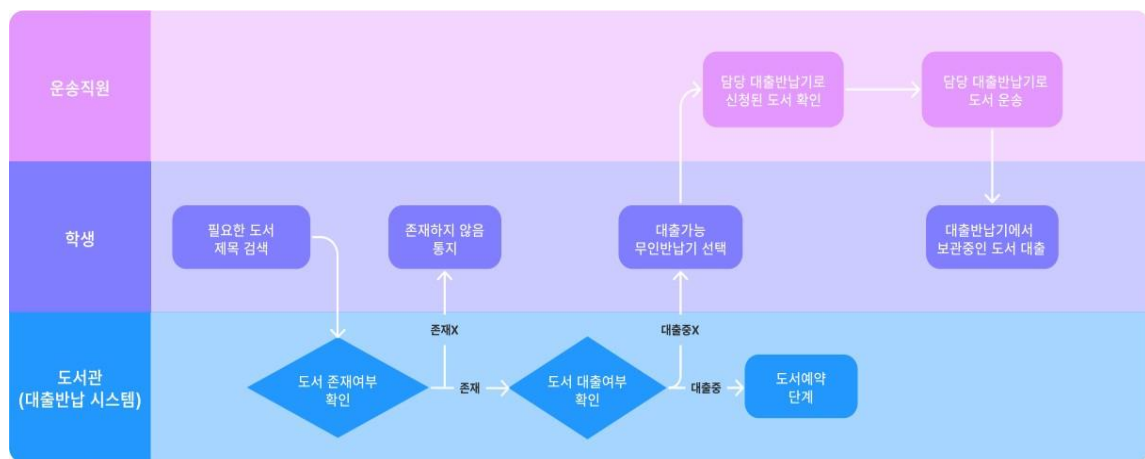
5. SQL 구현

- 5.1. SQL 테이블 정의 및 생성

1. 요구사항 분석

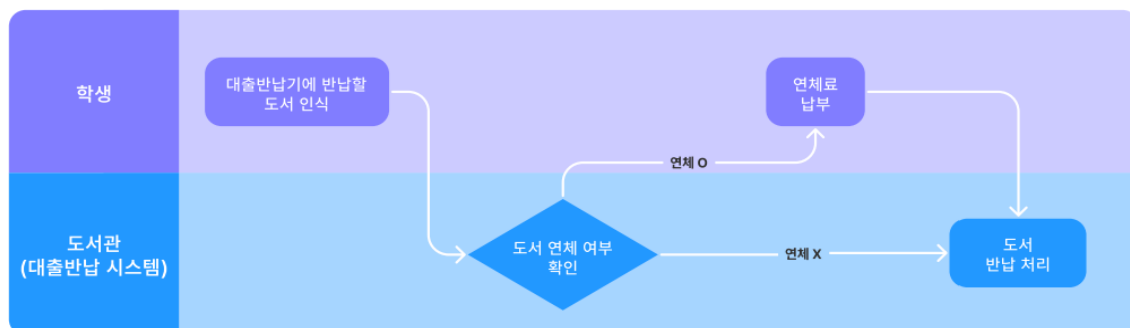
1.1 Swim Lane Chart (SLC)

(1) 대출: 먼저, 대출의 경우이다. 학생이 필요한 도서제목을 검색하고, 도서관을 통해 해당 도서가 도서관에 존재하는지 여부를 확인한다. 해당 도서가 도서관에 존재할 경우, 대출여부를 확인하고, 대출이 가능한 경우, 해당 도서를 가져올 대출반납기를 선택한다. 만일, 다른 학생이 도서를 대출한 상태여서 대출이 불가능할 경우에는 도서를 예약하며, 대출이 가능한 경우에는 학생이 선택한 대출반납기를 통해 도서를 대출을 신청한다. 신청을 확인한 운송직원은 해당 도서를 학생이 선택한 대출반납기로 운송하고, 이후에 학생은 대출반납기에서 보관중인 도서를 대출하게 된다.



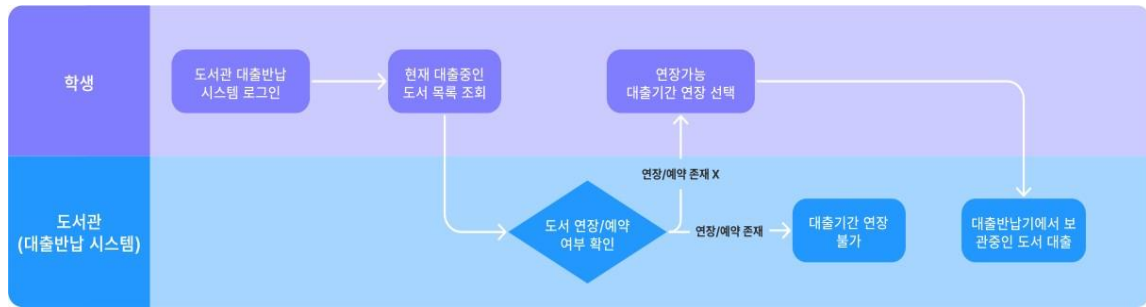
<그림1: 대출 SLC>

(2) 반납: 다음은 대출반납기를 통해 반납을 하는 경우이다. 반납은 대출한 학생이 대출반납기로 반납할 도서를 인식하고, 해당도서의 반납일을 확인하여, 연체여부를 파악한다. 연체가 되지 않았을 경우는 바로 반납 처리되며, 연체되었을 경우에는 연체기간을 파악하여 연체료를 납부한 후에 반납처리된다. 반납 처리가 된 도서는 다시 대출가능한 도서로 분류된다.



<반납 SLC>

(3) 대출 연장: 다음은 대출한 도서를 연장하는 경우이다. 먼저, 연장을 원하는 학생은 도서관 대출반납 시스템에 로그인하여, 자신이 현재 대출중인 도서를 조회한다. 그리고 연장을 하고 싶은 도서가 다른 학생이 예약 중이거나 이전에 연장한 횟수가 몇 회인지 파악한다. 예약이 없고, 연장을 3 회미만으로 했다면, 해당 도서는 연장이 가능한 상태로 분류되어, 학생은 대출기간을 연장할 수 있다. 대출연장을 선택하면 해당 도서는 연장 처리되어 대출기간이 늘어나게 된다.



<대출 연장 SLC>

1.2 시나리오

시나리오는 이용자 관점, 관리자 관점, 운송직원 관점으로 나누어서 고려해볼 수 있으며, 그 중 이용자 관점의 시나리오에서는 대출 / 연장 / 반납 측면으로 파악해 볼 수 있다. 각 단계의 과정은 아래와 같다.

(1) 시나리오1: 이용자 관점

대출	<ol style="list-style-type: none"> 1. 학생이 필요한 자료를 검색한다. 2. 도서가 대출상태인 경우, 예약을 한다. 3. 학생이 찾은 자료와 학생증을 인증한 후 본인이 원하는 대출반납기를 선택한다. 4. 책이 대출반납기에 도착했다는 메시지를 받으면 찾아간다.
연장	<ol style="list-style-type: none"> 1. 학생이 자료의 대출, 연장, 예약 상태를 확인한다. 2. 해당 자료가 연장 가능할 경우, 학생이 대출 연장을 한다.
반납	<ol style="list-style-type: none"> 1. 학생이 대출반납기를 이용해 도서를 인식한다. 2. 시스템은 도서의 연체 여부를 판단한다 3. 도서의 연체료가 있을 경우, 연체료 납부 후 반납처리한다. 4. 시스템은 반납 완료된 도서는 대출 가능한 도서로 분류한다.

(2) 시나리오 2: 관리자 관점

1. 학생이 대출을 신청하면 등록번호에 따라 해당 관리자에게 메시지를 보낸다.
2. 해당 관리자가 등록번호로 자료를 찾아서 모아 놓는다.
3. 관리자는 운반된 책을 제자리에 정리한다.

(3) 시나리오 3: 운송직원 관점

1. 시스템에서 자신이 담당하는 대출반납기로 들어온 대출 도서 목록을 확인한다.
2. 관리자가 모아둔 자료 중 일치하는 도서를 찾는다.
3. 책을 자신이 담당하는 대출반납기에 넣는다.
4. 반납된 도서는 전부 관리자에게 가져다준다.

1.3 Business Rules

Business Rules은 크게 대출/연장/예약/연체/운송 관련 룰로 나누어 고려해 볼 수 있다.

대출	<ul style="list-style-type: none">- 도서 대출은 최대 3 권까지 가능하다.- 도서 대출 기간은 3 주까지 가능하다.- 대출 시 대출반납기 보관일은 3 일이다.
연장	<ul style="list-style-type: none">- 연장은 1 주일간 가능하다.- 연장 횟수는 3 번까지 가능하다.- 이미 연체된 도서가 있을 경우 대출 연장은 불가하다.- 대출한 도서가 예약된 경우 대출 연장은 불가하다.
예약	<ul style="list-style-type: none">- 예약 유효일은 도서가 반납된 후 3 일이다.- 예약한 도서가 반납되었을 경우 학생의 전화번호로 도서가 대출가능상태임을 통지한다.- 연체된 도서가 있는 경우 예약은 불가하다.- 예약 한도 권수는 3 권이다.- 한 권당 예약은 1 명만 가능하다.- 예약 한도 권수를 초과한 경우 예약이 불가하다.
연체	<ul style="list-style-type: none">- 반납 예정일을 경과한 경우, 연체료를 납부해야 한다.- 연체료는 1 일 1 책 당 100 원이다.
운송	<ul style="list-style-type: none">- 운송직원은 인당 2 개의 대출반납기를 담당한다.

1.4 요구사항 명세서

해당 데이터베이스의 사용자는 도서관 이용 학생, 운송직원, 관리자로 규정되며, 학생의 도서 대출/반납 흐름에 의해 데이터가 변화한다. 아래의 전체 요구사항 명세서는 사용자의 요구사항을 바탕으로 작성되었다.

1. 학생이 필요한 도서의 제목이나 키워드를 검색한다.
2. 도서는 제목, 저자, 등록번호, 관리자 번호에 의해 식별된다.
3. 도서를 조회하여 학생이 필요로 하는 책이 존재하는지 확인한다.
4. 도서가 존재하는 경우, 대출 시스템을 확인한다.
5. 학생은 학번, 이름, 전화번호, 대출반납기번호에 의해 식별되어, 대출가능여부를 판단한다.
6. 예약현황은 도서제목, 등록번호, 예약 순번, 도착 통보일로 식별된다.
7. 대출이 가능한 도서의 경우, 도서의 등록번호에 따라 해당 관리자에게 메시지를 보낸다.
8. 해당 관리자가 본인이 관리하는 담당구역 번호에 해당하는 도서를 찾아서 모아둔다.
9. 관리자는 직원번호와 담당구역번호에 의해 식별된다.
10. 도서 대출이 가능한 경우, 학생은 학생증을 인증한 후 본인이 원하는 대출반납기를 지정한다.
11. 대출반납기는 대출반납기번호, 위치, 현재 보관 권수, 등록번호에 의해 식별된다.
12. 운송직원은 본인이 담당하는 무인 반납기로 운송할 대출도서 등록번호들을 확인 후 찾는다.
13. 운송직원은 찾은 도서를 본인 담당 무인 반납기로 운송한다.
14. 대출시스템은 학생에게 도서가 대출 준비되었다는 안내 메시지를 전송한다.
15. 학생은 안내메세지를 받은 후 본인이 지정한 무인 반납기에서 도서를 찾아간다.
16. 학생이 찾아간 도서를 대출시스템에서 대출처리한다.
17. 대출시스템은 등록번호, 보관일, 대출학생학번, 반납예정일, 대출반납기 번호, 연체료, 연장횟수로 식별된다.

2. 개념적 설계

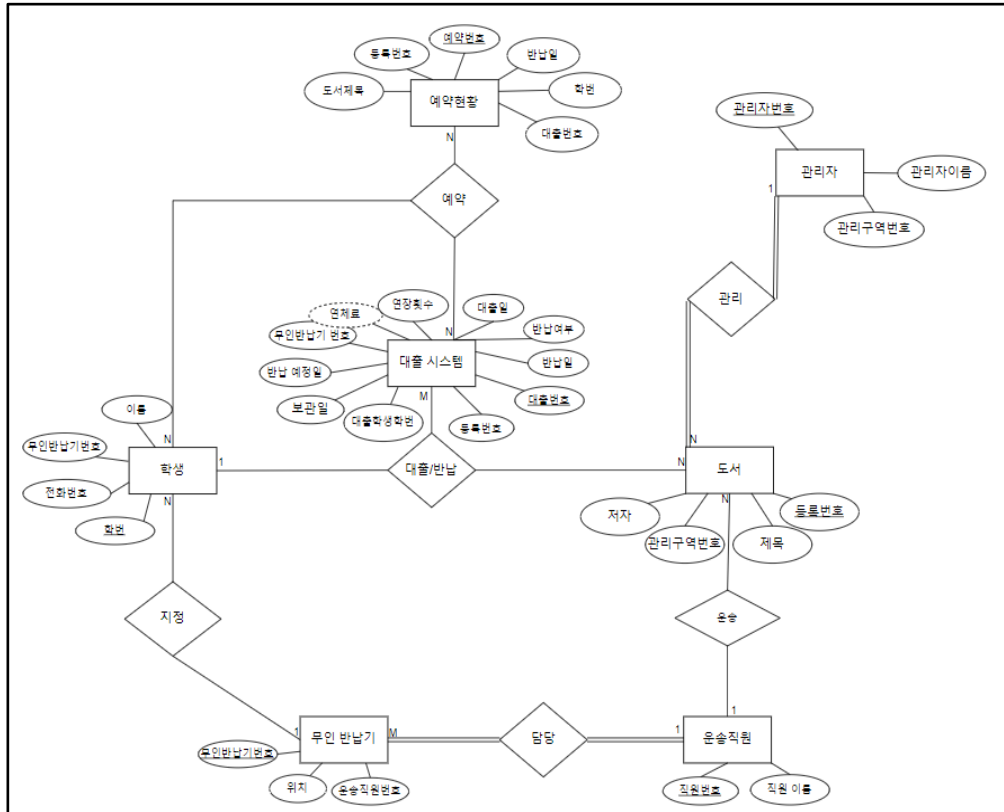
2.1. 개체/관계/속성 정의

개체/관계/속성 정의는 전체 요구사항 명세서를 바탕으로 정의되었다.

1. 학생이 필요한 도서의 제목을 검색한다.
2. 도서는 제목, 저자, 등록번호, 구역번호에 의해 식별된다.
3. 도서를 조회하여 학생이 필요로 하는 책이 존재하는지 확인한다.
4. 도서가 존재하는 경우, 대출 시스템을 확인한다.
5. 학생은 학번, 이름, 대출권수, 예약권수, 전화번호에 의해 식별되어, 대출가능여부를 판단한다.
6. 예약현황은 도서제목, 등록번호, 예약 순번, 도착 정보으로 식별된다.
7. 대출이 가능한 도서의 경우, 도서의 등록번호에 따라 해당 관리자에게 메시지를 보낸다.
8. 해당 관리자가 본인이 관리하는 담당구역 번호에 해당하는 도서를 찾아서 모아둔다.
9. 관리자는 직원 ID 번호와 담당구역번호에 의해 식별된다.
10. 도서 대출이 가능한 경우, 학생은 학생증을 인증한 후 본인이 원하는 무인 대출반납기를 지정한다.
11. 무인 대출반납기는 무인 대출반납기 ID, 대출반납기 위치, 담당 직원 번호에 의해 식별된다.
12. 문송직원은 본인이 담당하는 무인 대출반납기로 운송할 대출도서 등록번호들을 확인 후 도서를 찾는다.
13. 문송직원은 찾은 도서를 본안 담당 무인 대출반납기로 운송한다.
14. 대출시스템은 학생에게 도서가 대출 준비되었다는 안내 메시지를 전송한다.
15. 학생은 안내메시지를 받은 후 지정된 무인 대출반납기에서 도서를 찾아간다.
16. 학생이 찾아간 도서를 대출시스템에서 대출처리한다.
17. 대출시스템은 등록번호, 대출번호, 보관일, 대출학생학번, 반납예정일, 반납 여부, 무인 대출반납기 ID, 연체료, 연장횟수로 식별된다.

2.2 ER 다이어그램

2.1에서 정의한 내용을 토대로 ER 다이어그램을 그려볼 수 있다.



3. 논리적 설계

3.1. 개체변환

ER 다이어그램을 토대로, 기본키와 일반속성키를 구분해, 개체별로 표현해볼 수 있다. 각 테이블의 기본키는 밑줄로 표현하였고, ER 다이어그램을 기반으로 총 7 개의 개체 릴레이션이 표현되고, 변환하는 과정에서 SQL 구현 시 필요하다고 생각되는 관리구역 테이블을 추가하였다.

테이블명(속성)
학생 (학번, 이름, 전화번호, 대출반납기번호)
대출시스템 (대출번호, 등록번호, 학생 학번, 보관일, 대출일, 반납예정일, 대출반납기번호, 연체료, 연장횟수, 반납여부)
도서 (등록번호, 제목, 저자, 관리구역번호)
예약 (예약번호, 등록번호, 도서제목, 반납일, 대출번호, 학생 학번)
관리자 (관리자번호, 관리자 이름, 관리구역번호)
운송직원 (운송직원번호, 직원이름)
대출반납기 (대출반납기번호, 위치, 운송직원번호)

3.2 관계변환

관계 변환은 ER 다이어그램을 통해 파악한 관계를 토대로, 릴레이션 형태로 표현해보았다. 각 관계의 카디널리티, 차수, 관계 종속성에 따라 적용되는 관계 변환 규칙을 기반으로 관계변환을 진행했다. 관계 릴레이션의 기본키는 밑줄로, 외래키는 빨간색 글씨로 표현한 결과는 아래와 같다.

관계	관계 릴레이션
다원관계	대출/반납 (학생학번, 대출시스템대출번호, 도서등록번호)
일대다-부분참여관계	도서 (등록번호, 제목, 저자, 관리구역번호, 운송직원번호) 운송직원 (운송직원번호, 직원이름)
일대일-전체참여관계	대출반납기 (대출반납기번호, 위치, 운송직원번호, 담당직원번호) 운송직원 (운송직원번호, 직원이름, 담당대출반납기번호)
일대다-부분참여관계	학생 (학번, 이름, 전화번호, 대출반납기번호, 전화번호, 지정대출반납기번호) 대출반납기 (대출반납기번호, 위치, 운송직원번호)
다원관계	예약 (예약도서등록번호, 예약예약번호, 학생학번, 대출시스템대출번호)
일대다-전체참여관계	도서 (등록번호, 제목, 저자, 관리구역번호, 관리자관리자번호) 관리자 (관리자번호, 관리구역번호)

4. 물리적 설계

4.1. 영문변환

관계변환을 통해 완성된 관계 릴레이션을 기반으로, SQL에 적용시키기 위해 영문변환을 진행했다.

테이블명(속성)
checkout_sys (book_id, checkout_id, std_id, date_keep, date_ch, date_prereturn, date_return, machine_id, price_overdue, times_extension, return_0)
book (book_id, title, author, sector_id)
student (std_id, std_name, machine_id, std_phone)
selfs_machine (machine_id, machine_location, dlv_id)
checkout_return (book_id, checkout_id, std_id) reservation (reserv_id, book_id, checkout_id, std_id)
librarian (lib_id, lib_name, sector_id)
delivery (dlv_id, dlv_name)

5. SQL 구현

5.1. SQL 테이블 정의 및 생성

SQL 에 입력하기 위한 테이블을 정의하는 과정에서 관리구역을 구분하는 테이블도 추가적으로 필요할 것이라고 예상되어 추가하였다. 추가한 내용을 포함하여, 각 테이블에 필요한 내용을 정리한 결과이다.

테이블명	컬럼명	속성명	도메인	데이터타입	길이	null 여부	constraint	key
student	std_id	학번	학생 코드번호	varchar	10	NOT NULL		PK
	std_name	이름	이름	varchar	10	NOT NULL		
	machine_id	대출반납기번호	대출반납기 코드번호	varchar	10	NOT NULL		
	std_phone	전화번호	전화번호	varchar	15	NOT NULL		
checkout_return	std_id	학번	학생 코드번호	varchar	10	NOT NULL		PK
	checkout_id	대출번호	대출 코드번호	varchar	10	NOT NULL		PK
	book_id	등록번호	도서 코드번호	int	-	NOT NULL		PK
checkout_sys	book_id	등록번호	도서 코드번호	int	-	NOT NULL		
	checkout_id	대출번호	대출 코드번호	varchar	10	NOT NULL		PK
	std_id	학번	학생 코드번호	varchar	10	NOT NULL		

	date_keep	보관일	날짜	date	-	NULL		
	date_ch	대출일	날짜	date	-	NULL		
	date_prereturn	반납예정일	날짜	date	-	NULL		
	date_return	반납일	날짜	date	-	NULL		
	machine_id	대출반납기번호	대출반납기 코드번호	varchar	10	NOT NULL		
	price_overdue	연체료	-	int	-	NULL	100 원	
	times_extension	연장횟수	-	int	-	NULL	max 3 회	
	return_0	반납여부	-	int	-	NOT NULL	1 : 반납 완료	
book	book_id	등록번호	도서 코드번호	int	-	NOT NULL		PK
	title	제목	도서 제목	varchar	80	NOT NULL		
	author	저자	도서 저자	varchar	20	NOT NULL		
	sector_id	관리구역번호	관리구역 코드번호	varchar	10	NOT NULL		
reservation	reserv_id	예약번호	예약 코드번호	varchar	10	NOT NULL		PK
	book_id	등록번호	도서 코드번호	int	-	NOT NULL		
	title	제목	도서 제목	varchar	30	NOT NULL		
	date_return	반납일	날짜	date	-	NULL		
	checkout_id	대출번호	대출 코드번호	varchar	10	NOT NULL		
	std_id	학번	학생 코드번호	varchar	10	NOT NULL		
librarian	lib_id	관리자 번호	관리자 코드번호	varchar	10	NOT NULL		PK
	lib_name	관리자이름	이름	varchar	10	NOT NULL		
	sector_id	관리구역번호	관리구역 코드번호	varchar	10	NOT NULL		
delivery	dlv_id	운송직원번호	운송직원 코드번호	varchar	10	NOT NULL		PK
	dlv_name	운송직원이름	이름	varchar	10	NOT NULL		
selfs_machine	machine_id	대출반납기번호	대출반납기 코드번호	varchar	10	NOT NULL		PK
	machine_location	반납기위치	-	varchar	20	NOT NULL		
	dlv_id	운송직원번호	운송직원 코드번호	varchar	10	NOT NULL		
sector	sector_id	관리구역번호	관리구역 코드번호	varchar	10	NOT NULL		PK
	sector_name	관리구역분야	관리구역 분야이름	varchar	10	NOT NULL		
	lib_id	관리자 번호	관리자 코드번호	varchar	10	NOT NULL		

정의된 테이블 내용을 기반으로 SQL을 통해 테이블을 생성하는 것이 HW#2-2가 되겠습니다.

[추가 정보]

1. 외래키 설정

- 학생과 무인대출반납기 외래키: student.machine_id 는 selfs_machine.machine_id 참고
- 운송직원과 무인대출반납기 외래키: delivery.dlv_id는 selfs_machine.dlv_id 참고
- 대출시스템과 무인대출반납기 외래키: checkout_sys.machine_id는 selfs_machine.machine_id참고
- 도서와 구역 외래키: book.sector_id는 sector.sector_id 참고
- 관리자와 구역 외래키: librarian.sector_id는 sector.sector_id 참고

2. 데이터 삽입은 골격 코드에서 제공

- 도서의 경우 열린데이터광장에서 도서목록을 100개 가져왔으며, 4가지 분야로 분류함. 대출반납기는 2호선의 8개 지하철역을 통해 이용되는 상황을 가정하여 데이터를 추가함. 나머지 데이터는 임의로 생성.

3. 비즈니스 룰 및 제약 조건 함수 구현은 HW#2-3에서 해결할 것

4. 다음 쿼리를 통해 데이터베이스 구축이 잘 되었는지 확인할 것

쿼리	응답												
#박유찬 학생이 대출한 도서제목 확인 SELECT std_name, title FROM checkout_sys,student,book WHERE book.book_id=checkout_sys.book_id and student.std_id=checkout_sys.std_id and student.std_name='박유찬';	<table><tr><th></th><th>std_name</th><th>title</th></tr><tr><td>▶</td><td>박유찬</td><td>이기적 유전자</td></tr><tr><td></td><td>박유찬</td><td>내게 무해한 사람 :최은영 소설</td></tr></table>		std_name	title	▶	박유찬	이기적 유전자		박유찬	내게 무해한 사람 :최은영 소설			
	std_name	title											
▶	박유찬	이기적 유전자											
	박유찬	내게 무해한 사람 :최은영 소설											
#강민애 학생이 예약한 도서제목 확인 SELECT std_name, title,book.book_id FROM book, student, reservation WHERE book.book_id=reservation.book_id and student.std_id=reservation.std_id and student.std_name='강민애';	<table><tr><th></th><th>std_name</th><th>title</th><th>book_id</th></tr><tr><td>▶</td><td>강민애</td><td>지구 끝의 온실 :김초엽 장편소설</td><td>16</td></tr><tr><td></td><td>강민애</td><td>(보도 생퍼의) 이기는 습관 :불가능을 뛰어넘어 ...</td><td>59</td></tr></table>		std_name	title	book_id	▶	강민애	지구 끝의 온실 :김초엽 장편소설	16		강민애	(보도 생퍼의) 이기는 습관 :불가능을 뛰어넘어 ...	59
	std_name	title	book_id										
▶	강민애	지구 끝의 온실 :김초엽 장편소설	16										
	강민애	(보도 생퍼의) 이기는 습관 :불가능을 뛰어넘어 ...	59										
#도서 대출 #'불편한 편의점' 도서가 있는지 확인 select title, book_id from book where title='불편한 편의점';	<table><tr><th></th><th>title</th><th>book_id</th></tr><tr><td>▶</td><td>불편한 편의점</td><td>21</td></tr><tr><td>•</td><td>NULL</td><td>NULL</td></tr></table>		title	book_id	▶	불편한 편의점	21	•	NULL	NULL			
	title	book_id											
▶	불편한 편의점	21											
•	NULL	NULL											

<p>#학생 이름을 통해 각 학생이 대출한 책 목록 확인</p> <pre>select book_id,student.std_id from student, checkout_sys where student.std_name='김철수' and student.std_id=checkout_sys.std_id;</pre>	<table><tr><td></td><td>book_id</td><td>std_id</td></tr><tr><td>▶</td><td>16</td><td>B911423</td></tr><tr><td></td><td>59</td><td>B911423</td></tr></table>		book_id	std_id	▶	16	B911423		59	B911423
	book_id	std_id								
▶	16	B911423								
	59	B911423								
<p>#해당 날짜 대출 들어온 도서 book_id 도출</p> <pre>SELECT std_id,book_id from checkout_sys where date_keep='2023-05-13';</pre>	<table><tr><td></td><td>std_id</td><td>book_id</td></tr><tr><td>▶</td><td>B946271</td><td>99</td></tr></table>		std_id	book_id	▶	B946271	99			
	std_id	book_id								
▶	B946271	99								
<p>#관리자가 도서 테이블을 통해 본인 담당 책 확인 (당일 대출 신청 들어온 도서들)</p> <pre>SELECT book_id,lib_id FROM librarian, book WHERE librarian.sector_id = book.sector_id AND book_id IN (SELECT book_id FROM checkout_sys WHERE date_keep = '2023-05-13');</pre>	<table><tr><td></td><td>book_id</td><td>lib_id</td></tr><tr><td>▶</td><td>99</td><td>L2</td></tr></table>		book_id	lib_id	▶	99	L2			
	book_id	lib_id								
▶	99	L2								
<p>#운송직원이 담당무인반납기 책 조회하여 운송할 책 리스트 도출</p> <pre>select book_id, dlv_id from selfs_machine, checkout_sys where selfs_machine.machine_id=checkout_sys.machine_id and date_keep='2023-05-13' and dlv_id='D2';</pre>	<table><tr><td></td><td>book_id</td><td>dlv_id</td></tr><tr><td>▶</td><td>99</td><td>D2</td></tr></table>		book_id	dlv_id	▶	99	D2			
	book_id	dlv_id								
▶	99	D2								

	author	저자	도서 저자	varchar	20	NOT NULL		
	sector_id	관리구역번호	관리구역 코드번호	varchar	10	NOT NULL		
reservation	reserv_id	예약번호	예약 코드번호	varchar	10	NOT NULL		PK
	book_id	등록번호	도서 코드번호	int	-	NOT NULL		
	title	제목	도서 제목	varchar	30	NOT NULL		
	date_return	반납일	날짜	date	-	NULL		
	checkout_id	대출번호	대출 코드번호	varchar	10	NOT NULL		
	std_id	학번	학생 코드번호	varchar	10	NOT NULL		
librarian	lib_id	관리자 번호	관리자 코드번호	varchar	10	NOT NULL		PK
	lib_name	관리자이름	이름	varchar	10	NOT NULL		
	sector_id	관리구역번호	관리구역 코드번호	varchar	10	NOT NULL		
delivery	dlv_id	운송직원번호	운송직원 코드번호	varchar	10	NOT NULL		PK
	dlv_name	운송직원이름	이름	varchar	10	NOT NULL		
selfs_machine	machine_id	대출반납기번호	대출반납기 코드번호	varchar	10	NOT NULL		PK
	machine_location	반납기위치	-	varchar	20	NOT NULL		
	dlv_id	운송직원번호	운송직원 코드번호	varchar	10	NOT NULL		
sector	sector_id	관리구역번호	관리구역 코드번호	varchar	10	NOT NULL		PK
	sector_name	관리구역분야	관리구역 분야이름	varchar	10	NOT NULL		
	lib_id	관리자 번호	관리자 코드번호	varchar	10	NOT NULL		