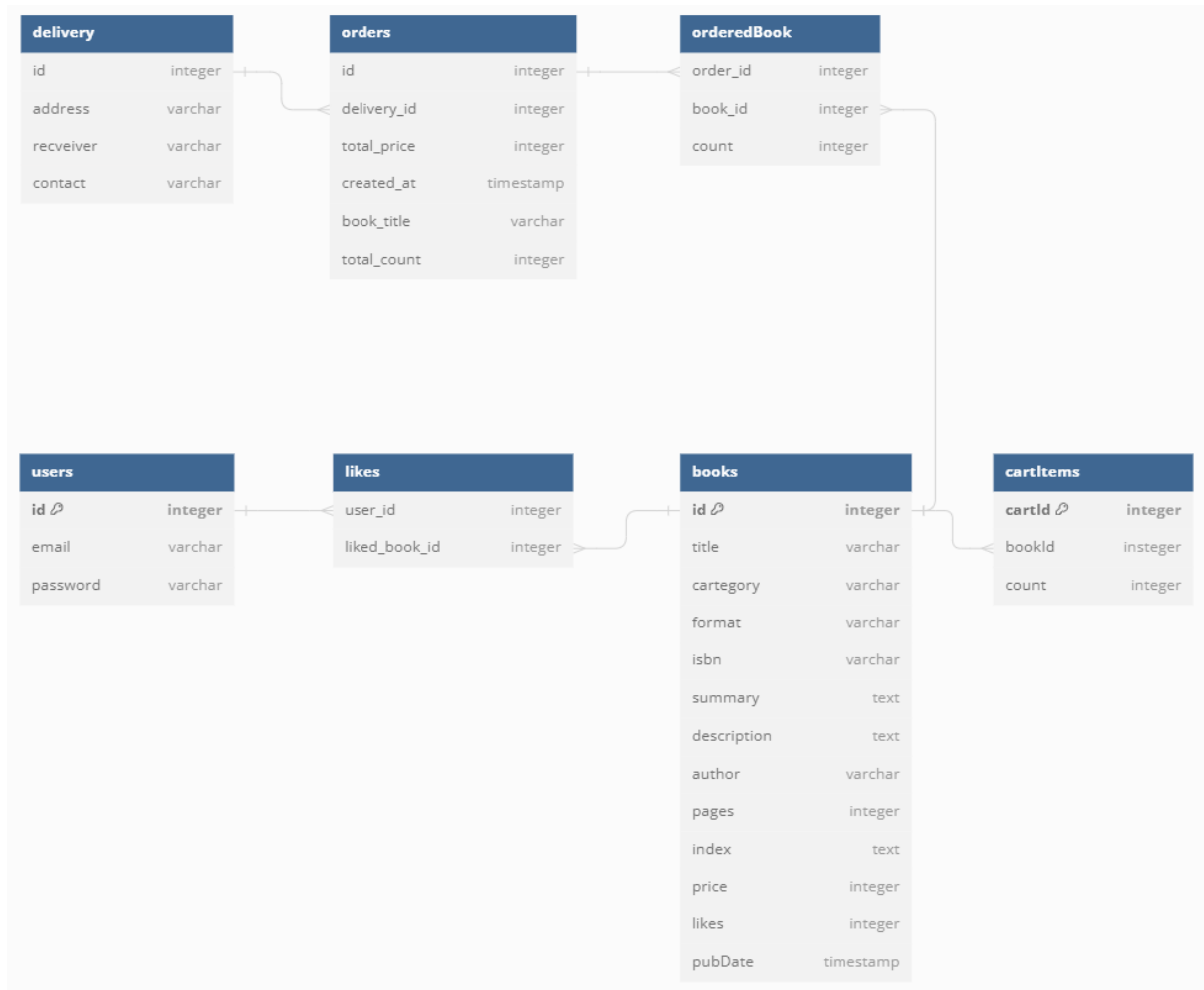


전체 ERD



회원 ERD 및 API

users	
id 🔗	integer
email	varchar
password	varchar

회원가입

Method	POST
URI	/join
HTTP Status Code	성공(201)
Request Body	{ email: "사용자가 입력한 이메일" password: "사용자가 입력한 비밀번호" }
Response Body	

로그인

Method	POST
URI	/login
HTTP Status Code	성공(200)
Request Body	{ email: "사용자가 입력한 이메일" password: "사용자가 입력한 비밀번호" }
Response Body	JWT token

비밀번호 초기화 요청

Method	POST
URI	/reset
HTTP Status Code	성공(200)
Request Body	{ email: "사용자가 입력한 이메일" }
Response Body	

비밀번호 초기화(수정)

Method	PUT
URI	/reset
HTTP Status Code	성공(200)
Request Body	{ password: "사용자가 입력한 비밀번호" }
Response Body	

도서 ERD 및 API

books	
id 	integer
title	varchar
category	varchar
format	varchar
isbn	varchar
summary	text
description	text
author	varchar
pages	integer
index	text
price	integer
likes	integer
pubDate	timestamp

전체 도서 조회

한 번에 8개씩 보내주는 것과 이미지 경로는 고려하지 않았다.

Method	GET
URI	/books
HTTP Status Code	성공(200)
Request Body	
Response Body	[{ id: 도서 id, title: "도서 제목", summary: '요약 설명', author: "작가", price: 가격, likes: "좋아요 수", pubDate: "출간일" }, ...]

개별 도서 조회

이미지 경로는 고려하지 않았다.

Method	GET
URI	/books/{bookId}
HTTP Status Code	성공(200)
Request Body	
Response Body	{ id: 도서 id, title: "도서 제목", category: "카테고리", format: "포맷", isbn: "isbn", summary: '요약 설명', description: "상세 설명" author: "작가", pages: 쪽 수, index: "목차", price: 가격, likes: "좋아요 수" liked: Boolean, pubDate: "출간일" }

카테고리별 도서 목록 조회

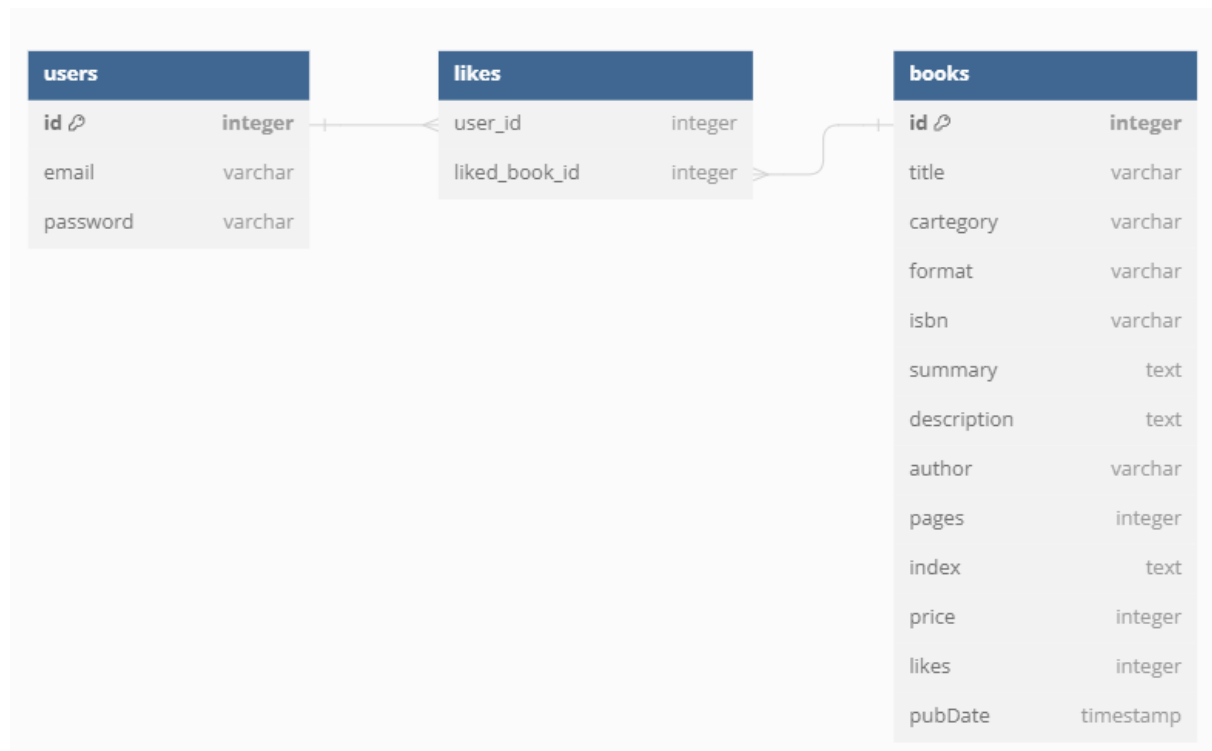
한 번에 8개씩 보내주는 것과 이미지 경로는 고려하지 않았다.

신간의 기준은 출간일 기준 30일 이내이다. new 쿼리를 통해 신간 조회 여부를 결정한다.

Method	GET
URI	/books?categoryId={categoryId}&new={boolean}
HTTP Status Code	성공(200)
Request Body	
Response Body	[{ id: 도서 id, title: "도서 제목", summary: '요약 설명', author: "작가", price: 가격, likes: "좋아요 수" pubDate: "출간일" }, ...]

좋아요 ERD 및 API

사용자 id와 책 id를 복합키로 지정하거나 고유 id 컬럼을 기본키로 설정할 예정이다.



좋아요 추가

likes 테이블에 좋아요를 누른 사용자 id와 책의 id를 삽입한다.

Method	POST
URI	/likes/{bookId}
HTTP Status Code	성공(201)
Request Body	
Response Body	

좋아요 취소

likes 테이블에서 해당 행을 삭제한다.

Method	DELETE
URI	/likes/{bookId}
HTTP Status Code	성공(200)
Request Body	
Response Body	

장바구니 API 설계



장바구니 담기

Method	POST
URI	/cart
HTTP Status Code	성공(201)
Request Body	{ bookId: 도서 id, count: 수량, }
Response Body	

장바구니 조회

Method	GET
URI	/cart
HTTP Status Code	성공(200)
Request Body	
Response Body	[{ cartId: 장바구니 도서 id, bookId: 도서 id title: "도서 제목", summary: "도서 요약", count: 수량, price: 가격 }, ...]

장바구니 삭제

Method	DELETE
URI	/cart/{cartId}
HTTP Status Code	성공(200)
Request Body	
Response Body	

주문 예상 상품 목록 조회

장바구니에서 선택한 상품들의 목록을 보여주는 API이다.

Method	GET
URI	/
HTTP Status Code	성공(200)
Request Body	[cartId, ...]
Response Body	[{ cartId: 장바구니 도서 id, bookId: 도서 id title: "도서 제목", summary: "도서 요약", count: 수량, price: 가격 }, ...]

주문(결제) ERD 및 API



결제하기

장바구니에서 주문된 상품은 장바구니 테이블(cartItems)에서 삭제한다.

결제하면 주문 테이블(orders)에 데이터가 삽입된다.

Method	POST
URI	/orders
HTTP Status Code	성공(201)
Request Body	<pre>{ items: [{ cartId: 장바구니 도서 id, bookId: 도서 id, count: 수량 }, ...], delivery: { address: "주소", receiver: "이름", contact: "010-0000-0000" }, totalPrice: 총 금액 }</pre>
Response Body	

주문 목록(내역) 조회

Method	GET
URI	/orders
HTTP Status Code	성공(200)
Request Body	
Response Body	[{ order_id: 주문 id, created_at: "주문 일자", delivery: { address: "주소", receiver: "이름", contact: "전화번호", }, bookTitle: "대표 책 제목", totalPrice: 결제 금액 totalCount: 총 수량 }, ...]

주문 상세 조회

Method	GET
URI	/orders/{orderId}
HTTP Status Code	성공(200)
Request Body	
Response Body	[{ bookId: 도서 id, bookTitle: "도서 제목", author: "작가명", price: 가격, count: 수량 }, ...]