

2024.04.17

주문 API 설계

1. 결제하기 = 주문하기

주문된 상품은 삭제되어야 함.

Method	POST
URI	/orders
HTTP status code	성공 200
Request Body	<pre>{ items : [{ cartItemId : 장바구니 도서 id bookId : 도서 id quantity : 수량 } { cartItemId : 장바구니 도서 id bookId : 도서 id quantity : 수량 }] delivery : { address : "주소" receiver : "이름 " contact "010-0000-0000" } totalPrice : 총 금액 }</pre>
Response Body	<pre>[]</pre>

2. 주문목록 조회

Method	GET
URI	/orders
HTTP status code	성공 200
Request Body	
Response Body	<pre>[{ created_at : "주문일자" delivery : { address : "주소" receiver : "이름" contact : "전화번호" } bookTitle : "대표 책 제목", titlePrice : 결제금액 totalCount : 총수량 }]</pre>

3. 주문 상세 조회

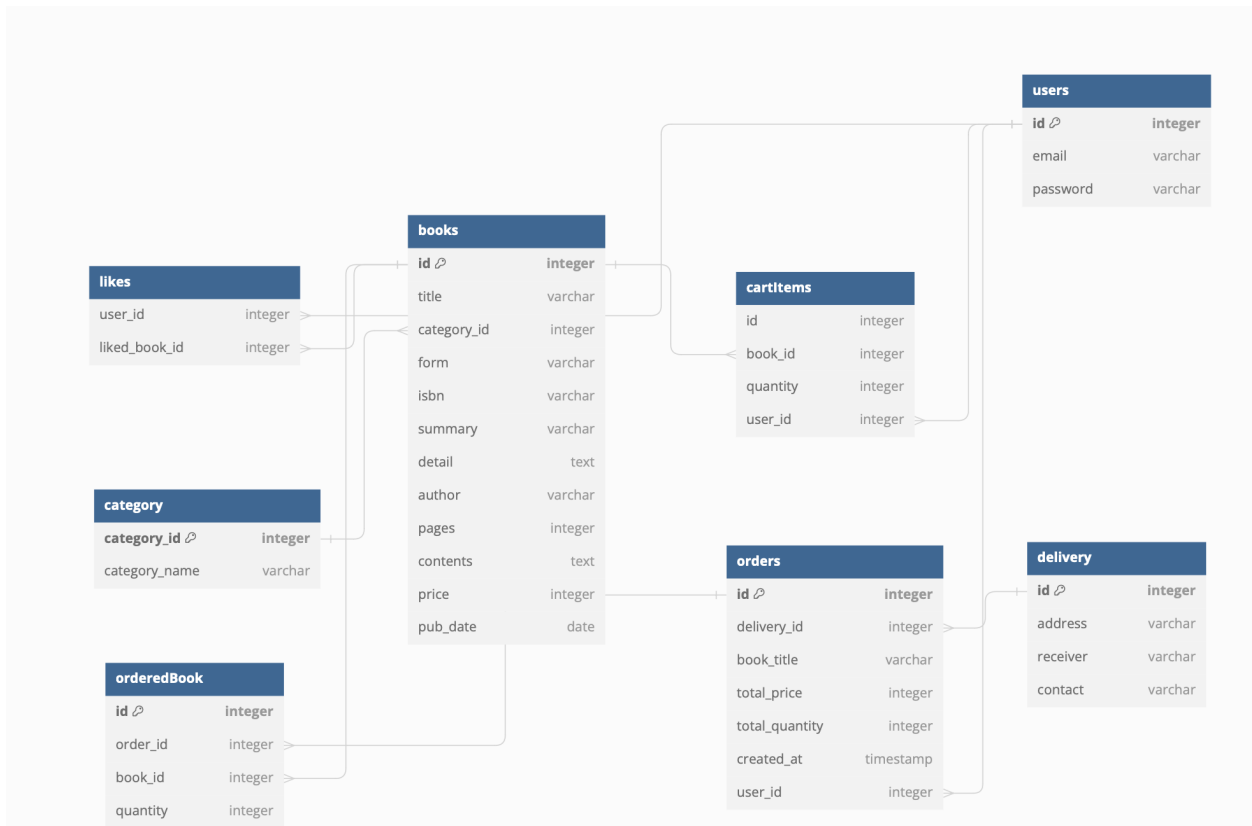
Method	GET
URI	/orders/{orderId}
HTTP status code	성공 200
Request Body	
Response Body	<pre>[{ bookId : 도서 id, bookTitle : "도서 제목" author : "작가명", price : 가격 quantity : 수량 }]</pre>

```

    }
  ]
}

```

주문 DB ERD 설계



방금 insert한 데이터 pk 가져오는 방법

- LAST_INSERT_ID() : 시간차 공격 >> 이전 값을 들고오는 간간히 오류가 남
- MAX()

```
sql = `INSERT INTO orders (book_title, total_quantity, total_price, user_id, delivery_id)
      VALUES (?, ?, ?, ?, ?)`;
values = [firstBookTitle, totalQuantity, totalPrice, userId, delivery_id]
conn.query(sql, values,
  (err, results) => {
    if(err) {
      console.log(err);
      return res.status(StatusCodes.BAD_REQUEST).end();
    }

    order_id = results.insertId;
    console.log(order_id);

    return res.status(StatusCodes.OK).json(results);
  })
```