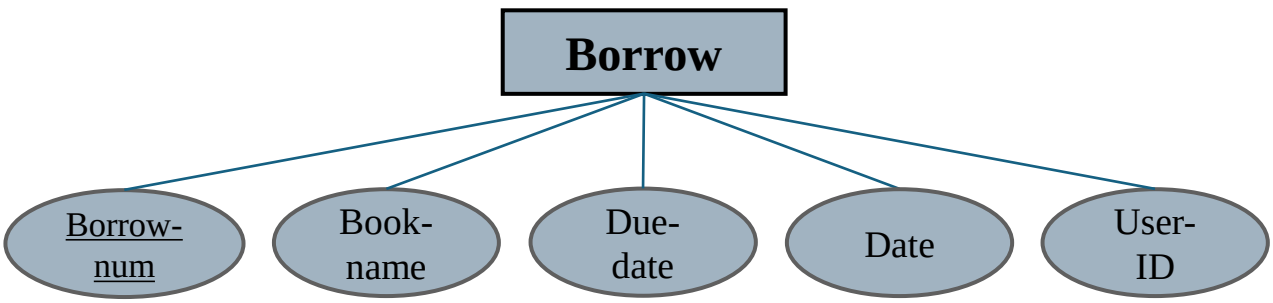
Spring 프로젝트 설계를 위한 DB 모델링

과 목 :	데이터베이스이론및실습
제출일자 :	2024.05.02
담당교수 :	
학 과 :	정보통신공학전공
학 번 :	
이 름 :	성연서

도서 추천 데이터베이스 구축

목차



DB 모델링 설계 주제 도서 대여관리 데이터베이스 설계(1)

- **제목 : 도서 대여관리 데이터베이스**
  - 사용자의 정보와 도서 대여 정보에 관한 정보를 보관하는 도서관 업무 시스템을 고려함.

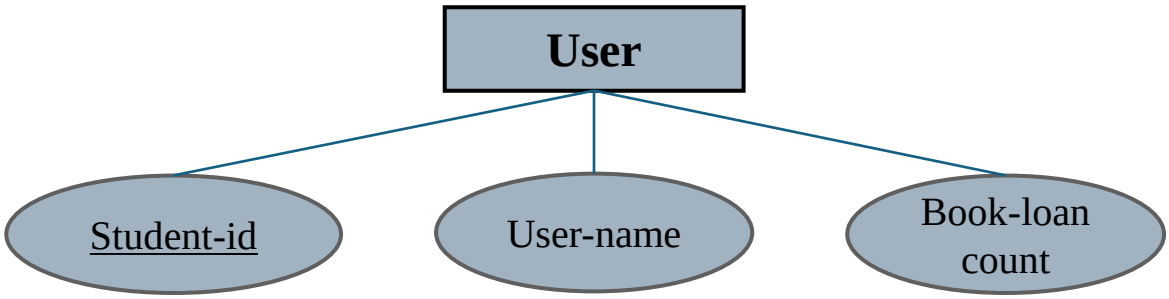
- **목적**
  - 많은 책과 사용자를 관리해야 하는 도서관에서 대여, 반납, 연체, 사용자 관리 등을 효과적으로 관리할 수 있게 하기 위해,
  - 책을 빌리는 곳이라는 도서관의 대표적 기능 뿐만이 아니라 학년별 추천 시스템을 통해 사용자의 흥미를 끌기 위해,
  - 데이터베이스 시스템을 활용해 대여, 반납의 간단한 시스템을 추천 시스템으로 연결되는 연결성을 위해,
  - 도서관 운영에 대한 데이터를 제공함으로 대여 추이, 인기 도서 등을 분석하여 도서관의 전략과 서비스를 개선하고자 한다.

- **도서관 업무**
  - 필수 요소
    - 사용자
      - 도서관 서비스를 이용할 사용자가 존재한다.
    - 도서목록
      - 사용자에게 제공할 도서가 존재한다.
    - 대여정보
      - 도서의 대여정보를 확인하고 대여할 수 있는지 알려주는 서비스가 존재한다.

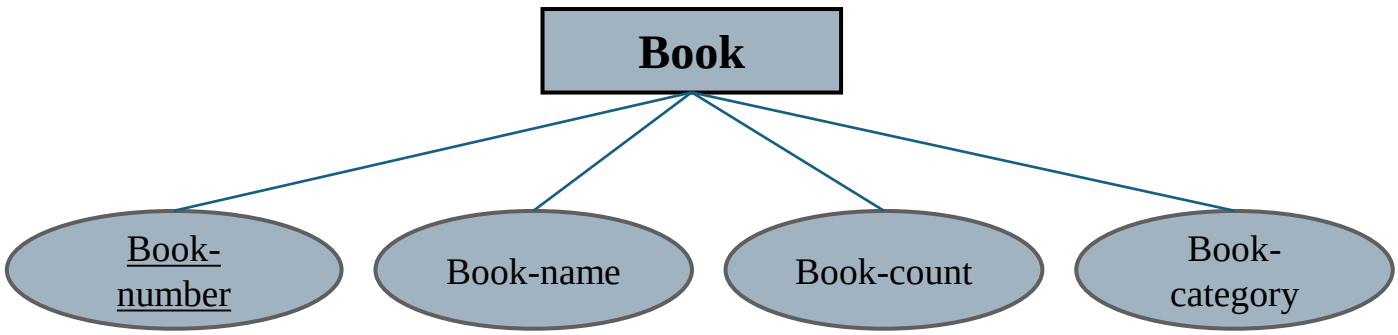
DB 모델링 Entity

도서 대여관리 데이터베이스 설계(3)

- **개체(엔티티, Entity)**
  - **사용자 (User)**
    - 학번 (Student-ID) : 주키(Primary Key)
    - 이름 (User-name)
    - 대출횟수 (Book-loan count)



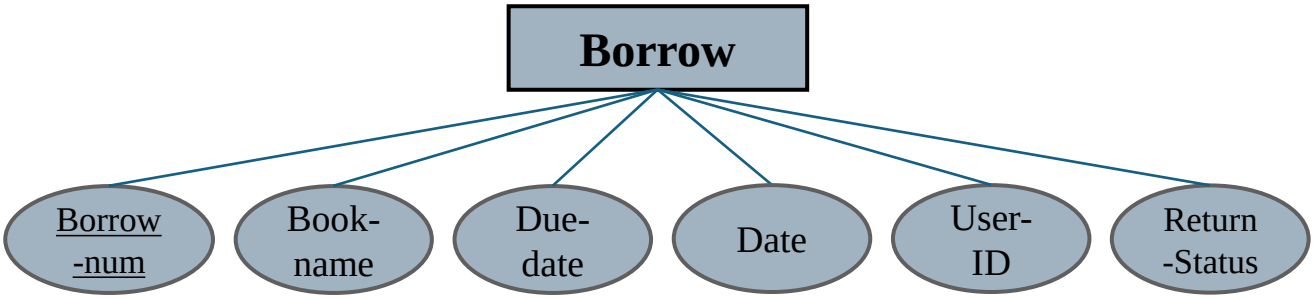
  - **도서 (Book)**
    - 도서 청구기호(Book-number) : 주키(Primary Key)
    - 도서 제목 (Book-name)
    - 도서 개수(Book-count) # 한 제목의 책이 2권 이상 있을 경우를 위해
    - 도서 주제 (Book-category)



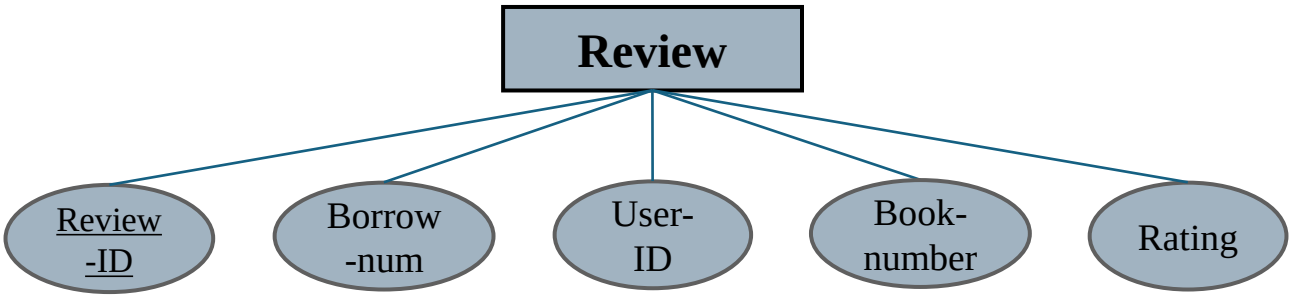
DB 모델링 Entity

도서 대여관리 데이터베이스 설계(3)

- **대출 (Borrow)**
  - 대출번호 (Borrow-num) : 주키(Primary Key)
  - 도서 제목 (Book-name) : 외래키(FK)
  - 날짜 (Date)
  - 반납 날짜 (Due-date)
  - 빌린 사용자 (User-ID) : 외래키(FK)
  - 반납여부 (Return-Status)



- **리뷰 (Review)**
  - 리뷰번호 (Review-ID) : 주키(Primary Key)
  - 대출번호 (Borrow-num) : 외래키(FK)
  - 사용자 (User-ID) : 외래키(FK)
  - 도서 청구기호 (Book-number) : 외래키(FK)
  - 평가 (Rating) #사용자의 도서 평가(예: 1~5 점)



- **관계(릴레이션, Relation)**
  - **대출 (Borrowing)**
    - 사용자가 도서관에서 책을 빌릴 때 발생하며, 한 명의 사용자가 여러 권의 책을 대출할 수 있고, 한 권의 책은 여러 번 대출될 수 있음.
    - 관련 엔티티 : 사용자(User), 도서(Book)
    - 대응수 : N 대 N (Many-to-Many)

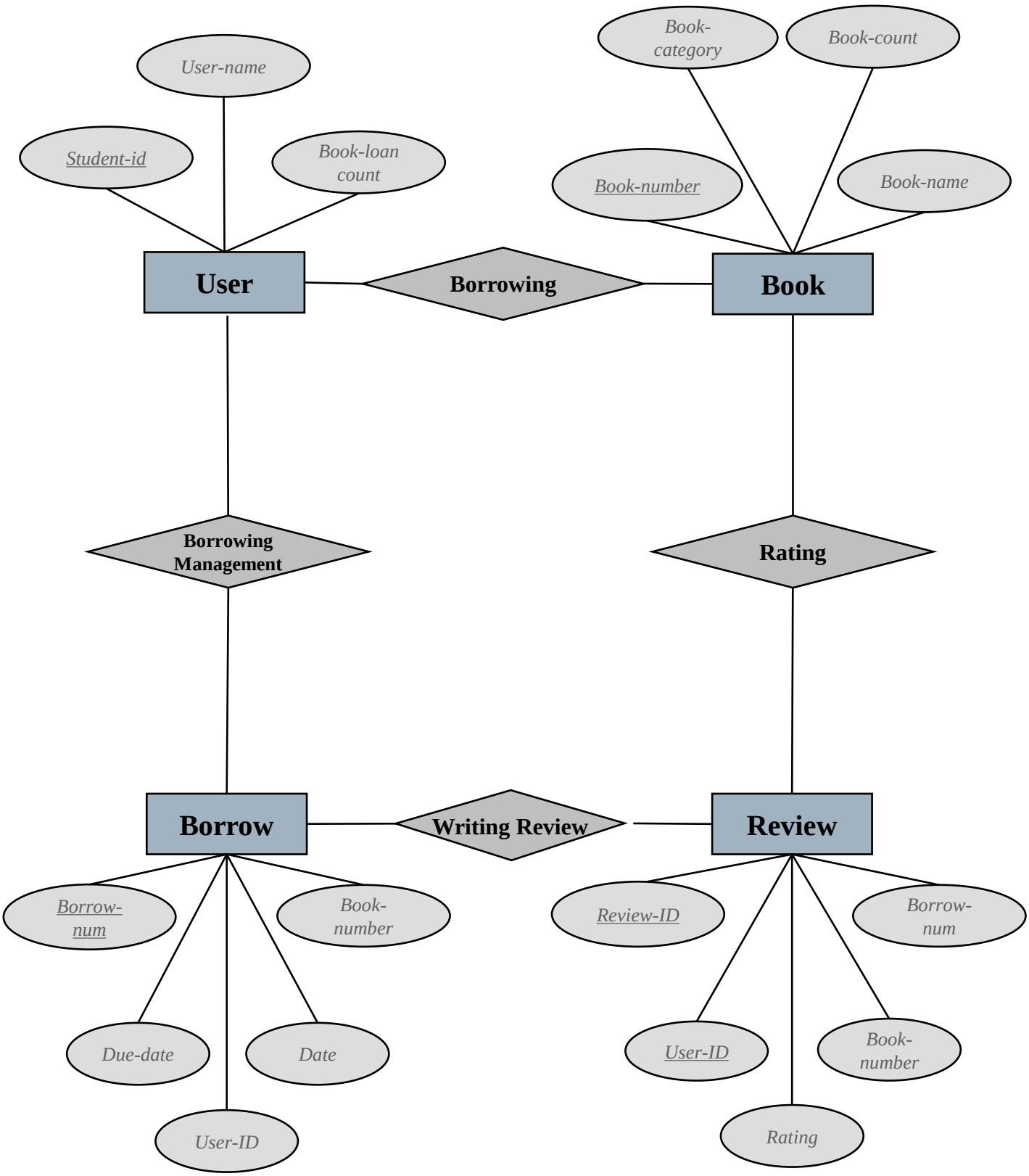
  - **리뷰 작성 (Writing Review)**
    - 사용자가 책을 빌린 후 그 책에 대한 리뷰를 작성하는 것을 의미하며, 한 번의 대출로 하나의 리뷰가 작성될 수 있음.
    - 관련 엔티티: 대출(Borrow), 리뷰(Review)
    - 대응수: 1대1 (One-to-One)

- **대출관계 (Borrowing Management)**
  - 한 명의 사용자가 여러 권의 책을 여러 번 대출할 수 있음을 의미함.
  - 엔티티: 사용자(User), 대출(Borrow)
  - 대응수: 1대 N (One-to-Many)

- **평가 (Rating)**
  - 사용자가 대출했던 책을 평가하는 것을 의미하며, 여러 리뷰가 한 권의 책을 평가할 수 있음
  - 관련 엔티티: 리뷰(Review), 도서(Book)
  - 대응수: 1대N (One-to-Many)

DB 모델링 Entity 도서 대여관리 데이터베이스 설계(3)

E-R 다이어그램



DB 모델링 Entity 도서 대여관리 데이터베이스 설계(3)

1차 테이블 구성

- user table

사용자 번호	이름	대출횟수
student_id	user_name	book_loancount
char(9)	varchar(10)	int(10)

- book table

도서 번호	제목	동일도서 번호	주제
book_number	book_name	book_count	book_category
varchar(20)	varchar(20)	int(10)	varchar(10)

- borrow table

날짜	대출번호	도서 번호	사용자 번호	반납기한
today_date	borrow_number	book_number	student_id	due_date
datetime	char(9)	varchar(20)	char(9)	datetime

- review table

리뷰번호	대출번호	사용자 번호	도서 번호	평가
review_num	borrow_number	student_id	book_number	rating
datetime	number(9)	varchar(20)	number(9)	number

1차 테이블 예시

- user table

student_id	*user_name*	*book_loancount*
202123972	Yeonseo	10
202216724	Eunha	13
202134687	Umji	5
202031484	Sinbi	12
202022141	Sowon	17
202216495	Yuju	1
202111643	Yerin	16
202134681	Hani	7
202031646	Jone	20
202216425	Curry	13

1차 테이블 예시

- book table

book_number	*book_name*	*book_count*	*book_category*
100001	1984	1	소설
100002	데미안	5	소설
100003	모비딕	2	소설
100004	죄와 벌	3	소설
100005	미드나잇 라이브러리	2	소설
100006	인류의 역사	5	역사
100007	동아시아 역사의 기원	4	역사
100008	역사의 역사	3	역사
100009	시대를 뛰어넘는 역사	3	역사
100010	확률론의 시작	2	과학
100011	우주를 보는 눈	3	과학
100012	인간 본성의 새로운	2	과학
100013	역사	1	과학
100014	사회인과 기계	4	자기계발
100015	좋은 사람이 되는 법	2	자기계발
100016	사소한 일의 기쁨	2	자기계발
100017	행복의 과학	3	자기계발
100018	관계의 기쁨	1	경제
100019	마이크로소프트 웨이	2	경제
100020	경영의 미학	3	IT
	운영체제		

1차 테이블 예시

- borrow table

today_date	*borrow_number*	*book_number*	*student_id*	*due_date*
20240401	b00001	100002	202215724	20240408
20240401	b00002	100002	202216495	20240408
20240415	b00003	100012	202123972	20240422
20240416	b00004	100019	202134681	20240423
20240422	b00005	100014	202216425	20240429
20240426	b00006	100020	202216495	20240503
20240429	b00007	100005	202022141	20240506
20240429	b00008	100014	202215724	20240506
20240506	b00009	100015	202031646	20240513
20240507	b00010	100020	202031646	20240514

- review table

review_num	*borrow_number*	*book_number*	*student_id*	*rating*
r00001	b00001	100002	202215724	5
r00002	b00002	100002	202216495	4
r00003	b00003	100012	202123972	5
r00004	b00004	100019	202134681	4.5
r00005	b00005	100014	202216425	4.5
r00006	b00006	100020	202216495	4.0
r00007	b00007	100005	202022141	3.5
r00008	b00008	100014	202215724	4.5
r00009	b00009	100015	202031646	5
r00010	b00010	100020	202031646	5

비즈니스 로직(API)

도서 대여관리 데이터베이스 설계(3)

User 릴레이션에서 User-ID가 “2021”로 시작하는 모든 학생들의 이름을 찾아라.

```
SELECT user_name FROM user
   WHERE student_id LIKE '2021%';
```

Borrow 릴레이션에서 User-ID가 “2021”로 시작하고 Date가 24년 4월인 모든 대출 기록을 찾아라.

```
SELECT * FROM borrow
   WHERE user_id LIKE '2021%' AND DATEPART(YEAR, today_date) = 2024
AND DATEPART(MONTH, today_date) = 4;
```

Book 릴레이션에서 Book-category를 중복을 제외하고 찾아라.

```
SELECT DISTINCT book_category
FROM book;
```

비즈니스 로직(API)

도서 대여관리 데이터베이스 설계(3)

Book 릴레이션에서 Book-category 속성에서 중복을 제외하고 Book-category 별 책의 수를 구하라.

```
SELECT book_category, COUNT(*) AS book_count
FROM book
GROUP BY book-category;
```

Borrow 릴레이션에서 24년 4월 동안 대출된 도서별 횟수 중에서 상위 10개를 조회해라.

```
SELECT book_number, COUNT(*) AS borrow_count
FROM borrow
WHERE DATEPART(YEAR, today_date) = 2024 AND DATEPART(MONTH, today_date) = 4
GROUP BY Book-number
ORDER BY COUNT(*) DESC
LIMIT 10;
```

비즈니스 로직(API) 도서 대여관리 데이터베이스 설계(3)

User-name이 "성연서"인 사용자의 가장 최근 대출 도서의 카테고리를 조회해라.

```
SELECT b.book_category
FROM borrow br
JOIN book b ON br.book_number = b.book_number
JOIN user u ON br. student_id = u. student_id
WHERE u.user_name = '성연서'
ORDER BY br.today_date DESC
LIMIT 1;
```

User-name이 "성연서"인 사용자와 User-id의 앞 4자리가 같은 User들의 최근 한달 간 대출 도서의 카테고리를 조회해라.

```
SELECT b.book_category
FROM borrow br
JOIN book b ON br.book_number = b.book_number
JOIN user u ON br.student_id = u.student_id
WHERE (u.user_name = '성연서' OR LEFT(u.student_id, 4) = LEFT((SELECT
student_id FROM user WHERE user-name = '성연서'), 4))
AND br.today_date >= DATEADD(MONTH, -1, GETDATE())
```

비즈니스 로직(API) 도서 대여관리 데이터베이스 설계(3)

최근 한달 간 대출 도서의 카테고리별 합을 구하고 그 중에서 상위 3개의 카테고리를 조회하여라.

```
SELECT b.book_category, COUNT(*) AS Category_Count
FROM borrow br
JOIN book b ON br.book_number = b. book_number
WHERE br.today_date >= DATEADD(MONTH, -1, GETDATE())
GROUP BY b. book_category
ORDER BY Category_Count DESC
LIMIT 3;
```

User-name이 "성연서"인 사용자와 User-id의 앞 4자리가 같은 User가 작성한 리뷰중에서 책의 평균 rating이 4.5점 이상인 책의 제목을 조회해라.

```
SELECT b.book_name
FROM review r
  JOIN book b ON r.book_number = b. book_number
  JOIN user u ON r.student_id = u. student_id
  WHERE (u.user_name = '성연서' OR LEFT(u. student_id , 4) = LEFT((SELECT
student_id  FROM user WHERE user_name = '성연서'), 4))
GROUP BY b.book_name
HAVING AVG(r.rating) >= 4.5;
```

비즈니스 로직(API) 도서 대여관리 데이터베이스 설계(3)

Rivew 릴레이션에서 도서별 평균 rating을 구하고, 평균 점수가 높은 순서대로 도서 청구기호, 제목, 주제를 조회해라.

```
SELECT b.book_number, b.book_name, b.book_category, AVG(r.rating) AS
Avg_Rating
FROM review r
JOIN book b ON r.book_number = b.book_number
GROUP BY b.book_number, b.book_name, b.book_category
ORDER BY Avg_Rating DESC;
```