데이터베이스설계및활용 개인과제 #2

20192208 김형훈

1번 문항

BCNF 정규화

다음에 제시된 BOOK_RENTAL 테이블을 BCNF로 만들어 보아라.

BOOK_RENTAL(대여ID, 회원ID, 회원이름, 도서ID, 도서제목, 저자, 출판사, 대여날짜, 반납날짜)

대여ID	회원ID	회원이름	도서ID	도서제목	저자	출판사	대여날짜	반납날짜
101	1001	김철수	B001	데이터베이스 개론	홍길동	미래출판사	2024-10-01	2024-10-08
102	1002	이영희	B002	알고리즘 입문	최영	정보서적	2024-10-02	2024-10-09
103	1001	김철수	B003	컴퓨터 네트워크	박찬호	네트워크북스	2024-10-03	2024-10-10
104	1003	박민수	B004	파이썬 프로그래밍	이수정	프로그래밍북	2024-10-05	2024-10-12
105	1004	최지현	B005	머신러닝 이론	정다은	데이터사이언스	2024-10-06	2024-10-13
106	1002	이영희	B001	데이터베이스 개론	홍길동	미래출판사	2024-10-07	2024-10-14
107	1005	오지훈	B006	빅데이터 분석	김민호	분석출판사	2024-10-08	2024-10-15
108	1006	송혜린	B007	클라우드 컴퓨팅	조영호	클라우드북스	2024-10-09	2024-10-16
109	1001	김철수	B002	알고리즘 입문	최영	정보서적	2024-10-10	2024-10-17
110	1003	박민수	B008	자바스크립트 마스터	이현우	웹개발서적	2024-10-11	2024-10-18

Boyce-Codd Normal Form에서 각각의 relation은 오직 하나의 theme을 가져야 한다.

Boyce-Codd 판별 과정을 거친 후, 제시된 테이블을 정규화해보도록 하겠다.

1. 모든 Candidate Key를 찾는다.

대여ID, (회원ID, 도서ID, 대여날짜)가 Candidate Key가 될 수 있다.

2. 모든 Functional Dependency를 찾는다.

- 대여ID -> (회원ID, 회원이름, 도서ID, 도서제목, 저자, 출판사, 대여날짜, 반납날짜)
- 회원ID -> (회원 이름)
- 도서ID -> (도서제목, 저자, 출판사)

3. functional dependency의 determinants를 검토한다.

회원ID, 도서ID는 candidate key가 아니므로, BCNF가 아니다.

따라서 다음의 과정을 거쳐 BCNF로 정규화한다.

A. Functional dependency의 columns를 새로운 relation에 넣는다.

- MEMBER (회원ID, 회원이름)
- BOOK (도서ID, 도서제목, 저자, 출판사)

B. functional dependency의 determinant를 새로운 relation의 primary key로 설정한다.

- MEMBER (회원ID, 회원이름)
- BOOK (도서ID, 도서제목, 저자, 출판사)

C. 기존의 relation에 determinant의 copy를 foreign key로 추가한다.

- BOOK_RENTAL(대여ID, 회원ID, 도서ID, 대여날짜, 반납날짜)

D. 기존의 relation과 새로운 relation 간의 referential integrity constraint를 생성한다.

- 회원ID in BOOK_RENTAL must exist in 회원ID in MEMBER
- 도서ID in BOOK_RENTAL must exist in 도서ID in BOOK

4. 모든 relation의 모든 determinant가 candidate key가 될 때까지 반복한다.

회원ID와 도서ID 관련 functional dependency를 새로운 relation으로 생상하고 과정을 종료한다.

정규화 과정을 거친 후의 TABLE

BCNF로 정규화한 후, TABLE은 다음과 같이 나타난다.

BOOK_RENTAL

대여ID	회원ID	도서ID	대여날짜	반납날짜
101	1001	B001	2024-10-01	2024-10-08
102	1002	B002	2024-10-02	2024-10-09
103	1001	B003	2024-10-03	2024-10-10
104	1003	B004	2024-10-05	2024-10-12
105	1004	B005	2024-10-06	2024-10-13
106	1002	B001	2024-10-07	2024-10-14
107	1005	B006	2024-10-08	2024-10-15
108	1006	B007	2024-10-09	2024-10-16
109	1001	B002	2024-10-10	2024-10-17
110	1003	B008	2024-10-11	2024-10-18

MEMBER

회원ID	회원이름
1001	김철수
1002	이영희
1003	박민수
1004	최지현
1005	오지훈
1006	송혜린

BOOK

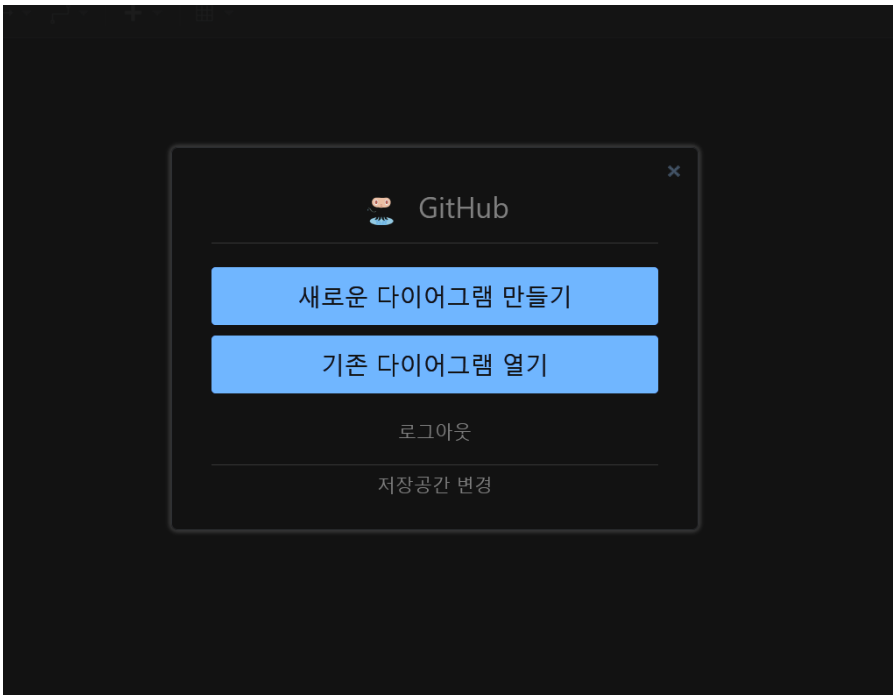
도서ID	도서제목	저자	출판사
B001	데이터베이스 개론	홍길동	미래출판사
B002	알고리즘 입문	최영	정보서적
B003	컴퓨터 네트워크	박찬호	네트워크북스
B004	파이썬 프로그래밍	이수정	프로그래밍북
B005	머신러닝 이론	정다은	데이터사이언스
B006	빅데이터 분석	김민호	분석출판사
B007	클라우드 컴퓨팅	조영호	클라우드북스
B008	자바스크립트 마스터	이현우	웹개발서적

2번 문항

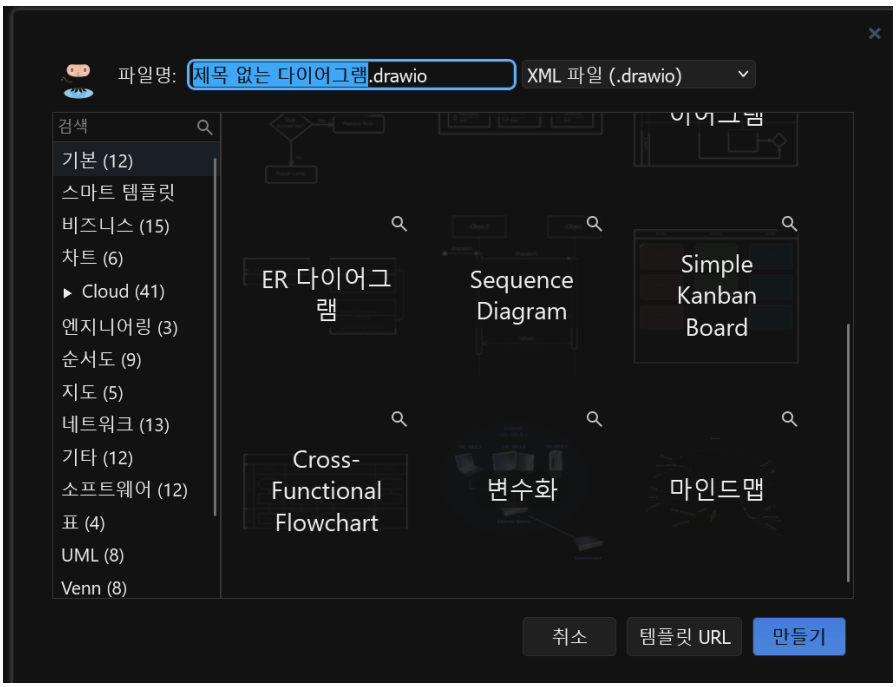
E-R 모델링

기본 전제

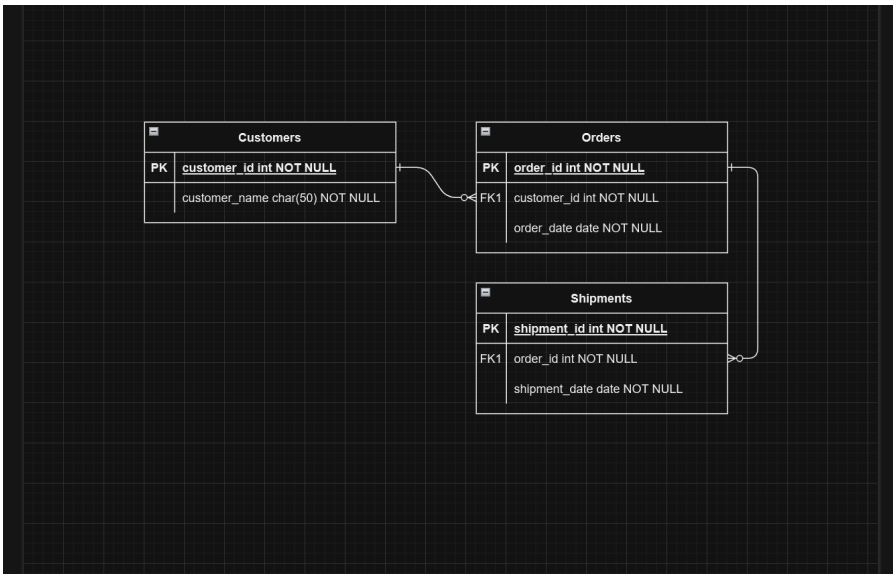
1번 문항에서 정규화한 TABLE로 ER Diagram을 작성해보면 다음과 같다.



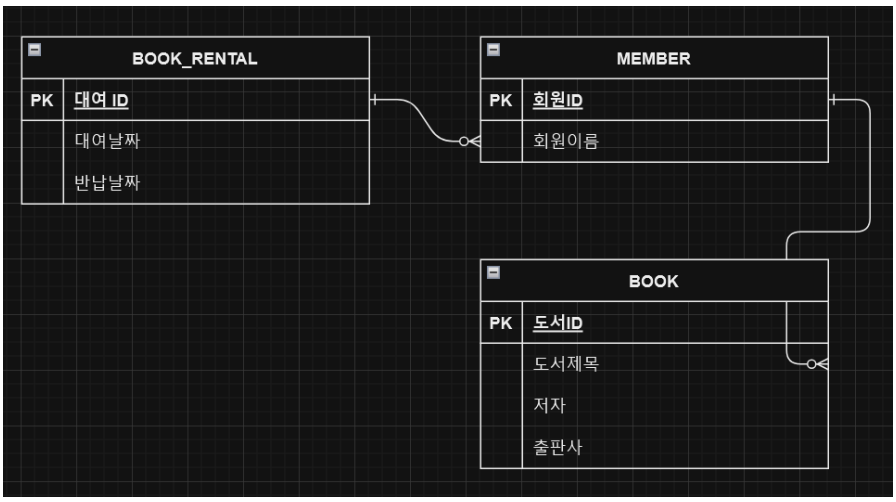
먼저 draw.io에 접속해서 새로운 diagram 만들기를 선택한다.



ER Diagram template으로 선택하겠다.



기본 화면. 이제 1번 문항의 TABLE을 생성해보겠다.

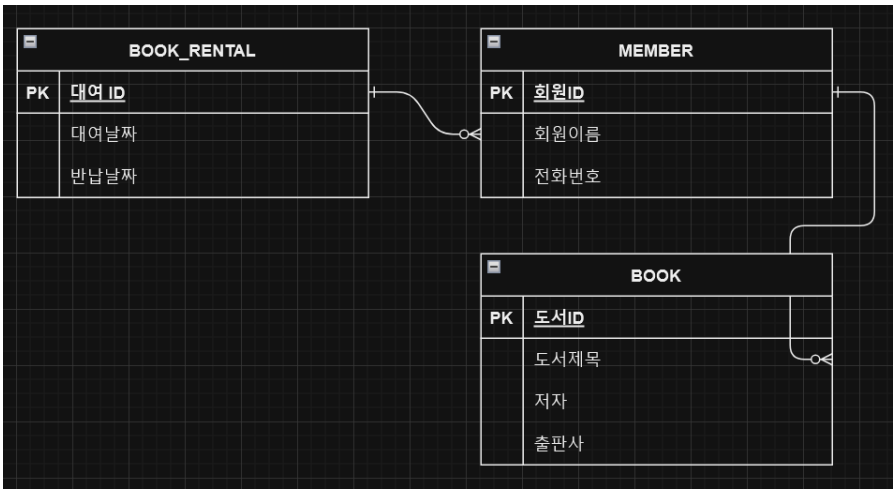


Entity 이름과 Attribute 값만 채워 넣어줬다.

개체 및 속성 식별

1번 문항에서 제시된 속성 이외에 필요하다고 판단되는 속성을 추가해주겠다.

MEMBER가 책을 빌리고 반납을 하지 않는 불상사를 대비하여, MEMBER의 Entity에 전화번호 attribute를 추가해주겠다.

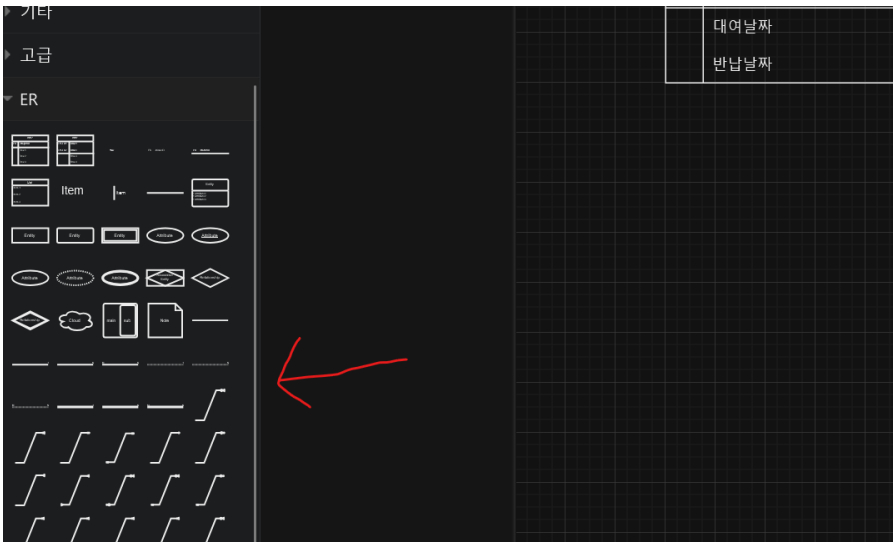


알맞게 설정이 된 모습.

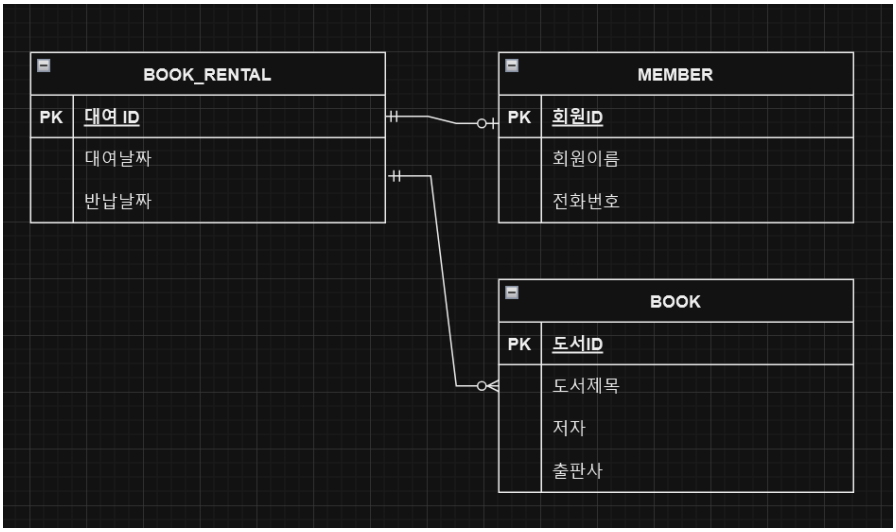
Strong/Weak Entity 및 Identifying/Non-Identifying Relationship을 표시함

이제 relationship을 추가해보겠다.

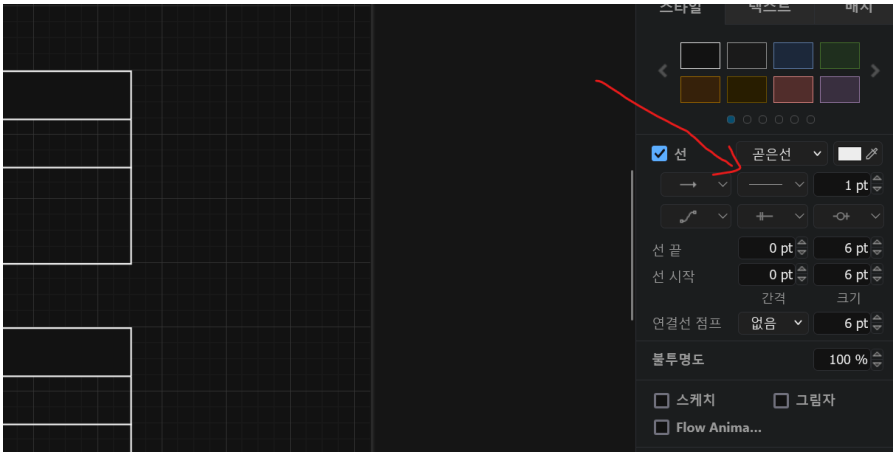
세개의 테이블 중, 다른 테이블의 identifier를 identifier로 가지는 경우가 없으므로, 모든 entity를 strong entity로 설정하고, non-identifying relationship으로 설정하겠다.



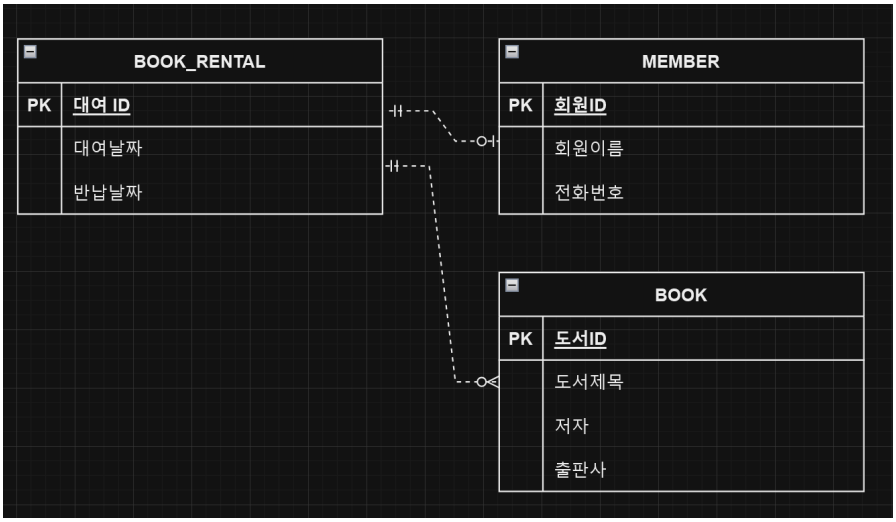
왼쪽 창에서 관계를 추가해준다.



그러면 이렇게 두개의 relationship을 추가할 수 있다.



이제 오른쪽 창에서 실선을 점선으로 변경해, non-identifying relationship을 표현해준다.



알맞게 설정이 된 모습.

Maximum/Minimum Cardinality 요구사항을 파악하여 모델에 반영

이제 Cardinality를 알맞게 설정해주겠다.

- **BOOK_RENTAL과 MEMBER의 관계**

  BOOK_RENTAL instance는 한명의 MEMBER 정보를 가질 수 있고, MEMBER instance는 대출을 여러 번 받을 수 있다.

  따라서, 두 테이블의 **Max-Cardinality는 1:N 관계**를 가진다.

  BOOK_RENTAL instance는 MEMBER에 대한 정보가 있어야 하고, MEMBER는 대출을 안할 수 있다.

  따라서, 두 테이블의 **Min-Cardinality는 Mandatory / Optional 관계**를 가진다.

- **BOOK_RENTAL과 BOOK의 관계**

  BOOK_RENTAL instance는 한권의 BOOK 정보를 가질 수 있고, BOOK instance는 여러 번 대출될 수 있다.

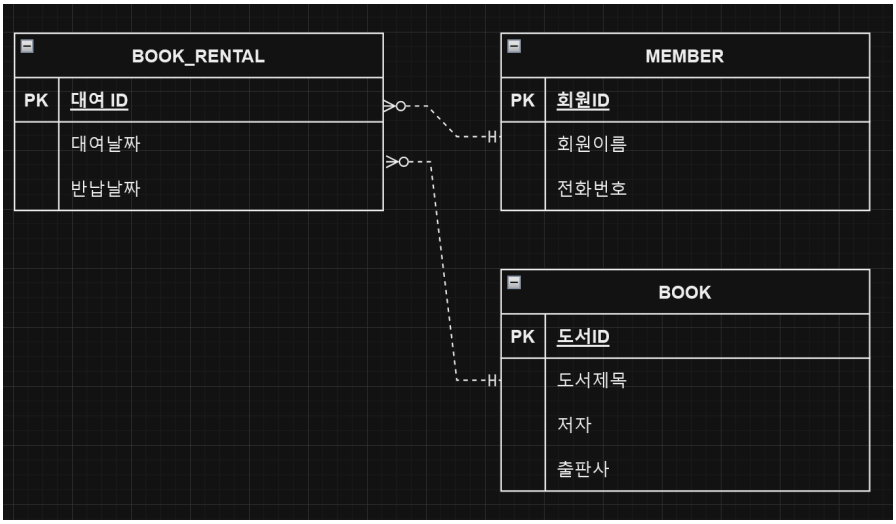
  따라서, 두 테이블의 **Max-Cardinality는 1:N 관계**를 가진다.

  BOOK_RENTAL instance는 BOOK에 대한 정보가 있어야 하고, BOOK는 대출되지 않을 수 있다.

  따라서, 두 테이블의 **Min-Cardinality는 Mandatory / Optional 관계**를 가진다.

위의 내용을 바탕으로 Cardinality를 설정해주겠다.

오른쪽 창에서 Cardinality를 설정해준다.



알맞게 설정이 된 모습.