**Database System**

Project #1 도서관 데이터베이스 설계 보고서

|  |  |
| --- | --- |
| 학과 | 컴퓨터공학과 |
| 학번 | 20181593 |
| 이름 | 계인혜 |

**<목차>**

**1. ER 다이어그램**

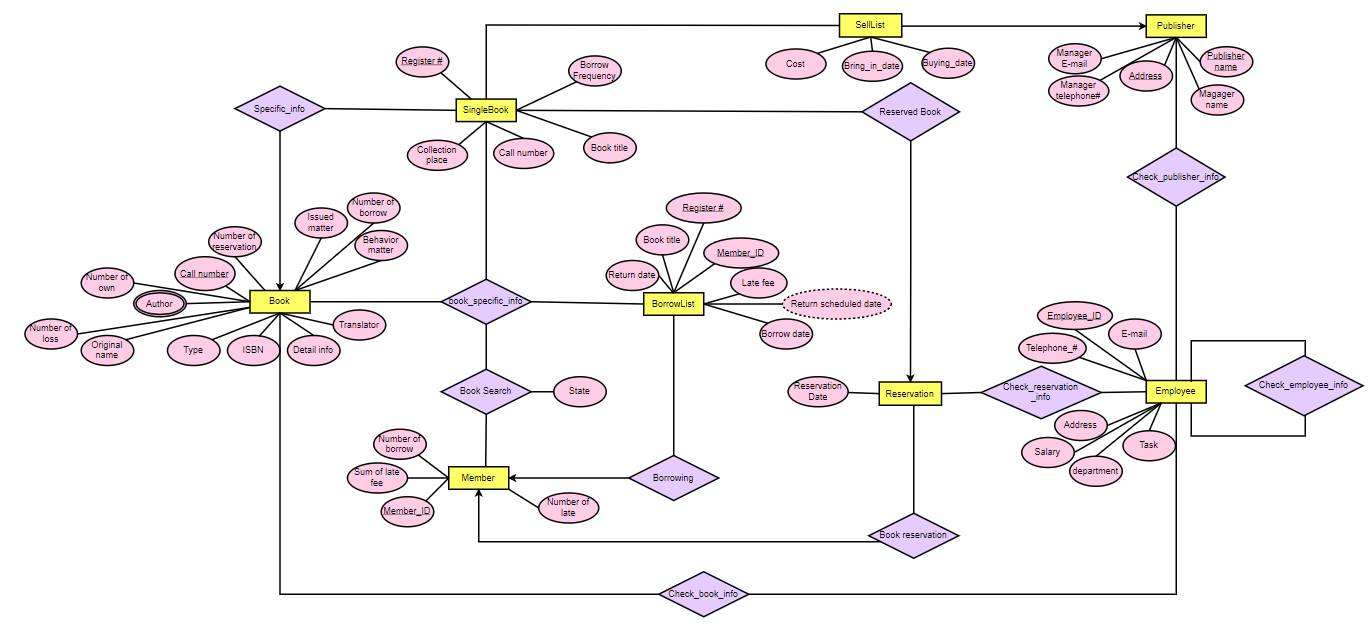
**2. Logical Mode**

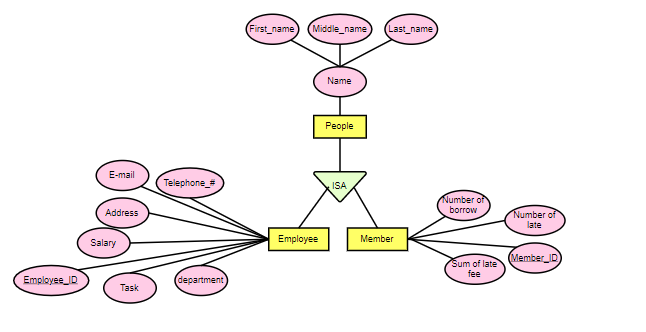
**3. Physical Mode**

**4. 시스템 요구사항 만족 관련 설명**

**5. 추가 설명**

1. **ER 다이어그램**

전체적인 ER 다이어그램은 다음과 같다.



먼저 위의 사진에는 Specialization이 없는데, 아래에만 있는 이유는 위에서 generalization을 표시하기에는 공간 배치가 애매했기 때문에 따로 표시했기 때문이다. 그러나 ERwin에서는 이를 같이 표시하였다. 그림을 보면 위의 ERD에서도 아래 사진의 Specialization을 고려하여 high level인 People Entity의 attribute인 name이 있기 때문에 Employee와 Member에는 Name 속성이 따로 없는 것을 확인할 수 있다. 명세서를 보면 Employee와 Member는 모두 이름 속성을 갖는다. 이를 이용하여 두 Entity를 People로 Generalization했기 때문에 각각의 entity가 따로 이름을 저장하지 않아도 된다.

1. ERD 세부사항 설명

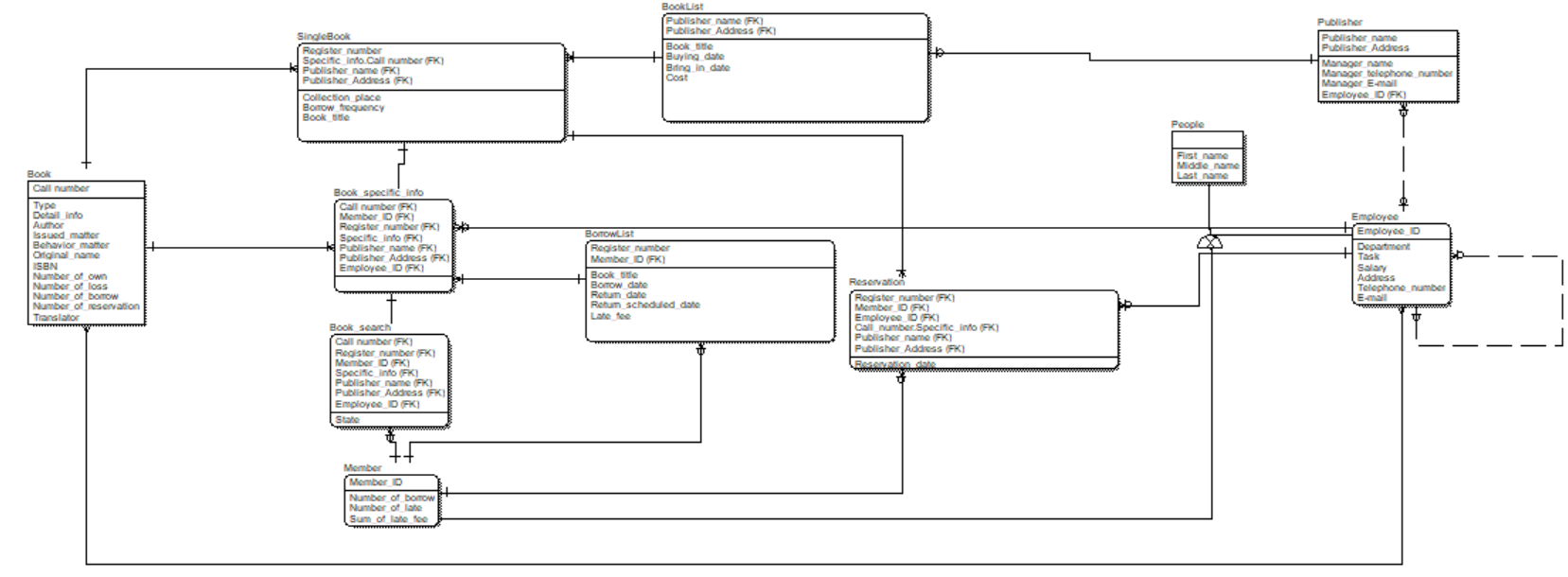
|  |  |
| --- | --- |
|  | Book entity의 author 속성을 multivalued로 설정하였다. 이로 인해 공동 저자가 있는 경우에도 저자를 무리 없이 표현할 수 있다. |
|  | BorrowList entity의 반납 예정일을derived attribute로 설정하였다. 대출일이 정해지면 그로부터 대부분 2주 후가 반납 예정일이 되기 때문이다. |
|  | Name을 component attribute로 설정하여 각각 First\_name, Middle\_name, Last\_name으로 나누어 구현하였다. |

1. **Logical Mode**
2. Entity와 각 Attributes에 대한 설명

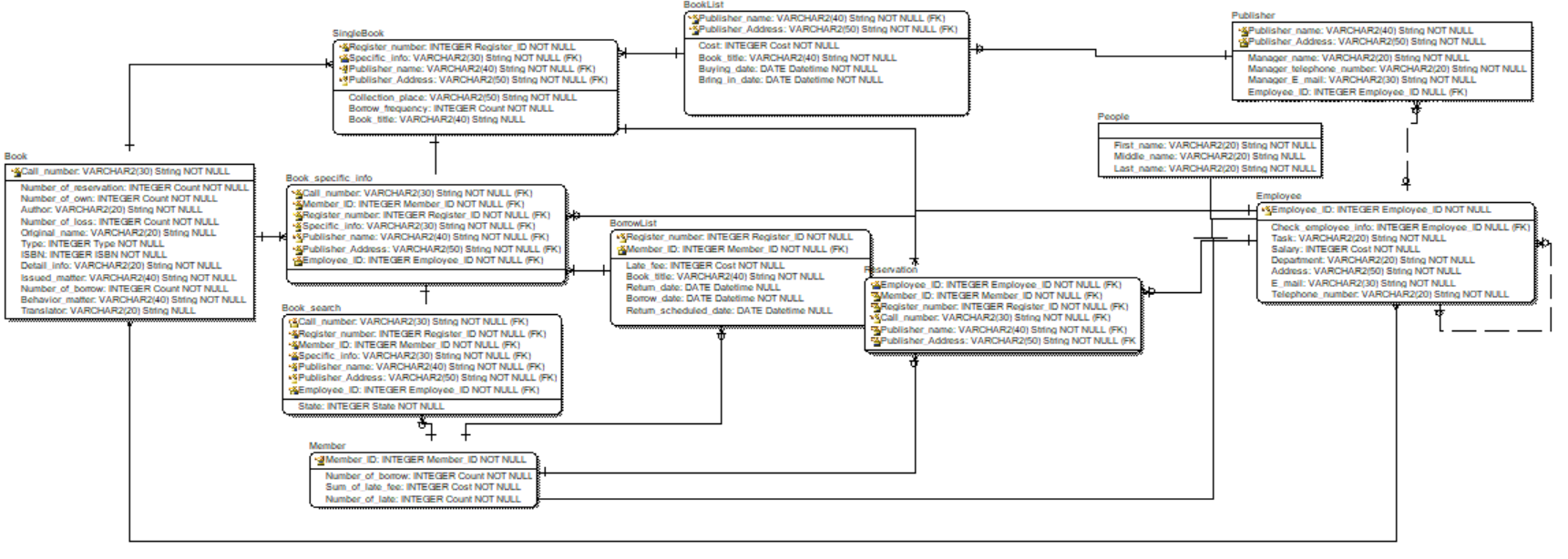
|  |  |
| --- | --- |
| **Entity** | **Attributes** |
|  | * Call number : 청구기호를 의미하며 Book enitiy의 primary key이다. * Type : 자료유형을 저장한다. * Detail\_info : 서명/저자사항을 저장한다. * Author : 저자를 저장한다. * Issued\_matter : 발행사항을 저장한다. * Behavior\_matter : 행태사항을 저장한다. * Original\_name : 필요한 경우 원서명을 저장한다. * ISBN : 국제 표준 도서 번호를 저장한다. * Number\_of\_own : 소장부수를 저장한다. * Number\_of\_loss : 분실건수를 저장한다. * Number\_of\_borrow : 대출건수를 저장한다. * Number\_of\_reservation : 예약건수를 저장한다. * Translator : 필요한 경우 옮긴이를 저장한다. |
|  | * 도서 검색 및 상세 정보 확인 화면을 위한 것으로 개별 서적, 대출 내역, 도서의 primary key를 속성으로 갖는다. |
|  | * Call number : Book entity의 attribute inheritance이며, 도서 검색 entity의 foreign key이다. * Register number : SingleBook entity의 attribute inheritance이며, 도서 검색 entity의 foreign key이다. * Member\_ID : Member entity의 attribute inheritance이며, 도서 검색 entity의 foreign key이다. * State : 도서 검색 시 도서의 상태를 저장한다.   (대출가능, 대출중, 분실, 예약) |
|  | * 출판사 정보 조회를 위한 entity로 출판사 정보 및 구매 리스트에 대한 정보를 포함한다. * Publisher\_name : Publisher entity의 attribute inheritance이며, BookList entity의 foreign key이다. * Publisher\_address : Publisher entity의 attribute inheritance이며, BookList entity의 foreign key이다. * Register\_number : SingleBook entity의 attribute inheritance이며, BookList entity의 foreign key이다. * Specific\_info : SingleBook entity의 attribute inheritance이며, BookList entity의 foreign key이다. * Book\_title : 서명을 저장한다. * Buying\_date : 구매일을 저장한다. * Bring\_in\_date : 반입일을 저장한다. * Cost : 금액을 저장한다. |
|  | * 대출 내역 조회를 위한 entity이다. * Register\_number : 등록번호를 저장한다. * Member\_ID : Member entity의 attribute inheritance이며, BorrowList entity의 foreign key이다. * Book\_title : 서명을 저장한다. * Borrow\_date : 대출일을 저장한다. * Return\_date : 반납일을 저장한다. * Return\_scheduled\_date : 반납예정일을 저장한다. * Late\_fee : 연체료를 저장한다. |
|  | * Employee\_ID : 사번을 의미하며 Employee enitiy의 primary key이다. * Department : 부서를 저장한다. * Task : 담당업무를 저장한다. * Salary : 연봉을 저장한다. * Address : 주소를 저장한다. * Telephone\_number : 전화번호를 저장한다. * E-mail : E-mail 주소를 저장한다. |
|  | * First\_name : 영문 이름을 저장한다. * Middle\_name : 미들네임이 있는 경우 이를 저장한다. * Last\_name : 영문 이름의 성을 저장한다. |
|  | * Member\_ID : 회원번호를 의미하며 Member enitiy의 primary key이다. * Number\_of\_borrow : 대출횟수를 저장한다. * Number\_of\_late : 연체횟수를 저장한다. * Sum\_of\_late\_fee : 총 연체금액을 저장한다. |
|  | * Publisher\_name : 출판사명을 의미하며 Publisher entity의 primary key이다. * Publisher\_address : 주소를 의미하며 Publisher entity의 primary key이다. 출판사 명만으로는 겹치는 경우가 있을 수도 있다고 생각하여 Entity의 uniqueness를 더욱 안정적으로 확보하기 위해 주소도 Primary key로 설정하게 되었다. * Manager\_name : 담당자명을 저장한다. * Manager\_telephone\_number : 전화번호를 저장한다. * Manager\_e\_mail : 담당직원의 이메일을 저장한다. * Employee\_ID : Employee entity의 attribute inheritance이며, Publisher entity의 foreign key이다. 담당직원의 사번을 의미한다. |
|  | * 도서 별 예약 정보를 위한 entity이다. * Register\_number : SingleBook entity의 attribute inheritance이며, Reservation entity의 foreign key이다.등록번호를 의미한다. * Member\_ID : Member entity의 attribute inheritance이며, Reservation entity의 foreign key이다. 회원번호를 의미한다. * Employee\_ID : Employee entity의 attribute inheritance이며, Reservation entity의 foreign key이다. * Call\_number : SingleBook entity의 attribute inheritance이며, Reservation entity의 foreign key이다. * Reservaton\_date : 예약일을 저장한다. |
|  | * 개별 서적을 위한 entity이다. * Register\_number : 등록번호를 의미하며 SingleBook entity의 primary key이다. * Call\_number : Book entity의 attribute inheritance이며, SingleBook entity의 foreign key이다. * Collection\_place : 소장처를 저장한다. * Borrow\_frequency : 대출빈도를 저장한다. * Book\_title : 서명을 저장한다. |

1. Entity간의 Relation과 mapping cardinality에 대한 설명

|  |  |
| --- | --- |
|  | |
| * 도서(Book)와 개별서적(SingleBook) 사이의 cardinality는 1:N인데, 한 도서는 여러 개의 개별 서적을 가질 수 있기 때문이다. Relation type은 identifying으로 이후에 서적 별 상세 정보 조회를 위해 도서의 청구기호를 Foreign key로 설정하였다. * Book\_specific\_info entity는 relationship 사이의 관계를 정의하기 위한 aggregation으로 도서 검색(Book\_search) relationship과 서적 별 상세 정보(Book\_specific\_info) relationship 사이의 관계를 정의한다. * 서적별 상세 정보(Book\_specific\_info)와 개별서적(SingleBook) 사이의 cardinality는 1:1인데, 개별 서적과 상세 정보는 하나씩 짝지어지기 때문이다. Relation type은 identifying으로 서적 별 상세 정보 조회를 위해 BorrowList의 등록번호를 Foreign key로 설정하였다. * 서적별 상세 정보(Book\_specific\_info)와 대출내역(BorrowList) 사이의 cardinality는 1:N인데, 한 대출 내역에는 여러 개의 개별 도서가 포함될 수 있기 때문이다. 즉, 서적 별 상세 정보도 여러 개가 있을 수 있다는 것이다. Relation type은 identifying으로 대출 내역 정보 조회를 위해 BorrowList의 회원번호를 Foreign key로 설정하였다. * 서적별 상세 정보(Book\_specific\_info)와 도서(Book) 사이의 cardinality는 N:1인데, 프로젝트 명세서를 보면 서적 별 상세 정보에 청구기호는 같지만 등록 번호가 다른 서적이 여러 개 출력되는 것으로 보아 도서 하나가 여러 개의 상세 정보를 가질 수 있다고 판단했다. | |
|  | |
| * 회원(Member)와 도서 검색(Book\_search) 사이의 cardinality는 1:N인데, 이는 한 회원이 도서 검색을 여러 번 할 수 있기 때문이다. 또한 회원이 도서 검색을 하지 않을 수도 있기 때문에 zero를 허용한다. Relation type은 identifying으로 대출 내역 정보 조회를 위해 Member의 회원번호를 Book\_search의 Foreign key로 설정하였다. * 회원(Member)와 대출 내역(BorrowList) 사이의 cardinality는 1:N인데, 이는 한 회원은 여러 개의 도서 대출이 가능하기 때문이다. 또한 회원이 도서 대출을 하지 않을 수도 있기 때문에 zero를 허용한다. Relation type은 identifying으로 대출 내역 정보 조회를 위해 Member의 회원번호를 BorrowList의 Foreign key로 설정하였다. | |
|  | |
| * 도서(Book)와 직원(Employee) 사이의 cardinality는 N:M인데, 이는 한 직원이 도서 정보 입력 및 수정을 여러 번 할 수 있기 때문이다. Relation type은 identifying으로 설정하였다. | |
|  | |
| * 구매내역(BookList)와 출판사(Publisher) 사이의 cardinality는 1:N인데, 이는 한 출판사가 도서관에 여러 개의 도서 판매를 할 수 있기 때문이다. 또한 한 출판사로부터 어떤 도서도 구매하지 않을 수 있기 때문에 zero를 허용한다. Relation type은 identifying으로 출판사 별 구입 도서 목록 조회를 위해 Publisher의 이름과 주소를 BookList의 Foreign key로 설정하였다. | |
|  | |
| * 구매내역(BookList)와 개별서적(SingleBook) 사이의 cardinality는 N:1인데, 이는 한 구매 내역이 여러 개의 개별 서적을 포함할 수 있기 때문이다. Relation type은 identifying으로 설정하였다. | |
|  | |
| * 회원(Member)과 예약(Reservation) 사이의 cardinality는 1:N인데, 이는 한 여러 개의 도서를 예약할 수 있기 때문이다. 또한 회원이 어떤 도서도 예약하지 않을 수 있기 때문에 zero를 허용한다. Relation type은 identifying으로 도서 별 예약 정보 조회를 위해 Member의 회원번호를 Reservation의 Foreign key로 설정하였다. | |
|  | | * 직원이 전체 직원 리스트 조회를 할 수 있도록 하기 위해 재귀적인 relationship을 설정하였다. 한 직원이 직원 리스트 조회를 하지 않을 수 있기 때문에 zero를 허용하였으며, 반대로 한 직원이 여러 명의 개별 직원 정보를 확인할 수 있기 때문에 cardinality는 1:N으로 설정하였다. Relationship은 non-identifying으로 설정하였다. |
|  | | * 직원이 출판사 정보 조회를 할 수 있도록 하기 위한 것으로 직원(Employee)와 출판사(Publisher) 사이의 cardinality는 1:N이다. 한 직원이 출판사 리스트 조회를 하지 않을 수 있기 때문에 zero를 허용하였으며, 반대로 한 직원이 여러 출판사의 개별 정보를 확인할 수 있기 때문이다. Relationship은 non-identifying으로 설정하였다. 직원이 없어도 출판사는 존재할 수 있으며 출판사가 없어도 직원이 존재할 수 있기 때문이다. |

1. Logical view
2. **Physical Mode**
3. 속성 값에 대한 자료형 및 제약사항 설정에 대한 설명

|  |  |
| --- | --- |
| **Entity** | **Attribute type & constraint** |
|  | * Call number : 청구기호의 도메인은 string으로 길이가 꽤 길 것 같아 30자를 최대로 두었으며, primary key이기 때문에 NULL을 허용하지 않았다. * Number\_of\_own, Number\_of\_loss, Number \_of\_borrow, Number\_of\_ reservation :   횟수를 나타내는 속성들은 0미만일 수 없기 때문에 Count라는 새로운 정수형 도메인을 생성하여 설정했다. Count는 최솟값이 0인 정수형 타입이다. 모두 NULL을 허용하지 않았다.   * Type : 자료유형은 단행본과 정기간행물 2가지로만 표시되기 때문에 Type이라는 새로운 정수형 도메인을 생성하여 설정했다. 0일 때는 단행본을, 1일 때는 정기 간행물을 의미한다. * Detail\_info, Author : 두 속성의 도메인은 string으로 20자를 최대 길이로 두었으며, NULL을 허용하지 않았다. * Issued\_matter, Behavior\_matter : 두 속성의 도메인은 string으로 길이가 꽤 길 것 같아 40자를 최대로 두었으며, NULL을 허용하지 않았다. * ISBN : 13자리 숫자로 표현하기 위해 ISBN이라는 새로운 정수형 도메인을 만들었다. Constraint는 min/max를 각각 0000000000000, 9999999999999로 설정하여 13자리 조건을 충족할 수 있었다. NULL은 허용하지 않았다. * Orinial\_name, Translator : 두 속성의 도메인은 string으로 20자를 최대 길이로 두었다. 명세서에는 원서명이 필요한 경우만 영문으로 표기하고, 필요한 경우 옮긴이를 보여준다고 명시되어 있기 때문에 두 속성은 NULL값을 허용하였다. |
|  | * State : 도서 상태는 다음과 같이 4가지 상태로만 표현되기 때문에 이를 효율적으로 다루기 위하여 State라는 새로운 정수형 도메인을 만들었다. 0일때는 대출가능, 1일때는 대출 중, 2일때는 분실, 3일때는 예약을 나타낸다. 모든 책은 다음 4가지 상태에 항상 속하게 되어있으므로 NULL값을 허용하지 않았다.   (대출가능, 대출 중, 분실, 예약) |
|  | * Cost : 도서 구입 비용을 나타내는 cost는 비용이므로 음의 값을 가질 수 없다. 따라서 이를 반영한 Cost라는 새로운 정수형 도메인을 만들었으며, 단위는 1원이다. 값이 없는 경우는 없으므로 NULL 값을 허용하지 않았다. * Book\_title : 도메인은 string으로 길이가 꽤 길 것 같아 40자를 최대로 두었으며, NULL을 허용하지 않았다. * Buying\_date, Bring\_in\_date : 두 속성은 날짜를 표시하기 위한 Datetime 도메인으로 설정하였으며, NULL을 허용하지 않았다. |
|  | * Late\_fee : 연체료를 나타내는 late\_fee는 비용이므로 음의 값을 가질 수 없다. 따라서 앞에서 언급한 Cost라는 도메인을 설정하였으며 단위는 마찬가지로 1원이다. 연체료가 없는 경우에도 0원으로 표시하기 때문에 NULL값은 허용하지 않았다. * Book\_title : 도메인은 string으로 길이가 꽤 길 것 같아 40자를 최대로 두었으며, NULL을 허용하지 않았다. * Borrow\_date, Return\_date, Return\_ scheduled\_date : 세 속성은 날짜를 표시하기 위한 Datetime 도메인으로 설정하였으며, 대출일은 NULL을 허용하지 않았다. 한편, 반납일은 현재도 대출 중인 경우 표시하지 않고, 반납 예정일은 반납이 완료되면 표시하지 않기 때문에 NULL을 허용하였다. |
|  | * Employee\_ID : 직원에게 부여되는 고유한 6자리 사번을 의미하며 이를 구현하기 위해 Employee\_ID라는 새로운 정수형 도메인을 만들었으며 MIN/MAX 값을 000000, 999999로 설정하여 조건을 충족할 수 있었다. 또한 Employee enitiy의 primary key이기 때문에 NULL을 허용하지 않았다. * Department, Task, Telephone\_number : 두 속성의 도메인은 string으로 20자를 최대 길이로 두었다. Null 값은 허용하지 않았다. * Salary : 연봉은 비용이므로 음의 값을 가질 수 없다. 따라서 앞에서 언급한 Cost라는 도메인을 설정하였으며 단위는 마찬가지로 1원이다. NULL값은 허용하지 않았다. * Address, E-mail : 두 속성의 도메인은 string으로 각각 최대 길이를 50, 30 자로 설정하였다. Null값은 허용하지 않았다. |
|  | * First\_name, Middle\_name, Last\_name : 세 속성의 도메인은 string으로 20자를 최대 길이로 두었다. First\_name, Last\_name은 NULL 값을 허용하지 않았지만 외국인이 아닌 이상 Middle\_name은 없을 수도 있기 때문에 Middle\_name은 예외적으로 NULL값을 허용하였다. |
|  | * Member\_ID : 회원에게 부여되는 고유한 8자리 숫자를 의미하며 이를 구현하기 위해 Member\_ID라는 새로운 정수형 도메인을 만들었으며 MIN/MAX 값을 00000000, 99999999로 설정하여 조건을 충족할 수 있었다. 또한 Member enitiy의 primary key이기 때문에 NULL을 허용하지 않았다.   Number\_of\_borrow, Number\_of\_late : 횟수를 나타내는 속성들은 0미만일 수 없기 때문에 Count라는 새로운 정수형 도메인을 생성하여 설정했다. Count는 최솟값이 0인 정수형 타입이다. 모두 NULL을 허용하지 않았다.   * Sum\_of\_late\_fee : 총 연체금액은 비용이므로 음의 값을 가질 수 없다. 따라서 앞에서 언급한 Cost라는 도메인을 설정하였으며 단위는 마찬가지로 1원이다. 연체를 한번도 하지 않은 경우에도 총 연체금액은 0이기 때문에 NULL값은 허용하지 않았다. |
|  | * Publisher\_name, Publisher\_Address : 두 속성의 도메인은 string으로 각각 40, 50자를 최대 길이로 두었다. 두 속성 모두 Publisher의 primary key이기 때문에 NULL값을 허용하지 않았다. * Manager\_name, Manager\_telephone\_ number : 두 속성의 도메인은 string으로 최대 길이를 20자로 두었으며 NULL값은 허용하지 않았다. * Manager\_E\_mail : string 도메인으로 최대 길이는 30자이며 NULL 값을 허용하지 않았다. |
|  | * Collection\_place, Book\_title : 두 속성의 도메인은 string으로 각각 50, 40자를 최대 길이로 두었다. NULL 값은 허용하지 않았다. * Borrow\_frequency : 대출빈도 즉, 횟수를 나타내는 속성은 0미만일 수 없기 때문에 Count라는 새로운 정수형 도메인을 생성하여 설정했다. Count는 최솟값이 0인 정수형 타입이다. NULL 값을 허용하지 않았다. |

1. Physical view
2. **시스템 요구사항 만족 관련 설명**
3. 직원 관리
4. Employee entity set에는 재귀 relationship이 있기 때문에 이를 통해 직원 리스트 및 개별 직원 정보를 열람할 수 있다.
5. 개별 직원 정보에 있는 모든 정보는 Employee entity attribute에 포함되어 있다.
6. 재고 관리
7. Employee entity set과 Publisher entity set은 one-to-many relationship으로 연결되어 있으며, 이를 통해 출판사 정보를 열람 및 수정할 수 있다.
8. 해당 출판사를 통해 구입한 도서 목록은 Employee와 Publisher 사이의 Relationship을 통해 Publisher와 BookList(구매 리스트) 사이의 Relationship으로 확인할 수 있다. 명세서에 나온 구매 리스트의 모든 정보는 BookList entity attribute에 포함되어 있다.
9. Employee entity set과 Book\_specific\_info entity set은 one-to-many relationship으로 연결되어 있으며, 이를 통해 도서에 대한 상세 정보를 열람 및 수정할 수 있다. Book\_specific\_info는 {Book(도서), SingleBook(개별서적), BorrowList(대출내역)}의 tenary relationship으로 도서와 개별 서적의 상세 정보 및 대출 내역까지도 조회할 수 있다.
10. 회원 관리
11. Employee entity set과 Member entity set은 one-to-many relationship으로 연결되어 있으며, 이를 통해 회원에 대한 상세 정보를 열람할 수 있다.
12. 회원 별 대출리스트는 Member entity set과 BorrowList(대출내역) entity set이 one-to-many relationship으로 연결되어 있으므로 이를 통해 확인할 수 있다.
13. 도서 검색
14. Member entity set과 Book\_search entity set은 one-to-many relationship으로 연결되어 있으며, 이를 통해 도서 검색 결과의 도서 상태를 열람할 수 있다. 이때 Book\_search enitiy에는 state 라는 속성이 있어 도서 상태를 확인할 수 있다.
15. 상태와 예약 현황을 제외한 도서 검색 결과는 Member와 Book\_search 사이의 Relationship을 통해 Book\_search와 Book\_specific\_info 사이의 Relationship으로 확인할 수 있다.
16. 도서 검색 결과의 예약 현황은 Member와 Reservation 사이의 Relationship을 통해 확인할 수 있다. 이때 Reservation에는 Reservation\_date(예약일) 속성이 포함되어 있다.
17. 대출 관리
18. Employee entity set과 Book\_specific\_info entity set은 one-to-many relationship으로 연결되어 있으며, Book\_specific\_info entity set은 다시 BorrowList, Book\_search와 relationship을 갖는다. 이를 통해 직원은 서적의 정보를 확인할 수 있고, 특히 Book\_search에 있는 state 속성을 이용해 도서 상태를 확인할 수 있다. 마지막으로 도서 별 예약 정보는 Book\_search와 Member 사이의 relationship을 통해 다시 Member와 Reservation 사이의 relationship을 이용함으로써 확인 가능하다.
19. **추가 설명**

|  |  |
| --- | --- |
|  | ERD 부분에서 이미 ISA에 대해 설명했지만, logical view에서도 다시 한번 언급하기 위한 것이다. Employee와 Member는 모두 사람이라는 공통적인 특성을 가지고 있고, 이름이라는 공통 속성을 갖는다. 따라서 이를 People Entity로 generalization하여 표현하였다. 그림을 보면 Employee, Member에는 따로 이름이라는 속성이 없다는 것을 확인할 수 있다. 또한 ERD에서 말했던 것처럼 이름 속성을 component attribute로 간주하여 First\_name, Middle\_name, Last\_name으로 나누었다. 이는 외국인 직원이 들어올 경우를 대비한 것이다. |