Databases Programming Project: **Final** **Report**

이름 (Name): 조현진

학과 (Department): 전기컴퓨터공학부

학번 (Student ID): 201824594

1. **프로젝트 개요 (Project topic)**

|  |
| --- |
| 사용한 언어: python,  사용한 라이브러리 :   * psycopg2 : PostgreSQL사용을 위한 라이브러리 * requests : API 요청을 위한 라이브러리   프로젝트 개요:  도서관 장서 관리 시스템 각 도서관의 장서정보와 대출정보를 조회할 수 있는 시스템이다. 이 시스템에서 각 도서관, 도서관이 보유한 장서에 대한 정보, 각 장서의 대출 정보가 포함되어 있다.  시스템의 사용자는 시스템 관리자, 도서관 관리자, 일반 이용자로 분류하여, 시스템 관리자는 사용자, 전체 도서관에 대한 관리, 전체 도서관에 포함된 장서에 대한 ISBN을 관리할 수 있다. 도서관 관리자는 담당하는 도서관의 정보와 도서관의 보유 장서를 관리하고, 일반 이용자의 장서 대출 요청을 처리한다. 일반 이용자는 장서에 대하여 대출, 반납을 신청할 수 있으며, 도서관 관리자의 승인 아래 장서를 대출, 반납할 수 있다. 또한 일반 사용자는 자신이 대출 중인 장서의 대출 기간을 연장할 수 있으며, 최대 연장 횟수가 지정되어 있다.  도서관 정보나루 API를 활용하여 최초 데이터를 입력받고, 이후 데이터 처리 기능을 구현하였다. |

1. **사용자 (역할) (Users / Roles)**

|  |
| --- |
| 1. 시스템 관리자  * 사용자 데이터 생성, 삭제, 변경할 수 있다. * 전체 도서관을 관리할 수 있으며, 신규 도서관의 개관 및 폐관에 대한 권한을 가진다 * ISBN 데이터를 관리할 수 있다  1. 도서관 관리자  * 각각의 도서관에 배정되어 담당 도서관의 데이터를 관리한다 * 담당 도서관의 장서 추가 및 삭제 등의 장서관련 데이터를 관리한다. * 장서 대출 및 반납을 처리한다. 장서 대출시 대출 기간, 대출 이용자를 갱신한다 * ISBN과 도서관이 보유한 장서의 데이터는 구분된다  1. 일반 이용자  * 각각의 도서관의 장서 및 대출 정보를 열람할 수 있다 * 장서에 대하여 대출을 예약, 반납을 요청할 수 있다. * 대출 중인 장서에 대하여 반납 기간을 연장할 수 있다 |

1. **기능 (Functions)**

|  |
| --- |
| 1. 시스템 관리자  * 시스템 내 등록된 데이터의 관리 * 신규 이용자의 데이터 관리 및 권한 부여 * 전체 도서관의 데이터 관리 * ISBN 데이터 관리  1. 도서관 관리자  * 담당 도서관에 한하여 데이터 관리 가능 * 자기 자신의 데이터 관리 가능 * 권한은 수정 불가능 * 신규 장서 추가 및 기존 장서 삭제 * 장서의 대출, 반납 처리  1. 일반 이용자  * 이용자 본인의 데이터 수정 * 각 도서관의 장서 조회 * 장서에 대하여, 대출, 반납 요청 * 대출한 장서의 대출 기간 연장 |

1. **데이터베이스 스키마 및 다이어그램 (Database schema / Schema diagram)**

|  |
| --- |
| Primary Key는 **볼드체**로 표기한다  Foreign Key는 밑줄로 표기한다   1. Users 테이블  * **UserCode**: SERIAL, PRIMARY KEY * ID: VARCHAR (20), NOT NULL * Password: VARCHAR (20), NOT NULL * AuthorCode: SMALLINT, NOT NULL, * (AuthorCode IN (0, 1, 2)) * LibCode: INT, NULLABLE * Name: VARCHAR (100), NOT NULL * Age: INT, NULLABLE * Gender: VARCHAR (1), NOT NULL * PhoneNumber: VARCHAR (15), NULLABLE * Email: VARCHAR (100), NULLABLE   Constraint:   * Gender IN ('M', 'F') * AuthorCode <> 1 OR LibCode IS NOT NULL  1. Books 테이블  * **ISBN**: CHAR (13) * BookName: VARCHAR (255), NOT NULL * Authors: VARCHAR (100), NOT NULL * Publisher: VARCHAR (100), NULLABLE * PublicationYear: INT * ClassNm: VARCHAR (20), NULLABLE * ClassNo: VARCHAR (255), NULLABLE   Constraint:   * CHECK (PublicationYear > 0)  1. Libraries 테이블  * **LibCode**: INT * LibName: VARCHAR(255), NOT NULL * Address: VARCHAR(255), NOT NULL * Tel: VARCHAR(30), NULLABLE * Fax: VARCHAR(30), NULLABLE * Latitude: DECIMAL(9, 6), NULLABLE * Longitude: DECIMAL(9, 6), NULLABLE * Homepage: VARCHAR(255), NULLABLE * Closed: BOOLEAN, NOT NULL DEFAULT FALSE * OperatingTime: VARCHAR(500), NULLABLE * BookCount: INT, DEFAULT 0  1. LibraryBooks 테이블  * **LibCode**: INT * REFERENCES Libraries(LibCode) * ON DELETE CASCADE * ON UPDATE CASCADE * **ISBN**: CHAR(13) * REFERENCES Books(ISBN) * ON DELETE CASCADE * ON UPDATE CASCADE * **Vol**: VARCHAR(5), NULLABLE * **CallNumber**: VARCHAR(100), NOT NULL * RegistrationDate: DATE, NOT NULL * DEFAULT CURRENT\_DATE * ISLoan: BOOLEAN, NOT NULL * DEFAULT FALSE   Constraint:   * PRIMARY KEY (LibCode, ISBN, Vol, CallNumber)  1. LoanProposals 테이블  * **ProposalID**: SERIAL * LibCode: INT, NOT NULL, * ISBN: CHAR(13), NOT NULL, * REFERENCES Books(ISBN) * ON DELETE CASCADE * ON UPDATE CASCADE * Vol: VARCHAR(10), NOT NULL * CallNumber: VARCHAR(100), NOT NULL * UserCode: INT, NOT NULL * LoanClassification: INT, NOT NULL * (0: 대출 신청, 1: 반납 신청)   Constraint:   * Foreign Key: LibraryBooks(LibCode, ISBN, Vol, CallNumber) * ON DELETE CASCADE * ON UPDATE CASCADE  1. Loans 테이블  * **LoanCode**: SERIAL * LibCode: INT, * REFERENCES Libraries(LibCode) * ON DELETE CASCADE * ON UPDATE CASCADE * ISBN: CHAR(13), NOT NULL * Vol: VARCHAR(10), NOT NULL * CallNumber: VARCHAR(100), NOT NULL * UserCode: INT * REFERENCES Users(UserCode) * ON DELETE CASCADE * ON UPDATE CASCADE * LoanDate: DATE, NOT NULL * DEFAULT CURRENT\_DATE * ReturnDate: DATE, NULLABLE * LoanExtensions: INT, * DEFAULT 0   Constraint:   * FOREIGN KEY: LibraryBooks(LibCode, ISBN, Vol, CallNumber) * ON DELETE CASCADE * ON UPDATE CASCADE |