2020-1학기 데이터베이스 시스템 01분반

대용량 관계 DB 기반 응용 개발 Project

중앙대학교 컴퓨터공학부 컴퓨터전공

20176758 박진영

-목차-

1. 개발환경 3

2. 응용 분야 4

1) 응용 분야 제목

2) 응용 분야 소개

3) 응용의 기능 리스트 및 설명

3. 관계 DB 스키마 및 설명 5

1) DB 스키마

2) 설명

4. 트랜잭션 설계 결과 및 지원 기능 7

1) 동적 SQL문 리스트

2) 트랜잭션 리스트 및 지원 기능 설명

5. 소스 코드 설명 12

1) Main.java

2) Transaction.java

3) 구현요건 충족에 관한 상세 설명

6. 테스트 레코드 생성 38

1) 랜덤 데이터 생성

2) 데이터 로드

3) 로드 결과

7. 응용 실행 데모 47

1) 도서 검색

2) 대출, 예약, 반납

3) 마이페이지

4) 관리

8. 소스 코드 실행 관련 필요사항 67

9. 사용자 매뉴얼 68

1. 개발 환경

- System OS: window 10
- DBMS 제품명 및 버전: Mysql community server 8.0.20
- 대화식 SQL 도구: MySQL 8.0 command line client
- Java version: 14
- Console application으로 개발

2. 응용 분야

1) 응용 분야 제목: 도서관

2) 응용 분야 소개: 대학 도서관의 도서 목록, 대출, 예약 기능 등을 제공한다.

3) 응용의 기능 리스트

① 도서 검색: 도서관에 소장 중인 도서 목록을 도서명, 출판사, 저자, 도서번호로 검색할 수 있다.

② 도서 대출: 학생은 도서 목록 중 대출 가능한 도서를 빌릴 수 있다.

③ 도서 반납: 대출한 도서를 반납한다.

④ 도서 예약: 대출을 원하는 도서를 타인이 빌리지 못하게 예약할 수 있다.

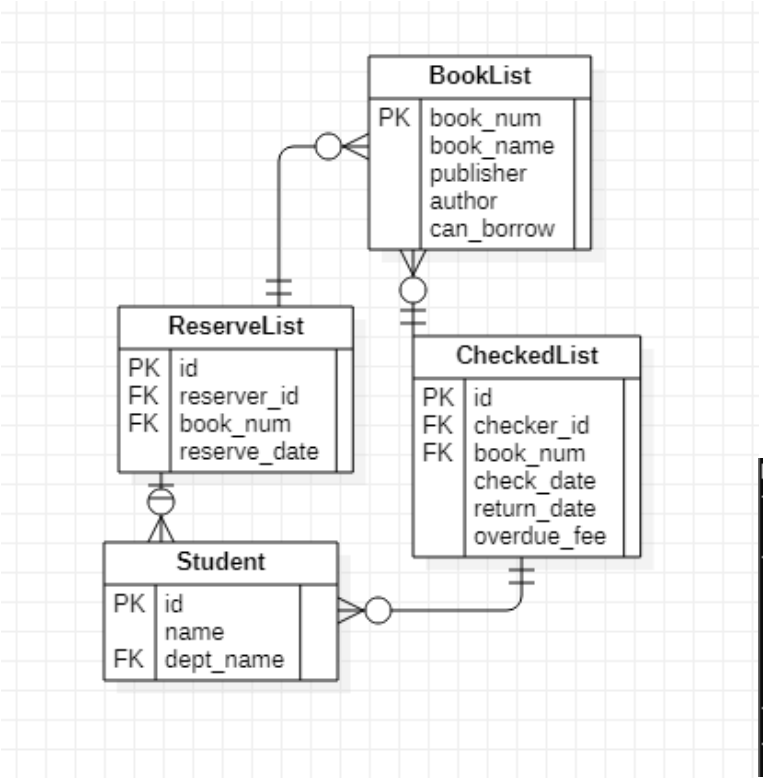
⑤ 내 정보: 내 대출, 예약, 연체료 정보를 확인할 수 있다.

⑥ 연체료 계산: 연체에 따라 발생한 요금을 계산한다. 해당 응용에선 반납 시에 결제하는 것으로 한정한다.

⑦ 도서 목록 수정: 존재하는 도서 목록에 대해 추가, 수정, 삭제를 할 수 있다.

3. 관계 DB 스키마 및 설명

1) DB 스키마 (왼: DB 스키마, 오: Table 생성 결과) : 테이블 생성 구문은 별도 첨부 (CreateTable.txt)




```
mysql> show tables;
+----------------------+
| Tables_in_projectdb |
+----------------------+
| booklist             |
| checkedlist          |
| reservelist          |
| student              |
+----------------------+
4 rows in set (0.00 sec)
```

2) 설명

테이블명	컬럼명	설명
Student	id (Int)	학생의 고유번호, Primary key
	name (Char)	학생의 이름
BookList	book_num (Int)	도서의 고유 번호, Primary key
	book_name (Char)	도서명
	publisher (Char)	출판사
	author (Char)	저자
	can_borrow (Boolean)	대출 가능한 지의 여부
CheckedList	id (Int)	대출 정보의 고유 번호, Primary key Auto Increasement
	checker_id (Int)	대출자 id Student Table의 id를 외래키로 사용
	book_num (Int)	도서 번호 BookList Table의 book_num을

		외래키로 사용
	check_date (Datetime)	대출 일자
	return_date (Datetime)	반납 일자
	overdue_fee (Int)	연체료
ReserveList	id (Int)	예약 정보의 고유 번호, Primary key Auto Increasement
	reserve_id (Int)	예약자 id Student Table의 id를 외래키로 사용
	book_num (Int)	도서 번호 BookList Table의 book_num을 외래키로 사용
	reserve_date (Datetime)	예약 일자

4. 트랜잭션 설계 결과 및 지원 기능

1) 동적 SQL 리스트

① S1

S1-1. 도서 목록에서 도서명으로 결과 검색

- "SELECT * FROM BookList WHERE book_name like ?"

S1-2. 도서 목록에서 출판사로 검색

- "SELECT * FROM BookList WHERE publisher like ?"

S1-3. 도서 목록에서 저자로 검색

- "SELECT * FROM BookList WHERE author like ?"

S1-4. 도서 목록에서 도서 고유 번호로 결과 검색

- "SELECT * FROM BookList WHERE book_num = ?"

S1-5. 도서 목록에서 도서명을 포함하는 개수 검색

- "SELECT count(*) FROM BookList WHERE book_name like ?"

S1-6. 도서 목록에서 출판사를 포함하는 개수 검색

- "SELECT count(*) FROM BookList WHERE publisher like ?"

S1-7. 도서 목록에서 저자를 포함하는 개수 검색

- "SELECT count(*) FROM BookList WHERE author like ?"

S1-8. 도서 목록에서 도서 고유 번호 개수 검색. 고유 번호가 리스트에 존재하는지 여부를 판단하기 위해 필요.

- "SELECT count(*) FROM BookList WHERE book_num = ?"

② S2

S2-1. 도서 목록에 도서 정보 추가

- "INSERT INTO BookList VALUES (?,?,?,?,?)"

S2-2. 도서 정보 수정

- "UPDATE BookList SET book_name=?, publisher=?, author=? WHERE book_num=?"

S2-3. 도서 목록에서 도서 고유 번호에 해당하는 row 삭제

- "DELETE FROM BookList WHERE book_num=?"

S2-4. 도서 고유 번호에 해당하는 row의 can_borrow 컬럼값 수정.

- "UPDATE BookList SET can_borrow=? WHERE book_num=?"

③ S3

S3-1. 대출 리스트에 대출 정보 추가

- "INSERT INTO CheckedList (checker_id,book_num) VALUES (?,?)"

S3-2. 대출 리스트에서 return_date 수정. 반납 기능에 필요.

- "UPDATE CheckedList SET return_date=? WHERE checker_id=? and book_num=? and return_date is NULL"

④ S4

S4-1. 예약 리스트에 예약 정보 추가

- "INSERT INTO ReserveList (reserver_id, book_num) VALUES (?,?)"

S4-2. 예약자 번호와 도서 번호로 예약 정보 조회

- "SELECT * FROM ReserveList WHERE reserver_id = ? and book_num=?"

S4-3. 예약 정보 삭제

- "DELETE FROM ReserveList WHERE id = ?"

⑤ S5

S5-1. 대출자 번호로 대출 정보 조회

- "SELECT * FROM CheckedList WHERE checker_id = ?"

S5-2. 예약자 정보로 예약 정보 조회

- "SELECT * FROM ReserveList WHERE reserver_id = ?"

S5-3. 대출자 번호로 대출 정보 개수 검색

- "SELECT count(*) FROM CheckedList WHERE checker_id = ?"

S5-4. 예약자 번호로 예약 정보 개수 검색

- "SELECT count(*) FROM ReserveList WHERE reserver_id = ?"

⑥ S6

S6-1. 대출자 번호로 반납 일자가 지난 대출 정보 개수 검색

- SELECT count(*) FROM CheckedList WHERE checker_id = ? and return_date is NULL and

date_format(check_date, '%Y-%m-%d')<?"

S6-2. 대출자 번호로 반납 일자가 지난 대출 정보 조회

- "SELECT * FROM CheckedList WHERE checker_id = ? and return_date is NULL and date_format(check_date, '%Y-%m-%d')<?"

S6-3. 대출 정보의 연체료 정보 수정

- "UPDATE CheckedList SET overdue_fee=? WHERE id=?"

2) 트랜잭션 리스트 및 지원 기능 설명

① T1 (검색 only)

(1) T1-1: 도서 목록 검색

- S1-1 ~ S1-8의 동적 SQL 질의문을 사용해 도서명, 출판사, 저자, 도서번호로 원하는 목록을 조회하는 것과 Pagination 기능을 구현한다.

(2) T1-2: 대출 정보 검색

- S5-1, S5-3의 동적 SQL 질의문을 사용해 대출자 번호에 따른 대출 정보 조회와 Pagination 기능을 구현한다.

(3) T1-3: 예약 정보 검색

- S5-2, S5-4의 동적 SQL 질의문을 사용해 예약자 번호에 따른 예약 정보 조회와 Pagination 기능을 구현한다.

② T2 (변경)

(1) T2-1: 도서 정보 추가

- 도서번호, 도서명, 출판사, 저자를 입력 받아 S2-1의 동적 SQL 변경문을 사용해 도서 정보를 추가한다.

(2) T2-2: 도서 정보 삭제

- 도서번호를 입력 받아 S2-3의 변경문을 사용해 도서 정보를 삭제한다.

(3) T2-3: 반납

- 도서번호와 대출자 ID를 입력 받아 S2-4 변경문을 사용해 도서 정보의 can_borrow 정보를 수정한다.

- S3-2 변경문으로 대출 정보에 반납 일자를 추가한다.

③ T3 (검색, 변경 혼합)

(1) T3-1: 도서 정보 수정

- 도서번호를 입력 받아 S1-4 질의문을 사용해 해당되는 도서 정보를 보여준다.
- 수정하기 원하는 컬럼을 선택한 후 새로운 값을 입력 받는다.
- 새로 입력 받은 값 이외의 정보는 질의문을 사용했을 때 얻은 정보를 사용한다.
- S2-2 변경문을 사용해 도서 정보를 수정한다.

(2) T3-2: 도서 대출

- 도서명을 입력 받고 도서명을 가진 정보가 1개 이상이라면 S1-1, S1-5 질의문을 사용해 도서 목록을 Pagination하여 보여준다.
- 빌리기 원하는 도서의 번호를 입력 받고 S1-4, S1-8 질의문을 사용해 해당 도서 정보를 가져온다.
- 도서명을 가진 정보가 1개라면 S1-1를 사용해 해당 도서 정보를 가져온다.
- 학생 ID를 입력 받고 S4-2 질의문을 사용해 같은 Id, 도서번호를 가진 예약 정보가 있는 지 조회한다.
- 예약 정보가 있다면 결과를 보여준 후 S4-3 변경문을 사용해 예약 정보를 삭제한다.
- S3-1 변경문으로 대출 정보를 추가한다.
- S2-4 변경문으로 도서의 대출 가능 값을 False로 수정한다.

(3) T3-3: 도서 예약

- 도서명을 입력 받고 도서명을 가진 정보가 1개 이상이라면 S1-1, S1-5 질의문을 사용해 도서 목록을 Pagination하여 보여준다.
- 예약하길 원하는 도서의 번호를 입력 받고 S1-4, S1-8 질의문을 사용해 해당 도서 정보를 가져온다.
- 도서명을 가진 정보가 1개라면 S1-1를 사용해 해당 도서 정보를 가져온다.
- 학생 ID를 입력 받고 S4-2 질의문을 사용해 예약 정보를 조회한다.

- 예약 정보가 없다면 S4-1 변경문을 사용해 예약 정보를 추가한다.

(4) T3-4: 연체료 업데이트 및 확인

- 학생 ID를 입력 받고 S6-1, S6-2 질의문을 사용해 반납 일자가 지난 대출 정보를 가져온다.
- 현재 날짜를 기준으로 연체료를 계산해 S6-3 변경문으로 각 대출 정보의 연체료 값을 수정한다.

(5) T3-5: 연체료 정산:

- 학생 ID를 입력 받고 S6-1, S6-2 질의문을 사용해 반납 일자가 지난 대출 정보를 가져온 후 Pagination하여 연체료와 함께 보여준다.
- 계산하기 원하는 대출 ID를 입력 받고 S6-3 변경문을 사용해 연체료 정보를 0으로 수정한다.

5. 소스 코드 설명

1) Main.java

코드	설명

`public class Main {`
`public static void main(String[] args) {`
`Transaction trans = new Transaction();`
`while(true) {`
`System.out.println("Welcome to libary!");`
`System.out.println("\n--------Select the option number(<0> to exit)---------");`
`System.out.println("  1. Search for books\r\n" +`
`"  2. Borrow a book\r\n" +`
`"  3. Reserve a book\r\n" +`
`"  4. Return a book\r\n" +`
`"  5. My page\r\n" +`
`"  6. Management");`
`System.out.print("\n >> ");`
`Scanner sc = new Scanner(System.in);`
`int func = sc.nextInt();`
`if (func==0) {`
`break;`
`}`
	Main() 시작
	각 옵션 번호를 입력하면 switch 문으로 해당되는 트랜잭션을 실행한다.

`switch(func) {`
`case 1: //search for books`
`while(true) {`
`clearScreen();`
`System.out.println("\n--------Search with..---------");`
`System.out.println("  1. Book name\r\n" +`
`"  2. Publisher\r\n" +`
`"  3. Author\r\n" +`
`"  4. Book number");`
`System.out.println("\nPlease Enter <0> to go main screen");`
`System.out.print(" >> ");`
`func = sc.nextInt();`
`if(func == 0 ) break;`
`trans.searchBookList(func);`
`}`
`break;`
 	
`case 2: //Borrow book`
`clearScreen();`
`trans.checkBook();`
`break;`
 	
`case 3: //Return book`
`clearScreen();`
`trans.reserveBook();`
`break;`
 	
`case 4: //Return book`
`clearScreen();`
`System.out.println("\nEnter the information to return the book (<0> to return main screen)");`
`System.out.print("  > book number: ");`
`int book_num = sc.nextInt();`
`if(book_num == 0) break;`
`System.out.print("  > student id: ");`
`int id = sc.nextInt();`
`if(id == 0) break;`
`trans.returnBook(book_num, id);`
`break;`
	Case1: 도서 검색
	Case2: 대출
	Case3: 예약
	Case4: 반납

```java
case 5: //Mypage
    clearScreen();
    System.out.println("------------------Mypage!-----------------");
    System.out.print(" > Please Enter your student ID: ");
    int student_id = sc.nextInt();
    while(true) {
        clearScreen();
        System.out.println("\n--------Select the option number---------");
        System.out.println("  1. Show check list\r\n" +
                "  2. Show reserve list\r\n" +
                "  3. Check overdue fee\r\n" +
                "  4. Pay overdue fee");
        System.out.println("\nPlease Enter <0> to go main screen");
        System.out.print(" >> ");
        int func_1 = sc.nextInt();
        if(func_1==0) break;
        switch(func_1) {
        case 1:
            trans.checklist(student_id);
            break;
        case 2:
            trans.reservelist(student_id);
            break;
        case 3:
            trans.overduelist(student_id);
            break;
        case 4:
            trans.payoverduefee(student_id);
            break;
        default:
            System.out.println("Wrong option number!");
            break;
        }
    }
    break;
```

Case5: Mypage

- ➔ 0을 입력 받기 전까지 실행
- ➔ Option을 입력 받아 switch문에 사용한다.
- ➔ Case1: 대출 리스트
- ➔ Case2: 예약 리스트
- ➔ Case3: 연체료 조회
- ➔ Case4: 연체료 계산

```java
case 6: //Management
    while(true) {
        clearScreen();
        System.out.println("\n--------Select the function number---------");
        System.out.println("  1. Add a book to List\r\n" +
                "  2. Modify a book's infomation\r\n" +
                "  3. Delete a book from List");
        System.out.println("\nPlease Enter <0> to go main screen");
        System.out.print(" >> ");
        func = sc.nextInt();
        if(func == 0 ) break;
        switch(func) {
        case 1:
            trans.addbook();
            break;
        case 2:
            trans.modifybook();
            break;
        case 3:
            trans.deletebook();
            break;
        default:
            System.out.println("Wrong option number!");
            break;
        }
    }
```

Case 6: Management

- ➔ 0을 입력 받기 전까지 실행한다..
- ➔ 도서목록을 관리할 수 있다.
- ➔ Option을 입력 받아 switch문에 사용한다.
- ➔ Case1: 도서 추가
- ➔ Case2: 도서 정보 수정
- ➔ Case3: 도서 삭제

2) Transaction.java

- 모든 트랜잭션 함수의 시작 부분에 `con.setAutoCommit(false);`을 실행하여 트랜잭션의 시작임을 알린다. 또한, `pstmt = con.prepareStatement(s2_2);`, `pstmt.setInt(4, book_num);`, `pstmt.executeQuery();`, `pstmt.executeUpdate();`를 사용해 동적 SQL문을 사용한다. 트랜잭션 마지막에 `con.commit();`을 실행해 변경 사항을 데이터베이스에 반영한다. 트랜잭션이 중단되어야 할 때 예외를 발생시켜 `con.rollback();`을 실행한다.

코드	설명

`public class Transaction {`
`static String url = "jdbc:mysql://localhost:3310/projectDB?characterEncoding=UTF-8&serverTimezone=UTC";`
`static String id = "root";`
`static String pw = "jinyoung22!m";`
`static String driver = "com.mysql.cj.jdbc.Driver";`
 	
`private Connection con;`
`private PreparedStatement pstmt;`
`private PreparedStatement pstmt2;`
`private ResultSet rset;`
`private ResultSet rset2;`
 	
`String s1_1 = "SELECT * FROM BookList WHERE book_name like ?";`
`String s1_2 = "SELECT * FROM BookList WHERE publisher like ?";`
`String s1_3 = "SELECT * FROM BookList WHERE author like ?";`
`String s1_4 = "SELECT * FROM BookList WHERE book_num = ?";`
`String s1_5 = "SELECT count(*) FROM BookList WHERE book_name like ?";`
`String s1_6 = "SELECT count(*) FROM BookList WHERE publisher like ?";`
`String s1_7 = "SELECT count(*) FROM BookList WHERE author like ?";`
`String s1_8 = "SELECT count(*) FROM BookList WHERE book_num = ?";`
 	
`String s2_1 = "INSERT INTO BookList VALUES (?,?,?,?,?)";`
`String s2_2 = "UPDATE BookList SET book_name=?, publisher=?, author=? WHERE book_num=?";`
`String s2_3 = "DELETE FROM BookList WHERE book_num=?";`
`String s2_4 = "UPDATE BookList SET can_borrow=? WHERE book_num=?";`
 	
`String s3_1 = "INSERT INTO CheckedList (checker_id,book_num) VALUES (?,?)";`
`String s3_2 = "UPDATE CheckedList SET return_date=? WHERE checker_id=? and book_num=? and return_date is NULL";`
 	
`String s4_1 = "INSERT INTO ReserveList (reserver_id, book_num) VALUES (?,?)";`
`String s4_2 = "SELECT * FROM ReserveList WHERE reserver_id = ? and book_num=?";`
`String s4_3 = "DELETE FROM ReserveList WHERE id = ?";`
 	
`String s5_1 = "SELECT * FROM CheckedList WHERE checker_id = ?";`
`String s5_2 = "SELECT * FROM ReserveList WHERE reserver_id = ?";`
`String s5_3 = "SELECT count(*) FROM CheckedList WHERE checker_id = ?";`
`String s5_4 = "SELECT count(*) FROM ReserveList WHERE reserver_id = ?";`
 	
`String s6_1 = "SELECT count(*) FROM CheckedList WHERE checker_id = ? and return_date is NULL and date_format(check_date, '%Y-%m-%d')<?";`
`String s6_2 = "SELECT * FROM CheckedList WHERE checker_id = ? and return_date is NULL and date_format(check_date, '%Y-%m-%d')<?";`
`String s6_3 = "UPDATE CheckedList SET overdue_fee=? WHERE id=?";`
 	
`Scanner scn = new Scanner(System.in);`
 	
`public static void clearScreen() {`
`for (int i = 0; i < 80; i++)`
`System.out.println("");`
`}`
	Connection, PreparedStatement, ResultSet, 동적SQL문을 미리 생성해두었다.

`public static void printBookList(ResultSet rs, int page, int full) throws SQLException {`
`// Ensure we start with first row`
`clearScreen();`
`System.out.printf("Current Page: %d/%d\n\n", page, full);`
`rs.absolute((page - 1) * 10 );`
 	
`System.out.print("\tBook num\t\t Book name\t\t\t Publisher\t\t\t Author\t\t\t  etc.\n");`
`System.out.print("----------\n");`
`while (rs.next()) {`
`// Retrieve by column name`
`int book_num = rs.getInt("book_num");`
`String book_name = rs.getString("book_name");`
`String publisher = rs.getString("publisher");`
`String author = rs.getString("author");`
`boolean can_borrow = rs.getBoolean("can_borrow");`
 	
`// Display values`
`System.out.print("\t"+book_num+"\t");`
`System.out.printf("%-25s\t",book_name);`
`System.out.printf("\t%-20s\t",publisher);`
`System.out.printf("\t%-20s\t",author);`
`if (can_borrow) {`
`System.out.println("  Available");`
`} else {`
`System.out.println(" Not Available");`
`}`
 	
`int rowNum = rs.getRow();`
`if (rowNum == (page) * 10 || rs.isLast()) {`
`break;`
`}`
`}`
`System.out.println();`
`}`
	도서 목록 pagination

```java
public static void printCheckedList(ResultSet rs, int page, int full) throws SQLException {
    // Ensure we start with first row
    clearScreen();
    System.out.printf("Current Page: %d/%d\n\n", page, full);
    rs.absolute((page - 1) * 10 );

    while (rs.next()) {
        // Retrieve by column name
        int id = rs.getInt("id");
        String book_num = rs.getString("book_num");
        String check_date = rs.getString("check_date");
        String return_date = rs.getString("return_date");

        // Display values
        System.out.print("Id: " + id);
        System.out.print(", Book num: " + book_num);
        System.out.print(", Checked date: " + check_date);
        if (return_date != null) {
            System.out.println(",  Return date: " + return_date);
        } else {
            System.out.println(",  Before return");
        }

        int rowNum = rs.getRow();
        if (rowNum == (page) * 10 || rs.isLast()) {
            break;
        }
    }
    System.out.println();
}
```

대출 목록

Pagination

```java
public static void printReserveList(ResultSet rs, int page, int full) throws SQLException {
    // Ensure we start with first row
    clearScreen();
    System.out.printf("Current Page: %d/%d\n\n", page, full);
    rs.absolute((page - 1) * 10 );

    while (rs.next()) {
        // Retrieve by column name
        int id = rs.getInt("id");
        String book_num = rs.getString("book_num");
        String reserve_date = rs.getString("reserve_date");

        // Display values
        System.out.printf("ID: %06d",id);
        System.out.print(", Book num: " + book_num);
        System.out.println(", Reserved date: " + reserve_date);

        int rowNum = rs.getRow();
        if (rowNum == (page) * 10 || rs.isLast()) {
            break;
        }
    }
    System.out.println();
}
```

예약 목록

Pagination

```java
public static void printOverdueList(ResultSet rs, int page, int full) throws SQLException {
    // Ensure we start with first row
    clearScreen();
    System.out.printf("Current Page: %d/%d\n\n", page, full);
    rs.absolute((page - 1) * 10 );

    while (rs.next()) {
        // Retrieve by column name
        int id = rs.getInt("id");
        String book_num = rs.getString("book_num");
        String check_date = rs.getString("check_date");
        String overdue_fee = rs.getString("overdue_fee");

        // Display values
        System.out.printf("ID: %06d", id);
        System.out.print(", Book num: " + book_num);
        System.out.print(", Checked date: " + check_date);
        System.out.println(", Overdue fee: " + overdue_fee);

        int rowNum = rs.getRow();
        if (rowNum == (page) * 10 || rs.isLast()) {
            break;
        }
    }
    System.out.println();
}
```

연체정보
Pagination

```java
// T1-1 검색only
public void searchBookList(int func)   {

    try {
        con.setAutoCommit(false);

        int rowCount = 0;

        if (func == 1) {

            System.out.print("Input book name: ");
            String book_name = scn.nextLine();
            book_name = "%"+book_name+"%";
            pstmt2 = con.prepareStatement(s1_5);
            pstmt2.setString(1, book_name);
            rset2 = pstmt2.executeQuery();
            rset2.next();
            rowCount = rset2.getInt(1);
            if (rowCount > 0) {
                pstmt = con.prepareStatement(s1_1, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
                pstmt.setString(1, book_name);
                rset = pstmt.executeQuery();
            } else {
                throw new Exception("No matched information!");
            }

        } else if (func == 2) {

            System.out.print("Input publisher: ");
            String publisher = scn.nextLine();

            publisher = "%"+publisher+"%";
            pstmt2 = con.prepareStatement(s1_6);
            pstmt2.setString(1, publisher);
            rset2 = pstmt2.executeQuery();
            rset2.next();
            rowCount = rset2.getInt(1);
            if (rowCount > 0) {
                pstmt = con.prepareStatement(s1_2, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
                pstmt.setString(1, publisher);
                rset = pstmt.executeQuery();
            } else {
                throw new Exception("No matched information!");
            }
```

T1-1. 도서검색

main에서 입력 받은 option

option=1:
도서명 검색

option=2:
출판사 검색

option=3
저자 검색

option=4
도서 번호 검색

검색 정보 크기가
0이면 예외 발생 시킴

검색 후

```java
        } else if (func == 3) {

            System.out.print("Input author: ");
            String author = scn.nextLine();

            author = "%"+author+"%";
            pstmt2 = con.prepareStatement(s1_7);
            pstmt2.setString(1, author);
            rset2 = pstmt2.executeQuery();
            rset2.next();
            rowCount = rset2.getInt(1);
            if (rowCount > 0) {
                pstmt = con.prepareStatement(s1_3, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
                pstmt.setString(1, author);
                rset = pstmt.executeQuery();
            } else {
                throw new Exception("No matched information!");
            }
        } else if (func == 4) {

            System.out.print("Input book number: ");
            int book_num = scn.nextInt();

            pstmt2 = con.prepareStatement(s1_8);
            pstmt2.setInt(1, book_num);
            rset2 = pstmt2.executeQuery();
            rset2.next();
            rowCount = rset2.getInt(1);
            if (rowCount > 0) {
                pstmt = con.prepareStatement(s1_4, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
                pstmt.setInt(1, book_num);
                rset = pstmt.executeQuery();
            } else {
                throw new Exception("No matched information!");
            }
        } else {
            throw new Exception("Wrong function number!");
        }

        int total_page = (int) Math.ceil((double)rowCount / 10.0);;
        printBookList(rset, 1, total_page);
        while (true) {
            System.out.print("\n\nEnter page number to move(<0> to exit) > ");
            int page = scn.nextInt();
            scn.nextLine();
            if (page == 0) {
                break;
            }
            else if(page>total_page) {
                System.out.println("\nExceed the total page number!");
                continue;
            }
            printBookList(rset, page, total_page);
        }
          con.commit();

      } catch (Throwable e) {
          System.out.println(e.getMessage());
          try {
              Thread.sleep(1000);
          } catch (InterruptedException e1) {
          }
          if (con != null) {
              try {
                  con.rollback();
              } catch (SQLException ex) {
              }
          }
      }
  }
```

Result Set을 PrintBookList() 사용해 입력받은 페이지에 따라 보여준다.

```java
// T1-2 검색only
public void checklist(int id)   {
    try {
        con.setAutoCommit(false);

        // s5_3 = "SELECT count(*) FROM CheckedList WHERE checker_id = ?";
        pstmt2 = con.prepareStatement(s5_3);
        pstmt2.setInt(1, id);
        rset2 = pstmt2.executeQuery();
        rset2.next();
        int rowCount = rset2.getInt(1);
        if (rowCount > 0) {
            // s5_1 = "SELECT * FROM CheckedList WHERE checker_id = ?"
            pstmt = con.prepareStatement(s5_1, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
            pstmt.setInt(1, id);
            rset = pstmt.executeQuery();
        } else {
            throw new Exception("No matching information!");
        }

        int full_page = (int) Math.ceil((double)rowCount / 10.0);
        if(full_page == 0) full_page = 1;
        printCheckedList(rset, 1, full_page);
        while (true) {
            System.out.print("\n\nEnter page number to move(<0> to exit) > ");
            int page = scn.nextInt();
            if (page == 0) {
                break;
            } else if (page > full_page) {
                System.out.println("\nExceed the total page number!");
                continue;
            }
            printCheckedList(rset, page, full_page);
        }

        con.commit();

    } catch (Throwable e) {
        System.out.println(e.getMessage());
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e1) {
        }
        if (con != null) {
            try {
                con.rollback();
            } catch (SQLException ex) {
            }
        }
    }
}
```

T1-2.
대출 목록 조회

입력받은 학생 ID를 가지는 대출 정보를 검색

개수 0이면 예외 발생 시킴

ResultSet을 PrintCheckedList()를 사용해 입력받은 Page에 따라 보여준다.

```java
// T1-3 검색only
public void reservelist(int id)   {
    try {
        con.setAutoCommit(false);

        // s5_4 = "SELECT count(*) FROM ReserveList WHERE reserver_id = ?";
        pstmt2 = con.prepareStatement(s5_4);
        pstmt2.setInt(1, id);
        rset2 = pstmt2.executeQuery();
        rset2.next();
        int rowCount = rset2.getInt(1);
        if (rowCount > 0) {
            // s5_2 = "SELECT * FROM ReserveList WHERE reserver_id = ?";
            pstmt = con.prepareStatement(s5_2, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
            pstmt.setInt(1, id);
            rset = pstmt.executeQuery();
        } else {
            throw new Exception("No matching information!");
        }

        int total_page = (int) Math.ceil((double)rowCount / 10.0);
        printReserveList(rset, 1, total_page);
        while (true) {
            System.out.print("\n\nEnter page number to move(<0> to exit) > ");
            int page = scn.nextInt();
            if (page == 0) {
                break;
            } else if (page > total_page) {
                System.out.println("\nExceed the total page number!");
                continue;
            }
            printReserveList(rset, page, total_page);
        }

        con.commit();

    } catch (Throwable e) {
        System.out.println(e.getMessage());
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e1) {
        }
        if (con != null) {
            try {
                con.rollback();
            } catch (SQLException ex) {
            }
        }
    }
}
```

T1-3.
예약 목록 조회

입력 받은 학생 ID를 가지는 예약 정보 조회.

개수 0이면 예외 발생시킴

ResultSet을 PrintReserveList()를 사용해 원하는 Page에 따라 보여준다.

```java
// T2-1 변경
public void addbook()   {
    try {
        con.setAutoCommit(false);

        pstmt = con.prepareStatement(s2_1);

        System.out.print("Input book number: ");
        int book_num = scn.nextInt();
        scn.nextLine();
        System.out.print("Input book name: ");
        String book_name = scn.nextLine();
        System.out.print("Input publisher: ");
        String publisher = scn.nextLine();
        System.out.print("Input author: ");
        String author = scn.nextLine();

        pstmt.setInt(1, book_num);
        pstmt.setString(2, book_name);
        pstmt.setString(3, publisher);
        pstmt.setString(4, author);
        pstmt.setInt(5, 1);

        pstmt.executeUpdate();

        con.commit();

        System.out.println("Success to add!");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e1) {
        }

    } catch (Throwable e) {
        System.out.println(e.getMessage());
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e1) {
        }
        if (con != null) {
            try {
                con.rollback();
            } catch (SQLException ex) {
            }
        }
    }
}
```

T2-1.
도서 추가

도서번호,
도서명,
출판사,
저자를
입력 받은 후

업데이트 실행

```java
// T2-2 변경
public void deletebook()   {
    try {
        con.setAutoCommit(false);

        pstmt = con.prepareStatement(s2_3);

        System.out.print("Input book number to delete: ");
        int book_num = scn.nextInt();

        pstmt.setInt(1, book_num);

        int s = pstmt.executeUpdate();

        if (s==0) {
            throw new Exception("No row to delete!");
        }

        con.commit();

        System.out.println("Success to delete!");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e1) {
        }

    } catch (Throwable e) {
        System.out.println(e.getMessage());
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e1) {
        }
        if (con != null) {
            try {
                con.rollback();
            } catch (SQLException ex) {
            }
        }
    }
}
```

T2-2
도서 삭제

도서 번호를 입력받아 해당되는 정보 삭제 실행.

영향 받은 row 가 없다면 예외 발생 시킴

```java
// T2-3 변경
public void returnBook(int book_num, int id)   {
    try {
        // s2_4 = "UPDATE BookList can_borrow=? WHERE book_num=?";
        con.setAutoCommit(false);

        pstmt = con.prepareStatement(s2_4);
        pstmt.setInt(1, 1);
        pstmt.setInt(2, book_num);
        int s = pstmt.executeUpdate();
        if(s==0) {
            throw new Exception("No rows matching the book number!"
        }

        // s3_2 = "UPDATE CheckedList SET return_date=? WHERE check
        // book_num=? and return_date is NULL";
        pstmt2 = con.prepareStatement(s3_2);
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd
        String format_time = format.format(System.currentTimeMillis
        pstmt2.setString(1, format_time);
        pstmt2.setInt(2, id);
        pstmt2.setInt(3, book_num);
        s = pstmt2.executeUpdate();
        if(s==0) {
            throw new Exception("No checked information!");
        }

        con.commit();

        System.out.println("Success to return book");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e1) {
        }

    } catch (Throwable e) {
        System.out.println(e.getMessage());
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e1) {
        }
        if (con != null) {
            try {
                con.rollback();
            } catch (SQLException ex) {
            }
        }
    }
}
```

T2-3.
도서 반납

도서번호, 학생 ID를 입력 받고

도서정보의 대출가능 여부 수정

대출정보의 반납일자 수정

각 수정에 대해 영향 받은 row가 없다면 예외발생

```java
// T3-1 혼합, b,e
public void modifybook()   {
    try {
        con.setAutoCommit(false);

        pstmt = con.prepareStatement(s1_4);

        System.out.print("Input book number to modify: ");
        int book_num = scn.nextInt();

        clearScreen();

        pstmt.setInt(1, book_num);

        rset = pstmt.executeQuery();

        String book_name = new String();
        String publisher = new String();
        String author = new String();
        boolean can_borrow;

        while (rset.next()) {
            // Retrieve by column name
            book_name = rset.getString("book_name");
            publisher = rset.getString("publisher");
            author = rset.getString("author");
            can_borrow = rset.getBoolean("can_borrow");

            System.out.print("\nBook name: " + book_name);
            System.out.print(", Publisher: " + publisher);
            System.out.print(", Author: " + author);
            if (can_borrow) {
                System.out.println(",  Available");
            } else {
                System.out.println(",  Not Available");
            }
        }

        System.out.println("\n--------Select number to modify---------");
        System.out.println("  1. Book name\r\n" + "  2. Publisher\r\n" + "  3. Author");
        System.out.println("\nPlease Enter <0> to go main screen");
        System.out.print("\n >> ");
        int func = scn.nextInt();
        scn.nextLine();
        if (func == 0)
            throw new Exception("Stop to modify the book information!");
```

T3-1.
도서정보 수정

도서 번호를 입력 받아 도서 정보 검색 후 출력

수정하기 원하는 컬럼을 선택

새로운 값을 입력 받고 업데이트 수행.

```java
        switch (func) {
        case 1:
            System.out.print("New book name: ");
            book_name = scn.nextLine();
            break;
        case 2:
            System.out.print("New publisher: ");
            publisher = scn.nextLine();
            break;
        case 3:
            System.out.print("New author: ");
            author = scn.nextLine();
            break;
        }

        pstmt = con.prepareStatement(s2_2);
        pstmt.setInt(4, book_num);
        pstmt.setString(1, book_name);
        pstmt.setString(2, publisher);
        pstmt.setString(3, author);

        pstmt.executeUpdate();

        con.commit();

        System.out.println("Success to modify!");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e1) {
        }

    } catch (Throwable e) {
        System.out.println(e.getMessage());
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e1) {
        }
        if (con != null) {
            try {
                con.rollback();
            } catch (SQLException ex) {
            }
        }
    }
}
```

```java
// T3-2 혼합, a~e
public void checkBook()   {
    try {
        con.setAutoCommit(false);

        System.out.print("Input book name to borrow: ");
        String book_name = scn.nextLine();
        book_name = "%"+book_name+"%";
        // SELECT count(*) FROM BookList WHERE book_name = ?
        pstmt = con.prepareStatement(s1_5);
        pstmt.setString(1, book_name);
        rset = pstmt.executeQuery();
        rset.next();
        int rowCount = rset.getInt(1);;
        if (rowCount>0) {}
        else
            throw new Exception("No information matching the book name!");

        if (rowCount > 1) {
            // SELECT * FROM BookList WHERE book_name = ?
            pstmt = con.prepareStatement(s1_1, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
            pstmt.setString(1, book_name);
            rset = pstmt.executeQuery();

            int total_page = (int) Math.ceil((double)rowCount / 10.0);;
            printBookList(rset, 1, total_page);
            while (true) {
                System.out.println("\nEnter <0> to select book number");
                System.out.print("Enter page number to move > ");
                int page = scn.nextInt();
                if (page == 0) {
                    break;
                }
                else if(page>total_page) {
                    System.out.println("\nExceed the total page number!");
                    continue;
                }
                printBookList(rset, page, total_page);
            }

            System.out.print("Input book number to borrow: ");
            int book_num = scn.nextInt();

            //s1_8 = "SELECT count(*) FROM BookList WHERE book_num = ?";
            pstmt = con.prepareStatement(s1_8);
            pstmt.setInt(1, book_num);
            rset = pstmt.executeQuery();
            rset.next();
            int book_i = rset.getInt(1);
            if(book_i!=1) {
                throw new Exception("No book matching!");
            }
```

T2-2. 대출

도서명 입력 받음

정보 개수가 0이면 예외 발생

정보가 1개보다 많다면 PrintBookList()를 사용해 원하는 페이지의 정보를 보여주고, 원하는 도서번호를 입력 받아 ResultSet 검색.

1개라면 도서명으로 ResultSet 검색

학생 ID를 입력 받고

앞의 ResultSet에서 얻은 도서 번호로 예약정보 검색

예약 정보가 있다면 삭제

```java
        // SELECT * FROM BookList WHERE book_num = ?
        pstmt = con.prepareStatement(s1_4);
        pstmt.setInt(1, book_num);
        rset = pstmt.executeQuery();
    } else {
        //s1_5 = "SELECT count(*) FROM BookList WHERE book_name = ?";
        pstmt = con.prepareStatement(s1_5);
        pstmt.setString(1, book_name);
        rset = pstmt.executeQuery();
        rset.next();
        int book_i = rset.getInt(1);
        if(book_i!=1) {
            throw new Exception("No book matching!");
        }

        // SELECT * FROM BookList WHERE book_name like ?
        pstmt = con.prepareStatement(s1_1);
        pstmt.setString(1, book_name);
        rset = pstmt.executeQuery();
    }

    int book_num = 0;
    String publisher = new String();
    String author = new String();
    String book_name_2 = new String();
    boolean can_borrow;

    while (rset.next()) {
        book_num = rset.getInt("book_num");
        book_name_2 = rset.getString("book_name");
        publisher = rset.getString("publisher");
        author = rset.getString("author");
        can_borrow = rset.getBoolean("can_borrow");
        if (!can_borrow) {
            throw new Exception("Can't borrow!");
        }
    }

    System.out.print("Input your student ID: ");
    int id = scn.nextInt();
```

대출 정보 업데이트

도서 정보의 can_borrow 값 False로 업데이트

```java
    // SELECT * FROM ReserveList WHERE reserver_id = ? and book_num=?
    pstmt2 = con.prepareStatement(s4_2);
    pstmt2.setInt(1, id);
    pstmt2.setInt(2, book_num);
    rset2 = pstmt2.executeQuery();
    while (rset2.next()) {
        // Retrieve by column name
        System.out.println("You have reservation information!");
        int reserve_id = rset.getInt("id");
        String reserve_date = rset.getString("reserve_date");

        System.out.print("Reserve ID: " + reserve_id);
        System.out.print(", Book name: " + book_name_2);
        System.out.print(", Publisher: " + publisher);
        System.out.print(", Author: " + author);
        System.out.print(", Reserved date: " + reserve_date);

        // DELETE FROM ReserveList WHERE id = ?
        pstmt = con.prepareStatement(s4_3);
        pstmt.setInt(1, reserve_id);
        pstmt.executeUpdate();

    }

    // INSERT INTO CheckedList (checker_id,book_num,check_date) VALUES (?,?,?)
    pstmt = con.prepareStatement(s3_1);
    pstmt.setInt(1, id);
    pstmt.setInt(2, book_num);
    pstmt.executeUpdate();

    // s2_4 = "UPDATE BookList can_borrow=? WHERE book_num=?";
    pstmt2 = con.prepareStatement(s2_4);
    pstmt2.setBoolean(1, false);
    pstmt2.setInt(2, book_num);
    pstmt2.executeUpdate();

    con.commit();

    System.out.println("\nSuccess to borrow book, " + book_name_2);
    try {
        Thread.sleep(2000);
    } catch (InterruptedException e1) {
    }

} catch (Throwable e) {
    System.out.println(e.getMessage());
    try {
        Thread.sleep(2000);
    } catch (InterruptedException e1) {
    }
    if (con != null) {
        try {
            con.rollback();
        } catch (SQLException ex) {
        }
```

```java
// T3-3 혼합, b,e
public void reserveBook()   {

    try {
        con.setAutoCommit(false);

        System.out.print("Input book name to reserve: ");
        String book_name = scn.nextLine();

        book_name = "%"+book_name+"%";
        // SELECT count(*) FROM BookList WHERE book_name = ?
        pstmt = con.prepareStatement(s1_5);
        pstmt.setString(1, book_name);
        rset = pstmt.executeQuery();
        rset.next();
        int rowCount = rset.getInt(1);
        if (rowCount>0) {}
        else
            throw new Exception("No information matching the book name!");

        if (rowCount > 1) {
            // SELECT * FROM BookList WHERE book_name = ?
            pstmt = con.prepareStatement(s1_1, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
            pstmt.setString(1, book_name);
            rset = pstmt.executeQuery();

            int total_page = (int) Math.ceil((double)rowCount / 10.0);;
            printBookList(rset, 1, total_page);
            while (true) {
                System.out.println("\nEnter <0> to select book number");
                System.out.print("Enter page number to move > ");
                int page = scn.nextInt();
                if (page == 0) {
                    break;
                }
                else if(page>total_page) {
                    System.out.println("\nExceed the total page number!");
                    continue;
                }
                printBookList(rset, page, total_page);
            }

            System.out.print("\nInput book number to reserve: ");
            int book_num = scn.nextInt();

            //s1_8 = "SELECT count(*) FROM BookList WHERE book_num = ?";
            pstmt = con.prepareStatement(s1_8);
            pstmt.setInt(1, book_num);
            rset = pstmt.executeQuery();
            rset.next();
            int book_i = rset.getInt(1);
            if(book_i!=1) {
                throw new Exception("No book matching!");
            }

            // SELECT * FROM BookList WHERE book_num = ?
            pstmt = con.prepareStatement(s1_4);
            pstmt.setInt(1, book_num);
            rset = pstmt.executeQuery();
        } else {
```

T3-3. 예약

도서명 입력 받음

정보 개수가 0이면 예외 발생

정보가 1개보다 많다면 PrintBookList()를 사용해 원하는 페이지의 정보를 보여주고, 원하는 도서번호를 입력 받아 ResultSet 검색.

1개라면 도서명으로 ResultSet 검색

학생 ID를 입력 받고

앞의 ResultSet에서 얻은 도서 번호로 예약정보 검색

예약 정보가 이미 존재한다면 예외 발생

if(book_i!=1) {
throw new Exception("No book matching!");
}
 	
// SELECT * FROM BookList WHERE book_num = ?
pstmt = con.prepareStatement(s1_4);
pstmt.setInt(1, book_num);
rset = pstmt.executeQuery();
} else {
 	
//s1_5 = "SELECT count(*) FROM BookList WHERE book_name = ?";
pstmt = con.prepareStatement(s1_5);
pstmt.setString(1, book_name);
rset = pstmt.executeQuery();
rset.next();
int book_i = rset.getInt(1);
if(book_i!=1) {
throw new Exception("No book matching!");
}
 	
// SELECT * FROM BookList WHERE book_name = ?
pstmt = con.prepareStatement(s1_1);
pstmt.setString(1, book_name);
rset = pstmt.executeQuery();
}
 	
int book_num = 0;
String book_name_2 = new String();
String publisher = new String();
String author = new String();
 	
while (rset.next()) {
book_num = rset.getInt("book_num");
publisher = rset.getString("publisher");
author = rset.getString("author");
}
 	
System.out.print("Input your student ID: ");
int id = scn.nextInt();
 	
// SELECT * FROM ReserveList WHERE reserver_id = ? and book_num=?
pstmt2 = con.prepareStatement(s4_2);
pstmt2.setInt(1, id);
pstmt2.setInt(2, book_num);
rset2 = pstmt2.executeQuery();
if (rset2.next())
throw new Exception("You already have reservation!");
else {
// INSERT INTO ReserveList (reserver_id, book_num, reserve_date) VALUES (?,?,?)
pstmt = con.prepareStatement(s4_1);
pstmt.setInt(1, id);
pstmt.setInt(2, book_num);
pstmt.executeUpdate();
}
 	
con.commit();
 	
System.out.println("\nSuccess to reserve book");
System.out.print(" >> Book name: " + book_name_2);
System.out.print(", Publisher: " + publisher);
System.out.print(", Author: " + author);
 	
try {
Thread.sleep(2000);
} catch (InterruptedException e1) {
}
 	
} catch (Throwable e) {
System.out.println(e.getMessage());
try {
Thread.sleep(2000);
} catch (InterruptedException e1) {
}
if (con != null) {
try {
con.rollback();
} catch (SQLException ex) {
}
}
}
}
	예약 정보 업데이트

```java
// T3-4 혼합
public void overduelist(int id)   {
    try {
        con.setAutoCommit(false);

        // s6_1 = "SELECT count(*) FROM CheckedList WHERE checker_id = ? and
        // return_date=NULL and check_date<str_to_date(?,'%Y-%m-%d %H:%m:%s')";
        pstmt2 = con.prepareStatement(s6_1);
        pstmt2.setInt(1, id);

        Date date = new Date();
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
        Calendar cal = Calendar.getInstance();
        cal.setTime(date);
        cal.add(Calendar.DATE, -14);
        String format_time = format.format(cal.getTime());
        pstmt2.setString(2, format_time);

        rset2 = pstmt2.executeQuery();
        rset2.next();
        int rowCount = rset2.getInt(1);
        if (rowCount > 0) {
            // s6_2 = "SELECT * FROM CheckedList WHERE checker_id = ? and return_date is NULL
            // and check_date<?";
            pstmt = con.prepareStatement(s6_2);
            pstmt.setInt(1, id);
            pstmt.setString(2, format_time);
            rset = pstmt.executeQuery();
        } else {
            throw new Exception("No overdue list!");
        }

        int current_fee = 0;

        // calculate overdue_fee
        while (rset.next()) {
            int overdue_fee = rset.getInt("overdue_fee");
            int check_id = rset.getInt("id");
            int book_num = rset.getInt("book_num");
            Date check_date = rset.getDate("check_date");

            // s6_3 = "UPDATE CheckedList SET overdue_fee=? WHERE id=?";
            pstmt2 = con.prepareStatement(s6_3);
            int fee_toadd = 0;
            long diff = date.getTime()- check_date.getTime();
            fee_toadd = (int) (diff / (24 * 60 * 60 * 1000));
            pstmt2.setInt(1, fee_toadd * 200);
            current_fee += fee_toadd * 200;
            pstmt2.setInt(2, check_id);
            int s = pstmt2.executeUpdate();
            if(s==0) {
                throw new Exception("Fail to update overdue fee!");
            }
        }

        con.commit();
        System.out.printf("Current overdue fee: %d\n", current_fee);
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e1) {
        }
```

T3-4.
연체료 업데이트 및 조회

메인에서 입력 받은 학생ID로 반납 일자가 지난 대출 정보 조회

결과Set으로 현재 일자를 기준으로 연체료 계산, 업데이트.

```java
// T3-5 혼합
public void payoverduefee(int id)   {
    try {
        con.setAutoCommit(false);

        // s6_1 = "SELECT count(*) FROM CheckedList WHERE checker_id = ? and
        // return_date=NULL and check_date<?";
        pstmt2 = con.prepareStatement(s6_1);
        pstmt2.setInt(1, id);

        Date date = new Date();
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        Calendar cal = Calendar.getInstance();
        cal.setTime(date);
        cal.add(Calendar.DATE, -14);
        String format_time = format.format(cal.getTime());
        pstmt2.setString(2, format_time);

        rset2 = pstmt2.executeQuery();
        rset2.next();
        int rowCount = rset2.getInt(1);
        if (rowCount > 0) {
            // s6_2 = "SELECT * FROM CheckedList WHERE checker_id = ? and return_date=NULL
            // and check_date<?";
            pstmt = con.prepareStatement(s5_1, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
            pstmt.setInt(1, id);
            rset = pstmt.executeQuery();
        } else {
            throw new Exception("No overdue fee information!");
        }

        int total_page = (int) Math.ceil((double)rowCount / 10.0);;
        printOverdueList(rset, 1, total_page);
        while (true) {
            System.out.print("\nEnter page number to move(<0> to exit) > ");
            int page = scn.nextInt();
            if (page == 0) {
                break;
            } else if (page > total_page) {
                System.out.println("\nExceed the total page number!");
                continue;
            }
            printOverdueList(rset, page, total_page);
        }

        System.out.print("Input check ID to pay the overdue fee (<0> to exit) > ");
        int check_id = scn.nextInt();
        if (check_id==0) {
            throw new Exception("Stop to pay!");
        }

        //s6_3 = "UPDATE CheckedList overdue_fee=? WHERE id=?";
        pstmt = con.prepareStatement(s6_3);
        pstmt.setInt(1, 0);
        pstmt.setInt(2, check_id);
        pstmt.executeUpdate();

        con.commit();

        System.out.println("Success to pay!");
```

T3-5
연체료 조회
및 정산

메인에서
입력 받은
학생 ID로
대출 정보 조회

ResultSet을
printOverdueList()를 통해
원하는 페이지
를 보여줌.

계산을 원하는
대출번호를
입력 받고

해당 번호의
Overduefee=0
으로 설정

3) 구현요건 충족에 대한 상세 설명

(1) 검색only 트랜잭션, 변경 트랜잭션, 혼합 트랜잭션은 위 4-2) 트랜잭션 리스트와 5-2) 소스코드 설명에서 확인할 수 있듯 2개 이상 포함되었다. (혼합 트랜잭션: T3-4, T3-5)

(2) (b), (e)를 만족하는 혼합 트랜잭션

① T3-1

동적SQL문	소스코드	설명

S1-4
질의문
	`pstmt = con.prepareStatement(s1_4);`
	
	`System.out.print("Input book number to modify: ");`
	`int book_num = scn.nextInt();`
	
	`clearScreen();`
	
	`pstmt.setInt(1, book_num);`
	
	`rset = pstmt.executeQuery();`
	
	`String book_name = new String();`
	`String publisher = new String();`
	`String author = new String();`
	`boolean can_borrow;`
	
	`while (rset.next()) {`
	`  // Retrieve by column name`
	`  book_name = rset.getString("book_name");`
	`  publisher = rset.getString("publisher");`
	`  author = rset.getString("author");`
	`  can_borrow = rset.getBoolean("can_borrow");`
	
	`  System.out.print("\nBook name: " + book_name);`
	`  System.out.print(", Publisher: " + publisher);`
	`  System.out.print(", Author: " + author);`
	`  if (can_borrow) {`
	`  System.out.println(",  Available");`
	`  } else {`
	`  System.out.println(",  Not Available");`
	`  }`
	`}`
		사용자로부터
		도서번호를
		입력받아
		S1-4질의문의
		값으로
		할당하였다.

S2-2
변경문
	``
	switch (func) {
	case 1:
	System.out.print("New book name: ");
	book_name = scn.nextLine();
	break;
	case 2:
	System.out.print("New publisher: ");
	publisher = scn.nextLine();
	break;
	case 3:
	System.out.print("New author: ");
	author = scn.nextLine();
	break;
	}
	
	pstmt = con.prepareStatement(s2_2);
	pstmt.setInt(4, book_num);
	pstmt.setString(1, book_name);
	pstmt.setString(2, publisher);
	pstmt.setString(3, author);
	
	pstmt.executeUpdate();
	``
		새로 입력 받은 한 컬럼을 제외하고
		위의 질의문에서 얻은 값들을 S2-2 변경문 값으로 할당하였다.

② T3-3

동적SQL문	소스코드	설명

S1-1
질의문
	``
	System.out.print("Input book name to reserve: ");
	String book_name = scn.nextLine();
	
	book_name = "%"+book_name+"%";
	
	// SELECT * FROM BookList WHERE book_name = ?
	pstmt = con.prepareStatement(s1_1);
	pstmt.setString(1, book_name);
	rset = pstmt.executeQuery();
	}
	
	int book_num = 0;
	String book_name_2 = new String();
	String publisher = new String();
	String author = new String();
	
	while (rset.next()) {
	book_num = rset.getInt("book_num");
	publisher = rset.getString("publisher");
	author = rset.getString("author");
	}
	``
		사용자로부터 도서명를 입력 받아
		S1-1 질의문의 값으로 할당하였다.

S4-1
변경문
	``
	// INSERT INTO ReserveList (reserver_id, book_num, reserve_date) VALUES (?,?,?)
	pstmt = con.prepareStatement(s4_1);
	pstmt.setInt(1, id);
	pstmt.setInt(2, book_num);
	pstmt.executeUpdate();
	``
		위에서 얻은 book_num 값을 S4-1 변경문의 값으로 할당하였다..

(3) (a)~(e)를 모두 만족하는 혼합 트랜잭션: T3-2

	동적 SQL 문	소스코드	설명

(a)	S1-5	`// SELECT count(*) FROM BookList WHERE book_name = ?`
		`pstmt = con.prepareStatement(s1_5);`
		`pstmt.setString(1, book_name);`
		`rset = pstmt.executeQuery();`
			2개 이상의 질의문을 사용하였다..

	S1-1	`// SELECT * FROM BookList WHERE book_name = ?`
		`pstmt = con.prepareStatement(s1_1, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);`
		`pstmt.setString(1, book_name);`
		`rset = pstmt.executeQuery();`	

	S1-8	`//s1_8 = "SELECT count(*) FROM BookList WHERE book_num = ?";`
		`pstmt = con.prepareStatement(s1_8);`
		`pstmt.setInt(1, book_num);`
		`rset = pstmt.executeQuery();`	

	S1-4	`// SELECT * FROM BookList WHERE book_num = ?`
		`pstmt = con.prepareStatement(s1_4);`
		`pstmt.setInt(1, book_num);`
		`rset = pstmt.executeQuery();`	

	S4-2	`// SELECT * FROM ReserveList WHERE reserver_id = ? and book_num=?`
		`pstmt2 = con.prepareStatement(s4_2);`
		`pstmt2.setInt(1, id);`
		`pstmt2.setInt(2, book_num);`
		`rset2 = pstmt2.executeQuery();`	

(b)	S1-5	`System.out.print("Input book name to borrow: ");`
		`String book_name = scn.nextLine();`
		`book_name = "%"+book_name+"%";`
		`// SELECT count(*) FROM BookList WHERE book_name = ?`
		`pstmt = con.prepareStatement(s1_5);`
		`pstmt.setString(1, book_name);`
		`rset = pstmt.executeQuery();`
			사용자에게 입력받은 도서명을 S1-5질의문 값에 할당하였다.

<table>
<tr>
<td>(c)</td>
<td>S1-1
S4-2</td>
<td>

```java
        // SELECT * FROM BookList WHERE book_name like ?
        pstmt = con.prepareStatement(s1_1);
        pstmt.setString(1, book_name);
        rset = pstmt.executeQuery();
    }

    int book_num = 0;
    String publisher = new String();
    String author = new String();
    String book_name_2 = new String();
    boolean can_borrow;

    while (rset.next()) {
        book_num = rset.getInt("book_num");
        book_name_2 = rset.getString("book_name");
        publisher = rset.getString("publisher");
        author = rset.getString("author");
        can_borrow = rset.getBoolean("can_borrow");
        if (!can_borrow) {
            throw new Exception("Can't borrow!");
        }
    }

    System.out.print("Input your student ID: ");
    int id = scn.nextInt();

    // SELECT * FROM ReserveList WHERE reserver_id = ? and book_num=?
    pstmt2 = con.prepareStatement(s4_2);
    pstmt2.setInt(1, id);
    pstmt2.setInt(2, book_num);
    rset2 = pstmt2.executeQuery();
```

</td>
<td>S1-1에서 얻은 book_num 값을 S4-2에 할당해 사용하였다.</td>
</tr>
<tr>
<td>(d)</td>
<td>S1-1
S4-2</td>
<td>

```java
        // SELECT * FROM BookList WHERE book_name like ?
        pstmt = con.prepareStatement(s1_1);
        pstmt.setString(1, book_name);
        rset = pstmt.executeQuery();
    }

    int book_num = 0;
    String publisher = new String();
    String author = new String();
    String book_name_2 = new String();
    boolean can_borrow;

    while (rset.next()) {
        book_num = rset.getInt("book_num");
        book_name_2 = rset.getString("book_name");
        publisher = rset.getString("publisher");
        author = rset.getString("author");
        can_borrow = rset.getBoolean("can_borrow");
        if (!can_borrow) {
            throw new Exception("Can't borrow!");
        }
    }

    System.out.print("Input your student ID: ");
    int id = scn.nextInt();

    // SELECT * FROM ReserveList WHERE reserver_id = ? and book_num=?
    pstmt2 = con.prepareStatement(s4_2);
    pstmt2.setInt(1, id);
    pstmt2.setInt(2, book_num);
    rset2 = pstmt2.executeQuery();
    while (rset2.next()) {
        // Retrieve by column name
        System.out.println("You have reservation information!");
        int reserve_id = rset.getInt("id");
        String reserve_date = rset.getString("reserve_date");

        System.out.print("Reserve ID: " + reserve_id);
        System.out.print(", Book name: " + book_name_2);
        System.out.print(", Publisher: " + publisher);
        System.out.print(", Author: " + author);
        System.out.print(", Reserved date: " + reserve_date);

        // DELETE FROM ReserveList WHERE id = ?
        pstmt = con.prepareStatement(s4_3);
        pstmt.setInt(1, reserve_id);
        pstmt.executeUpdate();

    }
```

</td>
<td>S1-1에서 얻은 Publisher, author값과

S4-2에서 얻은 Reserve_id, Reserve_date 값을 조합해 출력하였다.</td>
</tr>
</table>

(e)	S1-1
	S2-3
	S2-4
		
		// SELECT * FROM BookList WHERE book_name like ?
		pstmt = con.prepareStatement(s1_1);
		pstmt.setString(1, book_name);
		rset = pstmt.executeQuery();
		}
		
		int book_num = 0;
		String publisher = new String();
		String author = new String();
		String book_name_2 = new String();
		boolean can_borrow;
		
		while (rset.next()) {
		book_num = rset.getInt("book_num");
		book_name_2 = rset.getString("book_name");
		publisher = rset.getString("publisher");
		author = rset.getString("author");
		can_borrow = rset.getBoolean("can_borrow");
		if (!can_borrow) {
		throw new Exception("Can't borrow!");
		}
		}
		
		// INSERT INTO CheckedList (checker_id,book_num,check_date) VALUES (?,?,?)
		pstmt = con.prepareStatement(s3_1);
		pstmt.setInt(1, id);
		pstmt.setInt(2, book_num);
		pstmt.executeUpdate();
		
		// s2_4 = "UPDATE BookList can_borrow=? WHERE book_num=?";
		pstmt2 = con.prepareStatement(s2_4);
		pstmt2.setBoolean(1, false);
		pstmt2.setInt(2, book_num);
		pstmt2.executeUpdate();
			S1-1에서 얻은 Book_num 값을
			
			S2-3, S2-4 변경문 값에 할당하였다.

6. 테스트 레코드 생성

1) 랜덤 데이터 생성: 완전 랜덤한 값을 생성한 레포트3과 달리, 데이터 검색에 용이하게 값을 생성해주었다.

코드	설명

`public static int createCSV(String[] list, String title, String filepath) {`
`int resultCount = 0;`
`try {`
`FileWriter fw = new FileWriter(filepath+"/"+title+".csv");`
 	
`for(String dom : list) {`
`fw.append(dom);`
`fw.append('\n');`
`resultCount++;`
`}`
 	
`fw.flush();`
`fw.close();`
 	
`}catch(Exception e) {`
`e.printStackTrace();`
`}`
`return resultCount;`
`}`
	입력받은 String배열을 담은 CSV 파일을 생성하는 함수

`public static String randomStr(int length) {`
 	
`Random rnd = new Random();`
`String randomStr = new String();`
 	
`for(int i=0; i<rnd.nextInt(length)+5; i++) {`
`int rindex = rnd.nextInt(2);`
`if(rindex==0) {`
`randomStr += (char)((int)(rnd.nextInt(26))+97);`
`}`
`else {`
`randomStr += (char)((int)(rnd.nextInt(26))+65);`
`}`
`}`
`return randomStr;`
`}`
	입력받은 length+5 길이의 랜덤 영문 문자열 생성 함수

`public static int randomInt(int range) {`
 	
`Random rnd = new Random();`
 	
`return rnd.nextInt(range);`
`}`
	0~(range-1) 범위의 랜덤 정수를 생성하는 함수

`public static int randomIntLength(int length) {`
 	
`Random rnd = new Random();`
`int temp = 0;`
 	
`for(int i=0; i<length; i++) {`
`temp += Math.pow(10, length-i-1)*(randomInt(8)+1);`
`}`
`return temp;`
`}`
	입력받은 길이의 랜덤 정수를 생성하는 함수

`public static Date randomDate(int start, int end) {`
 	
`Date tempDate = new Date();`
`Calendar startDate = Calendar.getInstance();`
`startDate.setTime(tempDate);`
`startDate.add(Calendar.MONTH, start);`
`Calendar endDate = Calendar.getInstance();`
`endDate.setTime(tempDate);`
`endDate.add(Calendar.MONTH, end);`
 	
`long random = ThreadLocalRandom.current().nextLong(startDate.getTimeInMillis(), endDate.getTimeInMillis());`
`Date date = new Date(random);`
 	
`return date;`
`}`
	(Start)개월 전부터 (end) 개월 후까지의 "2020-**-** **:**:*" 형식으로 랜덤 Datetime을 생성하는 함수

- Student.csv 생성

Student Table의 데이터 생성	`static String[] student_id = new String[100000];`
	
	`public static void createStudent() {`
	
	`String[] list = new String[100000];`
	`String temp = new String();`
	`String id_temp = new String();`
	`int[] id = {2016, 2017, 2018, 2019};`
	
	`for(int i=0; i<100000; i++) {`
	
	`id_temp = Integer.toString(id[randomInt(4)]) + String.format("%05d", i); //id`
	`student_id[i] =new String(id_temp);//book_num`
	`temp = id_temp;`
	`temp += ",";`
	`temp += randomStr(10); //name`
	
	`list[i] = new String(temp);`
	`}`
	
	`createCSV(list, "Student", "C:/Users/pjpp8/MySQL/mysql-sandboxes/3310/mysql-files/");`
	`}`

설명	10만개 데이터 생성 ID: 길이 4인 랜덤 인트 + i값 5자리 ID 생성 후 다른 Table의 외래키로 사용하기 위해 배열에 저장

	Name: 길이 15이하의 랜덤 문자열
데이터 생성 결과 (일부)	

	A	B
1	201800000	zWFGgOB
2	201600001	puckfEO
3	201700002	BGSjancpR
4	201900003	iCdZXmhtA
5	201800004	WggwNYT
6	201900005	SVPuraL
7	201800006	vNIHtYps
8	201800007	NPPOzlZQz
9	201700008	jIrVrJlg

A: IDInt), 학생 번호

B: *name*(char), 학생 이름

- BookList.csv 생성

BookList Table의 데이터 생성

```
static String[] book_number = new String[100000];

public static void createBookList() {

    String[] list = new String[100000];
    String temp = new String();
    String book_num_temp = new String();
    String[] name = {"book", "name", "bn", "computer", "science", ""};
    String[] publish_pre = {"red", "pearson", "oxford", "ybm", "cambridge", "happy", "fun"};
    String[] and = {" ", "&", ""};
    String[] publish_suf = {"publish", "brick", "build", "logman", "future","education","press","learning", "university"};
    String[] author_list = {"Anabel", "Anastasia", "Adam", "Jacob", "Mason", "William", "Jayden", "Noah", "Michael", "Ethan",
    for(int i=0; i<100000; i++) {

        book_num_temp = Integer.toString(randomIntLength(3)) + String.format("%05d", i);
        book_number[i] =new String(book_num_temp);//book_num
        temp = book_num_temp;
        temp += ",";
        temp += name[randomInt(6)]+randomStr(5)+name[randomInt(6)]; //bookname
        temp += ",";
        temp += publish_pre[randomInt(7)]+and[randomInt(3)]+publish_suf[randomInt(9)]; //publisher
        temp += ",";
        temp += author_list[randomInt(15)]+randomStr(3); //author
        temp += ",";

        if (randomInt(2)==0) {
            temp += "0";
        }
        else {
            temp+="1";
        }

        list[i] = new String(temp);
    }

    createCSV(list, "BookList", "C:/Users/pjpp8/MySQL/mysql-sandboxes/3310/mysql-files/");
}
```

설명	10만개의 데이터 생성 Book_num: 길이 4인 랜덤 정수 + i값을 길이 5로 채워서 생성 Book_num은 다른 Table에서 외래키로 사용하므로 배열에 저장해둠 Book_name, Publisher, Author: 위와 같이 문자열을 생성해둔 후 랜덤하게 사용

	Can_borrow: 0 또는 1로 생성
데이터 생성 결과 (일부)	

	A	B	C	D	E
1	42200000	bookloKuPxscience	ybm&education	JaydenVGYMLZ	1
2	62900001	bnxPyuYBmMbn	oxford&university	AnabelqwcnszE	0
3	29500002	bookZqkDbscience	funpress	Noahciyrps	0
4	34600003	nameaZLEitTKname	happy&build	EthanvypALg	0
5	98800004	scienceTDYmGFcomp	red&press	DanielOkUdM	1
6	23200005	booksNyrPgname	ybm publish	AlexanderiHpaLN	1
7	82200006	bookEnRJAbbook	ybm publish	EthanHjrwXY	0
8	29400007	namemOjiucomputer	happy university	JacobUZZWI	1
9	64600008	nametyuxdL	oxford&university	AnastasiaCcbmj	0
10	78200009	EvQvuGTJpname	redpress	MichaelYDuqaIA	1

A: *book_num*(Int), 도서 고유번호

B: *bookname*(Varchar), 도서명

C: *publisher*(Varchar), 출판사

D: *author*(Varchar), 작가

E: *can_borrow*(Boolean), 현재 대출 가능한지 여부

- CheckedList.csv 생성

CheckedList Table의 데이터 생성

```java
public static void createCheckedList() {

    String[] list = new String[250000];
    String temp = new String();

    for(int i=0; i<250000; i++) {

        temp = Integer.toString(i+1);
        temp += ",";
        temp += student_id[randomInt(100000)]; //checker_id
        temp += ",";
        temp += book_number[randomInt(100000)]; //book_num
        temp += ",";
        Date date = randomDate(-6,0);
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        String format_time = format.format(date.getTime());
        temp += format_time; //check_date
        temp += ",";
        if(randomInt(2)==1) {
            Calendar returndate = Calendar.getInstance();
            returndate.setTime(date);
            returndate.add(Calendar.DATE, randomInt(90));
            SimpleDateFormat format_2 = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            String format_time_2 = format_2.format(returndate.getTime());
            temp += format_time_2; //return_date
        }
        else {
            temp += "\\N";
        }
        temp += ", 0";

        list[i] = new String(temp);
    }

    createCSV(list, "CheckedList", "C:/Users/pjpp8/MySQL/mysql-sandboxes/3310/mysql-files/");
}
```

설명	25만개의 데이터 생성 Checker_id: Student 생성에 사용한 값을 사용 Book_num: BookList 생성에 사용한 값을 사용 Check_date: 6달 전부터 지금까지 랜덤한 날짜 생성 Retrun_date: Check_date 이후의 랜덤 날짜나 Null로 생성 Overdue_fee: 0으로 초기화
데이터 생성 결과 (일부)	

	A	B	C	D	E	F
1	1	201743361	47449696	2020-01-19 19:06	₩N	0
2	2	201837203	44160442	2020-01-09 20:05	₩N	0
3	3	201995650	83267956	2020-03-03 17:05	₩N	0
4	4	201897206	45743495	2020-03-26 14:12	2020-06-18 14:12	0
5	5	201809205	45322424	2020-03-26 0:19	2020-06-20 0:19	0
6	6	201890056	91670531	2020-04-23 23:13	₩N	0
7	7	201890334	69310407	2019-12-21 12:42	₩N	0
8	8	201614985	93391686	2019-12-15 15:29	2020-01-31 15:29	0
9	9	201974033	37538466	2020-06-06 20:23	2020-07-10 20:23	0
10	10	201899027	13432679	2020-02-12 8:58	2020-04-11 8:58	0

A: *id*(Int), PK

B: *checker_id*(Int), 대출자 id

C: *book_num*(Int), 대출도서 번호, 외래키

D: *check_date*(Datetime), 대출 날짜

E: *return_date*(Datetime), 반납 날짜

F: *overdue_fee*(Int), 연체료

- ReserveList.csv 생성

ReserveList Table의 데이터 생성

```java
public static void createReserveList() {

    String[] list = new String[250000];
    String temp = new String();

    for(int i=0; i<250000; i++) {

        temp = Integer.toString(i+1);
        temp += ",";
        temp += student_id[randomInt(100000)]; //reserver_id
        temp += ",";
        temp += book_number[randomInt(100000)]; //book_num
        temp += ",";
        Date date = randomDate(-4,0);
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        String format_time = format.format(date.getTime());
        temp += format_time; //reserve_date

        list[i] = new String(temp);
    }

    createCSV(list, "ReserveList", "C:/Users/pjpp8/MySQL/mysql-sandboxes/3310/mysql-files/");
}
```

설명

<table>
<tr><td></td><td>25만개의 데이터 생성
reserver_id: Student 생성에 사용한 값을 사용
Book_num: BookList 생성에 사용한 값을 사용
Reserve_date: 4달 전부터 지금까지 랜덤한 날짜 생성</td></tr>
<tr><td>데이터 생성
결과
(일부)</td><td>
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>1</td><td>1</td><td>201728657</td><td>47848090</td><td>2020-04-07 15:46</td></tr>
<tr><td>2</td><td>2</td><td>201894723</td><td>67951884</td><td>2020-05-16 21:21</td></tr>
<tr><td>3</td><td>3</td><td>201766660</td><td>61842334</td><td>2020-04-30 2:04</td></tr>
<tr><td>4</td><td>4</td><td>201653788</td><td>95342835</td><td>2020-03-11 3:44</td></tr>
<tr><td>5</td><td>5</td><td>201670855</td><td>57107631</td><td>2020-04-22 15:16</td></tr>
<tr><td>6</td><td>6</td><td>201738781</td><td>19307604</td><td>2020-02-28 0:15</td></tr>
<tr><td>7</td><td>7</td><td>201899511</td><td>89775662</td><td>2020-02-28 0:02</td></tr>
<tr><td>8</td><td>8</td><td>201754293</td><td>83248235</td><td>2020-06-06 9:25</td></tr>
<tr><td>9</td><td>9</td><td>201979501</td><td>63146194</td><td>2020-03-17 16:44</td></tr>
<tr><td>10</td><td>10</td><td>201647901</td><td>16260253</td><td>2020-05-27 5:53</td></tr>
</table>
A: id(Int), PK
B: reserver_id(Int), 예약자 id
C: book_num(Int), 대출도서 번호, 외래키
D: reserve_date(Datetime), 예약 날짜
</td></tr>
</table>

2) 데이터 로드: 아래와 같이 JDBC 코드를 통해 데이터를 로드했다.

```
public static void LoadDataInfileExcute() throws Exception {

    Connection conn = null;

    Statement stmt = null;

    Class.forName(driver);

    try {

        conn = DriverManager.getConnection(url, id, pw);

        stmt = conn.createStatement();

        String path = "C:/Users/pjpp8/MySQL/mysql-sandboxes/3310/mysql-files/Student.csv"; //파일경로 및 파일명

                String sql = "load data infile '"+ path +

                "' INTO TABLE Student " +

                "CHARACTER SET euckr " + //인코딩

                "FIELDS TERMINATED BY ',' " + // ',' 기준으로 파싱

                "LINES TERMINATED BY '\n'";

                //라인끝

        boolean result = stmt.execute(sql);

        path = "C:/Users/pjpp8/MySQL/mysql-sandboxes/3310/mysql-files/BookList.csv"; //파일경로 및 파일명

        sql = "load data infile '"+ path +

                "' INTO TABLE BookList " +

                "CHARACTER SET euckr " + //인코딩

                "FIELDS TERMINATED BY ',' " + // ',' 기준으로 파싱

                "LINES TERMINATED BY '\n'";

                //라인끝

       result = stmt.execute(sql);
```

```
    path = "C:/Users/pjpp8/MySQL/mysql-sandboxes/3310/mysql-files/CheckedList.csv"; //파일경로 및 파일명

     sql = "load data infile '"+ path +
             "' INTO TABLE CheckedList " +
             "CHARACTER SET euckr " + //인코딩
             "FIELDS TERMINATED BY ',' " + // ',' 기준으로 파싱
             "LINES TERMINATED BY '\n'";
             //라인끝
    result = stmt.execute(sql);
     path = "C:/Users/pjpp8/MySQL/mysql-sandboxes/3310/mysql-files/ReserveList.csv"; //파일경로 및 파일명

     sql = "load data infile '"+ path +
             "' INTO TABLE ReserveList " +
             "CHARACTER SET euckr " + //인코딩
             "FIELDS TERMINATED BY ',' " + // ',' 기준으로 파싱
             "LINES TERMINATED BY '\n'";
             //라인끝
    result = stmt.execute(sql);
} catch(Exception e){
    System.out.println("e: " + e);
} finally {
    conn.close();
    stmt.close();
}
```

3) 로드 결과: 아래와 같이 Select질의문을 통해 데이터가 로드된 것을 확인하였다.

```
mysql> select count(*) from student;
+----------+
| count(*) |
+----------+
|   100000 |
+----------+
1 row in set (0.01 sec)
```

```
mysql> select count(*) from bookList;
+----------+
| count(*) |
+----------+
|   100000 |
+----------+
1 row in set (0.01 sec)
```

```
mysql> select count(*) from CheckedList;
+----------+
| count(*) |
+----------+
|   250000 |
+----------+
1 row in set (0.01 sec)
```

```
mysql> select count(*) from ReserveList;
+----------+
| count(*) |
+----------+
|   250000 |
+----------+
1 row in set (0.01 sec)
```

-

7. 응용 실행 데모

1) 도서 검색

메인 화면에서 1을 입력하면 도서 검색이 가능하다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:26:15)
Welcome to libary!

--------Select the option number---------
  1. Search for books
  2. Borrow a book
  3. Reserve a book
  4. Return a book
  5. My page
  6. Management

 >> 1
```

① 도서명으로 검색

1번을 선택한 후 도서명을 입력하면 입력 받은 값을 포함하는 모든 레코드를 볼 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:26:15)




--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 1
Input book name: computer
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:26:15)

Current Page: 1/3066

        Book num                Book name                    Publisher                    Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11101146            computerYVnXCUoubn               pearsonbuild                 DanielTIBfXeC            Not Available
        11101294            IWVlTfcomputer                   fun&future                   DanielpplFns             Not Available
        11102451            bookIHjdSnpcomputer              fun&press                    MasonVFkDPmW             Not Available
        11107528            computerUCVehtbook               pearson&learning             MichaelciCIV             Not Available
        11108024            yVZSEaBcomputer                  pearson&press                AnabelSgaKyg               Available
        11109456            computerehRDmEBbook              happyuniversity              AnastasiaBMTvH             Available
        11111726            namedFjSLcomputer                oxfordeducation              MichaelcAmzw               Available
        11114447            scienceHDmBWAcomputer            happy&build                  NoahjaHLnR               Not Available
        11116310            bnGsLDeqcomputer                 ybmbrick                     JaydenrtRTPi             Not Available
        11116878            namevVLsabcomputer               pearson&logman               JacobuMANv               Not Available


Enter page number to move(<0> to exit) >
<
```

아래에 원하는 페이지 번호를 입력하면 다음과 같이 해당 페이지의 값들을 볼 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:26:15)
Current Page: 1/3066

        Book num                Book name                       Publisher                       Author                  etc.
-------------------------------------------------------------------------------------------------------------------------------------
        11101146            computerYVnXCUoubn              pearsonbuild                    DanielTIBfXeC           Not Available
        11101294            IWVlTfcomputer                  fun&future                      DanielpplFns            Not Available
        11102451            bookIHjdSnpcomputer             fun&press                       MasonVFkDPmW            Not Available
        11107528            computerUCVehtbook              pearson&learning                MichaelciCIV            Not Available
        11108024            yVZSEaBcomputer                 pearson&press                   AnabelSgaKyg              Available
        11109456            computerehRDmEBbook             happyuniversity                 AnastasiaBMTvH            Available
        11111726            namedFjSLcomputer               oxfordeducation                 MichaelcAmzw              Available
        11114447            scienceHDmBWAcomputer           happy&build                     NoahjaHLnR              Not Available
        11116310            bnGsLDeqcomputer                ybmbrick                        JaydenrtRTPi            Not Available
        11116878            namevVLsabcomputer              pearson&logman                  JacobuMANv              Not Available


Enter page number to move(<0> to exit) > 5
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:26:15)
Current Page: 5/3066

        Book num                Book name                       Publisher                       Author                  etc.
-------------------------------------------------------------------------------------------------------------------------------------
        11188404            bnbxKeJYecomputer               funfuture                       AidenXkXKTtc            Not Available
        11192610            computerXrDPqbn                 oxforduniversity                MichaelUNObH              Available
        11195130            computerTyfZbTVbn               ybmfuture                       AndrewVjjLd               Available
        11204023            computerieKmijBbn               oxford&learning                 MichaelxoUeOs           Not Available
        11208625            computerXgLGASname              oxfordlearning                  MasontgMGwO             Not Available
        11209305            fpKYeFcomputer                  funfuture                       WilliamODJPzx             Available
        11211774            computertqlcjscience            red&learning                    DanielWawvv               Available
        11213926            namextQsluncomputer             ybm university                  MasontUqlhNV            Not Available
        11215965            computernXZrqpQIbook            ybm build                       AdamrPUgYzz             Not Available
        11216992            WLgNlEvcomputer                 oxfordbrick                     DanielgoVDG               Available


Enter page number to move(<0> to exit) >
```

② 출판사로 검색

2번을 선택한 후 검색할 값을 입력하면 아래와 같이 값을 포함하는 모든 레코드를 볼 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:31:34)



--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 2
Input publisher: university
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:31:34)
Current Page: 1/1115

        Book num                Book name                     Publisher                     Author                    etc.
-----------------------------------------------------------------------------------------------------------------------------
        11109456            computerehRDmEBbook            happyuniversity               AnastasiaBMTvH             Available
        11126744            nameLlRwAbn                    happy&university              WilliamoWQRxQ            Not Available
        11127841            NYavpySObook                   red university                WilliamwGdKvv            Not Available
        11132588            nameEOSTbKgname                red university                MasonwqLWtVG             Not Available
        11134557            scienceBSWSrVCcomputer         red&university                MasonPzATTp                Available
        11137632            computerxBKycDL                oxford university             MasonazNNFF                Available
        11139541            bookBsLFXdname                 ybm&university               EthanZFTrmo                Available
        11141011            bnMIbzjkbook                   happyuniversity               AnabelFZVqY                Available
        11142621            computerOhrWaAcomputer         fun&university                TylerFberufd               Available
        11149058            sciencewgxewtObn               red&university                Williamjyinec            Not Available


Enter page number to move(<0> to exit) >
```

또한 pagination에서 전체 페이지 크기보다 큰 값을 입력하면 아래와 같이 문구가 출력된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:41:40)
Current Page: 1/1446

        Book num                Book name                     Publisher                     Author                    etc.
-----------------------------------------------------------------------------------------------------------------------------
        11101294            IWVlTfcomputer                 fun&future                    DanielpplFns             Not Available
        11102451            bookIHjdSnpcomputer            fun&press                     MasonVFkDPmW             Not Available
        11107381            bnaGSgFmZbook                  fun&learning                  AidenCoWzm               Not Available
        11107526            scienceEVBVzfWscience          fun&education                 AnastasiaMjKxV             Available
        11117386            scienceaiDTFfLZcomputer        funfuture                     AdamkGrnU                  Available
        11126318            bnevGTSGnLbook                 funlearning                   TylerLcXuxK              Not Available
        11128564            scienceSpkQCgsbn               funbrick                      AdamCYLYEW                 Available
        11134581            bngqKsDbn                      fun build                     TylerczIRIQ                Available
        11135482            scienceluBnBQnS                fun logman                    EthanxZgMDM                Available
        11142569            bookBpKaawname                 funbrick                      DanielxvGTfB             Not Available


Enter page number to move(<0> to exit) > 2000

Exceed the total page number!


Enter page number to move(<0> to exit) >
```

③ 저자로 검색

3번을 선택한 후 검색하기 원하는 값을 입력하면 해당 값을 포함하는 레코드들을 모두 볼 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:38:33)



--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 3
Input author: Adam
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:38:33)
Current Page: 1/659

        Book num                Book name                    Publisher                    Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11114270            gcboIKK                      red&publish                  AdamgNPwh                 Available
        11116308            scienceIEDbgRebook           pearson&publish              AdamiIEGdG                Available
        11117386            scienceaiDTFfLZcomputer      funfuture                    AdamkGrnU                 Available
        11128564            scienceSpkQCgsbn             funbrick                     AdamCYLYEW                Available
        11133469            bnmCaBvrCcomputer            oxford learning              AdamQpDlBY                Available
        11142837            bookmsmAuqhbook              pearson&brick                AdamFiXTe             Not Available
        11152058            bnJcOvyxpname                pearson future               AdamcPyfWNo               Available
        11153230            bnKpIkPlscience              ybm brick                    AdamFytWpj                Available
        11170746            scienceKwuggobook            cambridgepress               AdamLMmhp             Not Available
        11174700            computerSeCDydJname          cambridge future             Adampagzr                 Available


Enter page number to move(<0> to exit) >
```

④ 도서 번호로 검색

4번을 선택한 후 도서 번호를 입력하면 해당 번호를 가진 도서 정보를 볼 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:41:40)



--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 4
Input book number: 11114270
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:41:40)

Current Page: 1/1
        Book num                Book name                    Publisher                    Author                    etc.
-----------------------------------------------------------------------------------------------------------------------
        11114270            gcboIKK                         red&publish                  AdamgNPwh                 Available

Enter page number to move(<0> to exit) >
<
```

2) 대출, 예약, 반납

① 대출

메인 화면에서 2번을 선택한 후 대출하길 원하는 도서명을 입력한다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:41:40)

Welcome to libary!

--------Select the option number(<0> to exit)---------
  1. Search for books
  2. Borrow a book
  3. Reserve a book
  4. Return a book
  5. My page
  6. Management

 >> 2
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:41:40)

Input book name to borrow: science
```

해당 값을 포함하는 도서 정보가 여러 개라면, 아래와 같이 레코드들을 페이지별로 확인할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:41:40)



Current Page: 1/3054

        Book num                Book name                     Publisher                     Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11104620          scienceaAuEftbn                happylogman                   AnastasiafaUxIM          Not Available
        11107526          scienceEVBVzfWscience          fun&education                 AnastasiaMjKxV             Available
        11108520          nameqILAkGscience              oxford education              WilliamDugAlGE           Not Available
        11114447          scienceHDmBWAcomputer          happy&build                   NoahjaHLnR               Not Available
        11116308          scienceIEDbgRebook             pearson&publish               AdamiIEGdG                 Available
        11117386          scienceaiDTFfLZcomputer        funfuture                     AdamkGrnU                  Available
        11118999          scienceRQWjmSname              happy build                   MasonDowbvwU               Available
        11120824          EcLglEcscience                 ybmlogman                     AidenWtLgnm              Not Available
        11122198          bnDJQSQuscience                cambridge&future              AndrewKRAMjSQ            Not Available
        11127591          bookGcYbbbTSscience            oxford&build                  AndrewCwVfTi               Available


Enter <0> to select book number
Enter page number to move > 3
```

목록에서 도서 번호를 확인한 후 대출할 도서 번호를 아래와 같이 입력한다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:58:07)

Current Page: 3/3054

        Book num                Book name                     Publisher                     Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11153230          bnKpIkPlscience                ybm brick                     AdamFytWpj                 Available
        11159722          computeruZdsKwlscience         cambridge future              WilliamIkodMD            Not Available
        11160013          scienceRRSpHi                  pearson&university            MichaelrasYo             Not Available
        11163121          WqtJnbnscience                 pearson publish               MichaelJIDTC             Not Available
        11166778          scienceLIrPuSbn                funpress                      JacobJRcriT                Available
        11167566          scienceeoFZwGmucomputer        oxford press                  EthanQoPaJ                 Available
        11170746          scienceKwuggobook              cambridgepress                AdamLMmhp                Not Available
        11172783          computerBqlFMBscience          red future                    TyleroEnZGJ                Available
        11177047          scienceZDxWlmcomputer          happy brick                   MasonsMBdjtl             Not Available
        11177655          sciencewulzNname               pearson university            TylerXjjaGmV             Not Available


Enter <0> to select book number
Enter page number to move > 0
Input book number to borrow: 11159722
Can't borrow!
```

학생 ID를 입력하면 성공적으로 도서를 대출했다는 메시지가 출력된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:59:23)

Current Page: 3/3054

        Book num                Book name                     Publisher                     Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11153230          bnKpIkPlscience                ybm brick                     AdamFytWpj                 Available
        11159722          computeruZdsKwlscience         cambridge future              WilliamIkodMD            Not Available
        11160013          scienceRRSpHi                  pearson&university            MichaelrasYo             Not Available
        11163121          WqtJnbnscience                 pearson publish               MichaelJIDTC             Not Available
        11166778          scienceLIrPuSbn                funpress                      JacobJRcriT                Available
        11167566          scienceeoFZwGmucomputer        oxford press                  EthanQoPaJ                 Available
        11170746          scienceKwuggobook              cambridgepress                AdamLMmhp                Not Available
        11172783          computerBqlFMBscience          red future                    TyleroEnZGJ                Available
        11177047          scienceZDxWlmcomputer          happy brick                   MasonsMBdjtl             Not Available
        11177655          sciencewulzNname               pearson university            TylerXjjaGmV             Not Available


Enter <0> to select book number
Enter page number to move > 0
Input book number to borrow: 11153230
Input your student ID: 201600001

Success to borrow book, bnKpIkPlscience
```

만약 도서 번호를 가진 도서가 존재하지 않는다면 아래와 같이 에러 메시지가 출력된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 1:56:24)

Current Page: 1/10000

        Book num                Book name                       Publisher                       Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11101146            computerYVnXCUoubn              pearsonbuild                    DanielTIBfXeC             Not Available
        11101294            IWVlTfcomputer                  fun&future                      DanielpplFns              Not Available
        11101445            bnljBcSwname                    redlogman                       JacobAAZKy                Not Available
        11102451            bookIHjdSnpcomputer             fun&press                       MasonVFkDPmW              Not Available
        11104620            scienceaAuEftbn                 happylogman                     AnastasiafaUxIM           Not Available
        11105752            bnWSObYname                     cambridge&press                 WilliamGGxtc              Not Available
        11106757            bookogipYIjbook                 ybm&education                   AnabelbdaWI               Not Available
        11107381            bnaGSgFmZbook                   fun&learning                    AidenCoWzm                Not Available
        11107526            scienceEVBVzfWscience           fun&education                   AnastasiaMjKxV              Available
        11107528            computerUCVehtbook              pearson&learning                MichaelciCIV              Not Available


Enter <0> to select book number
Enter page number to move > 0
Input book number to borrow: 11111111
No book matching!
```

만약 입력 받은 도서명을 가진 정보의 개수가 1개라면 바로 학생 ID를 입력 받은 후 대출할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:01:21)

Input book name to borrow: booksNyrPgname
Input your student ID: 201800000

Success to borrow book, booksNyrPgname
```

도서명을 가진 정보가 존재하지 않는다면 아래와 같이 에러 메시지가 출력된 후 종료된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 7:23:22)

Input book name to borrow: asdfadfadfasdfdf
No information matching the book name!
```

② 예약

메인 화면에서 3번을 선택한 후 예약을 원하는 도서명을 입력한다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:02:42)
Welcome to libary!

--------Select the option number(<0> to exit)---------
  1. Search for books
  2. Borrow a book
  3. Reserve a book
  4. Return a book
  5. My page
  6. Management

 >> 3
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:02:42)

Input book name to reserve: book
```

도서명을 가진 도서 정보의 개수가 1개보다 많다면 아래와 같이 해당 레코드들을 페이지로 볼 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:02:42)

Current Page: 30/3036

        Book num                Book name                     Publisher                     Author                   etc.
-----------------------------------------------------------------------------------------------------------------------------
        11770093            bookOfeyUhKrcomputer            redlearning                   AnabelAcGItn            Not Available
        11775251            bookDVGzpEbn                    oxford learning               JacobXNbrpkB              Available
        11775557            bookIBElNgbook                  pearson&publish               JaydenmYuVlG              Available
        11776884            booksGpfkcomputer               pearsonlogman                 MasonJSqKmO             Not Available
        11776964            bookbdjnirVname                 ybm learning                  AlexanderqHKANE           Available
        11778627            computerZxRcqdnbook             oxford&publish                JaydenBpJEVw            Not Available
        11781894            bookRTgvQyuVYcomputer           oxford build                  AnastasiafxtVfJ           Available
        11788044            bookAMEyQXzname                 happy&university              AidengwdHeW             Not Available
        11788473            bookWPRtmvQ                     pearsonuniversity             AnabelXueUyv            Not Available
        11789705            bookmGEAwname                   fun&logman                    MasonMnysaU               Available


Enter <0> to select book number
Enter page number to move >
```

목록에서 도서 번호를 확인한 후 예약을 원하는 도서의 번호를 입력한다. 이어서 학생 ID를 입력한 후 예약을 할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)

Current Page: 30/3036

        Book num                Book name                     Publisher                     Author                   etc.
-----------------------------------------------------------------------------------------------------------------------------
        11770093            bookOfeyUhKrcomputer            redlearning                   AnabelAcGItn            Not Available
        11775251            bookDVGzpEbn                    oxford learning               JacobXNbrpkB              Available
        11775557            bookIBElNgbook                  pearson&publish               JaydenmYuVlG              Available
        11776884            booksGpfkcomputer               pearsonlogman                 MasonJSqKmO             Not Available
        11776964            bookbdjnirVname                 ybm learning                  AlexanderqHKANE           Available
        11778627            computerZxRcqdnbook             oxford&publish                JaydenBpJEVw            Not Available
        11781894            bookRTgvQyuVYcomputer           oxford build                  AnastasiafxtVfJ           Available
        11788044            bookAMEyQXzname                 happy&university              AidengwdHeW             Not Available
        11788473            bookWPRtmvQ                     pearsonuniversity             AnabelXueUyv            Not Available
        11789705            bookmGEAwname                   fun&logman                    MasonMnysaU               Available


Enter <0> to select book number
Enter page number to move > 0

Input book number to reserve: 11776884
Input your student ID: 201700002

Success to reserve book
 >> Book name: , Publisher: pearsonlogman, Author: MasonJSqKmO
```

만약 해당 도서번호, 예약자 ID값을 갖는 예약 정보가 이미 있다면 아래와 같이 에러 메시지가 출력된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:08:40)


Current Page: 1/3066

        Book num                Book name                     Publisher                     Author                   etc.
-------------------------------------------------------------------------------------------------------------------------------
        11101146           computerYVnXCUoubn              pearsonbuild                  DanielTIBfXeC           Not Available
        11101294           IwVlTfcomputer                  fun&future                    DanielpplFns            Not Available
        11102451           bookIHjdSnpcomputer             fun&press                     MasonVFkDPmW            Not Available
        11107528           computerUCVehtbook              pearson&learning              MichaelciCIV            Not Available
        11108024           yVZSEaBcomputer                 pearson&press                 AnabelSgaKyg              Available
        11109456           computerehRDmEBbook             happyuniversity               AnastasiaBMTvH            Available
        11111726           namedFjSLcomputer               oxfordeducation               MichaelcAmzw              Available
        11114447           scienceHDmBWAcomputer           happy&build                   NoahjaHLnR              Not Available
        11116310           bnGsLDeqcomputer                ybmbrick                      JaydenrtRTPi            Not Available
        11116878           namevVLsabcomputer              pearson&logman                JacobuMANv              Not Available


Enter <0> to select book number
Enter page number to move > 0

Input book number to reserve: 11101146
Input your student ID: 201849211
You already have reservation!
```

③ 반납

메인 화면에서 4번을 선택하면 반납을 진행할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)


Welcome to libary!

--------Select the option number(<0> to exit)---------
  1. Search for books
  2. Borrow a book
  3. Reserve a book
  4. Return a book
  5. My page
  6. Management

 >> 4
```

아래는 반납 기능을 실행하기 전에 대화식SQL 도구를 통해 확인한 값이다.

```
mysql> select * from checkedlist where checker_id = 201890056 and book_num=91670531 and return_date is null;
+----+------------+----------+---------------------+-------------+-------------+
| id | checker_id | book_num | check_date          | return_date | overdue_fee |
+----+------------+----------+---------------------+-------------+-------------+
|  6 |  201890056 | 91670531 | 2020-04-23 23:13:20 | NULL        |           0 |
+----+------------+----------+---------------------+-------------+-------------+
1 row in set (0.00 sec)
```

아래와 같이 도서 번호와 학생 ID를 입력하면 반납이 완료된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)

Enter the information to return the book (<0> to return main screen)
   > book number: 91670531
   > student id: 201890056
Success to return book
```

반납 기능을 실행한 후 return_date에 실행 당시 시간이 추가된 것을 확인할 수 있다.

```
mysql> select * from checkedlist where checker_id = 201890056 and book_num=91670531 and return_date is null;
Empty set (0.00 sec)

mysql> select * from checkedlist where checker_id = 201890056 and book_num=91670531;
+----+------------+----------+---------------------+---------------------+-------------+
| id | checker_id | book_num | check_date          | return_date         | overdue_fee |
+----+------------+----------+---------------------+---------------------+-------------+
|  6 |  201890056 | 91670531 | 2020-04-23 23:13:20 | 2020-06-12 02:19:54 |           0 |
+----+------------+----------+---------------------+---------------------+-------------+
1 row in set (0.00 sec)
```

만약 도서 번호가 존재하지 않는다면 아래와 같이 메시지가 출력된 후 종료된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)

Enter the information to return the book (<0> to return main screen)
   > book number: 10000000
   > student id: 201600001
No rows matching the book number!
```

대출 정보가 존재하지 않는다면 아래와 같이 메시지가 출력된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)

Enter the information to return the book (<0> to return main screen)
   > book number: 47449696
   > student id: 20000000
No checked information!
```

3) 마이페이지

메인 화면에서 5번을 선택하고, 학생 ID를 입력하면 마이페이지 기능을 이용할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)


Welcome to libary!

--------Select the option number(<0> to exit)---------
  1. Search for books
  2. Borrow a book
  3. Reserve a book
  4. Return a book
  5. My page
  6. Management

 >> 5
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)



------------------Mypage!-----------------
 > Please Enter your student ID: 201700002
```

① 대출 정보 조회

1번을 선택해 자신의 대출 정보를 조회할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)



--------Select the option number---------
  1. Show check list
  2. Show reserve list
  3. Check overdue fee
  4. Pay overdue fee

Please Enter <0> to go main screen
 >> 1
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:11:43)


Current Page: 1/1

Id: 87349, Book num: 55836911, Checked date: 2020-02-23 09:42:03,  Before return
Id: 132730, Book num: 33560552, Checked date: 2020-03-06 03:26:34,  Before return
Id: 158538, Book num: 34686331, Checked date: 2020-05-21 05:47:14,  Return date: 2020-08-14 05:47:14
Id: 229207, Book num: 11728391, Checked date: 2019-12-27 12:41:26,  Return date: 2020-01-25 12:41:26
Id: 230710, Book num: 57917964, Checked date: 2020-04-10 22:25:11,  Return date: 2020-07-07 22:25:11



Enter page number to move(<0> to exit) >
```

② 예약 정보 조회

2번을 선택해 자신의 예약 정보를 조회할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:26:20)



--------Select the option number---------
  1. Show check list
  2. Show reserve list
  3. Check overdue fee
  4. Pay overdue fee

Please Enter <0> to go main screen
 >> 2
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:26:20)

Current Page: 1/1

ID: 012600, Book num: 17257285, Reserved date: 2020-05-05 23:08:15
ID: 019090, Book num: 79889830, Reserved date: 2020-04-21 10:25:14
ID: 116630, Book num: 92930243, Reserved date: 2020-05-20 19:29:23
ID: 124620, Book num: 87174660, Reserved date: 2020-03-10 04:55:50
ID: 196409, Book num: 48657023, Reserved date: 2020-06-08 12:39:54
ID: 204685, Book num: 26310336, Reserved date: 2020-03-16 20:33:33
ID: 205878, Book num: 78113779, Reserved date: 2020-02-15 01:33:00
ID: 250005, Book num: 11770093, Reserved date: 2020-06-12 02:04:29
ID: 250007, Book num: 11776884, Reserved date: 2020-06-12 02:12:39



Enter page number to move(<0> to exit) >
```

③ 연체료 업데이트 및 조회

3번을 선택하면 아래와 같이 해당 시점까지의 자신의 전체 연체료를 확인할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:26:20)

--------Select the option number---------
  1. Show check list
  2. Show reserve list
  3. Check overdue fee
  4. Pay overdue fee

Please Enter <0> to go main screen
 >> 3
Current overdue fee: 41600
```

아래는 기능을 사용하기 전이고,

```
mysql> select * from checkedlist where checker_id=201700002 and return_date is NULL;
+--------+------------+----------+---------------------+-------------+-------------+
| id     | checker_id | book_num | check_date          | return_date | overdue_fee |
+--------+------------+----------+---------------------+-------------+-------------+
|  87349 |  201700002 | 55836911 | 2020-02-23 09:42:03 | NULL        |           0 |
| 132730 |  201700002 | 33560552 | 2020-03-06 03:26:34 | NULL        |           0 |
+--------+------------+----------+---------------------+-------------+-------------+
2 rows in set (0.00 sec)
```

기능을 사용한 후 연체료 정보가 업데이트된 것을 확인할 수 있다.

```
mysql> select * from checkedlist where checker_id=201700002 and return_date is NULL;
+--------+------------+----------+---------------------+-------------+-------------+
| id     | checker_id | book_num | check_date          | return_date | overdue_fee |
+--------+------------+----------+---------------------+-------------+-------------+
|  87349 |  201700002 | 55836911 | 2020-02-23 09:42:03 | NULL        |       22000 |
| 132730 |  201700002 | 33560552 | 2020-03-06 03:26:34 | NULL        |       19600 |
+--------+------------+----------+---------------------+-------------+-------------+
2 rows in set (0.00 sec)
```

④ 연체 정보 조회 및 정산

4번을 선택하면 자신의 대출 정보와 함께 연체 정보를 확인할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:28:55)

--------Select the option number---------
  1. Show check list
  2. Show reserve list
  3. Check overdue fee
  4. Pay overdue fee

Please Enter <0> to go main screen
 >> 4
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:28:55)

Current Page: 1/1

ID: 087349, Book num: 55836911, Checked date: 2020-02-23 09:42:03, Overdue fee: 22000
ID: 132730, Book num: 33560552, Checked date: 2020-03-06 03:26:34, Overdue fee: 19600
ID: 158538, Book num: 34686331, Checked date: 2020-05-21 05:47:14, Overdue fee: 0
ID: 229207, Book num: 11728391, Checked date: 2019-12-27 12:41:26, Overdue fee: 0
ID: 230710, Book num: 57917964, Checked date: 2020-04-10 22:25:11, Overdue fee: 0

Enter page number to move(<0> to exit) >
```

아래와 같이 정산하길 원하는 대출 번호를 입력하면 해당 연체료를 정산할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:28:55)
Current Page: 1/1

ID: 087349, Book num: 55836911, Checked date: 2020-02-23 09:42:03, Overdue fee: 22000
ID: 132730, Book num: 33560552, Checked date: 2020-03-06 03:26:34, Overdue fee: 19600
ID: 158538, Book num: 34686331, Checked date: 2020-05-21 05:47:14, Overdue fee: 0
ID: 229207, Book num: 11728391, Checked date: 2019-12-27 12:41:26, Overdue fee: 0
ID: 230710, Book num: 57917964, Checked date: 2020-04-10 22:25:11, Overdue fee: 0

Enter page number to move(<0> to exit) > 0
Input check ID to pay the overdue fee (<0> to exit) > 087349
Success to pay!
```

다시 4번을 선택해 정보를 조회해보면 값이 0으로 바뀐 것을 확인할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:28:55)


Current Page: 1/1

ID: 087349, Book num: 55836911, Checked date: 2020-02-23 09:42:03, Overdue fee: 0
ID: 132730, Book num: 33560552, Checked date: 2020-03-06 03:26:34, Overdue fee: 19600
ID: 158538, Book num: 34686331, Checked date: 2020-05-21 05:47:14, Overdue fee: 0
ID: 229207, Book num: 11728391, Checked date: 2019-12-27 12:41:26, Overdue fee: 0
ID: 230710, Book num: 57917964, Checked date: 2020-04-10 22:25:11, Overdue fee: 0


Enter page number to move(<0> to exit) >
```

4) 관리

메인 화면에서 6번을 선택하면 관리 메뉴를 사용할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:36:11)
Welcome to libary!

--------Select the option number(<0> to exit)---------
  1. Search for books
  2. Borrow a book
  3. Reserve a book
  4. Return a book
  5. My page
  6. Management

 >> 6
```

① 도서 정보 추가

1번을 선택하고, 도서 번호, 도서명, 출판사, 저자를 입력하면 도서 정보를 추가할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:36:11)
--------Select the function number---------
  1. Add a book to List
  2. Modify a book's infomation
  3. Delete a book from List

Please Enter <0> to go main screen
 >> 1
Input book number: 20176758
Input book name: helloworld!
Input publisher: jin-publish
Input author: jinyoung
Success to add!
```

아래는 추가한 정보를 대화식SQL도구에서 확인한 결과이다.

```
mysql> select * from booklist where book_num=20176758;
+----------+-------------+-------------+----------+------------+
| book_num | book_name   | publisher   | author   | can_borrow |
+----------+-------------+-------------+----------+------------+
| 20176758 | helloworld! | jin-publish | jinyoung |          1 |
+----------+-------------+-------------+----------+------------+
1 row in set (0.00 sec)
```

다음과 같이 응용 기능을 통해서도 확인할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 7:23:22)

Current Page: 1/1
        Book num                Book name                    Publisher                    Author                   etc.
-------------------------------------------------------------------------------------------------------------------------
        20176758            helloworld!                   jin-publish                  jinyoung                 Available

Enter page number to move(<0> to exit) >
```

② 도서 정보 수정

2번을 선택하고 수정하길 원하는 도서 번호를 입력한다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:48:34)


--------Select the function number---------
  1. Add a book to List
  2. Modify a book's infomation
  3. Delete a book from List

Please Enter <0> to go main screen
 >> 2
Input book number to modify: 20176758
```

다음과 같이 도서 번호에 해당되는 도서 정보를 확인할 수 있다. 수정하길 원하는 옵션 번호를 입력한 후 새로운 값을 입력하면 수정이 완료된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:48:34)

Book name: helloworld!, Publisher: jin-publish, Author: jinyoung,  Available

--------Select number to modify---------
  1. Book name
  2. Publisher
  3. Author

Please Enter <0> to go main screen

 >> 1
New book name: newbookname
Success to modify!
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:48:34)

Book name: newbookname, Publisher: jin-publish, Author: jinyoung,  Available

--------Select number to modify---------
  1. Book name
  2. Publisher
  3. Author

Please Enter <0> to go main screen

 >> 2
New publisher: newpublish
Success to modify!
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:48:34)

Book name: newbookname, Publisher: newpublish, Author: jinyoung,  Available

--------Select number to modify---------
  1. Book name
  2. Publisher
  3. Author

Please Enter <0> to go main screen

 >> 3
New author: jin
Success to modify!
```

아래와 같이 응용의 메인 1번 기능을 통해 수정된 값을 확인할 수 있다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:48:34)



--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 4
Input book number: 20176758
```

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:48:34)
Current Page: 1/1

        Book num                Book name                    Publisher                    Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        20176758            newbookname                     newpublish                    jin                      Available


Enter page number to move(<0> to exit) >
```

③ 도서 정보 삭제

3번을 선택한 후 삭제할 도서의 번호를 입력하면 성공적으로 삭제가 완료된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 2:48:34)

--------Select the function number---------
  1. Add a book to List
  2. Modify a book's infomation
  3. Delete a book from List

Please Enter <0> to go main screen
 >> 3
Input book number to delete: 20176758
Success to delete!
```

다음은 대화식SQL도구에서 삭제된 결과를 확인한 캡처 화면이다.

```
mysql> select * from booklist where book_num=20176758;
Empty set (0.00 sec)
```

아래와 같이 존재하지 않는 도서번호를 입력하면 에러 메시지가 출력된 후 종료된다.

```
Console
Main [Java Application] C:₩Program Files₩Java₩jdk-14₩bin₩javaw.exe  (2020. 6. 12. 오전 7:23:22)

--------Select the function number---------
  1. Add a book to List
  2. Modify a book's infomation
  3. Delete a book from List

Please Enter <0> to go main screen
 >> 3
Input book number to delete: 000000000
No row to delete!
```

8. 소스 코드 실행 관련 필요사항

- Mysql driver 파일(.jar)을 빌드 패스 라이브러리에 추가한다.
- Createtable.java를 실행해 테스트 데이터를 생성한다.(경로 수정 필요)
  -> 사용한 테스트 데이터는 폴더에 첨부 (TestData 폴더)
- MysqlLoadDataInfile.java를 실행해 데이터를 테이블에 로드한다.(경로 수정 필요)
- Transaction.java에서 url, id, pw를 변경해준다.
- Main.java의 main()함수로 실행한다.

제출 폴더

Java 파일들은 DBProject 폴더 안에 존재.

Createtable.java, MysqlLoadDateInfile.java는 LoadData 폴더(패키지) 안에 존재

Main.java, Transaction.java는 Library 폴더(패키지) 안에 존재

TestData 폴더 안 프로젝트에 사용한 테스트 데이터 파일(.csv) 존재

CreateTable.txt : 테이블 생성 구문

레포트는 word파일과 pdf파일 두 버전.

9. 사용자 매뉴얼

0)

```
Welcome to libary!

--------Select the option number(<0> to exit)---------
  1. Search for books
  2. Borrow a book
  3. Reserve a book
  4. Return a book
  5. My page
  6. Management

 >>
```

➔ **메인 화면. 원하는 옵션의 번호를 입력한다. 0 입력 시 종료**

1)

```
--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >>
```

➔ **메인 1번. 검색을 원하는 옵션 번호를 입력한다. 0 입력 시 메인 화면으로 돌아간다.**

```
--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 1
Input book name:
```

```
--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 2
Input publisher:
```

```
--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 3
Input author:
```

```
--------Search with..---------
  1. Book name
  2. Publisher
  3. Author
  4. Book number

Please Enter <0> to go main screen
 >> 4
Input book number:
```

➔ 메인 1번. 옵션 선택 시 검색 값을 입력할 수 있다.

```
Current Page: 1/3036

        Book num                Book name                    Publisher                    Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11102451          bookIHjdSnpcomputer            fun&press                    MasonVFkDPmW            Not Available
        11106757          bookogipYIjbook                ybm&education                AnabelbdaWI             Not Available
        11107381          bnaGSgFmZbook                  fun&learning                 AidenCoWzm              Not Available
        11107528          computerUCVehtbook             pearson&learning             MichaelciCIV            Not Available
        11109456          computerehRDmEBbook            happyuniversity              AnastasiaBMTvH            Available
        11115988          bookjzEsQKname                 happyfuture                  AidenCpildEA            Not Available
        11116148          bookKocHjf                     ybm&build                    JacobVsbKR                Available
        11116308          scienceIEDbgRebook             pearson&publish              AdamiIEGdG                Available
        11116539          mfPqNUnVbook                   cambridge&publish            AidenHYqZZ                Available
        11116974          MhsQirVbook                    ybm&future                   AlexanderlEdShA           Available


Enter page number to move(<0> to exit) >
```

➔ 메인 1번. 검색 값 입력 시 다음과 같은 목록을 볼 수 있고, 페이지를 입력해 원하는 페이지로 이동할 수 있다. 0 입력 시 메인 1번으로 돌아간다.

2)

```
Input book name to borrow: |
```

➔ 메인 2번. 빌리기 원하는 책 이름을 입력한다.

```
Current Page: 1/3054

        Book num                Book name                    Publisher                    Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11104620          scienceaAuEftbn                happylogman                  AnastasiafaUxIM         Not Available
        11107526          scienceEVBVzfWscience          fun&education                AnastasiaMjKxV            Available
        11108520          nameqILAkGscience              oxford education             WilliamDugAlGE          Not Available
        11114447          scienceHDmBWAcomputer          happy&build                  NoahjaHLnR              Not Available
        11116308          scienceIEDbgRebook             pearson&publish              AdamiIEGdG                Available
        11117386          scienceaiDTFfLZcomputer        funfuture                    AdamkGrnU                 Available
        11118999          scienceRQWjmSname              happy build                  MasonDowbvwU              Available
        11120824          EcLglEcscience                 ybmlogman                    AidenWtLgnm             Not Available
        11122198          bnDJQSQuscience                cambridge&future             AndrewKRAMjSQ           Not Available
        11127591          bookGcYbbbTSscience            oxford&build                 AndrewCwVfTi              Available


Enter <0> to select book number
Enter page number to move > 0
Input book number to borrow: |
```

➔ 메인 2번. 입력한 책 이름이 여러 개일 경우 목록으로 확인할 수 있다. 0을 입력해 빌릴 책의 번호를 입력할 수 있다.

```
Current Page: 1/3054

        Book num                Book name                    Publisher                    Author                    etc.
-------------------------------------------------------------------------------------------------------------------------------
        11104620          scienceaAuEftbn                happylogman                  AnastasiafaUxIM         Not Available
        11107526          scienceEVBVzfWscience          fun&education                AnastasiaMjKxV            Available
        11108520          nameqILAkGscience              oxford education             WilliamDugAlGE          Not Available
        11114447          scienceHDmBWAcomputer          happy&build                  NoahjaHLnR              Not Available
        11116308          scienceIEDbgRebook             pearson&publish              AdamiIEGdG                Available
        11117386          scienceaiDTFfLZcomputer        funfuture                    AdamkGrnU                 Available
        11118999          scienceRQWjmSname              happy build                  MasonDowbvwU              Available
        11120824          EcLglEcscience                 ybmlogman                    AidenWtLgnm             Not Available
        11122198          bnDJQSQuscience                cambridge&future             AndrewKRAMjSQ           Not Available
        11127591          bookGcYbbbTSscience            oxford&build                 AndrewCwVfTi              Available


Enter <0> to select book number
Enter page number to move > 0
Input book number to borrow: 11107526
Input your student ID: 201700002

Success to borrow book, scienceEVBVzfWscience
```

➔ 메인 2번. 책의 번호와 학생 ID를 입력하면 대출할 수 있다.

```
Input book name to borrow: scienceIEDbgRebook
Input your student ID: 201700002

Success to borrow book, scienceIEDbgRebook
```

➔ 메인 2번. 입력한 책 이름이 1개일 경우 바로 학생 ID를 입력해 대출할 수 있다.

3)

➔ 메인 3번. 메인 2번과 동일하게 사용한다.

4)

```
Enter the information to return the book (<0> to return main screen)
   > book number: 20176758
   > student id: 201700002
```

➔ 메인 4번. 반납할 책의 번호와 학생 ID를 입력해 책을 반납한다. 0을 입력하면 메인 화면으로 돌아간다.

5)

```
------------------Mypage!-----------------
 > Please Enter your student ID: |
```

➔ 메인 5번. 학생 ID를 입력해 마이 페이지에 들어갈 수 있다.

```
--------Select the option number---------
  1. Show check list
  2. Show reserve list
  3. Check overdue fee
  4. Pay overdue fee

Please Enter <0> to go main screen
 >>
```

➔ 메인 5번. 옵션을 선택해 해당 기능을 이용할 수 있다. 0을 선택하면 메인 화면으로 돌아간다.

➔ 메인 5번, 옵션 1,2는 메인 1번과 같이 각 목록을 페이지로 조회할 수 있다.

```
--------Select the option number---------
  1. Show check list
  2. Show reserve list
  3. Check overdue fee
  4. Pay overdue fee

Please Enter <0> to go main screen
 >> 3
Current overdue fee: 41600
```

➔ 메인 5번. 옵션 3을 선택하면 연체료를 확인할 수 있다.

```
Current Page: 1/1

ID: 087349, Book num: 55836911, Checked date: 2020-02-23 09:42:03, Overdue fee: 0
ID: 132730, Book num: 33560552, Checked date: 2020-03-06 03:26:34, Overdue fee: 19600
ID: 158538, Book num: 34686331, Checked date: 2020-05-21 05:47:14, Overdue fee: 0
ID: 229207, Book num: 11728391, Checked date: 2019-12-27 12:41:26, Overdue fee: 0
ID: 230710, Book num: 57917964, Checked date: 2020-04-10 22:25:11, Overdue fee: 0
ID: 250013, Book num: 11107526, Checked date: 2020-06-12 08:51:45, Overdue fee: 0
ID: 250014, Book num: 11116308, Checked date: 2020-06-12 08:52:55, Overdue fee: 0


Enter page number to move(<0> to exit) > 0
Input check ID to pay the overdue fee (<0> to exit) > 132730
```

➔ 메인 5번. 옵션 4를 선택하면 목록을 확인할 수 있고 원하는 ID를 입력하여 정산한다.

6)

```
--------Select the function number---------
  1. Add a book to List
  2. Modify a book's infomation
  3. Delete a book from List

Please Enter <0> to go main screen
 >>
```

➔ 메인 6번. 옵션 번호를 선택해 도서 목록 관리 기능을 사용할 수 있다. 0을 선택하면 메인 화면으로 돌아간다.

```
--------Select the function number---------
  1. Add a book to List
  2. Modify a book's infomation
  3. Delete a book from List

Please Enter <0> to go main screen
 >> 1
Input book number: 99999999
Input book name: newname
Input publisher: newpublisher
Input author: newauthor
Success to add!
```

➔ **메인 6번. 옵션 1번을 선택해 정보를 입력하여 도서를 추가할 수 있다.**

```
--------Select the function number---------
  1. Add a book to List
  2. Modify a book's infomation
  3. Delete a book from List

Please Enter <0> to go main screen
 >> 2
Input book number to modify: 20176758
```

➔ **메인 6번. 옵션 2번을 선택해 수정하길 원하는 도서 번호를 입력한다.**

```
Book name: helloworld!, Publisher: jin-publish, Author: jinyoung,  Available

--------Select number to modify---------
  1. Book name
  2. Publisher
  3. Author

Please Enter <0> to go main screen

 >>
```

➔ **메인 6번. 옵션 2번(이어서) 정보를 확인하고 수정하길 원하는 옵션을 선택해 수정할 수 있다.**

```
--------Select the function number---------
  1. Add a book to List
  2. Modify a book's infomation
  3. Delete a book from List

Please Enter <0> to go main screen
 >> 3
Input book number to delete:
```

➔ **메인 6번. 옵션 3번을 선택, 도서 번호를 입력한 후 삭제할 수 있다.**