

## A. HLL-Report

학과 : IT공학전공

### 1. 프로그램 설명 :: 온라인 서점 `BookStore`

온라인 서점 `BookStore`에서 회원가입/책 주문/회원정보 확인,수정,탈퇴 등의 기능을 제공하는 프로그램을 구현하였다.

테이블은 다음과 같이 회원, 도서, 주문내역, 출판사 총 4개의 테이블을 구상하였다.

분홍색 셀은 각 테이블의 주 키를 의미하고, 초록색 셀은 외래 키를 의미한다.

회원	ID	PWD	이름	가입날짜	도서	ID	책 제목	저자	출판사 ID	가격
hi	0	이효원	2020-11-02		1	OpenGL로 배우는 3차원 컴퓨터 그래픽스	주우석		1	32000
you	123	김민지	2012-04-02		2	자료구조와 함께 배우는 알고리즘 입문	보요 시바타		2	19800
are	1	신경원	2018-12-03		3	자료구조와 C	이석호		3	30000
so	12	한채연	2019-09-13		4	안드로이드 앱 프로그래밍	정재곤		2	36000
beauti	1234	김서희	2021-10-12		5	HTML5 웹 프로그래밍 입문	임순범		4	24000
ful	34	차수정	2021-07-22		6	경영정보시스템	Kenneth		5	39000
and	45	김도영	2022-04-03		7	회계원리	손성규		6	40000
look	573	김경민	2021-05-16		8	컴퓨터와 IT기술의 이해	최윤철		4	27000
happy	24	정현정	2022-11-30		9	Warming-Up C Programming	전정아		8	25000
now	9	송혜진	2021-12-15		10	이것이 리눅스다	우재남		7	28800

주문내역	ID	회원ID	책 ID	총 가격	주문일	출판사	ID	출판사이름	주소	전화번호
1	hi	3	30000	2022-03-10 10:16:21		1	한빛아카데미	서울 서대문구 연희로2길 62	02-336-7122	
2	hi	7	40000	2022-03-10 10:17:30		2	이지스퍼블리싱	서울 마포구 전지로 109	02-325-1722	
3	hi	10	28800	2022-03-10 10:20:34		3	정익사	서울 서대문구 동일로 39가길 47	02-396-4077	
4	look	6	39000	2021-11-20 16:42:08		4	생능	경기도 파주시 광인사길 143	031-955-0761	
5	beauti	4	36000	2021-12-28 22:36:12		5	시그마프레스	서울 영등포구 영평로22길 21	02-323-4845	
6	you	7	40000	2018-09-04 14:17:11		6	법문사	경기도 파주시 회동길 37-29	031-955-6500	
7	now	10	28800	2022-04-25 18:19:20		7	한빛미디어	서울 서대문구 연희로2길 62	02-325-0384	
8	and	5	24000	2022-05-01 23:21:18		8	연두매디션	경기도 고양시 일산동구 동국로 32	031-932-9896	
9	you	1	32000	2015-06-06 17:28:37		9	오랜미디어	서울 성동구 아차산로 92	0505-522-2024	
10	hi	2	19800	2022-04-14 18:30:27		10	길벗	서울 마포구 월드컵로 10길 56	02-332-0931	

스키마 제작 과정은 다음과 같다.

book 테이블 생성

```
id int not null auto_increment,
title varchar(30),
author varchar(20),
pid int,
price int,
primary key(id),
foreign key(pid) references publisher(id) on update cascade on delete cascade
);
```

book 인스턴스 10개 삽입

```
insert into book (title, author, pid, price) values
('OpenGL로 배우는 3차원 컴퓨터 그래픽스', '주우석', 1, 32000);
insert into book (title, author, pid, price) values
('자료구조와 함께 배우는 알고리즘 입문', '보요 시바타', 2, 19800);
insert into book (title, author, pid, price) values
('자료구조와 C', '이석호', 3, 30000);
insert into book (title, author, pid, price) values
('안드로이드 앱 프로그래밍', '정재곤', 2, 36000);
insert into book (title, author, pid, price) values
('HTML5 웹 프로그래밍 입문', '임순범', 4, 24000);
insert into book (title, author, pid, price) values
('경영정보시스템', 'Kenneth', 5, 39000);
insert into book (title, author, pid, price) values
('회계원리', '손성규', 6, 40000);
insert into book (title, author, pid, price) values
('컴퓨터와 IT기술의 이해', '최윤철', 4, 27000);
insert into book (title, author, pid, price) values
('Warming-Up C Programming', '전정아', 8, 25000);
insert into book (title, author, pid, price) values
('이것이 리눅스다', '우재남', 7, 28800);
```

먼저 book 테이블은 `BookStore`에 존재하는 책 정보를 저장하는 테이블로, 책 번호(id), 책 제목(title), 저자(author), pid(출판사 번호), 가격(price)로 구성되어있고, id가 주키이다.

### member 테이블 생성

```
create table member
(
  id varchar(20),
  pwd varchar(20),
  name varchar(20),
  mdate date,
  primary key(id)
);
```

### member 인스턴스 10개 삽입

```
insert into member(id, pwd, name, mdate) values
('hi', '0', '이효원', '2020-11-02');
insert into member(id, pwd, name, mdate) values
('you', '0123', '김민지', '2012-04-02');
insert into member(id, pwd, name, mdate) values
('are', '1', '신경원', '2018-12-03');
insert into member(id, pwd, name, mdate) values
('so', '12', '함재현', '2019-09-13');
insert into member(id, pwd, name, mdate) values
('beauti', '1234', '김서희', '2021-10-12');
insert into member(id, pwd, name, mdate) values
('ful', '34', '차수경', '2021-07-22');
insert into member(id, pwd, name, mdate) values
('and', '45', '김도영', '2022-04-03');
insert into member(id, pwd, name, mdate) values
('look', '573', '김경민', '2021-05-16');
insert into member(id, pwd, name, mdate) values
('happy', '024', '정현정', '2022-11-30');
insert into member(id, pwd, name, mdate) values
('now', '9', '송해진', '2021-12-15');
```

member 테이블은 'BookStore'의 회원 정보를 저장하는 테이블로, 회원 아이디(id), 회원 비밀번호(pwd), 회원 이름(name), 가입날짜(mdate)로 구성되어있고, id가 주키이다.

### orders 테이블 생성

```
create table orders
(
  id int not null auto_increment,
  mid varchar(20),
  bid int,
  totalPrice int,
  odate timestamp,
  primary key(id),
  foreign key(mid) references member(id) on update cascade on delete cascade,
  foreign key(bid) references book(id) on update cascade on delete cascade
);
```

### orders 인스턴스 10개 삽입

```
insert into orders(mid, bid, totalPrice, odate) values
('hi', 3, 30000, '2022-03-10 10:16:21');
insert into orders(mid, bid, totalPrice, odate) values
('hi', 7, 40000, '2022-03-10 10:17:30');
insert into orders(mid, bid, totalPrice, odate) values
('hi', 10, 28800, '2022-03-10 10:20:34');
insert into orders(mid, bid, totalPrice, odate) values
('look', 6, 39000, '2021-11-20 16:42:08');
insert into orders(mid, bid, totalPrice, odate) values
('beauti', 4, 36000, '2021-12-28 22:36:12');
insert into orders(mid, bid, totalPrice, odate) values
('you', 7, 40000, '2018-09-04 14:17:11');
insert into orders(mid, bid, totalPrice, odate) values
('now', 10, 28800, '2022-04-25 18:19:20');
insert into orders(mid, bid, totalPrice, odate) values
('and', 5, 24000, '2022-05-01 23:21:18');
insert into orders(mid, bid, totalPrice, odate) values
('you', 1, 32000, '2015-06-06 17:28:37');
insert into orders(mid, bid, totalPrice, odate) values
('hi', 2, 19800, '2022-04-14 18:30:27');
```

orders 테이블은 'BookStore'의 주문 내역을 저장하는 테이블로, 주문 번호(id), 회원 아이디(mid), 책 번호(bid), 주문금액(totalPrice), 주문날짜(odate)로 구성되어있고, id가 주키이다. 그리고 mid, bid는 각각 member 테이블 id와 book 테이블의 id의 외래키이다.

#### publisher 테이블 생성

```
create table publisher(  
id int not null auto_increment,  
name varchar(20),  
addr varchar(20),  
tel varchar(20),  
primary key(id)  
);
```

#### publisher 인스턴스 10개 삽입

```
insert into publisher(name, addr, tel) values  
( '한빛아카데미', '서울 서대문구 연희로2길 62', '02-336-7122');  
insert into publisher(name, addr, tel) values  
( '이지스퍼블리싱', '서울 마포구 잔다리로 109', '02-325-1722');  
insert into publisher(name, addr, tel) values  
( '정익사', '서울 서대문구 통일로 39가길 47', '02-396-4077');  
insert into publisher(name, addr, tel) values  
( '상능', '경기도 파주시 광인사길 143', '031-955-0761');  
insert into publisher(name, addr, tel) values  
( '시그마프레스', '서울 영등포구 양평로22길 21', '02-323-4845');  
insert into publisher(name, addr, tel) values  
( '법문사', '경기도 파주시 회동길 37-29', '031-955-6500');  
insert into publisher(name, addr, tel) values  
( '한빛미디어', '서울 서대문구 연희로2길 62', '02-325-0384');  
insert into publisher(name, addr, tel) values  
( '연두에디션', '경기도 고양시 일산동구 통곡로 32', '031-932-9896');  
insert into publisher(name, addr, tel) values  
( '오렌지미디어', '서울 성동구 아차산로 92', '0505-522-2024');  
insert into publisher(name, addr, tel) values  
( '길벗', '서울 마포구 월드컵로 10길 56', '02-332-0931');
```

publisher 테이블은 출판사 정보를 저장하는 테이블로, 출판사 등록 번호(id), 출판사 이름(name), 출판사 주소(addr), 출판사 전화번호(tel)로 구성되어있고, id가 주키이다.

## 2. 메소드 설명

- showMenu() : 메뉴를 보여주고 선택한 메뉴 번호를 반환하는 함수
- order() : 주문서를 보여주고 주문을 처리하는 함수
- memInfo() : 회원 정보와 과거 주문 내역을 확인하는 함수
- editInfo() : 비밀번호를 수정하는 함수
- signUp() : 회원가입을 진행하는 함수
- resign() : 회원탈퇴를 진행하는 함수

## 3. HLL ESQl 구현 코드 상세 설명 - Python

#### showMenu() - 메뉴 출력 및 반환

```
def showMenu():  
    print("-----")  
    print("메뉴 선택 페이지입니다.")  
    print("\n")  
    print("1. 책 주문")  
    print("2. 회원정보 확인")  
    print("3. 비밀번호 수정")  
    print("4. 회원가입")  
    print("5. 회원탈퇴")  
    print("6. 종료")  
    print("\n")  
  
    menu = input("메뉴 번호를 입력해주세요: ")  
  
    return menu
```

## order() - 주문 처리 함수

```
# 주문하기
def order():
    print("-----")
    print("주문 페이지입니다.\n")

# 데베 연결
db = pymysql.connect(host='localhost', user='root', password='try1234',
db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

# 커서 이용해서 sql문 실행
cur = db.cursor()
sql = "select * from book"
cur.execute(sql)

rows = cur.fetchall()
books = pd.DataFrame(rows)
print(tabulate(books, headers='keys', tablefmt='psql', showindex=False))

userID = input("사용자 ID를 입력해주세요: ")
bookID = input("책 ID를 입력해주세요: ")

sql = "select * from member where id=%s"
cur.execute(sql, userID)
rows = cur.fetchall()

sql2 = "select * from book where id=%s"
cur.execute(sql2, bookID)
rows2 = cur.fetchall()

if (rows and rows2):

    bookPrice = books.iloc[int(bookID)-1, 4]

    sql = "insert into orders (mid, bid, totalPrice, odate) values (%s, %s, %s, %s)"
    val = (userID, bookID, bookPrice, datetime.now().strftime("%Y-%m-%d %H:%M:%S"))
    cur.execute(sql, val)
    rows = cur.fetchall()

    sql2 = "select * from orders order by odate desc limit 1"
    cur.execute(sql2)
    rows = cur.fetchall()
    db.close()

    now_order = pd.DataFrame(rows)

    print("\n주문이 완료되었습니다.")

    print(tabulate(now_order, headers='keys', tablefmt='psql', showindex=False))
    print("\n")

else:
    print("\n존재하지 않는 사용자 ID이거나 책 ID를 잘못 입력하였습니다.\n")
```

DB 연결

커서 객체 생성 후, 작성한  
**select문 실행**, book table  
의 모든 값을 검색

fetchall() 이용해서 rows  
변수에 저장.

pandas 객체의  
DataFrame() 함수 이용해  
서 데이터값을 데이터프레  
임 형태로 출력.

tabulate 함수 이용해서 테  
이블 형태로 출력.

존재하는 회원 아이디와 책 아이디인  
지 확인하는 sql문

book 테이블에서 해당 셀 값 검색

**insert문 실행**, orders 테  
이블에 주문 내역 추가

select문 이용, 가장 최근  
주문 내역을 가져와서 화  
면에 출력

먼저 DB를 연결하고, 커서 객체를 생성한다. select문으로 book table에 있는 모든 값들을 검색하여 테이블 형태로 출력한다. 사용자가 입력한 userID와 bookID가 유효한지 확인하고, 유효하면 insert문으로 orders 테이블에 userID, bookID, totalPrice, 주문시각의 값들을 가진 레코드를 삽입한다. 그리고 주문이 완료되었다는 문구를 출력한다. 입력값이 유효하지 않은 경우에는 존재하지 않는 사용자 ID이거나 책 ID를 잘못 입력하였다는 문구를 출력한다.

### memInfo() - 회원정보 확인 함수

```
# 회원 정보 확인하기
def memInfo():
    print("-----")
    print("회원 정보 확인 페이지입니다.\n")

    userID = input("사용자 ID를 입력해주세요: ")

    # 데베 연결
    db = pymysql.connect(host='localhost', user='root', password='try1234',
        db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

    # 커서 생성
    cur = db.cursor()

    sql = "select * from member where id=%s"
    cur.execute(sql, userID)
    rows = cur.fetchall()

    if (rows):

        # 회원 정보
        sql = "select * from member where id = %s"
        cur.execute(sql, userID)
        rows = cur.fetchall()

        memInfo = pd.DataFrame(rows)
        print("\n회원정보입니다.")
        print(tabulate(memInfo, headers='keys', tablefmt='psql', showindex=False))

        # 과거 주문 내역
        sql2 = "select orders.id, book.title, book.author, book.price, publisher.name, publisher.addr, publisher.tel, orders.odate from member, book, publisher, orders where member.id=%s and orders.mid=member.id and orders.bid=book.id and book.pid=publisher.id"
        cur.execute(sql2, userID)
        rows = cur.fetchall()
        db.close()

        order_list = pd.DataFrame(rows)
        print("\n과거 주문 내역입니다.")
        print(tabulate(order_list, headers='keys', tablefmt='psql', showindex=False))
        print("\n")

    else:
        print("\n일치하는 ID가 없습니다.")
        print("회원가입을 먼저 해주세요.\n")
```

3중 조인 연산

코드가 잘려서 아래  
텍스트로 다시 삽입

사용자 ID가 존재하는 회원 ID이면 회원 정보와 과거 주문 내역을 출력한다.

회원 정보를 출력할 때는 select문을 이용하여 member 테이블에서 id값이 입력한 userID값과 같은 레코드를 검색한다.

### 3중 조인 연산

```
sql2 = "select orders.id, book.title, book.author, book.price, publisher.name, publisher.addr, publisher.tel, orders.odate from member, book, publisher, orders where member.id=%s and orders.mid=member.id and orders.bid=book.id and book.pid=publisher.id";
```

과거 주문 내역을 출력할 때는 select문을 이용하여 member, orders, publisher, book 총 4개의 테이블들을 모두 조인하여 member id값이 userID인 레코드에서 주문 번호, 책 제목, 책 저자, 책 가격, 출판사 이름, 출판사 전화번호, 주문 시각등의 값들을 검색하여 가져오도록 하였다.



### editInfo() - 비밀번호 수정 함수

```
# 비밀번호 수정하기
def editInfo():
    print("-----")
    print("비밀번호 수정 페이지입니다.\n")

    # 데베 연결
    db = pymysql.connect(host='localhost', user='root', password='try1234',
                        db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

    # 커서 생성
    cur = db.cursor()

    userID = input("사용자 ID를 입력해주세요: ")
    userPWD = input("기존 비밀번호를 입력해주세요: ")
    newPWD = input("새 비밀번호를 입력해주세요: ")

    sql = "select * from member where id=%s and pwd=%s"
    val = (userID, userPWD)
    cur.execute(sql, val)
    rows = cur.fetchall()

    if (rows):
        sql2 = "update member set pwd=%s where id=%s"
        val2 = (newPWD, userID)
        cur.execute(sql2, val2)

        print("\n수정이 완료되었습니다.")
        sql3 = "select * from member where id=%s"
        cur.execute(sql3, userID)
        rows = cur.fetchall()
        db.close()

        userInfo = pd.DataFrame(rows)
        print(tabulate(userInfo, headers='keys', tablefmt='psql', showindex=False))
        print("\n")
    else:
        print("\n존재하지 않는 사용자 ID이거나 비밀번호를 잘못 입력하였습니다.\n")
```

**update문 실행**, member 테이블에서 id값이 userID인 레코드의 pwd값을 newPWD로 수정

userID값과 userPWD값이 일치하는 레코드가 member 테이블에 존재할 경우에만 수정을 진행한다. id가 userID인 레코드의 pwd값을 newPWD로 수정하는 update문을 작성하고 실행한다. 이후 수정이 완료되었다는 문구를 출력하고, 수정이 완료된 것을 확인할 수 있도록 id값이 userID인 레코드를 member 테이블에서 검색하는 select문을 작성하고 실행하여 테이블 형태로 보여준다.

### signUp() - 회원 가입 함수

```
# 회원 가입하기
def signUp():
    print("-----")
    print("회원가입 페이지입니다.\n")

    # 데베 연결
    db = pymysql.connect(host='localhost', user='root', password='try1234',
                        db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

    # 커서 생성
    cur = db.cursor()

    newID = input("사용할 ID를 입력해주세요: ")
    newPWD = input("사용할 비밀번호를 입력해주세요: ")
    userName = input("이름을 입력해주세요: ")

    sql = "select * from member where id=%s"
    cur.execute(sql, newID)
    rows = cur.fetchall()

    if (rows):
        print("\n이미 존재하는 ID입니다.\n")
    else:
        sql2 = "insert into member (id, pwd, name, mdate) values (%s, %s, %s, %s)"
        val = (newID, newPWD, userName, datetime.now().strftime("%Y-%m-%d"))
        cur.execute(sql2, val)
        rows = cur.fetchall()

        print("\n가입이 완료되었습니다.\n")
```

**insert문 실행**, member 테이블에 새로운 회원 정보 삽입

member 테이블에 입력받은 id와 같은 id가 존재하지 않을 경우에만 insert문을 실행하여 member 테이블에 입력받은 id, pwd, name 값을 가진 레코드를 삽입한다.

#### resign() - 회원 탈퇴 함수

```
# 회원 탈퇴하기
def resign():
    print("-----")
    print("회원 탈퇴 페이지입니다.\n")

    # 데베 연결
    db = pymysql.connect(host='localhost', user='root', password='try1234',
                        db='bookstore', charset='utf8', autoccommit=True, cursorclass=pymysql.cursors.DictCursor)

    # 커서 생성
    cur = db.cursor()

    userID = input("사용자 ID를 입력해주세요: ")
    userPWD = input("비밀번호를 입력해주세요: ")

    sql = "select * from member where id=%s and pwd=%s"
    val = (userID, userPWD)
    cur.execute(sql, val)
    rows = cur.fetchall()

    if (rows):
        sql2 = "delete from member where id=%s"
        cur.execute(sql2, userID)

        print("\n회원 탈퇴가 완료되었습니다.\n")

    else:
        print("\n존재하지 않는 사용자 ID이거나 비밀번호를 잘못 입력하였습니다.\n")
```

delete문 실행, member  
테이블에서 해당 id값을 가  
진 회원정보 삭제

member 테이블에 userID와 userPWD가 모두 일치하는 경우에만 delete문을 실행한다.  
member테이블에서 id값이 userID인 레코드를 삭제하는 delete문을 작성하고 실행한 후 회  
원 탈퇴가 완료되었다는 문구를 출력한다.

#### 메인 코드

```
print("\nBookStore에 오신 것을 환영합니다!\n")

while(True):

    menu = showMenu()

    if menu == "1":
        order()
    elif menu == "2":
        memInfo()
    elif menu == "3":
        editInfo()
    elif menu == "4":
        signUp()
    elif menu == "5":
        resign()
    elif menu == "6":
        print("안녕히 가십시오.")
        break
    else:
        print("1~6 사이의 메뉴 번호를 입력해주세요")
```

무한 루프로 메뉴 화면을 출력해주고 입력한 메뉴 번호에 따라 함수를 실행하는 프로세스를 반복한다. 사용자가 6을 입력했을 때는 break문을 통해 프로그램이 종료되고, 1~6 이외의 값  
을 입력했을 때는 1~6 사이의 메뉴 번호를 입력해달라는 경고문을 출력한다.

#### 4. 실행 및 결과 화면

##### 1) signup() - 회원 가입

1-1) 중복되지 않는 아이디로 가입할 경우 -> 정상 가입 완료

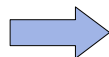
```
BookStore에 오신 것을 환영합니다!  
-----  
메뉴 선택 페이지입니다.  
  
1.책 주문  
2.회원정보 확인  
3.비밀번호 수정  
4.회원가입  
5.회원탈퇴  
6.종료  
  
메뉴 번호를 입력해주세요: 4  
-----  
회원가입 페이지입니다.  
  
사용할 ID를 입력해주세요: win  
사용할 비밀번호를 입력해주세요: 2015399  
이름을 입력해주세요: 이효원  
  
가입이 완료되었습니다.
```

1-2) 중복된 아이디로 가입할 경우 -> 가입 실패

```
메뉴 번호를 입력해주세요: 4  
-----  
회원가입 페이지입니다.  
  
사용할 ID를 입력해주세요: hi  
사용할 비밀번호를 입력해주세요: 342  
이름을 입력해주세요: 이효원  
  
이미 존재하는 ID입니다.  
-----  
메뉴 선택 페이지입니다.
```

1-3) 데이터베이스 member 테이블 변화

	id	pwd	name	mdate
▶	and	45	김도영	2022-04-03
	are	1	신경원	2018-12-03
	beauti	1234	김서희	2021-10-12
	ful	34	차수정	2021-07-22
	happy	024	정현정	2022-11-30
	hi	0	이효원	2020-11-02
	look	573	김경민	2021-05-16
	now	9	송혜진	2021-12-15
	so	12	한채연	2019-09-13
	you	0123	김민지	2012-04-02
●	NULL	NULL	NULL	NULL



삽입

	id	pwd	name	mdate
▶	and	45	김도영	2022-04-03
	are	1	신경원	2018-12-03
	beauti	1234	김서희	2021-10-12
	ful	34	차수정	2021-07-22
	happy	024	정현정	2022-11-30
	hi	2015399	이효원	2020-11-02
	look	573	김경민	2021-05-16
	now	9	송혜진	2021-12-15
	so	12	한채연	2019-09-13
	win	2015399	이효원	2022-05-22
	you	0123	김민지	2012-04-02
●	NULL	NULL	NULL	NULL



## 2) order() - 책 주문

### 2-1) 존재하는 회원 ID와 올바른 책 ID를 입력할 경우 -> 주문 완료

메뉴 번호를 입력해주세요: 1

주문 페이지입니다.

id	title	author	pid	price
1	OpenGL로 배우는 3차원 컴퓨터 그래픽스	주우석	1	32000
2	자료구조와 함께 배우는 알고리즘 입문	보요 시바타	2	19800
3	자료구조와 C	이석호	3	30000
4	안드로이드 앱 프로그래밍	정재곤	2	36000
5	HTML5 웹 프로그래밍 입문	임순범	4	24000
6	경영정보시스템	Kenneth	5	39000
7	회계원리	손성규	6	40000
8	컴퓨터와 IT기술의 이해	최윤철	4	27000
9	Warming-Up C Programming	천정아	8	25000
10	이것이 리눅스다	우재남	7	28800

사용자 ID를 입력해주세요: win  
책 ID를 입력해주세요: 8

주문이 완료되었습니다.

id	mid	bid	totalPrice	odate
14	win	8	27000	2022-05-22 02:16:56

### 2-2) 존재하지 않는 회원 ID나 올바른 책 ID가 아닐 경우 -> 주문 실패

메뉴 번호를 입력해주세요: 1

주문 페이지입니다.

id	title	author	pid	price
1	OpenGL로 배우는 3차원 컴퓨터 그래픽스	주우석	1	32000
2	자료구조와 함께 배우는 알고리즘 입문	보요 시바타	2	19800
3	자료구조와 C	이석호	3	30000
4	안드로이드 앱 프로그래밍	정재곤	2	36000
5	HTML5 웹 프로그래밍 입문	임순범	4	24000
6	경영정보시스템	Kenneth	5	39000
7	회계원리	손성규	6	40000
8	컴퓨터와 IT기술의 이해	최윤철	4	27000
9	Warming-Up C Programming	천정아	8	25000
10	이것이 리눅스다	우재남	7	28800

사용자 ID를 입력해주세요: whattimeisit  
책 ID를 입력해주세요: 3

존재하지 않는 사용자 ID이거나 책 ID를 잘못 입력하였습니다.

### 2-3) 데이터베이스 orders 테이블 변화

id	mid	bid	totalPrice	odate
1	hi	3	30000	2022-03-10 10:16:21
2	hi	7	40000	2022-03-10 10:17:30
3	hi	10	28800	2022-03-10 10:20:34
4	look	6	39000	2021-11-20 16:42:08
5	beauti	4	36000	2021-12-28 22:36:12
6	you	7	40000	2018-09-04 14:17:11
7	now	10	28800	2022-04-25 18:19:20
8	and	5	24000	2022-05-01 23:21:18
9	you	1	32000	2015-06-06 17:28:37
10	hi	2	19800	2022-04-14 18:30:27
NULL	NULL	NULL	NULL	NULL

삽입

id	mid	bid	totalPrice	odate
1	hi	3	30000	2022-03-10 10:16:21
2	hi	7	40000	2022-03-10 10:17:30
3	hi	10	28800	2022-03-10 10:20:34
4	look	6	39000	2021-11-20 16:42:08
5	beauti	4	36000	2021-12-28 22:36:12
6	you	7	40000	2018-09-04 14:17:11
7	now	10	28800	2022-04-25 18:19:20
8	and	5	24000	2022-05-01 23:21:18
9	you	1	32000	2015-06-06 17:28:37
10	hi	2	19800	2022-04-14 18:30:27
14	win	8	27000	2022-05-22 02:16:56
NULL	NULL	NULL	NULL	NULL

### 3) memInfo() - 회원정보 확인

3-1) 존재하는 회원 아이디일 경우 -> 회원 정보 및 과거 주문 내역 출력

```

메뉴 번호를 입력해주세요: 2
회원 정보 확인 페이지입니다.
사용자 ID를 입력해주세요: win
회원정보입니다.
+-----+
| id | pwd | name | mdate |
+-----+
| win | 2015399 | 이효원 | 2022-05-22 |
+-----+

과거 주문 내역입니다.
+-----+
| id | title | author | price | name | addr | tel | odate |
+-----+
| 14 | 컴퓨터와 IT기술의 이해 | 최윤철 | 27000 | 생능 | 경기도 파주시 광인사길 143 | 031-955-0761 | 2022-05-22 02:16:56 |
+-----+

```

3-2) 존재하는 회원 아이디가 아닐 경우 -> 경고문 출력

메뉴 번호를 입력해주세요: 2

---

회원 정보 확인 페이지입니다.

사용자 ID를 입력해주세요: depth

일치하는 ID가 없습니다.

회원가입을 먼저 해주세요.

#### 4) editInfo() - 비밀번호 수정

4-1) 올바른 사용자 ID와 비밀번호를 입력한 경우 -> 수정 완료

```
메뉴 번호를 입력해주세요 : 3
-----
비밀번호 수정 페이지입니다 .

사용자 ID를 입력해주세요 : win
기존 비밀번호를 입력해주세요 : 2015399
새 비밀번호를 입력해주세요 : 0522

수정이 완료되었습니다 .
-----+-----+-----+-----+
| id      | pwd      | name      | mdate     |
|-----+-----+-----+-----+
| win     | 0522    | 이효원   | 2022-05-22 |
|-----+-----+-----+-----+
```

4-2) 올바른 사용자 ID와 비밀번호가 아닌 경우 -> 경고문 출력

```
메뉴 번호를 입력해주세요: 3
-----
비밀번호 수정 페이지입니다.

사용자 ID를 입력해주세요: win
기존 비밀번호를 입력해주세요: 33
새 비밀번호를 입력해주세요: 02

존재하지 않는 사용자 ID이거나 비밀번호를 잘못 입력하였습니다.
```

#### 4-3) 데이터베이스 member 레코드 변화

	id	pwd	name	mdate
▶	and	45	김도영	2022-04-03
	are	1	신경원	2018-12-03
	beauti	1234	김서희	2021-10-12
	ful	34	차수정	2021-07-22
	happy	024	정현정	2022-11-30
	hi	2015399	이효원	2020-11-02
	look	573	김경민	2021-05-16
	now	9	송혜진	2021-12-15
	so	12	한채연	2019-09-13
	win	2015399	이효원	2022-05-22
	you	0123	김민지	2012-04-02
●	NULL	NULL	NULL	NULL

수정

	id	pwd	name	mdate
▶	and	45	김도영	2022-04-03
	are	1	신경원	2018-12-03
	beauti	1234	김서희	2021-10-12
	ful	34	차수정	2021-07-22
	happy	024	정현정	2022-11-30
	hi	2015399	이효원	2020-11-02
	look	573	김경민	2021-05-16
	now	9	송혜진	2021-12-15
	so	12	한채연	2019-09-13
	win	0522	이효원	2022-05-22
	you	0123	김민지	2012-04-02
●	NULL	NULL	NULL	NULL

#### 5) resign() - 회원 탈퇴

##### 5-1) 올바른 사용자 ID와 비밀번호일 경우 -> 탈퇴 완료

```

메뉴 번호를 입력해주세요: 5
-----
회원 탈퇴 페이지입니다.

사용자 ID를 입력해주세요: win
비밀번호를 입력해주세요: 0522

회원 탈퇴가 완료되었습니다.
    
```

##### 5-2) 올바른 사용자 ID와 비밀번호가 아닌 경우 -> 경고문 출력

```

메뉴 번호를 입력해주세요: 5
-----
회원 탈퇴 페이지입니다.

사용자 ID를 입력해주세요: hi
비밀번호를 입력해주세요: 234

존재하지 않는 사용자 ID이거나 비밀번호를 잘못 입력하였습니다.
    
```

#### 5-3) 데이터베이스 member 테이블 변화

	id	pwd	name	mdate
▶	and	45	김도영	2022-04-03
	are	1	신경원	2018-12-03
	beauti	1234	김서희	2021-10-12
	ful	34	차수정	2021-07-22
	happy	024	정현정	2022-11-30
	hi	2015399	이효원	2020-11-02
	look	573	김경민	2021-05-16
	now	9	송혜진	2021-12-15
	so	12	한채연	2019-09-13
	win	0522	이효원	2022-05-22
	you	0123	김민지	2012-04-02
●	NULL	NULL	NULL	NULL

삭제

	id	pwd	name	mdate
▶	and	45	김도영	2022-04-03
	are	1	신경원	2018-12-03
	beauti	1234	김서희	2021-10-12
	ful	34	차수정	2021-07-22
	happy	024	정현정	2022-11-30
	hi	2015399	이효원	2020-11-02
	look	573	김경민	2021-05-16
	now	9	송혜진	2021-12-15
	so	12	한채연	2019-09-13
	you	0123	김민지	2012-04-02
●	NULL	NULL	NULL	NULL

[\*\*\* Cross Check 점수 신청: 고급언어-->웹 검증: 1점 신청 \*\*\*]

## 1) Update

### HLL에서 회원 비밀번호 수정

메뉴 번호를 입력해주세요: 3

비밀번호 수정 페이지입니다.

사용자 ID를 입력해주세요: win


기존 비밀번호를 입력해주세요: 2015399

새 비밀번호를 입력해주세요: 0522

수정이 완료되었습니다.

id	pwd	name	mdate
win	0522	이효원	2022-05-22

### Web에서 변경된 회원 비밀번호 확인

 Home Order My Page Sign Up

마이 페이지

아이디:

회원 정보 확인

주문 내역 확인

회원 정보

아이디	비밀번호	이름	가입날짜
win	0522	이효원	2022-05-22

## 2) Insert

### HLL에서 책 주문

메뉴 번호를 입력해주세요: 1

주문 페이지입니다.

id	title	author	pid	price
1	OpenGL로 배우는 3차원 컴퓨터 그래픽스	주우석	1	32000
2	자료구조와 함께 배우는 알고리즘 입문	보요 시바타	2	19800
3	자료구조와 C	이석훈	3	30000
4	안드로이드 앱 프로그래밍	정재곤	2	36000
5	HTML5 웹 프로그래밍 입문	임순범	4	24000
6	경영정보시스템	Kenneth	5	39000
7	회계원리	손성규	6	40000
8	컴퓨터와 IT기술의 이해	최윤철	4	27000
9	Warming-Up C Programming	천정아	8	25000
10	이것이 리눅스다	우재남	7	28800

사용자 ID를 입력해주세요: win

책 ID를 입력해주세요: 8

주문이 완료되었습니다.

id	mid	bid	totalPrice	odate
14	win	8	27000	2022-05-22 02:16:56

### Web에서 주문 내역 확인

BookStore

Home Order My Page Sign Up

마이 페이지

아이디:

win 검색 후

회원 정보 확인

주문 내역 확인

과거 주문 내역

주문번호	책 제목	책 저자	책 가격	출판사 이름	출판사 주소	출판사 전화번호	주문 날짜
14	컴퓨터와 IT기술의 이해 최윤철	27000	생능	경기도 파주시 광인사길 143 031-955-0761			2022-05-22 02:16:56

### 3) Delete

#### HLL에서 회원 탈퇴

메뉴 번호를 입력해주세요: 5  
-----  
회원 탈퇴 페이지입니다.  
사용자 ID를 입력해주세요: win  
비밀 번호를 입력해주세요: 0522  
회원 탈퇴가 완료되었습니다.

#### Web에서 해당 회원 정보 검색 -> 정보 출력 X

BookStore

Home Order My Page Sign Up

마이 페이지

아이디:

회원 정보 확인

주문 내역 확인

회원 정보

아이디 비밀번호 이름 가입날짜



## 5. 소스 코드

### bookstore.py

```
import pymysql
import pandas as pd
from tabulate import tabulate
from datetime import datetime

# 메뉴 출력하기
def showMenu():
    print("-----")
    print("메뉴 선택 페이지입니다.")
    print("\n")
    print("1.책 주문")
    print("2.회원정보 확인")
    print("3.비밀번호 수정")
    print("4.회원가입")
    print("5.회원탈퇴")
    print("6.종료")
    print("\n")

    menu = input("메뉴 번호를 입력해주세요: ")

    return menu

# 주문하기
def order():
    print("-----")
    print("주문 페이지입니다.\n")

    # 데베 연결
    db = pymysql.connect(host='localhost', user='root', password='try1234',
                        db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

    # 커서 이용해서 sql문 실행
    cur = db.cursor()
    sql = "select * from book"
    cur.execute(sql)

    rows = cur.fetchall()
    books = pd.DataFrame(rows)
    print(tabulate(books, headers='keys', tablefmt='psql', showindex=False))

    userID = input("사용자 ID를 입력해주세요: ")
    bookID = input("책 ID를 입력해주세요: ")

    # 사용자 아이디 확인
    sql = "select * from member where id=%s"
```

```

cur.execute(sql, userID)
rows = cur.fetchall()

# 책 아이디 확인
sql2 = "select * from book where id=%s"
cur.execute(sql2, bookID)
rows2 = cur.fetchall()

if (rows and rows2):

    bookPrice = books.iloc[int(bookID)-1, 4]

    # 주문 내역 삽입
    sql = "insert into orders (mid, bid, totalPrice, odate) values (%s, %s, %s, %s)"
    val = (userID, bookID, bookPrice, datetime.now().strftime("%Y-%m-%d %H:%M:%S"))

    cur.execute(sql, val)
    rows = cur.fetchall()

    # 주문 내역 출력
    sql2 = "select * from orders order by odate desc limit 1"
    cur.execute(sql2)
    rows = cur.fetchall()
    db.close()

    now_order = pd.DataFrame(rows)

    print("\n주문이 완료되었습니다.")

    print(tabulate(now_order, headers='keys', tablefmt='psql', showindex=False))
    print("\n")

else:
    print("\n존재하지 않는 사용자 ID이거나 책 ID를 잘못 입력하였습니다.\n")

# 회원정보 확인하기
def memInfo():
    print("-----")
    print("회원 정보 확인 페이지입니다.\n")

    userID = input("사용자 ID를 입력해주세요: ")

# 데베 연결
db = pymysql.connect(host='localhost', user='root', password='try1234',
db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

# 커서 생성
cur = db.cursor()

```

```

sql = "select * from member where id=%s"
cur.execute(sql, userID)
rows = cur.fetchall()

if (rows):

    # 회원 정보 출력
    sql = "select * from member where id = %s"
    cur.execute(sql, userID)
    rows = cur.fetchall()

    memInfo = pd.DataFrame(rows)
    print("\n회원정보입니다.")
    print(tabulate(memInfo, headers='keys', tablefmt='psql', showindex=False))

    # 과거 주문 내역 출력
    sql2 = "select orders.id, book.title, book.author, book.price, publisher.name, publisher.addr,
publisher.tel, orders.odate from member, book, publisher, orders where member.id=%s and
orders.mid=member.id and orders.bid=book.id and book.pid=publisher.id";
    cur.execute(sql2, userID)
    rows = cur.fetchall()
    db.close()

    order_list = pd.DataFrame(rows)
    print("\n과거 주문 내역입니다.")
    print(tabulate(order_list, headers='keys', tablefmt='psql', showindex=False))
    print("\n")

else:
    print("\n일치하는 ID가 없습니다.")
    print("회원가입을 먼저 해주세요.\n")

# 비밀번호 수정하기
def editInfo():
    print("-----")
    print("비밀번호 수정 페이지입니다.\n")

    # 데베 연결
    db = pymysql.connect(host='localhost', user='root', password='try1234',
db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

    # 커서 생성
    cur = db.cursor()

    userID = input("사용자 ID를 입력해주세요: ")
    userPWD = input("기존 비밀번호를 입력해주세요: ")
    newPWD = input("새 비밀번호를 입력해주세요: ")

    # 아이디, 비밀번호 확인

```

```

sql = "select * from member where id=%s and pwd=%s"
val = (userID, userPWD)
cur.execute(sql, val)
rows = cur.fetchall()

if (rows):
    # 비밀번호 수정
    sql2 = "update member set pwd=%s where id=%s"
    val2 = (newPWD, userID)
    cur.execute(sql2, val2)

    print("\n수정이 완료되었습니다.")
    sql3 = "select * from member where id=%s"
    cur.execute(sql3, userID)
    rows = cur.fetchall()
    db.close()

    userInfo = pd.DataFrame(rows)
    print(tabulate(userInfo, headers='keys', tablefmt='psql', showindex=False))
    print("\n")

else:
    print("\n존재하지 않는 사용자 ID이거나 비밀번호를 잘못 입력하였습니다.\n")

# 회원 가입하기
def signUp():
    print("-----")
    print("회원가입 페이지입니다.\n")

    # 데베 연결
    db = pymysql.connect(host='localhost', user='root', password='try1234',
    db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

    # 커서 생성
    cur = db.cursor()

    newID = input("사용할 ID를 입력해주세요: ")
    newPWD = input("사용할 비밀번호를 입력해주세요: ")
    userName = input("이름을 입력해주세요: ")

    # 중복 아이디 유무 판단
    sql = "select * from member where id=%s"
    cur.execute(sql, newID)
    rows = cur.fetchall()

    if (rows):
        print("\n이미 존재하는 ID입니다.\n")
    else:
        # 회원 정보 삽입

```

```

        sql2 = "insert into member (id, pwd, name, mdate) values (%s, %s, %s, %s)"
        val = (newID, newPWD, userName, datetime.now().strftime("%Y-%m-%d"))
        cur.execute(sql2, val)
        rows = cur.fetchall()

        print("\n가입이 완료되었습니다.\n")

# 회원 탈퇴하기
def resign():
    print("-----")
    print("회원 탈퇴 페이지입니다.\n")

    # 데베 연결
    db = pymysql.connect(host='localhost', user='root', password='try1234',
        db='bookstore', charset='utf8', autocommit=True, cursorclass=pymysql.cursors.DictCursor)

    # 커서 생성
    cur = db.cursor()

    userID = input("사용자 ID를 입력해주세요: ")
    userPWD = input("비밀번호를 입력해주세요: ")

    # 아이디, 비밀번호 확인
    sql = "select * from member where id=%s and pwd=%s"
    val = (userID, userPWD)
    cur.execute(sql, val)
    rows = cur.fetchall()

    if (rows):
        # 회원 정보 삭제
        sql2 = "delete from member where id=%s"
        cur.execute(sql2, userID)

        print("\n회원 탈퇴가 완료되었습니다.\n")

    else:
        print("\n존재하지 않는 사용자 ID이거나 비밀번호를 잘못 입력하였습니다.\n")

print("\nBookStore에 오신 것을 환영합니다!\n")

while(True):

    menu = showMenu()

    if menu == "1":
        order()
    elif menu == "2":
        memInfo()

```



```
elif menu == "3":
    editInfo()
elif menu == "4":
    signUp()
elif menu == "5":
    resign()
elif menu == "6":
    print("안녕히 가십시오.")
    break
else:
    print("1~6 사이의 메뉴 번호를 입력해주세요")
```

## 6. Progress Report(고급 언어)

**5/16**

서점 관련 프로그램을 구현하기로 결정했고, 오후 3시 학교에서 수업이 끝나고 남은 시간에 데이터베이스 테이블 구상을 완료하였다.

**5/20**

고급 언어는 웹 언어에서 구현해놓았던 것을 형식만 바꾸는 것 때문인지, 웹 언어를 구현할 때보다 빠르게 구현할 수 있었다. 1학년 이후로 파이썬을 이용하여 프로그래밍해본 것이 처음이라 1학년 기억도 나고 좋았다. 점심 2~3시에 시작해서 저녁 8시까지 픽업픽업 함수를 작성하다가 모든 함수 구현을 마쳤고, 약간의 예외 처리로 마무리하였다.

**5/22**

고급 언어 보고서를 작성하는 중이다. 지금 시각은 새벽 3시이다. 거의 다 작성한 것 같다. 아침에 다시 검토하고 아마 자정에 다 정리해서 낼 것 같다.

### **\*\* 요구 점수 \*\***

#### **1. 최종보고서 - 요구점수 : 6점**

최종보고서에 포함되어야 하는 내용을 모두 작성하였다. 프로그램 및 스키마 설명, 구현 코드 설명, 실행 및 결과 화면 캡처 및 데이터베이스 결과 변화 화면 캡처, Progress Report 등을 내용에 포함하여 사실에 기반하여 구체적으로 작성하였다.

#### **2. 고급/웹 언어 구현 성숙도 - 요구점수 : 10점**

모든 메뉴에 잘못된 입력을 줄 경우, 다른 화면을 출력해내도록 예외 처리를 구현하였다. 예를 들어, 주문하기 메뉴에서 사용자 ID와 책 ID를 잘못 입력할 경우 주문 내역 처리가 되지 않도록 하였고, 회원 가입하기 메뉴에서 중복된 사용자 ID로 가입을 시도하려고 한다면 경고문이 출력되도록 하였다. 회원 정보 확인하기에서도 입력한 사용자 ID가 존재하지 않는 경우에 경고문이 출력되도록 하였고, 회원 탈퇴하기, 비밀번호 수정하기 메뉴에서 사용자 ID와 비밀번호 ID가 일치하는 회원 정보가 없는 경우 또한 삭제와 수정 모두 처리되지 않고 경고문이 출력되도록 하였다.

#### **3. 테이블/조인 기능 존재 여부 - 요구점수 : 2점**

1번 프로그램 설명 단계에서 스키마 사진을 첨부하여 3개 이상의 속성을 가진 총 4개의 테이블을 구상하였음을 보였고, 각 테이블에 총 10개의 인스턴스를 삽입하는 화면을 캡처하여 제시하였다.

회원 정보 확인하기 메뉴에서 과거 주문 내역을 출력하는 과정에서 member, book, publisher, orders 4개의 테이블의 삼중 조인 연산을 진행하였다.

#### **4. HLL-Web ESQL Cross Check 여부 - 요구점수 : 2점**

보고서 내에 [\*\*\* Cross Check 점수 신청: 고급언어-->웹 검증:1점 신청 \*\*\*] 이라는 label로 점수를 신청하였고, 고급 언어에서의 update/delete/insert에 대한 연산 결과를 웹 상에서 확인한 화면을 모두 첨부하였다.