**[스프링 프레임워크]**

**기말고사 과제 : 개발 보고서**

컴퓨터과학과

2016301019

김진우

**목차 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. 프로젝트 개요
2. 개발 구조
3. 개발 내용
4. 테이블 구조
5. 향후 활동 /후기

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1-1. 소개

이 시스템은 도서관 관리 시스템입니다. 다중 사용자가 스프링 mvc로 관리되어지는 도서 목록을 검색하고 대여하며 반납하는 기능을 이용할 수 있도록 개발하는 것이 이 과제의 목표입니다. 사용자는 정보 입력과 수정을 할 수 있으며 신규 가입과 로그인 서비스로 도서 서비스를 이용할 수 있습니다. 또한 추가적인 기능으로 중고 서적 등록 서비스를 구현하였으며 사용자는 자신의 중고 서적을 등록하여 다른 사용자가 사용할 수 있는 서비스를 이용할 수 있습니다. 다른 추가 서비스로는 도서 예약하기가 있으며 사용자는 도서가 이미 대여 중일 때 도서 예약하기 기능으로, 다른 도서 대여자가 도서를 반납하면 자동으로 도서가 대여되는 서비스입니다. 또 다른 추가적인 기능으로 관리자 모드 사용이 있으며, 이 기능은 관리자 비밀번호를 입력해서 로그인에 성공하면, 도서를 수정하고 삭제하고 등록 할 수 있는 도서관 관리자의 서비스 입니다. 데이터베이스와 연동하여 동작하며 사용자 정보는 member, 도서 정보는 book, 도서 대여와 예약 서비스는 orderb table에 저장되어집니다. 데이터 베이스 이름은 spring5fs이며 Toad for MySQL을 사용하였습니다.

1-2. 요구사항

저는 개발 요구사항을 충족하기 위해서 Spring MVC버전으로 이 도서관 관리 시스템을 개발하였으며, 도서정보와 사용자 정보, 도서 대여와 예약 등의 정보는 MySQL로 저장되며 Spring JDBC를 사용하였습니다. 각 정보들은 Controller에 의해서 해당하는 페이지를 jsp파일로 실행되어 질 수 있게 프로그램 하였습니다 (Controller\_book ,Controller\_member,Controller\_order). 또한 Spring MVC로 프로그램이 동작하게 하기 위해서 이클립스 내에서 여러 설정을 진행하였습니다.

1-3. 개발 일정

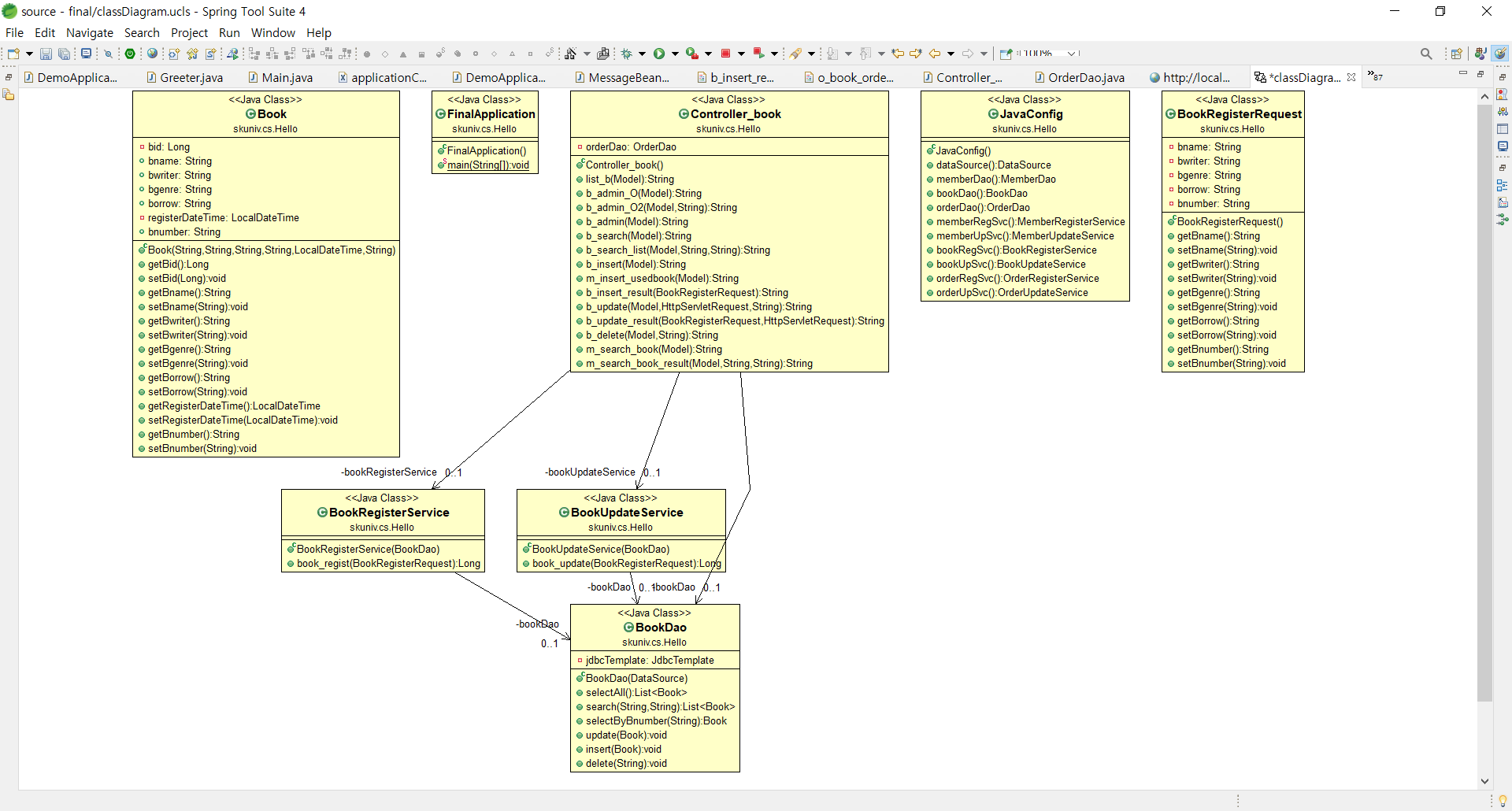
|  |  |  |
| --- | --- | --- |
| **작업명** | **일정** | **체크** |
| 프로그램 구상 | 05.25 |  |
| 개발환경 준비, 세팅 | 05.26 |  |
| 주요 기능 공부, 복습 | 05.27~05.28 |  |
| 도서 서비스 위한 코드 작성 | 05.29~05.30 |  |
| 사용자 서비스와 관리자 서비스를 위한 코드 작성 | 05.31~06.01 |  |
| 도서대여, 중고 기증하기, 도서 예약을 위한 코드 작성 | 06.02~06.03 |  |
| 코트 수정 및 테스트 | 06.04 |  |
| 보고서 작성 | 06.05~06.08 |  |

1-4. 개발 환경

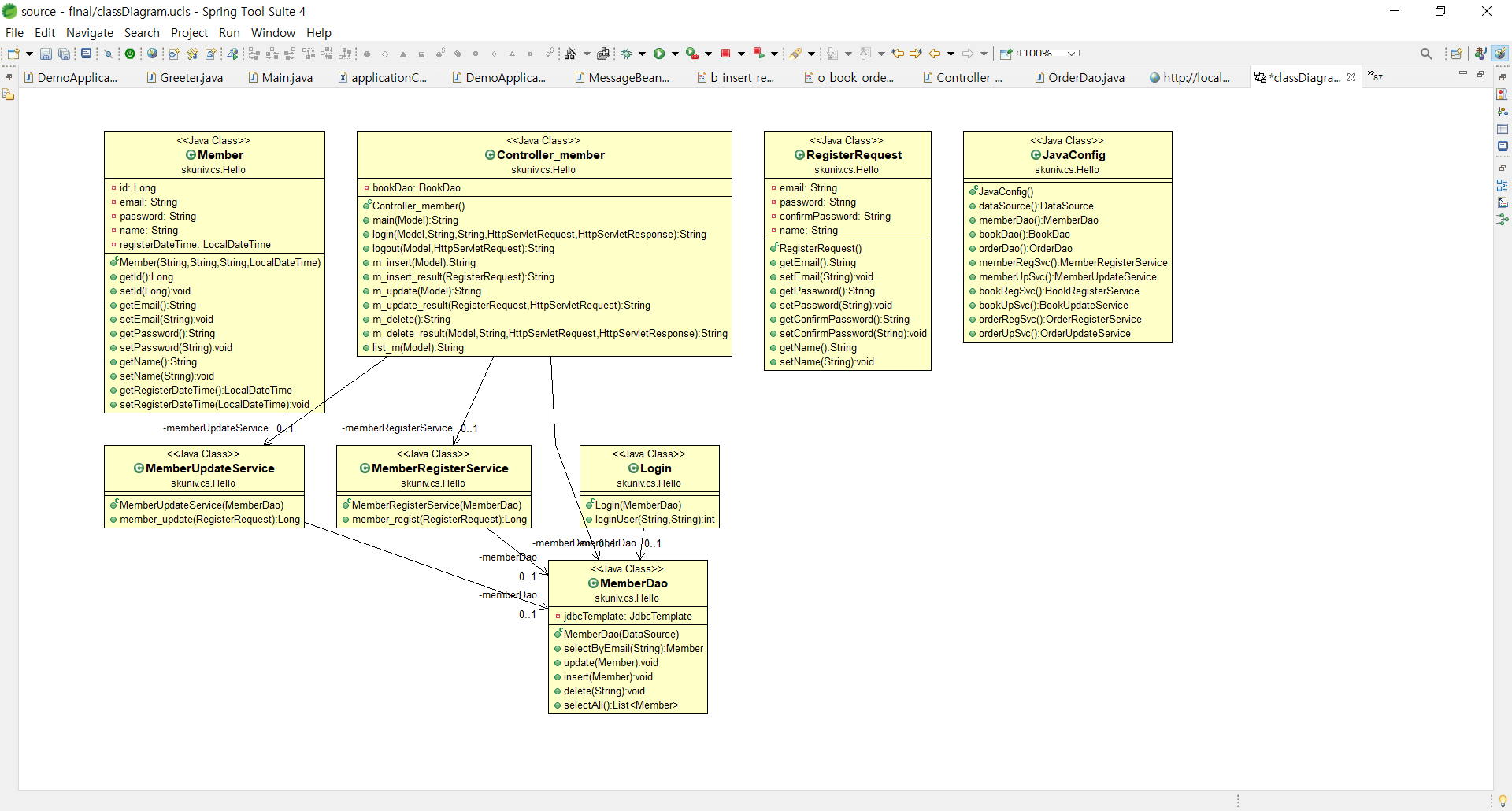
|  |  |
| --- | --- |
| **개발 언어** | **Java, xml** |
| **개발 환경** | **SpringToolSuite4,**  **Toad for MySQL** |
| **라이브러리** | **Maven, Spring MVC, JSP** |
| **참고서적** | **초보 웹 개발자를 위한 스프링5 프로그래밍 입문** |
| **문서 편집** | **파워포인트** |
| **word** |

**2. 개발 구조\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

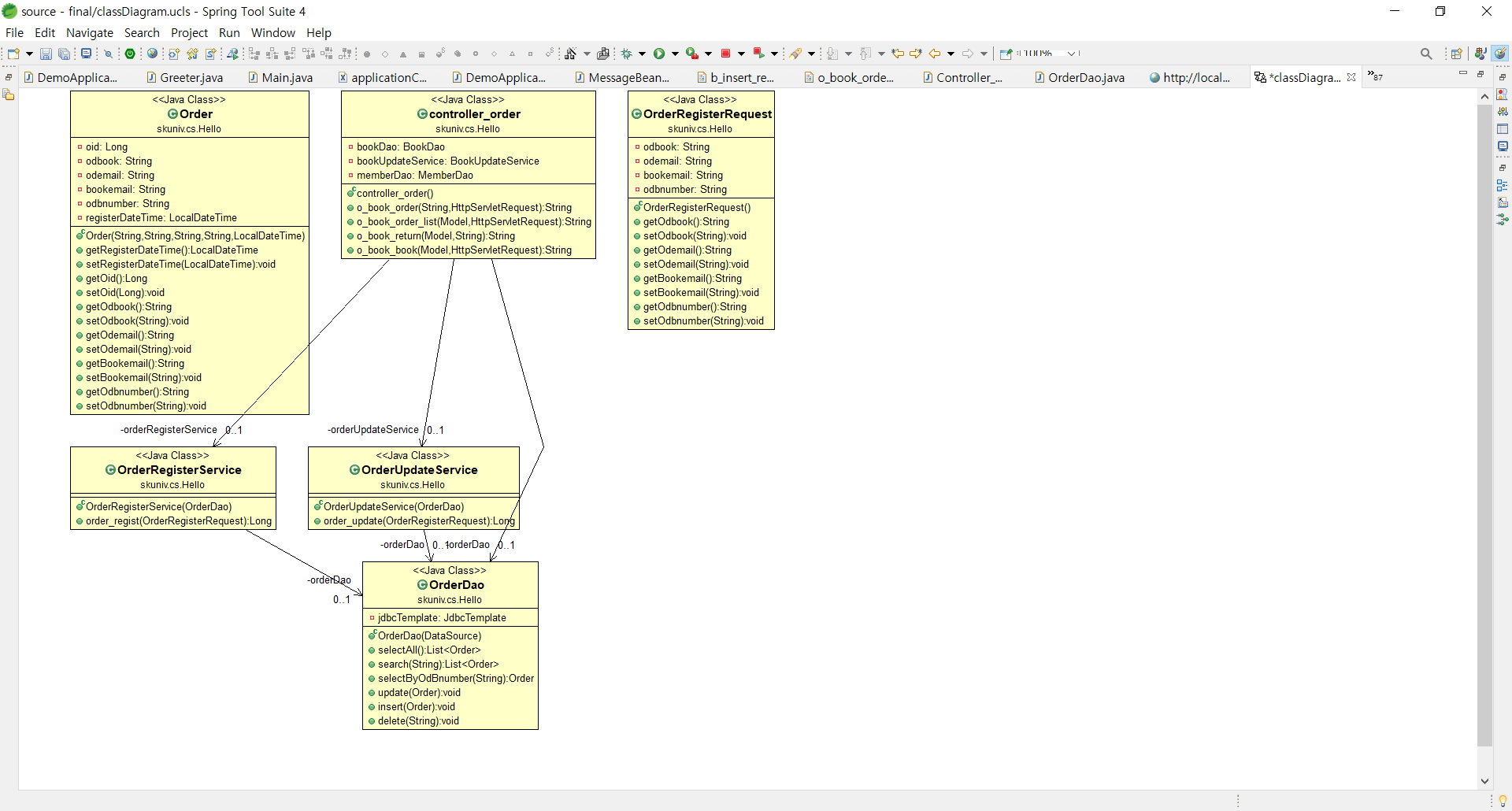
클래스 다이어그램 <Book>



클래스 다이어그램 <Member>



클래스 다이어그램 <Order>



**3. 개발 내용\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

3-1. Book.java 클래스

책 정보를 저장하는 클래스입니다. 책 클래스는 bid, bnumber(책번호), bname(책 제목), bwriter(책 작가), bgenre(책 장르), borrow(대여가능/대여불가능),registerDateTime(책 등록 시간)을 객체로 가집니다. 그리고 Book클래스는 getters 와 setters를 각 객체의 메서드로 가집니다.

3-2 BookDao.java 클래스

책 정보를 보관하고 데이터를 수정, 삭제 검색, 등록하기 위한 클래스입니다. JdbTemplate를 사용하여 외부의 MySQL에접근 할 수 있습니다.

**public** List<Book> selectAll()

해당 메서드를 통해서 외부 DB에 있는 모든 Book 테이블정보를 불러올 수 있습니다.

**public** List<Book> search(String op , String search)

이 메서드는 특정 쿼리를 불러올 수 있는 메서드입니다.

책을 검색할 때 사용하며,

"select \* from BOOK where "+op+" LIKE ?" ,,,,,, , ("%" + search + "%")

해당 쿼리문을 통해서 op(옵션을 선택하는 변수, 책의 장르로 검색할지 , 이름으로 검색할지 등 사용자가 선택하는 옵션 값) , search(사용자가 입력한 검색어)를 통해서 사용자가 검색하려고 하는 목록을 보여줄 때 사용됩니다.

**public** Book selectByBnumber(String bnumber)

이 메서드는 bnumber(책 번호 = 고유번호 )를 통해서 책 정보를 읽어오는 메서드입니다.

그 외에 update, insert, delete 메서드를 통해서 수정, 등록, 삭제를 진행할 수 있습니다.

3-3 BookRegisterRequest.java 클래스

이 클래스는 book의 요소들을 set, get할 수 있는 메서드들을 가지고 있는 클래스입니다. set함수들을 통해서 book 요소를 설정하거나 get함수들을 사용해서 book의 요소를 return받을 수 있습니다.

3-4 BookRegisterService.java 클래스

이 클래스는 도서를 등록하기 위한 클래스입니다.

**public** Long book\_regist(BookRegisterRequest breq)

해당 메서드가 도서를 등록하기 위한 regist 메서드이며, 내부에서 중요한 코드를 살펴보자면,

**while** (**true**) {

**int** rand = (**int**) Math.*floor*(Math.*random*() \* 99999999);

String randnum = Integer.*toString*(rand);

Book book = bookDao.selectByBnumber(randnum);

해당 코드가 있습니다. 이는 booknumber(책번호)를 랜덤으로 생성하고 해당 책번호를 가지는 책 정보가 없다면 새로운 책을 등록하기 위한 코드입니다. 책번호는 바코드 번호처럼 랜덤번호, 고유 값으로 설정하기 위해서 해당 코드를 작성하였습니다. 만약에 중복된 책번호가 생성되었을 경우 while문이 반복적으로 동작해서 0~9까지 8자리의 랜덤 책번호를 생성하게 됩니다. 이 후에 BookDao의 isert문을 사용해서 책 정보를 등록하게 됩니다.

3-5 BookUpdateService.java 클래스

이 클래스는 BookRegsterService와 아주 유사하며 BookDao의 update문을 사용해서 책 정보를 수정하는 클래스입니다.

3-6 Controller\_book.java 클래스

이 클래스는 MVC로 프로그램이 동작하기 위한 Controller중에서 책 데이터에 관련된 작업을 수행하는 클래스입니다.

메서드를 중요한 것만 살펴보자면,

@GetMapping(value = "/b\_list")

**public** String list\_b(Model model) {

모든 책 정보를 읽어와서 list\_b.jsp에 나타내기 위한 메서드입니다.

@GetMapping(value = "b\_admin\_O")

**public** String b\_admin\_O(Model model) {

@RequestMapping(value = "/b\_admin\_O2", method = RequestMethod.***POST***)

**public** String b\_admin\_O2(Model model, @RequestParam(value = "admin\_pwd", required = **false**) String admin\_pwd) {

책을 수정, 삭제, 등록할 수 있는 관리자 서비스를 이용하기 위한 것 입니다. 관리자 서비스를 이용하기 위해서는 비밀번호가 필요한데, admin\_pwd를 객체로 받아서 비밀번호의 유효성을 판단하는 코드입니다.

@RequestMapping(value = "/b\_update", method = RequestMethod.***POST***)

**public** String b\_update(Model model,HttpServletRequest request

, @RequestParam(value = "check", required = **false**) String check) {

@PostMapping("b\_update\_result")

**public** String b\_update\_result(BookRegisterRequest bregReq ,HttpServletRequest request) {

책을 업데이트 하는 메서드입니다.

중요한 점은, 사용자가 정보를 수정할 때 수정을 원하지 않는 부분은 아무런 입력을 하지 않게 됩니다. 그러면 update문을 실행할 때 빈 값이 data로 들어가기 때문에, 입력되지 않은 값에 해당하는 객체는 미리 저장되어진 세션에 있는 값을 불러와서 값이 그대로 보존될 수 있도록 해줍니다.

그 외에 책을 검색하고 검색결과를 보여주고, 책을 삭제하는 등의 대부분 관리자 서비스를 이용할 수 있는 클래스입니다.

3-7 Controller\_member.java 클래스

이 클래스는 MVC로 프로그램이 동작하기 위한 Controller중에서 멤버 데이터에 관련된 작업을 수행하는 클래스입니다.

메서드를 중요한 것만 살펴보자면,

@RequestMapping(value = "/login", method = RequestMethod.***POST***)

**public** String login(Model model, @RequestParam(value = "strEmail", required = **false**) String strEmail,

@RequestParam(value = "strPwd", required = **false**) String strPwd,

HttpServletRequest request, HttpServletResponse response) {

위 메서드는 사용자가 로그인 하기 위한 메서드입니다. 만약에 아이디와 비밀번호 값이 정상적으로 입력되었다면, 세션을 생성하여 로그인 속성에 1을 대입하고 정상적으로 입력되지 않았다면 로그인 속성에 0을 대입합니다. 이후에 main.jsp (도서 관리 서비스의 기본 화면)에서 로그인 속성이 1이라면 사용자가 로그인 되어진 것이므로 로그인 후에 이용할 수 있는 서비스를 나타내고, 로그인 속성이 0이라면 사용자가 로그인 하지 않은 것이므로, 회원가입이나 관리자 서비스만 이용할 수 있도록 프로그램 하였습니다. 책 컨트롤러와 마찬가지로 사용자 정보를 update할 때 기존 값을 세션에 저장한 후에 입력되지 않은 값을 기존 값으로 입력될 수 있게 프로그램 하였습니다.

3-8 Controller\_order.java 클래스

이 클래스는 MVC로 프로그램이 동작하기 위한 Controller중에서 대여 /예약 데이터에 관련된 작업을 수행하는 클래스입니다. 메서드를 중요한 것만 살펴보자면,

@RequestMapping(value = "/o\_book\_order", method = RequestMethod.***POST***)

**public** String o\_book\_order(@RequestParam(value = "check", required = **false**) String check, HttpServletRequest request) {

이 메서드는 책을 대여하기 위한 메서드입니다. Book 테이블에 있는 borrow(대여 가능한지 여부)를 조회하여, 해당 책을 대여할 수 있다면 borrow\_book 세션이 1값을 가지고, 만약에 대여할 수 없다면 borrow\_book 세션이 0값을 가지도록 프로그램 하였습니다. 이 후에 jsp화면에서 저장된 세션 값이 1이라면 사용자는 책을 대여할 수 있고, 세션 값이 0이라면 사용자는 책을 예약하거나 대여를 취소할 수 있습니다.

사용자가 책을 대여할 경우 book 테이블의 borrow 객체는 “대여불가능”으로 변경되어집니다. 이후에 만약 예약을 진행하는 것을 원한다면

@GetMapping("/o\_book\_book") **public** String o\_book\_book(Model model,

HttpServletRequest request)

위 메서드를 통해서 예약을 진행합니다. 하지만 책이 예약마저 등록되어 있다면 사용자는 아무런 서비스를 이용할 수 없도록 프로그램 하였습니다.

@RequestMapping(value = "/o\_book\_return", method = RequestMethod.***POST***)

**public** String o\_book\_return(Model model, @RequestParam(value = "check", required = **false**) String check)

위 메서드는 책을 반납하는 메서드입니다. 위 메서드는 사용자가 자신이 대여한 책 목록 중에서 하나를 선택하면 check메서드에 선택된 책의 책번호를 전달받습니다. 이 후에 해당 책의 대여(order)를 삭제하고, 해당 책의 borrow(대여 가능도)를 대여가능으로 변경하고, 만약에 이전의 order에 예약자가 있을 경우 새로운 대여(order)를 생성합니다. (예약자가 대여자로 바뀌는 것)

3-9JavaConfig.java 클래스

이 클래스는 데이터베이스를 접근하여 테이블 값을 관리하는 Dao와 연동할 수 있도록 해주는 클래스입니다.

3-10 Login.java 클래스

이 클래스는 사용자가 로그인 할 수 있도록 해주는 클래스입니다. 아이디와 비밀번호가 모두 맞을 경우 1을 반환합니다.

3-11 Member.java 클래스

사용자 정보를 저장하는 클래스입니다. 사용자 클래스는 id, email(사용자 이메일), password(사용자 비밀번호), name(사용자 이름),registerDateTime(사용자 가입 시간)을 객체로 가집니다. 그리고 member 클래스는 getters 와 setters를 각 객체의 메서드로 가집니다.

3-12 MemberDao.java 클래스

사용자 정보를 보관하고 데이터를 수정, 삭제 검색, 등록하기 위한 클래스입니다. JdbTemplate를 사용하여 외부의 MySQL에접근 할 수 있습니다.

3-13 MemberRegisterService.java 클래스

이 클래스는 사용자를 등록하는 클래스입니다. selectByEmail이라는 MemberDao의 메서드를 사용해서 해당하는 이메일의 사용자가 등록되어 있지 않으면, 새로운 사용자를 등록할 수 있도록 해주는 서비스입니다.

3-14 MemberUpdateService.java 클래스

이 클래스는 사용자를 수정하는 클래스입니다. selectByEmail을 사용해서 해당 사용자를 읽어오고, 해당 사용자가 존재한다면 수정을 진행하는 클래스입니다.

3-15 Order.java 클래스

주문(대여, 예약) 정보를 저장하는 클래스입니다. 오더 클래스는 oid, odbook(주문 책 이름), odemail(주문자 이메일), bookemail(예약자 이메일), odnumber(주문 책 번호) registerDateTime(대여 or 예약시간)을 객체로 가집니다.

그리고 order 클래스는 getters 와 setters를 각 객체의 메서드로 가집니다.

3-16 OrderDao.java 클래스

주문(대여, 예약) 정보를 보관하고 데이터를 수정, 삭제 검색, 등록하기 위한 클래스입니다. JdbTemplate를 사용하여 외부의 MySQL에접근 할 수 있습니다.

3-17 OrderRegisterRequest.java 클래스

이 클래스는 order의 요소들을 set, get할 수 있는 메서드들을 가지고 있는 클래스입니다. set함수들을 통해서 order 요소를 설정하거나 get함수들을 사용해서 order의 요소를 return받을 수 있습니다.

3-18 OrderRegisterService.java 클래스

이 클래스는 주문 정보를 등록하는 클래스입니다. selectByOdBnumber이라는 OrderDao의 메서드를 사용해서 해당하는 책번호의 주문이 없다면, 새로운 대여 or 예약을 진행하는 서비스입니다.

3-19 OrderUpdateService.java 클래스

이 클래스는 주문 정보를 변경하는 클래스입니다. 만약에 예약자가 새로 등록되어야 한다면 이미 대여되어진 책의 주문에 예약자 정보를 표시해야 하므로 그 때 이 클래스를 사용합니다. 예약자 이메일인 bookEmail에 현재 사용자의 이메일 값을 넣어주는 클래스입니다.

3-20 RegisterRequest.java 클래스

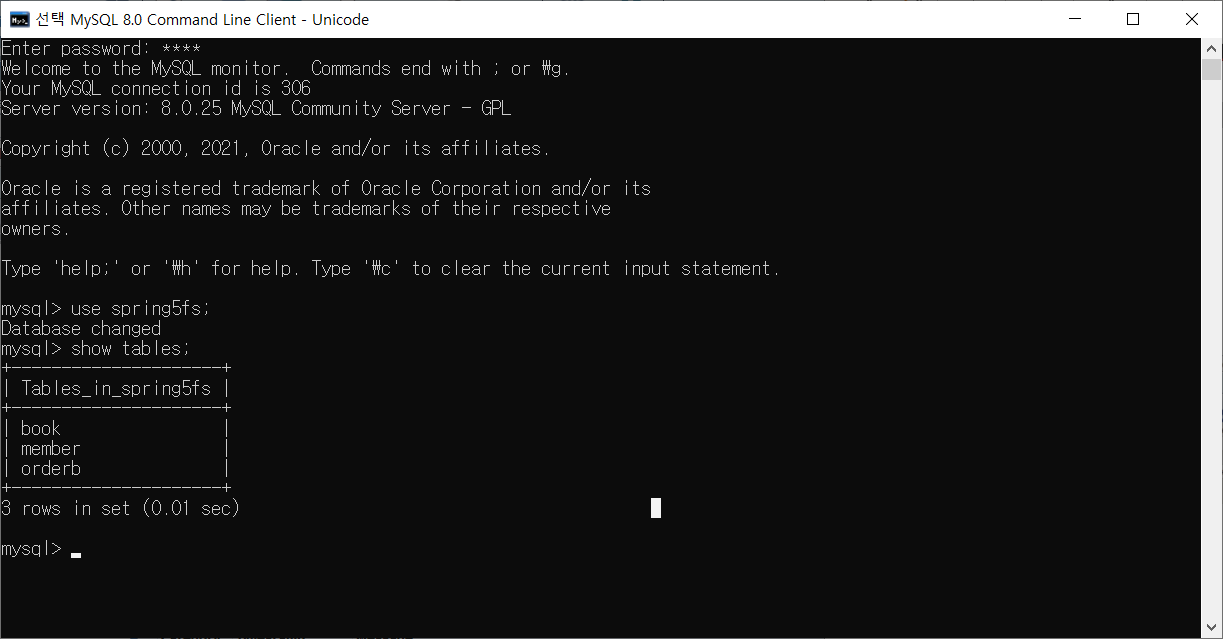
이 클래스는 member의 요소들을 set, get할 수 있는 메서드들을 가지고 있는 클래스입니다. set함수들을 통해서 member요소를 설정하거나 get함수들을 사용해서 member의 요소를 return받을 수 있습니다.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

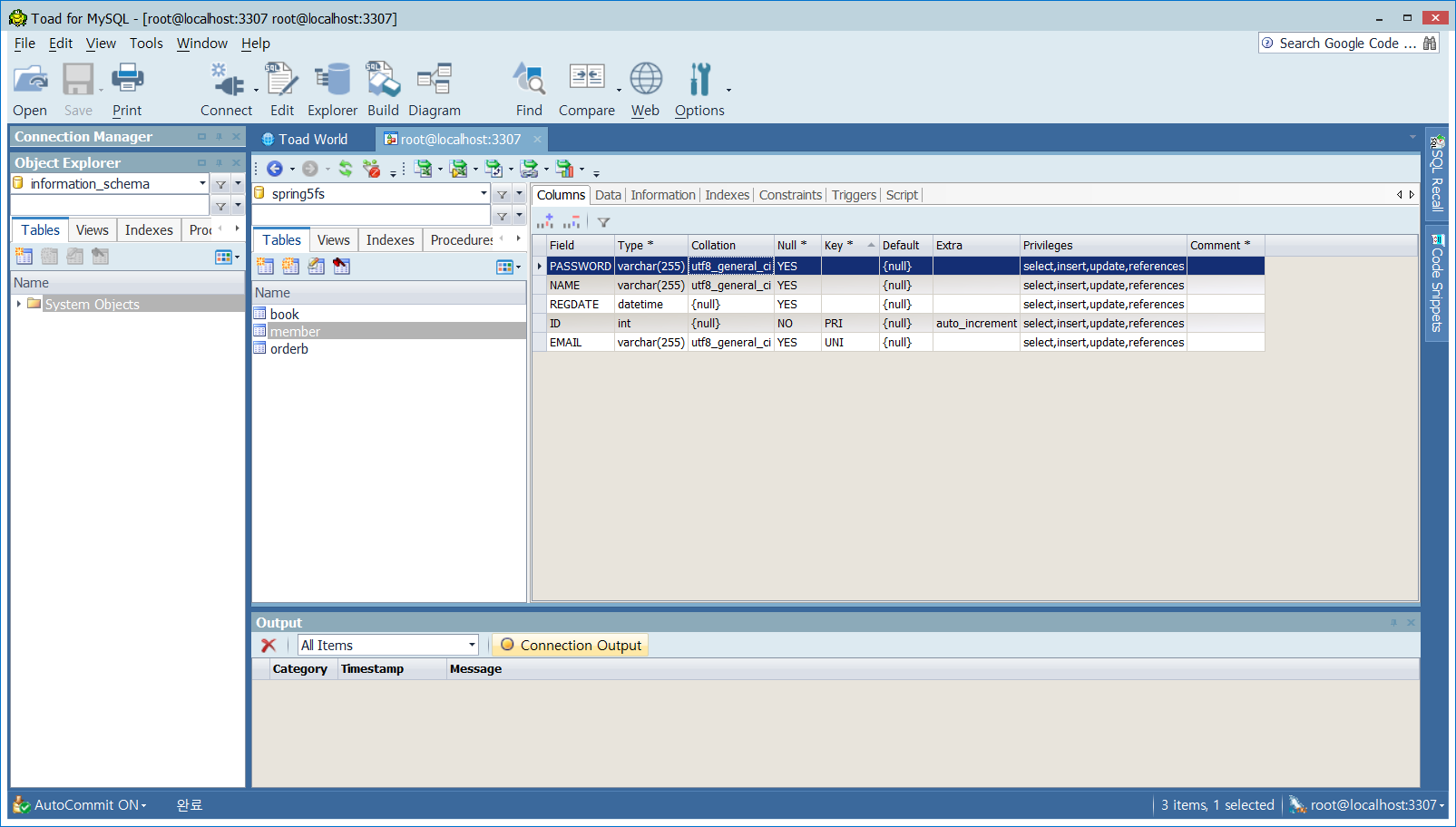
**4. 테이블 구조\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

모든 테이블은 Toad for Mysql에서 작성되었으며,

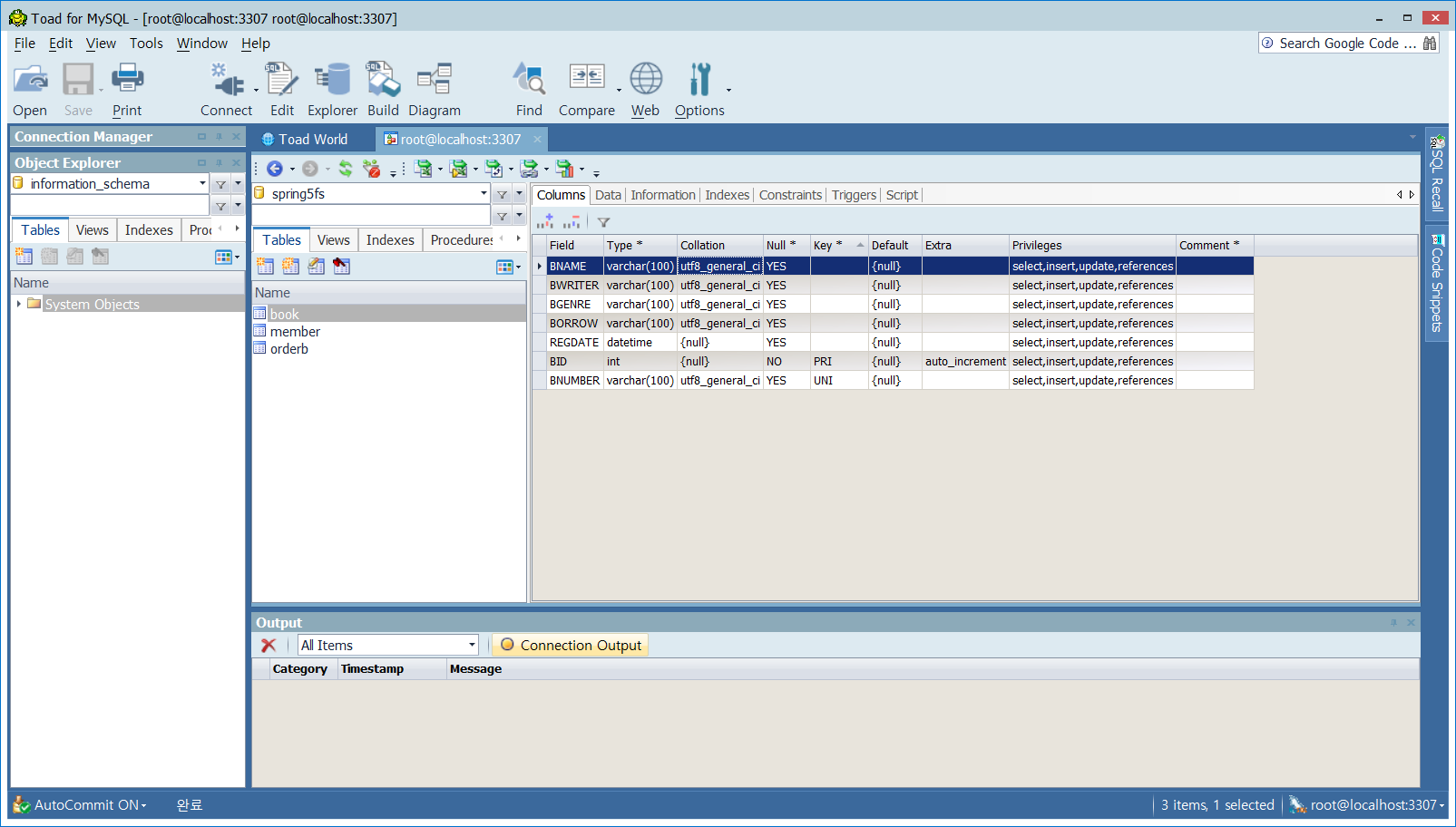
spring5fs라는 데이터베이스에 등록되어져 있습니다.



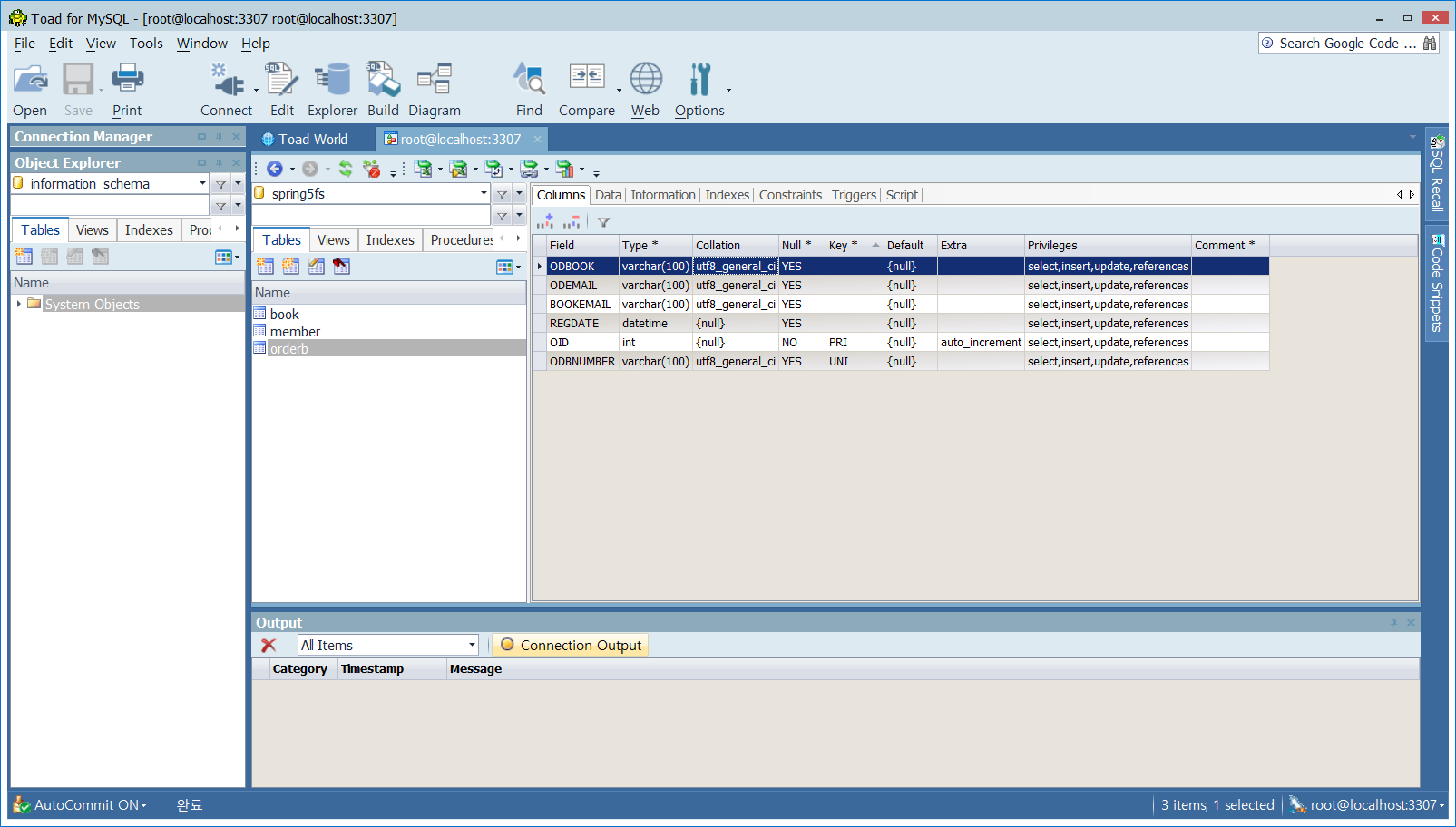
4-1. member table



4-2. book table



4-1. order table



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

5. 향후 활동/후기

처음에 중간 프로젝트를 개발하였을 때, 기말 프로젝트에 대한 부담감과 어려움이 많이 있었습니다. 이유는 제가 데이터베이스와 jsp 관련 부분에서 매우 취약하다고 느끼고 있었기 때문입니다. 따라서 학기중에 조금씩 시간을 내서 제가 부족한 부분을 공부하였습니다. 그래도 개발하는데 어려움이 많았지만 개인 시간을 투자한 것이 효과를 많이 본 것 같습니다. Spring MVC관련해서 프로그래밍 하는 것은 처음 하는 것 이어서 거부감이 있었지만 강의 자료로 제공된 ppt와 실습을 통해서 쉽고 빠르게 개발 능력을 키울 수 있었던 것 같습니다. 제 프로그램이 완벽하지 않고 부족한 점이 많을 수도 있지만, 저는 이 프로젝트를 개발해 봄으로써 개발 능력 향상을 얻었고, 이는 졸업작품이나 다른 프로젝트 개발에 많은 도움이 될 것 같습니다. 또한 향후에 spring을 이용한 개발에 자신감이 많이 생길 것 같습니다. 마지막으로 저는 사용자 보고서, 개발보고서, 기획보고서 등 보고서 작성에 많은 시간을 사용하였습니다. 보고서 작성이 미숙해서 더 많이 신경 쓰고 고민한 것 같습니다. 보고서 작성 능력도 향상된 것 같아서 많은 도움이 되었습니다.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**