

数据库实习报告

——图书管理系统的数据库设计

基于图书馆的基本业务，我们设计了图书馆的图书管理系统，选择了拓展图书预约功能以及用户留言评论功能，即用户可以对任意一本图书发表评论，同时可以对被借出的书进行预约。在需求分析与概念设计阶段，我的工作辅助进行数据库的设计，我们一起设计了数据库的概念模型。

在逻辑设计阶段，我的工作为将E-R图中的实体与联系转化为关系模式中的关系，并且设定各个属性的类型，关系的主键以及外键关联。在主键的设定中，有很多我们都是设定的自动增长的类型，同时为了数据插入，更新，删除的可靠性，我们设定了一系列的外键约束，来保证操作的一致性。

在数据库实施阶段，我的工作为建立实际的数据库，以及编写DAO层，编写测试代码来检测系统的功能。

实际数据库建立如下：

DUANWEIWEI6D24.L...ry - dbo.BookInfo			
	列名	数据类型	允许 Null 值
▶	BookId	int	<input type="checkbox"/>
	ISBN	varchar(20)	<input type="checkbox"/>
	BookName	varchar(50)	<input checked="" type="checkbox"/>
	BookShelfId	varchar(10)	<input checked="" type="checkbox"/>
	BookStatus	varchar(50)	<input checked="" type="checkbox"/>


DUANWEIWEI6D24.L... - dbo.BookShelf			
	列名	数据类型	允许 Null 值
▶	ShelfId	int	<input type="checkbox"/>
	ShelfNo	varchar(10)	<input checked="" type="checkbox"/>
	ShelfName	varchar(30)	<input checked="" type="checkbox"/>
	Capacity	int	<input checked="" type="checkbox"/>
	Location	varchar(10)	<input checked="" type="checkbox"/>
	RoomId	int	<input checked="" type="checkbox"/>


DUANWEIWEI6D24....y - dbo.BookType			
	列名	数据类型	允许 Null 值
▶	TypeId	int	<input type="checkbox"/>
	TypeNO	varchar(50)	<input checked="" type="checkbox"/>
	TypeName	varchar(100)	<input checked="" type="checkbox"/>

DUANWEIWEI6D24.L... - dbo.BorrowInfo		DUANWEIWEI6D24.L...
列名	数据类型	允许 Null 值
 Id	int	<input type="checkbox"/>
BookId	int	<input checked="" type="checkbox"/>
UserId	int	<input checked="" type="checkbox"/>
BorrowTime	date	<input checked="" type="checkbox"/>

DUANWEIWEI6D24.Library - dbo.BT		DUANWEIWEI6D24.L...
列名	数据类型	允许 Null 值
 Id	int	<input type="checkbox"/>
TypeId	int	<input type="checkbox"/>
BookId	int	<input type="checkbox"/>

DUANWEIWEI6D24....y - dbo.Comment		DUANWEIWEI6D24.L...
列名	数据类型	允许 Null 值
 CommentId	int	<input type="checkbox"/>
UserId	int	<input type="checkbox"/>
BookId	int	<input type="checkbox"/>
CommentContext	varchar(500)	<input checked="" type="checkbox"/>
CommentTime	datetime	<input checked="" type="checkbox"/>

DUANWEIWEI6D24.L....DetailedBookInfo		DUANWEIWEI6D24.L...
列名	数据类型	允许 Null 值
 Id	int	<input type="checkbox"/>
BookId	int	<input checked="" type="checkbox"/>
FirstAuthor	varchar(100)	<input checked="" type="checkbox"/>
Description	varchar(500)	<input checked="" type="checkbox"/>
Price	float	<input checked="" type="checkbox"/>
Catalog	varchar(500)	<input checked="" type="checkbox"/>
ContactWay	varchar(50)	<input checked="" type="checkbox"/>
ISOService	varchar(100)	<input checked="" type="checkbox"/>
LoaderShape	varchar(100)	<input checked="" type="checkbox"/>
PublishingInformation	varchar(200)	<input checked="" type="checkbox"/>
Topic	varchar(50)	<input checked="" type="checkbox"/>
SecondAuthor	nvarchar(100)	<input checked="" type="checkbox"/>

DUANWEIWEI6D24.Library - dbo.Reader		DUANWEIWEI6D24.L...
列名	数据类型	允许 Null 值
 UserId	int	<input type="checkbox"/>
UserName	varchar(20)	<input checked="" type="checkbox"/>
Password	varchar(20)	<input type="checkbox"/>
UserType	varchar(10)	<input checked="" type="checkbox"/>
Email	varchar(50)	<input checked="" type="checkbox"/>
BarCode	varchar(20)	<input checked="" type="checkbox"/>
IDCard	varchar(20)	<input type="checkbox"/>
Sex	varchar(10)	<input checked="" type="checkbox"/>
Phone	varchar(20)	<input checked="" type="checkbox"/>
BorrowTimes	int	<input checked="" type="checkbox"/>

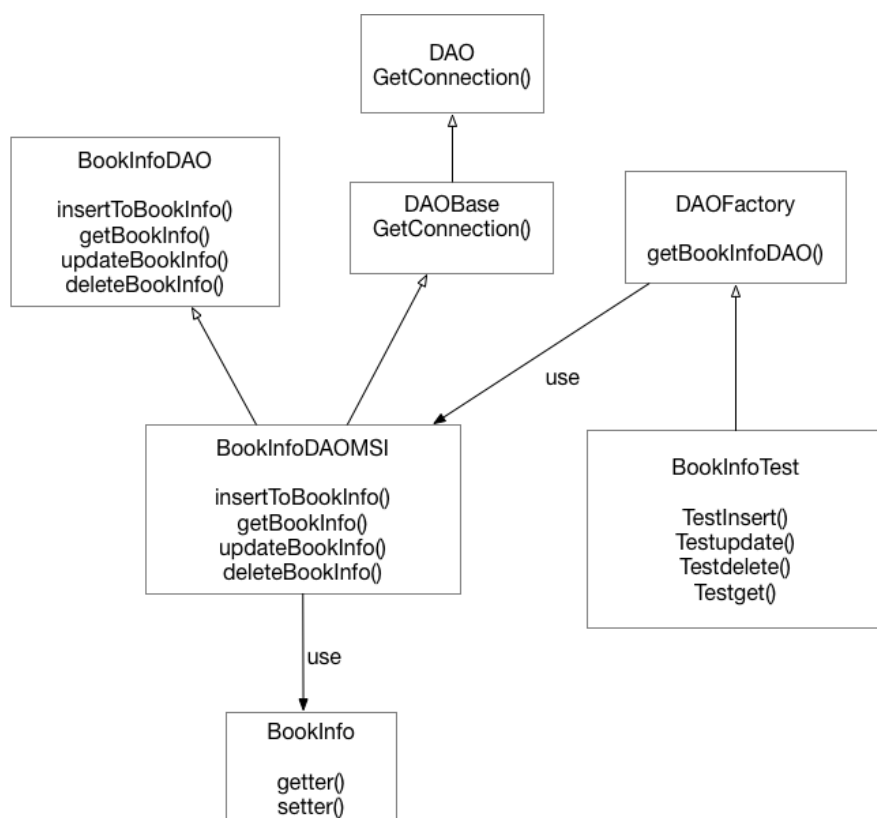
DUANWEIWEI6D24.L... dbo.Reservation			
	列名	数据类型	允许 Null 值
PK	RId	int	<input type="checkbox"/>
	UserId	int	<input type="checkbox"/>
	BookId	int	<input type="checkbox"/>
	ReserveTime	datetime	<input type="checkbox"/>

DUANWEIWEI6D24.L... - dbo.ReturnInfo			
	列名	数据类型	允许 Null 值
PK	Id	int	<input type="checkbox"/>
	UserId	int	<input type="checkbox"/>
	BookId	int	<input type="checkbox"/>
	ReturnTime	date	<input type="checkbox"/>

DUANWEIWEI6D24.Library - dbo.Room			
	列名	数据类型	允许 Null 值
PK	RoomId	int	<input type="checkbox"/>
	RoomNo	varchar(10)	<input checked="" type="checkbox"/>
	RoomName	varchar(10)	<input checked="" type="checkbox"/>
	RoomLocation	varchar(10)	<input checked="" type="checkbox"/>
	RoomCapacity	int	<input checked="" type="checkbox"/>

DAO层的类图如下所示：（以

BookInfo为例）



其中最主要的实现类BookInfoDAOMSI的实现代码如下：

```
public class BookInfoDAOMSI extends DAOBase implements BookInfoDAO {

    private static final String insertBookSQL="insert into
BookInfo(ISBN,BookName,BookShelfId,BookStatus) values(?,?,?,?)";

    @Override

    public int insertBookInfo(BookInfo book) {

        Connection conn = null;

        PreparedStatement pstmt = null;

        try{

            conn = getConnection();

            pstmt = conn.prepareStatement(insertBookSQL);

            pstmt.setString(1, book.getISBN());

            pstmt.setString(2, book.getBookName());

            pstmt.setString(3,book.getShelfId());

            pstmt.setString(4, book.getBookStatus());

            pstmt.executeUpdate();

            pstmt.close();

            conn.close();

            return 1;

        }catch(Exception e){

            e.printStackTrace();

        }

        return -1;

    }
```

```
    private static final String updateBookSQL="update BookInfo set
ISBN=?,BookName=?,BookShelfId=?,BookStatus=? where BookId=?";
```

```
    @Override
```

```

public int updateBookInfo(BookInfo book) {

    Connection conn = null;

    PreparedStatement pstmt = null;

    try{

        conn = getConnection();

        pstmt = conn.prepareStatement(updateBookSQL);

        pstmt.setString(1, book.getISBN());

        pstmt.setString(2, book.getBookName());

        pstmt.setString(3, book.getShelfId());

        pstmt.setString(4, book.getBookStatus());

        pstmt.setInt(5, book.getBookId());


        pstmt.executeUpdate();

        pstmt.close();

        conn.close();

        return 1;

    }catch(Exception e){

        e.printStackTrace();

    }

    return -1;

}

private static final String deleteBookSQL="delete from BookInfo where BookId=?";

@Override

public int deleteBookInfo(BookInfo book) {

    Connection conn = null;

    PreparedStatement pstmt = null;

    try{

```

```

        conn = getConnection();

        pstmt = conn.prepareStatement(deleteBookSQL);
        pstmt.setInt(1,book.getBookId());

        pstmt.executeUpdate();

        pstmt.close();
        conn.close();

        return 1;
    }catch(Exception e){
        e.printStackTrace();
    }
    return -1;
}

```

```

private static final String getBookSQL="select
BookId,ISBN,BookName,BookShelfId,BookStatus from BookInfo where BookId=?";

```

@Override

```

public BookInfo getBookInfo(int id) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    try{
        conn = getConnection();

        pstmt = conn.prepareStatement(getBookSQL);

        pstmt.setInt(1,id );

        rs=pstmt.executeQuery();

        BookInfo temp = null;

```

```

        while(rs.next())

            temp = new

BookInfo(rs.getInt("BookId"),rs.getString("ISBN"),rs.getString("BookName"),rs.getString("BookShe
lfId"),rs.getString("BookStatus"));

            rs.close();

            pstmt.close();

            conn.close();

            return temp;

        }catch(Exception e){

            e.printStackTrace();

        }

        return null;

    }

```

```

private static final String getBookSQL2="select
BookId,ISBN,BookName,BookShelfId,BookStatus from BookInfo where BookName=?";

```

@Override

```

public BookInfo getBookInfo(String BookName) {

    Connection conn = null;

    PreparedStatement pstmt = null;

    ResultSet rs = null;

    try{

        conn = getConnection();

        pstmt = conn.prepareStatement(getBookSQL2);

        pstmt.setString(1,BookName );

    }
}

```

```

        rs=pstm.executeQuery();

        BookInfo temp = null;

        while(rs.next())

            temp = new

BookInfo(rs.getInt("BookId"),rs.getString("ISBN"),rs.getString("BookName"),rs.getString("BookShe
lfId"),rs.getString("BookStatus"));

        rs.close();

        pstm.close();

        conn.close();

        return temp;

    }catch(Exception e){

        e.printStackTrace();

    }

    return null;}}

```

我们使用JUnit测试，测试报告如下：

1简介

1.1编写目的

实例：本测试报告为图书管理系统数据库测试报告，目的在于总结测试阶段的测试以及分析测试结果，描述该数据库是否实现对数据表的增删改查需求，预期参考人员包括用户、测试人员、开发人员、项目管理者、其他质量管理人员和需要阅读本报告的高层经理。

1.2项目背景

图书管理系统是当前各高校的主要业务系统。请结合北京林业大学图书管理系统lib.bjfu.edu.cn进行数据库设计，并完成相应的DAO实现及测试工作。

1.3数据库简介

数据库类型SQLSERVER2008

运行环境：Windows7 64位操作系统

1.4数据表

a) BookInfo:图书信息表：

BookInfo：书籍基本信息表							
字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
Bookid	书籍ID	int		NOT NULL	主键		AUTO_INCREMENT
ISBN	ISBN编号	varchar	20	NOT NULL		NULL	
BookName	书籍名	varchar	50			NULL	
BookShelfId	书架id	varchar	10			NULL	
BookStatus	书籍状态	varchar	50			NULL	

b) BookShelf书架信息表：

BookShelf书架信息表							
字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
ShelfId	书架ID	int		NOT NULL	主键		AUTO_INCREMENT
ShelfNo	书架编号	varchar	10			NULL	
ShelfName	书架名	varchar	30			NULL	
Capacity	书架容量	int				NULL	
Location	书架位置	varchar	10			NULL	
RoomId	存放阅览室ID	int			外键	NULL	

c) BookType：类别表：

BookType：类别表							
字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
TypeID	类别ID	int		NOT NULL	主键		AUTO_INCREMENT
TypeNO	类别编号	varchar	50			NULL	
TypeName	类别名	varchar	100			NULL	

c) BorrowInfo借阅信息表：

BorrowInfo: 借阅信息表							
字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
Id	借阅id	int		NOT NULL	主键		AUTO_INCREMENT
bookId	书籍id	int		NOT NULL	外键	NULL	
UserId	用户id	int		NOT NULL	外键	NULL	
Borrowtime	借阅时间	date		NOT NULL		NULL	

e) BT 图书类别表：

BT: 图书类别表							
字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
Id	ID	int		NOT NULL	主键		AUTO_INCREMENT
TypeId	类别ID	int			外键	NULL	
BookId	书籍ID	int			外键	NULL	

f) Comment评论表：

Comment: 评论表							
字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
CommentId	收藏ID	int		NOT NULL	主键		AUTO_INCREMENT
UserId	用户id	int		NOT NULL	外键	NULL	
BookId	书籍id	int		NOT NULL	外键	NULL	
CommentContext	评论内容	varchar	500			NULL	
CommentTime	评论时间	datetime				NULL	

g) DetailedBookInfo 图书详情表：

MessageInfo: 留言表							
字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
Id	ID	int		NOT NULL	主键		AUTO_INCREMENT
BookId	图书id	int		NOT NULL	外键	NULL	
FirstAuthor	第一作者	varchar	100			NULL	
Description	简介	varchar	500			NULL	
Price	价格	float				NULL	
Catalog	目录	varchar	500			NULL	

ContactWay	联系方式	varchar	50			NULL	
ISOService	光盘镜像服务	varchar	100			NULL	
LoaderShape	载体形态项	varchar	100			NULL	
PublishingInformation	出版信息	varchar	200			NULL	
Topic	学科主题	varchar	50			NULL	
SecondAuthor	第二作者	varchar	100			NULL	

Room信息表：

字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
RoomId	阅览室ID	int		NOT NULL	主键		AUTO_INCREMENT
RoomNo	阅览室编号	varchar (10)	10			NULL	
RoomName	阅览室名	varchar (10)	10			NULL	
RoomLocation	阅览室位置	varchar (10)	10			NULL	
RoomCapacity	阅览室容量	int	11			NULL	

预约信息表

字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
RId	预约ID	int		NOT NULL	主键		AUTO_INCREMENT
UserId	用户ID	int		NOT NULL			
BookId	预订书籍ID	int		NOT NULL			
ReserveTime	预定日期	datetime		NOT NULL			

归还信息表

字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
Id	归还信息ID	int		NOT NULL	主键		AUTO_INCREMENT
BookId	归还书籍ID	int		NOT NULL			
UserId	归还用户ID	int		NOT NULL			
ReturnTime	归还日期	date		NOT NULL			

读者详情表

||
||
||

字段名称	描述	类型	长度	允许空值	主外键	默认值	是否自增
UserId	用户ID	int		NOT NULL	主键		AUTO_INCREMENT
UserName	用户姓名	varchar(20)	20			NULL	
Password	用户密码	varchar(20)	20	NOT NULL			
UserType	用户类型	varchar(10)	10			NULL	
Email	用户邮箱	varchar(50)	50			NULL	
BarCode	用户条码号	varchar(20)	20			NULL	
IDNO	用户证件号	varchar(20)	20	NOT NULL			
Sex	用户性别	varchar(10)	10			NULL	
Phone	用户电话号码	varchar(20)	20			NULL	
BorrowTime	借出时间	Int				NULL	

1.对Reader表的增删改查测试：

原Reader表如下图所示：

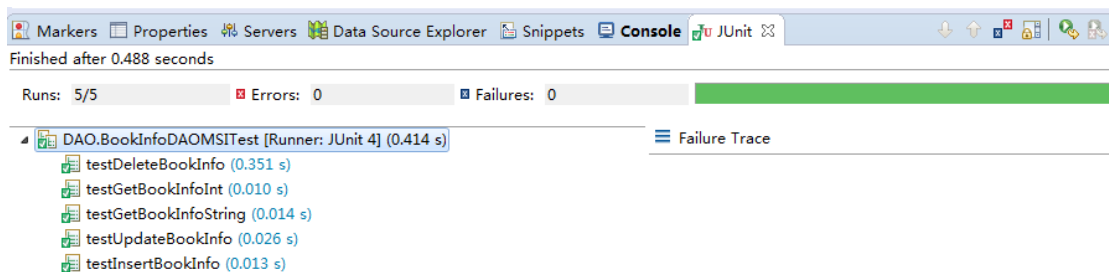
210A.Library - dbo.Reader							
	UserId	UserName	Password	UserType	Email	BarCode	IDNO
▶	1	Sheldon Lee Coope	123	教师	Shearduocx@ya...	164923456	742532147
	2	Leslie Winkle	123	教师	Leslieweng46@y...	162873248	726385636
	3	Melissa Rauch	123	教师	Melissabeauty@...	162034568	754239456
	4	Emma Green	123	学生	LitterEmma34@...	162702145	723452567
	5	Bob Martin	123	学生	Bob123@126.com	161253742	735873735
	6	Willian Sheeran	123	学生	SuperWillian@16...	163802193	732839423
	7	James Foster	123	教师	JameFT@126.com	163893723	782937268
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

测试代码：（增加了读者Betty以及她的详细信息，更新第一个读者的信息（将其名字改为Peter2），查询第一个读者的用户ID）删除了第六个读者的信息。

```

9-  @Test
10-  public void testInsertReader() {
11-      Reader r=new Reader();
12-      r.setUserName("Betty");
13-      r.setPassword("1234");
14-      r.setBarCode("2222");
15-      r.setBorrowTimes(0);
16-      r.setEmail("Betty@2222");
17-      r.setIDCard("77777");
18-      r.setPhone("123456");
19-      r.setSex("女");
20-      r.setUserType("学生");
21-      ReaderDAO MSI rd=new ReaderDAO MSI();
22-      assertEquals(1,rd.insertReader(r));
23-  }
24-
25-  @Test
26-  public void testUpdateReader() {
27-      ReaderDAO MSI rd=new ReaderDAO MSI();
28-      Reader r=rd.getReader(1);
29-      r.setUserName("Peter2");
30-      assertEquals(1,rd.updateReader(r));
31-  }
32-
33-  @Test
34-  public void testGetReader() {
35-      ReaderDAO MSI rd=new ReaderDAO MSI();
36-      Reader r=rd.getReader(1);
37-      assertEquals(1,r.getUserId());
38-  }
39-
40-  @Test
41-  public void testDeleteReader() {
42-      ReaderDAO MSI rd=new ReaderDAO MSI();
43-      assertEquals(1,rd.deleteReader(6));
44-  }

```



更改插入删除查询成功：

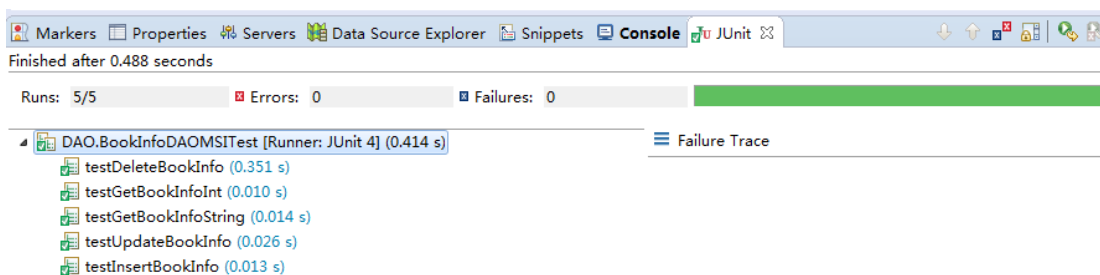
Userid	UserName	Password	UserType	Email	BarCode	IDCard	Sex	Phone	BorrowTimes
1	Peter2	123	教师	Shearduocx@ya...	164923456	742532147	男	199999999	33
2	Leslie Winkle	123	教师	Lesleweng46@y...	162873248	726385636	女	155555555	36
3	Melissa Rauch	123	教师	Melissabeauty@...	162034568	754239456	女	133333333	38
4	Emma Green	123	学生	LitterEmma34@...	162702145	723452567	女	177777777	19
5	Bob Martin	123	学生	Bob123@126.com	161253742	735873735	男	188888888	21
7	James Foster	123	教师	JameFT@126.com	163893723	782937268	男	100000000	28
9	Peter	1234	学生	Peter@11111	111111	888888	男	123456	0
10	Betty	1234	学生	Betty@2222	2222	77777	女	123456	0

2.对BookInfo表的增删改查测试：

原Bookinfo表：

	BookId	ISBN	BookName	BookShelfId	BookStatus
▶	1	978-7-5366-929...	《三体》	A111	可借
	2	978-7-5302-101...	《平凡的世界》	A123	可借
	3	978-7-115-2757...	《JavaScript高...	B216	可借
	4	7-80120-859-5	《了不起的盖...	B227	可借
	5	978-7-5063-654...	《活着》	C149	可借
	6	978-7-121-2105...	《MATLAB编程...	D213	借出-应归还日...
	7	7-5063-0974-2	《苏菲的世界》	C127	借出-应归还日...
	8	7-112-05043-X	《西方现代景...	D423	可借
	9	7-5357-3230-5	《时间简史》	D207	可借
	10	978-7-5680-077...	《国防论》	D312	可借
	11	978-7-5080-510...	《非暴力沟通》	D324	借出-应归还日...
※	NULL	NULL	NULL	NULL	NULL

测试代码:



在BookInfo表里增加名为bookname的书还有它的详细信息，更新bookid为3的书籍信息（将其名字改为数据库原理与应用），删除id为10的书籍信息，查询id为5的图书信息

```

10
11 public void testInsertBookInfo() {
12     BookInfoDAOMSI bidm=new BookInfoDAOMSI();
13     BookInfo b=new BookInfo();
14     b.setBookName("bookname");
15     b.setISBN("9999999");
16     b.setBookStatus("可借");
17     b.setShelfId("A111");
18     assertEquals(1,bidm.insertBookInfo(b));
19 }
20 @Test
21 public void testUpdateBookInfo() {
22     BookInfoDAOMSI bidm=new BookInfoDAOMSI();
23     BookInfo b=bidm.getBookInfo(3);
24     b.setBookName("《数据库原理》");
25     b.setShelfId("A123");
26     assertEquals(1,bidm.updateBookInfo(b));
27 }
28 @Test
29 public void testDeleteBookInfo() {
30     BookInfoDAOMSI bidm=new BookInfoDAOMSI();
31     BookInfo b=bidm.getBookInfo(10);
32     assertEquals(1,bidm.deleteBookInfo(b));
33 }
34
35 @Test
36 public void testGetBookInfoInt() {
37     BookInfoDAOMSI bidm=new BookInfoDAOMSI();
38     BookInfo b=bidm.getBookInfo(7);
39     assertEquals(7,b.getBookId());
40 }
41 @Test
42 public void testGetBookInfoString() {
43     BookInfoDAOMSI bidm=new BookInfoDAOMSI();
44     BookInfo b=bidm.getBookInfo("《活着》");
45     assertEquals(5,b.getBookId());
46 }

```

插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：

BookId	ISBN	BookName	BookShelfId	BookStatus
2	978-7-5302-101...	《平凡的世界》	A123	可借
3	978-7-115-2757...	数据库原理	A123	可借
4	7-80120-859-5	《了不起的盖...	B227	可借
5	978-7-5063-654...	《活着》	C149	可借
6	978-7-121-2105...	《MATLAB编程...	D213	借出-应归还日...
7	7-5063-0974-2	《苏菲的世界》	C127	借出-应归还日...
8	7-112-05043-X	《西方现代景...	D423	可借
9	7-5357-3230-5	《时间简史》	D207	可借
10	978-7-5680-077...	《国防论》	D312	可借
11	978-7-5080-510...	《非暴力沟通》	D324	借出-应归还日...

DUANWEIWEI6D24.L...ry - dbo.BookInfo					
	BookId	ISBN	BookName	BookShelfId	BookStatus
▶	2	978-7-5302-101...	《平凡的世界》	A123	可借
	3	978-7-115-2757...	《追风筝的人》	A666	可借
	4	7-80120-859-5	《了不起的盖...	B227	可借
	5	978-7-5063-654...	《活着》	C149	可借
	6	978-7-121-2105...	《MATLAB编程...	D213	借出-应归还日...
	7	7-5063-0974-2	《苏菲的世界》	C127	借出-应归还日...
	8	7-112-05043-X	《西方现代景...	D423	可借
	9	7-5357-3230-5	《时间简史》	D207	可借
	10	978-7-5680-077...	《国防论》	D312	可借
	12	9999999	bookname	A111	可借

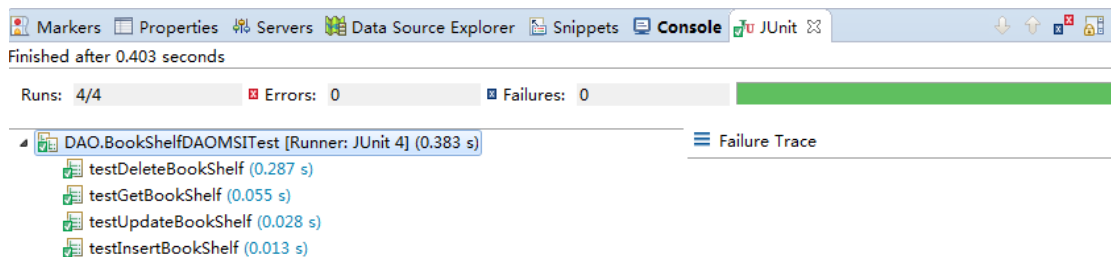
(3) 对BookShelf表的增删改查测试

查询书架名“A111”的书架信息，插入301第一书架的书架信息，更新书架名为A111的书架信息(将其容量改为70)，删除书架id为2的书架信息。

```

11 @Test
12 public void testGetBookShelf() {
13     BookShelfDAO bsdm=new BookShelfDAO();
14     BookShelf bs=bsdm.getBookShelf("A111");
15     assertEquals("A111",bs.getShelfNo());
16 }
17
18 @Test
19 public void testInsertBookShelf() {
20     BookShelfDAO bsdm=new BookShelfDAO();
21     BookShelf bs=new BookShelf();
22     bs.setCapacity(20);
23     bs.setLocation("301阅览室");
24     bs.setRoomId(1);
25     bs.setShelfName("301第一书架");
26     bs.setShelfNo("A001");
27     assertEquals(1,bsdm.insertBookShelf(bs));
28 }
29
30 @Test
31 public void testUpdateBookShelf() {
32     BookShelfDAO bsdm=new BookShelfDAO();
33     BookShelf bs=bsdm.getBookShelf("A111");
34     bs.setCapacity(70);
35     assertEquals(1,bsdm.updateBookShelf(bs));
36 }
37
38 @Test
39 public void testDeleteBookShelf() {
40     BookShelfDAO bsdm=new BookShelfDAO();
41
42     assertEquals(1,bsdm.deleteBookShelf(2));
43 }
44
45 }

```

插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：

ShelfId	ShelfNo	ShelfName	Capacity	Location	RoomId
1	A111	301 第十一书架	70	301阅览室	1
3	B216	302 第十六书架	50	302阅览室	2
4	B227	302 第二十七...	40	302阅览室	2
5	C119	401 第十九书架	40	401阅览室	4
6	D213	402 第十三书架	50	402阅览室	5
7	C123	401 第二十三...	40	401阅览室	4
8	D423	密集书库 第二...	50	密集书库	3
9	D207	402 第七书架	50	402阅览室	5
10	D312	403 第十二书架	50	403阅览室	6
14	A001	301第一书架	20	301阅览室	1

ShelfId	ShelfNo	ShelfName	Capacity	Location	RoomId
1	A111	301 第十一书架	40	301阅览室	1
2	A123	301 第二十三...	40	301阅览室	1
3	B216	302 第十六书架	50	302阅览室	2
4	B227	302 第二十七...	40	302阅览室	2
5	C119	401 第十九书架	40	401阅览室	4
6	D213	402 第十三书架	50	402阅览室	5
7	C123	401 第二十三...	40	401阅览室	4
8	D423	密集书库 第二...	50	密集书库	3
9	D207	402 第七书架	50	402阅览室	5
10	D312	403 第十二书架	50	403阅览室	6

4.对ReturnInfo表的增删改查测试

查询归还ID为1的归还记录信息，增加用户ID为5的用户归还信息，更新用户ID为2的用户归还信息（将其用户ID设为2，书籍ID设为4）删除归还ID为5的归还记录信息。

原ReturnInfo表：

列名	数据类型	允许 Null 值
Id	int	<input type="checkbox"/>
BookId	int	<input type="checkbox"/>
UserId	int	<input type="checkbox"/>
ReturnTime	date	<input type="checkbox"/>

```

1 package DAO;
2
3 import static org.junit.Assert.*;
4
5 public class ReturnInfoDAOMSITest {
6
7     @Test
8     public void testInsertReturnInfo() { //add
9         ReturnInfoDAOMSI rdm=new ReturnInfoDAOMSI();
10        ReturnInfo r=new ReturnInfo();
11        r.setUserId(5);
12        r.setBookId(3);
13        assertEquals(1,rdm.insertReturnInfo(r));
14    }
15
16    @Test
17    public void testUpdateReturnInfo() { //update the returninfo where id=1 ,set userid=2 and bookid=4
18        ReturnInfoDAOMSI rdm=new ReturnInfoDAOMSI();
19        ReturnInfo r=rdm.getReturnInfo(1);
20        r.setUserId(2);
21        r.setBookId(4);
22        assertEquals(1,rdm.updateReturnInfo(r));
23    }
24
25    @Test
26    public void testDeleteReturnInfo() { //delete where id=5
27        ReturnInfoDAOMSI rdm=new ReturnInfoDAOMSI();
28        assertEquals(1,rdm.deleteReturnInfo(5));
29    }
30
31    @Test
32    public void testGetReturnInfo() { //find where id=1
33        ReturnInfoDAOMSI rdm=new ReturnInfoDAOMSI();
34        assertEquals(1,rdm.getReturnInfo(1).getId());
35    }
36 }

```

Markers Properties Servers Data Source Explorer Snippets Console JUnit

Finished after 0.507 seconds

Runs: 4/4 Errors: 0 Failures: 0

DAO.ReturnInfoDAOMSITest [Runner: JUnit 4] (0.426 s)

- testGetReturnInfo (0.364 s)
- testDeleteReturnInfo (0.013 s)
- testUpdateReturnInfo (0.039 s)
- testInsertReturnInfo (0.010 s)

Failure Trace

插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：

Id	UserId	BookId	ReturnTime
1	1	2	2018-08-06
2	3	3	2018-09-13
3	2	5	2018-09-20
5	5	4	2018-11-15

Id	UserId	BookId	ReturnTime
1	2	4	2018-08-06
2	3	3	2018-09-13
3	2	5	2018-09-20
6	5	3	2018-12-22

5.对Reservation表的增删改查测试

原Reservation表：

210A.Library - dbo.Reservation				
	RId	UserId	BookId	ReserveTime
▶	1	1	6	2018-12-19 07:...
	2	4	7	2018-12-20 12:...
	3	2	11	2018-12-21 17:...
*	NULL	NULL	NULL	NULL

插入UserID为5的预定图书ID为6的信息，更新预定ID为一的预定记录（将其BookID设为7）删除预定ID为2的预定记录，查询预定ID为1的预定记录。

测试代码：

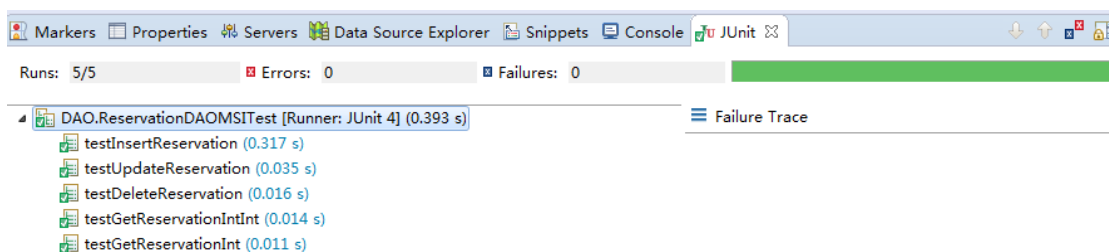
```
@Test
public void testInsertReservation() {
    ReservationDAOMSI rd=new ReservationDAOMSI();
    Reservation r=new Reservation();
    r.setBookId(6);
    r.setUserId(5);
    assertEquals(1,rd.insertReservation(r));
}

@Test
public void testUpdateReservation() {
    ReservationDAOMSI rd=new ReservationDAOMSI();
    Reservation r=rd.getReservation(1);
    r.setBookId(7);
    assertEquals(1,rd.updateReservation(r));
}

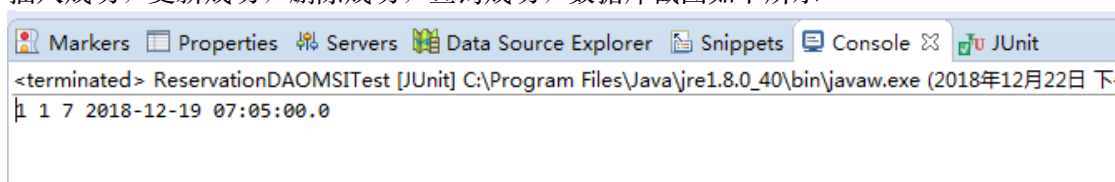
@Test
public void testDeleteReservation() {
    ReservationDAOMSI rd=new ReservationDAOMSI();
    assertEquals(1,rd.deleteReservation(2));
}

@Test
public void testGetReservationIntInt() {
    ReservationDAOMSI rd=new ReservationDAOMSI();
    assertEquals(1,rd.getReservation(1).getRId());
}

@Test
public void testGetReservationInt() {
    ReservationDAOMSI rd=new ReservationDAOMSI();
    List<Reservation> lr=rd.getReservation(1, 7);
    Reservation r=lr.get(0);
    System.out.println(r.getRId()+" "+r.getUserId()+" "+r.getBookId()+" "+r.getReserveTime());
    assertEquals(1,lr.size());
}
```



插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：



DUANWEIWEI6D24.L... dbo.Reservation		DUANWEIWEI6D24.L....DetailedB		
	RId	UserId	BookId	ReserveTime
▶	1	1	6	2018-12-19 07:...
	2	4	7	2018-12-20 12:...

RId	UserId	BookId	ReserveTime
1	1	7	2018-12-19 07:...
4	5	6	2018-12-22 14:...

6.对Comment表的增删改查测试

原Comment表：

210A.Library - dbo.Comment		210A.Library - dbo.Reservation			
	CommentId	UserId	BookId	CommentContext	CommentTime
▶	1	3	1	最牛逼的科幻...	2018-09-14 20:...
	2	4	5	接踵而来的命...	2018-10-15 15:...
	3	3	3	很不错，全面...	2018-10-21 18:...
	4	5	4	盖茨比爱的不...	2018-11-01 21:...
	5	1	2	一直都看的很...	2018-11-25 14:...
	6	7	9	宇宙学科的入...	2018-11-28 08:...
	9	6	10	总体战的时代...	2018-12-01 09:...
	10	2	6	书如其名，大...	2018-12-03 11:...
*	NULL	NULL	NULL	NULL	NULL

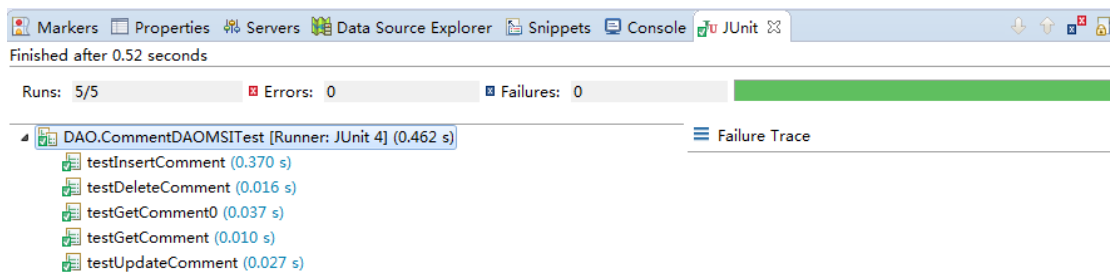
插入用户UserID为5的用户对书籍ID为2的评论，更新用户CommentID为2的留言信息（将其留言内容改为Wonderful）删除留言ID为11的留言信息，查询留言ID为2的留言信息。

测试代码：

```

4 public void testInsertComment() { //add
5     CommentDAO MSI cdm=new CommentDAO MSI();
6     Comment c=new Comment();
7     c.setUserId(5);
8     c.setBookId(2);
9     c.setCommentContext("射人杀马");
10    assertEquals(1,cdm.insertComment(c));
11 }
12 @Test
13 public void testUpdateComment() { //update the comment where id=2 ,set context="Wonderful"
14     CommentDAO MSI cdm=new CommentDAO MSI();
15     Comment c=cdm.getComment(2);
16     c.setCommentContext("Wonderful");
17     assertEquals(1,cdm.updateComment(c));
18 }
19 @Test
20 public void testDeleteComment() { //delete the comment where id=11
21     CommentDAO MSI cdm=new CommentDAO MSI();
22     assertEquals(1,cdm.deleteComment(11));
23 }
24 @Test
25 public void testGetComment() { //find the comment where id=2
26     CommentDAO MSI cdm=new CommentDAO MSI();
27     Comment c=cdm.getComment(2);
28     assertEquals(2,c.getId());
29 }
30 @Test
31 public void testGetComments() { //find the comments where userid=3 and bookid=3
32     CommentDAO MSI cdm=new CommentDAO MSI();
33     List<Comment> lc=cdm.getComments(3, 3);
34     for(int i=0;i<lc.size();i++){
35         Comment c=lc.get(i);
36         System.out.println(c.getId()+" "+c.getUserId()+" "+c.getBookId()+" "+c.getCommentContext()+" "+c.getCommentTime());
37     }
38     assertEquals(3,lc.size());
39 }

```



插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：

CommentId	UserId	BookId	CommentContext	CommentTime
2	4	5	文学著作，值...	2018-10-15 15:...
3	3	3	很不错，全面...	2018-10-21 18:...
4	5	4	盖茨比爱的不...	2018-11-01 21:...
5	1	2	一直都看的很...	2018-11-25 14:...
6	7	9	宇宙学科的入...	2018-11-28 08:...
11	4	4	it Worth reading	2018-12-22 12:...

CommentId	UserId	BookId	CommentContext	CommentTime
2	4	5	Wonderful	2018-10-15 15:...
3	3	3	很不错，全面...	2018-10-21 18:...
4	5	4	盖茨比爱的不...	2018-11-01 21:...
5	1	2	一直都看的很...	2018-11-25 14:...
6	7	9	宇宙学科的入...	2018-11-28 08:...
12	5	2	耐人寻味	2018-12-22 13:...

7.对BT表的增删改查测试

原BT表：

Id	TypeId	BookId
1	1	1
2	1	2
3	5	5
4	2	6
5	3	8
6	4	7
7	6	4
8	7	10
9	2	3
10	7	10
11	9	11
* NULL	NULL	NULL

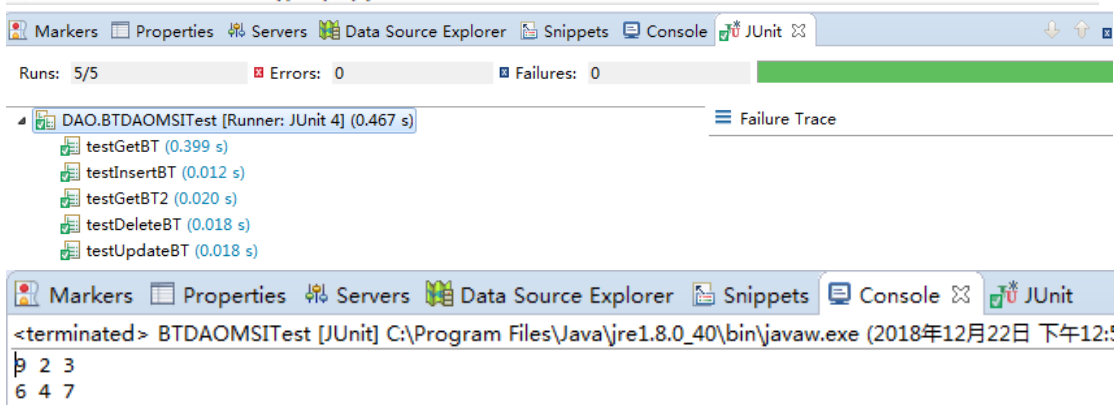
测试代码：

插入TypeId为2，BookID为5的图书信息，更新BTID为2的图书类别信息（将其类别ID设为7），将BTID为

10的那条记录信息，查询BTID为3的记录信息。

```
~
public void testInsertBT() { //add
    BTDAOMSI btdm=new BTDAOMSI();
    BT bt=new BT();
    bt.setTypeId(2);
    bt.setBookId(5);
    assertEquals(1,btdm.insertBT(bt));
}
@Test
public void testUpdateBT() { //update the BT where bookId=2 ,set TypeId=7
    BTDAOMSI btdm=new BTDAOMSI();
    List<BT> lb=btdm.getBT(2);
    BT bt=lb.get(0);
    bt.setTypeId(7);
    assertEquals(1,btdm.updateBT(bt));
}
@Test
public void testDeleteBT() {
    BTDAOMSI btdm=new BTDAOMSI();
    assertEquals(1,btdm.deleteBT(10));
}
@Test
public void testGetBT() {
    BTDAOMSI btdm=new BTDAOMSI();
    List<BT> lb=btdm.getBT(3);
    for(int i=0;i<lb.size();i++){
        BT b=lb.get(i);
        System.out.println(b.getId()+" "+b.getTypeId()+" "+b.getBookId());
    }
    assertEquals(9,lb.get(0).getId());
}
@Test
public void testGetBT2() {
    BTDAOMSI btdm=new BTDAOMSI();
    List<BT> lb=btdm.getBT2(4);
    for(int i=0;i<lb.size();i++){
        BT b=lb.get(i);

```



插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：

	Id	TypeId	BookId
	2	1	2
	3	5	5
	4	2	6
	5	3	8
	6	4	7
	7	6	4
	8	7	10
	9	2	3
	10	7	10

	Id	TypeId	BookId
	2	7	2
	3	5	5
	4	2	6
	5	3	8
	6	4	7
	7	6	4
	8	7	10
	9	2	3
	12	2	5

8.对BorrowInfo表的增删改查测试

原Borrowinfo表：

210A.Library - dbo.BorrowInfo				
	Id	BookId	UserId	BorrowTime
▶	1	2	1	2018-07-12
	2	3	3	2018-08-22
	4	5	2	2018-09-15
	5	1	4	2018-10-17
	6	4	5	2018-10-26
	7	6	6	2018-12-18
	8	7	4	2018-12-19
	9	11	7	2018-12-20
*	NULL	NULL	NULL	NULL

插入用户为2，借阅书籍ID为10的借阅信息。更新借阅ID为1的记录的信息（将其日期设为12月20日），删除借阅ID为8的记录，查询借阅ID为3的借阅信息。

```

BorrowInfoDAOMSTest.java BorrowInfoDAOMSI.java
15 @Test
16 public void testInsertBorrowInfo() { //Add
17     BorrowInfoDAOMSI bidm=new BorrowInfoDAOMSI();
18     BorrowInfo bi=new BorrowInfo();
19     bi.setBookId(10);
20     bi.setUserId(2);
21     assertEquals(1,bidm.insertBorrowInfo(bi));
22 }
23 @Test
24 public void testUpdateBorrowInfo() { //update BorrowInfo where id=1,set Date="2018-12-20"
25     BorrowInfoDAOMSI bidm=new BorrowInfoDAOMSI();
26     BorrowInfo b=bidm.getBorrowInfo(1);
27     Date date=new Date(2018-12-20);
28     b.setBorrowTime(date);
29     assertEquals(1,bidm.updateBorrowInfo(b));
30 }
31 @Test
32 public void testDeleteBorrowInfo() { //delete BorrowInfo where id=8
33     BorrowInfoDAOMSI bidm=new BorrowInfoDAOMSI();
34     assertEquals(1,bidm.deleteBorrowInfo(8));
35 }
36 @Test
37 public void testGetBorrowInfo() { //find BorrowInfo where id=2
38     BorrowInfoDAOMSI bidm=new BorrowInfoDAOMSI();
39     BorrowInfo b=bidm.getBorrowInfo(2);
40     assertEquals(2,b.getId());
41 }
42 @Test
43 public void testGetBorrowInfo2() { //get BorrowInfo where userId=3
44     BorrowInfoDAOMSI bidm=new BorrowInfoDAOMSI();
45     List<BorrowInfo> lb=bidm.getBorrowInfo(3);
46     for(int i=0;i<lb.size();i++){
47         BorrowInfo b=lb.get(i);
48         System.out.println(b.getId()+" User:"+b.getUserId()+" BookId:"+b.getBookId()+" BorrowTime:"+b.getBorrowTime());
49     }
50     assertEquals(2,lb.get(0).getId());

```

Markers Properties Servers Data Source Explorer Snippets Console JUnit

Runs: 5/5 Errors: 0 Failures: 0

DAO.BorrowInfoDAOMSTest [Runner: JUnit 4] (0.434 s)

testGetBorrowInfo2 (0.357 s)

testGetBorrowInfo (0.011 s)

testDeleteBorrowInfo (0.019 s)

testUpdateBorrowInfo (0.036 s)

testInsertBorrowInfo (0.011 s)

插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：

Markers Properties Servers Data Source Explorer Snippets Console

<terminated> BorrowInfoDAOMSTest [JUnit] C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (2018-08-22 10:22)

User:3 BookId:3 BorrowTime:2018-08-22

DUANWEIWEI6D24.L... - dbo.BorrowInfo		DUANWEIWEI6D24....y - dbo.BookType		
	Id	BookId	UserId	BorrowTime
▶	1	2	1	2018-07-12
	2	3	3	2018-08-22
	4	5	2	2018-09-15
	6	4	5	2018-10-26
	8	7	4	2018-12-19

DUANWEIWEI6D24.L... - dbo.BorrowInfo		DUANWEIWEI6D24....y - dbo.BookType		
	Id	BookId	UserId	BorrowTime
	1	2	1	2018-12-20
	2	3	3	2018-08-22
	4	5	2	2018-09-15
	6	4	5	2018-10-26
▶	10	10	2	2018-12-22

9.对DetailedBookinfo表的增删改查测试

原detailedinfo表：

210A.Library - dbo.DetailedBookInfo		210A.Library - dbo.BorrowInfo		210A.Library - dbo.BT		
	Id	BookId	FirstAuthor	Description	Price	ContactWay
▶	1	1	刘慈欣	文化大革命如...	23	1. 科学边界 ...
	2	2	路遥	《平凡的世界...	88	第一部
	3	3	Nicholas C. Zakas	所属分类: 计...	99	第1章 JavaScr...
	4	4	菲茨杰拉德	经典的美国梦...	19.8	第一章 第二章 ...
	5	5	余华	《活着》是一...	20	第1章
	6	6	赵书兰	《MATLAB编程...	59	1.1 MATLAB的...
	7	7	乔斯坦·费德	14岁的少女苏...	26.8	1.伊甸园-在某...
	8	8	王向荣, 林菁著	《西方现代景...	210	1 西方的园林...
	9	9	史蒂芬·霍金	史蒂芬·霍金生...	45	第一章 我们的...
	10	10	蒋百里	1931年“九一八”...	30	第一篇 国防...
	11	11	马歇尔·卢森堡	著名的马歇尔·...	29	第一章 让爱融...
*	NULL	NULL	NULL	NULL	NULL	NULL

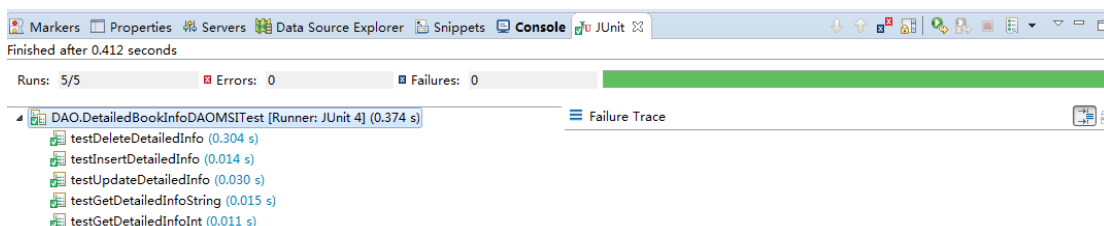
插入目录为aaa的书籍的详细信息，更新ID为2的书籍的详细信息，将其目录设为bbb，将其作者设为CCC，将其主题设为ttt查询ID为2书籍的详细信息，查询图书类别为宇宙学的图书详细信息，删除ID为10的记录信息。

```

9      d.setCatalog("aaaa");
10     d.setContactWay("contactway");
11     d.setDescription("description");
12     d.setISOService("iso");
13     d.setLoaderShape("16X16");
14     d.setPrice(100);
15     d.setPublishingInformation("publishing");
16     d.setSecondAuthor("bbb");
17     d.setTopic("ttt");
18     assertEquals(1,dbi.insertDetailedInfo(d));
19 }
20 @Test
21 public void testUpdateDetailedInfo() { //update the info where id=2 ,set catalog=bbbb and set firstauthor=CCCC
22     DetailedBookInfoDAO dbi=new DetailedBookInfoDAO();
23     DetailedBookInfo d=dbi.getDetailedInfo(2);
24     d.setCatalog("bbbb");
25     d.setAuthor("CCCC");
26     assertEquals(1,dbi.updateDetailedInfo(d));
27 }
28 @Test
29 public void testDeleteDetailedInfo() { //delete the info where id=10
30     DetailedBookInfoDAO dbi=new DetailedBookInfoDAO();
31     assertEquals(1,dbi.deleteDetailedInfo(10));
32 }
33 @Test
34 public void testGetDetailedInfoInt() { //find the info where id=2
35     DetailedBookInfoDAO dbi=new DetailedBookInfoDAO();
36     DetailedBookInfo d=dbi.getDetailedInfo(2);
37     assertEquals(2,d.getId());
38 }
39 @Test
40 public void testGetDetailedInfoString() {
41     DetailedBookInfoDAO dbi=new DetailedBookInfoDAO();
42     List<DetailedBookInfo> ld=dbi.getDetailedInfo("宇宙");
43     assertEquals(9,ld.get(0).getId());
44 }

```

插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：



Id	BookId	FirstAuthor	Description	Price	Catalog	ContactWay	ISOService	LoaderShape	PublishingInfor...	Topic	Si
2	2	CCCC	《平凡的世界...	88	bbbb	QQ:1924835570...	020-38250260	3册 (463, 465, ...	北京:北京十月...	长篇小说-中国...	M
3	3	Nicholas C. Zakas	所属分类: 计...	99	第1章 JavaScr...	QQ:1924835570...	020-38250260	730页:图;24cm	北京:人民邮电...	JAVA语言-程序...	李
4	4	菲茨杰拉德	经典的美国梦...	19.8	第一章 第二章 ...	QQ:1924835570...	020-38250260	186页;21cm	北京:中国华侨...	长篇小说-美国...	成
5	5	余华	《活着》是一...	20	第1章	QQ:1924835570...	020-38250260	17,191页;21cm	北京:作家出版...	长篇小说-中国...	M
6	6	赵书兰	《MATLAB编程...	59	1.1 MATLAB的...	QQ:1924835570...	020-38250260	xii, 445页:图;26cm	北京:电子工业...	Matlab软件-程...	M
7	7	乔斯坦·费德	14岁的少女苏...	26.8	1.伊甸园-在某...	QQ:1924835570...	020-38250260	575页;20cm	北京:作家出版...	哲学史-欧洲通...	萧
8	8	王向荣, 林菁著	《西方现代景...	210	1 西方的园林...	QQ:1924835570...	020-38250260	289页:图;30cm	北京:中国建筑...	景观-环境设计...	M
9	9	史蒂芬·霍金	史蒂芬·霍金生...	45	第一章 我们的...	QQ:1924835570...	020-38250260	245页:图, 肖像...	长沙:湖南科学...	宇宙学	许

DUANWEIWEI6D24.L...DetailedBookInfo											DUANWEIWEI6D24.L... - dbo.BookInfo*
Id	BookId	FirstAuthor	Description	Price	Catalog	ContactWay	ISOService	LoaderShape	PublishingInfor...	Topic	
2	2	CCCC	《平凡的世界...	88	bbbb	QQ:1924835570...	020-38250260	3册 (463, 465, ...	北京:北京十月...	长篇小说-中国...	
3	3	Nicholas C. Zakas	所属分类: 计...	99	第1章 JavaScr...	QQ:1924835570...	020-38250260	730页:图;24cm	北京:人民邮电...	JAVA语言-程序...	
4	4	菲茨杰拉德	经典的美国梦...	19.8	第一章 第二章 ...	QQ:1924835570...	020-38250260	186页;21cm	北京:中国华侨...	长篇小说-美国...	
5	5	余华	《活着》是一...	20	第1章	QQ:1924835570...	020-38250260	17,191页;21cm	北京:作家出版...	长篇小说-中国...	
6	6	赵书兰	《MATLAB编程...	59	1.1 MATLAB的...	QQ:1924835570...	020-38250260	xii, 445页:图;26cm	北京:电子工业...	Matlab软件-程...	
7	7	乔斯坦·费德	14岁的少女苏...	26.8	1.伊甸园-在某...	QQ:1924835570...	020-38250260	575页;20cm	北京:作家出版...	哲学史-欧洲通...	
8	8	王向荣, 林菁著	《西方现代景...	210	1 西方的园林...	QQ:1924835570...	020-38250260	289页:图;30cm	北京:中国建筑...	景观-环境设计...	
11	11	AAA	description	100	aaaa	contactway	iso	16x16	publishing	ttt	

10.对bookType表的增删改查测试

原bookType表:

210A.Library - dbo.BookType				210A.Library - dbo.Detail			
	TypeId	TypeNO	TypeName				
▶	1	I247.5	中国文学 小说 ...				
	2	TP312	工业技术 自动...				
	3	TU-856	工业技术 建筑...				
	4	I533.45	外国哲学, 通...				
	5	I247.5	中国图书, 长...				
	6	I712.45	各国文学				
	7	E115	世界军事 军事...				
	8	P159	天文学、地球...				
	9	C912.1-49	社会学 社会结...				
*	NULL	NULL	NULL				

查询TypeID为1的类别信息, 插入TypeNO为A111.2,类别名为外国文学小说的类别信息, 更新TypeID为1的记录将其编号设为B222,类别名设为俄国文学。

```

8
9 public class BookTypeDAO姆斯ITest {
0
1     @Test
2     public void testGetBookType() {
3         BookTypeDAO姆斯I btdm=new BookTypeDAO姆斯I();
4         BookType bt=btdm.getBookType(1);
5         assertEquals(1,bt.getTypeId());
6     }
7
8     @Test
9     public void testInsertBookType() {
0         BookTypeDAO姆斯I btdm=new BookTypeDAO姆斯I();
1         BookType bt=new BookType();
2         bt.setTypeNO("A111.2");
3         bt.setTypeNO("外国文学. 小说");
4         assertEquals(1,btdm.insertBookType(bt));
5     }
6
7     @Test
8     public void testUpdateBookType() {
9         BookTypeDAO姆斯I btdm=new BookTypeDAO姆斯I();
0         BookType bt=btdm.getBookType(1);
1         bt.setTypeNO("B222");
2         bt.setTypeNO("外国文学");
3         assertEquals(1,btdm.updateBookType(bt));
4     }
5
6     @Test
7     public void testDeleteBookType() {
8         BookTypeDAO姆斯I btdm=new BookTypeDAO姆斯I();
9         assertEquals(1,btdm.deleteBookType(8));
0
1     }
2 }

```

插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：

Markers Properties Servers Data Source Explorer Snippets Console JUnit

Finished after 0.408 seconds

Runs: 4/4 Errors: 0 Failures: 0

DAO.BookTypeDAO姆斯ITest [Runner: JUnit 4] (0.375 s)

- testDeleteBookType (0.310 s)
- testGetBookType (0.019 s)
- testUpdateBookType (0.033 s)
- testInsertBookType (0.013 s)

Failure Trace

DUANWEIWEI6D24....y - dbo.BookType		DUANWEIWEI6D24....y - dbo.BookType	
	TypeId	TypeNO	TypeName
▶	1	B111.3	英国文学
	2	TP312	工业技术 自动化技术、计算机技术 计算机软...
	3	TU-856	工业技术 建筑科学 房屋建筑设计 机电设备 ...
	4	I533.45	外国哲学，通俗读物，欧洲
	5	I247.5	中国图书，长篇小说，中国当代
	6	I712.45	各国文学
	7	E115	世界军事 军事建设与战备 边防、海防、空防
	8	P159	天文学、地球科学 天文学 恒星天文学、星系...

DUANWEIWEI6D24....y - dbo.BookType			
	TypeId	TypeNO	TypeName
▶	1	B222	俄国文学
	2	TP312	工业技术 自动...
	3	TU-856	工业技术 建筑...
	4	I533.45	外国哲学，通...
	5	I247.5	中国图书，长...
	6	I712.45	各国文学
	7	E115	世界军事 军事...
	10	A111.2	外国文学，小说

11.对Roominfo表的增删改查测试

原Roominfo表：

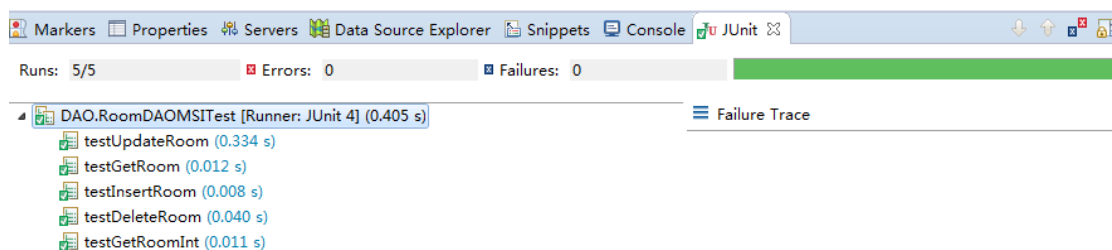
210A.Library - dbo.ReturnInfo				
	Id	BookId	UserId	ReturnTime
▶	1	2	1	2018-08-06
	2	3	3	2018-09-13
	3	5	2	2018-09-20
	4	1	4	2018-10-28
	5	4	5	2018-11-15
*	NULL	NULL	NULL	NULL

插入RoomNo为304的阅览室信息，更新RoomID为1的阅览室信息将其容量设为40，删除RoomID为6的阅览室信息，查询每条阅览室信息。

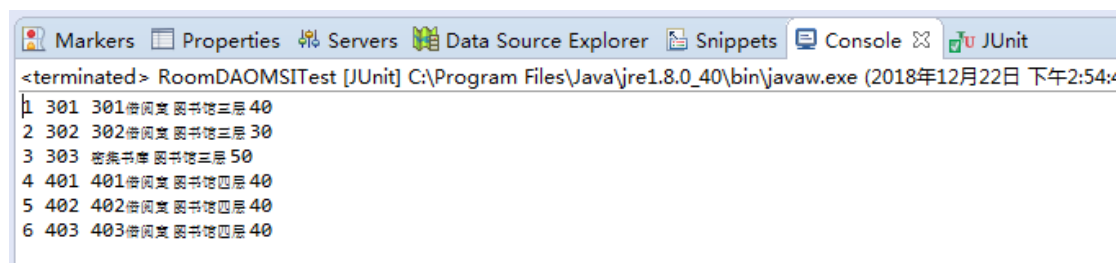
```

12 @Test
13 public void testInsertRoom() {
14     RoomDAO MSI rdm=new RoomDAO MSI();
15     Room r=new Room();
16     r.setRoomNo("304");
17     r.setRoomName("304阅览室");
18     r.setRoomLocation("图书馆三楼");
19     r.setRoomCapacity(40);
20     assertEquals(1,rdm.insertRoom(r));
21 }
22 @Test
23 public void testUpdateRoom() {
24     RoomDAO MSI rdm=new RoomDAO MSI();
25     Room r=rdm.getRoom(1);
26     r.setRoomCapacity(40);
27     assertEquals(1,rdm.updateRoom(r));
28 }
29 @Test
30 public void testDeleteRoom() {
31     RoomDAO MSI rdm=new RoomDAO MSI();
32     assertEquals(1,rdm.deleteRoom(6));
33 }
34 @Test
35 public void testGetRoomInt() {
36     RoomDAO MSI rdm=new RoomDAO MSI();
37     assertEquals(2,rdm.getRoom(2).getRoomId());
38 }
39 @Test
40 public void testGetRoom() {
41     RoomDAO MSI rdm=new RoomDAO MSI();
42     List<Room> lr=rdm.getRoom();
43     for(int i=0;i<lr.size();i++){
44         Room r=lr.get(i);
45         System.out.println(r.getRoomId()+" "+r.getRoomNo()+" "+r.getRoomName()+" "+r.getRoomLocation()+" "+r.getRoomCapaci
46     }
47     assertEquals(6,lr.size());

```



插入成功，更新成功，删除成功，查询成功，数据库截图如下所示：



RoomId	RoomNo	RoomName	RoomLocation	RoomCapacity
1	301	301借阅室	图书馆三层	30
2	302	302借阅室	图书馆三层	30
3	303	密集书库	图书馆三层	50
4	401	401借阅室	图书馆四层	40
5	402	402借阅室	图书馆四层	40
6	403	403借阅室	图书馆四层	40

RoomId	RoomNo	RoomName	RoomLocation	RoomCapacity
1	301	301借阅室	图书馆三层	40
2	302	302借阅室	图书馆三层	30
3	303	密集书库	图书馆三层	50
4	401	401借阅室	图书馆四层	40
5	402	402借阅室	图书馆四层	40
7	304	304借阅室	图书馆三层	40

收获与体验：

通过这次的图书馆管理系统的数据库设计，使我们真正地将数据库的一些设计原理，规范化理论与实际的实践相结合。

在需求设计与概念设计阶段，我们需要充分分析图书馆管理系统涉及的功能，产生的实体，以及实体之间的联系，完成某项功能应该考虑实体的哪些属性等等。

在逻辑设计与物理设计阶段，该用什么作为主键，是自动增长还是自定义类型，以及外键条件的约束，Null/Not Null约束，还有在设置外键的时候我们也遇到了一些问题，比如外键应该设置在哪里，或者外键怎样设置才能实现级联的要求等等。

在数据库实施阶段，我们要考虑各个关系中属性的类型，编号等尽量使用字符串的形式以增加灵活度，以及在测试用例中对于DAO层的编写，DAO层之间的继承与实现的关系，实际上就是封装了对数据库一系列操作的借口，使得用户更加方便地操作数据库。

这次课程设计对我而言意义非凡，不仅锻炼了我对数据库操作的能力，更是培养了一些好的设计数据库的思维逻辑，让我受益匪浅。

Github项目链接：<https://github.com/duanweiwe/DataBase-NancyDww.git>