# NMPMF00Z{V~W9@$H{AW]51T 沈阳工学院校内规划项目报告册

**项目 报 告 册**

**课程名称：**  **Hibernate应用开发**

**专 业： 计算机科学与技术**

**班级学号： 1530110331**

**姓 名： 张东源**

**指导教师： 靳 新**

**2017 ~ 2018 学年 第 一 学期**

**沈阳工学院**

项目报告成绩评定标准

|  |  |  |
| --- | --- | --- |
| 成绩等级 | 评分标准 | 说明 |
| A | 1. 准备充分，掌握项目原理，明确项目任务； 2. 能准确、按时完成项目任务； 3. 能独立解决项目实施中的问题和错误； 4. 项目报告内容完整、书写规范整洁，操作过程完整，实验结果正确； 5. 讨论有一定见解； 6. 遵守项目操作规范，无违章现象发生。 |  |
| B | 1. 准备较充分，理解项目原理，明确项目任务； 2. 在教师的指导下能准确、按时完成项目任务； 3. 能够解决项目实施中的一般问题或错误； 4. 项目报告内容完整、版面整洁，操作过程完整，实验结果基本正确； 5. 报告中有讨论内容； 6. 遵守项目操作规范，无违章现象发生。 |  |
| C | 1. 能进行预习和准备，理解项目原理，明确项目任务； 2. 基本按时完成项目任务； 3. 能发现并解决简单的问题和错误； 4. 项目报告内容基本完整、书写规范整洁，操作过程基本完整，实验结果基本正确； 5. 遵守项目操作规范，无违章现象发生。 |  |
| D | 1. 能进行预习和准备，了解项目原理，明确项目任务； 2. 在教师的指导下基本按时完成多个项目任务； 3. 项目报告内容基本完整、有项目实施过程，书写较规范，实验结果基本正确； 4. 遵守项目操作规范，无违章现象发生。 |  |
| E | 1. 不能完成基本的项目任务； 2. 项目报告内容不完整、项目实施过程不完整，书写不规范，无实验结果等； 3. 实验报告有抄袭情况； 4. 严重违反相关学科实验教学中心规章制度并造成不良后果。 |  |

**在本门课程中，项目报告成绩所占比重：**

预 习 报 告

项目名称：

基于Hibernate的图书管理系统的设计——图书信息管理

预习内容：

1、hibernate检索方式

HQL基础：HQ了是面向对象的查询语言，它和SQL查询语言有些相似。在hibernate提供的各种检索方式，HQL是一种广泛的检索方式。既有如下功能：

（1）、在查询语句中设定各种查询条件。

（2）、支持各种投影查询，即检索出对象的部分属性。

（3）、支持分页查询。

（4）、支持分组查询。允许使用having和group by 关键字。

（5）、提供内置聚集函数，如sum()、min()、max()。

（6）支持子查询，即嵌入式查询。

（7）、支持动态绑定函数。

HQL检索方式方法步骤如下：

（1）、有session对象的createQuery()方法创建一个Query对象，该对象包含一个HQL查询语句，使用占位符。

（2）、有Query对象的setXxx和setParameter()方法参数赋值，其中setXxx()方法的第一个参数表示参数占位符名称，第二个参数表示参数值；setParameter()方法的第一个参数表示参数占位符的位置，0表示第一个？，1表示第二个？。

（3）、有Query对象的list（）方法执行查询语句。该方法返回list类型的结果集，在list集合中存放了符合查询条件的持久化对象。

2、QBC检索方式

QBC是hibernate提供的一个查询接口。采用HQL检索方式时，在应用程序中需要定义基于字符串形式的HQL接口查询语句。QBC提供了检索对象的另一种方式，它主要有criteria接口、critertion接口和restrictions类组成，它支持在运行时动态查询语句。

QBC查询方式步骤如下：

（1）、创建criteria对象。有session对象的createCriteria()方法创建一个criteria对象。

（2）、设定查询条件。一个criterion实例代表一个查询条件，restrictions等类提供了很多用于设定查询条件的方法。

（3）、添加查询条件。由criteria对象的add()方法添加包含查询添加的criterion实例。

（4）、执行查询返回结果。由criteria对象的list()方法执行查询并返回查询结果，在list集合中存放着符合条件的持久化对象

3、hibernatet的事务管理

事务是指一组相互依赖的操作行为，在每个事务结束时，都能够保持数据一致性。事务是指作为单个逻辑工作单元执行的一系列操作。数据库向用户提供保存前程序状态的方法叫事务提交（commit）；当事务执行过程中，使数据忽略当前的状态并回到前面保存状态的方法叫事务回滚（RollBack）。

项 目 报 告

|  |  |
| --- | --- |
| 项目名称 | 基于Hibernate的图书管理系统的设计——图书信息管理 |
| 实验室名称 | 计算机实训室2 |
| 日期 | 2017年12月5日 |

目的与要求（及主要实验仪器、设备）：

1、了解Hibernate基本特性；

2、掌握Hibernate框架的组成；

3、掌握Hibernate中核心接口和类；

4、掌握Hibernate主键生成策略的特性；

5、掌握Hibernate中OID、QBC和HQL等的基本操作语句；

6、掌握Hibernate中一对一、一对多、多对多的关联关系。

7、了解Hibernate缓存和事务处理机制。

本次项目采用的主要设备：自带电脑。

项目实施方法及原理概要分析：

1、HQL检索方式方法步骤如下：

（1）、有session对象的createQuery()方法创建一个Query对象，该对象包含一个HQL查询语句，使用占位符。

（2）、有Query对象的setXxx和setParameter()方法参数赋值，其中setXxx()方法的第一个参数表示参数占位符名称，第二个参数表示参数值；setParameter()方法的第一个参数表示参数占位符的位置，0表示第一个？，1表示第二个？。

（3）、有Query对象的list（）方法执行查询语句。该方法返回list类型的结果集，在list集合中存放了符合查询条件的持久化对象。

项目实施过程与数据：

1. 创建数据库

1.1、 booktype表存放图书类别的信息，包括： booktypeid、booktypename2个字段。如表1.1所示：

表1.1 booktype表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 是否为空 | 说明 |
| booktypeid | int | Not null | 图书类型编号 |
| booktypename | Varchar | Not null | 图书类型姓名 |

1.2、bookinfo表存放图书的信息，包括： bookid、bookname等7个字段。如表1.2所示：

表1.2 bookinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 是否为空 | 说明 |
| bookid | int | Not null | 图书编号 |
| bookname  booktypename  pubname  author  isbn  price | Varchar  Varchar  Varchar  Varchar  Varchar  double | Not null  Not null  Not null  Not null  Not null  Not null | 图书名称  图书类型名称  出版社名称  作者  编码  价格 |

2、关于booktype关键代码如下：

2.1、pojo/booktype.hbm.xml关键代码

<class name=*"pojo.BookType"* table=*"BOOKTYPE"*> <!-- 类对应一个表-->

<id name=*"booktypeid"* column=*"BOOKTYPEID"*>

<generator class=*"assigned"*/> <!-- 主键的生成策略--> </id>

<property name=*"booktypename"* column=*"BOOKTYPENAME"* /></class>

2.2、hibernate.cfg.xml关键代码

<mapping resource=*"pojo/BookType.hbm.xml"* />

2.3、booktypedao关键代码

（1）、查询全部：public List<BookType> findAllBookType() throws Exception;

（2）、根据uname查询当前类别信息：public BookType queryOneBookType(int booktypeid) throws Exception;

（3）、修改：public boolean updateBookType(BookType booktype) throws Exception;

（4）、删除：public boolean deleteBookType(int booktypeid) throws Exception;

（5）、添加：public boolean addBookType(BookType booktype) throws Exception;

2.4、booktypeDAoImpl关键代码

private Session session = null;

private Transaction tran = null;

（1）、构造方法，得到事务对象

public BookTypeDAOImpl(){

session = HibernateSessionFactory.*getSession*();

tran = session.beginTransaction(); }

（2）、查询

String sql = "from BookType";

Query query = session.createQuery(sql);

List<BookType> list = query.list();

tran.commit();return list;

（3）、根据id查询修改文件

BookType booktype = (BookType) session.get(BookType.class, booktypeid);

tran.commit();return booktype;

（4）、修改

String sql = "update BookType set booktypename = ? where booktypeid = ?";

Query query = session.createQuery(sql);

query.setString(0, booktype.getBooktypename());

query.setInteger(1, booktype.getBooktypeid());

int i = query.executeUpdate();

tran.commit();return true;

（5）、删除

BookType booktype = (BookType) session.get(BookType.class, booktypeid);

session.delete(booktype);tran.commit(); return true;

（6）、添加

booktype.setBooktypename(booktype.getBooktypename());

booktype.setBooktypeid(booktype.getBooktypeid());

session.save(booktype);tran.commit();return true;

3、关于bookinfo关键代码

3.1、pojo/bookinfo.Hbm.xml关键代码

<class name=*"pojo.BookInfo"* table=*"BOOKINFO"*> <!-- 类对应一个表-->

<id name=*"bookid"* column=*"BOOKID"*>

<generator class=*"assigned"*/> <!-- 主键的生成策略--></id>

<property name=*"bookname"* column=*"BOOKNAME"* />

<property name=*"author"* column=*"AUTHOR"* />

<property name=*"isbn"* column=*"ISBN"* />

<property name=*"price"* column=*"PRICE"* />

<property name=*"pubname"* column=*"PUBNAME"* />

<property name=*"booktypename"* column=*"BOOKTYPENAME"* /></class>

3.2、hibernate.cfg.xml关键代码

<mapping resource=*"pojo/BookInfo.hbm.xml"* />

3.3、bookinfoDAO关键代码

（1）、查询全部：public List<BookInfo> findAllBookInfo() throws Exception;

（2）、根据id查询修改用户：public BookInfo queryOneBookInfo(int bookid) throws Exception;

（3）、修改用户：public boolean updateBookInfo(BookInfo bookinfo) throws Exception;

（4）、删除：public boolean deleteBookInfo(int bookid) throws Exception;

（5）、添加：public boolean addBookInfo(BookInfo bookinfo) throws Exception;

3.4、bookinfoDAOImpl关键代码

private Session session = null;private Transaction tran = null;

（1）、构造方法，得到事务对象

public BookInfoDAOImpl(){

session = HibernateSessionFactory.*getSession*();

tran = session.beginTransaction(); }

（2）、查询

String sql = "from BookInfo";

Query query = session.createQuery(sql);

List<BookInfo> list = query.list();tran.commit();return list;

（3）、删除

BookInfo bookinfo = (BookInfo)session.get(BookInfo.class, bookid);

session.delete(bookinfo);tran.commit(); return true;

（4）、根据id查询当前修改用户的信息

BookInfo bookinfo = (BookInfo)session.get(BookInfo.class, bookid);

tran.commit();return bookinfo;

（5）、对要修改的用户进行修改

String sql = "update BookInfo set bookname = ?,author=?,pubname=?,booktypename=?,isbn=?,price=? where bookid = ?";

Query query = session.createQuery(sql);

query.setString(0, bookinfo.getBookname());

query.setString(1, bookinfo.getAuthor());

query.setString(2, bookinfo.getPubname());

query.setString(3, bookinfo.getBooktypename());

query.setString(4, bookinfo.getIsbn());

query.setDouble(5, bookinfo.getPrice());

query.setInteger(6, bookinfo.getBookid());

int i = query.executeUpdate();

tran.commit();return true;

（6）、添加

bookinfo.setBookname(bookinfo.getBookname());

bookinfo.setBookid(bookinfo.getBookid());

bookinfo.setAuthor(bookinfo.getAuthor());

bookinfo.setBooktypename(bookinfo.getBooktypename());

bookinfo.setIsbn(bookinfo.getIsbn());

bookinfo.setPrice(bookinfo.getPrice());

bookinfo.setPubname(bookinfo.getPubname());

session.save(bookinfo);tran.commit();return true;

项目实施结果：

1. booktype实验结果图如下：

1.1、点击图书管理-图书类型，即可查询所有图书类型的信息。结果图如图1.1所示：

图1.1

1.2、选择要删除的图书类型，点击删除按钮，即可删除信息。结果图如图1.2所示：

图1.2

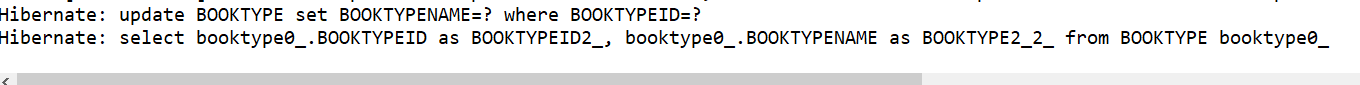
1.3、选择修改的图书类型信息，点击修改按钮，即可修改图书类型的类型编号、类型名称，点击保存按钮，即可完成信息修改。效果图如图1.3所示:

图1.3

1.4、点击添加图书类型信息，进入添加界面，即可添加图书类型的信息。效果图如图1.4所示：

图1.4

2、bookinfo实验结果如下：

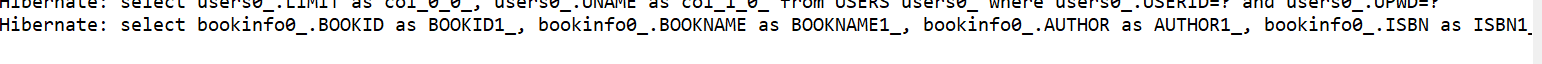
2.1、点击图书管理-图书档案，即可查询所有图书的信息。结果图如图1.5所示：

图1.5

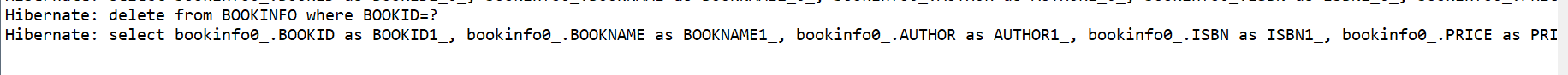
2.2、选择要删除的图书信息，点击删除按钮，即可删除图书信息。结果图如图1.6所示：

图1.6

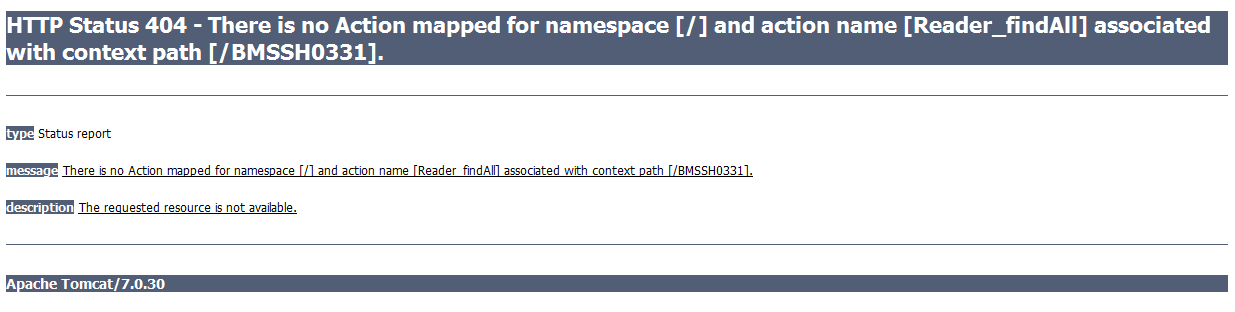
2.3、选择修改的图书信息，点击修改按钮，即可修改图书的编号、名称等7个类别，点击保存按钮，即可完成信息修改。效果图如图1.7所示:

图1.7

2.4、点击添加图书信息，进入添加界面，即可添加图书的信息。效果图如图1.8所示：

图1.8

思考与讨论：

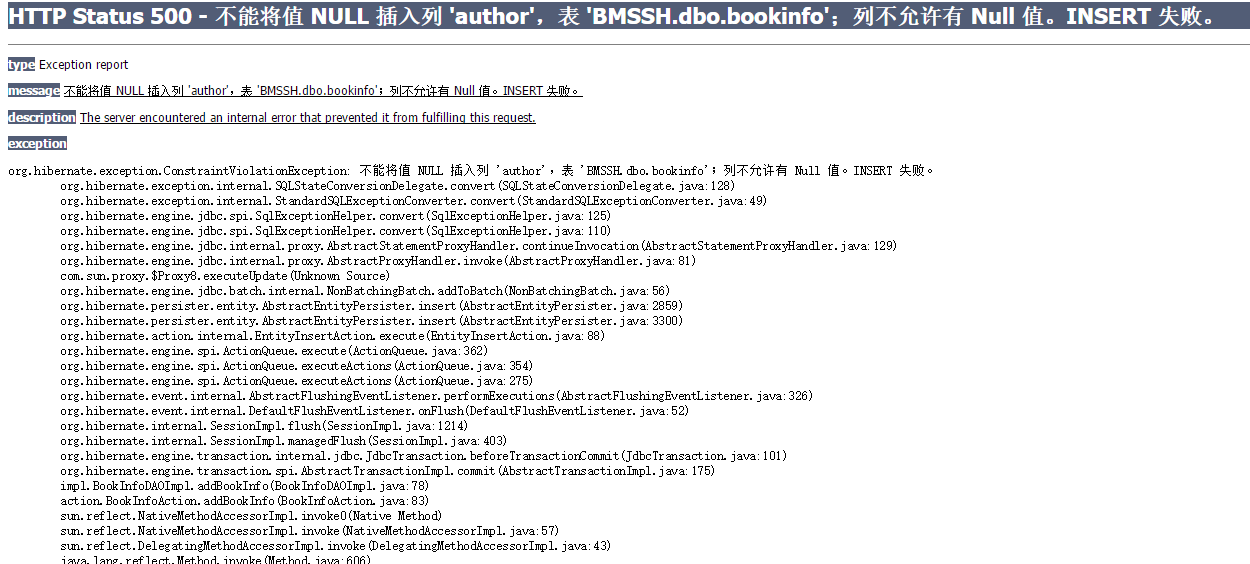
 1、查询出现404错误：

（1）、首先检查menu.JS中a href响应的action是否正确；

（2）、struts中method方法与dao方法是否相同；

（3）、jsp查询界面标签s:property value的值是否正确；

2、添加时出现不能讲空值插入列中：

C:\Users\asus\AppData\Local\Temp\enhtmlclip\1.PNGJSP中标签的s:textfield name属性是否正确；

|  |  |
| --- | --- |
| 项目成绩 |  |
| 批阅教师 |  |
| 批阅日期 | 年 月 日 |