**桂林航天工业学院学生实验报告**

**实验十一、十二**

https://github.com/ALI1416/Library-Management-System/

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | 数据库编程 | 实验日期 | **2019.5.7/9/14/16/21** |
| 实验目的 | 在安装有SQL SERVER2008和.net 2010以上版本的环境中，实现对数据库的数据进行增加、删除、修改、查询操作 | | |
| 实验内容 | 编程实现图书借阅信息管理系统。系统有系统管理、读者信息管理、图书信息管理、借阅信息管理模块。系统管理有用户管理和密码管理2个窗体；读者信息管理有读者信息增加、读者信息删除、读者信息修改和读者信息查询功能4个窗体；图书信息管理有图书信息增加、删除、修改和图书信息查询2个窗体；借阅信息管理有借阅信息增删改查1个窗体。 | | |
| **实验方法及步骤**  **需求分析：**  读者信息管理 类名：Reader   1. 读者信息增加 方法名：addReader  * 输入读者姓名、性别、出生年、身份号码、联系方式、地址、id（自增）   表名：reader  字段：name, gender, year, id\_card, tel, addr, id   1. 读者信息修改、删除 方法名：changeReader, delReader   查询读者信息后   * 可以修改读者信息，除了id * 也可以删除该读者  1. 读者信息查询 方法名：reader  * ~~可以根据读者信息的id、姓名、身份证号码筛选读者~~ * 也可以查询全部的读者   图书信息管理 类名：Book   1. 图书信息增加、删除、修改 方法名：addBook, belBook, changeBook  * 可以添加图书，需要输入图书的书名、作者、出版社、ISBN编号、数量、剩余数量、id（自增）   表名：book 字段：name, author, publishing, isbn, count, remain, id  查询图书信息后   * + 可以修改图书信息，除了id   + 也可以删除图书  1. 图书信息查询 方法名：book  * ~~可以根据图书的id、书名、作者、出版社、ISBN编号、数量筛选图书~~ * 也可以查询全部的图书   借阅信息管理 类名：Borrow   1. 借阅信息增加、删除、修改、查询   方法名：addBorrow, delBorrow, changeBorrow, borrow   * 可以增加借阅信息，需要输入读者id、图书id、借阅时间、是否归还、id   表名：borrow 字段：reader\_id, book\_id, borrow\_time, return\_book   * ~~可以根据读者的id、姓名、身份证号码和图书的id、书名、作者、出版社、ISBN编号、数量（范围）和借阅信息的借阅时间（范围）、是否归还来筛选借阅信息~~ * 也可以查询全部借阅信息   查询借阅信息后   * 可以修改是否归还 * 也可以删除该借阅信息   系统管理 类名：Admin   1. 用户管理（仅限root用户）  * 可以增加管理员 方法名：addAdmin   需要输入用户名、密码、id（自增）  表名：admin 字段：user, pwd, id   * 可以查询所有管理员的账号密码 方法名：admin   查询后   * + 可以删除管理员 方法名：delAdmin   需要输入要删除用户名   * + 也可以修改管理员信息 方法名：changeAdmin   可以修改管理员的用户名   1. 修改密码 方法名：changePwd  * 需要输入用户名、旧密码、新密码   数据库创建：   |  |  |  |  | | --- | --- | --- | --- | | 表名 | 字段名 | 类型 | 约束 | | reader | id | int unsigned | PRIMARY AUTO\_INCREMENT | | name | varchar(16) |  | | gender | tinyint unsigned |  | | year | int unsigned |  | | id\_card | char(18) |  | | tel | char(11) |  | | addr | varchar(256) |  | | book | id | int unsigned | PRIMARY AUTO\_INCREMENT | | name | varchar(128) |  | | author | varchar(16) |  | | publishing | varchar(16) |  | | isbn | char(13) |  | | count | int unsigned |  | | remain | int unsigned |  | | borrow | id | int unsigned | PRIMARY AUTO\_INCREMENT | | reader\_id | int unsigned | FOREIGN reader.id | | book\_id | int unsigned | FOREIGN book.id | | borrow\_time | datetime |  | | return\_book | tinyint unsigned |  | | admin | id | int unsigned | PRIMARY AUTO\_INCREMENT | | user | varchar(16) | UNIQUE | | pwd | varchar(16) |  |   SQL语句：  CREATE DATABASE ck go  USE ck go  CREATE TABLE admin (  id int NOT NULL identity,  user1 varchar(16) UNIQUE NOT NULL,  pwd varchar(16) NOT NULL,  PRIMARY KEY (id)  ) go  CREATE TABLE book (  id int NOT NULL identity,  name varchar(128) NOT NULL,  author varchar(16) NOT NULL,  publishing varchar(16) NOT NULL,  isbn char(13) NOT NULL,  count1 int NOT NULL,  remain int NOT NULL,  PRIMARY KEY (id)  ) go  CREATE TABLE reader (  id int NOT NULL identity,  name varchar(16) NOT NULL,  gender tinyint NOT NULL,  year1 int NOT NULL,  id\_card char(18) NOT NULL,  tel char(11) NOT NULL,  addr varchar(128) NOT NULL,  PRIMARY KEY (id)  ) go  CREATE TABLE borrow (  id int NOT NULL identity,  reader\_id int NOT NULL,  book\_id int NOT NULL,  borrow\_time datetime NOT NULL,  return\_borrow tinyint NOT NULL DEFAULT '0',  PRIMARY KEY (id),  FOREIGN KEY (reader\_id) REFERENCES reader (id),  FOREIGN KEY (book\_id) REFERENCES book (id)  ) go  INSERT INTO admin(user1,pwd) VALUES ('root', 'root');  INSERT INTO admin(user1,pwd) VALUES ('ck', 'chengkai');  INSERT INTO admin(user1,pwd) VALUES ('txf', 'txf');  INSERT INTO book(name,author,publishing,isbn,count1,remain) VALUES ('计算机网络', '谢希仁', '电子工业出版社', '9787121302954', '10', '10');  INSERT INTO book(name,author,publishing,isbn,count1,remain) VALUES ('SqlServer2008数据库应用技术', '刘卫国、刘泽星', '人民邮电出版社', '9787115377302', '25', '24');  INSERT INTO book(name,author,publishing,isbn,count1,remain) VALUES ('JavaEE企业级应用开发教程', '黑马程序员', '人民邮电出版社', '9787115461025', '17', '16');  INSERT INTO book(name,author,publishing,isbn,count1,remain) VALUES ('Android应用开发教程', '钟元生、高成珍', '江西高校出版社', '9787549317066', '13', '12');  INSERT INTO reader(name,gender,year1,id\_card,tel,addr) VALUES ('ck', '1', '1998', '370811199805250000', '18807700000', '山东济宁');  INSERT INTO reader(name,gender,year1,id\_card,tel,addr) VALUES ('txf', '0', '1996', '451000000000000000', '18888888888', '广西玉林');  INSERT INTO reader(name,gender,year1,id\_card,tel,addr) VALUES ('zk', '1', '1995', '510000000000000000', '17700000000', '四川绵阳');  INSERT INTO reader(name,gender,year1,id\_card,tel,addr) VALUES ('hhd', '0', '1996', '310000000000000000', '11100000000', '福建漳州');  INSERT INTO borrow(reader\_id,book\_id,borrow\_time,return\_borrow) VALUES ('1', '1', '2019-05-08 11:03:35', '1');  INSERT INTO borrow(reader\_id,book\_id,borrow\_time,return\_borrow) VALUES ('2', '2', '2019-05-08 11:04:36', '0');  INSERT INTO borrow(reader\_id,book\_id,borrow\_time,return\_borrow) VALUES ('1', '3', '2019-05-08 11:11:24', '0');  INSERT INTO borrow(reader\_id,book\_id,borrow\_time,return\_borrow) VALUES ('1', '4', '2019-05-08 11:12:10', '0');  go  create trigger book\_borrow on borrow for insert as  declare @bookId int  select @bookId=book\_id from inserted  update book set remain=remain-1 where id=@bookId  go  create trigger book\_return on borrow for update as  declare @newRB int, @oldRB int, @bookId int  select @newRB=return\_borrow from inserted  select @oldRB=return\_borrow from deleted  select @bookId=book\_id from deleted  if(@newRB!=@oldRB)  begin  if(@newRB=1)  begin  update book set remain=remain+1 where id=@bookId  end  if(@newRB=0)  begin  update book set remain=remain-1 where id=@bookId  end  end  **页面错误信息：**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 页面 | 错误程度 | 错误码 | 错误标题 | 错误信息 | | login | warning | loginErr | 登录失败 | 账号或密码错误 | | changePwd | success | pwdOk | 修改成功 | 密码修改成功 | | warning | pwdErr | 修改失败 | 密码错误 | | admin | danger | delForbid | 无法删除 | 禁止删除本用户 | | success | delOk | 删除成功 | 用户已删除 | | warning | delErr | 删除失败 | 用户不存在 | | danger | changeForbid | 无法修改 | 已存在相同用户名 | | success | changeOk | 修改成功 | 用户已修改 | | warning | changeErr | 修改失败 | 用户不存在或未做修改 | | danger | addForbid | 无法添加 | 已存在相同用户名 | | success | addOk | 添加成功 | 添加成功 | | warning | addErr | 添加失败 | 未知错误，请重新添加 | | book | danger | delForbid | 无法删除 | 不可删除！请先删除所有借该书的记录 | | success | delOk | 删除成功 | 图书已删除 | | warning | delErr | 删除失败 | 图书不存在 | | danger | changeForbid | 无法修改 | 输入格式错误 | | success | changeOk | 修改成功 | 图书已修改 | | warning | changeErr | 修改失败 | 图书不存在或信息未做修改 | | danger | addForbid | 无法添加 | 输入格式错误 | | success | addOk | 添加成功 | 图书已添加 | | warning | addErr | 添加失败 | 未知错误，请重新添加 | | reader | danger | delForbid | 无法删除 | 不可删除！请先删除所有借书记录 | | success | delOk | 删除成功 | 读者已删除 | | warning | delErr | 删除失败 | 读者不存在 | | danger | changeForbid | 无法修改 | 输入格式错误 | | success | changeOk | 修改成功 | 读者已修改 | | warning | changeErr | 修改失败 | 读者不存在或信息未做修改 | | danger | addForbid | 无法添加 | 输入格式错误 | | success | addOk | 添加成功 | 读者已添加 | | warning | addErr | 添加失败 | 未知错误，请重新添加 | | borrow | danger | delForbid | 待定 |  | | success | delOk | 删除成功 | 借阅信息已删除 | | warning | delErr | 删除失败 | 借阅信息不存在 | | danger | changeForbid | 待定 |  | | success | changeOk | 修改成功 | 借阅信息已修改 | | warning | changeErr | 修改失败 | 借阅信息不存在或信息未做修改 | | danger | addForbid | 无法添加 | 不存在该读者或该图书或该书已借完 | | success | addOk | 添加成功 | 借阅信息已添加 | | warning | addErr | 添加失败 | 未知错误，请重新添加 |   **关键代码：**  **AdminDao.xml**  <mapper namespace="com.ck.dao.AdminDao">  <select id="login" parameterType="Admin" resultType="Integer">  SELECT id FROM admin WHERE user = #{user}AND pwd = #{pwd}  </select>  <select id="findAdmin" resultType="Admin">  SELECT id,user,pwd FROM admin  </select>  <select id="findAdminById" parameterType="Integer" resultType="Admin">  SELECT id, user, pwd FROM admin WHERE id = #{id}  </select>  <select id="findAdminByIdList" parameterType="List" resultType="Admin">  SELECT id, user, pwd FROM admin WHERE id in  <foreach item="id" index="index" collection="list" open="("  separator="," close=")">  #{id}  </foreach>  </select>  <insert id="addAdmin" parameterType="Admin" keyProperty="id"  useGeneratedKeys="true"> INSERT admin (user, pwd) VALUES (#{user}, #{pwd})  </insert>  <delete id="delAdmin" parameterType="Integer">  DELETE FROM admin WHERE id = #{id}  </delete>  <delete id="delAdminList" parameterType="List">  DELETE FROM admin WHERE id in  <foreach item="id" index="index" collection="list" open="("  separator="," close=")">  #{id}  </foreach>  </delete>  <update id="changeAdmin" parameterType="Admin">  UPDATE admin  <set>  <if test="id != null and id != ''">  id=#{id}, </if>  <if test="pwd != null and pwd != ''">  pwd=#{pwd}, </if>  </set>  WHERE id = #{id}  </update>  </mapper>  **BookDao.xml**  <mapper namespace="com.ck.dao.BookDao">  <select id="findBook" resultType="Book">  SELECT id, name, author, publishing, isbn, count, remain FROM book  </select>  <select id="findBookById" parameterType="Integer" resultType="Book">  SELECT id, name, author, publishing, isbn, count, remain FROM book WHERE id = #{id}  </select>  <select id="findBookByIdList" parameterType="List" resultType="Book">  SELECT id, name, author, publishing, isbn, count, remain FROM book WHERE id in  <foreach item="id" index="index" collection="list" open="("  separator="," close=")">  #{id}  </foreach>  </select>  <insert id="addBook" parameterType="Book" keyProperty="id"  useGeneratedKeys="true">  INSERT book (name, author, publishing, isbn, count, remain )VALUES(#{name}, #{author}, #{publishing}, #{isbn}, #{count}, #{remain})  </insert>  <delete id="delBook" parameterType="Integer">  DELETE FROM book WHERE id = #{value}  </delete>  <delete id="delBookList" parameterType="List">  DELETE FROM book WHERE id in  <foreach item="id" index="index" collection="list" open="("  separator="," close=")">  #{id}  </foreach>  </delete>  <update id="changeBook" parameterType="Book">  UPDATE book  <set>  <if test="name != null and name != ''">  name=#{name}, </if>  <if test="author != null and author != ''">  author=#{author}, </if>  <if test="publishing != null and publishing != ''">  publishing=#{publishing}, </if>  <if test="isbn != null and isbn != ''">  isbn=#{isbn}, </if>  <if test="count != null and count != '' or count == 0">  count=#{count}, </if>  <if test="remain != null and remain != '' or remain == 0">  remain=#{remain}, </if>  </set>  WHERE id = #{id}  </update>  </mapper>  **BorrowDao.xml**  <mapper namespace="com.ck.dao.BorrowDao">  <select id="findBorrow" resultType="Borrow">  SELECT borrow.id, borrow.reader\_id AS readerId, reader.name AS readerName, borrow.book\_id AS bookId, book.name AS bookName, borrow.borrow\_time AS borrowTime, borrow.return\_borrow AS returnBorrow FROM borrow, reader, book WHERE borrow.reader\_id = reader.id AND borrow.book\_id = book.id  </select>  <select id="findBorrowById" parameterType="Integer" resultType="Borrow">  SELECT borrow.id, borrow.reader\_id AS readerId, reader.name AS readerName, borrow.book\_id AS bookId, book.name AS bookName, borrow.borrow\_time AS borrowTime, borrow.return\_borrow AS returnBorrow FROM borrow, reader, book WHERE borrow.reader\_id = reader.id AND borrow.book\_id = book.id AND borrow.id = #{id}  </select>  <select id="findBorrowByIdList" parameterType="List" resultType="Borrow">  SELECT id, reader\_id AS readerId, book\_id AS bookId, borrow\_time AS borrowTime, return\_borrow AS returnBorrow FROM borrow WHERE id in  <foreach item="id" index="index" collection="list" open="("  separator="," close=")">  #{id}  </foreach>  </select>  <insert id="addBorrow" parameterType="Borrow" keyProperty="id"  useGeneratedKeys="true">  INSERT borrow (reader\_id, book\_id, borrow\_time)VALUES(#{readerId}, #{bookId}, now())  </insert>  <delete id="delBorrow" parameterType="Integer">  DELETE FROM borrow WHERE id = #{id}  </delete>  <delete id="delBorrowList" parameterType="List">  DELETE FROM borrow WHERE id in  <foreach item="id" index="index" collection="list" open="("  separator="," close=")">  #{id}  </foreach>  </delete>  <update id="changeBorrow" parameterType="Borrow">  UPDATE borrow  <set>  <if test="readerId != null and readerId != ''">  reader\_id=#{readerId}, </if>  <if test="bookId != null and bookId != ''">  book\_id=#{bookId}, </if>  <if test="borrowTime != null and borrowTime != ''">  borrow\_time=#{borrowTime}, </if>  <if  test="returnBorrow != null and returnBorrow != '' or returnBorrow == 0">  <!-- 注意！0会判断成空值 -->  return\_borrow=#{returnBorrow}, </if>  </set>  WHERE id = #{id}  </update>  </mapper>  **ReaderDao.xml**  <mapper namespace="com.ck.dao.ReaderDao">  <select id="findReader" resultType="Reader">  SELECT id, name, gender, year, id\_card AS idCard, tel, addr FROM reader  </select>  <select id="findReaderById" parameterType="Integer" resultType="Reader">  SELECT id, name, gender, year, id\_card AS idCard, tel, addr FROM reader WHERE id = #{id}  </select>  <select id="findReaderByIdList" parameterType="List" resultType="Reader">  SELECT id, name, gender, year, id\_card AS idCard, tel, addr FROM reader WHERE id in  <foreach item="id" index="index" collection="list" open="("  separator="," close=")">  #{id}  </foreach>  </select>  <insert id="addReader" parameterType="Reader" keyProperty="id"  useGeneratedKeys="true">  INSERT reader (name, gender, year, id\_card, tel, addr)VALUES(#{name}, #{gender}, #{year}, #{idCard}, #{tel}, #{addr})  </insert>  <delete id="delReader" parameterType="Integer">  DELETE FROM reader WHERE id = #{id}  </delete>  <delete id="delReaderList" parameterType="List">  DELETE FROM reader WHERE id in  <foreach item="id" index="index" collection="list" open="("  separator="," close=")">  #{id}  </foreach>  </delete>  <update id="changeReader" parameterType="Reader">  UPDATE reader  <set>  <if test="name != null and name != ''">  name=#{name}, </if>  <if test="gender != null and gender != '' or gender == 0">  gender=#{gender}, </if>  <if test="year != null and year != '' or year == 0">  year=#{year}, </if>  <if test="idCard != null and idCard != ''">  id\_card=#{idCard}, </if>  <if test="tel != null and tel != ''">  tel=#{tel}, </if>  <if test="addr != null and addr != ''">  addr=#{addr}, </if>  </set>  WHERE id = #{id}  </update>  </mapper>  **运行截图：**    输入错误的密码，登录失败。再次输入正确的账号密码    登录成功后，进入图书管理页面，进行添加图书信息    图书添加成功，上方显示绿色提示框    在输入框内直接修改信息，点击修改按钮    修改成功 删除1号图书，无法删除，因为有外键约束 删除5号图书，删除成功    新增读者信息    添加成功 删除1号读者，删除失败，因为有外键约束 删除5号读者    直接对输入框内信息进行修改，点击修改按钮    修改成功    进行借阅    借阅成功    可以修改用户是否已还书    修改成功 删除4号读者借阅信息    新增管理员用户    添加成功    修改账号和密码    修改成功 删除4号管理员用户    非id为1的用户(root)，只可以修改密码，其他功能相同    原密码错误，修改失败 原密码正确，修改成功  点击右上角退出登录后，进入登录页面 | | | |