# 实验五

## 一、相关知识点

1. JDBC基本概念
2. JDBC简单查询、连接查询、嵌套查询、集函数查询等

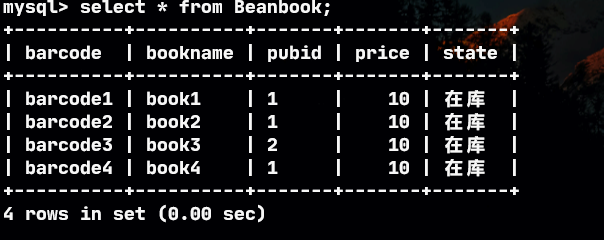
## 二、实验目的

理解Statement对象、ResultSet对象。

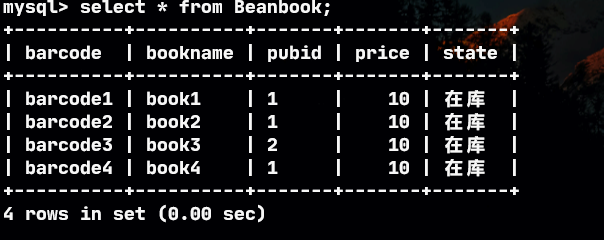
## 三、实验内容

### 1、在booklib工程的BookManager类中增加如下函数（要求采用Statement完成相关查询），并在main函数中进行测试，在实验报告中将代码补上:

public int getBookCount(String pubid) throws BaseException{  
 //要求返回该出版社的图书数量  
 int result = 0;  
 Connection conn = null;  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select \* from beanpublisher where pubid=?";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 pst.setString(1, pubid);  
 java.sql.ResultSet rs = pst.executeQuery();  
 if (!rs.next()) throw new BusinessException("出版社不存在");  
 sql = "select count(1) from beanbook where pubid = ?" +   
 "group by pubid";  
 pst = conn.prepareStatement(sql);  
 pst.setString(1, pubid);  
 rs = pst.executeQuery();  
 if (rs.next()) result = rs.getInt(1);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
 return result;  
}

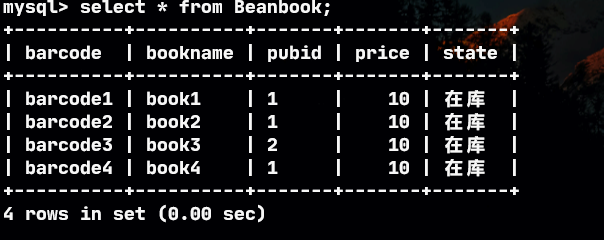


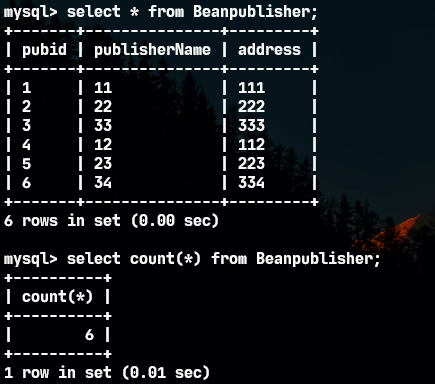
public int getPublisherCount() throws BaseException{  
 //要求返回图书表中出现过的出版社数量  
 int result = 0;  
 Connection conn = null;  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select count(distinct pubid) from beanbook";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 java.sql.ResultSet rs = pst.executeQuery();  
 if(rs.next()) result = rs.getInt(1);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
 return result;  
}





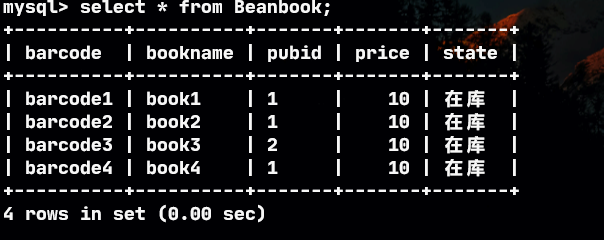
public int getNoneBookPublisherCount()throws BaseException{  
 //要求返回没有图书的出版社数量  
 int result = 0;  
 Connection conn = null;  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select count(\*) from beanpublisher";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 java.sql.ResultSet rs = pst.executeQuery();  
 if(rs.next()) result = rs.getInt(1);  
 result -= (new BookManager()).getPublisherCount();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
 return result;  
}

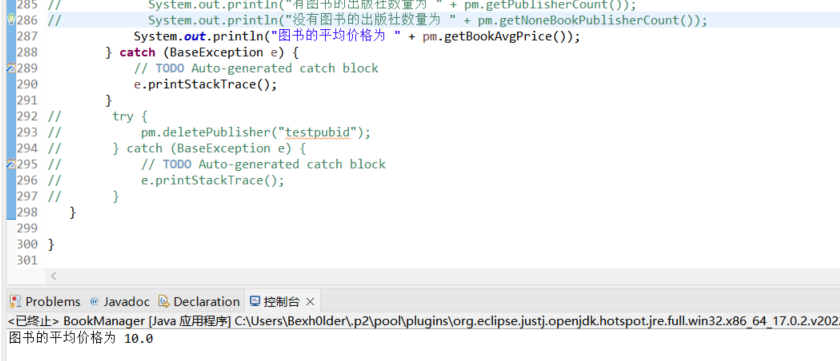






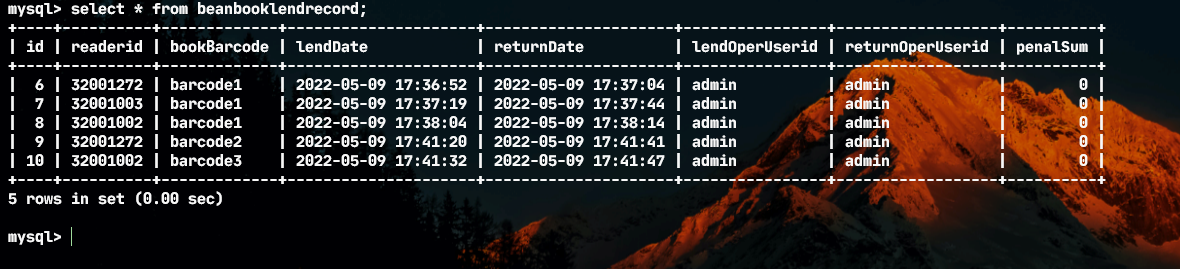
public double getBookAvgPrice()throws BaseException{  
 //要求返回图书的评价价格  
 double result = 0;  
 Connection conn = null;  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select avg(price) from Beanbook";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 java.sql.ResultSet rs = pst.executeQuery();  
 if(rs.next()) result = rs.getDouble(1);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
 return result;  
}





### 2、在booklib工程的BookLendManager类中增加如下函数，并在main函数中进行测试，在实验报告中将代码补上:

public String loadBookLendOperator(String barcode)throws BaseException{  
 //参数为图书条码，返回这本图书最近一次被借出时的操作员姓名，要求采用连接查询实现。难点：如何识别出最近一次？假设不允许用mysql的limit关键字，也不能用嵌套查询，应该如何完成？  
 BeanBook book = (new BookManager()).loadBook(barcode);  
 if (book == null) throw new BusinessException("图书不存在");  
 String result = "";  
 Connection conn = null;  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select \* from BeanBookLendRecord where bookBarcode=?";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 pst.setString(1, barcode);  
 java.sql.ResultSet rs = pst.executeQuery();  
 if (!rs.next()) {  
 throw new BusinessException("该图书没有借阅记录");  
 }  
 sql = "select username,max(lendDate)" +  
 " from beanbooklendrecord" +  
 " left outer join beansystemuser on lendOperUserid = userid" +  
 " where bookBarcode=?";  
 sql += " group by bookBarcode";  
 pst = conn.prepareStatement(sql);  
 pst.setString(1, barcode);  
 rs = pst.executeQuery();  
 if(rs.next()) result = rs.getString(1);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
 return result;  
}





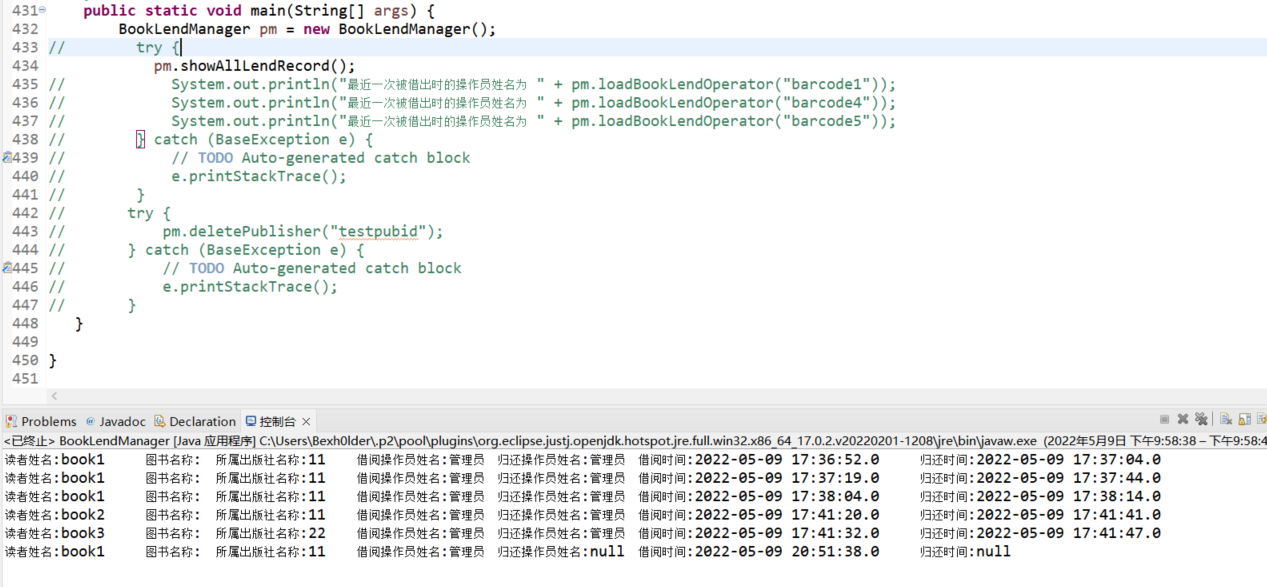
### 3、完成题2中的功能，要求采用嵌套查询实现。

public String loadBookLendOperator(String barcode)throws BaseException{  
 //参数为图书条码，返回这本图书最近一次被借出时的操作员姓名，要求采用连接查询实现。难点：如何识别出最近一次？假设不允许用mysql的limit关键字，也不能用嵌套查询，应该如何完成？  
 BeanBook book = (new BookManager()).loadBook(barcode);  
 if (book == null) throw new BusinessException("图书不存在");  
 String result = "";  
 Connection conn = null;  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select \* from BeanBookLendRecord where bookBarcode=?";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 pst.setString(1, barcode);  
 java.sql.ResultSet rs = pst.executeQuery();  
 if (!rs.next()) {  
 throw new BusinessException("该图书没有借阅记录");  
 }  
 //连接查询  
// sql = "select username,max(lendDate)" +  
// " from beanbooklendrecord" +  
// " left outer join beansystemuser on lendOperUserid = userid" +  
// " where bookBarcode=?";  
// sql += " group by bookBarcode";  
 //嵌套查询  
 sql = "select username from beansystemuser where userid in (" +   
 "select lendOperUserid from beanbooklendrecord where lendDate in (" +  
 "select max(lendDate) from beanbooklendrecord where bookBarcode = ? group by bookBarcode))";  
 pst = conn.prepareStatement(sql);  
 pst.setString(1, barcode);  
 rs = pst.executeQuery();  
 if(rs.next()) result = rs.getString(1);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
 return result;  
}



### 4、在booklib工程的BookLendManager类中增加如下函数，并在main函数中进行测试，在实验报告中将代码补上:

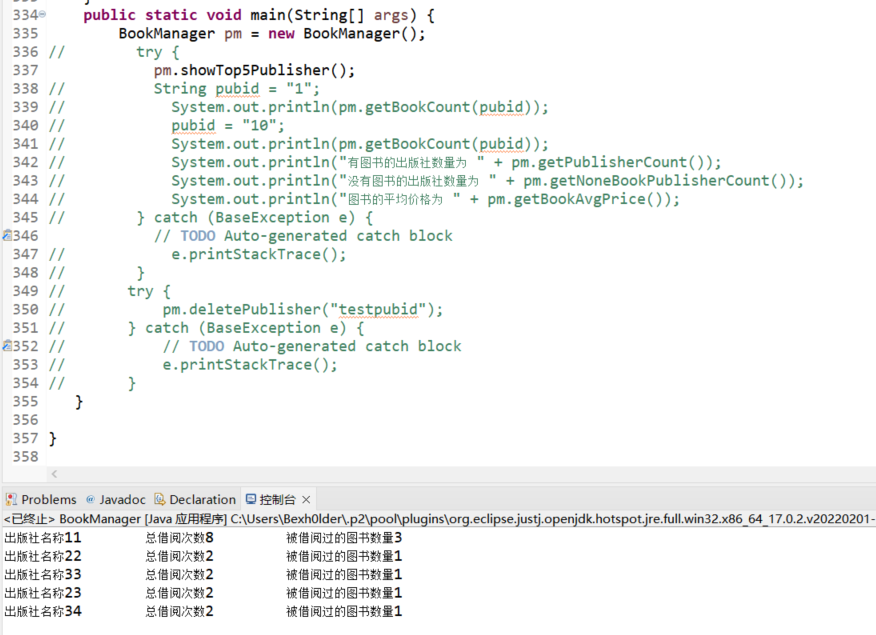
public void showAllLendRecord(){  
 //通过System.out.println方法，输出所有借阅记录的明细数据，要求结果中包括读者姓名、图书名称、所属出版社名称、借阅操作员姓名、归还操作员姓名、借阅时间、归还时间等  
 //注意：需要注意未归还图书的情况  
 Connection conn = null;  
 try {  
 List<BeanBookLendRecord> result = new ArrayList<BeanBookLendRecord>();  
 conn = DBUtil.getConnection();  
 String sql = "select \* from BeanBookLendRecord";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 java.sql.ResultSet rs = pst.executeQuery();  
 while(rs.next()) {  
 String readerName = "";  
 String bookName = "";  
 String publisherName = "";  
 String lendOperUserName = null;  
 String returnOperUserName = null;  
 BeanBookLendRecord r = new BeanBookLendRecord();  
 r.setId(rs.getInt(1));  
 r.setReaderid(rs.getString(2));  
 r.setBookBarcode(rs.getString(3));  
 r.setLendDate(rs.getTimestamp(4));  
 r.setReturnDate(rs.getTimestamp(5));  
 r.setLendOperUserid(rs.getString(6));  
 r.setReturnOperUserid(rs.getString(7));  
 r.setPenalSum(rs.getDouble(8));  
 result.add(r);  
 sql = "select readerName from beanreader where readerid = ?";  
 pst = conn.prepareStatement(sql);  
 pst.setString(1, r.getReaderid());  
 java.sql.ResultSet xs = pst.executeQuery();  
 if(xs.next()) readerName = xs.getNString(1);  
 sql = "select bookName, publisherName from beanbook,beanpublisher where barcode = ? and beanpublisher.pubid = beanbook.pubid";  
 pst = conn.prepareStatement(sql);  
 pst.setString(1, r.getBookBarcode());  
 xs = pst.executeQuery();  
 if(xs.next()){   
 readerName = xs.getNString(1);  
 publisherName = xs.getNString(2);  
 }  
 if(r.getLendOperUserid() != null)  
 {  
 sql = "select username from beansystemuser where userid = ?";  
 pst = conn.prepareStatement(sql);  
 pst.setString(1, r.getLendOperUserid());  
 xs = pst.executeQuery();  
 if(xs.next()) lendOperUserName = xs.getNString(1);  
 }  
 if(r.getReturnOperUserid() != null)  
 {  
 sql = "select username from beansystemuser where userid = ?";  
 pst = conn.prepareStatement(sql);  
 pst.setString(1, r.getReturnOperUserid());  
 xs = pst.executeQuery();  
 if(xs.next()) returnOperUserName = xs.getNString(1);  
 }  
 System.out.println("读者姓名:" + readerName + "\t图书名称:" +  
 bookName + "\t所属出版社名称:" + publisherName + "\t借阅操作员姓名:" + lendOperUserName  
 + "\t归还操作员姓名:" + returnOperUserName + "\t借阅时间:" +  
 r.getLendDate() + "\t归还时间:" + r.getReturnDate());  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
// throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
}



### 5、在booklib工程的BookManager类中增加如下函数，并在main函数中进行测试，在实验报告中将代码补上:

public void showTop5Books(){  
 //通过System.out.println方法，输出借阅次数最多的5本图书及其借阅次数  
 Connection conn = null;  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select bookname, count(1) as x "  
 + "from beanbooklendrecord, beanbook "  
 + "where barcode = bookBarcode "  
 + "group by bookname order by x DESC limit 5;";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 java.sql.ResultSet rs = pst.executeQuery();  
 while(rs.next())  
 {  
 System.out.println("图书名称" + rs.getString(1) + "\t借阅次数" + rs.getInt(2));  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
// throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
}

public void showTop5Publisher(){  
 //通过System.out.println方法，输出被借阅图书次数最多的5个出版名称及其总借阅次数和被借阅过的图书次数  
 Connection conn = null;  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select publishername,count(\*) as x, count(distinct bookbarcode) "  
 + "from beanbooklendrecord "  
 + "left outer join (beanbook left outer join beanpublisher on beanbook.pubid = beanpublisher.pubid) on bookbarcode = barcode "  
 + "group by beanbook.pubid order by x desc limit 5;";  
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 java.sql.ResultSet rs = pst.executeQuery();  
 while(rs.next())  
 {  
 System.out.println("出版社名称" + rs.getString(1) + "\t总借阅次数" + rs.getInt(2) + "\t被借阅过的图书数量" + rs.getInt(3));  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
// throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
}



### 6、在BookLendManager中增加函数public void printDateLendRecord(String date)throws DbException，并在main函数中调用测试;要求通过该函数输出指定日期的所有借阅记录，，输出格式如下：

readerId=\***,bookBarcode=**\*\*,lendDate=2020-05-01 15:17:01,returnDate=未归还

readerId=\***,bookBarcode=**\*\*,lendDate=2020-05-01 15:17:01,returnDate=2020-05-12 12:00:00

说明：每个借阅记录1行输出，如果returnDate为空，则输出：“未归还”

注：时间的输出格式请使用java.text.SimpleDateFormat类实现

请提供函数代码及运行结果截图：

public void printDateLendRecord(String date)throws DbException{  
 Connection conn = null;  
 SimpleDateFormat DateFor = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");  
 try {  
 conn = DBUtil.getConnection();  
 String sql = "select readerid, bookBarcode, lendDate, returnDate from beanbooklendrecord where lendDate = ?";   
 java.sql.PreparedStatement pst = conn.prepareStatement(sql);  
 pst.setString(1, date);  
 java.sql.ResultSet rs = pst.executeQuery();  
 if(rs.next())  
 if(rs.getDate(4) != null)  
 System.out.println("readerId= " + rs.getString(1) + ",bookBarcode= " + rs.getString(2) + ",lendDate= " + DateFor.format(rs.getDate(3)) + ",returnDate=" + DateFor.format(rs.getDate(4)));  
 else  
 System.out.println("readerId= " + rs.getString(1) + ",bookBarcode= " + rs.getString(2) + ",lendDate= " + DateFor.format(rs.getDate(3)) + ",未归还");  
 } catch (SQLException e) {  
 e.printStackTrace();  
 throw new DbException(e);  
 } finally {  
 if (conn != null)  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
}

