目录

1.需求分析说明

1.1 概述

数据库管理系统是位于用户与操作系统之间的一层数据管理软件。数据库管理系统和操作系统一样是计算机的基础软件。而本次课程设计的目的是让我们利用所学的 JAVA 语言知识和数据库的相关知识来开发一个小型 DBMS 系统，以实现简单数据库的创建、存储、显示、查询、更新和删除以及用户权限等功能。

1.2 基本功能

（1）设计特定的数据结构，用于存储数据表、视图等数据库对象的信息，即建立数据库系统的数据字典；

（2）设计特定的数据结构，用于存储数据表中的数据；

（3）设计特定的数据结构，分别用于存储用户和访问权限的信息；（4）输入“help database”命令，输出所有数据表、视图和索引的

信息，同时显示其对象类型；输入“help table 表名”命令，输出数据表中所有属性的详细信息；输入“help view 视图名”命令，输出视图的定义语句；

（5）解析 CREATE、SELECT、INSERT、DELETE、UPDATE 等 SQL 语句的内容；

（6）检查 SQL 语句中的语法错误和语义错误；

（7）执行 CREATE 语句，创建数据表、视图、索引等数据库对象；

（8）执行 SELECT 语句，从自主设计的数据表中查询数据，并输出结果；

（9）执行 INSERT、DELETE 和 UPDATE 语句，更新数据表的内容；提示数据表更新成功。

（10）执行 GRANT 语句，为用户授予 CREATE、SELECT、INSERT、DELETE、UPDATE 权限；执行 REVOKE 语句，收回上述权限；

（11）用户登录时，需要输入用户名；如果用户没有被授权，则拒绝执行用户操作，并给出提示信息。

1.3 系统所需环境

运行环境：java jdk1.8
开发工具：eclipse neon2，IDEA

2.概要设计说明

2.1 设计思路

程序运行初始，建立数据库和数据表、视图、索引，数据库用文件夹表示，数据库用 xls 表示，视图用 txt 表示；初始 root、guest 用户并赋予其初始密码，同时建立用户数据字典存放在“user.xls”文件中。将数据库管理系统的各个功能划分成模块，主要包括数据库、表和视图的创建模块，对表的修改、插入、删除、查询以及显示其结构，对用户的权限进行授予或取消授权模块。

2.2 数据字典

为了实现建立数据库的数据字典和存储数据表中的数据以及用户的信息，使用 jxl 库的工作台数据格式存储数据表等数据库对象信息。用户的信息，采用字符串的形式存放在 xls。

2.2.1 数据库数据字典

Cell 的数据字典

```
public interface Cell {
    int getRow();

    int getColumn();

    CellType getType();

    boolean isHidden();

    String getContents();

    CellFormat getCellFormat();

    CellFeatures getCellFeatures();
}
```

SHEET 的数据字典类型

```
public interface Sheet {
    Cell getCell(int var1, int var2);

    int getRows();
```

```
    int getColumns();

    Cell[] getRow(int var1);

    Cell[] getColumn(int var1);

    String getName();
}
```

Workbook 的数据字典类型

```
public abstract class Workbook {

    protected Workbook() {
    }


    public abstract Sheet[] getSheets();

    public abstract String[] getSheetNames();

    public abstract Sheet getSheet(int var1) throws IndexOutOfBoun
dsException;


    public abstract Sheet getSheet(String var1);

    public abstract int getNumberOfSheets();

    public abstract Cell findCellByName(String var1);
```

```
 public abstract Range[] findByName(String var1);

 public abstract String[] getRangeNames();

public abstract boolean isProtected();


 protected abstract void parse() throws BiffException,
PasswordExc eption;

    public abstract void close();

    public static Workbook getWorkbook(File file) throws
IOExcepti on, BiffException {
        return getWorkbook(file, new WorkbookSettings());
    }
```

Lable 的数据字典类型

```
            public class Label extends LabelRecord implements
                        WritableCell, La belCell {
             public Label(int c, int r, String cont)
                           { super(c, r, cont);
                                 }

        public Label(int c, int r, String cont, CellFormat st)
                          { super(c, r, cont, st);
                                 }

            protected Label(int col, int row, Label l)
                           { super(col, row, l);
```

```
                                    }

public Label(LabelCell lc) {
        super(lc);
    }

    public void setString(String s)
        { super.setString(s);
    }

    public WritableCell copyTo(int col, int row)
        { return new Label(col, row, this);
```

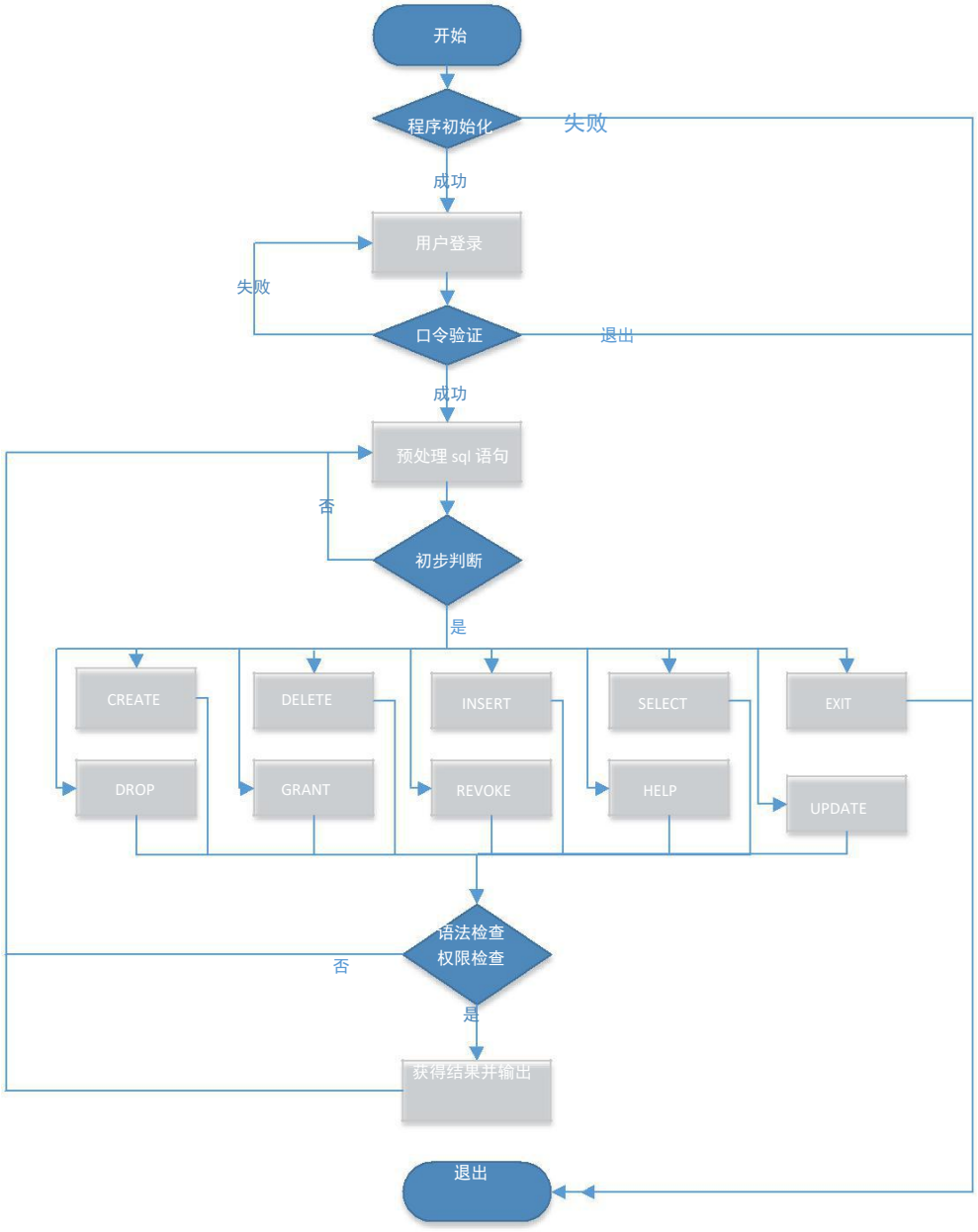
2.2.2 用户数据字典

存在 xls 文件中

user	permission	password
root	all	root
guest	insert create select of A	null

user	permission	password
root	all	root
guest	delete insert on student	null

2.3 程序流程图


开始
程序初始化
失败
成功
用户登录
失败
口令验证
退出
成功
预处理 sql 语句
否
初步判断
是
CREATE
DELETE
INSERT
SELECT
EXIT
DROP
GRANT
REVOKE
HELP
UPDATE
语法检查
权限检查
否
是
获得结果并输出
退出

3.详细设计说明

3.1 主方法模块

主方法模块的 main 函数是整个系统的入口，系统启动后，首先执行初始化操作，加载必要的数据到内存，初始化成功后进入登录模块，若初始化失败，则抓到异常，退出系统。

3.2 预处理模块

用户输入的 sql 语句是随机的、不确定的，可能存在语句中包含多余的换行符和空格的情况，因此，在获取到输入的 sql 语句之后，先对 sql 语句进行预处理，将 sql 语句转换成容易识别、拆分的语句格式，后续操作均使用预处理过的 sql 语句，预处理具体过程包括：

(1) 将所有的换行符，制表符替换成单个空格符；

(2) 将一个以上的连续的空格符替换成单个空格符；

预处理后，对 sql 语句进行分词，根据第一个关键字，调用不同的方法对 sql 语句做进一步处理。主要代码如下：

```
  String[] ss = sentence.split("\\s+");
while (sentence.lastIndexOf(";") != sentence.length() - 1) {
                    sentence = sentence + " " + input4.nextLine();
             }
```

3.3 用户登录模块

首先，用户输入用户名和密码进行登录，程序从“user.xls”中获取用户名和密码，与输入的用户名和密码进行匹配。若匹配成功则登录，进行下一步操作，否则登录失败重新加载 login 函数。登录函数如下：

```
public static void sqllogin(){
        System.out.println("Please enter user name£");
        Scanner input = new Scanner(System.in); String
        name = input.next();

        username = name;

        if(name .equals("root")){
        System.out.println("The user name is correct, please
enter the password£º");
        Scanner input1 = new Scanner(System.in);
        String password = input1.next();
        if(password.equals("root")){
        System.out.println("The password is entered
correctly£¬welcome to Amazsql !");
                    sql.sqldbroot();
              }
        else{
   System.out.println("Password input error, please re-login£");
                    sqllogin();
              }
        }else if(name.equals("guest")){
              System.out.println("welcome to Amazsql !");
              sql.sqldbvis();
        }

        else{
   System.out.println("The user does not exist, please re-enter
the user name:");
              sqllogin();
        }
   }
```

3.4 创建模块

3.4.1 创建用户模块

在系统中默认储存了 root、guest 用户，root 用户对所的操作和数据都具有最高权限。

3.4.2 创建数据库模块

系统支持创建多个数据库，通过语句 CREATE DATEBASE DATEBASE_NAME 可创建相应的数据库。具体过程如下：

(1) 用字符串匹配进行语法匹配。

(2) 检查当前用户是否有权限。

(3) 检查数据库是否已经存在。

当所有条件均满足后，将新的数据库信息写入数据字典中，并以新数据库名称为 mydatabases 文件夹名在根目录创建一个空的文件夹。

3.4.3 创建基本表模块

创建基本表过程如下：

(1) 检查是否选择数据库，判断该表是否存在。

(2) 检查当前用户是否有权限；

(3) 字符串匹配语法，匹配成功则返回 true，否则返回 false。

(4) 对语句进行分词，匹配以后，可获得相应的组，获得组的内容即获得表的名字，表中含有的的属性、类型及约束；

(5) 检查新的表名在数据库中是否已存在

(6) 执行创建，将获得表的信息存入数据字典中，同时在当前数据库下创建以表的名字为名字的空文件用来存储表的数据。

```
workbook = Workbook.createWorkbook(file);

 if (workbook != null) {
```

```
WritableSheet sheet = workbook.createSheet(filename1, 0);
     for (String str:ss1) {
                                System.out.println(str);
                           }
for (int j = 0; j < createtb.column; j++) {
                  Label label = new Label(j, 0, ss1[j]);
                  sheet.addCell(label);
}
                  workbook.write();
                  workbook.close(); }
```

3.4.4 创建视图模块

通过 CREATE VIEW VIEW_NAME SELECT …语句来创建视图，视图是虚表，创建以后并不创建实际的表，而是将视图信息存入 txt。创建过程如下：

(1) 检查是否选择数据库

(2) 检查视图在数据库中是否已存在。

执行创建

3.5 删除模块

通过 DROP DATABASE DATABASE_NAME 语句可删除数据库，DROP TABLE TABLE_NAME 语句可删除执行过程如下：

(1) 语法检查；

(2) 判断删除的是数据库、表。

(3) 权限检查；

删除相应的信息、文件或文件夹。

3.6 权限授予模块

可授予某一用户对某个数据库的 create、 select、insert、delete、update 权限。

执行过程如下：

（1）检查语法正确性

（2）检查用户是否具有权限

（3）检查数据库是否存在

（4）检查用户是否存在

（5）执行

```
File file1 = new File("user.xls");
Workbook.getWorkbook(file1);

WritableWorkbook wwb = Workbook.createWorkbook(file1, wb);
                WritableSheet ws = wwb.getSheet(1);
int columnum = ws.getColumns();
int rownum = ws.getRows();
WritableCell cell1 = null;

for (int i = 0; i < ss1.length; i++) {

                        sss1 += ss1[i] + " ";

                                                }
       cell1 = new Label(1, 2, sss1+" "+"on "+ss[3]+" of "+ss[5]);
                        ws.addCell(cell1);
                        wwb.write();
System.out.println("grant permissions of table to guest
successfully !");
wwb.close();
```

3.7 help 模块

HELP DATABASE 语句可查看当前数据库下的所有的表，视图信息；HELP TABLE TABLE_NAME 可查看表的详细信息；HELP VIEW VIEW_NAME 可查看视图的详细信息。匹配语法后会有信息后按格式输出。

3.8 插入模块

通过语句 insert into table_name(NAME1,NAME2) values(value1, value2…)向表或视图中插入数据。执行过程如下：

(1) 检查是否选择数据库；

配语法，代码如下

```
Workbook wb = Workbook.getWorkbook(file)

WritableWorkbook wwb = Workbook.createWorkbook(file,wb);
      WritableSheet ws = wwb.getSheet(0);
       int columnum = ws.getColumns(); int
      rownum = ws.getRows();
      if(attributes.length==columnum){ for
      (int j = 0; j <columnum ; j++) {
      Cell cell = ws.getCell(j, 0);
 WritableCell cell1 = null;
      String content = cell.getContents();
        String shuxing = content.substring(0, content.indexOf("
"));
              if(attributes[j].equals(shuxing)){
      cell1 = new Label(j, rownum, values[j].substring(1,
values[j].length()-1));

                ws.addCell(cell1);

              }

}
System.out.println("insert table "+tbname+"successfully !");
                                    }

 else{
 System.out.println("The number of requirements is not the same as
the actual quantity !");
                                    }
            wwb.write();

            wwb.close();
```

3.9 权限移除模块

可移除某一用户对某个数据库的 create、select、insert、delete、update 权限。

执行过程如下：

（1）检查语法正确性

（2）检查用户是否具有权限
（3）检查数据库是否存在
（4）检查用户是否存在
（5）执行

```
File file1 = new File("user.xls");
      Workbook wb = Workbook.getWorkbook(file1);


      WritableWorkbook wwb = Workbook.createWorkbook(file1, wb);
            WritableSheet ws = wwb.getSheet(1);
            int columnum = ws.getColumns();
      int rownum = ws.getRows();
            Sheet sheet = wb.getSheet(1);


            Cell cell = sheet.getCell(1, 2);
            String content = cell.getContents();//Ô-±íÖÐ´æµÄÈ¨ÏÞ
select drop create on tb_name of db_name


            WritableCell cell1 = null;


            for (int i = 0; i < ss1.length; i++) {


                  sss1 += ss1[i] + " ";


                        }
            content = content.replaceAll(sss1, "");
            cell1 = new Label(1, 2, content);
                        ws.addCell(cell1);
                        wwb.write();
System.out.println("revoke permissions of table to guest
successfully !");
                  wwb.close();


 File file1 = new File("user.xls");


            Workbook wb = Workbook.getWorkbook(file1);


WritableWorkbook wwb = Workbook.createWorkbook(file1, wb);
      WritableSheet ws = wwb.getSheet(0);
                        int columnum = ws.getColumns();
```

```
                    int rownum = ws.getRows();
                    Sheet sheet = wb.getSheet(0);
                WritableCell cell1 = null;
                Cell cell = sheet.getCell(1, 2);
      String content = cell.getContents(); for
      (int i = 0; i < ss1.length; i++) {

                sss1 += ss1[i] + " ";
                  drop create

                }
      content = content.replaceAll(sss1, "");
            cell1 = new Label(1, 2, content);
                    ws.addCell(cell1);
                    wwb.write();
      System.out.println("revoke permissions of database to guest
successfully !");
                    wwb.close();
```

3.10 查询模块

查询模块语法较复杂，可能出现带有 where 子句的

(1) select * from TABLE_NAME，匹配代码如下：

```
    jxl.Workbook workbook;
    try {
                workbook = Workbook.getWorkbook(file);
                Sheet sheet = workbook.getSheet(0);

                int columnum = sheet.getColumns();
                int rownum = sheet.getRows();
                for (int j = 0; j < columnum; j++) {
                Cell cell = sheet.getCell(j, 0);

                    String content = cell.getContents();

                    String shuxing = content.substring(0,
    content.indexOf(" "));

          System.out.print(shuxing + "         ");
```

```
                                    }
                        System.out.println();
            for (int i = 1; i < rownum; i++) {
                  for (int j = 0; j < columnum; j++) {

                        Cell cell = sheet.getCell(j, i);

                  String content = cell.getContents();



            System.out.print(content + "        ");

                                                }
                        System.out.println();


      }
```


(2) 匹配成功后，采用层析解析的方式，如果带有 where 子句，则对结果集进行筛选，获得包含所有属性的结果集，解析 where 子句内容具有通用性，：

```
if (ss[0].equals("select") && ss[2].equals("from") &&
ss[4].equals("where")) {

File file = new File("mydatabase//" + chooseAnddo.str, ss[3] +
".xls");
      if (file.exists()) {
      if (ss[1].equals("*")) {
      jxl.Workbook workbook;
      try {
      workbook = Workbook.getWorkbook(file);
      Sheet sheet = workbook.getSheet(0); int
      columnum = sheet.getColumns(); int
      rownum = sheet.getRows(); int flag1 =
      0;//
      int flag2 = 0;
      int flag3 = 0;//

      for (int j = 0; j < columnum; j++) {
                  Cell cell = sheet.getCell(j, 0);
                  String content = cell.getContents();
                  String shuxing = content.substring(0,
content.indexOf(" "));
```

```
                    String yueshu =
content.substring(content.indexOf(" ") + 1);

      if (ss[5].equals(shuxing)) {
                    flag2 = j;
                                }

                    }
      for (int i = 0; i < rownum; i++) {

      Cell c1 = sheet.getCell(flag2, i);

if (c1.getContents().equals(ss[7].substring(1,
ss[7].lastIndexOf("'")))) {
                    flag3 = i;

            }
}
      System.out.println("The contents of the query is :");
      for (int j = 0; j < columnum; j++) {
```

```
Cell cell1 = sheet.getCell(j, flag3+2);
content = cell1.getContents();

System.out.print(content + " ");
System.out.println();
                                        }


      workbook.close();
}
```

(3) 匹配成功后，采用层析解析的方式，查询单个属性如果带有 where 子句而且匹配多个属性，则对结果集进行筛选，获得包含所有属性的结果集，代码如下：

```
jxl.Workbook workbook;

try {

workbook = Workbook.getWorkbook(file);

Sheet sheet = workbook.getSheet(0); int

columnum = sheet.getColumns(); int

rownum = sheet.getRows(); int flag1 =

0;//

int flag2 = 0;//

int flag3 = 0;//

for (int j = 0; j < columnum; j++) {

Cell cell = sheet.getCell(j, 0);

String content = cell.getContents();

String shuxing = content.substring(0, content.indexOf(" "));

String yueshu = content.substring(content.indexOf(" ") + 1);

if (ss[1].equals(shuxing)) {

flag1 = j;

}

if (ss[5].equals(shuxing)) {

flag2 = j;
```

```
    }

    }


    for (int i = 0; i < rownum; i++) {
    Cell c1 = sheet.getCell(flag2, i);

    if (c1.getContents().equals(ss[7].substring(1,
ss[7].lastIndexOf("'")))) {

    flag3 = i;

    }

    }



    Cell cell1 = sheet.getCell(flag1, flag3+2);

    System.out.println("The contents of the query --" + ss[1] + "--
is :");

    content = cell1.getContents();

    System.out.println(content);

    workbook.close();
```

(4) 匹配成功后，采用层析解析的方式，查询多个属性如果带有 where 子句而且匹配多个属性，则对结果集进行筛选，获得包含所有属性的结果集，代码如下：

```
String[] ss1 = ss[1].split(",");

    // ss1[0] ss1[1] ...

    jxl.Workbook workbook;

    try {


    workbook = Workbook.getWorkbook(file);

    Sheet sheet = workbook.getSheet(0); int

    columnum = sheet.getColumns(); int

    rownum = sheet.getRows(); int flag1 =

    0;//

    int[] flag = new int[ss1.length];

    int flag2 = 0;//

    int flag3 = 0;//


    for (int i = 0; i < ss1.length; i++) {

    for (int j = 0; j < columnum; j++) {

    Cell cell = sheet.getCell(j, 0);

    String content = cell.getContents();


    String shuxing = content.substring(0, content.indexOf(" "));

    String yueshu = content.substring(content.indexOf(" ") + 1);

    if (ss1[i].equals(shuxing)) {
```

```
    flag[i] = j;

    }

    if (ss[5].equals(shuxing)) {

    flag2 = j;

    }

    }

    }

    for (int i = 0; i < rownum; i++) {



    Cell c1 = sheet.getCell(flag2, i);

    if (c1.getContents().equals(ss[7].substring(1,
ss[7].lastIndexOf("'")))) {

    flag3 = i;

    }

    }

    for (int i = 0; i < ss1.length; i++) {

    Cell cell1 = sheet.getCell(flag[i], flag3+2);

    System.out.println("The contents of the query --" + ss1[i] + "-
- is :");

    content = cell1.getContents();

    System.out.println(content);

}

    workbook.close();
```

```
    } catch (BiffException e) {

    // TODO Auto-generated catch block

    e.printStackTrace();

    } catch (IOException e) {

    // TODO Auto-generated catch block

    e.printStackTrace();

    }

    }

    } else {

    System.out.println("the data table is not exist !");

    }

    } else {

    System.out.println(

    "Sql statement input error, Please re-select your operating
options");

    }

}
```

3.11 show 模块

用户输入 SHOW DATABASE 可显示系统中的所有数据表，输入 SHOW TABLE 可以显示选中数据库中的所有表结构，输入 SHOW VIEW 可以显示选中数据库中的所有视图。

3.12 更新模块

(1) 检查是否选择数据库
(2) 检查语法
(3) 检查基本表是否存在
(4) 检查权限
(5) 获得所有的数据
(6) 如果有 where 子句，匹配符合条件的元组
(7) 获得要更新的属性及值

```
workbook = Workbook.getWorkbook(file);

    Sheet sheet = workbook.getSheet(0);

    int columnum = sheet.getColumns();/

    int rownum = sheet.getRows();

    int flag1 = 0;//

    int flag2 = 0;//

    int flag3 = 0;//

     for (int j = 0; j < columnum; j++) {

    Cell cell = sheet.getCell(j, 0);

    String content = cell.getContents();

    String shuxing = content.substring(0, content.indexOf(" "));

    String yueshu = content.substring(content.indexOf(" ") + 1);

       if(ss[3].equals(shuxing)){

         flag1 = j;

      }

    if (ss[7].equals(shuxing)) {flag2 = j;
```

```
    }

    }

    for (int i = 0; i < rownum; i++) {
    Cell c1 = sheet.getCell(flag2, i);

    if (c1.getContents().equals(ss[9].substring(1,
ss[9].lastIndexOf("'"))) {

    flag3 = i;

    }

    }

    WritableWorkbook wwb = Workbook.createWorkbook(file, workbook);
    WritableSheet ws = wwb.getSheet(0);

    Label label = new Label(flag1,flag3,ss[5].substring(1,
ss[5].lastIndexOf("'")));

    ws.addCell(label);

    wwb.write();

            System.out.println("update the table successfully !");
            wwb.close();

    workbook.close();

    } catch (BiffException e) {

    e.printStackTrace();

    } catch (IOException e) {
```

```
e.printStackTrace();
}catch (RowsExceededException e) {
e.printStackTrace();
} catch (WriteException e) {
e.printStackTrace();
}
        }else{
        System.out.println("the data table is not exists !");
        }
        }else{
        System.out.println("Sql statement input error, select the database failed! Please re-select your operating options£º");
        }
```

3.13 选择数据库模块

用户输入 USE DATABASE_NAME 即可选中某一个数据库。

4.调试分析

4.1 正确的 SQL 语句的处理

当用户输入正确的 sql 语句时，系统可对语句进行解析，执行并获得正确

```
Please select in the following operations:
---------------------------
1.build database
2.show  databases
3.delete  database
4.select the database you want to operate and operate it
5.Grant permissions of database to a user
6.Grant permissions of table to a user
7.Revoke permissions of database from a user
8.Revoke permissions of table from a user
9.show permissions of a user
10.re_login
---------------------------
Please enter the options:
```

```
You have chosen the database: A;
Please select your operation under this database:
---------------------------
1.create table
2.Show the tables under the current database
3.delete table
4.Show the data structure in the data table
5.Insert the contents into the data table
6.Query the contents of the data table
7.update the table
8.delete some contents from the table
9.exit current database
10.view
---------------------------
11.help databases
12.help table
13.help view
14.help index
---------------------------
Please enter the options:
```

4.2 出现语法或语义错误 SQL 语句的处理

当 sql 语句中出现语法及语义错误时，系统应该能够检测出来提示错误并拒绝执行操作。在系统中提供了完整的语法及语义错误检测机制，当出现错误时，可以报告具体的错误类型反馈给用户并拒绝执行操作。主要错误类型如下：

4.2.1 用户权限不足

```
8
Permission denied
```

4.2.2 语法错误

当用户语法不符合 sql 标准时提示语法错误。

```
Please enter the options:
2
Please enter the query in the following format: show databases;
show databasd;
Sql Statement input error, query failed! Please reselect your operating options:
```

5.用户使用说明

5.1 概述

在本数据库管理系统中，实现了用户登录、退出，创建数据库、数据表、视图、用户，删除数据库、数据表、视图、用户，向表中插入数据，按条件删除数据、更新数据，给用户授予和收回权限，按条件查找数据，

系统在输入获得原始的 sql 语句之后，会对 sql 语句做预处理，去除冗余的换行符、制表符、空格等符号.

5.2 使用说明

5.2.1 用户登录及退出

5.2.1.1 用户登录

语句格式: username password;
例句： root root

5.2.1.2 用户退出

语法格式：exit;

5.2.2 选择数据库

语法格式: use database_name;
例句：use A;

5.2.3 创建操作

5.2.3.1 创建数据库

语法格式：create database database_name;
例句：create database A;

5.2.3.2 创建基本表

语法格式：create table table_name (属性名 属性类型 约束条件);
例句：create table amazing(sno int(10) not_null primary_key,sname varchar(20),sage int(4)
);

5.2.3.3 创建视图

语法格式：create view view_name as select 属性名 1, 属性名 2…from table table_name;
例句：create view student2(select sno from student where sage=’19’
);

5.2.4 删除操作

5.2.4.1 删除数据库

语法格式：drop database database_name;
例句：drop database db2;

5.2.4.2 删除基本表

语法格式：drop table table_name;
例句：drop table t1;

5.2.5 插入数据

语法格式：insert into table_name values(value1, value2, value3);
例句：insert into
student(sno,sname,sage)values('1601060120','pengg','20');

5.2.6 删除数据

5.2.6.1 不带 where 子句

语法格式：delete from table_name;
例句：delete from student;

5.2.6.2 带有 where 子句

语法格式：delete from table_name where …;
例句： delete from student where sage = '20';

5.2.7 更新数据

5.2.7.1 不带 where 子句

语法格式：update table_name set…;
例句：upodate student set sage = 20;

5.2.7.2 带有 where 子句

语法格式：update table_name set… where …;
例句：update student set sname = 'lh' where sno = '1601060101';

5.2.8 查找数据

5.2.8.1 查看所有的数据

语法格式：select * from table_name;
例句：select * from student;

5.2.8.2 查看符合 where 子句条件的数据

语法格式：select s1 from table_name where…;
例句：select sname from student where sage= ' 19';

5.2.8.3 查看符合 where 子句条件的数据

语法格式：select s1,s2 from table_name where…;
例句：select sname,sno from student where sage= ' 20';

5.2.9 权限管理

5.2.9.1 授予权限

语法格式：grant （select, delete …) on table t1, t2 … to …;

例句 1：revoke update on A from guest;例句 2：grant on table t1 to u1;

5.2.9.2 收回权限

语法格式：revoke （select, delete …) on table t1, t2 … from…;

例句 1：revoke delete , update on table student from guest;

例句 1：revoke delete , update on table t1 from guest;

5.2.10 HELP 操作

5.2.10.1 输出当前数据库下所有的表和视图的信息

语法格式：help database;
例句：help database;

5.2.10.2 输出指定基本表的详细信息

语法格式：help table table_name;
例句：help table student;

5.2.10.3 输出指定视图的详细信息

语法格式：help view view_name;
例句：help view v_student;