



院 系 数据科学与计算机学院

学 号 15352237

姓 名 刘顺宇

【实验题目】Java 实验(2)

【实验目的】学习具有数据库操作的 Java 编程。

【编辑环境】

- (1)用记事本编辑和用命令行运行：点击运行 a.bat 进入控制台，再运行 b.bat 设置 classpath
- (2)用 Eclipse 或 netbeans 编辑和运行
- (3)数据库管理系统可以用 mysql 自带的，也可以用 navicat（在软件下载中有）。

【数据库系统】

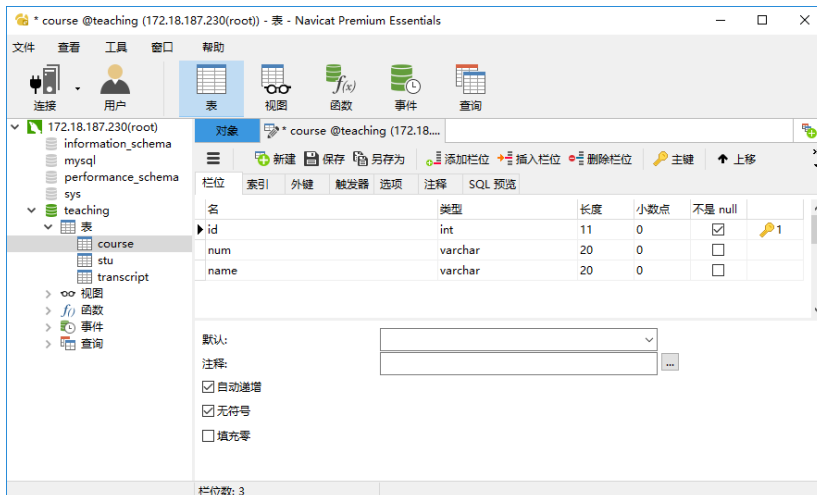
可以直接使用服务器 172.18.187.230~172.18.187.232 上建立的数据库 teaching、teaching1、...、teaching18。服务器上的数据库 teaching 的连接方法：主机 IP 地址为 172.18.187.230（替换连接字符串中的 localhost），端口号为 53306，用户名为 user，密码为 123。

可以用 Navicat 新建连接：



建立或使用学生名单 stu 和课程名单，它们的字段(field)都是：id, num, name，其中，id 为自动增长的 int 类型，num（学号）和 name（姓名）都为 32 个字符长的 varchar 类型，id 为 primary key, num 要求唯一（建立唯一性索引）。建立或使用成绩单 transcript，它的字段是：id int auto_increment primary, stuid int, courseid int，并建立两个外键让 stuid 连至 stu 的 Id，courseid 连至 course 的 Id。

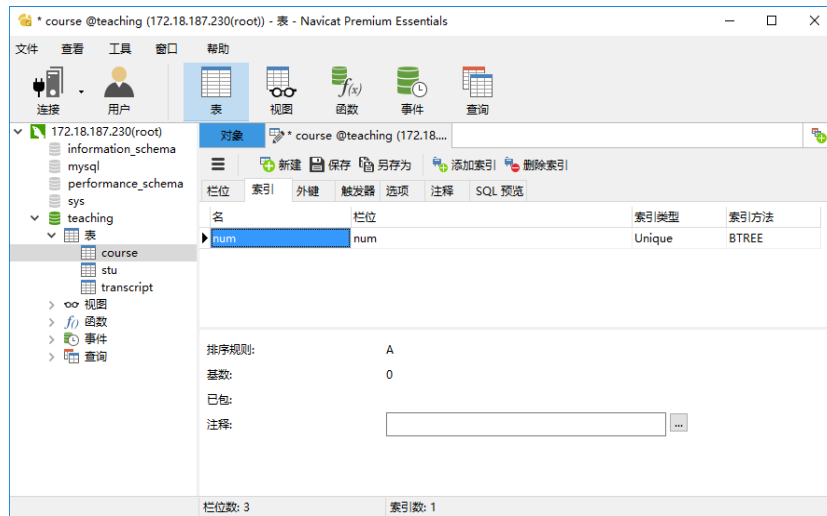
建立表和字段，设置自动增长的主键。



* 主键也可以使用多个字段，主键会自动建唯一性索引。

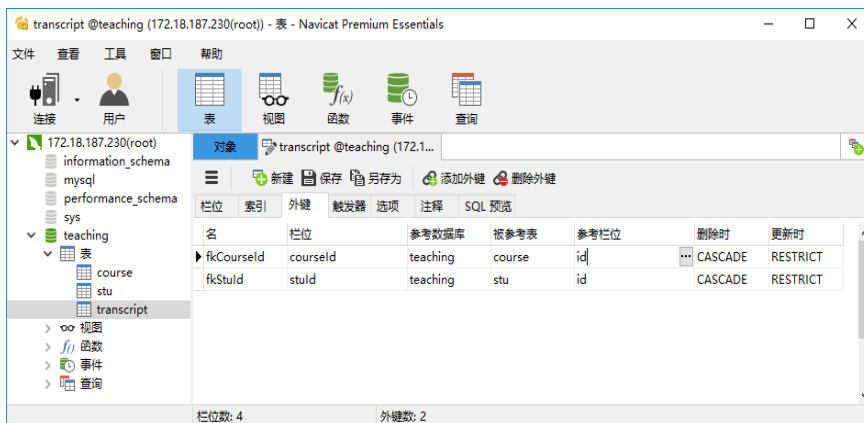


建立唯一性索引：



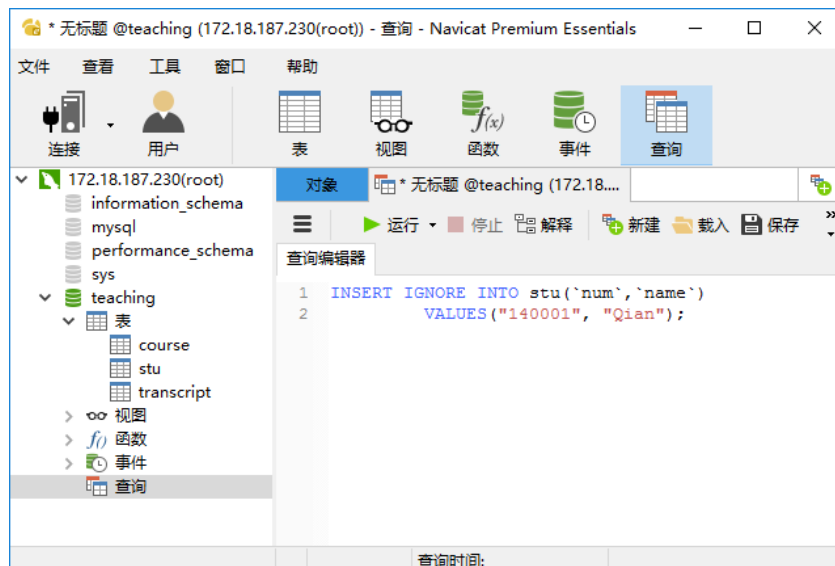
* 建立索引可以加快查询速度，唯一性索引限制插入的课程号必须唯一，否则出错。

为 transcript 建立外键：



* 外键的作用[参考](#)

进入查询页：查询/新建查询。查询页中可以输入和执行任何 SQL 语句：





* 可以直接运行 SQL.txt 中的 SQL 语句插入实验所需的数据。

【参考资料】

1、《Java 程序设计（基础）》的控制台输入输出和数据库设计。

2、SQL 语句：

<http://bbs.51cto.com/thread-450360-1-1.html>

<http://www.cnblogs.com/ggjucheng/archive/2012/11/05/2754938.html>

【程序运行参考】

可执行在 ref 目录下的.class 文件，访问的是 172.18.187.230 上的 MySQL 数据库。

【实验内容】

1、编程(QueryStu.java)实现查询操作。输入举例：

```
>1345          查出具有某些包含输入字符串的记录
                SELECT * FROM stu WHERE num LIKE "%1345%" OR name="%1345%" ORDER BY
Y num
> *            查出所有记录
                SELECT * FROM stu ORDER BY num
> -88          查出 id 为 88 的记录
                SELECT * FROM stu WHERE id=88

>exit          退出 (空行也退出)
```

参考程序：InputData.java, ShowStudents.java 先读懂并编译运行这两个程序。

生成数据：在 Navicat 的查询界面执行 SQL.data 中的语句（在数据不够时运行）

运行结果：可以直接截屏，也可以把输出内容保存到文件“c:\java QueryStu >result.txt”

进入控制台：* 点击 a.bat 出现 cmd，然后执行 b.bat 设置 classpath

```
C:\Windows\system32\cmd.exe
C:\java\ex4>cmd
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。
C:\java\ex4>b
C:\java\ex4>set classpath=.;.\mysql-connector-java-5.1.39-bin.jar
C:\java\ex4>java QueryStu
查询学号和姓名（部分匹配），以-开头查询id，*查出所有记录，exit或空行退出。

1> -183
183 140008 Li
[1条记录]

2> *
176 140001 Qian
177 140002 Li
178 140003 Wang
179 140004 Zhang
180 140005 Zhao
181 140006 Xu
182 140007 Liang
183 140008 Li
[8条记录]

3> Li
177 140002 Li
182 140007 Liang
183 140008 Li
[3条记录]

4> 002
177 140002 Li
[1条记录]

5>
C:\java\ex4>
```



完成后运行截屏（如果记录太多，可以每执行查询一次截一个图，并且只进行部分截屏）

```
C:\Windows\system32\cmd.exe
D:\学习\大三\移动web\实验\java2\code>cmd
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation。保留所有权利。

D:\学习\大三\移动web\实验\java2\code>b
D:\学习\大三\移动web\实验\java2\code>set classpath=.;.\mysql-connector-java-5.1.42-bin.jar
D:\学习\大三\移动web\实验\java2\code>javac QueryStu.java
D:\学习\大三\移动web\实验\java2\code>java QueryStu
查询学号和姓名（部分匹配），以-开头查询id，*查出所有记录，exit或空行退出。

1> -12346982
12346982 140001 Qian
[1条记录]

2> *
12346982 140001 Qian
12346984 140003 Wang
12346985 140004 Zhang
12346986 140005 Zhao
12346987 140006 Xu
12346992 140088 Ge
[6条记录]

3> Zh
12346985 140004 Zhang
12346986 140005 Zhao
[2条记录]

4> 003
12346984 140003 Wang
[1条记录]

5>
D:\学习\大三\移动web\实验\java2\code>
```

源程序：

```
1 import java.sql.*;
2 import java.util.*;
3 import java.io.*;
4 import java.io.InputStreamReader;
5
6 public class QueryStu {
7     static private Connection conn;
8     static int cnt = 0;
9     public static void main(String args[]) {
10         System.out.println("查询学号和姓名（部分匹配），以-开头查询id，*查出所有记录，exit或空行退出。\\n");
11
12         int i = 1;
13         while(true){
14             System.out.print(i + "> ");
15             i++;
16
17             Scanner sc = new Scanner(System.in);
18             String str = new String();
19             str = sc.nextLine();
20
21
22             if (connect()) {
23                 if(str.length() == 0 || str.equals("exit")){
24                     break;
25                 }
26                 else if(str.charAt(0) == '-'){
27                     ResultSet rs = executeQuery("SELECT * FROM stu WHERE id="
28                         + str.substring(1, str.length()) + ";");
29                     showStudents(rs);
30                 }
31                 else if(str.charAt(0) == '*'){
32                     ResultSet rs = executeQuery("SELECT * FROM stu;");
33                     showStudents(rs);
34                 }
35                 else if(str.charAt(0) >= '0' && str.charAt(0) <= '9'){
36                     ResultSet rs = executeQuery("SELECT * FROM stu WHERE num LIKE"
37                         + "\\%" + str + "%\\");
38                     showStudents(rs);
39                 }
40                 else if((str.charAt(0) >= 'a' && str.charAt(0) <= 'z') || (str.charAt(0) >= 'A' && str.charAt(0) <= 'Z')){
41                     ResultSet rs = executeQuery("SELECT * FROM stu WHERE name LIKE"
42                         + "\\%" + str + "%\\");
43                     showStudents(rs);
44                 }
45                 System.out.println();
46             }
47         }
48     }
49 }
```



```
44     System.out.println();
45 }
46 else System.out.println("Connect Error!");
47 }
48 }
49
50 private static boolean connect() {
51     String connectString = "jdbc:mysql://172.18.187.230:53306/teaching13"
52         + "?autoReconnect=true&useUnicode=true&characterEncoding=UTF-8&useSSL=false";
53     try {
54         Class.forName("com.mysql.jdbc.Driver");
55         conn = DriverManager.getConnection(connectString, "user", "123");
56         return true;
57     } catch (Exception e) {
58         System.out.println(e.getMessage());
59     }
60     return false;
61 }
62
63 static private ResultSet executeQuery(String sqlSentence) {
64     Statement stat;
65     ResultSet rs = null;
66
67     try {
68         stat = conn.createStatement();
69         rs = stat.executeQuery(sqlSentence);
70     } catch (Exception e) {
71         System.out.println(e.getMessage());
72     }
73     return rs;
74 }
75
76 private static void showStudents(ResultSet rs){
77     try {
78         int cnt = 0;
79         while(rs.next()){
80             System.out.println(rs.getString("id") + " "
81                 + rs.getString("num") + " "
82                 + rs.getString("name"));
83             cnt++;
84         }
85         System.out.println("[ " + cnt + " 条记录]");
86     }
87     catch (Exception e) {
88         System.out.println(e.getMessage());
89     }
90 }
91 }
```

2、编程(InsertStu.java)实现新增学生记录的操作。

SQL 语句:

> 13405601 Wang Qiang

INSERT INTO stu(num,name)VALUES('13405601','Wang Qiang');

运行参考:

```
C:\Windows\system32\cmd.exe
C:\java\ex4>Java InsertStu
输入学号和姓名(用空格间隔), exit或空行退出.
1> 140066 Ding
1条记录被加入!
2> 140067 Chen
1条记录被加入!
3> 140002 Jia
Duplicate entry '140002' for key 'num'
0条记录被加入!
4> 140088 Ge
1条记录被加入!
5>
C:\java\ex4>
```

插入记录前运行 QueryStu 截屏:

信息	结果1	概况	状态
	id	num	name
▶	12346982	140001	Qian
	12346984	140003	Wang
	12346985	140004	Zhang
	12346986	140005	Zhao
	12346987	140006	Xu
	12346992	140088	Ge



运行 InsertStu 截屏(运行时可以另开控制台运行 QueryStu):

```
C:\Windows\system32\cmd.exe
D:\学习\大三\移动web\实验\java2\code>cmd
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation. 保留所有权利。

D:\学习\大三\移动web\实验\java2\code>b
D:\学习\大三\移动web\实验\java2\code>set classpath=..\mysql-connector-java-5.1.42-bin.jar
D:\学习\大三\移动web\实验\java2\code>javac InsertStu.java
D:\学习\大三\移动web\实验\java2\code>java InsertStu
输入学号和姓名(用空格间隔), exit或空行退出.

1> 140066 Ding
1条记录被加入!

2> 140067 Chen
1条记录被加入!

3> 140006 Xu
Duplicate entry '140006' for key 'num'
0条记录被加入!

4> 140089 GG
1条记录被加入!

5>
D:\学习\大三\移动web\实验\java2\code>
```

插入记录后运行 QueryStu 截屏(标注自己插入的记录):

信息	结果1	概况	状态
	id	num	name
	12346982	140001	Qian
	12346984	140003	Wang
	12346985	140004	Zhang
	12346986	140005	Zhao
	12346987	140006	Xu
	12346992	140088	Ge
	12347014	140066	Ding
	12347015	140067	Chen
	12347017	140089	GG

源程序:

```
1 import java.sql.*;
2 import java.util.*;
3 import java.io.*;
4 import java.io.InputStreamReader;
5
6 public class InsertStu{
7     static private Connection conn;
8     static int cnt = 0;
9     public static void main(String args[]) {
10         System.out.println("输入学号和姓名(用空格间隔), exit或空行退出.\n");
11         int i = 1;
12         while(true){
13             System.out.print(i+"> ");
14             i++;
15
16             Scanner sc = new Scanner(System.in);
17             String str = new String();
18             str = sc.nextLine();
19             if (connect()) {
20                 if(str.length() == 0 || str.equals("exit")){
21                     break;
22                 }
23                 else{
24                     int pos = str.indexOf(" ");
25                     boolean rs = executeUpdate("INSERT INTO stu(num,name)VALUES(\""
26                         + str.substring(0,pos) + "\",\""+
27                         + str.substring(pos + 1,str.length()) + "\");");
28                 }
29             }
30             System.out.println();
31         }
32         else System.out.println("Connect Error!");
33     }
34 }
35
```



```
36 private static boolean connect() {
37     String connectString = "jdbc:mysql://172.18.187.230:53306/teaching13"
38     + "?autoReconnect=true&useUnicode=true&characterEncoding=UTF-8&useSSL=false";
39     try {
40         Class.forName("com.mysql.jdbc.Driver");
41         conn = DriverManager.getConnection(connectString, "user", "123");
42         return true;
43     } catch (Exception e) {
44         System.out.println(e.getMessage());
45     }
46     return false;
47 }
48
49 private static boolean executeUpdate(String sqlSentence) {
50     Statement stat;
51     cnt = -1;
52     try {
53         stat = conn.createStatement(); // 根据连接获取一个执行sql语句的对象
54         cnt = stat.executeUpdate(sqlSentence); // 执行sql语句,返回所影响行记录的个数
55     }
56     catch (Exception e) {
57         System.out.println(e.getMessage());
58     }
59     if(cnt >= 0) System.out.println(cnt + "条记录被加入! ");
60     else System.out.println("0条记录被加入! ");
61     return (cnt >= 0);
62 }
63 }
```

- 3、(选做) 编程(UpdateStu.java)实现修改操作: 通过输入 id、num 和 name 修改该输入 id 对应记录的 num 字段和 name 字段的值, 或者通过输入 num、name, 修改 num 对应的 name。

SQL 语句:

> 88 1234567 张三

UPDATE stu SET num='1234567', name='张三' WHERE id=88

> 1234567 张三

UPDATE stu SET name='张三' WHERE num= '1234567'

运行参考:

```
C:\Windows\system32\cmd.exe
C:\java\ex4>Java UpdateStu
输入学号、姓名或者id、学号 姓名, exit或空行退出.

1> 140030 Ren
1个记录被修改.

2> 140031 Hua
1个记录被修改.

3> 178 140050 Jin
1个记录被修改.

4> 180 140007 He
Duplicate entry '140007' for key 'num'
0个记录被修改.

5> 180 140077 He
1个记录被修改.

6>

C:\java\ex4>
```

修改记录前运行 QueryStu 截屏 (标注自己要修改的记录):

信息	结果1	概况	状态
id	num	name	
12346982	140001	Qian	
12346984	140003	Wang	
12346985	140004	Zhang	
12346986	140005	Zhao	
12346987	140006	Xu	
12346992	140088	Ge	
12347014	140066	Ding	
12347015	140067	Chen	
12347017	140089	GG	

运行 UpdateStu 截屏:



```
C:\Windows\system32\cmd.exe
D:\学习\大三\移动web\实验\java2\code>cmd
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation。保留所有权利。

D:\学习\大三\移动web\实验\java2\code>b
D:\学习\大三\移动web\实验\java2\code>set classpath=.;.\mysql-connector-java-5.1.42-bin.jar
D:\学习\大三\移动web\实验\java2\code>javac UpdateStu.java
D:\学习\大三\移动web\实验\java2\code>java UpdateStu
输入学号、姓名或者id、学号 姓名, exit或空行退出。

1> 140089 HH
1个记录被修改。

2> 140067 Liu
1个记录被修改。

3> 12346982 140003 HE
Duplicate entry '140003' for key 'num'
0个记录被修改。

4> 12346982 140073 HE
1个记录被修改。

5> 12347017 140099 huang
1个记录被修改。

6>
D:\学习\大三\移动web\实验\java2\code>
```

修改记录后运行 QueryStu 截屏（标注自己修改的记录）：

信息	结果1	概况	状态
id	num	name	
▶ 12346982	140073	HE	
12346984	140003	Wang	
12346985	140004	Zhang	
12346986	140005	Zhao	
12346987	140006	Xu	
12346992	140088	Ge	
12347014	140066	Ding	
12347015	140067	Liu	
12347017	140099	huang	

4、(选做) 用 Java 编程(DeleteStu.java) 删除某 id 对应的记录或者删除学号和姓名中部分匹配的记录。

SQL 语句: >-88

```
DELETE FROM stu WHERE id=88
```

>123

```
DELETE FROM stu WHERE num LIKE '%123%' OR name LIKE '%123%'
```

运行参考:

```
C:\Windows\system32\cmd.exe
C:\java\ex4>java DeleteStu
输入学号或姓名（部分匹配）以删除记录，输入以-开头数字则按id删除，exit或空行退出。

1> -189
1个记录被删除。

2> Li
3个记录被删除。

3> 0006
1个记录被删除。

4>
C:\java\ex4>
```




删除记录前运行 QueryStu 截屏（标注要修改的记录）：

信息	结果1	概况	状态
id	num	name	
12346982	140073	HE	
12346984	140003	Wang	
12346985	140004	Zhang	
12346986	140005	Zhao	
12346987	140006	Xu	
12346992	140088	Ge	
12347014	140066	Ding	
12347015	140067	Liu	
12347017	140099	huang	

运行 DeleteStu 截屏：

```
C:\Windows\system32\cmd.exe
D:\学习\大三\移动web\实验\java2\code>cmd
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation。保留所有权利。

D:\学习\大三\移动web\实验\java2\code>b
D:\学习\大三\移动web\实验\java2\code>set classpath=..\mysql-connector-java-5.1.42-bin.jar
D:\学习\大三\移动web\实验\java2\code>javac DeleteStu.java
D:\学习\大三\移动web\实验\java2\code>java DeleteStu
输入学号或姓名（部分匹配）以删除记录，输入以-开头数字则按id删除，exit或空行退出。
1> -12347017
1个记录被删除。
2> Zh
2个记录被删除。
3> 006
3个记录被删除。
4>
D:\学习\大三\移动web\实验\java2\code>
```

删除记录后运行 QueryStu 截屏：

信息	结果1	概况	状态
id	num	name	
12346982	140073	HE	
12346984	140003	Wang	
12346992	140088	Ge	

源程序：

```
1 import java.sql.*;
2 import java.util.*;
3 import java.io.*;
4 import java.io.InputStreamReader;
5
6 public class DeleteStu{
7     static private Connection conn;
8     static int cnt = 0;
9     public static void main(String args[]) {
10         System.out.println("输入学号或姓名（部分匹配）以删除记录，输入以-开头数字则按id删除，exit或空行退出.\n");
11         int i = 1;
12         while(true){
13             System.out.print(i + "> ");
14             i++;
15
16             Scanner sc = new Scanner(System.in);
17             String str = new String();
18             str = sc.nextLine();
19             if (connect()) {
20                 if(str.length() == 0 || str.equals("exit")){
21                     break;
22                 } else{
23                     boolean rs;
24                     if(str.charAt(0) == '-'){
25                         rs = executeUpdate("DELETE FROM stu WHERE id="
26                             + str.substring(1,str.length()) + ";");
27                     } else{
28                         rs = executeUpdate("DELETE FROM stu WHERE num like '%"
29                             + str+"%' OR name LIKE '%" + str + "%'");
30                     }
31                 }
32                 System.out.println();
33             }
34         }
35     }
36 }
```



```
34     else System.out.println("Connect Error!");
35 }
36 }
37 private static boolean connect() {
38     String connectString = "jdbc:mysql://172.18.187.230:53306/teaching13"
39         + "?autoReconnect=true&useUnicode=true&characterEncoding=UTF-8&useSSL=false";
40     try {
41         Class.forName("com.mysql.jdbc.Driver");
42         conn = DriverManager.getConnection(connectString, "user", "123");
43         return true;
44     } catch (Exception e) {
45         System.out.println(e.getMessage());
46     }
47     return false;
48 }
49
50 private static boolean executeUpdate(String sqlSentence) {
51     Statement stat;
52     cnt=-1;
53     try {
54         stat = conn.createStatement(); // 根据连接获取一个执行sql语句的对象
55         cnt = stat.executeUpdate(sqlSentence); // 执行sql语句,返回所影响行记录的个数
56     }
57     catch (Exception e) {
58         System.out.println(e.getMessage());
59     }
60     if(cnt >= 0) System.out.println(cnt+"个记录被删除.");
61     else System.out.println("0个记录被删除.");
62     return (cnt >= 0);
63 }
64 }
```

【完成情况】

是否完成了这些步骤? (√完成 ×未做 -未做完)

1 [√] 2 [√] 3 [√] 4 [√]

【实验体会】

写出实验过程中遇到的问题，解决方法和自己的思考；并简述实验体会（如果有的话）。

- 在做第一题的时候进行查询操作使用到了 `executeQuery()` 函数，但是发现在后面几题这个函数就不能使用了，会报错，后来查询了相关资料，知道了数据库有三种执行函数，分别为 `execute`，`executeQuery` 和 `executeUpdate`，它们之间的区别如下：
 - `executeQuery()`：直接从数据库中获取数据，只能执行 `select` 语句。通常用于读取数据库的内容。以结果集的形式输出数据。一般使用于 `SELECT` 语句。
 - `executeUpdate()`：对数据库进行 `insert`，`delete` 或者 `update`。通常用于改变数据库。一般来说 `DROP TABLE` 或数据库，插入到表，更新表，从 `TABLE` 语句删除将在此使用。输出将在 `INT` 的形式。此 `int` 值表示受到查询的行数。
 - `execute()`：所有对数据库的操作都可以使用此方法。如果用户不知道要用于执行 `SQL` 语句该方法，可以使用这种方法。这将返回一个布尔值。 `TRUE` 表示结果是一个 `ResultSet`，`FALSE` 表示它有它表示受查询的行数的 `int` 值。
- 在字符串中定义一些特殊的字符，如单引号，双引号，都要加上转义字符`\`。

【交实验报告】

(a) 每位同学单独完成本实验内容并填写实验报告上传一份。

(b) 截止时间：2017 年 11 月 23 日（周四）22:30

上交作业网站：<http://172.18.187.11/netdisk/default.aspx?vm=15web>

文件夹：/实验上交/java2

上传文件：学号_姓名_java2.doc（实验报告）

学号_姓名_java2.rar（所有.java 和.class 文件）