



## 【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 netbeas 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】

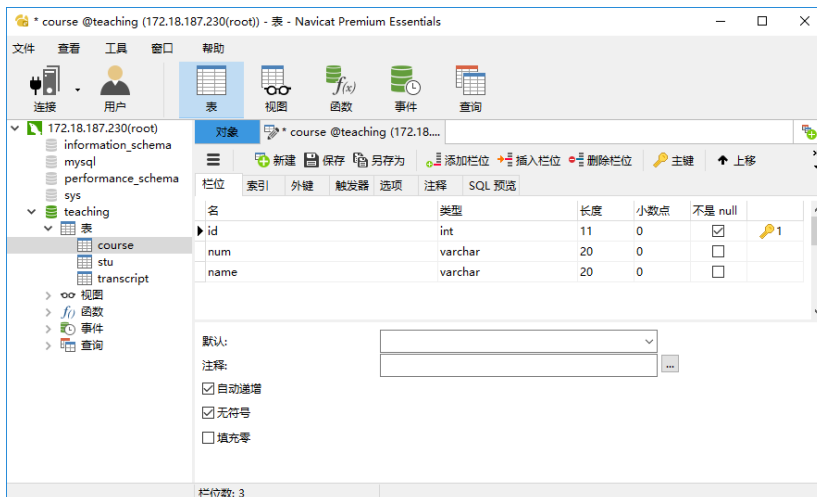
可以直接使用服务器 172.18.187.230~172.18.187.232 上建立的数据库 teaching、teaching1、...、teaching18。服务器上的数据库 teaching 的连接方法：主机 IP 地址为 172.18.187.230（替换连接字符串中的 localhost），端口号为 53306，用户名为 user，密码为 123。

可以用 Navicat 新建连接：



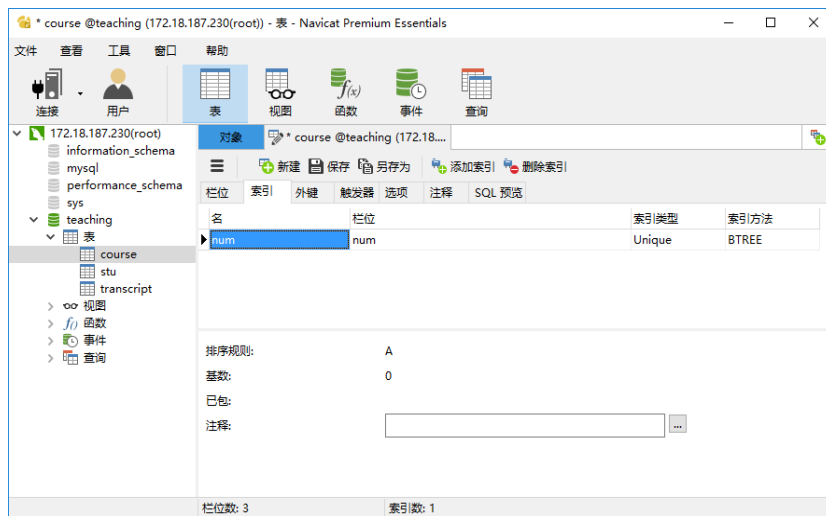
建立或使用学生名单 stu 和课程名单，它们的字段(field)都是：id, num, name，其中，id 为自动增长的 int 类型，num（学号）和 name（姓名）都为 32 个字符长的 varchar 类型，id 为 primary key，num 要求唯一（建立唯一性索引）。建立或使用成绩单 transcript，它的字段是：id int auto\_increment primary, stuId int, courseId int，并建立两个外键让 stuId 连至 stu 的 Id，courseId 连至 course 的 Id。

建立表和字段，设置自动增长的主键。



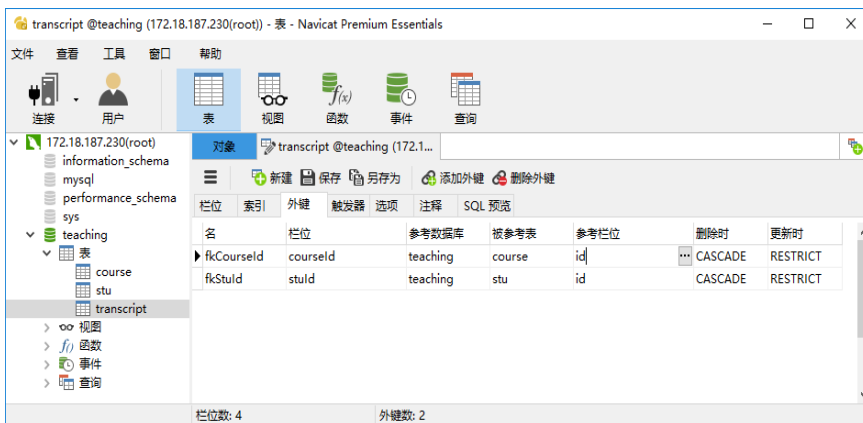
\* 主键也可以使用多个字段，主键会自动建唯一性索引。

建立唯一性索引：



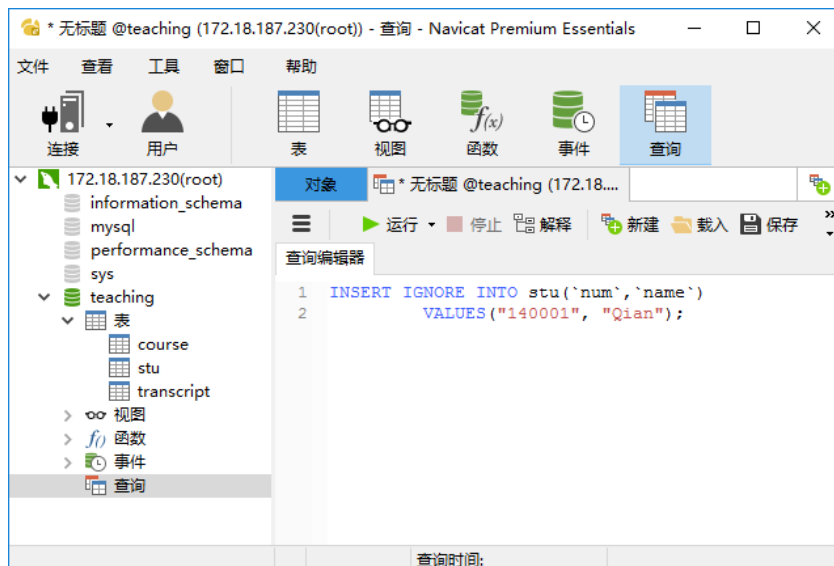
\* 建立索引可以加快查询速度，唯一性索引限制插入的课程号必须唯一，否则出错。

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 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

E:\Web\JAVA2\code>java QueryStu
查询学号和姓名(部分匹配)，以-开头查询id，*查出所有的记录，exit或空行退出。

1> *
12347062 140001 Qian
12347063 140002 Li
12347064 140003 Wang
12347065 140004 Zhang
12347066 140005 Zhao
12347067 140006 Xu
12347068 140007 Liang
12347069 140008 Li
[8条记录]

2> -12347063
12347063 140002 Li
[1条记录]

3> Li
12347063 140002 Li
12347068 140007 Liang
12347069 140008 Li
[3条记录]

4> 004
12347065 140004 Zhang
[1条记录]

5>

E:\Web\JAVA2\code>_
```

源程序:

```
import java.util.*;
import java.sql.*;

public class QueryStu {
    static private Connection conn;
    static int cnt = 0;
    static Scanner in = new Scanner(System.in);
    public static void main(String args[]) {
        if (!connect()) {
            System.out.println("Connect Error!");
            return;
        }
        String s;
        while ((s=getNextLineWords())!=null) {
            ResultSet rs;
            if(s.substring(0,1).equals("*"))
                rs = executeQuery("select * from stu;");
            else if(s.substring(0,1).equals("-"))
                rs = executeQuery("SELECT * FROM stu WHERE id="+s.substring(1,s.length()) + ";");
            else
                rs = executeQuery("SELECT * FROM stu WHERE num LIKE \"%"+s+"%\" OR name LIKE \"%"+s+"%\" ORDER BY num;");
            showStuInfo(rs);
        }
    }

    public static String getNextLineWords() {
        if (cnt++ == 0)
```



```
        System.out.println("查询学号和姓名(部分匹配)，以-开头查询id，*查出所有的记录，exit或空行退出.\r\n");
        System.out.print(cnt+"> ");
        if (!in.hasNextLine()) // 是否还有输入
            return null;
        String line = in.nextLine().replace(" ", ""); // 读取下一行并去所有空格
        if (line.equals("exit") || line.length() == 0)
            return null;
        return line;
    }

    private static boolean connect() { // 建立连接
        String connectString = "jdbc:mysql://172.18.187.231:53306/teaching13"
            + "?autoReconnect=true&useUnicode=true"
            + "&characterEncoding=UTF-8&useSSL=false";

        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(connectString, "user", "123");
            return true;
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return false;
    }

    static private ResultSet executeQuery(String sqlSentence) { // 执行SQL查询语句，返回结果集
        Statement statement;
        ResultSet rs = null;
        try {
            statement = conn.createStatement(); // 获取执行sql语句的对象
            rs = statement.executeQuery(sqlSentence); // 执行sql查询，返回结果集
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return rs;
    }

    private static void showStuInfo(ResultSet rs) { // 显示查询结果
        try {
            int count = 0;
            while(rs.next()){
                System.out.println(rs.getString("id")+" "+rs.getString("num")+" "+rs.getString("name"));
                count++;
            }
            System.out.println("[~+count+条记录]\n");
        }
        catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

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:

```
E:\Web\JAVA2\code>java QueryStu
查询学号和姓名(部分匹配), 以-开头查询id, *查出所有的记录, exit或空行退出.

1> *
12347062 140001 Qian
12347063 140002 Li
12347064 140003 Wang
12347065 140004 Zhang
12347066 140005 Zhao
12347067 140006 Xu
12347068 140007 Liang
12347069 140008 Li
[8条记录]

2>

E:\Web\JAVA2\code>
```

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

```
E:\Web\JAVA2\code>java InsertStu
输入学号和姓名(用空格间隔), exit或空行退出.

1> 140001 Zhang
Duplicate entry '140001' for key 'num'
0个记录被加入

2> 140009 Zhang
1个记录被加入

3> 140010 Chen
1个记录被加入

4>

E:\Web\JAVA2\code>
```

插入记录后运行 QueryStu 截屏 (标注自己插入的记录):



```
E:\Web\JAVA2\code>java QueryStu
查询学号和姓名(部分匹配)，以-开头查询id，*查出所有的记录，exit或空行退出。
```

```
1> *
12347081 140001 Qian
12347082 140002 Li
12347083 140003 Wang
12347084 140004 Zhang
12347085 140005 Zhao
12347086 140006 Xu
12347087 140007 Liang
12347088 140008 Li
12347090 140009 Zhang
12347091 140010 Chen
[10条记录]
```

```
2>
```

```
E:\Web\JAVA2\code>_
```

源程序:

```
import java.util.*;
import java.sql.*;

public class InsertStu {
    static private Connection conn;
    static int cnt = 0;
    static int count_data=-1;
    static Scanner in = new Scanner(System.in);
    public static void main(String args[]) {
        if (!connect()) {
            System.out.println("Connect Error!");
            return;
        }
        String []s;
        while ((s=getNextLineWords())!=null)
            executeUpdate("INSERT INTO stu(num,name)VALUES('"+s[0]+"','"+s[s.length-1]+"')");
    }
    public static String[] getNextLineWords() {
        if (cnt++ == 0)
            System.out.println("输入学号和姓名(用空格间隔)，exit 或空行退出.\r\n");
        System.out.print(cnt+"> ");
        if (!in.hasNextLine()) // 是否还有输入
            return null;
        String s = in.nextLine().trim();//读取下一行
        if (s.equals("exit") || s.length() == 0)
            return null;
        return s.split(" ");
    }
    private static boolean connect() { // 建立连接
        String connectString = "jdbc:mysql://172.18.187.231:53306/teaching13"
            + "?autoReconnect=true&useUnicode=true"
            + "&characterEncoding=UTF-8&useSSL=false";
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(connectString, "user", "123");
            return true;
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return false;
    }
    static private void executeUpdate(String sqlSentence) { //执行 SQL 查询语句，返回结果集
        Statement statement;
        ResultSet rs = null;
        try {
            statement = conn.createStatement(); //获取执行 sql 语句的对象
            count_data = statement.executeUpdate(sqlSentence); //执行 sql 插入，返回结果集
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        if(count_data>=0)System.out.println(count_data + "个记录被加入\n");
    }
}
```



```
else System.out.println("0 个记录被加入\n");  
}  
}
```

- 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）：

```
E:\Web\JAVA2\code>java QueryStu  
查询学号和姓名(部分匹配)，以-开头查询id，*查出所有的记录, exit或空行退出.  
1> *  
12347081 140001 Qian  
12347082 140002 Li  
12347083 140003 Wang  
12347084 140004 Zhang  
12347085 140005 Zhao  
12347086 140006 Xu  
12347087 140007 Liang  
12347088 140008 Li  
12347090 140009 Zhang ← name -> Jiang  
12347091 140010 Chen ← name -> Zhang  
[10条记录] num -> 140010  
2>  
E:\Web\JAVA2\code>_
```

运行 UpdateStu 截屏：





```
E:\Web\JAVA2\code>java UpdateStu
输入学号,姓名或者id,学号,姓名, exit或空行退出.
```

```
1> 12347091 140001 Jiang
Duplicate entry '140001' for key 'num'
0个记录被修改
```

```
2> 12347091 140010 Zhang
1个记录被修改
```

```
3> 140009 Jiang
1个记录被修改
```

```
4>
```

```
E:\Web\JAVA2\code>java QueryStu
```

修改记录后运行 QueryStu 截屏 (标注自己修改的记录):

```
E:\Web\JAVA2\code>java QueryStu
查询学号和姓名(部分匹配), 以-开头查询id, *查出所有的记录, exit或空行退出.
```

```
1> *
12347081 140001 Qian
12347082 140002 Li
12347083 140003 Wang
12347084 140004 Zhang
12347085 140005 Zhao
12347086 140006 Xu
12347087 140007 Liang
12347088 140008 Li
12347090 140009 Jiang
12347091 140010 Zhang
[10条记录]
```

```
2>
```

```
E:\Web\JAVA2\code>
```

源程序:

```
import java.util.*;
import java.sql.*;
public class UpdateStu {
    static private Connection conn;
    static int cnt = 0;
    static int count_data=0;
    static Scanner in = new Scanner(System.in);
    public static void main(String args[]) {
        if (!connect()) {
            System.out.println("Connect Error!");
            return;
        }
        String []s;
        while ((s=getNextLineWords())!=null) {
            if(s.length==3)
                executeUpdate("UPDATE stu SET num='"+s[1]+"' , name='"+s[2]+"' WHERE id='"+s[0]+"'");
            else
                executeUpdate("UPDATE stu SET name='"+s[1]+"' WHERE num= '"+s[0]+"'");
        }
    }
    public static String[] getNextLineWords() {
        if (cnt++ == 0)
            System.out.println("输入学号,姓名或者 id,学号,姓名, exit 或空行退出.\r\n");
        System.out.print(cnt+"> ");
        if (!in.hasNextLine()) // 是否还有输入
            return null;
        String s = in.nextLine().trim();//读取下一行
        if (s.equals("exit") || s.length() == 0)
            return null;
        return s.split(" ");
    }
    private static boolean connect() { // 建立连接
```



```
String connectString = "jdbc:mysql://172.18.187.231:53306/teaching13"
    + "?autoReconnect=true&useUnicode=true"
    + "&characterEncoding=UTF-8&useSSL=false";

try {
    Class.forName("com.mysql.jdbc.Driver");
    conn = DriverManager.getConnection(connectString, "user", "123");
    return true;
} catch (Exception e) {
    System.out.println(e.getMessage());
}

return false;
}

static private void executeUpdate(String sqlSentence) { //执行 SQL 查询语句，返回结果集
    Statement statement;
    ResultSet rs = null;
    try {
        statement = conn.createStatement(); //获取执行 sql 语句的对象
        count_data = statement.executeUpdate(sqlSentence); //执行 sql 插入，返回结果集
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }

    if(count_data>=0) System.out.println(count_data + "个记录被修改\n");
    else System.out.println("0 个记录被修改\n");
}
}
```

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）:



```
E:\Web\JAVA2\code>java QueryStu
查询学号和姓名(部分匹配)，以-开头查询id，*查出所有的记录，exit或空行退出。
```

```
1> *
12347081 140001 Qian
12347082 140002 Li
12347083 140003 Wang
12347084 140004 Zhang
12347085 140005 Zhao
12347086 140006 Xu
12347087 140007 Liang
12347088 140008 Li
12347090 140009 Jiang
12347091 140010 Zhang
[10条记录]
```

```
2>
```

```
E:\Web\JAVA2\code>java DeleteStu
```

运行 DeleteStu 截屏：

```
E:\Web\JAVA2\code>java DeleteStu
输入学号或姓名(部分匹配)以删除记录，输入以-id开头数字则按id删除，exit或空行退出
```

```
1> -12347091
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>
```

```
E:\Web\JAVA2\code>java QueryStu
```

删除记录后运行 QueryStu 截屏：

```
查询学号和姓名(部分匹配)，以-开头查询id，*查出所有的记录，exit或空行退出。
```

```
1> *
12347081 140001 Qian
12347083 140003 Wang
12347084 140004 Zhang
12347085 140005 Zhao
12347090 140009 Jiang
[5条记录]
```

```
2>
```

```
E:\Web\JAVA2\code>_
```

源程序：

```
import java.util.*;
import java.sql.*;

public class DeleteStu {
    static private Connection conn;
    static int cnt = 0;
    static int count_data=-1;
    static Scanner in = new Scanner(System.in);
    public static void main(String args[]) {
        if (!connect()) {
            System.out.println("Connect Error!");
            return;
        }
        String s;
        while ((s=getNextLineWords())!=null) {
            if(s.substring(0,1).equals("-")) {
                s = s.substring(1, s.length());
                executeUpdate("DELETE FROM stu WHERE id=" + s + ";");
            }
        }
    }
}
```



```
        else
            executeUpdate("DELETE FROM stu WHERE num LIKE '%" + s + "%' OR name LIKE '%" + s + "%'");
    }
}

public static String getNextLineWords() {
    if (cnt++ == 0)
        System.out.println("输入学号或姓名(部分匹配)以删除记录, 输入以-id 开头数字则按 id 删除, exit 或空行退出.\r\n");
    System.out.print(cnt + "> ");
    if (!in.hasNextLine()) // 是否还有输入
        return null;
    String s = in.nextLine().replace(" ", ""); // 读取下一行
    if (s.equals("exit") || s.length() == 0)
        return null;
    return s;
}

private static boolean connect() { // 建立连接
    String connectString = "jdbc:mysql://172.18.187.231:53306/teaching13"
        + "?autoReconnect=true&useUnicode=true"
        + "&characterEncoding=UTF-8&useSSL=false";
    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(connectString, "user", "123");
        return true;
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return false;
}

static private void executeUpdate(String sqlSentence) { // 执行 SQL 查询语句, 返回结果集
    Statement statement;
    ResultSet rs = null;
    try {
        statement = conn.createStatement(); // 获取执行 sql 语句的对象
        count_data = statement.executeUpdate(sqlSentence); // 执行 sql 插入, 返回结果集
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    if (count_data >= 0) System.out.println(count_data + "个记录被删除\n");
    else System.out.println("0 个记录被删除\n");
}
}
```