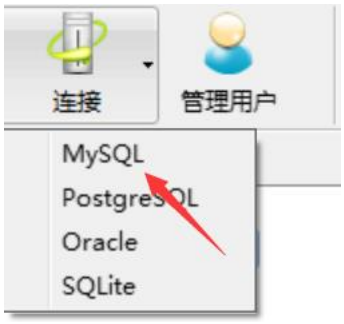
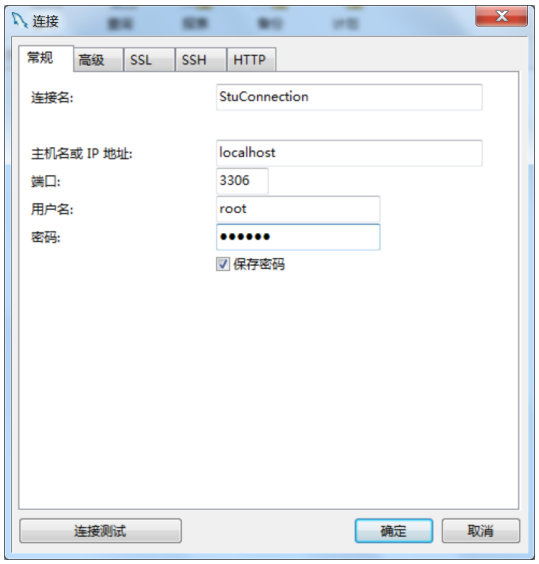
第 9 章 JSP 操作数据库

9.1 JDBC 查询数据库

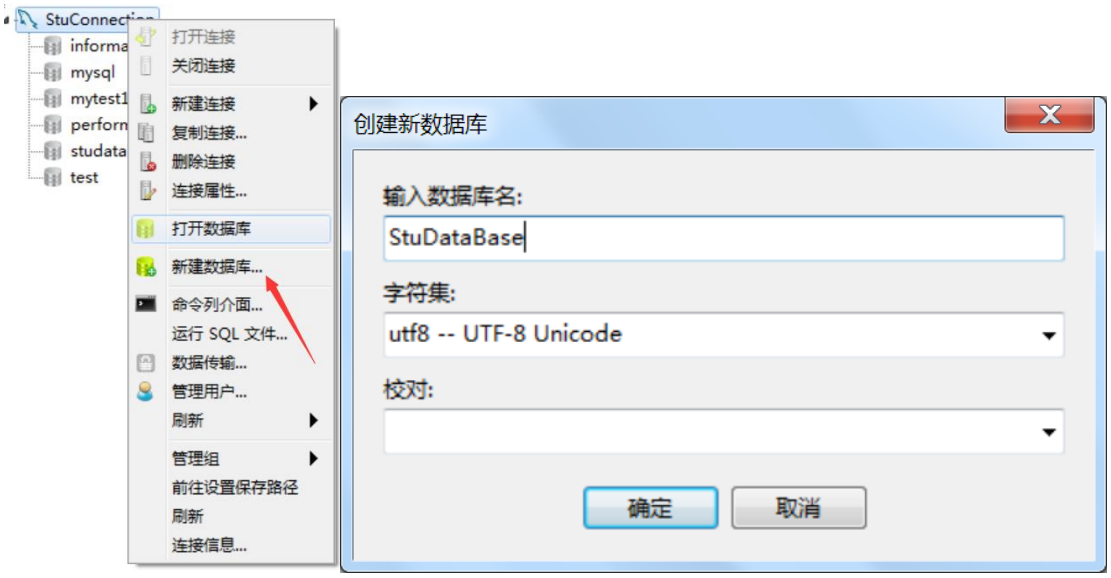
1. 拷贝数据库文件

2. MySQL 建立数据连接

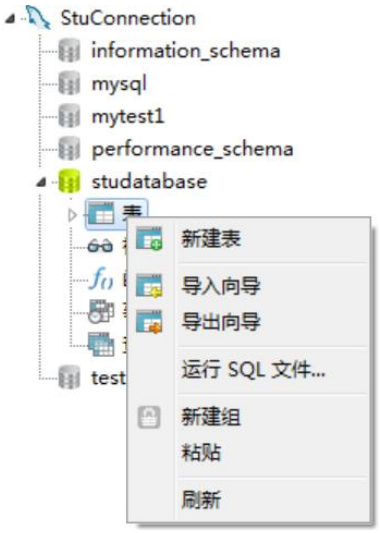


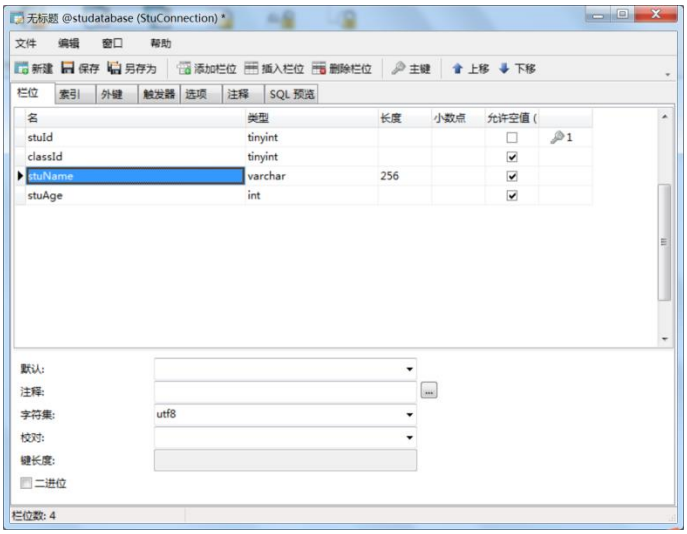


3. MySQL 建立数据库



4. MySQL 建立数据表





设置主键、字段自动递增

5.导入 JDBC 驱动包（jar 包）：mysql-connector-java-5.1.46.jar

（1）导入到 lib 文件夹下

（2）右键→Build Path→Add to Build Path

6. 使用 JDBC 查询数据表基本代码及流程说明（**新建 Query.java**）

（1）导入包

import java.sql.*;

（2）定义变量

① 驱动器字符串

```
private static String driver = "com.mysql.jdbc.Driver";
```

② 数据库连接地址、用户名、密码等字符串

```
private static String uri = "jdbc:mysql://localhost:3306/ 数据库?characterEncoding=utf8 ";
private static String user = "root";
private static String password = "123456";
```

③//查询字符串

String sqlstr = "select * from 数据表";

（3）加载驱动器（try-catch 处理-**ClassNotFoundException**）

Class.forName(driver).newInstance();

（4）获取数据库链接

Connection conn = DriverManager.getConnection(uri,user,password);

（5）获取 SQL 语句对象

Statement stmt = conn.createStatement();

（6）获取结果集

ResultSet rs = stmt.executeQuery(sqlstr);

（7）循环输出记录

```
while (rs.next()) {
    out.print(rs.getString(2))//获取第 2 列;
}
```

注意：第 4、5、6、7 步需 try-catch 处理-SQLException

9.2 添加记录

1. **新建 Add.java**，拷贝 **Query.java** 的代码，删除 9.1 节（2）③中定义的变量：

（2）③//查询字符串

~~String sqlstr = "select * from Student";~~

```
//敲写插入记录字符串
String insertstr = "INSERT INTO Student(学号,班级,姓名,性别) VALUES(‘’,’’,’’,’’)";
```

2. 改变 9.1 节（6）中获取结果集为执行更新

```
int num = stmt.executeUpdate(insertstr);
//解释 num 类型为什么为 int?
```

3. 删除（7）代码

9.3 编辑记录

1. **新建 Edit.java**，拷贝 **Add.java** 的代码，删除 9.1 节（2）③中定义的变量：

```
//敲写编辑记录字符串
String updatestr = "UPDATE Student SET 姓名=’Tomcat’ WHERE 学号=’’";
```

2. 改变 9.1 节（6）中获取结果集为执行更新

```
int num = stmt.executeUpdate(updatestr);
```

9.4 删除记录

1. **新建 Delete.java**，拷贝 **Add.java** 的代码，删除 9.1 节（2）③中定义的变量：

```
//敲写删除记录字符串
String deletestr = "DETELE FROM Student WHERE 学号=’’";
```

2. 改变 9.1 节（6）中获取结果集为执行更新

```
int num = stmt.executeUpdate(deletestr);
```

9.5 带游标的查询操作

1. 游标可以随意定位

【外引】小卒一去不回头、风萧萧兮易水寒

2. **新建 QueryScale.java**，拷贝 **Query.java** 的代码，修改程序代码。

（1）9.1 节（5）中代码

```
Statement stmt = conn.createStatement();
```

（2）改后代码

```
// 添加两个变量
int type = ResultSet.TYPE_SCROLL_SENSITIVE;
int concur = ResultSet.CONCUR_UPDATABLE;
//调用重载方法
Statement stmt = conn.createStatement(type,concur);
```

3. 统计表中记录个数

（1）游标定位到表中末条记录

```
rs.last();
```

（2）获取末条记录行号

```
int num = rs.getRow();
out.print("1、表中共有"+num+"条记录<br/>");
out.print("<hr/>");
```

4. 获取第 1 个记录信息*

```
游标定位到表中第 1 条记录
rs.absolute(1);
//输出第 1 条记录信息
out.print("2、第 1 条记录信息：<br/>");
```

```
out.print(rs.getString(1)+"\t"+rs.getString(2) + "\t" + rs.getString(3) + "\t" +
rs.getString(4)    + "<br/>");
out.print("<hr/>");

//随机输出一个记录
Int num = (int)((b-a)*Math.random()+a);
游标定位到表中第 num 条记录
rs.absolute(num);
```

5.逆序输出记录信息

```
（1）游标定位到表中末条记录之后
rs.afterLast();
（2）逆序输出记录信息
out.print("3、逆序输出记录信息：<br/>");
while (rs.previous()) {
     out.print(rs.getString(1) + "\t" + rs.getString(2) + "\t"+ rs.getString(3) + "\t" +
rs.getString(4) + "<br/>");
}
```

~~9.6 预处理 SQL 语句（选讲）~~

1.添加记录

（1）插入记录字符串

String insertstr = "INSERT INTO Student(学号,班级,姓名,性别) VALUES(?,?,?,?)";

（2）获取 SQL 预处理语句对象

PreparedStatement pstmt = conn.prepareStatement(insertstr);

（3）设置各字段值

pstmt.setString(1, "1");

pstmt.setString(2, "2");

pstmt.setString(3, "3");

pstmt.setString(4, "4");

（4）执行更新

int num = pstmt.executeUpdate();

2.编辑记录

（1）修改记录字符串

String editstr = "UPDATE Student SET 姓名 = ? WHERE 学号 = ?";

（2）获取 SQL 语句对象

PreparedStatement pstmt = conn.prepareStatement(editstr);

（3）设置各字段值

pstmt.setString(1, "韩梅梅");

pstmt.setString(2, "1");

（4）执行更新

int num = pstmt.executeUpdate();

3.删除记录

（1）删除记录字符串

String delstr = "DELETE FROM Student WHERE 学号 = ?";

（2）获取 SQL 语句对象

PreparedStatement pstmt = conn.prepareStatement(delstr);

（3）设置各字段值

pstmt.setString(1, "1");

（4）执行更新

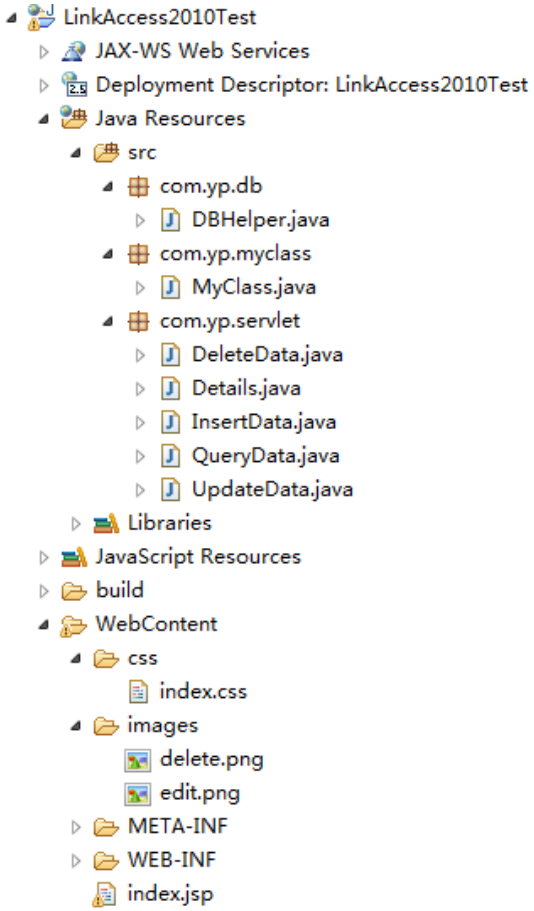
int num = pstmt.executeUpdate();

~~9.7 整合数据库编程代码（选讲）~~

1. 项目中文件列表

文件名	所在文件夹	文件作用	
index.jsp	WebContent	显示表中数据	执行/展示
index.css	WebContent\css	主页样式表文件	展示
delete.png	WebContent\images	删除图标	展示
edit.png	WebContent\images	编辑图标	展示
DBHelper.java	src\com.yp.db	操作数据库方法	展示
MyClass.java	src\ com.yp.myclass	自定义类（自动生成姓名）	

QueryData.java	src\ com.yp.servlet	查询数据类	
InsertData.java	src\ com.yp.servlet	插入数据类	
UpdateData.java	src\ com.yp.servlet	编辑数据类	
DeleteData.java	src\ com.yp.servlet	删除数据类	
Details.java	src\ com.yp.servlet	详细信息类	



2.DBHelper.java 文件代码

```
public class DBHelper {
（1）//驱动器字符串
private static String driver = "com.mysql.jdbc. Driver";
（2）//数据库连接地址、字符串、密码字符串
private static String uri = "jdbc:mysql://localhost:3306/ 数据库?characterEncoding=utf8 ";
private static String user = "root";
private static String password = "123456";
```

```
（3）//数据库连接方法
public static Connection getConnection() throws ClassNotFoundException, SQLException {
    Class.forName(driver);
     return DriverManager.getConnection(uri,user,password);
}
（4）//创建 SQL 语句
public static Statement getStatement() throws ClassNotFoundException, SQLException{  return getConnection().createStatement();}
（5）//获取结果集（查询记录）
public static ResultSet getResult(String sqlstr) throws ClassNotFoundException, SQLException{  return getStatement().executeQuery(sqlstr);}
（6）//插入记录
public static int insertData(String sqlstr) throws ClassNotFoundException, SQLException{  return getStatement().executeUpdate(sqlstr);}
（7）//编辑记录
public static int updateData(String sqlstr) throws ClassNotFoundException, SQLException{  return getStatement().executeUpdate(sqlstr);}
（8）//删除记录
public static int deleteData(String sqlstr) throws ClassNotFoundException, SQLException{  return getStatement().executeUpdate(sqlstr);}}
```

3. index.jsp 页面代码

```
<a href="InsertData">添加</a><hr/>
<%
    String sqlstr = "SELECT * FROM R133";
    ResultSet rs = DBHelper.getResult(sqlstr);
    out.print("<table border=1 cellspacing=1>");
    while (rs.next()) {
    String Rid = rs.getString(1);
```

```
        String StuName = rs.getString(2);
        out.print("<tr>");
        out.print("<td><a href='Details?Rid=" + Rid + "'>" + StuName
        + "</a></td>");
        out.print("<td><a href='UpdateData?Rid=" + Rid + "'>
         <img src='images/edit.png'></a></td>");
        out.print("<td><a href='DeleteData?Rid=" + Rid + "'>
         <img src='images/delete.png'></a></td>");
        out.print("</tr>");
            }
        out.print("<table>");
%>
```

4. MyClass.java 文件代码

```
public class MyClass {
    //随机生成姓名
    public static String getStuName() {
        String FamilyName[] = { "赵", "钱", "孙", "李" };
        int i = (int) (4 * Math.random());
        String familyName = FamilyName[i];
        String FirstName[] = { "卫", "忠", "建", "伟" };
        int j = (int) (4 * Math.random());
        String firstName = FirstName[j];
        String SecondName[] = { "强", "宝", "国", "庆" };
        int k = (int) (4 * Math.random());
        String secondName = SecondName[k];
        return familyName + firstName + secondName;
    }
}
```

5. 添加（InsertData.java）

```
protected void doGet(HttpServletRequest request,
                HttpServletResponse response) throws ServletException, IOException {
  response.setContentType("text/html;charset=gb2312");
  PrintWriter out = response.getWriter();
  //获取随机姓名
  String StuName = MyClass.getStuName();
  //定义 SQL 插入语句
  String sqlstr = "INSERT INTO R133(StuName) VALUES('" + StuName + "')";
  try {
    //调用 DBHelper 类的 insertData 方法
    DBHelper.insertData(sqlstr);
    //页面提示信息
    out.print("添加成功！ ");
    out.print(" <a href='index.jsp'>返回</a>");
  } catch (ClassNotFoundException e) {e.printStackTrace();}
    catch (SQLException e) {e.printStackTrace();      }
}
```

6.修改 Servlet

```
protected void doGet(HttpServletRequest request,
                HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=gb2312");
    PrintWriter out = response.getWriter();
    //获取前一页面传递参数
    int Rid = Integer.parseInt(request.getParameter("Rid"));
    //获取随机姓名
    String StuName = MyClass.getStuName();
    //定义 SQL 修改语句
    String sqlstr = "UPDATE R133 SET StuName = '" + StuName
                    + "' WHERE Rid = " + Rid;
```

```
    try {
        //调用 DBHelper 类的 updateData 方法
        DBHelper.updateData(sqlstr);
        //页面提示信息
        out.print("编辑成功！ ");
        out.print(" <a href='index.jsp'>返回</a>");
    } catch (ClassNotFoundException e) {e.printStackTrace();}
      catch (SQLException e) {e.printStackTrace();}
}
```

7. 删除 Servlet

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=gb2312");
    PrintWriter out = response.getWriter();
    //获取前一页面传递参数
    int Rid = Integer.parseInt(request.getParameter("Rid"));
    //定义 SQL 删除语句
    String sqlstr = "DELETE FROM R133 WHERE Rid = " + Rid;
    try {
        //调用 DBHelper 类的 deleteData 方法
        DBHelper.deleteData(sqlstr);
        //页面提示信息
        out.print("删除成功！ ");
        out.print(" <a href='index.jsp'>返回</a>");
    } catch (ClassNotFoundException e) {e.printStackTrace();}
      catch (SQLException e) {e.printStackTrace();}
}
```

8.查询 Servlet

```
protected void doGet(HttpServletRequest request,
```

```
            HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=gb2312");
    PrintWriter out = response.getWriter();
    //定义查询语句
    String sqlstr = "SELECT * FROM R133";
    try {
        //调用 DBHelper 类的 getResult 方法
        ResultSet rs = DBHelper.getResult(sqlstr);
        out.print("<b>姓名</b><br/>");
        //循环查询
        while (rs.next()) {
            out.print("<a href='Details?Rid=" + rs.getString(1) + "'>"
                    + rs.getString(2) + "</a><br/>");
        }
    } catch (ClassNotFoundException e) {    e.printStackTrace();}
            catch (SQLException e) {e.printStackTrace();}
}
```

9.详细信息代码

```
protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=gb2312");
    PrintWriter out = response.getWriter();
    //获取上一页面传递的 Rid 参数
    String Rid = request.getParameter("Rid");
    //定义查询语句
    String sqlstr = "SELECT * FROM R133 WHERE Rid = "+ Integer.parseInt(Rid);
    try {
        //调用 DBHelper 类的 getResult 方法
        ResultSet rs = DBHelper.getResult(sqlstr);
```

```
        out.print("详细信息：<br/>");
        //输出详细信息
        if (rs.next()) {
            out.print("序号：" + rs.getString(1) + "\t 姓名：" + rs.getString(2)+"<br/>");
        }
        out.print("<a href='index.jsp'>返回</a>");
    } catch (ClassNotFoundException e) {    e.printStackTrace();}
        catch (SQLException e) {e.printStackTrace();}
}
```