



实 验 伍建霖 学 20337251 日 期: 2022.11.24
人: 号:

式:

一. 实验目的

二. 实验内容和要求

三. 实验重点和难点

实验难点：不同的数据库应用开发工具具有不同的开发框架和模式。能够较为熟练地使用所选择的应用开发工具，是实现本实验的难点。

四. 实验工具

MySQL、SQL Server、Navicat、NetBeans IDE 8.2（或其他开发工具）

五. 实验准备

(一) 编写实验程序的基本目标

在本实验中，以 NetBeans IDE（或其他开发工具）、MySQL 和 SQLServer 数据库为例，实现一个完整的示例程序。该程序实现把 MySQL 数据源中的 university 数据库中的 instructor 表数据复制到 SQLServer 的 university 数据库中的 T72_instructor 表中。

（二） 下载 JDBC 驱动程序

在 Java 程序中要使用 JDBC 访问某个数据库，必须在 Java 项目中连接这个数据库系统所提供的 JDBC 驱动程序包。本实验要实现对 MySQL 数据库和 SQLServer 数据库的访问，因此要下载以下 JDBC 驱动程序：

（1） 下载连接 MySQL 数据库的 JDBC 驱动程序包（如：

mysql-connector-java-8.0.30.jar）

参阅：<https://www.runoob.com/java/java-mysql-connect.html>

（2） 下载连接 SQLserver 2019 数据库的 JDBC 驱动程序包（如：

mssql-jdbc-11.2.0.jre8.jar）

参阅：

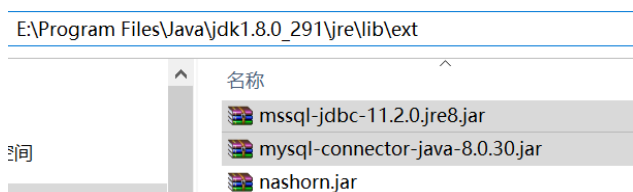
https://blog.csdn.net/qg_45687268/article/details/125017185

https://blog.csdn.net/m0_68693215/article/details/124788010

https://blog.csdn.net/qg_46110556/article/details/116765473

<https://learn.microsoft.com/zh-cn/sql/connect/jdbc/step-3-proof-of-concept-connecting-to-sql-using-java?view=sql-server-ver16>（注：对于参数 connectionUrl，有多个例子）

- **注：**为了使新建的 Java 程序能够直接使用下载的 JDBC 驱动程序包，可以将它复制到安装 JDK 的子目录（即：jdk\jre\lib\ext\）中，如：



(三) JDBC 程序的基本结构

(1) 基于 JDBC 驱动的数据库连接方法

```
/*step 1 加载驱动程序: Class.forName(driverName); */  
  
/* step 2 连接数据库 : Connection con = DriverManager.getConnection(dbURL,  
userName, userPwd); */
```

(2) 基于 JDBC 驱动的数据库数据操纵方法

```
/* step 3 创建 Statement/PreparedStatement 对象, 用来执行 sql 语句  
  
/* step 3.1 Statement stmt = con.createStatement(); */  
  
/* step 3.2 PreparedStatement ps=prepareStatement(sql); */  
  
/* step 4 给占位符赋值*/  
  
/* step 5 执行 sql 语句*/  
  
/* step 5.1 接收结果集: ResultSet rs = stmt.executeQuery(sqlStr); */  
  
/* step 5.2 更新操作: ps.executeUpdate(); */  
  
/* step 6 处理结果:遍历结果集*/
```

(3) 中断基于 JDBC 驱动的数据库连接

```
/* step 7 完成后关闭各个对象 */
```

六. 实验过程

1. 编写程序 T72A, 显示 MySQL 数据源中的 university 数据库中 instructor 表中的所有记录。

Step 1: 创建项目 T72A, 如图:



Step 2: 在程序文件T72A. java中输入以下代码:

```
package t72a;
import java.sql.*;
public class T72A {
    static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    static final String DB_URL =
"jdbc:mysql://localhost:3306/university?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC";
    // 数据库的用户名与密码，需要根据自己的设置
    static final String USER = "root";
    static final String PASS = "123456";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
            // step 1 加载驱动程序: Class.forName(driverName);
            Class.forName(JDBC_DRIVER);
            // step 2 获得数据库连接: Connection con =
            DriverManager.getConnection(dbURL, userName, userPwd);
            System.out.println("连接数据库..." + DB_URL);
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
            ResultSet rs;
            // step 3 创建Statement/PreparedStatement对象, 用来执行sql语句
            // step 3.1 创建Statement: Statement stmt = con.createStatement();
            // step 3.2 创建PreparedStatement对象
            stmt = conn.createStatement();
            //step 4 给占位符赋值
            //step 5 执行sql语句(接收结果集): ResultSet rs =
            stmt.executeQuery(sqlStr);
            String sql;
```

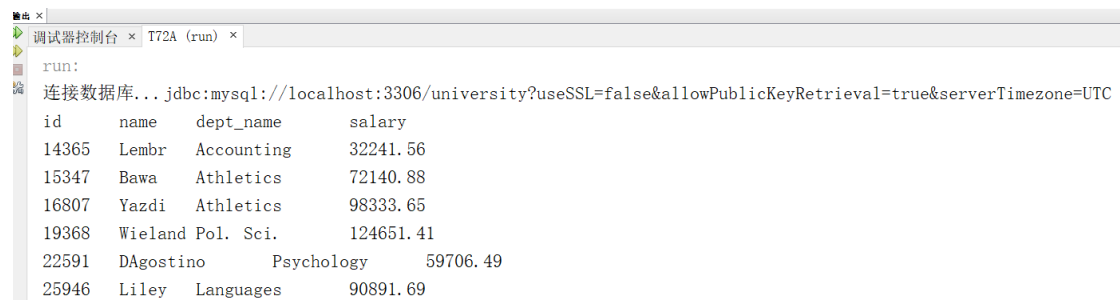
```

        sql = "select id,name,dept_name,salary from instructor";
        rs = stmt.executeQuery(sql);
        // step 6 处理结果:遍历结果集
        System.out.print("id\tname\tdept_name\tsalary\n");
        while(rs.next()){
            // 通过字段检索
            String inst_id = rs.getString("id");
            String inst_name = rs.getString("name");
            String dept_name = rs.getString("dept_name");
            Float salary = rs.getFloat("salary");
            // 输出数据
            System.out.print(inst_id);
            System.out.print("\t" + inst_name);
            System.out.print("\t" + dept_name);
            System.out.print("\t" + salary);
            System.out.print("\n");
        }
        // step 7 完成后关闭各个对象
        rs.close();
        stmt.close();
        conn.close();
    }catch(SQLException se){
        // 处理 JDBC 错误
        se.printStackTrace();
    }catch(Exception e){
        // 处理 Class.forName 错误
        e.printStackTrace();
    }finally{
        // 关闭资源
        try{
            if(stmt!=null) stmt.close();
        }catch(SQLException se2){
        }// 什么都不做
        try{
            if(conn!=null) conn.close();
        }catch(SQLException se){
            se.printStackTrace();
        }
    }
    System.out.println("Goodbye!");
}
}

```

Step 3: 运行将显示

验证截图如下：



The screenshot shows a Java IDE with a tab labeled 'T72A (run)'. The output window displays the following table:

id	name	dept_name	salary
14365	Lembr	Accounting	32241.56
15347	Bawa	Athletics	72140.88
16807	Yazdi	Athletics	98333.65
19368	Wieland	Pol. Sci.	124651.41
22591	D'Agostino	Psychology	59706.49
25946	Liley	Languages	90891.69

2. 编写程序 T72B，根据程序 T72A 设计函数

```
public static void showTableByJDBC(String JDBC_DRIVER,String DB_URL,String
USER,String PASS);
```

该函数显示由 JDBC_DRIVER 加载的数据源 DB_URL 中的 university 数据库中 instructor 表中的所有记录。

Step 1: 创建项目 T72B

Step 2: 在程序文件 T72B.java 中输入以下代码：

```
package t72b;
import java.sql.*;
public class T72B {
    public static void main(String[] args) {
        showTableByJDBC("com.mysql.cj.jdbc.Driver"
            , "jdbc:mysql://localhost:3306/university?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC"
            , "root", "123456");

        showTableByJDBC("com.microsoft.sqlserver.jdbc.SQLServerDriver"
            , "jdbc:sqlserver://localhost:1433;DatabaseName=university;"
            + "encrypt=true;"
            + "trustServerCertificate=true;"
            + "loginTimeout=30;"
            , "sa", "123456");
    }

    public static void showTableByJDBC(String JDBC_DRIVER, String DB_URL, String
    USER, String PASS) {
```

```
Connection conn = null;
Statement stmt = null;
try{
    // step 1 加载驱动程序: Class.forName(driverName);
    Class.forName(JDBC_DRIVER);
    // step 2 获得数据库连接 : Connection con =
DriverManager.getConnection(dbURL, userName, userPwd);
    System.out.println("连接数据库..." + DB_URL);
    conn = DriverManager.getConnection(DB_URL, USER, PASS);
    ResultSet rs;

    // step 3 创建Statement/PreparedStatement对象, 用来执行sql语句
    // step 3.1 创建Statement: Statement stmt = con.createStatement();
    // step 3.2 创建PreparedStatement对象
    stmt = conn.createStatement();

    //step 4 给占位符赋值
    //step 5 执行sql语句(接收结果集): ResultSet rs =
stmt.executeQuery(sqlStr);
    String sql;
    sql = "select id,name,dept_name,salary from instructor";
    rs = stmt.executeQuery(sql);
    // step 6 处理结果:遍历结果集
    System.out.print("id\tname\tdept_name\tsalary\n");
    while(rs.next()){
        // 通过字段检索
        String inst_id = rs.getString("id");
        String inst_name = rs.getString("name");
        String dept_name = rs.getString("dept_name");
        Float salary = rs.getFloat("salary");
        // 输出数据
        System.out.print(inst_id);
        System.out.print("\t" + inst_name);
        System.out.print("\t" + dept_name);
        System.out.print("\t" + salary);
        System.out.print("\n");
    }
    // step 7 完成后关闭各个对象
    rs.close();
    stmt.close();
    conn.close();
}catch(SQLException se){
    // 处理 JDBC 错误
```

```

        se.printStackTrace();
    }catch(Exception e){
        // 处理 Class.forName 错误
        e.printStackTrace();
    }finally{
        // 关闭资源
        try{
            if(stmt!=null) stmt.close();
        }catch(SQLException se2){
        }// 什么都不做
        try{
            if(conn!=null) conn.close();
        }catch(SQLException se){
            se.printStackTrace();
        }
    }
    System.out.println("Goodbye!");
}
}

```

Step 3: 运行将显示

验证截图如下:

问题	输出	调试控制台	终端
72553	Yin	English	46397.59
73623	Sullivan	Elec. Eng.	90038.09
74420	Voronina	Physics	121141.99
74426	Kenje	Marketing	106554.73
77346	Mahmoud	Geology	99382.59
78699	Pingr	Statistics	59303.62
79081	Ullman	Accounting	47307.1
79653	Levine	Elec. Eng.	89805.83
80759	Queiroz	Biology	45538.32
81991	Valtchev	Biology	77036.18
90376	Bietzk	Cybernetics	117836.5
90643	Choll	Statistics	57807.09
95030	Arinb	Statistics	54805.11
95709	Sakurai	English	118143.98
96895	Mird	Marketing	119921.41
97302	Bertolino	Mech. Eng.	51647.57
99052	Dale	Cybernetics	93348.83
Goodbye!			
(base) PS D:\CodeField\Database\lab\7\72>			

3. 编写程序 T72C, 设计函数

```
public static void copyTableByJDBC(String src_JDBC_DRIVER, String
src_DB_URL, String src_USER, String src_PASS,
String target_JDBC_DRIVER, String target_DB_URL, String target_USER, String
target_PASS )
```

该函数功能是：

- 先在目标数据源 target_DB_URL 创建 T72_instructor 表（注：与 instructor 表结构一样）中，
- 然后将源数据源 src_DB_URL 中 instructor 表中的所有记录复制到目标数据源 target_dsn 中的 T72_instructor 表中。

Step 1: 创建项目 T72C

Step 2: 在程序文件 T72C.java 中输入以下代码：

```
package t72c;
import java.sql.*;
public class T72C {
    public static void main(String[] args) {
        copyTableByJDBC("com.mysql.cj.jdbc.Driver"
            , "jdbc:mysql://localhost:3306/university?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC"
            , "root", "123456",
            "com.microsoft.sqlserver.jdbc.SQLServerDriver"
            , "jdbc:sqlserver://localhost:1433;DatabaseName=university;"
            + "encrypt=true;"
            + "trustServerCertificate=true;"
            + "loginTimeout=30;"
            , "sa", "123456");
        copyTableByJDBC("com.microsoft.sqlserver.jdbc.SQLServerDriver"
            , "jdbc:sqlserver://localhost:1433;DatabaseName=university;"
            + "encrypt=true;"
            + "trustServerCertificate=true;"
            + "loginTimeout=30;"
            , "sa", "123456"
            , "com.mysql.cj.jdbc.Driver"
            , "jdbc:mysql://localhost:3306/university?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC"
```

```

        , "root", "123456"
    );
}

    public static void copyTableByJDBC(String src_JDBC_DRIVER, String
src_DB_URL, String src_USER, String src_PASS,
    String target_JDBC_DRIVER, String target_DB_URL, String target_USER, String
target_PASS )
    {
        Connection src_conn = null;
        Statement src_stmt = null;
        Connection target_conn = null;
        PreparedStatement target_ps=null;
        try{
            System.out.println("连接数据库..." +src_DB_URL);
            System.out.println("连接数据库..." +target_DB_URL);
            // step 1 加载驱动程序: Class.forName(driverName);
            Class.forName(src_JDBC_DRIVER);
            Class.forName(target_JDBC_DRIVER);
            // step 2 连接数据库 : Connection con =
DriverManager.getConnection(dbURL, userName, userPwd);
            src_conn =
DriverManager.getConnection(src_DB_URL, src_USER, src_PASS);
            target_conn =
DriverManager.getConnection(target_DB_URL, target_USER, target_PASS);
            ResultSet src_rs;
            // step 3 创建Statement/PreparedStatement对象, 用来执行sql语句
            // step 3.1 创建Statement: Statement stmt = con.createStatement();
            // step 3.2 创建PreparedStatement对象: PreparedStatement
ps=prepareStatement(sql);
            src_stmt = src_conn.createStatement();
            target_ps = target_conn.prepareStatement("create table T72_instructor
(ID varchar(5), name varchar(20) not null, dept_name varchar(20), salary
numeric(8,2) )");
            target_ps.executeUpdate();

            target_ps = target_conn.prepareStatement("INSERT INTO
T72_instructor(ID,name,dept_name, salary) VALUES(?, ?, ?, ?)");

            //step 5 执行sql语句(接收结果集): ResultSet rs =
stmt.executeQuery(sqlStr);
            String select_sql;
            select_sql = "select id,name,dept_name,salary from instructor";

```

```

src_rs = src_stmt.executeQuery(select_sql);
// step 6 处理结果:遍历结果集
System.out.print("id\tname\tdept_name\tsalary\n");
while(src_rs.next()){
    // 通过字段检索
    String inst_id = src_rs.getString("id");
    String inst_name = src_rs.getString("name");
    String dept_name = src_rs.getString("dept_name");
    Float salary = src_rs.getFloat("salary");
    // 输出数据
    System.out.print(inst_id);
    System.out.print("\t" + inst_name);
    System.out.print("\t" + dept_name);
    System.out.print("\t" + salary);
    System.out.print("\n");

    //step 4 给占位符赋值
    target_ps.setString(1, inst_id); //给第1个占位符赋值(ID)
    target_ps.setString(2, inst_name); //给第2个占位符赋值(name)
    target_ps.setString(3, dept_name); //给第3个占位符赋值
(dept_name)
    target_ps.setFloat(4, salary); //给第4个占位符赋值(salary)
//step 5.2 执行sql语句(更新操作): ps.executeUpdate();
    target_ps.executeUpdate();
}

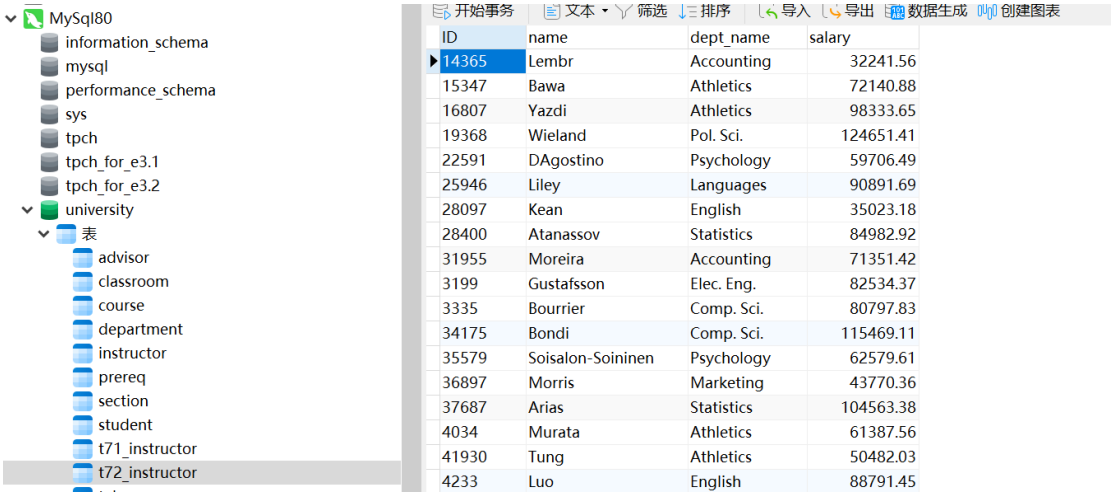
// step 7 完成后关闭各个对象
src_rs.close();
src_stmt.close();
src_conn.close();

target_ps.close();
target_conn.close();
}catch(SQLException se){
    // 处理 JDBC 错误
    se.printStackTrace();
}catch(Exception e){
    e.printStackTrace();
}
System.out.println("Goodbye!");
}
}

```

Step 3: 运行将显示

验证截图 1: 在 My SQL 数据库中的表 T72_instructor



The screenshot shows the MySQL 8.0 graphical user interface. On the left, the 'university' database is selected, and the 'T72_instructor' table is highlighted under the '表' (Tables) folder. The main window displays the table's data with columns: ID, name, dept_name, and salary. The table contains 20 rows of instructor data.

ID	name	dept_name	salary
14365	Lembr	Accounting	32241.56
15347	Bawa	Athletics	72140.88
16807	Yazdi	Athletics	98333.65
19368	Wieland	Pol. Sci.	124651.41
22591	D'Agostino	Psychology	59706.49
25946	Liley	Languages	90891.69
28097	Kean	English	35023.18
28400	Atanassov	Statistics	84982.92
31955	Moreira	Accounting	71351.42
3199	Gustafsson	Elec. Eng.	82534.37
3335	Bourrier	Comp. Sci.	80797.83
34175	Bondi	Comp. Sci.	115469.11
35579	Soisalon-Soininen	Psychology	62579.61
36897	Morris	Marketing	43770.36
37687	Arias	Statistics	104563.38
4034	Murata	Athletics	61387.56
41930	Tung	Athletics	50482.03
4233	Luo	English	88791.45

验证截图 2: 在 SQL Server 数据库中的表 T72_instructor

对象	T72_instructor @university.dbo (SQL...		
开始事务	文本	筛选	排序
导入	导出	数据	
ID	name	dept_name	salary
14365	Lembr	Accounting	32241.56
15347	Bawa	Athletics	72140.88
16807	Yazdi	Athletics	98333.65
19368	Wieland	Pol. Sci.	124651.41
22591	D'Agostino	Psychology	59706.49
25946	Liley	Languages	90891.69
28097	Kean	English	35023.18
28400	Atanassov	Statistics	84982.92
31955	Moreira	Accounting	71351.42
3199	Gustafsson	Elec. Eng.	82534.37
3335	Bourrier	Comp. Sci.	80797.83
34175	Bondi	Comp. Sci.	115469.11
35579	Soisalon-Soininen	Psychology	62579.61
36897	Morris	Marketing	43770.36
37687	Arias	Statistics	104563.38
4034	Murata	Athletics	61387.56
41930	Tung	Athletics	50482.03
4233	Luo	English	88791.45
42782	Vicentino	Elec. Eng.	34272.67
43779	Romero	Astronomy	79070.08
48507	Lent	Mech. Eng.	107978.47
48570	Sarkar	Pol. Sci.	87549.80
50330	Shuming	Physics	108011.81
50885	Konstantinides	Languages	32570.50
52647	Bancilhon	Pol. Sci.	87958.01
57180	Hau	Accounting	43966.29
58558	Dusserre	Marketing	66143.25

+ - ✓ ✕ ↺ ■

SELECT *, 0 AS _NAV_ORDER_F_ FROM [dbo].[T72_instructor] ORDER BY _I

七. 与实验结果相关的文件

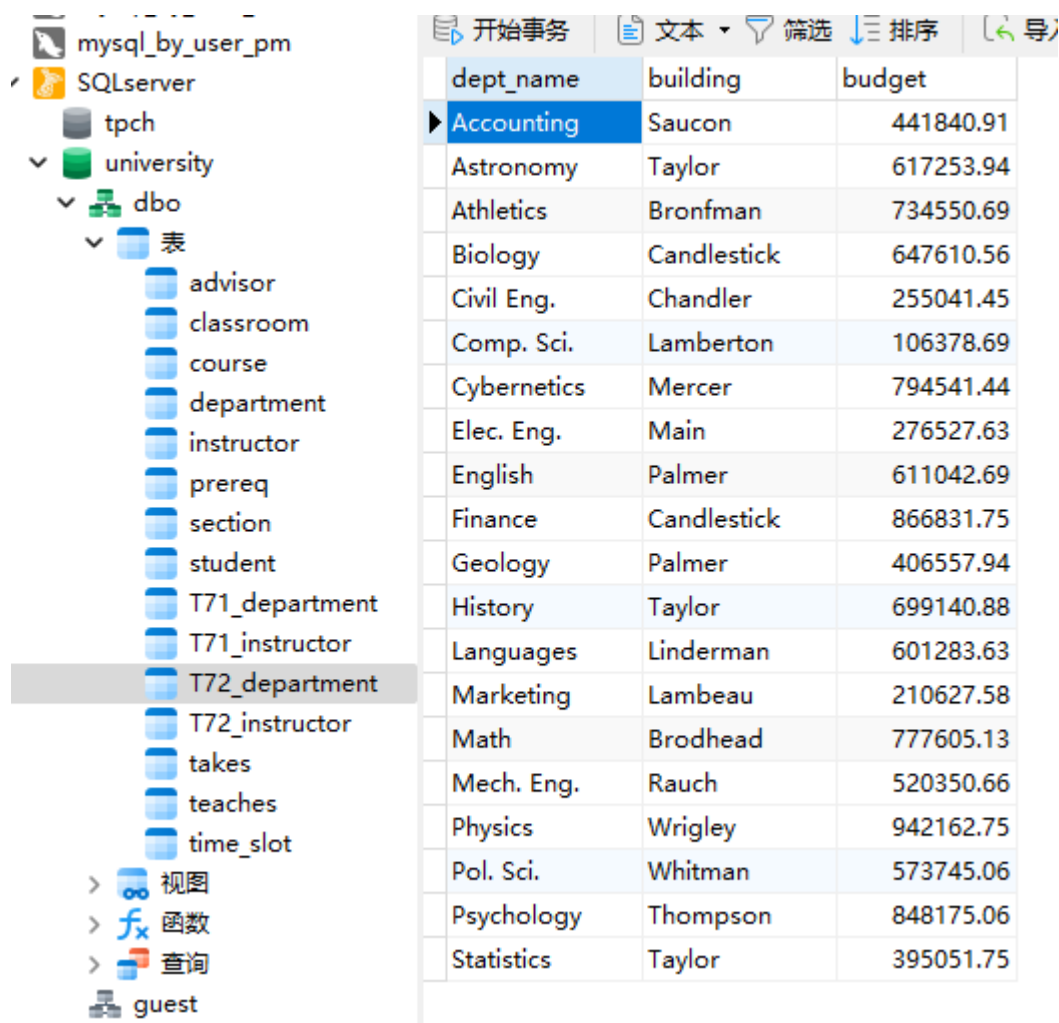
- 程序 T72A 文件
- 程序 T72B 文件
- 程序 T72C 文件

- 程序 T72More1 文件

八. 思考题

(1) 尝试编写程序 T72More1 (注: 参考以上程序), 实现将数据库 university 中的表 department 从数据源 university_in_mysql80 复制到数据源 university_in_sqlserver 中的 T72_department。

答:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'university' database is expanded, showing the 'dbo' schema and a list of tables. The 'T72_department' table is highlighted. On the right, a query result table is displayed with the following data:

dept_name	building	budget
Accounting	Saucon	441840.91
Astronomy	Taylor	617253.94
Athletics	Bronfman	734550.69
Biology	Candlestick	647610.56
Civil Eng.	Chandler	255041.45
Comp. Sci.	Lamberton	106378.69
Cybernetics	Mercer	794541.44
Elec. Eng.	Main	276527.63
English	Palmer	611042.69
Finance	Candlestick	866831.75
Geology	Palmer	406557.94
History	Taylor	699140.88
Languages	Linderman	601283.63
Marketing	Lambeau	210627.58
Math	Brodhead	777605.13
Mech. Eng.	Rauch	520350.66
Physics	Wrigley	942162.75
Pol. Sci.	Whitman	573745.06
Psychology	Thompson	848175.06
Statistics	Taylor	395051.75

```
import java.sql.*;

public class T72More1 {
    public static void main(String[] args) {
        copyTableByJDBC("com.mysql.cj.jdbc.Driver",
```

```

        "jdbc:mysql://localhost:3306/university?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC",
        "root", "dsbdsb",
        "com.microsoft.sqlserver.jdbc.SQLServerDriver",
        "jdbc:sqlserver://localhost:1433;DatabaseName=university";
    }

    + "encrypt=true;"
    + "trustServerCertificate=true;"
    + "loginTimeout=30;",
    "sa", "dsbdsb");
    //
copyTableByJDBC("com.microsoft.sqlserver.jdbc.SQLServerDriver"
    // , "jdbc:sqlserver://localhost:1433;DatabaseName=university;"
    // + "encrypt=true;"
    // + "trustServerCertificate=true;"
    // + "loginTimeout=30;"
    // , "sa", "dsbdsb"
    // , "com.mysql.cj.jdbc.Driver"
    // , "jdbc:mysql://localhost:3306/university?useSSL=false&allowPu
blicKeyRetrieval=true&serverTimezone=UTC"
    // , "root", "dsbdsb"
    // );
}

public static void copyTableByJDBC(String src_JDBC_DRIVER, String
src_DB_URL, String src_USER, String src_PASS,
    String target_JDBC_DRIVER, String target_DB_URL, String
target_USER, String target_PASS) {
    Connection src_conn = null;
    Statement src_stmt = null;
    Connection target_conn = null;
    PreparedStatement target_ps = null;
    try {

        System.out.println("连接数据库..." + src_DB_URL);
        System.out.println("连接数据库..." + target_DB_URL);
        // step 1 加载驱动程序: Class.forName(driverName);
        Class.forName(src_JDBC_DRIVER);
        Class.forName(target_JDBC_DRIVER);
        // step 2 连接数据库 : Connection con =
        DriverManager.getConnection(dbURL, userName,
            // userPwd);
    }
}

```

```

        src_conn = DriverManager.getConnection(src_DB_URL, src_USER,
src_PASS);
        target_conn = DriverManager.getConnection(target_DB_URL,
target_USER, target_PASS);
        ResultSet src_rs;

        // step 3 创建 Statement/PreparedStatement 对象, 用来执行 sql 语
句
        // step 3.1 创建 Statement: Statement stmt =
con.createStatement();
        // step 3.2 创建 PreparedStatement 对象: PreparedStatement
ps=prepareStatement(sql);
        src_stmt = src_conn.createStatement();
        target_ps = target_conn.prepareStatement(
            "create table T72_department (dept_name varchar(20)
not null, building varchar(15), budget numeric(12,2))");
        target_ps.executeUpdate();

        target_ps = target_conn
            .prepareStatement("INSERT INTO T72_department
(dept_name, building, budget) VALUES(?, ?, ?)");

        // step 5 执行 sql 语句(接收结果集): ResultSet rs =
stmt.executeQuery(sqlStr);
        String select_sql;
        select_sql = "select dept_name, building, budget from
department";
        src_rs = src_stmt.executeQuery(select_sql);

        // step 6 处理结果:遍历结果集
        System.out.print("dept_name\tbuilding\tbudget\n");
        while (src_rs.next()) {
            // 通过字段检索
            String dept_name = src_rs.getString("dept_name");
            String dept_building = src_rs.getString("building");
            Float dept_budget = src_rs.getFloat("budget");
            // 输出数据
            System.out.print(dept_name);
            System.out.print("\t" + dept_building);
            System.out.print("\t" + dept_budget);
            System.out.print("\n");

            // step 4 给占位符赋值

```



```

        target_ps.setString(1, dept_name); // 给第 3 个占位符赋值
(dept_name)
        target_ps.setString(2, dept_building); // 给第 1 个占位符赋
值(ID)
        target_ps.setFloat(3, dept_budget); // 给第 4 个占位符赋值
(salary)

        // step 5.2 执行 sql 语句(更新操作): ps.executeUpdate();
        target_ps.executeUpdate();
    }

    // step 7 完成后关闭各个对象
    src_rs.close();
    src_stmt.close();
    src_conn.close();

    target_ps.close();
    target_conn.close();
} catch (SQLException se) {
    // 处理 JDBC 错误
    se.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
System.out.println("Goodbye!");
}
}

```

(2) 请调查目前比较流行的软件开发环境在基于 JDBC 驱动开发数据库应用方面各有哪些优缺点?

答: 现在常用的 java 开发工具有 eclipse, myeclipse, netbeans, jdk 等等。

eclipse, myeclipse, netbeans 可跨平台, jdk 简单易学。

九. 实验总结