

死记硬背：

Java连接数据库基本步骤：

1. 加载 j d b c 驱动程序：

```
Class.forName("COM.ibm.db2.jdbc.app.DB2Driver");
```

2. 创建数据库的连接

```
String url = "jdbc:db2:sample";//欲连接的数据库路径
```

```
String userid=" db2admin" ;//用户名
```

```
String passwd=" db2admin" ;//密码
```

```
Connection sample=DriverManager.getConnection(url,userid,passwd);
```

3. 写sql语句，并创建一个 PreparedStatement/Statement语句，并执行存入ResultSet结果集。

知识：用到的sql语句分两类，查询，更新，删除，插入

Sql语句如果是select查询类型的，查询之后的行数列数内容就用ResultSet存储。

Sql如果是插删改的话，返回值是插入/修改/删除的行数。

ResultSet executeQuery(String sqlString)：执行查询数据库的SQL语句，返回一个结果集 (ResultSet) 对象。

int executeUpdate(String sqlString)：用于执行INSERT、UPDATE或删除DELETE语句以及SQL DDL语句，如：CREATE TABLE和DROP TABLE等

1.PreparedStatement：//一般用于sql语句有？的，不可以直接执行的语句。

```
String sql="insert into templ(empno,firstnme,lastname,edlevel) values (?,?,?,?)";//sql语句
```

```
PreparedStatement pstmt = sample.prepareStatement(sql);//创建PreparedStatement语句
```

```
pstmt.setString(1,deptno);//记住，从1开始，不是从0开始set，规定。deptno是一个字符串，自己定义即可。
```

```
pstmt.setString(2,deptno);
```

```
pstmt.setString(3,deptno);
```

```
pstmt.setString(4,deptno);
```

```
int rows=pstmt.executeUpdate();//返回更新过的行数。
```

```
// 插入数据
```

```
String sql = null;
```

```
sql = "INSERT INTO TEMPL (EMPNO, LASTNAME, SALARY)" + "VALUES ('000110', 'Roth', 50000)",
```

```
stmt = con.createStatement();
```

```
insertCount = stmt.executeUpdate( sql );
```

```
// 删除数据
```

```
PreparedStatement pstmt = null;
```

```
Stringeno="000110";
```

```

Stringmysql=null;
mysql= "DELETE FROM TEMPLWHEREEMPNO=?";
pstmt=con.prepareStatement(mysql);
pstmt.setString(1,eno);

deleteCount = pstmt.executeUpdate();


// 更新数据库
Statement updStmt = con.createStatement(); int numRows = upd-
Stmt.executeUpdate

        ("UPDATE TEMPL " +
         " SET LASTNAME = 'Stohl' " +
         " WHERE EMPNO = '000110' ");
System.out.println("Number of rows updated " + numRows);

```

4. 这一步，如果是插删改的话，不需要了，如果是查询语句的话，那么就要遍历结果集ResultSet，将查到的内容提取出来。

2.Statement：//一般用于sql语句没有？的，可以直接执行的语句

```

String sql ="select NAME, JOB, SALARY from staff Where ID = 10" ; //
sql语句
Statement stmt = sample.createStatement();//创建statement语句
ResultSet rs = stmt.executeQuery( sql);//执行，并存入结果集
语句：
while(rs.next()){
    String data1 = rs.getString(1); //从1开始
    String data2 = rs.getString(2);
    String data3 = rs.getString(3);
    String data4= rs.getString(4);
}

```

结果集有很多库函数，getString最常用，指取出的那一列属性的是字符串类型，

如果那一列属性是整数类型，就用getInt

5. 处理异常，关闭jdbc对象资源

操作完成以后要把所有使用的JDBC对象全都关闭，以释放JDBC资源，关闭顺序和声

明顺序相反：

- 1、先关闭ResultSet
- 2、再关闭preparedStatement
- 3、最后关闭连接对象connection

```
if(rs != null){ // 关闭记录集
try{
    rs.close();
}catch(SQLException e){
    e.printStackTrace();
}
}
if(stmt != null){ // 关闭声明
try{
    stmt.close();
}catch(SQLException e){
    e.printStackTrace();
}
}
if(conn != null){ // 关闭连接对象
try{
    conn.close();
}catch(SQLException e){
    e.printStackTrace();
}
}
```

```
sampleconn.setAutoCommit(false);
```

```
pstmt = con.prepareStatement( sql );
```

```
pstmt.setNull (1, java.sql.Type.SMALLINT);
```

rs是结果集

```
rs.wasNull()
```

实验报告知识点：

任务1：没东西

任务2：没东西

任务3：没东西

任务4：

```
String inputValue = JOptionPane.showInputDialog("Please input a value");//输入的东西保存在inputValue
```

```
JOptionPane.showMessageDialog(null, "\nNumber of rows updated: " + updateCount);//弹出一个输出框
```

任务5：表格的相关操作，创建表格

```
String[] Names = { "编号", "名字", "姓氏", "职位"};//表头
```

```
Object[][] Info = new Object[4][4];//表格内容，几行几列
```

```
JTable table = new JTable(playerInfo, Names);//创建表格
```

表格主要的库函数：

1. 取单元格数据

```
String cellValue=(String) tableModel.getValueAt(row, column);// 取单元格数据,row是行号,column是列号
```

2. 给单元格数据赋值，即给二维数组赋值，一般用ResultSet结果集赋值上面的Info

```
for(int i=0;i<4;i++)
```

```
for(int j=0;j<4;j++)
```

```
{
    System.out.println(j+"ci");
    info[i][j]=rs.getString(j+1);
}
}
```

单行插入 `String single_insert="insert into templ(empno,firstnme,lastname,edlevel) values (?, ?, ?, ?);"`

多行插入 `int count =4 ; //用户想插入四行`

```
String sql="insert into templ(empno,firstnme,lastname,edlevel) values (?, ?, ?, ?);"
```

```
for(int i=0;i<3;i++)
```

```
{
    sql=sql+",(?, ?, ?, ?)";
}
}
```

通过子查询插入 `String find_insert="insert into templ(empno,firstnme,lastname,edlevel) select empno,firstnme,lastname,edlevel from employee where empno=? and firstnme=? and lastname=? and edlevel=?";`

任务6：ResultSet的wasNull()方法

表示它的下一列是不是空

任务7：目前不知道

任务8：

第一问:没东西

第二问：实现对结果集的任意行，任意列的修改

```
* 方法一： String finalstr="";
*          String selectforupdate="select
empno,firstnme,lastname,edlevel,job from employee"
* +" for update";//sql语句，不要忘记加上for update！！
int finalrow=0;
int finalcolmn=0;
* int r= table.getSelectedRow();//得到表格被选中的行数，与结
果集对应 finalrow=r;
int c=table.getSelectedColumn();//得到表格被选中的列数，与结果集
对应 finalcolmn=c;
*
```

```
stmt=sample.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,Res
ultSet.CONCUR_UPDATABLE); //创建可滚动，可更新的结果集
* ResultSet select_rs=stmt.executeQuery(selectforupdate);
* for(int i=0;i<finalrow;i++)
* {
* select_rs.next();//结果集移动到第finalrow行
* }
* if(select_rs.next())
* {
* select_rs.updateObject(finalcolmn,finalstr);//更新结果集的
第finalcolumn列，更新为finalstr
* }
* select_rs.updateRow();//将结果集落实到数据库
*
```

```
* 方法二： * String finalstr="";
*          String selectforupdate="select
empno,firstnme,lastname,edlevel,job from employee"
* +" for update";//sql语句，不要忘记加上for update！！ int
finalrow=0; int finalcolmn=0;
* int r= table.getSelectedRow();//得到表格被选中的行数，与结
果集对应
* finalrow=r;
* int c=table.getSelectedColumn();//得到表格被选中的列数，
与结果集对应
* finalcolmn=c;
* stmt=sample.createStatement(ResultSet.TYPE_SCROLL_IN-
```

SENSITIVE,ResultSet.

```
* CONCUR_UPDATABLE);
* ResultSet select_rs=stmt.executeQuery(selectforupdate);
* select_rs.absolute(finalrow+1); //直接定位到第finalrow+1行
* select_rs.updateObject(finalcolmn+1,finalstr);
* select_rs.updateRow();
*
*/
```

任务9：

Section3.2：创建可滚动的结果集，使用last，previous，first函数遍历

String sql="select NAME,JOB,SALARY from staff";

P r e p a r e d S t a t e m e n t

stmt=sample.prepareStatement(sql,rs.TYPE_SCROLL_INSENSITIVE,rs.CONCUR_READ_ONLY); //为了创建可滚动，只读的结果集

ResultSet rs = stmt.executeQuery();

rs.last();

rs.previous();

rs.first();

知识：

rs.absolute()//绝对位置，负数表示从后面数

rs.first()第一条

rs.last()最后一条

rs.previous()前一条

rs.next()后一条

Section3.3：批处理操作，同时执行多条语句

Statement接口里有两个方法：

void addBatch(String sql)

将给定的 SQL 命令添加到此 Statement 对象的当前命令列表中。通过调用方法 executeBatch 可以批量执行此列表中的命令。

int[] executeBatch()

将一批命令提交给数据库来执行，如果全部命令执行成功，则返回更新计数组成的数组。

返回：

包含批中每个命令的一个元素的更新计数所组成的数组（数组中的每个元素为：成功处理了命令后，执行命令所影响数据库中行数的更新计数）。数组的元素根据将命令添加到批中的顺序排序。

批处理操作，即同时执行多条语句

代码：

Statement stmt=sample.createStatement();

stmt.addBatch("INSERT INTO DB2ADMIN.DEPARTMENT " +

```

        "VALUES ('BT6','BATCH6 NEWYORK','BBBBB1','BTT','NEW
YORK CITY6')");
stmt.addBatch("INSERT INTO DB2ADMIN.DEPARTMENT " +
        "VALUES ('BT7','BATCH7 NEWYORK','BBBBB2','BT2','NEW
YORK CITY7')");
int []updateCounts=stmt.executeBatch();
sample.commit();

```

Section3.4：取数据库结构

DatabaseMetaData dbmd=sample.getMetaData();//获取数据库的结构，存储等信息。

```
String []tableTypes={"TABLE","VIEW"};
```

```
ResultSet rs=dbmd.getTables(null,"UDBA","%",tableTypes);
```

知识： dbmd.getTables (String catalog,String schema,String tableName,String[] types) ,

这个方法带有四个参数，它们表示的含义如下：

String catalog：要获得表所在的编目。" "" "意味着没有任何编目，Null表示所有编目。

String schema：要获得表所在的模式。" "" "意味着没有任何模式，Null表示所有模式。

String tableName：指出要返回表名与该参数匹配的那些表。

String types：一个指出返回何种表的数组。

可能的数组项是："TABLE"、"VIEW"、"SYSTEM TABLE"，"GLOBAL TEMPORARY"，"LOCAL TEMPORARY"，"ALIAS"，"SYNONYM"。

通过getTables () 方法返回的结果集中的每个表都有下面是10字段的描述信息，而且只有10个。通常我们用到的也就是标红的几个字段。而且在结果集中直接使用下面字段前面的序号即可获取字段值。

1.TABLE_CAT (String) => 表所在的编目(可能为空)

2.TABLE_SCHEM (String) => 表所在的模式(可能为空)

3.TABLE_NAME (String) => 表的名称

任务10：blob类型对象的读取和插入

Blob类型的赋值，用setBinaryStream(int n, java.io.InputStream x, int length)类型赋值，n指示参数编号的 int。InputStream 对象。length指示字节数的 int。

查询图片代码：

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

```
ResultSet rs=st.executeQuery("select image from employee");
```

```
while(rs.next())
```

```

{
//读取Blob对象
Blob blob= (Blob) rs.getBlob(1);
//Blob对象转化为InputStream流
java.io.InputStream inputStream =blob.getBinaryStream();
//要写入的文件
File fileOutput = new File("c:/backa.jpg");
//文件的写入流的定义
FileOutputStream fo = new FileOutputStream(fileOutput);
int c;
//读取流并写入到文件中
while ((c = inputStream.read()) != -1)
fo.write(c);
//流的关闭:
fo.close();
}

```

插入图片代码：

```

PreparedStatement preparedStatement = conn
    .prepareStatement("insert into emp_photo values('000130','jpeg',?)");
//创建文件对象:
File file=new File("e:/123.jpg");//参数是本地图片的路径名
// 创建流对象:
BufferedInputStream imageInput = new BufferedInputStream(
    new FileInputStream(file));
//参数赋值:
preparedStatement.setBinaryStream(1, imageInput,(int) file.length());//
第二个参数InputStream 对象，第三个参数字节数int
//执行语句
preparedStatement.executeUpdate();

```

任务11：clob类型的读取

Clob resumelob = null;//定义clob类型

resumelob = rs3.getClob(3);//将结果集中第三个位置的clob取出


```
String detailinfo = resumelob.getSubString((long)1,  
(int)resumelob.length());//将clob类型转化为字符串类型，参数固定，基本上  
就这么写
```

还用到的sql函数

POSSTR(RESUME,'Personal')//查看Personal这个单词在RESUME中的位置

SUBSTR(RESUME,1,length)//取resume中1-length长度的字段