

JDBC:

- 面向应用的API，供开发人员调用。
- 面向数据库的API，供数据库开发厂商开发对应的数据库驱动。

具体开发步骤

- 加载数据库驱动，Java程序和数据库的桥梁。
- 获取Connection，表示Java程序与数据库的一次连接，每执行一个操作都必须建立在一次连接上，Connection 通过 DriverManager 来产生。
- 通过Statement来执行SQL语句，Statement通过Connection产生。
- ResultSet，当执行查询操作时需要用ResultSet来接收结果集，将数据库中的数据转行成Java对象，并且是用集合来存储的。如果执行的是新增，删除，修改操作就不需要用到ResultSet，返回的是int类型的数据，表示对于数据库数据影响的行数。

```
Class.forName("com.mysql.cj.jdbc.Driver");
String url = "jdbc:mysql://localhost:3306/dbname";
String user = "root";
String pwd = "root";
Connection connection = DriverManager.getConnection(url,user,pwd);
//查询
String sql = "select * from user";
Statement statment = connection.createStatement();
ResultSet resultSet = statement.executeQuery(sql);
while(resultSet.next()){
    ...
}
//新增、修改、删除
String sql = "delete from user where id = 1";
Statement statement = connection.createStatement();
int row = statement.executeUpdate(sql);

resultSet.close();
statement.close();
connection.close();
```

实际开发中一般不会使用Statment来执行SQL语句，而是选择其子类 PreparedStatement 来执行SQL。

PreparedStatement 提供了SQL占位符的功能，定义SQL语句时可以先使用?来标识要被替换的参数，而不需要去拼接字符串。

PreparedStatement 首先可以解决频繁拼接SQL字符串的问题。

PreparedStatement 还可以避免SQL注入的风险。

SQL注入：利用某些系统没有对用户输入的数据进行充分检查，在用户输入的数据中注入非法的SQL语句，从而利用系统漏洞完成恶意行为的做法。

PreparedStatement 执行效率更高。

将图片保存到数据库中

原理：将图片转换成二进制流，然后将二进制流保存到数据库中。

要求存储图片的字段的数据类型为二进制类型。

MySQL中共有4种二进制的数据类型（除了存储的最大信息量不同外，没有区别）

blob 存储上限是65KB

tinyblob 存储上限是255KB

mediumblob 存储上限是16MB

longblob 存储上限是4GB

- 在数据表中添加blob类型的字段
- 通过Java程序完成图片的保存：使用InputStream将图片以流的形式读取到Java程序中，再通过JDBC将二进制数据存入数据库。

```
Connection connection = null;
PreparedStatement preparedStatement = null;
ResultSet resultSet = null;
InputStream inputStream = null;
try {
    //通过InputStream读入图片
    inputStream = new FileInputStream("src/1.png");
    Class.forName("com.mysql.cj.jdbc.Driver");
    String url = "jdbc:mysql://localhost:3306/test";
    String user = "root";
    String password = "root";
    connection = DriverManager.getConnection(url, user, password);
    String sql = "insert into t_user(username,img) values(?,?)";
    preparedStatement = connection.prepareStatement(sql);
    preparedStatement.setString(1,"图片");
    preparedStatement.setBlob(2,inputStream);
    int row = preparedStatement.executeUpdate();
    if(row == 1){
        System.out.println("图片保存成功！");
    }else{
        System.out.println("图片保存失败！");
    }
} catch (Exception e){
```

```

        e.printStackTrace();
    } finally {
        try {
            connection.close();
            preparedStatement.close();
            inputStream.close();
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

从数据库中读取二进制数据转成图片。

```

Connection connection = null;
PreparedStatement preparedStatement = null;
ResultSet resultSet = null;
InputStream inputStream = null;
OutputStream outputStream = null;
try {
    //通过IupterStream读入图片
    inputStream = new FileInputStream("src/1.png");
    Class.forName("com.mysql.cj.jdbc.Driver");
    String url = "jdbc:mysql://localhost:3306/test";
    String user = "root";
    String password = "root";
    connection = DriverManager.getConnection(url, user, password);
    String sql = "select img from t_user where id = ?";
    preparedStatement = connection.prepareStatement(sql);
    preparedStatement.setInt(1,17);
    resultSet = preparedStatement.executeQuery();
    if(resultSet.next()){
        Blob blob = resultSet.getBlob(1);
        inputStream = blob.getBinaryStream();
        outputStream = new FileOutputStream("src/copy.png");
        int temp = 0;
        while ((temp = inputStream.read())!=-1){
            outputStream.write(temp);
        }
    }
} catch (Exception e){
    e.printStackTrace();
} finally {
    try {
        connection.close();
        preparedStatement.close();
    }
}

```

```

        resultSet.close();
        inputStream.close();
        outputStream.close();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

JDBCTools:

```

package com.southwind.util;

import java.sql.*;

public class JDBCTools {
    private static String driver = "com.mysql.cj.jdbc.Driver";
    private static String url = "jdbc:mysql://localhost:3306/test";
    private static String username = "root";
    private static String password = "root";

    static {
        try {
            Class.forName(driver);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    public static Connection getConnection(){
        Connection connection = null;
        try {
            connection =
DriverManager.getConnection(url,username,password);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return connection;
    }

    public static void release(Connection connection, Statement statement,
ResultSet resultSet){
        try {
            if(connection != null){
                connection.close();
            }
        }
    }
}

```

```
        if(statement != null){
            statement.close();
        }
        if(resultSet != null){
            resultSet.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

}
```