

1. JDBC: java database connect *****

1. 加载驱动 (利用drivermanager类可省略)
2. 创建链接: ip, port, username, password
3. 创建执行器 (Statement)
4. 执行sql, 处理结果
5. 关闭资源

```
package com.etc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;

public class TestMain1 {

    public static void main(String[] args) throws ClassNotFoundException,
    SQLException {

        // 1. 加载驱动
        Class.forName("com.mysql.jdbc.Driver");
        Class.forName("oracle.jdbc.OracleDriver");
        // 读取 META-INF/service/java.sql.Driver文件

        // 2. 创建链接
        Connection con =
        DriverManager.getConnection("jdbc:mysql://47.93.34.175:3306/my_mazheng", "root",
        "123456");
        Connection con1 =
        DriverManager.getConnection("jdbc:oracle:thin:@39.105.100.118:1521:helowin",
        "scott", "123456");

        // 3. 创建执行器
        Statement stat = con.createStatement();
        Statement stat1 = con1.createStatement();

        // 4. 执行SQL, 处理结果
        String sql = "select id, ename from stu";
        String sql1 = "select empno, ename, job, mgr, hiredate, sal from emp";
        // 执行sql查询, 获取结果ResultSet
        ResultSet rs = stat.executeQuery(sql);

        while (rs.next()) { // 是否有下一条记录
            int id = rs.getInt(1); // 根据第几列
            String ename = rs.getString("ename"); // 根据列名字
            System.out.println(id + "@" + ename);
        }

        ResultSet rs1 = stat1.executeQuery(sql1);
        while (rs1.next()) { // 是否有下一条记录
```

```

        int empno = rs1.getInt(1); // 根据第几列
        String ename = rs1.getString("ename"); // 根据列名字
        String job = rs1.getString(3);
        int mgr = rs1.getInt(4);
        java.sql.Date d1 = rs1.getDate(5);
        java.util.Date d2 = new java.util.Date(d1.getTime());
        String d3 = new SimpleDateFormat("yyyy-MM-dd").format(d2);
        System.out.println(empno + "@" + ename + "@" + job + "@"
            + mgr + "@" + d3);
    }

    // 5. 关闭资源
    con.close();
    con1.close();

}
}

```

- Class.forName("类全名"); 把驱动类加载到jvm中
- DriverManager: 驱动管理能够根据 url, 用户名, 密码 创建链接对象
- Statement执行器; 根据链接获取执行器。
- 执行SQL: 执行器的 executeQuery (R), executeUpdate (CUD)
- ResultSet:
 - boolean next(): 判断是否还有下一行, 如果有指向这一行
 - XXX getXXX(); XXX合适数据类型 ,
 - getXX(第几列)
 - getXXX("列名")
- 执行器关闭, 关闭链接。释放资源

1.1 Maven : 流行的jar的管理库

- groupid: 公司名, alibaba, tencent, accentue, ibm.....
- artifactid: 项目名
- version: 版本

- packaging:
 - jar 库文件, 可执行jar
 - pom 父级项目
 - war: web项目

```
47     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
48     xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">
49   <!-- localRepository
50   | The path to the local repository maven will use to store artifacts.
51   |
52   | Default: ${user.home}/.m2/repository
53   <localRepository>/path/to/local/repo</localRepository>
54   >
55   <localRepository>E:\Store</localRepository>
56   <!-- interactiveMode
57   | This will determine whether maven prompts you when it needs input. If set to false,
58   | maven will use a sensible default value, perhaps based on some other setting, for
59   | the parameter in question.
60   |
61   | Default: true
62   <interactiveMode>true</interactiveMode>
63   -->
64
65   <!-- offline
66   | Determines whether maven should attempt to connect to the network when executing a build.
```