



链滴

# MYSQL简单命令

作者: [zadenii](#)

原文链接: <https://ld246.com/article/1438594860820>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

常用mysql操作指令：

1. 连接：`mysql -u用户名 -p密码`
2. 退出：`mysql>exit;`
3. 建数据库：`mysql>create database ???;`
4. 显示数据库：`mysql>show databases;//(-s)`
5. 删除数据库：`mysql>drop database ???;`
6. 连接数据库：`mysql>use ???;`
7. 创建数据表：`mysql>create table ???(字段名 类型, 字段名 类型);`
8. 删除表：`mysql>drop table ???;`
9. 表插入数据：`mysql>insert into 表名 value(?,?,...);`与建表元素个数相同，类型相符
10. 表查数据：`mysql>select ??? from 表名 where ??;` `mysql>select * from 表名;`
11. 表查前几行数据：`mysql>select * from 表名 order by id limit 0,2;`
12. 删除表中数据：`mysql>delete from 表名 where ???;`
13. 修改表中数据：`mysql>update 表名 set name="???" where id=1?`

如若以上内容不够用，可以参考[这里](#)。

在实际的JAVA WEB中，就应当把数据库的驱动，连接，用户名和密码写在配置文件中。这样才便于护。

以Properties读取配置文件：

```
java.properties:
driver=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/test
user=root
password=5938
```

引入：

```
import java.io.InputStream;
import java.util.Properties;
```

读取操作：

```
static{
Properties prop = new Properties();
InputStream in = 类名.class.getClassLoader().getResourceAsStream("jdbc.properties");
try{
prop.load(in);
} catch(IOException e){
e.printStackTrace();
}
}
```

```
driver=prop.getProperty("driver");
```

```
url=prop.getProperty("url");
user=prop.getProperty("user");
password=prop.getProperty("password");
}
```

完整的链接程序如下：

```
package ppj.demo01.jdbc;

import java.io.InputStream;
import java.util.Properties;

import java.io.IOException;

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

/**
 * test to connection sql
 * 单线程链接
 * @author pengpj
 *
 */
public class JDBCConnectionTest01 {
    /**
     * 数据库连接设置
     * driver 数据库驱动
     * url 数据库链接字符
     * user 用户名
     * password 密码
     */
    public static String driver=null;//"com.mysql.jdbc.Driver";
    public static String url = null;//"jdbc:mysql://localhost:3306/test";
    public static String user = null;//"root";
    public static String password = null;//"5938";
    /**
     * 在加载类时，将jdbc链接数据库的信息获取
     * 获取途径：jdbc.properties
     */
    static {
        Properties prop = new Properties();
        InputStream in = JDBCConnectionTest01.class.getClassLoader().getResourceAsStream("jdbc.properties");
        try{
            prop.load(in);
        }catch( IOException e){
            e.printStackTrace();
        }
        driver = prop.getProperty("driver");
        url = prop.getProperty("url");
    }
}
```

```

    user = prop.getProperty("user");
    password = prop.getProperty("password");
}
/*
 * 加载驱动
 */
{
    try{
        Class.forName(driver);
    }catch(ClassNotFoundException e){
        System.out.println("连接失败! \nnot found the exception");
        e.printStackTrace();
    }
}
/*
 * 获取链接
 */
public static Connection getConne(){
    Connection conn = null;
    try{
        conn = DriverManager.getConnection(url, user, password);
    }catch(SQLException e){
        e.printStackTrace();
    }
    return conn;
}
/*
 * 关闭结果集
 */
public static void close(ResultSet rs){
    if(null!=rs){
        try{
            rs.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
    }
}
/*
 * 关闭封装SQL命令的对象
 */
public static void close(Statement stmt){
    if(null!=stmt){
        try{
            stmt.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
    }
}
/*
 * 关闭链接
 */
public static void close(Connection conn){

```

```

        if(null!=conn){
            try{
                conn.close();
            }catch(SQLException e){
                e.printStackTrace();
            }
        }
    }
}

```

测试类如下:

```

package ppj.demo01;

import ppj.demo01.jdbc.JDBCConnectionTest01;

import static org.junit.Assert.*;
import org.junit.Test;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;

public class testJDBCConnectionTest01 {
    @Test
    public void testGetConne() throws Exception{
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String sqlstr = "select * from user";
        try{
            conn = JDBCConnectionTest01.getConne();
            stmt = conn.createStatement();
            rs = stmt.executeQuery(sqlstr);
            while(rs.next()){
                int id = rs.getInt("id");
                String userName = rs.getString("userName");
                String addr = rs.getString("addr");
                System.out.println("id:"+id+" userName:"+userName+" address:"+addr);
            }
        }catch(Exception e){

        }finally{
            JDBCConnectionTest01.close(conn);
            JDBCConnectionTest01.close(rs);
            JDBCConnectionTest01.close(stmt);
        }
        assertEquals("false",JDBCConnectionTest01.getConne(),JDBCConnectionTest01.getConne());
    }
}

```

注:

在数据库中已经建立有库test与表user，所以可以在程序中直接链接调用而不用再建表。