

JDBC 技术 connection :

- 架构介绍、Driver、DriverManager、Connectin
- Statement Sql 注入
- preparedStatement 基本用法 参数处理技巧
- ResultSet 结果集处理
- 时间类型 CLOB BLOB
- CallableStatement
- 批处理
- 事务处理
- 经典 JDBC 代码总结
- 使用 properties 文件存储 JDBC 链接

MySql 数据库命令行操作 :

- 配置环境变量
 - 将bin目录配置到path中
- 命令行操作

登陆操作	mysql -hlocalhost -uroot -p123456
退出操作	exit
数据库操作	建库: create testjdbc; 卸载库: drop testjdbc; 显示所有数据库: show databases; 选择库: use testjdbc;
表操作	建表的sql语句; 显示库中所有表: show tables; 显示某个表的内容: describe testjdbc;

注意 !: 一个小错误 , 应该是 `create database testjdbc;`
`drop database testjdbc`

JDBC 访问数据库流程

- 1、加载 JDBC 驱动程序。
- 2、建立与数据库的链接。
 - * 建立连接是比较耗时间的。
 - * 后期开发中为了提高效率，都是使用连接池来管理连接对象。
- 3、发送 SQL 语句。
- 4、得到查询结果。

JDBC 常用接口

1、Driver 接口

• Driver接口

- Driver接口由数据库厂家提供，对于java开发者而言，只需要使用Driver接口就可以了。
- 在编程中要连接数据库，必须先装载特定厂商的数据库驱动程序。不同的数据库有不同的装载方法。
- 驱动：就是各个数据库厂商实现的Sun公司提出的JDBC接口。即对Connection等接口的实现类的jar文件
- 装载MySQL驱动
 - `Class.forName("com.mysql.jdbc.Driver");`
- 装载Oracle驱动
 - `Class.forName("oracle.jdbc.driver.OracleDriver");`

2、DriverManager 接口

• DriverManager接口

- DriverManager是JDBC的管理层，作用于用户和驱动程序之间。
- DriverManager跟踪可用的驱动程序，并在数据库和相应的驱动程序之间建立连接。

3、Connection 接口

- **Connection接口**

- Connection与特定数据库的连接（会话），在连接上下文中执行 SQL 语句并返回结果。
- DriverManager的getConnection()方法建立在JDBC URL中定义的数据库Connection连接上
- 连接MySQL数据库：
 - `Connection con = DriverManager.getConnection("jdbc:mysql://host:port/database", "user", "password");`
- 连接ORACLE数据库：
 - `Connection con = DriverManager.getConnection("jdbc:oracle:thin:@host:port:database", "user", "password");`

4、Statement 接口

- **Statement接口**

- 用于执行静态 SQL 语句并返回它所生成结果的对象。
- 三种Statement类：
 - Statement：
 - 由createStatement创建，用于发送简单的SQL语句。（不带参数的）
 - PreparedStatement：
 - 继承自Statement接口，由prepareStatement创建，用于发送含有一个或多个输入参数的sql语句。PreparedStatement对象比Statement对象的效率更高，并且可以防止SQL注入。我们一般都用PreparedStatement。
 - CallableStatement：
 - 继承自PreparedStatement。由方法prePareCall创建，用于调用存储过程。
- 常用的Statement方法：
 - execute()：运行语句，返回是否有结果集。
 - executeQuery()：运行select语句，返回ResultSet结果集。
 - executeUpdate()：运行insert/update/delete操作，返回更新的行数。

5、ResultSet 接口

- **ResultSet接口**

- Statement执行SQL语句时返回ResultSet结果集。
- ResultSet提供的检索不同类型字段的方法，常用的有：
 - getString():获得在数据库里是varchar、char等数据类型的对象。
 - getFloat():获得杂数据库里是Float类型的对象。
 - getDate():获得在数据库里面是Date类型的数据。
 - getBoolean():获得在数据库里面是Boolean类型的数据

- 依序关闭使用之对象及连接:

- ResultSet → Statement → Connection

事务

事务基本概念

- **一组要么同时执行成功，要么同时执行失败的SQL语句。是数据库操作的一个执行单元！**
- **事务开始于：**
 - 连接到数据库上，并执行一条DML语句(INSERT、UPDATE或DELETE)。
 - 前一个事务结束后，又输入了另外一条DML语句。
- **事务结束于：**
 - 执行COMMIT或ROLLBACK语句。
 - 执行一条DDL语句，例如CREATE TABLE语句；在这种情况下，会自动执行COMMIT语句。
 - 执行一条DCL语句，例如GRANT语句；在这种情况下，会自动执行COMMIT语句。
 - 断开与数据库的连接。
 - 执行了一条DML语句，该语句却失败了；在这种情况下，会为此无效的DML语句执行ROLLBACK语句。

事务的四大特点 (ACID)

- atomicity (原子性)
 - 表示一个事务内的所有操作是一个整体，要么全部成功，要么全失败；
- consistency (一致性)
 - 表示一个事务内有一个操作失败时，所有的更改过的数据都必须回滚到修改前的状态；
- isolation (隔离性)
 - 事务查看数据时数据所处的状态，要么是另一开发事务修改它之前的状态，要么是另一事务修改它之后的状态，事务不会查看中间状态的数据。
- durability (持久性)
 - 持久性事务完成之后，它对于系统的影响是永久性的。

CLOB (文本大对象)

- CLOB (Character Large Object)
 - 用于存储大量的文本数据
 - 大字段有些特殊，不同数据库处理的方式不一样，大字段的操作常常是以流的方式来处理的。而非一般的字段，一次即可读出数据。
- Mysql中相关类型：
 - TINYTEXT最大长度为255(2^8-1)字符的TEXT列。
 - TEXT[(M)]最大长度为65,535($2^{16}-1$)字符的TEXT列。
 - MEDIUMTEXT最大长度为16,777,215($2^{24}-1$)字符的TEXT列。
 - LONGTEXT最大长度为4,294,967,295或4GB($2^{32}-1$)字符的TEXT列。

BLOB (二进制大对象)

- BLOB (Binary Large Object)
 - 用于存储大量的二进制数据
 - 大字段有些特殊，不同数据库处理的方式不一样，大字段的操作常常是以流的方式来处理的。而非一般的字段，一次即可读出数据。
- Mysql中相关类型：
 - TINYBLOB最大长度为255(2^8-1)字节的BLOB列。
 - BLOB[(M)]最大长度为65,535($2^{16}-1$)字节的BLOB列。
 - MEDIUMBLOB最大长度为16,777,215($2^{24}-1$)字节的BLOB列。
 - LONGBLOB最大长度为4,294,967,295或4GB($2^{32}-1$)字节的BLOB列。