

# JDBC

@M了个J  
李明杰

<https://github.com/CoderMJLee>

<https://space.bilibili.com/325538782>

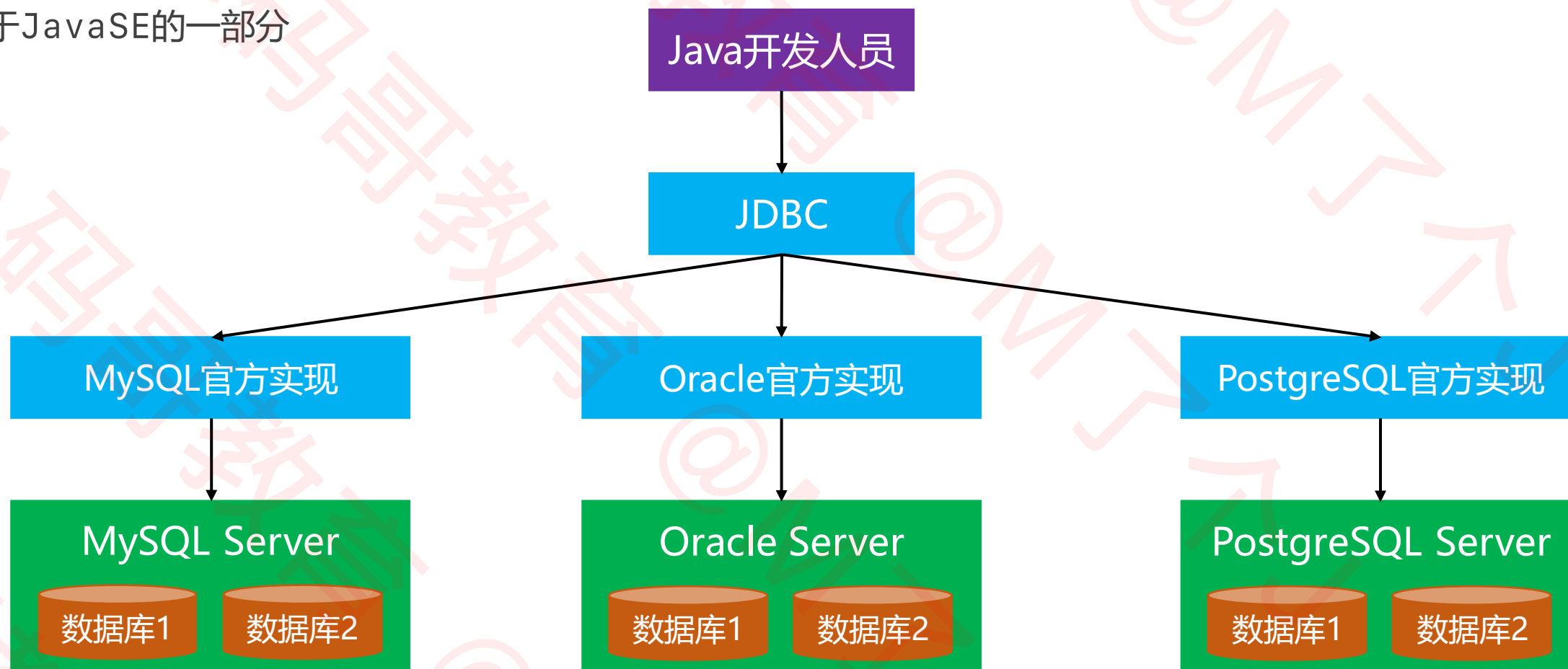


实力IT教育 [www.520it.com](http://www.520it.com)



# 如何通过Java操作数据库?

- JDBC, 全称是Java Database Connectivity
- 在Java中用来规范如何访问关系型数据库, 由各大数据库厂商去实现它
- 属于JavaSE的一部分



# 下载MySQL的JDBC实现 (jar, 驱动包)

■ 8.0版本的驱动包同时支持MySQL 5.7、8.0

□ <https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-versions.html>

■ 下载地址

□ <https://dev.mysql.com/downloads/connector/j/>

## Connector/J 8.0.20

Select Operating System:

Platform Independent

Looking for previous GA versions?

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.20.tar.gz)	8.0.20	3.7M	<a href="#">Download</a>
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.20.zip)	8.0.20	4.5M	<a href="#">Download</a>

# JDBC使用步骤

- ① 将Driver (驱动程序) 注册到DriverManager (驱动程序管理者)
- ② 利用DriverManager创建Connection (数据库连接)
- ③ 利用Connection创建Statement (语句)
- ④ 利用Statement执行SQL语句
- ⑤ 关闭资源 (关闭Statement、Connection等)

■ MySQL的url格式是

□ `jdbc:mysql://IP地址:端口号/数据库名`

□ 比如`jdbc:mysql://localhost:3306/xmg`

	MySQL 驱动包 6.x以前	MySQL 驱动包 6.x开始
驱动类	<code>com.mysql.jdbc.Driver</code>	<code>com.mysql.cj.jdbc.Driver</code>
时区		需要指定一个确定的时区 比如在url后加上参数 <code>serverTimezone=UTC</code>

# JDBC版本

- 从JDBC 4.0开始，显式注册驱动程序是可选的
- 我们只需要将供应商的Jar放在类路径中，然后DriverManager就可以自动检测并加载驱动程序
- JDBC版本
  - ▣ JDBC 4.0包含在Java SE 6中
  - ▣ JDBC 4.1包含在Java SE 7中
  - ▣ JDBC 4.2包含在Java SE 8中
  - ▣ JDBC 4.3包含在Java SE 9中

# Statement的常用API

■ `ResultSet` `executeQuery(String sql)`

▣ 执行DQL语句

■ `int` `executeUpdate(String sql)`

▣ 执行DML、DDL语句

▣ 如果是DML语句，返回值代表影响的记录数量；如果数据库没有任何返回值，通常返回0

# ResultSet的常用API

■ `boolean next()`

□ 让游标指向下一行。如果指向的这行有数据，就返回true，否则，返回false

■ `XX getX(int columnIndex)`、`XX getX(String columnLabel)`

□ 获取当前行（游标指向的那行）某一系列的数据

□ `columnIndex`的数值从1开始



# PreparedStatement

- PreparedStatement接口继承自Statement接口

- 建议使用PreparedStatement替代Statement

- PreparedStatement的优点

- 可以防止SQL注入

- 执行速度比Statement快

- 支持批量处理

# 配置文件

- 一些经常动态修改的值，建议放入到配置文件中，不要写死在Java代码中

- Java中常见的配置文件

- properties：比较单一，适合量小、简单的配置

- xml：比较灵活，适合量大、复杂的配置

- properties文件

- key、value的分隔符是=、：

- 建议分隔符左右不要留空格

- #、!开头是单行注释

- 可以用反斜线\连接多行内容

# 数据库连接池

■ 数据库连接池，可以提高访问数据库的性能，负责创建、分配、管理和释放数据库连接

## ■ 基本思想

□ 在初始化时，创建一定数量的数据库连接对象存储在内存中

□ 当需要访问数据库时，并非建立一个新的连接，而是从连接池中取出一个已创建的空闲连接对象

□ 使用完毕后，用户也并非将连接关闭，而是将连接放回连接池中，以供下一个请求访问使用

□ 当连接的空闲时间已超过最大空闲时间时，将会释放掉该连接

□ 可以通过设置参数来控制初始连接数、连接的上下限数以及每个连接的最大使用次数、最大空闲时间

□ 可以通过其自身的管理机制来监视数据库连接的数量、使用情况

## ■ Java中常见的开源的数据库连接池

□ C3P0、Proxool、DBCP、BoneCP、Druid等

# Druid

- 产自阿里巴巴，项目地址：<https://github.com/alibaba/druid/>
- jar包下载：<https://mvnrepository.com/artifact/com.alibaba/druid>

# Spring JDBC

■ [Spring JDBC](#)框架可以帮助开发者节省大量开发工作，自动去处理一些低级细节，比如

□ 异常处理、打开和关闭资源 (`Connection`、`Statement`、`ResultSet`)

■ 需要的jar包

□ `spring-jdbc`

□ `spring-beans`

□ `spring-core`

□ `spring-tx`

□ `spring-jcl`

# Spring JDBC核心类：JdbcTemplate

## ■ 构造方法

□ `public JdbcTemplate(DataSource dataSource)`

## ■ 执行DDL、DML语句

□ `int update(String sql, Object... args)`

## ■ 执行DQL语句

□ `<T> List<T> query(String sql, RowMapper<T> rowMapper, Object... args)`

□ `<T> T queryForObject(String sql, RowMapper<T> rowMapper, Object... args)`