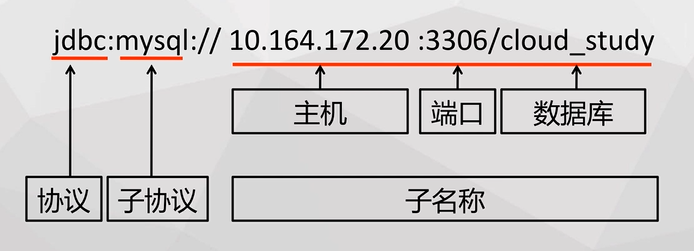
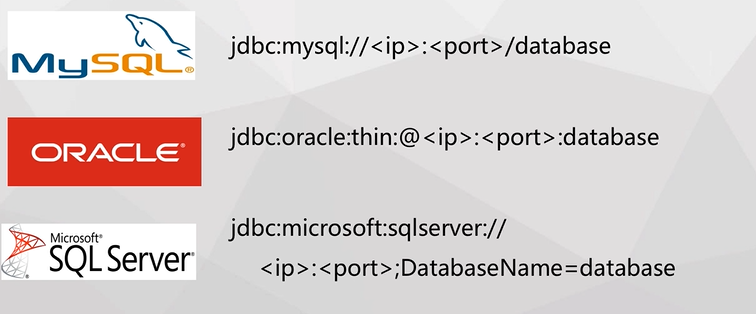
JDBC笔记

# JDBC基础

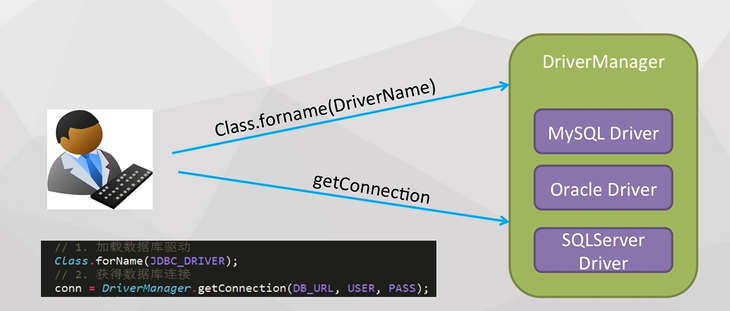
## JDBC URL



1. 常用jdbc的url

## JDBC API

### Driver & DriverManager



### Connection



1. 常用方法
   1. Statement stmt = conn.createStatement();

### Statement

1. SQL的容器
2. 常用方法
   1. ResultSet rs = stmt..executeQuery(“SELECT username FROM user”);

### ResultSet

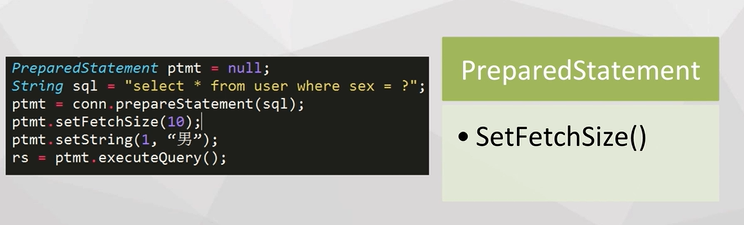
1. 方法
   1. .next()
   2. .previous()
   3. .absolute()
   4. .beforeFirst()
   5. .afterLast()
2. 其他方法
   1. .getString(ColumnName/Index)
   2. .getInt(ColumnName/Index)
   3. .getObject(ColumnName/Index)

# 解决数据库数据传输到浏览器乱码问题

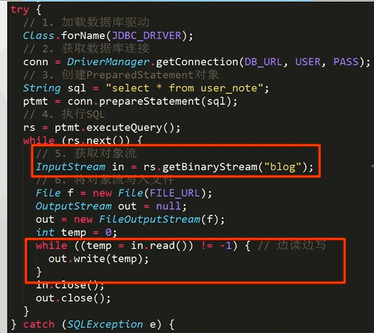
1. 设置response对象的数据类型
   1. resp.setContentType(“text/html; charset=utf-8”);
   2. 即设置HTTP报文头文件解码方式

# JDBC进阶

## 游标

* 1. 提供了一种客户端读取部分服务器数据端结果集的机制
  2. 如何使用游标
     1. 设置
     2. PreparedStatement对象代替Statement
        1. SetFetchSize()
        2. 支持格式化参数?

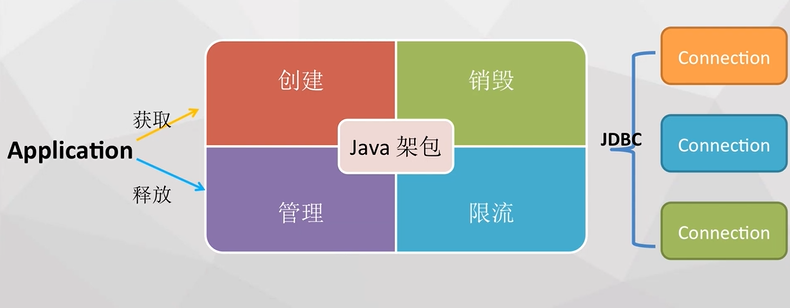
## 流方式

* 1. 二进制流方式大对象读取
  2. ResultSet.getBinaryStream()
  3. 

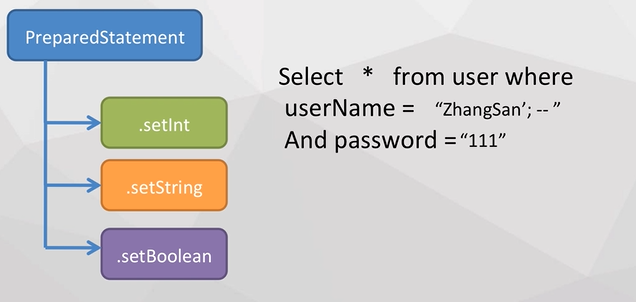
## 批处理

* 1. 插入大量数据
  2. Statement
     1. addBatch()
     2. executeBatch()
     3. clearBatch()

# 数据库连接池DBCP

1. 连接池
2. DBCP——使用最多
   1. commons-dbcp.jar
   2. commons-pool.jar
   3. commons-logging.jar
3. 创建线程池对象BasicDataSource
   1. .setDriverName()
   2. .setUrl()
   3. .setUsername()
   4. .setPassword()
4. 获取数据库连接
   1. GetConnection()
5. 释放连接
   1. 重写了Connection.close()方法
   2. 归还连接而不是关闭连接
6. 高级配置
   1. .setInitialSize()
      1. 初始化连接数量
   2. .setMaxTotal()
      1. 最大连接数量
   3. .setMaxWaitMillis()
      1. 设置线程最大等待时间，当没有空闲连接时
   4. .setMaxIdle()
      1. 最大空闲连接数，自动销毁空闲连接
   5. .setMinIdle()
      1. 设置最小连接数，防止空闲连接不够
      2. 最好和上者设置同样的值，避免频繁创建和销毁连接
   6. DBCP定期检查
      1. .setTestWhileIdle(True)
         1. 数据库服务器会自动关闭空闲时间超过一定时间的连接
         2. 客户端连接池并不知道，会将失效的连接租借给应用程序
         3. 开启自动检查空闲连接是否有效
      2. .setMinEvictableIdleTimeMillies()
         1. 最小空闲时间，超时会销毁该空闲连接
         2. 要小于数据库自动关闭空闲连接的时间
      3. .setTimeBetweenEvictionRunsMillies()
         1. 检测运行时间的间隔

# SQL注入与防范

1. SQL注入
2. 解决方案
   1. .preparedStatement(sql)格式化参数
      1. .setInt()
      2. .setString()
      3. .setBoolean()
3. 其他注意事项
   1. 严格的数据库权限管理
      1. 仅给予Web应用访问数据库的最小权限
      2. 避免Drop table等权限
   2. 封装数据库错误
      1. 禁止直接将后端数据库异常信息暴露给用户！
      2. 对后端异常信息进行必要的封装，避免用户直接查看到后端异常！
   3. 机密文件禁止明文存储
      1. 涉密信息需要加密处理
      2. 使用AES\_ENCRYPT/AES\_DECRYPT加密和解密

# JDBC事务

1. Connection
   1. .setAutoCommit(false) 开启事务
   2. .commit() 提交事务
   3. .rollback() 回滚事务
2. 检查点
   1. .setSavePoint()
   2. .rollback(SavePoint savePoint)
3. 事务隔离级别
   1. 读未提交（read uncommitted）
   2. 读提交（read committed）
   3. 重复读（repeatable read）
   4. 串行读（serializable）
   5. MySQL默认事务隔离级别为repeatable read
   6. 事务隔离级别越高，数据库性能越差