

河南中医药大学信息技术学院（智能医疗行业学院）产教协同课程建设成果
智能医学工程专业《医疗信息系统开发》课程

第03章：数据库操作

黄子杰

河南方和信息科技有限公司

河南中医药大学信息技术学院（智能医疗行业学院）智能医疗教研室

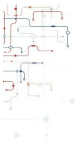
<https://aitcm.hactcm.edu.cn>

2025/9/18

本章概要

- 引言与核心概念
 - 为什么需要数据库
 - 数据库两大阵营
 - Java数据库连接 - JDBC
- 关系型数据库操作 (JDBC核心)
 - 环境准备与基础流程
 - 核心接口深度解析
 - 事务管理 (Transaction)
 - 数据库连接池 (C3P0, HikariCP, DBCP)
- ORM框架 - MyBatis & JPA (Hibernate)
 - 为什么需要ORM
 - MyBatis 简介
- 非关系型数据库操作
 - Redis
 - MongoDB

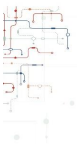




引言与核心概念

为什么需要数据库

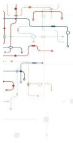
- 数据持久化的必要性（对比于内存、文件存储）。
- 数据库的优势：并发控制、数据一致性、安全、高效查询等。



引言与核心概念

数据库两大阵营

- 关系型数据库 (RDBMS)
 - 核心概念：表、行、列、主键、外键、关系（一对一，一对多，多对多）。
 - 语言：SQL (Structured Query Language)。
 - 代表产品：Oracle, MySQL, PostgreSQL, SQL Server。
 - 特点：强一致性、ACID事务、结构化数据。
- 非关系型数据库 (NoSQL)
 - 核心思想：打破关系模型，适应多样化数据结构和海量数据。
 - 分类及代表：
 - 键值型 (Key-Value): Redis** (极高性能，内存存储)
 - 文档型 (Document): MongoDB** (类JSON文档，灵活模式)



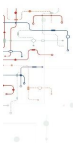
引言与核心概念

Java数据库连接

- JDBC是什么？（Java Database Connectivity）

是 Java 语言中用来规范客户端程序如何访问数据库的应用程序编程接口（API）。提供了一组标准的 Java 类和接口，允许 Java 程序（如 Java 应用程序、JSP、Servlet 等）与各种不同的关系型数据库进行连接和操作。
- JDBC架构：应用程序 -> JDBC API -> JDBC Driver -> 数据库。
- 驱动类型

类型	名称	平台无关性	是否需要本地库	性能	部署难度	推荐度
Type 1	JDBC-ODBC 桥	否	是 (ODBC)	差	困难	已淘汰
Type 2	本地API驱动	否	是 (DB Client)	较好	困难	不推荐
Type 3	网络协议驱动	是	否	中等	复杂（需中间件）	少见
Type 4	纯Java驱动	是	否	好	简单（一个JAR）	推荐



关系型数据库操作

环境准备与基础流程

- 环境准备与基础流程

引入数据库驱动JAR包（mysql-connector-java, ojdbcX for Oracle）。

标准步骤：

 - 1注册驱动（Class.forName(), 现代版本可省略）。
 - 2获取连接 (DriverManager.getConnection(url, user, password))。
 - 3创建Statement对象 (createStatement())。
 - 4执行SQL语句 (executeQuery, executeUpdate)。
 - 5处理结果集 (ResultSet 对象, next(), getXxx())。
 - 6释放资源 (close(), 强调在finally或使用try-with-resources)。
- 核心接口

Connection: 与数据库的会话。

Statement: 用于执行静态SQL语句，存在SQL注入风险。

PreparedStatement

预编译SQL，性能更高。

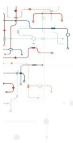
有效防止SQL注入（使用?作为占位符，setXxx()方法传参）。



关系型数据库操作

事务管理 (Transaction)

- JDBC中如何控制事务：
 - setAutoCommit(false): 关闭自动提交。
 - commit(): 提交事务。
 - rollback(): 回滚事务。
 - 应用场景：模拟银行转账，确保原子性。



关系型数据库操作

数据库连接池 (C3P0, HikariCP, DBCP)

- 为什么需要连接池？（减少创建连接的开销，管理连接资源）。
- 常见连接池
- 如何使用HikariCP、C3P0



03-1 HikariCP连接池添加依赖.xml



03-1 C3P0连接池添加依赖.xml



03-2 HikariCP基础用法.txt



03-2 C3P0基础用法.txt



hikari.properties.txt



C3P0通过XML配置文件配置.xml



使用属性文件的Java代码.txt



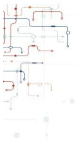
C3P0使用XML配置文件的Java代码.txt



集成到项目架构中.txt



C3P0集成到项目架构中.txt



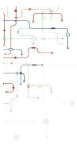
ORM框架

为什么需要ORM

- 优势

解决JDBC手动映射ResultSet到Java对象的繁琐和冗余。

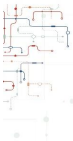
Object-Relational Mapping: 将数据库表记录自动映射为Java对象。



ORM框架

MyBatis 简介

- 半自动化ORM，SQL由开发者控制，灵活性高。
- 核心组件：SqlSessionFactory, SqlSession, Mapper接口。
- XML映射文件 (<select>, <insert>, <update>, <delete>) 或注解开发。
- 动态SQL (<if>, <where>, <foreach>)。



非关系型数据库操作

Redis

- 内存中的数据结构存储，用作缓存、消息队列等。
- 操作五种基本数据类型：String, List, Set, Hash, ZSet。
- 示例：用Redis实现一个简单的缓存功能。



添加依赖pom.xml



创建 Redis 缓存工具类.txt



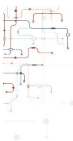
创建用户服务类（模拟数据库操作）.txt



创建使用缓存的用户服务包装类.txt



调用方法.txt



非关系型数据库操作

MongoDB

- 面向文档的数据库
- 基础操作：
 - 连接MongoDB。
 - 获取集合 (getCollection)。
 - 文档的CRUD：insertOne, find, updateOne, deleteOne。
 - 使用Document对象构建查询和更新条件。



MD添加依赖.xml



MD创建实体类.txt



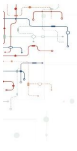
MD创建工具类.txt



MD用户服务类.txt



MD主类.txt



本章总结

- JDBC核心编程步骤

- 注册驱动 (`Class.forName()`, 可省略)
- 获取连接 (`DriverManager.getConnection()`)
- 创建Statement对象 (`createStatement()`)
- 执行SQL (`executeQuery` / `executeUpdate`)
- 处理结果集 (`ResultSet`, `next()`, `getXxx()`)
- 释放资源 (`close()`), 必须使用try-with-resources或finally块确保资源被释放。

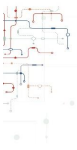
- 事务管理 (Transaction)

- `connection.setAutoCommit(false)`: 关闭自动提交。
- `connection.commit()`: 提交事务。
- `connection.rollback()`: 回滚事务。

- 数据库连接池

- 关系型数据库操作

- 非关系型数据库操作



本章作业

- 操作关系型数据库实现一个单表的增、删、改、查功能。



河南中医药大学信息技术学院（智能医疗行业学院）智能医疗教研室
河南中医药大学医疗健康信息工程技术研究所