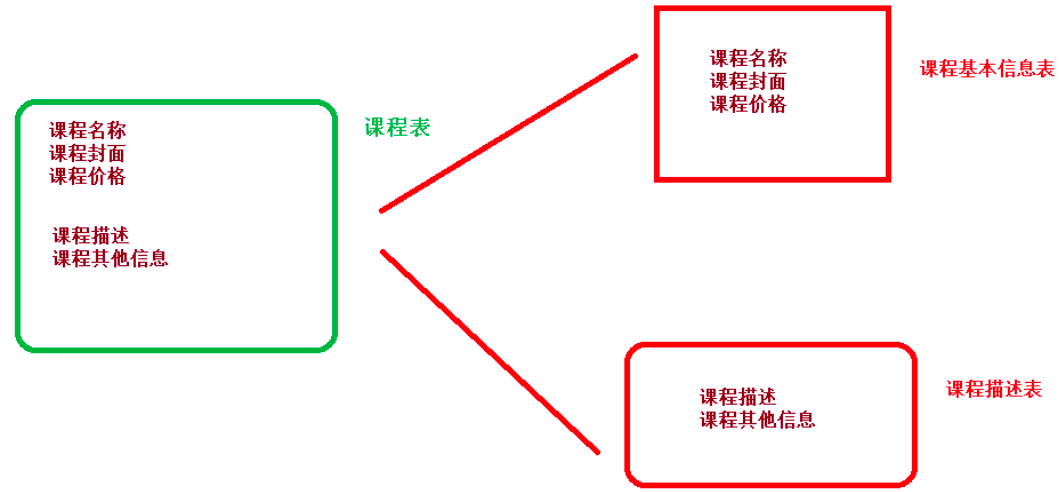
Sharding Sphere笔记

# 基本概念

一套开源的分布式数据库中间件解决方案。有三个产品：Sharding-JDBC 和 Sharding-Proxy。定位为关系型数据库中间件，合理在分布式环境下使用关系型数据库操作。分库分表有两种方式：垂直切分和水平切分。垂直切分：垂直分表和垂直分库。水平切分：水平分表和水平分库

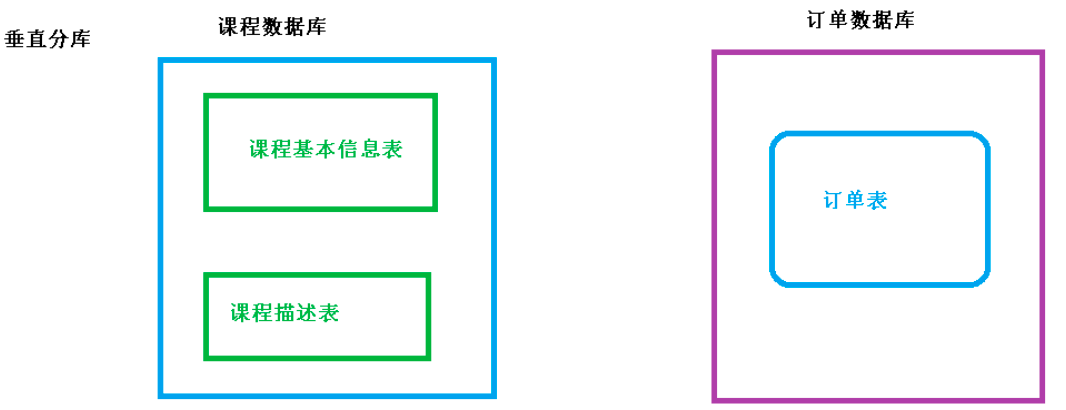
### 垂直分表

操作数据库中某张表，把这张表中一部分字段数据存到一张新表里面，再把这张表另一部分字段数据存到另外一张表里面。



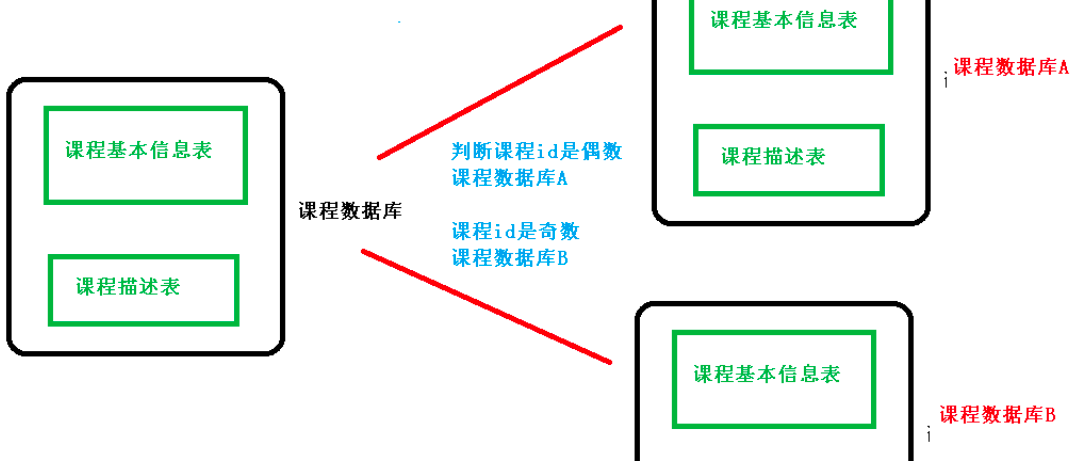
### 垂直分库

把单一数据库按照业务进行划分，专库专表。



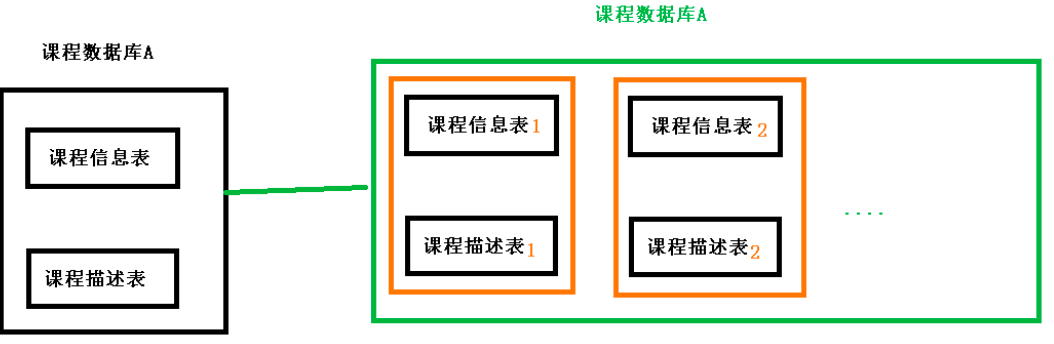
### 水平分库

库中所有表的结构都相同，数据分片放在分库的相同表名称下。



### 水平分表

分表的表结构不变，数据分片放在不同的表中。



### 应用

（1）在数据库设计时候考虑垂直分库和垂直分表。

（2）随着数据库数据量增加，不要马上考虑做水平切分，首先考虑缓存处理，读写分离，使用索引等等方式，如果这些方式不能根本解决问题了，再考虑做水平分库和水平分表。

### 分库分表问题

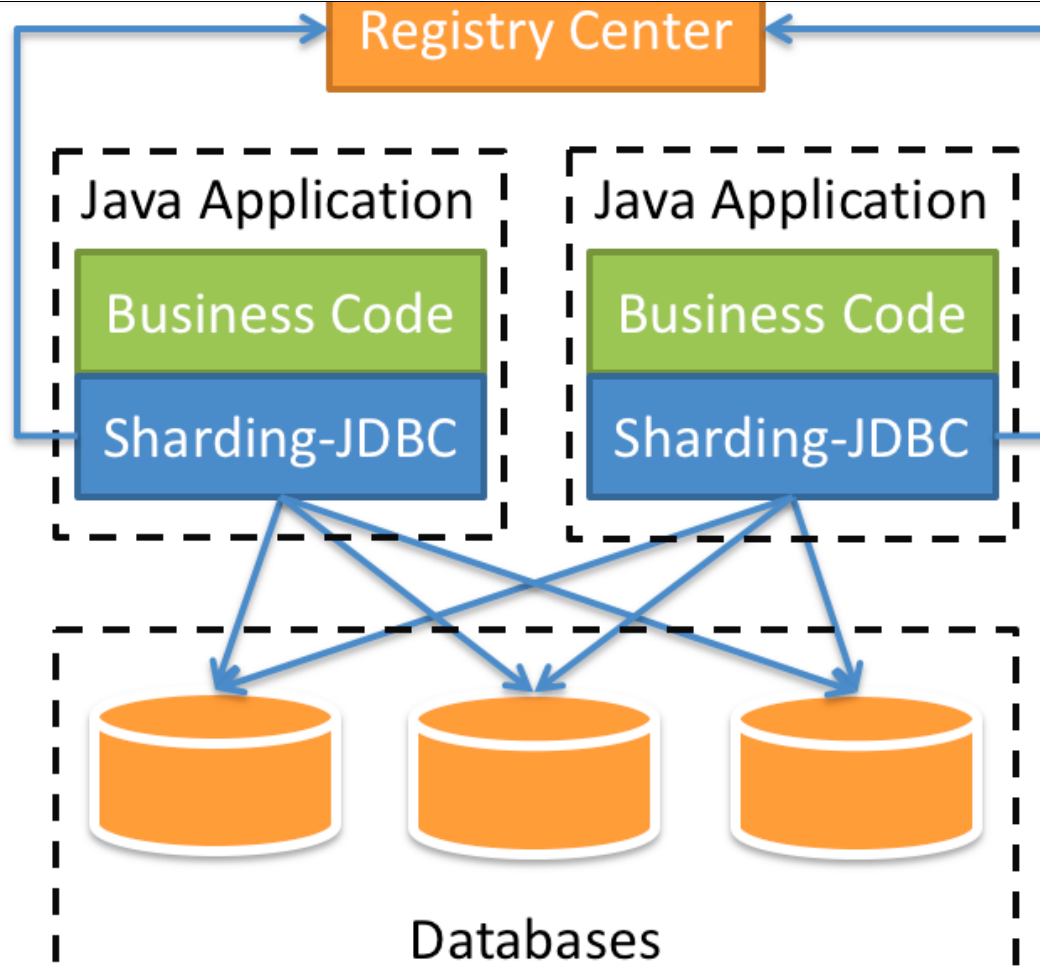
（1）跨节点连接查询问题（分页、排序）

（2）多数据源管理问题

# 第二章Sharding-JDBC

## 一 介绍

是轻量级的 java 框架，是增强版的 JDBC 驱动。主要目的是简化对分库分表之后数据相关操作。



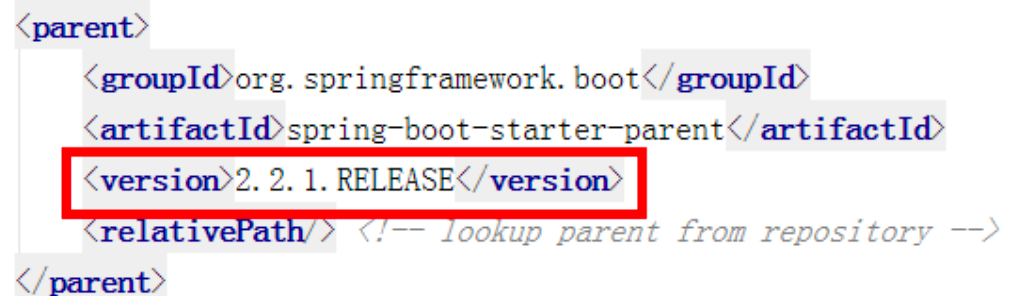
## 二Sharding-JDBC 实现水平分表

### 1、搭建环境

（1）技术：SpringBoot 2.2.1+ MyBatisPlus + Sharding-JDBC + Druid 连接池

（2）创建 SpringBoot 工程

（3）修改工程 SpringBoot 版本 2.2.1



（4）引入需要的依赖



### 2、按照水平分表的方式，创建数据库和数据库表

（1）创建数据库 course\_db。

（2）在数据库创建两张表 course\_1 和 course\_2。

（3）约定规则：如果添加课程 id 是偶数把数据添加 course\_1，如果奇数添加到 course\_2。

### 3、编写代码实现对分库分表后数据的操作

（1）创建实体类，mapper。

### 4、配置 Sharding-JDBC 分片策略

（1）在项目 application.properties 配置文件中进行配置.

*# shardingjdbc 分片策略*

*# 配置数据源，给数据源起名称*

**spring.shardingsphere.datasource.names**=**m1**

***# 一个实体类对应两张表，覆盖***

**spring.main.allow-bean-definition-overriding=true**

*#配置数据源具体内容，包含连接池，驱动，地址，用户名和密码*

**spring.shardingsphere.datasource.m1.type**=**com.alibaba.druid.pool.DruidDataSourc**

**e**

**spring.shardingsphere.datasource.m1.driver-class-name**=**com.mysql.cj.jdbc.Driver**

**spring.shardingsphere.datasource.m1.url**=**jdbc:mysql://localhost:3306/course\_db?**

**serverTimezone=GMT%2B8**

**spring.shardingsphere.datasource.m1.username**=**root**

**spring.shardingsphere.datasource.m1.password**=**root**

*#指定 course 表分布情况，配置表在哪个数据库里面，表名称都是什么 m1.course\_1 , m1.course\_2*

**spring.shardingsphere.sharding.tables.course.actual-data-nodes**=**m1.course\_$-**

**>{1..2}**

*# 指定 course 表里面主键 cid 生成策略 SNOWFLAKE*

**spring.shardingsphere.sharding.tables.course.key-generator.column**=**cid**

**spring.shardingsphere.sharding.tables.course.key-generator.type**=**SNOWFLAKE**

*# 指定分片策略 约定 cid 值偶数添加到 course\_1 表，如果 cid 是奇数添加到 course\_2 表*

**spring.shardingsphere.sharding.tables.course.table-strategy.inline.sharding**

**column**=**cid**

**spring.shardingsphere.sharding.tables.course.table-strategy.inline.algorithm**

**expression**=**course\_$->{cid % 2 + 1}**

*# 打开 sql 输出日志*

**spring.shardingsphere.props.sql.show**=**true**

### 5、编写测试代码