闪聚支付项目 扩展-分库分表

1 订单数据分库分表

1.1 需求分析

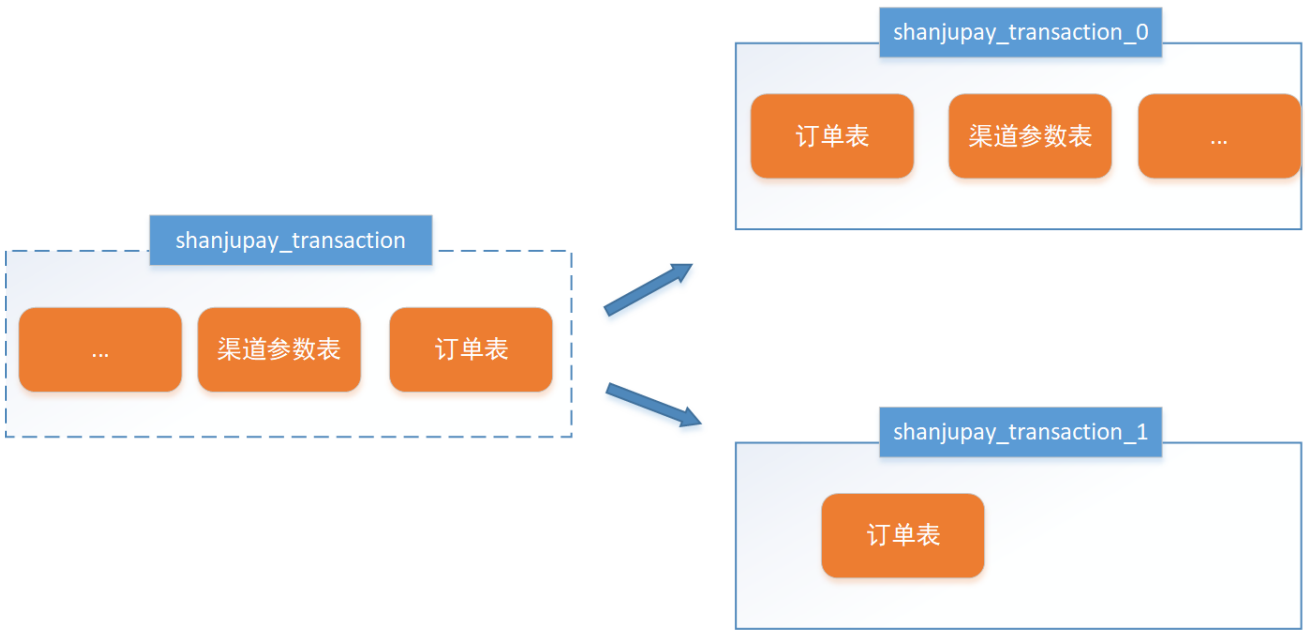
随着公司业务快速发展，接入的商户也越来越多，数据库中订单表的数据量猛增，由于所有表都在一个数据库中，导致服务器存储快满了，访问性能也变慢了，优化迫在眉睫。针对现状目前采用的解决方案是对订单表进行分库分表。

1.2 分库分表技术

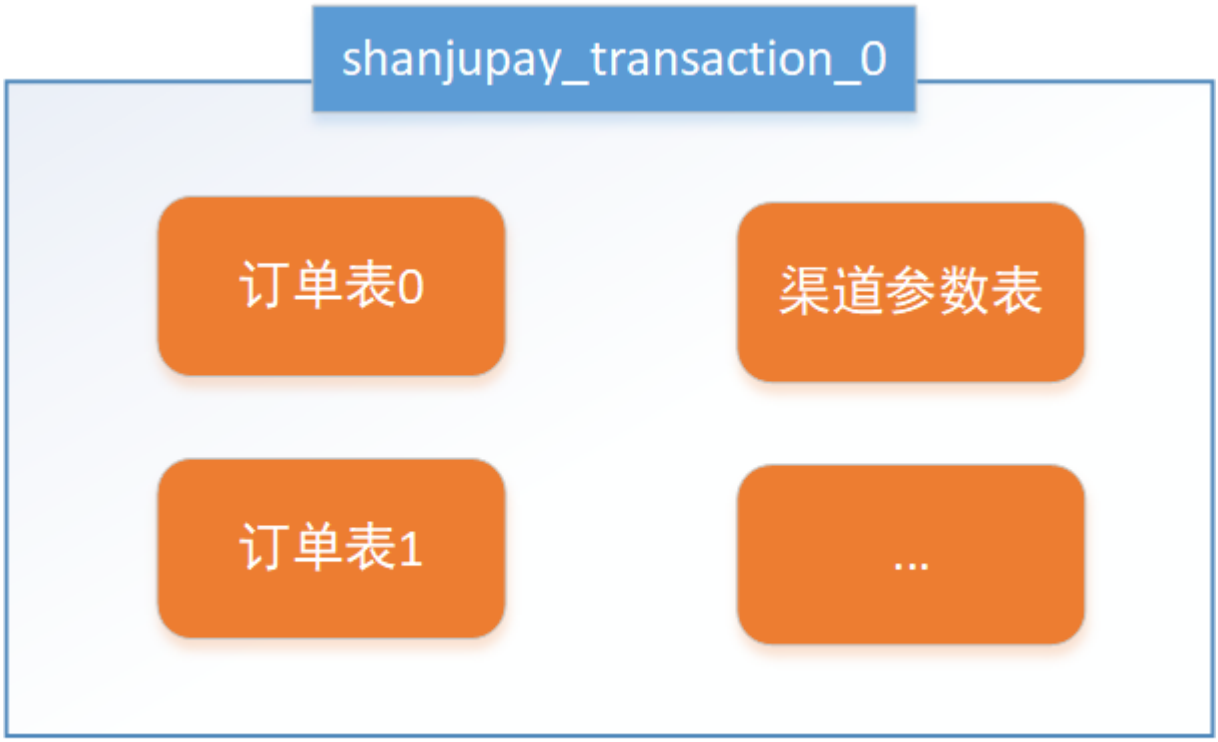
参考"Sharding-jdbc专题-v1.0.pdf"

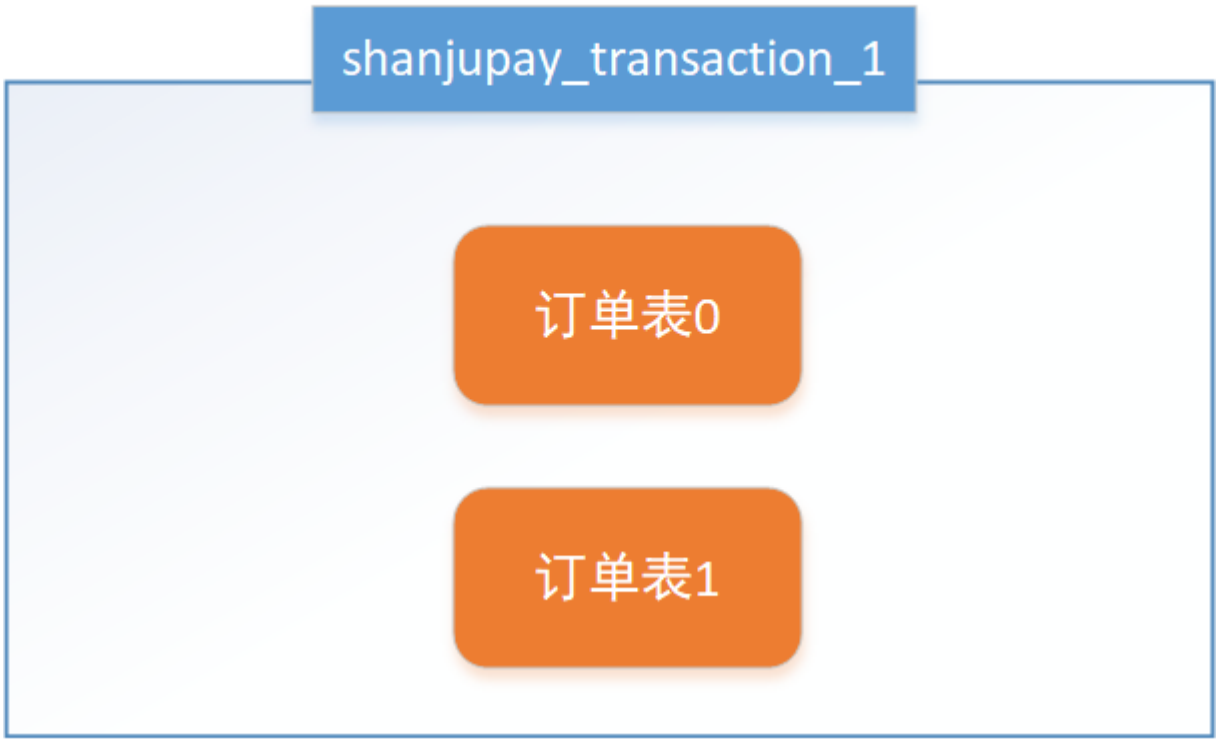
1.3 数据库设计

1、shanjupay_transaction数据库(交易服务)会存储订单数据(pay_order表)。首先对该数据库进行分库，相同商户的数据最好不要分散，否则查询相关信息要跨库，因此以商户ID作为分片键，分片策略采取商户ID % 2。其中其他表不做分库处理



2、然后对库内的订单表进行分表，根据需求此两表会以订单ID作为关联键联合查询， 因此以订单ID作为分片键，分片策略采取订单ID % 2， 最终形成如下数据库设计:





导入如下数据库脚本：

1、导入资料下shanjupay_transaction_0.sql

2、导入资料下shanjupay_transaction_1.sql

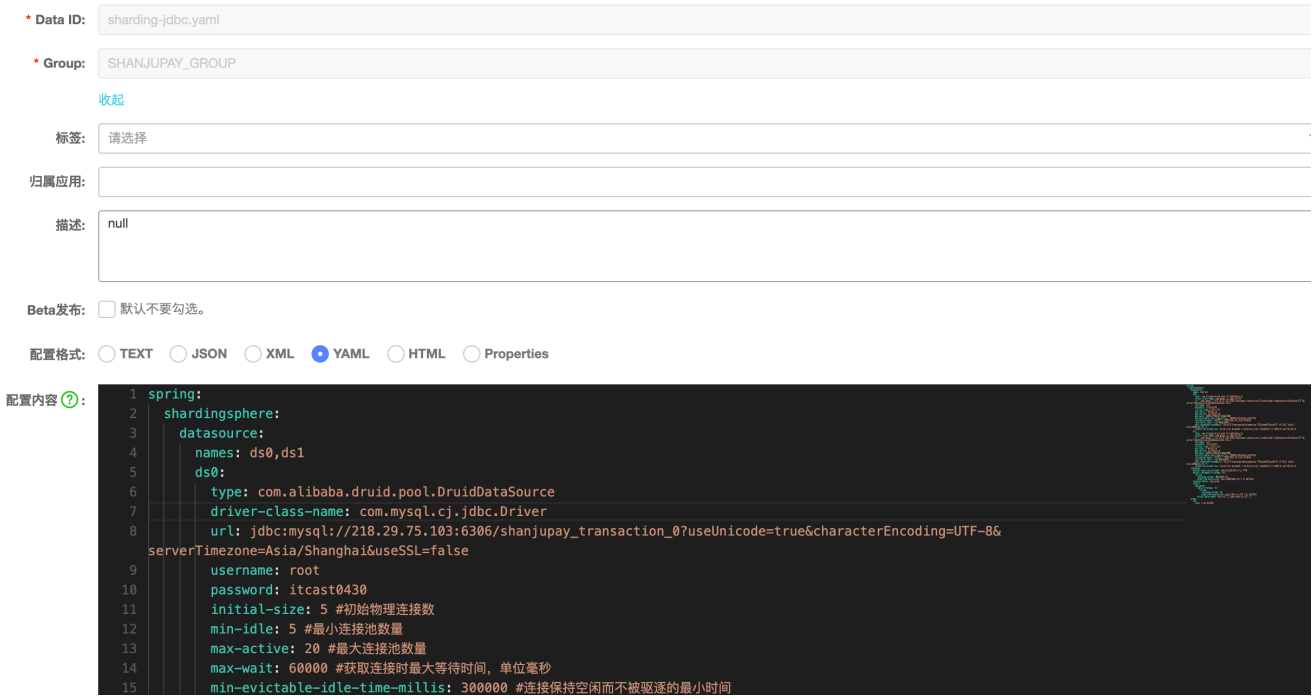
1.4 分库分表配置

1、在交易服务添加sharding-jdbc的依赖

```
<dependency>
    <groupId>org.apache.shardingsphere</groupId>
    <artifactId>sharding-jdbc-spring-boot-starter</artifactId>
    <version>4.0.0-RC1</version>
</dependency>
```

2、在Nacos中新建sharding-jdbc.yaml配置，Group: SHANJUPAY_GROUP

将下面配置中数据库连接地址、用户名、密码修改为自己的配置

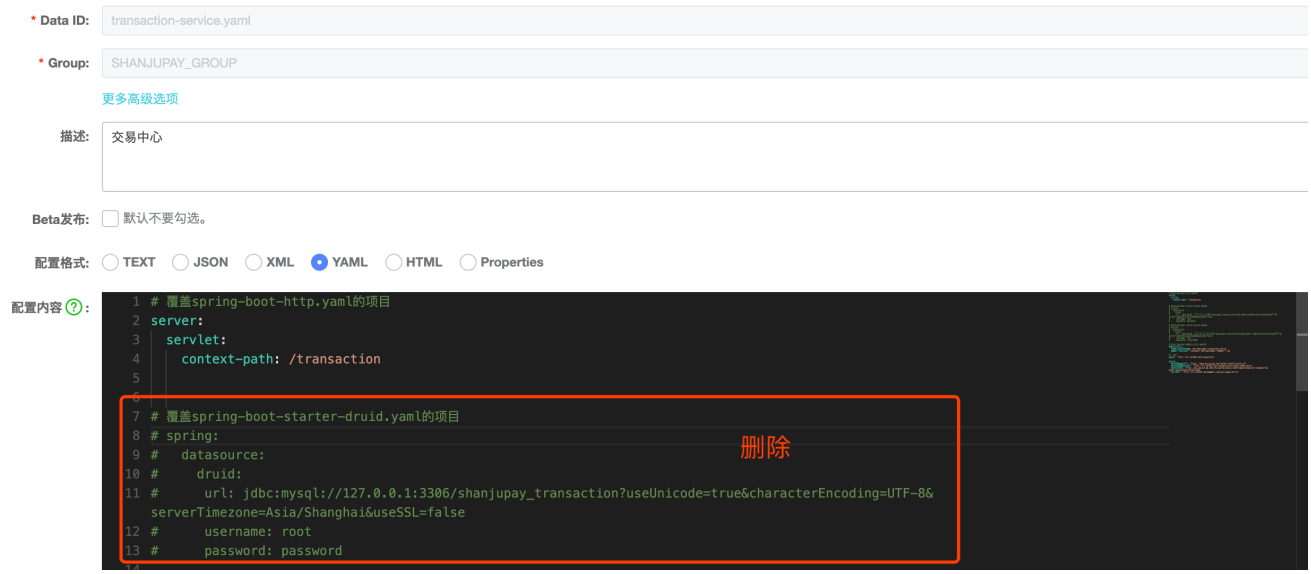



```
spring:
  shardingsphere:
    datasource:
      names: ds0,ds1
      ds0:
        type: com.alibaba.druid.pool.DruidDataSource
        driver-class-name: com.mysql.cj.jdbc.Driver
        url: jdbc:mysql://127.0.0.1:3306/shanjupay_transaction_0?
useUnicode=true&characterEncoding=UTF-8&serverTimezone=Asia/Shanghai&useSSL=false
        username: root
        password: password
        initial-size: 5 #初始物理连接数
        min-idle: 5 #最小连接池数量
        max-active: 20 #最大连接池数量
        max-wait: 60000 #获取连接时最大等待时间，单位毫秒

        min-evictable-idle-time-millis: 300000 #连接保持空闲而不被驱逐的最小时间
```

```
        validation-query: SELECT 1 FROM DUAL #用来检测连接是否有效的sql
        test-while-idle: true #申请连接的时候检测
        pool-prepared-statements: false #是否缓存preparedStatement，也就是PSCache。PSCache对支持游标的数据库性能提升巨大，比如说oracle。在mysql下建议关闭。
        connection-properties: druid.stat.mergeSql:true;druid.stat.slowSqlMillis:5000 #打开sql合并和慢sql记录
      ds1:
        type: com.alibaba.druid.pool.DruidDataSource
        driver-class-name: com.mysql.cj.jdbc.Driver
        url: jdbc:mysql://127.0.0.1:3306/shanjupay_transaction_1?useUnicode=true&characterEncoding=UTF-8&serverTimezone=Asia/Shanghai&useSSL=false
        username: root
        password: password
        initial-size: 5 #初始物理连接数
        min-idle: 5 #最小连接池数量
        max-active: 20 #最大连接池数量
        max-wait: 60000 #获取连接时最大等待时间，单位毫秒
        min-evictable-idle-time-millis: 300000 #连接保持空闲而不被驱逐的最小时间
        validation-query: SELECT 1 FROM DUAL #用来检测连接是否有效的sql
        test-while-idle: true #申请连接的时候检测
        pool-prepared-statements: false #是否缓存preparedStatement，也就是PSCache。PSCache对支持游标的数据库性能提升巨大，比如说oracle。在mysql下建议关闭。
        connection-properties: druid.stat.mergeSql:true;druid.stat.slowSqlMillis:5000 #打开sql合并和慢sql记录
    sharding:
      default-data-source-name: ds0 #未配置分片规则的表将通过默认数据源定位
      default-database-strategy: #分库策略
        inline:
          sharding-column: MERCHANT_ID
          algorithm-expression: ds$->{MERCHANT_ID % 2} #商户ID%2
      tables:
        pay_order:
          table-strategy: #分表策略
            inline:
              sharding-column: ID
              algorithm-expression: pay_order_$->{ID % 2} #订单ID%2
          actual-data-nodes: ds$->{0..1}.pay_order_$->{0..1}
    props:
      sql:
        show: true #打印SQL
```

3、在Nacos中编辑transaction-service.yaml，把原来的数据库配置删除



4、在shanjupay-transaction-service工程的bootstrap.yml中配置引入sharding-jdbc.yaml，并将删除原来的数据库配置spring-boot-starter-druid.yaml的引入

```
删除
- refresh: true
  data-id: spring-boot-starter-druid.yaml # spring boot starter druid配置
  group: COMMON_GROUP # 通用配置组
- refresh: true
  data-id: spring-boot-mybatis-plus.yaml # spring boot mybatisplus配置
  group: COMMON_GROUP # 通用配置组
- refresh: true
  data-id: spring-boot-freemarker.yaml # spring boot freemarker配置
  group: COMMON_GROUP # 通用配置组
- refresh: true
  data-id: spring-boot-redis.yaml # redis配置
  group: COMMON_GROUP # 通用配置组
- refresh: true
  data-id: xxl-job.yaml # xxl-job配置
  group: COMMON_GROUP # 通用配置组
- refresh: true
  data-id: spring-boot-starter-rocketmq.yaml # rocketmq配置
  group: COMMON_GROUP # 通用配置组
新增
- refresh: true
  data-id: sharding-jdbc.yaml # sharding-jdbc配置
  group: SHANJUPAY_GROUP # 闪聚配置组
```

```
- refresh: true
  data-id: sharding-jdbc.yaml # sharding-jdbc配置
  group: SHANJUPAY_GROUP # 闪聚配置组
```

1.5 测试

1、分别使用不同的商户（偶数id和奇数id）生成两个二维码。

2、分别扫这两个二维码，进行立即支付

3、观察shanjupay_transaction_0和shanjupay_transaction_1两个数据库中的订单信息保存情况。