5-6 | 引入 ShardingJDBC 配置标签记录表

分表配置：

```
Java
dataSources:
  user_master:  ##新表，重建的分表
    dataSourceClassName: com.zaxxer.hikari.HikariDataSource
    driver-class-name: com.mysql.cj.jdbc.Driver
    jdbcUrl: jdbc:mysql://cloud.db:8808/qiyu_live_user?
useUnicode=true&characterEncoding=utf8
    username: root
    password: root

  user_slave0:  ##新表，重建的分表
    dataSourceClassName: com.zaxxer.hikari.HikariDataSource
    driver-class-name: com.mysql.cj.jdbc.Driver
    jdbcUrl: jdbc:mysql://cloud.db:8809/qiyu_live_user?
useUnicode=true&characterEncoding=utf8
    username: root
    password: root

rules:
    - !READWRITE_SPLITTING
      dataSources:
        user_ds:
         staticStrategy:
           writeDataSourceName: user_master
           readDataSourceNames:
              - user_slave0
    - !SINGLE
      defaultDataSource: user_ds ## 不分表分分库的默认数据源
    - !SHARDING
      tables:
        t_user:
          actualDataNodes: user_ds.t_user_${(0..99).collect()
{it.toString().padLeft(2,'0')}}
          tableStrategy:
```

```
            standard:
              shardingColumn: user_id
              shardingAlgorithmName: t_user-inline
        t_user_tag:
          actualDataNodes: user_ds.t_user_tag_${(0..99).collect()
{it.toString().padLeft(2,'0')}}
          tableStrategy:
            standard:
              shardingColumn: user_id
              shardingAlgorithmName: t_user_tag-inline
      shardingAlgorithms:
        t_user-inline:
          type: INLINE
          props:
            algorithm-expression: t_user_${(user_id %
100).toString().padLeft(2,'0')}
        t_user_tag-inline:
          type: INLINE
          props:
            algorithm-expression: t_user_tag_${(user_id %
100).toString().padLeft(2,'0')}
props:
  sql-show: true
```