**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 |