Mycat

尚硅谷 JAVA 研究院

版本：V1.1

一、 Mycat 介绍

1、 是什么

1.1、 数据库中间件

前身是阿里的 cobar

2、 干什么的

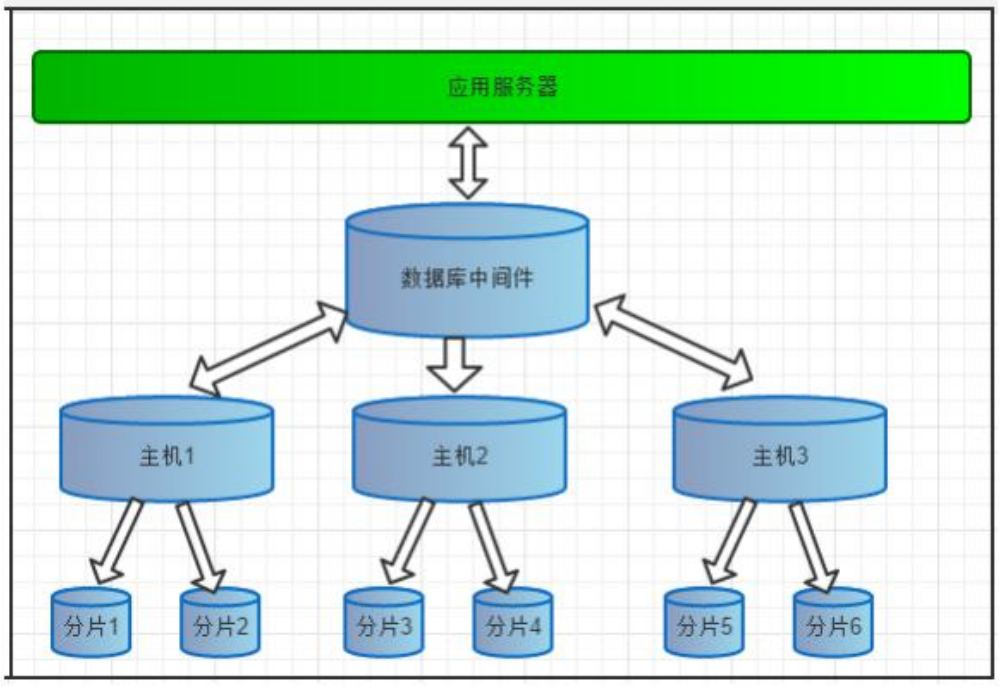
2.1、 读写分离

2.2、 数据分片

垂直拆分

水平拆分

垂直+水平拆分

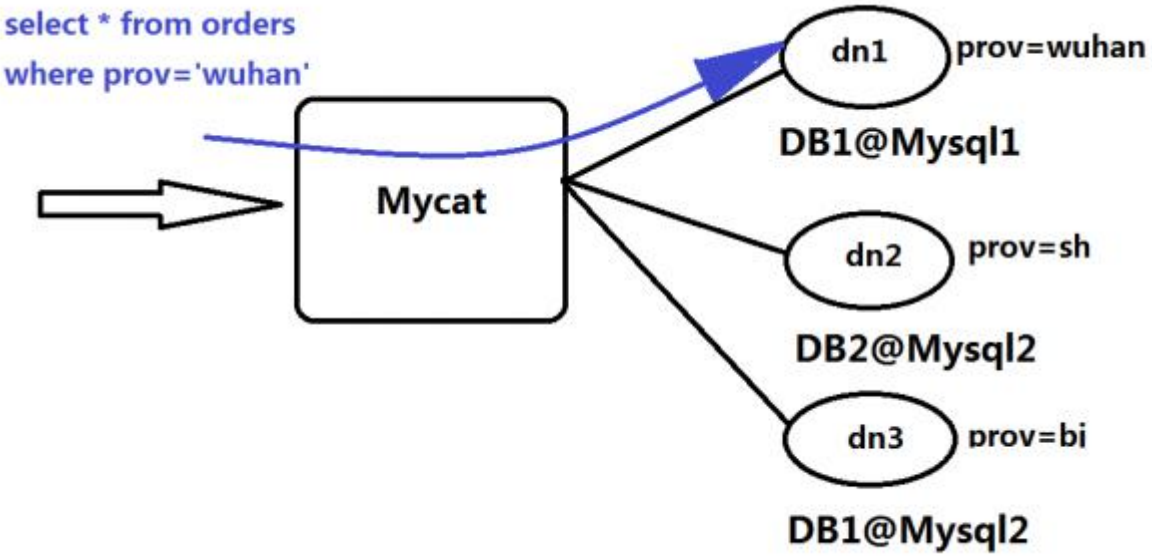


2.3、 多数据源整合

3、 原理

3.1、 “拦截”

Mycat 的原理中最重要的一个动词是“拦截”，它拦截了用户发送过来的 SQL 语句，首先对 SQL 语句做了一些特定的分析：如分片分析、路由分析、读写分离分析、缓存分析等，然后将此 SQL 发往后端的真实数据库，并将返回的结果做适当的处理，最终再返回给用户



这种方式把数据库的分布式从代码中解耦出来，程序员察觉不出来后台使用 mycat 还是 mysql。

二、安装启动

1、 安装解压

解压缩文件拷贝到 linux 下 /usr/local/

2、 配置文件介绍

（1） schema.xml

定义逻辑库，表、分片节点等内容

（2） rule.xml

定义分片规则

（3） server.xml

定义用户以及系统相关变量，如端口等

3、 配置文件修改

（1） 修改配置文件 schema.xml

```
<?xml version="1.0"?>
<!DOCTYPE mycat:schema SYSTEM "schema.dtd">
<mycat:schema xmlns:mycat="http://io.mycat/">
        <!--逻辑库 name 名称, checkSQLschema sqlMaxLimit 末尾是否要加 limit xxx-->
    <schema name="TESTDB" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1"> </schema>
        <!--逻辑库 name 名称, dataHost 引用的哪个 dataHost database:对应 mysql 的 database-->
    <dataNode name="dn1" dataHost="localhost1" database="db1" />
    <dataHost name="localhost1" maxCon="1000" minCon="10" balance="0"
                writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">
            <heartbeat>select user()</heartbeat>
            <!-- can have multi write hosts -->
            <writeHost host="hostM1" url="localhost:3306" user="root"
                        password="123456">
            </writeHost>
    </dataHost>
</mycat:schema>
```

（2） 修改配置文件 server.xml

```
<user name="root">
            <property name="password">654321</property>
            <property name="schemas">TESTDB</property>
```

```
</user>
```

4、 验证数据库访问情况

通过命令远程访问 mycat 涉及的机器上的 mysql 数据库是否可以正常登陆。

```
mysql -u 用户名 -p 密码 -h 机器 1IP -P3306

mysql -u 用户名 -p 密码 -h 机器 1IP -P3306
```

5、 启动程序

（1） 控制台启动 ：去 mycat/bin 目录下 mycat console

（2） 后台启动 ：去 mycat/bin 目录下 mycat start

6、 启动时问题解决

（1） 问题：域名解析失败

```
root@jack bin]# ./mycat console
Running Mycat-server...
wrapper  | --> Wrapper Started as Console
wrapper  | Launching a JVM...
wrapper  | JVM exited while loading the application.
jvm 1    | 错误: 代理抛出异常错误: java.net.MalformedURLException: Local host name unknown: java.net.UnknownHostException: jack.atguigu: jack.atguigu: 域名解析暂时失败
wrapper  | Launching a JVM...
```

（2） 解决办法

用 vim 修改 /etc/hosts 文件，在 127.0.0.1 后面增加你的机器名

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4 cocoon.atguigu
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
~
```

修改后重新启动网络服务

```
[root@jack ~]# service network restart
正在关闭接口 eth0：  设备状态：3（断开连接）
                                                           [确定]
关闭环回接口：                                             [确定]
弹出环回接口：                                             [确定]
弹出界面 eth0：  活跃连接状态：激活中
活跃连接路径：/org/freedesktop/NetworkManager/ActiveConnection/1
状态：激活的
连接被激活
                                                           [确定]
[root@jack ~]#
```

7、 登录

（1） 后台管理窗口

登录命令：

mysql -u 用户名 -p 密码 -P9066 -h 启动 mycat 的机器 IP

操作命令：

show database

```
mysql> show database;
+----------+
| DATABASE |
+----------+
| TESTDB   |
+----------+
1 row in set (0.01 sec)
```

show @@help

```
mysql> show @@help;
+-----------------------------------------+-----------------------------------------+
| STATEMENT                               | DESCRIPTION                             |
+-----------------------------------------+-----------------------------------------+
| show @@time.current                     | Report current timestamp                |
| show @@time.startup                     | Report startup timestamp                |
| show @@version                          | Report Mycat Server version             |
| show @@server                           | Report server status                    |
| show @@threadpool                       | Report threadPool status                |
| show @@database                         | Report databases                        |
| show @@datanode                         | Report dataNodes                        |
| show @@datanode where schema = ?        | Report dataNodes                        |
| show @@datasource                       | Report dataSources                      |
| show @@datasource where dataNode = ?    | Report dataSources                      |
| show @@datasource.synstatus             | Report datasource data synchronous      |
| show @@datasource.syndetail where name=? | Report datasource data synchronous detail |
| show @@datasource.cluster               | Report datasource galary cluster variables |
| show @@processor                        | Report processor status                 |
| show @@command                          | Report commands status                  |
| show @@connection                       | Report connection status                |
| show @@cache                            | Report system cache usage               |
| show @@backend                          | Report backend connection status        |
| show @@session                          | Report front session details            |
| show @@connection.sql                   | Report connection sql                   |
| show @@sql.execute                      | Report execute status                   |
| show @@sql.detail where id = ?          | Report execute detail status            |
| show @@sql                              | Report SQL list                         |
| show @@sql.high                         | Report Hight Frequency SQL              |
| show @@sql.slow                         | Report slow SQL                         |
| show @@sql.resultset                    | Report BIG RESULTSET SQL                |
```

（2） 数据窗口

登录命令：

mysql -u 用户名 -p 密码 -P8066 -h 启动 mycat 的机器 IP

操作命令：

同 mysql

三、读写分离

1、 配置文件修改

（1） 修改配置文件 schema.xml

```
<?xml version="1.0"?>
<!DOCTYPE mycat:schema SYSTEM "schema.dtd">
<mycat:schema xmlns:mycat="http://io.mycat/">

<schema name="TESTDB" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1">
        </schema>
        <dataNode name="dn1" dataHost="host1" database="atguigu_mc" />
        <dataHost name="host1" maxCon="1000" minCon="10" balance="2"
                writeType="0" dbType="mysql" dbDriver="native" switchType="1"
slaveThreshold="100">
                <heartbeat>select user()</heartbeat>
                <writeHost host="hostm1" url="192.168.67.1:3306" user="root"
                        password="123123">
```

```
<!--读库（从库）的配置 -->
                    <readHost host="hosts1" url="192.168.67.131:3306" user="root"
                              password="123123">
                    </readHost>
                </writeHost>
            </dataHost>
</mycat:schema>
```

Balance——负载均衡类型，目前的取值有 4 种：

1. balance="0"，不开启读写分离机制，所有读操作都发送到当前可用的 writeHost 上。

2. balance="1"，全部的 readHost 与 stand by writeHost 参与 select 语句的负载均衡，简单的说，当双主双从模式(M1->S1，M2->S2，并且 M1 与 M2 互为主备)，正常情况下，M2,S1,S2 都参与 select 语句的负载均衡。

3. balance="2"，所有读操作都随机的在 writeHost、readhost 上分发。

4. balance="3"，所有读请求随机的分发到 readhost 执行，writerHost 不负担读压力

2、 重启 Mycat

重启 Mycat，让新配置生效

3、 验证

（1） 创建表

```
create table t_replica
(       id    int auto_increment ,
        name varchar(200)
);
```

（2） 插入数据

分别在两个库下插入：insert into t_replica(name) values (@@hostname);

（3） 验证

然后再 mycat 下执行：select * from t_replica;

能够验证读写分离

四、分库

1、 选择分库表

选择不会出现关联查询的表分到别的库里。

分离出来的表为：客户表

建表语句：

```
#客户表   rows:20 万
CREATE TABLE customer(
    id INT AUTO_INCREMENT,
    NAME VARCHAR(200),
    PRIMARY KEY(id)
);
------------------------

#订单表    rows:600 万
CREATE TABLE orders(
    id INT AUTO_INCREMENT,
```

```
    order_type INT,
    customer_id INT,
    amount DECIMAL(10,2),
    PRIMARY KEY(id)
);

#订单详细表        rows:600 万
CREATE TABLE orders_detail(
    id INT AUTO_INCREMENT,
    detail VARCHAR(2000),
    order_id INT,
    PRIMARY KEY(id)
);

#订单状态字典表     rows:20
CREATE TABLE dict_order_type(
    id INT AUTO_INCREMENT,
    order_type VARCHAR(200),
    PRIMARY KEY(id)
);


 select o.*,od.detail,d.order_type
from orders   o
inner join orders_detail   od on o.id =od.order_id
 inner join dict_order_type d on o.order_type=d.id
where o.customer_id=xxxx
```

2、 配置文件修改

（1） 修改配置文件 schema.xml

```
<mycat:schema xmlns:mycat="http://io.mycat/">
        <schema name="TESTDB" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1">
                <table name="customer" dataNode="dn2" ></table>
        </schema>
        <dataNode name="dn1" dataHost="host1" database="atguigu_mc" />
        <dataNode name="dn2" dataHost="host2" database="atguigu_sm" />
        <dataHost name="host1" maxCon="1000" minCon="10" balance="0"
                        writeType="0"       dbType="mysql"       dbDriver="native"       switchType="1"
slaveThreshold="100">
                <heartbeat>select user()</heartbeat>
                <writeHost host="hostm1" url="192.168.67.1:3306" user="root"
                                password="123123">
                </writeHost>
        </dataHost>
        <dataHost name="host2" maxCon="1000" minCon="10" balance="0"
                        writeType="0"       dbType="mysql"       dbDriver="native"       switchType="1"
slaveThreshold="100">
                <heartbeat>select user()</heartbeat>
                <writeHost host="hostm2" url="192.168.67.145:3306" user="root"
                                password="123123">
                </writeHost>
        </dataHost>

</mycat:schema>
```

3、 重启 Mycat

重启 Mycat，让新配置生效

4、 验证

（1） 连接 Mycat 执行 4 个表的建表语句

（2） 执行成功后，分别去两个数据库查看，客户表在一个库，订单相关 3 张表在另一个库

五、水平分表

1、 选择水平分表

（1） 订单表 orders 表已经有 600 万数据超过 MySQL 单表数据瓶颈，需要进行水平分表

（2） 鉴于订单表特性，无论是按照 id、时间进行分表均不合适，应该保证每个人的订单在同一张表里。所以应该按照客户 id（customer_id）进行分表，具体做法：按照客户 id 取模平均分配到两张表里。

2、 配置文件修改

（1） 修改配置文件 schema.xml

```
<?xml version="1.0"?>
<!DOCTYPE mycat:schema SYSTEM "schema.dtd">
<mycat:schema xmlns:mycat="http://io.mycat/">

        <schema name="TESTDB" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1">
                <table name="customer" dataNode="dn2" ></table>
                <table name="orders" dataNode="dn1,dn2"   rule="mod_rule" ></table>
        </schema>
        <dataNode name="dn1" dataHost="host1" database="atguigu_mc" />
        <dataNode name="dn2" dataHost="host2" database="atguigu_sm" />
        <dataHost name="host1" maxCon="1000" minCon="10" balance="2"
```

```
                                writeType="0"        dbType="mysql"        dbDriver="native"        switchType="1"
slaveThreshold="100">
                        <heartbeat>select user()</heartbeat>
                        <writeHost host="hostm1" url="192.168.67.1:3306" user="root"
                                                password="123123">
                        </writeHost>
                </dataHost>
                <dataHost name="host2" maxCon="1000" minCon="10" balance="0"
                                writeType="0"        dbType="mysql"        dbDriver="native"        switchType="1"
slaveThreshold="100">
                        <heartbeat>select user()</heartbeat>
                        <writeHost host="hostm2" url="192.168.67.1:3306" user="root"
                                                password="123123">
                        </writeHost>
                </dataHost>
```

（2） 修改配置文件 rule.xml

```
    <tableRule name="mod_rule">
                        <rule>
                                <columns>customer_id</columns>
                                <algorithm>mod-long</algorithm>
                        </rule>
                </tableRule>
        ……


    <function name="mod-long" class="io.mycat.route.function.PartitionByMod">
                        <!-- how many data nodes -->
```

```
                <property name="count">2</property>
            </function>
```

3、 重启 Mycat

先在另一个库里创建订单表，之后重启 Mycat，让新配置生效

5、 验证

（1） 连接 Mycat 向订单表插入数据

```
注：表名后必须加上相关字段，告知 Mycat，哪个字段是 customer_id
INSERT INTO orders(id,order_type,customer_id,amount) values(1,101,100,100100);
INSERT INTO orders(id,order_type,customer_id,amount) VALUES(2,101,100,100300);
INSERT INTO orders(id,order_type,customer_id,amount) VALUES(3,101,101,120000);
INSERT INTO orders(id,order_type,customer_id,amount) VALUES(4,101,101,103000);
INSERT INTO orders(id,order_type,customer_id,amount) VALUES(5,102,101,100400);
INSERT INTO orders(id,order_type,customer_id,amount) VALUES(6,102,100,100020);
```

（2） 执行成功后，分别去两个数据库查看，两个客户的订单已分到两个表里。

六、跨库关联查询

1、 ER 表

为了相关联的表的行尽量分在一个库下,订单详情表（orders_detail）与订单表紧密关联，订单表分表，为保证关联查询结果正确，必须对订单详情表进行分表，用 ER 表的方式参考订单表的分表进行分表。

（1） 修改配置文件 schema.xml

```
<?xml version="1.0"?>
<!DOCTYPE mycat:schema SYSTEM "schema.dtd">
<mycat:schema xmlns:mycat="http://io.mycat/">

        <schema name="TESTDB" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1">
                <table name="customer" dataNode="dn2" ></table>
                <table name="orders" dataNode="dn1,dn2"   rule="mod_rule" >
                    <childTable  name="orders_detail"  primaryKey="id"  joinKey="order_id"  parentKey="id"
/>
                </table>

        </schema>
        <dataNode name="dn1" dataHost="host1" database="atguigu_mc" />
        <dataNode name="dn2" dataHost="host2" database="atguigu_sm" />
    ……
```

（2） 重启 Mycat

先在另一个库里创建订单详情表，之后重启 Mycat，让新配置生效

（3） 验证

连接 Mycat 向订单表插入数据

```
注：表名后必须加上相关字段，告知 Mycat 关键字段
INSERT INTO orders_detail(id,detail,order_id) values(1,'detail1',1);
INSERT INTO orders_detail(id,detail,order_id) VALUES(2,'detail1',2);
INSERT INTO orders_detail(id,detail,order_id) VALUES(3,'detail1',3);
INSERT INTO orders_detail(id,detail,order_id) VALUES(4,'detail1',4);
INSERT INTO orders_detail(id,detail,order_id) VALUES(5,'detail1',5);
```

```
INSERT INTO orders_detail(id,detail,order_id) VALUES(6,'detail1',6);
```

执行成功后，分别去两个数据库查看，两个客户的订单详情已分到两个表里。可以把订单表和订单详情表关联查询，可以查询到正确结果。

2、 全局表

设定为全局的表，会直接复制给每个数据库一份，所有写操作也会同步给多个库。

所以全局表一般不能是大数据表或者更新频繁的表。

一般是字典表或者系统表为宜。

（1） 修改配置文件 schema.xml

```
<?xml version="1.0"?>
<!DOCTYPE mycat:schema SYSTEM "schema.dtd">
<mycat:schema xmlns:mycat="http://io.mycat/">

        <schema name="TESTDB" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1">
            <table name="customer" dataNode="dn2" ></table>
            <table name="orders" dataNode="dn1,dn2"    rule="mod_rule" ></table>
             <table name="dict_order_type" dataNode="dn1,dn2" type="global" ></table>
        </schema>
        <dataNode name="dn1" dataHost="host1" database="atguigu_mc" />
        <dataNode name="dn2" dataHost="host2" database="atguigu_sm" />
……
```

（2） 重启 Mycat

先在另一个库里创建订单状态字典表，之后重启 Mycat，让新配置生效

（3） 验证

连接 Mycat 向订单状态字典表插入数据

注：表名后必须加上相关字段，告知 Mycat 关键字段

```
INSERT INTO dict_order_type(id,order_type) values(101,'type1');

INSERT INTO dict_order_type(id,order_type) VALUES(102,'type2');
```

执行成功后，分别去两个数据库查看，两个订单状态字典表都有全量数据。

七、全局序列

1、 本地文件方式

不推荐，如在 Mycat 主机中用本地文件方式创建全局序列，当这台机器宕机时会出现，序列文件丢失，造成序列冲突问题

2、 数据库方式

2.1、 原理

利用数据库一个表 来进行计数累加。但是并不是每次生成序列都读写数据库，这样效率太低 mycat 会预加载一部分号段到 mycat 的内存中，这样大部分读写序列都是在内存中完成的。如果内存中的号段用完了 mycat 会再向数据库要一次。

问题：那如果 mycat 崩溃了 ，那内存中的序列岂不是都没了？

是的。如果是这样，那么 mycat 启动后会向数据库申请新的号段，原有号段会弃用。也就是说如果 mycat 重启，那么损失是当前的号段没用完的号码，但是不会因此出现主键重复。

2.2、 创建脚本

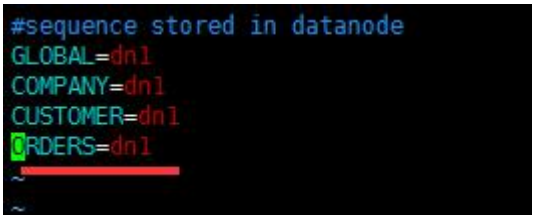
```
#win10
#建表
CREATE TABLE MYCAT_SEQUENCE (NAME VARCHAR(50) NOT NULL,current_value INT NOT
NULL,increment INT NOT NULL DEFAULT 100, PRIMARY KEY(NAME)) ENGINE=INNODB;

#新建函数
DELIMITER $$

CREATE FUNCTION mycat_seq_currval(seq_name VARCHAR(50)) RETURNS VARCHAR(64)
DETERMINISTIC
BEGIN
DECLARE retval VARCHAR(64);
SET retval="-999999999,null";
SELECT CONCAT(CAST(current_value AS CHAR),",",CAST(increment AS CHAR)) INTO retval FROM
MYCAT_SEQUENCE WHERE NAME = seq_name;
RETURN retval;
END $$
DELIMITER;

DELIMITER $$
CREATE FUNCTION mycat_seq_setval(seq_name VARCHAR(50),VALUE INTEGER) RETURNS VARCHAR(64)
DETERMINISTIC
BEGIN
UPDATE MYCAT_SEQUENCE
SET current_value = VALUE
```

```
WHERE NAME = seq_name;
RETURN mycat_seq_currval(seq_name);
END $$
DELIMITER ;

DELIMITER $$
CREATE FUNCTION mycat_seq_nextval(seq_name VARCHAR(50)) RETURNS VARCHAR(64)
DETERMINISTIC
BEGIN
UPDATE MYCAT_SEQUENCE
SET current_value = current_value + increment WHERE NAME = seq_name;
RETURN mycat_seq_currval(seq_name);
END $$
DELIMITER;


SELECT * FROM MYCAT_SEQUENCE

TRUNCATE TABLE MYCAT_SEQUENCE

##增加要用的序列
INSERT INTO MYCAT_SEQUENCE(NAME,current_value,increment) VALUES ('ORDERS', 400000,
100);
```

2.3、 修改配置

（1） 修改配置文件 sequence_db_conf.properties

vim sequence_db_conf.properties



意思是 ORDERS 这个序列在 dn1 这个节点上，具体 dn1 节点是哪台机子，请参考 schema.xml

（2） 修改配置文件 server.xml

vim server.xml

（3） 重启 Mycat

2.4、 验证

多次插入数据，验证全局序列 ID

```
insert into `orders`(id,amount,customer_id,order_type) values(next value for MYCATSEQ_ORDERS,1000,101,102);
```

重启 Mycat 后，再次插入数据，再验证。

3、 时间戳方式

用时间戳方式不会出现冲突问题，但长度有 18 位，比较长

4、 自主生成

可以根据业务逻辑组合，或者可以利用 redis 的单线程原子性 incr 来生成序列。