

TTS 11.0 COOKBOOK

(NSD RDBMS2 DAY03)

版本编号 11.0

2019-06

达内 IT 培训集团

NSD RDBMS2 DAY03

1. 案例 1：部署 MyCAT 服务

• 问题

- 数据库服务器 192.168.4.53 使用 db1 库存储数据
- 数据库服务器 192.168.4.54 使用 db2 库存储数据
- 数据库服务器 192.168.4.55 使用 db3 库存储数据
- 主机 192.168.4.56 运行 mycat 服务，逻辑库名称 TESTDB，连接用户名为 adminplj，密码 123qqq...A
- 客户端 192.168.4.50 访问 mycat 服务测试配置

• 方案

准备 5 台虚拟主机；其中主机 192.168.4.56 作为 mycat 服务器，192.168.4.53、192.168.4.54、192.168.4.55 运行数据库服务，192.168.4.50 作为客户端。具体如图-1 所示：

主机名	角色	数据库	IP地址
client50	客户端	无	192.168.4.50/24
mycat56	分片服务器	无	192.168.4.56/24
mysql53	数据库服务器	db1	192.168.4.53/24
mysql54	数据库服务器	db2	192.168.4.54/24
mysql55	数据库服务器	db3	192.168.4.55/24

图-1

• 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置分片服务器（192.168.4.56）

1) 部署 MyCat 运行环境

```
]# yum -y install java-1.8.0-openjdk //安装 JDK
已安装：
```

```
java-1.8.0-openjdk.x86_64 1:1.8.0.161-2.b14.el7
```

作为依赖被安装:

```
alsa-lib.x86_64 0:1.1.4.1-2.el7
copy-jdk-configs.noarch 0:3.3-2.el7
giflib.x86_64 0:4.1.6-9.el7
java-1.8.0-openjdk-headless.x86_64 1:1.8.0.161-2.b14.el7
javapackages-tools.noarch 0:3.4.1-11.el7
libXtst.x86_64 0:1.2.3-1.el7
libxslt.x86_64 0:1.1.28-5.el7
lksctp-tools.x86_64 0:1.0.17-2.el7
python-javapackages.noarch 0:3.4.1-11.el7
python-lxml.x86_64 0:3.2.1-4.el7
tzdata-java.noarch 0:2018c-1.el7
```

完毕!

```
[root@mycat56 ~]# which java //查看命令
/usr/bin/java
[root@mycat56 ~]# java -version //显示版本
openjdk version "1.8.0_161"
OpenJDK Runtime Environment (build 1.8.0_161-b14)
OpenJDK 64-Bit Server VM (build 25.161-b14, mixed mode)
```

2) 安装提供服务的软件包

```
[root@mycat56 ~]# tar -zxvf Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz //
解压源码
```

```
mycat/bin/wrapper-linux-ppc-64
.....
mycat/version.txt
mycat/conf/log4j2.xml
mycat/bin/init_zk_data.sh
mycat/bin/startup_nowrap.sh
mycat/bin/dataMigrate.sh
mycat/bin/rehash.sh
mycat/logs/
mycat/catlet/
[root@mycat56 ~]#
```

```
[root@mycat56 ~]# mv mycat /usr/local/ //移动目录 (非必须操作)
```

```
[root@mycat56 ~]# ls /usr/local/mycat/ //查看文件列表
bin catlet conf lib logs version.txt
[root@mycat56 ~]#
```

3) 设置连账号

```
]# vim /usr/local/mycat/conf/server.xml
<user name="root"> //连接 mycat 服务时使用的用户名
    <property name="password">123456</property> //用户连接 mycat 用户时使用的密码
    <property name="schemas">TESTDB</property> //逻辑库名
</user>
<user name="user">
    <property name="password">user</property>
    <property name="schemas">TESTDB</property>
    <property name="readOnly">true</property> //只读权限, 连接 mycat 服务
```

后只有读记录的权限,不写这一行则是可读可写

```
</user>
:wq
```

4) 配置数据分片

使用 sed 删除不需要的配置行(可选操作)

```
[root@mycat56 conf]# wc -l schema.xml //删除前查看总行数
77 /root/schema.xml

[root@mycat56 conf]# sed -i '56,77d' schema.xml //删除无关的配置行
[root@mycat56 conf]# sed -i '39,42d' schema.xml
[root@mycat56 conf]# sed -i '16,18d' schema.xml

[root@mycat56 conf]# wc -l schema.xml //删除后查看总行数
48 schema.xml

[root@mycat56 conf]# vim /usr/local/mycat/conf/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"> //对 TESTDB
库下的表做分片存储
        <!-- auto sharding by id (long) -->

        <table name="travelrecord" dataNode="dn1,dn2,dn3"
rule="auto-sharding-long" /> //对 travelrecord 表做分片存储

        <!-- global table is auto cloned to all defined data nodes ,so can
join
        with any table whose sharding node is in the same data node
-->
        <table name="company" primaryKey="ID" type="global"
dataNode="dn1,dn2,dn3" /> //对 company 表做分片存储

        <table name="goods" primaryKey="ID" type="global"
dataNode="dn1,dn2,dn3" />

        <!-- random sharding using mod sharind rule -->

        <table name="hotnews" dataNode="dn1,dn2,dn3"
rule="mod-long" />

        <table name="employee" primaryKey="ID" dataNode="dn1,dn2,dn3"
rule="sharding-by-intfile" />

        <table name="customer" primaryKey="ID" dataNode="dn1,dn2,dn3"
rule="sharding-by-intfile">
            <childTable name="orders" primaryKey="ID"
joinKey="customer_id"
parentKey="id">
                <childTable name="order_items" joinKey="order_id"
parentKey="id" />
            </childTable>
            <childTable name="customer_addr" primaryKey="ID"
joinKey="customer_id"
parentKey="id" />
        </table>
        <!-- <table name="oc_call" primaryKey="ID" dataNode="dn1$0-743"
rule="latest-month-calldate">
```

```

        /> -->
    </schema>

    //定义数据库主机名及存储数据的库
    <dataNode name="dn1" dataHost="localhost53" database="db1" />
    <dataNode name="dn2" dataHost="localhost54" database="db2" />
    <dataNode name="dn3" dataHost="localhost55" database="db3" />

    //定义 localhost53 主机名对应的数据库服务器 ip 地址
    <dataHost name="localhost53" maxCon="1000" minCon="10" balance="0"
        writeType="0" dbType="mysql" dbDriver="native"
switchType="1" slaveThreshold="100">
        <heartbeat>select user()</heartbeat>
        <writeHost host="hostM53" url="192.168.4.53:3306" user="adminplj"
            password="123qqq...A">
        </writeHost>
    </dataHost>

    //定义 localhost54 主机名对应的数据库服务器 ip 地址
    <dataHost name="localhost54" maxCon="1000" minCon="10" balance="0"
        writeType="0" dbType="mysql" dbDriver="native" switchType="1"
slaveThreshold="100">
        <heartbeat>select user()</heartbeat>
        <writeHost host="hostM54" url="192.168.4.54:3306" user="adminplj"
            password="123qqq...A">
        </writeHost>
    </dataHost>

    //定义 localhost55 主机名对应的数据库服务器 ip 地址
    <dataHost name="localhost55" maxCon="1000" minCon="10" balance="0"
        writeType="0" dbType="mysql" dbDriver="native" switchType="1"
slaveThreshold="100">
        <heartbeat>select user()</heartbeat>
        <writeHost host="hostM55" url="192.168.4.55:3306" user="adminplj"
            password="123qqq...A">
        </writeHost>
    </dataHost>
:wq

```

5) 配置数据库服务器

根据分片文件的设置在对应的数据库服务器上创建存储数据的数据库

```

mysql> create database db1;    //在数据库 53 上, 创建 db1 库
mysql> create database db2;    //在数据库 54 上, 创建 db2 库
mysql> create database db3;    //在数据库 55 上, 创建 db3 库

```

根据分片文件配置, 在对应的数据库服务器上创建授权用户(3 台数据库服务器都要添加, 在数据库服务器本机管理员 root 用户登录后执行授权命令)

```

mysql> grant all on *.* to adminplj@"%" identified by "123qqq...A" ; //在数据库服
务器 192.168.4.53 执行

mysql> grant all on *.* to adminplj@"%" identified by "123qqq...A" ; //在数据库服
务器 192.168.4.54 执行

mysql> grant all on *.* to adminplj@"%" identified by "123qqq...A" ; //在数据库服
务器 192.168.4.55 执行

```

6) 启动 mycat 服务

测试授权用户：在 192.168.4.56 主机，使用授权用户分别连接 3 台数据库服务器，若连接失败，请检查数据库服务器是否有对应的授权用户。

```
[root@mycat56 ~]# which mysql || yum -y install mariadb //安装提供 mysql 命令的软件包
```

```
//连接数据库服务器 192.168.4.53
```

```
[root@mycat56 ~]# mysql -h192.168.4.53 -uadminplj -p123qqq...A
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 5.7.17 MySQL Community Server (GPL)
```

```
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> exit; //连接成功 断开连接
Bye
[root@mycat56 ~]#
```

```
//连接数据库服务器 192.168.4.54
```

```
[root@mycat56 ~]# mysql -h192.168.4.54 -uadminplj -p123qqq...A
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 5.7.17 MySQL Community Server (GPL)
```

```
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> exit; //连接成功 断开连接
Bye
[root@mycat56 ~]#
```

```
//连接数据库服务器 192.168.4.54
```

```
[root@mycat56 ~]# mysql -h192.168.4.55 -uadminplj -p123qqq...A
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 49
Server version: 5.7.17 MySQL Community Server (GPL)
```

```
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> exit ; //连接成功 断开连接
```

启动服务

```
[root@mycat56 ~]# /usr/local/mycat/bin/mycat start
Starting Mycat-server...
[root@mycat56 ~]#
```

查看服务状态

```
[root@mycat56 ~]# netstat -utnlp | grep :8066 //查看端口
tcp6      0      0 :::8066          :::*              LISTEN      2924/java
[root@mycat56 ~]#

[root@mycat56 ~]# ps -C java //查看进程
  PID TTY          TIME CMD
 2924 ?        00:00:01 java
[root@mycat56 ~]#
```

步骤二：测试配置

1) 客户端访问

在客户端 192.168.4.50 连接分片服务器，访问数据

命令：mysql -hmycat 主机的 IP -P 端口号 -u 用户 -p 密码

```
[root@client50 ~]# mysql -h192.168.4.56 -P8066 -uroot -p123456
mysql> show databases; //显示已有的库
+-----+
| DATABASE |
+-----+
| TESTDB   |
+-----+
1 row in set (0.00 sec)

mysql> USE TESTDB; //进入 TESTDB 库
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
mysql> show tables; //显示已有的表，配置文件里定义的表名
+-----+
| Tables in TESTDB |
+-----+
| company          |
| customer          |
| customer_addr    |
| employee         |
| goods            |
| hotnews          |
| orders           |
| order_items      |
| travelrecord     |
+-----+
9 rows in set (0.00 sec)

mysql>exit; //断开连接
```

2. 案例 2：连接分片服务器存储数据

- 问题：具体要求如下

- 1) 练习 sharding-by-intfile 分片规则的使用
- 2) 练习 mod-long 分片规则的使用

- 步骤

实现此案例需要按照如下步骤进行。

步骤一：练习 sharding-by-intfile 分片规则的使用

1) 查看配置文件，得知使用 sharding-by-intfile 分片规则的表名

```
[root@mycat56 ~]# vim /usr/local/mycat/conf/schema.xml
<table name="employee" primaryKey="ID" dataNode="dn1,dn2,dn3"
      rule="sharding-by-intfile" />
:wq
```

2) 查看规则文件，得知 sharding-by-intfile 分片规则使用的函数

```
[root@mycat56 ~]# vim /usr/local/mycat/conf/rule.xml
<tableRule name="sharding-by-intfile">
  <rule>
    <columns>sharding_id</columns> //数据分片字段名
    <algorithm>hash-int</algorithm> //使用的函数名
  </rule>
</tableRule>

<function name="hash-int"
  class="io.mycat.route.function.PartitionByFileMap">
  <property name="mapFile">partition-hash-int.txt</property> //函数调
用的配置文件
</function>
:wq
```

3) 修改函数配置文件,添加 dn3 数据节点

```
[root@mycat56 ~]# vim /usr/local/mycat/conf/partition-hash-int.txt
10000=0 //当 sharding_id 字段的值是 10000 时,数据存储在数据节点 dn1 里
10010=1 //当 sharding_id 字段的值是 10010 时,数据存储在数据节点 dn2 里
10020=2 //当 sharding_id 字段的值是 10020 时,数据存储在数据节点 dn3 里
:wq
```

4) 重启 mycat 服务，使其修改有效

```
[root@mycat56 ~]# /usr/local/mycat/bin/mycat stop //停止服务
Stopping Mycat-server...
Stopped Mycat-server.
[root@mycat56 conf]# netstat -utnlp | grep :8066 //无端口
[root@mycat56 conf]# ps -C java //无进程
[root@mycat56 conf]#
```



```
[root@mycat56 conf]# /usr/local/mycat/bin/mycat start //启动服务
Starting Mycat-server...
[root@mycat56 conf]#
[root@mycat56 conf]# netstat -utnlp | grep :8066 //有端口
tcp6      0      0 :::8066          :::*              LISTEN      1364/java
[root@mycat56 conf]#
[root@mycat56 conf]# ps -C java //有进程
PID TTY          TIME CMD
 1125 ?            00:00:01 java
[root@mycat56 conf]#
```

5) 客户端连接分片服务器, 存取数据

```
]#mysql -h192.168.4.56 -P8066 -uroot -p123456 //访问服务
mysql> use TESTDB; //进入 TESTDB 库
mysql> create table employee( ID int primary key , sharding_id int,
-> name char(15) , age int ); //建表
Query OK, 0 rows affected (0.68 sec)
```

```
mysql> desc employee; //查看表结构
```

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	
sharding_id	int(11)	YES		NULL	
name	char(15)	YES		NULL	
age	int(11)	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
Mysql>insert into employee(ID,sharding_id,name,age) //插入表记录
values
```

```
(1,10000,"bob",19), //存储在 53 服务器的 db1 库的 employee 表里
```

```
(2,10010,"tom",21), //存储在 54 服务器的 db2 库的 employee 表里
```

```
(3,10020,"lucy2",16); //存储在 55 服务器的 db3 库的 employee 表里
```

```
Query OK, 3 rows affected (0.07 sec)
```

```
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> select * from employee; //查看表记录
```

ID	sharding_id	name	age
1	10000	bob	19
2	10010	tom	21
3	10020	lucy	16

```
3 rows in set (0.06 sec)
```

```
mysql>insert into employee(ID,sharding_id,name,age)
values
```

```
(4,10000,"bob2",19), //存储在 53 服务器的 db1 库的 employee 表里
```

```
(5,10000,"tom2",21), //存储在 53 服务器的 db1 库的 employee 表里
```

```
(6,10000,"lucy2",16); //存储在 53 服务器的 db1 库的 employee 表里
```

```
Query OK, 3 rows affected (0.07 sec)
```

```
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> select * from employee; //查看表记录
```

```
+-----+-----+-----+-----+
```

ID	sharding_id	name	age
1	10000	bob	19
4	10000	bob2	19
5	10000	tom2	21
6	10000	lucy2	16
3	10020	lucy	16
2	10010	tom	21

6 rows in set (0.00 sec)

6) 在数据库服务器本机，查看表记录

在数据库服务器 192.168.4.53 查看数据

```
[root@host53 ~]# mysql -uroot -p123qqq...A -e "select * from db1.employee"
mysql: [Warning] Using a password on the command line interface can be insecure.
+----+-----+-----+-----+
| ID | sharding_id | name | age |
+----+-----+-----+-----+
| 1  | 10000       | bob  | 19  |
| 4  | 10000       | bob2 | 19  |
| 5  | 10000       | tom2 | 21  |
| 6  | 10000       | lucy2| 16  |
+----+-----+-----+-----+
[root@host53 ~]#
```

在数据库服务器 192.168.4.54 查看数据

```
[root@host54 ~]# mysql -uroot -p123qqq...A -e "select * from db2.employee"
mysql: [Warning] Using a password on the command line interface can be insecure.
+----+-----+-----+-----+
| ID | sharding_id | name | age |
+----+-----+-----+-----+
| 2  | 10010       | tom  | 21  |
+----+-----+-----+-----+
[root@host54 ~]#
```

在数据库服务器 192.168.4.55 查看数据

```
[root@host55 ~]# mysql -uroot -p123qqq...A -e "select * from db3.employee"
mysql: [Warning] Using a password on the command line interface can be insecure.
+----+-----+-----+-----+
| ID | sharding_id | name | age |
+----+-----+-----+-----+
| 3  | 10020       | lucy | 16  |
+----+-----+-----+-----+
[root@host55 ~]#
```

步骤二：练习 mod-long 分片规则的使用

1) 查看配置文件，得知使用 mod-long 分片规则的表名

注意要删除 primaryKey="ID" autoIncrement="true" 不然无法存储数据

```
[root@mycat56 ~]# vim /usr/local/mycat/conf/schema.xml
<table name="hotnews" dataNode="dn1,dn2,dn3" rule="mod-long" />
:wq
```

2) 查看规则文件，得知 mod-long 分片规则使用的函数

```
[root@mycat56 ~]# vim /usr/local/mycat/conf/rule.xml
<tableRule name="mod-long">
  <rule>
```

```

        <columns>id</columns> //数据分片字段
        <algorithm>mod-long</algorithm> //函数名
    </rule>
</tableRule>
<function name="mod-long" class="io.mycat.route.function.PartitionByMod">
    <!-- how many data nodes -->
    <property name="count">3</property> //指定求模数字
</function>
: wq

```

3) 重启 mycat 服务，使其修改有效

```

[root@mycat56 ~]# /usr/local/mycat/bin/mycat stop //停止服务
Stopping Mycat-server...
Stopped Mycat-server.

[root@mycat56 conf]# netstat -utnlp | grep :8066 //无端口

[root@mycat56 conf]# ps -C java //无进程
[root@mycat56 conf]#

[root@mycat56 conf]# /usr/local/mycat/bin/mycat start //启动服务
Starting Mycat-server...
[root@mycat56 conf]#

[root@mycat56 conf]# netstat -utnlp | grep :8066 //有端口
tcp6      0      0 :::8066          :::*              LISTEN      1364/java
[root@mycat56 conf]#

[root@mycat56 conf]# ps -C java //有进程
PID TTY          TIME CMD
 1125 ?            00:00:01 java
[root@mycat56 conf]#

```

4) 客户端连接分片服务器，存取数据

```

]#mysql -h192.168.4.56 -P8066 -uroot -p123456 //访问服务
mysql> use TESTDB; //进入 TESTDB 库
mysql> create table hotnews(id int ,title char(30),comment char(200)); //建表
Query OK, 0 rows affected (0.79 sec)

mysql> desc hotnews; //查看表结构
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) | YES | | NULL | |
| title | char(30) | YES | | NULL | |
| comment | char(200) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> insert into hotnews(id,title,comment)values(9,"sc","xxxxx"); //插入第 1 条表
记录, 9 和 3 取余 余 0 记录存储在 53 服务器的 db1 库里
Query OK, 1 row affected (0.11 sec)

mysql> insert into hotnews(id,title,comment)values(10,"xx","haha");//插入第 2 条表
记录, 10 和 3 取余 余 1 记录存储在 54 服务器的 db2 库里
Query OK, 1 row affected (0.05 sec)

```

```
mysql> insert into hotnews(id,title,comment)values(11,"yy","zz");//插入第 3 条表记录, 11 和 3 取余 余 2 记录存储在 55 服务器的 db3 库里
Query OK, 1 row affected (0.03 sec)
```

```
mysql> select * from hotnews; //查看表记录
```

id	title	comment
11	yy	zz
10	xx	haha
9	sc	xxxxx

3 rows in set (0.01 sec)

5) 在数据库服务器本机, 查看表记录

在数据库服务器 192.168.4.53 查看数据

```
[root@host53 ~]# mysql -uroot -p123qqq...A -e "select * from db1.hotnews"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+-----+-----+
| id | title | comment |
+-----+-----+-----+
| 9 | sc | xxxxx |
+-----+-----+-----+
[root@host53 ~]#
```

在数据库服务器 192.168.4.54 查看数据

```
[root@host54 ~]# mysql -uroot -p123qqq...A -e "select * from db2.hotnews"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+-----+-----+
| id | title | comment |
+-----+-----+-----+
| 10 | xx | haha |
+-----+-----+-----+
[root@host54 ~]# [root@host54 ~]#
```

在数据库服务器 192.168.4.55 查看数据

```
[root@host55 ~]# mysql -uroot -p123qqq...A -e "select * from db3.hotnews"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+-----+-----+
| id | title | comment |
+-----+-----+-----+
| 11 | yy | zz |
+-----+-----+-----+
[root@host55 ~]#
```

3. 案例 3: 连接分片服务器存储数据

• 问题: 具体要求如下

- 逻辑库名 BBSDB
- 逻辑表名 company2 数据不分片, 把数据存储到 3 台数据库服务器上。
- 逻辑表名 employee2 使用枚举法分片规则把数据存储到 3 台数据库服务器

上。

• 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置 Mycat 服务器

1) 添加新库

```
[root@mycat56 ~]# vim /usr/local/mycat/conf/server.xml
<user name="root">
    .....
    <property name="schemas">TESTDB,BBSDb</property> //指定逻辑库名
</user>
:wq
```

2) 添加新表

```
[root@mycat56 ~]# vim /usr/local/mycat/conf/schema.xml
<mycat:schema xmlns:mycat="http://io.mycat/">
    <schema name="BBSDb" checkSQLSchema="false" sqlMaxLimit="100">
        <table name="company2" primaryKey="ID" type="global"
dataNode="dn1,dn2,dn3" /> //指定逻辑表名 company2
        <table name="employee2" primaryKey="ID" dataNode="dn1,dn2,dn3"
rule="sharding-by-intfile" /> //指定逻辑表名 employee2
    </schema>
    <schema name="TESTDB" checkSQLSchema="false" sqlMaxLimit="100">
        .....
    </mycat:schema >
:wq
```

3) 重启 mycat 服务

```
[root@mycat56 ~]# /usr/local/mycat/bin/mycat stop //停止服务
Stopping Mycat-server...
Stopped Mycat-server.

[root@mycat56 conf]# netstat -utnlp | grep :8066 //无端口
[root@mycat56 conf]# ps -C java //无进程
[root@mycat56 conf]#

[root@mycat56 conf]# /usr/local/mycat/bin/mycat start //启动服务
Starting Mycat-server...
[root@mycat56 conf]#
[root@mycat56 conf]# netstat -utnlp | grep :8066 //有端口
tcp6      0      0 :::8066          :::*              LISTEN      1364/java
[root@mycat56 conf]#
[root@mycat56 conf]# ps -C java //有进程
PID TTY          TIME CMD
1125 ?            00:00:01 java
[root@mycat56 conf]#
```

步骤二：测试配置

1) 连接 mycat 服务器、建表、插入记录

```
[root@host50 ~]# mysql -h192.168.4.56 -P8066 -uroot -p123456 //连接 mycat 服务器
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

Your MySQL connection id is 1
Server version: 5.6.29-mycat-1.6-RELEASE-20161028204710 MyCat Server (OpenCloudDB)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases; //显示已有的数据库

```
+-----+
| DATABASE |
+-----+
| BBSDB    | //逻辑库 BBSDB
| TESTDB   |
+-----+
2 rows in set (0.00 sec)
```

mysql> use BBSDB; //切换到 BBSDB 库

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed

mysql> show tables; //查看表

```
+-----+
| Tables in BBSDB |
+-----+
| company          | //逻辑表
| employee          |
+-----+
2 rows in set (0.00 sec)
```

mysql> create table company(ID int primary key,name char(50),addr char(50)); //建表

Query OK, 0 rows affected (1.01 sec)

mysql> desc company; //查看表结构

```
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID    | int(11)| NO   | PRI | NULL    |       |
| name  | char(50)| YES  |     | NULL    |       |
| addr  | char(50)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

mysql> insert into company(ID,name,addr)values(1,"tarena","beijing"); //插入记录
Query OK, 1 row affected (0.10 sec)

mysql> insert into company(ID,name,addr)values(2,"tmall","beijing");
Query OK, 1 row affected (0.15 sec)

mysql> insert into company(ID,name,addr)values(3,"sina","beijing");
Query OK, 1 row affected (0.13 sec)

mysql> select * from company; //查看表记录

```
+-----+-----+-----+
| ID | name  | addr  |
+-----+-----+-----+
| 1  | tarena | beijing |
```

```
| 2 | tmall | beijing |
| 3 | sina  | beijing |
+-----+
3 rows in set (0.04 sec)
```

2) 在数据库服务器本机, 查看表记录

在数据库服务器 53 本机查看

```
[root@host53 ~]# mysql -uroot -p123qqq...A -e "select * from db1.company2"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+
| ID | name | addr |
+-----+
| 1 | tarena | beijing |
| 2 | tmall  | beijing |
| 3 | sina   | beijing |
+-----+
[root@host53 ~]#
```

在数据库服务器 54 本机查看

```
[root@host54 ~]# mysql -uroot -p123qqq...A -e "select * from db2.company2"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+
| ID | name | addr |
+-----+
| 1 | tarena | beijing |
| 2 | tmall  | beijing |
| 3 | sina   | beijing |
+-----+
[root@host54 ~]#
```

在数据库服务器 55 本机查看

```
[root@host55 ~]# mysql -uroot -p123qqq...A -e "select * from db3.company"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+
| ID | name | addr |
+-----+
| 1 | tarena | beijing |
| 2 | tmall  | beijing |
| 3 | sina   | beijing |
+-----+
[root@host55 ~]#
```