

医嘱冷热发布方案-数据迁移v1_停机迁移

1.准备

1.1.库备份

备份现有数据库，找DBA~

1.2.Shardingsphere Proxy

下载

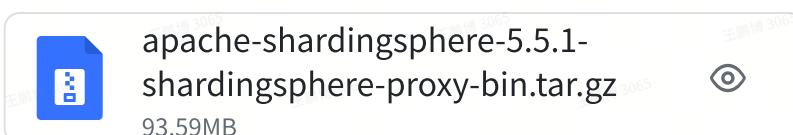
官方文档：<https://shardingsphere.apache.org/document/current/cn/quick-start/shardingsphere-proxy-quick-start/>

这里我们采用二进制包的方式进行部署，二进制包下载页面：

- <https://www.apache.org/dyn/closer.lua/shardingsphere/5.4.1/apache-shardingsphere-5.4.1-shardingsphere-proxy-bin.tar.gz>
- <https://www.apache.org/dyn/closer.lua/shardingsphere/5.5.1/apache-shardingsphere-5.5.1-shardingsphere-proxy-bin.tar.gz>

这里我已经准备好了，点击下载即可。

[apache-shardingsphere-5.4.1-shardingsphere-proxy-bin.tar.gz](#)



部署

一般推荐部署离MySQL服务比较近的机器上，比如就在MySQL所在服务器部署，但是可能MySQL所在服务器资源比较紧张可以部署在其它机器上，根据实际情况决定。

前提：必须安装JDK1.8环境！一般MySQL所在服务器是没有该环境的！

- 要么安装JDK1.8环境
- 要么在业务服务所在机器（业务服务已经安装了JDK1.8环境）直接部署

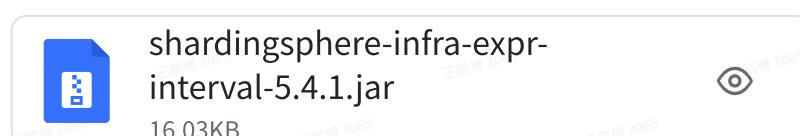
```
1 $ mkdir -p /usr/local/sharding-proxy
2 $ 将apache-shardingsphere-5.5.1-shardingsphere-proxy-bin.tar.gz上传
至/usr/local/sharding-proxy目录下
3 $ cd /usr/local/sharding-proxy
4 $ tar -zxvf apache-shardingsphere-5.5.1-shardingsphere-proxy-bin.tar.gz
5 $ mv apache-shardingsphere-5.5.1-shardingsphere-proxy-bin shardingsphere-proxy
```

上传扩展依赖包

将如下包上传至/usr/local/sharding-proxy/shardingsphere-proxy/ext-lib目录下：



1. (根据我们使用的MySQL版本决定，**如果是5.x版本为必须，如果是8.x版本无需上传**)



2. (**5.4.1版本必须 / 5.5.1版本非必须**)

初始没有/usr/local/sharding-proxy/shardingsphere-proxy/ext-lib目录，创建即可：

```
1 $ mkdir -p /usr/local/sharding-proxy/shardingsphere-proxy/ext-lib
```

配置

备份原始配置文件：

```
1 $ cp server.yaml server.yaml.default
2 $ cp config-sharding.yaml config-sharding.yaml.default
```

注意：检查配置中的地址信息是否正确，需要与实际相符！

修改上述两个配置：server.yaml和config-sharding.yaml，其配置内容如下：

server.xml

```
1 mode:
2   type: Cluster
3   repository:
4     type: ZooKeeper
5     props:
6       namespace: governance_ds
7       # 需要修改成对应环境的ZK服务地址信息
8       server-lists: 10.2.3.66:2181
9       retryIntervalMilliseconds: 500
10      timeToLiveSeconds: 60
11      maxRetries: 3
12      operationTimeoutMilliseconds: 500
13 authority:
14   users:
15   - user: root@%
16     password: root
17   - user: sharding
18     password: sharding
19 privilege:
20   type: ALL_PERMITTED
21
22 props:
23   max-connections-size-per-query: 1
24   sql-show: true
```

注意：分片策略需要与EMR或者同步程序保持一致！

注意：分片策略需要与EMR或者同步程序保持一致！

注意：分片策略需要与EMR或者同步程序保持一致！

config-sharding.xml

```
1  databaseName: windranger_emr
2
3  dataSources:
4    ds_0:
5      url: jdbc:mysql://10.2.3.157:3306/windranger_emr?
6        characterEncoding=utf8&serverTimezone=GMT%2B8&allowMultiQueries=true
7      username: user
8      password: Lachesis-mh_1024
9      connectionTimeoutMilliseconds: 30000
10     idleTimeoutMilliseconds: 60000
11     maxLifetimeMilliseconds: 1800000
12     maxPoolSize: 50
13     minPoolSize: 20
14
15   #props:
16   #  default-data-source: ds_0
17   rules:
18     - !SINGLE
19       tables:
20         - ds_0.*
21       defaultDataSource: ds_0
22     - !SHARDING
23       tables:
24         pat_inhos_order_out:
25           actualDataNodes:
26             <INTERVAL>P=ds_0.pat_inhos_order_out_;SP=yyyyMM;DIA=1;DIU=Months;DL=202101;DU=
27             202412
28             tableStrategy:
29               standard:
30                 shardingColumn: out_date
31                 shardingAlgorithmName: out_date_interval
32             pat_inhos_order_group_out:
33               actualDataNodes:
34                 <INTERVAL>P=ds_0.pat_inhos_order_group_out_;SP=yyyyMM;DIA=1;DIU=Months;DL=2021
35                 01;DU=202412
36                 tableStrategy:
37                   standard:
38                     shardingColumn: out_date
```

```
34     shardAlgorithmName: out_date_interval
35     shardAlgorithms:
36         out_date_interval:
37             type: INTERVAL
38             props:
39                 datetime-pattern: 'yyyy-MM-dd HH:mm:ss'
40                 datetime-lower: '2021-01-01 00:00:00'
41                 datetime-upper: '2024-12-31 23:59:59'
42                 shardings-suffix-pattern: 'yyyyMM'
43                 datetime-interval-amount: 1
44                 datetime-interval-unit: MONTHS
```

启动

进行启动并查看启动日志：

```
1 $ sh bin/start.sh
2 we find java version: java8, full_version=1.8.0_121,
full_path=/usr/local/java/jdk1.8.0_121/bin/java
3 The classpath is /usr/local/sharding-proxy/shardingsphere-
proxy/conf:/usr/local/sharding-proxy/shardingsphere-
proxy/conf:::/usr/local/sharding-proxy/shardingsphere-
proxy/lib/*:/usr/local/sharding-proxy/shardingsphere-proxy/ext-lib/*
4 main class org.apache.shardingsphere.proxy.Bootstrap -1 /usr/local/sharding-
proxy/shardingsphere-proxy/conf 0.0.0.0 false
5 Starting the ShardingSphere-Proxy ... PID: 9035
6 Please check the STDOUT file: /usr/local/sharding-proxy/shardingsphere-
proxy/logs/stdout.log
```

```
1 $ tailf logs/stdout.log
2 [INFO ] 2024-02-19 14:08:38.071 [main] org.apache.zookeeper.ZooKeeper -
Initiating client connection, connectString=10.2.3.66:2181
sessionTimeout=60000 watcher=org.apache.curator.ConnectionState@1a75e76a
3 [INFO ] 2024-02-19 14:08:38.094 [main] org.apache.zookeeper.common.X509Util -
Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-
initiated TLS renegotiation
4 [INFO ] 2024-02-19 14:08:38.109 [main] o.apache.zookeeper.ClientCnxnSocket -
jute.maxbuffer value is 1048575 Bytes
```

```
5 [INFO ] 2024-02-19 14:08:38.136 [main] org.apache.zookeeper.ClientCnxn -  
zookeeper.request.timeout value is 0. feature enabled=false  
6 [INFO ] 2024-02-19 14:08:38.154 [main] o.a.c.f.imps.CuratorFrameworkImpl -  
Default schema  
7 [INFO ] 2024-02-19 14:08:38.155 [main-SendThread(10.2.3.66:2181)]  
org.apache.zookeeper.ClientCnxn - Opening socket connection to server  
10.2.3.66/10.2.3.66:2181.  
8 [INFO ] 2024-02-19 14:08:38.155 [main-SendThread(10.2.3.66:2181)]  
org.apache.zookeeper.ClientCnxn - SASL config status: Will not attempt to  
authenticate using SASL (unknown error)  
9 [INFO ] 2024-02-19 14:08:38.219 [main-SendThread(10.2.3.66:2181)]  
org.apache.zookeeper.ClientCnxn - Socket connection established, initiating  
session, client: /10.2.3.173:46122, server: 10.2.3.66/10.2.3.66:2181  
10 [INFO ] 2024-02-19 14:08:38.243 [main-SendThread(10.2.3.66:2181)]  
org.apache.zookeeper.ClientCnxn - Session establishment complete on server  
10.2.3.66/10.2.3.66:2181, session id = 0x10000324fe8017e, negotiated timeout  
= 40000  
11 [INFO ] 2024-02-19 14:08:38.267 [main-EventThread]  
o.a.c.f.state.ConnectionStateManager - State change: CONNECTED  
12 [INFO ] 2024-02-19 14:09:14.300 [main] o.apache.curator.utils.Compatibility -  
Using org.apache.zookeeper.server.quorum.MultipleAddresses  
13 [INFO ] 2024-02-19 14:09:14.584 [main] o.a.s.p.v.ShardingSphereProxyVersion -  
Database type is `MySQL`, version is `5.6.51-log`, database name is  
'windranger_emr'  
14 [INFO ] 2024-02-19 14:09:14.586 [main] o.a.s.p.frontend.ssl.ProxySSLContext -  
Proxy frontend SSL/TLS is not enabled.  
15 [INFO ] 2024-02-19 14:09:14.875 [main] o.a.s.p.frontend.ShardingSphereProxy -  
ShardingSphere-Proxy Cluster mode started successfully
```

至此可以执行下一个环节了，后续的重置为非必要环境，只有需要时才执行~

重置（非必要）

何为重置？就是还原出厂设置。因为我们使用到了ZK辅助数据迁移，所以ShardingSphere Proxy的元数据会存储在ZK中，因此在重置ShardingSphere Proxy时，需要严格按照以下步骤进行操作。

1. 停止ShardingSphere Proxy服务

```
1 $ cd /usr/local/sharding-proxy/shardingsphere-proxy  
2 $ sh bin/stop.sh
```

如果上述命令无法停止时，可强杀进程：

```
1 $ ps -ef | grep "shardingsphere-proxy"
2 $ kill -9 {pid}
```

2.清空ZK服务的元数据信息

注意：先确认配置的ZK服务所在机器，一般找到机器之后，ZK的路径为：/usr/local/zookeeper

```
1 $ cd /usr/local/zookeeper/
2 $ sh bin/zkCli.sh
3
4 [zk: localhost:2181(CONNECTED) 0] ls /
5 [dubbo, governance_ds, zookeeper]
6 [zk: localhost:2181(CONNECTED) 1] deleteall /governance_ds
7 [zk: localhost:2181(CONNECTED) 2] ls /
8 [dubbo, zookeeper]
```

3.启动ShardingSphere Proxy服务

```
1 $ cd /usr/local/sharding-proxy/shardingsphere-proxy
2 $ sh bin/start.sh
3 $ tailf logs/stdout.log
```

1.3.MySQL配置修改

开启binlog，基于Proxy迁移必备条件，后续可进行关闭~

开启binlog配置

修改/usr/my.cnf配置（存在对应配置项时覆盖，不存在新增，切记在[mysqld]配置项下）

```
1 [mysqld]
2 server_id = 1
3 log-bin=/var/lib/mysql/master-bin.log
4 max_binlog_size = 1G
5 binlog-format=ROW
6 expire_logs_days = 6
```

如果配置后重启MySQL失败可找DBA宝哥联系解决！切记后续迁移完成之后关闭binlog！

特别说明

问题：不开启的话，对于历史数据的迁移是否支持，支持的话，此处的binlog可以不用开启，待验证~

答案：验证完毕，必须开启binlog，否则在使用Proxy自带迁移功能时，报错如下：

```
| 0 required BINLOG FORMAT <= ROW , now is < STATEMENT | | | | | org.apache.shardingsphere.data.pipeline.core.exception.job.PrepareJobWithInvalidDataSourceException: Source data source req
at org.apache.shardingsphere.data.pipeline.mysql.check.dataSource.MySQLDataSourceChecker.lambda$checkVariables2$0(MySQLDataSourceChecker.java:95)
at org.apache.shardingsphere.infra.executor.ShardingSpherePreconditions.checkNotNull(ShardingSpherePreconditions.java:41)
at org.apache.shardingsphere.data.pipeline.mysql.check.dataSource.MySQLDataSourceChecker.checkVariable(MySQLDataSourceChecker.java:94)
at org.apache.shardingsphere.data.pipeline.core.prepare.PipelineJobPreparer.fillEnderecoCheckVariables(PipelineJobPreparer.java:124)
at org.apache.shardingsphere.data.pipeline.core.executor.PipelineJobPreparer.fillCheckVariables(PipelineJobPreparer.java:152)
at org.apache.shardingsphere.data.pipeline.scenario.migration.prepare.MigrationJobPreparer.prepare(MigrationJobPreparer.java:93)
at org.apache.shardingsphere.data.pipeline.scenario.migration.MigrationJob.doPrepare(MigrationJob.java:75)
at org.apache.shardingsphere.data.pipeline.core.job.AbstractPipelineJob.prepare(AbstractPipelineJob.java:91)
at org.apache.shardingsphere.elasticjob.simple.executor.SimpleJobExecutor.execute(SimpleJobExecutor.java:61)
at org.apache.shardingsphere.elasticjob.simple.executor.SimpleJobExecutor.process(SimpleJobExecutor.java:33)
at org.apache.shardingsphere.elasticjob.executor.ElasticJobExecutor.process(ElasticJobExecutor.java:173)
at org.apache.shardingsphere.elasticjob.executor.ElasticJobExecutor.execute(ElasticJobExecutor.java:170)
at org.apache.shardingsphere.elasticjob.executor.ElasticJobExecutor.execute(ElasticJobExecutor.java:124)
at org.apache.shardingsphere.elasticjob.executor.ElasticJobExecutor.execute(ElasticJobExecutor.java:100)
at org.apache.shardingsphere.elasticjob.lite.internal.schedule.LiteJob.execute(LiteJob.java:35)
at org.quartz.core.JobRunShell.run(JobRunShell.java:202)
at org.quartz.impl.SimpleThreadPool.workerThread.run(SimpleThreadPool.java:573)
```

1.4.数据校验工具准备

下载

- 官方文档：<https://docs.pingcap.com/zh/tidb/stable/sync-diff-inspector-overview>
- 下载页面：<https://docs.pingcap.com/zh/tidb/stable/download-ecosystem-tools>

因为该工具Linux二进制包已经超过了100MB，上传不了附件，可以在10.2.3.173下的去获取， 对应服务的路径：/usr/local/sharding-proxy。

```
total 1255484
-rw-r--r-- 1 root root 97737013 Jan 18 10:49 apache-shardsphere-5.4.1-shardsphere-proxy-bin.tar.gz
drwxr-xr-x 2 root root 6 Feb 28 14:57 dbsyncer
drwxr-xr-x 10 root root 164 Feb 19 13:42 shardsphere-proxy
drwxr-xr-x 4 root root 4096 Feb 28 14:22 tidb-community-toolkit
-rw-r--r-- 1 root root 1191886466 Feb 19 17:50 tidb-community-toolkit-v7.5.0-linux-amd64.tar.gz
[root@localhost shardsphere-proxy]# pwd
/usr/local/sharding-proxy
[root@localhost shardsphere-proxy]#
```

部署

```
1 $ mkdir -p /usr/local/sharding-proxy
2 $ 将tidb-community-toolkit-v7.5.0-linux-amd64.tar.gz上传至/usr/local/sharding-
proxy目录下
3 $ cd /usr/local/sharding-proxy
4 $ tar -zxvf tidb-community-toolkit-v7.5.0-linux-amd64.tar.gz
5 $ mv tidb-community-toolkit-v7.5.0-linux-amd64 tidb-community-toolkit
```

配置

进入tidb-community-toolkit目录下，创建如下两个配置文件： config_pat_order_group_out.toml 和config_pat_order_out.toml。

注意：检查配置中的地址信息是否正确，需要与实际相符！

config_pat_order_out.toml

```
1 ##### Global config #####
2 check-thread-count = 4          # 检查数据的线程数量
3 export-fix-sql = true          # 如果开启，若表数据存在不一致，则输出用于修复的
SQL语句
4 check-struct-only = false       # 只对比表结构而不对比数据
5
6 ##### Datasource config #####
7 [data-sources.mysql1]           # 上游MySQL数据库配置（源端）
```

```

8      host = "10.2.3.173"
9      port = 3306
10     user = "user"
11     password = "Lachesis-mh_1024"
12     route-rules = ["rule1"]
13 [data-sources.mysql2]          # 下游MySQL数据库配置（目标端）
14     host = "10.2.3.173"
15     port = 3306
16     user = "user"
17     password = "Lachesis-mh_1024"
18
19 ##### Routes #####
20 # 如果需要对比大量的不同库名或者表名的表的数据，或者用于校验上游多个分表与下游总表的数据，可以通过table-rule来设置映射关系
21 [routes.rule1]
22 schema-pattern = "windranger_emr"           # 匹配数据源的库名，支持通配符"*"和"?"
23 table-pattern = "pat_inhos_order_out_202*"  # 匹配数据源的表名，支持通配符"*"和"?"
24 target-schema = "windranger_emr"             # 目标库名
25 target-table = "pat_inhos_order_out_tmp"    # 目标表名
26
27 ##### Task config #####
28 [task]
29     output-dir = "./output_order_out_sharding"
30     source-instances = ["mysql1"]           # 上游数据库，内容是data-sources声明的唯一标识id，分库分表场景下支持多个上游数据库，如：["mysql10", "mysql20"]
31     target-instance = "mysql2"              # 下游数据库，内容是data-sources声明的唯一标识id
32     target-check-tables = ["windranger_emr.pat_inhos_order_out_tmp"]   # 需要比对的下游数据库的表，每个表需要包含数据库名和表名，两者由.隔开

```

config_pat_order_group_out.toml

```

1 ##### Global config #####
2 check-thread-count = 4           # 检查数据的线程数量
3 export-fix-sql = true           # 如果开启，若表数据存在不一致，则输出用于修复的SQL语句
4 check-struct-only = false        # 只对比表结构而不对比数据
5
6 ##### Datasource config #####
7 [data-sources.mysql1]           # 上游MySQL数据库配置（源端）
8     host = "10.2.3.173"
9     port = 3306

```

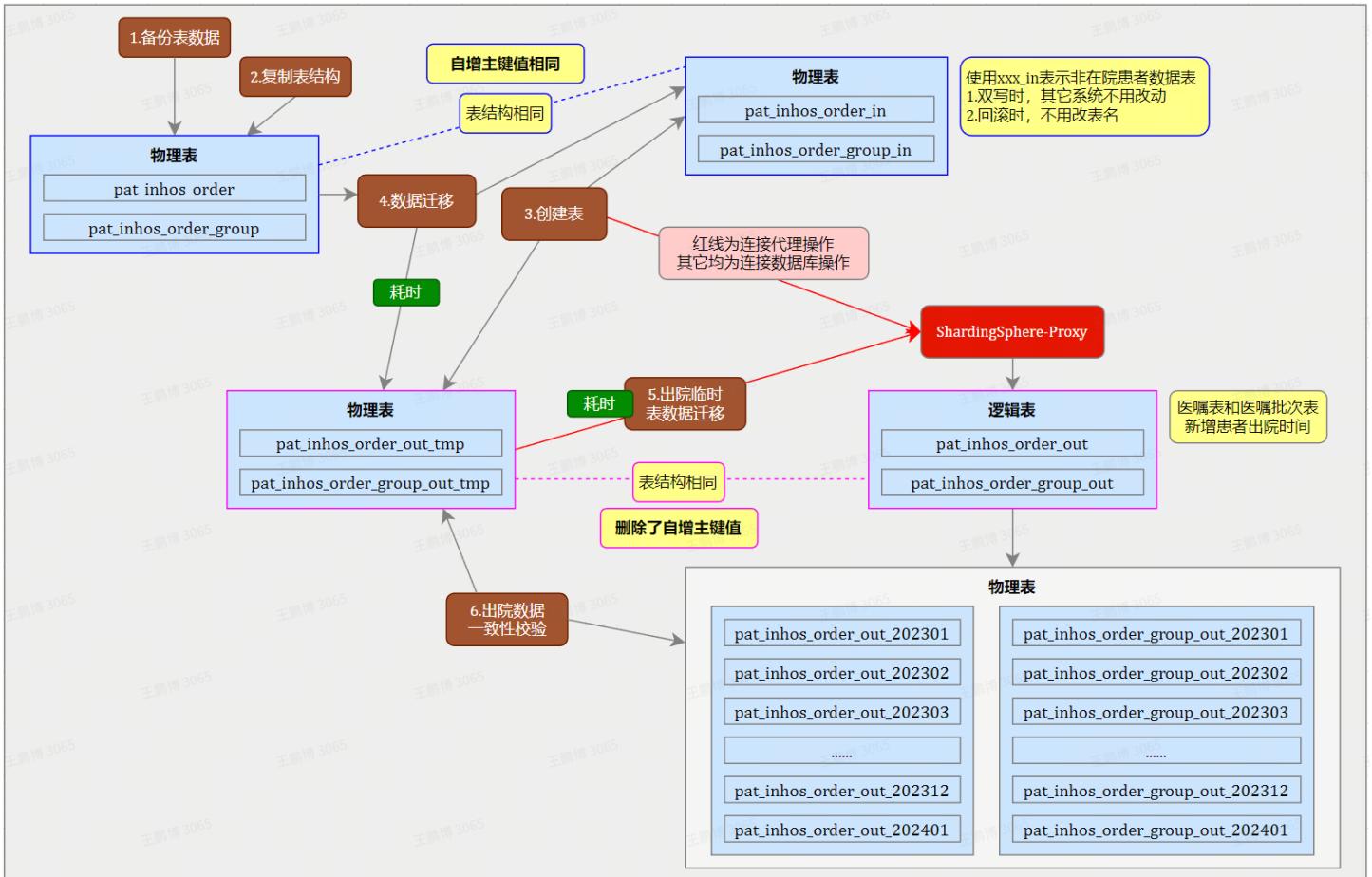
```

10      user = "user"
11      password = "Lachesis-mh_1024"
12      route-rules = ["rule1"]
13      [data-sources.mysql2]          # 下游MySQL数据库配置（目标端）
14          host = "10.2.3.173"
15          port = 3306
16          user = "user"
17          password = "Lachesis-mh_1024"
18
19 ##### Routes #####
20 # 如果需要对比大量的不同库名或者表名的表的数据，或者用于校验上游多个分表与下游总表的数据，可以通过table-rule来设置映射关系
21 [routes.rule1]
22 schema-pattern = "windranger_emr"           # 匹配数据源的库名，支持通配符"*"和"?""
23 table-pattern = "pat_inhos_order_group_out_202*"    # 匹配数据源的表名，支持通配符"*"和"?""
24 target-schema = "windranger_emr"            # 目标库名
25 target-table = "pat_inhos_order_group_out_tmp"    # 目标表名
26
27 ##### Task config #####
28 [task]
29     output-dir = "./output_order_group_out_sharding"
30     source-instances = ["mysql1"]      # 上游数据库，内容是data-sources声明的唯一标识id，分库分表场景下支持多个上游数据库，如: ["mysql10", "mysql20"]
31     target-instance = "mysql2"        # 下游数据库，内容是data-sources声明的唯一标识id
32     target-check-tables = ["windranger_emr.pat_inhos_order_group_out_tmp"]
# 需要比对的下游数据库的表，每个表需要包含数据库名和表名，两者由.隔开

```

2. 数据迁移

数据迁移整体流程



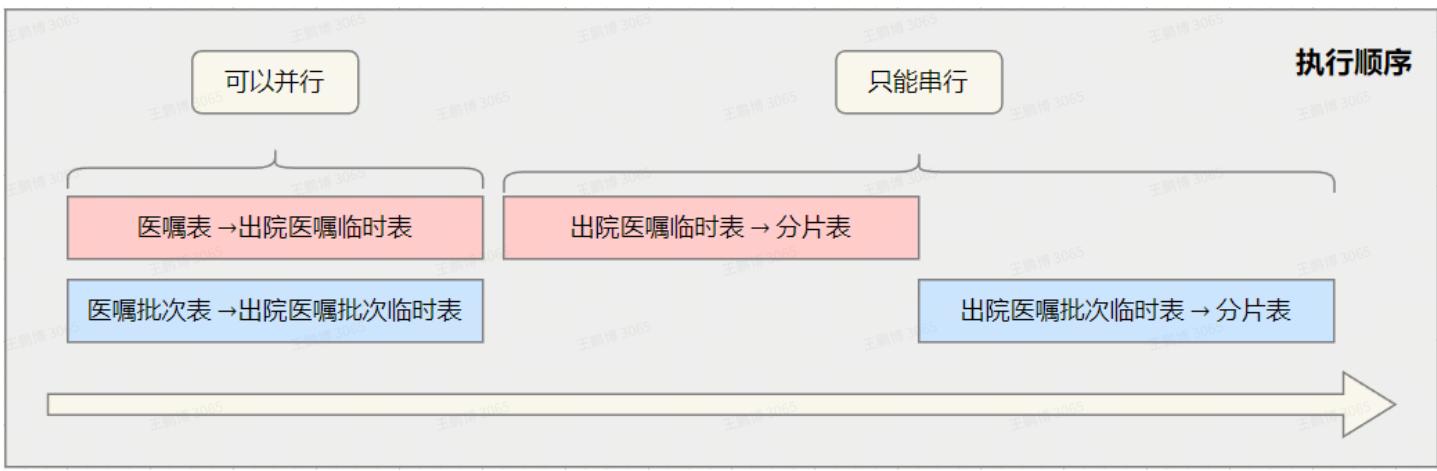
特别说明：目前还没有找到性能更优的迁移工具，所以暂时使用Proxy进行数据迁移工作~

出院临时表存在的意义

- 方便数据通过Proxy迁移至分片表，因为Proxy不支持SQL查询进行迁移
- 方便数据迁移完成后，进行数据校验

耗时操作顺序性问题

因为Proxy在调研阶段发现多个任务同时执行迁移时仅会有一个任务最终迁移完成，其它任务均会被终止，所以在此通过串行方式追个迁移。



2.0.一键停服

```
1 ./startToStopSpringBootAll stop
```

2.1.备份表数据

将原有表 (pat_inhos_order表和pat_inhos_order_group表) 先备份一份。

这里的备份方法比较简单，即将原有的表结构复制一份出来，重名下表名，比如 pat_inhos_order_bak202403/pat_inhos_order_group_bak202403，然后将原有表数据复制到新表中去：

```
1 insert into pat_inhos_order_bak202403 (seq_id, order_code, inhos_code,
pat_code, order_group_no, order_sub_no, repeat_indicator, order_class_code,
order_class, order_text, order_text_abbr, order_remark, item_code,
item_specification, item_price, total dosage, total dosage_units, dosage,
dosage_units, administration_code, administration, start_date_time,
stop_date_time, duration, duration_units, frequency_code, frequency,
freq_counter, freq_interval, freq_interval_unit, freq_detail,
is_self_prepare, is_emergent, is_continue, need_skintest, perform_result,
ordering_dept, doctor, stop_doctor, nurse, confirm_date_time, stop_nurse,
enter_date_time, stop_order_date_time, order_status, skintest_type,
is_dispense, order dosage, firstday_times, lastday_times, nurse_remark,
firstday_plan_time, endday_plan_time, dripping_speed, create_time,
create_person, update_time, update_person, ORDERING_DEPT_CODE, DOCTOR_CODE,
STOP_DOCTOR_CODE, NURSE_CODE, STOP_NURSE_CODE)
```

```
2 select pioa.seq_id, pioa.order_code, pioa.inhos_code, pioa.pat_code,
pioa.order_group_no, pioa.order_sub_no, pioa.repeat_indicator,
pioa.order_class_code, pioa.order_class, pioa.order_text,
pioa.order_text_abbr, pioa.order_remark, pioa.item_code,
pioa.item_specification, pioa.item_price, pioa.total_dosage,
pioa.total_dosage_units, pioa.dosage, pioa.dosage_units,
pioa.administration_code, pioa.administration, pioa.start_date_time,
pioa.stop_date_time, pioa.duration, pioa.duration_units, pioa.frequency_code,
pioa.frequency, pioa.freq_counter, pioa.freq_interval,
pioa.freq_interval_unit, pioa.freq_detail, pioa.is_self_prepare,
pioa.is_emergent, pioa.is_continue, pioa.need_skintest, pioa.perform_result,
pioa.ordering_dept, pioa.doctor, pioa.stop_doctor, pioa.nurse,
pioa.confirm_date_time, pioa.stop_nurse, pioa.enter_date_time,
pioa.stop_order_date_time, pioa.order_status, pioa.skintest_type,
pioa.is_dispense, pioa.order dosage, pioa.firstday_times, pioa.lastday_times,
pioa.nurse_remark, pioa.firstday_plan_time, pioa.endday_plan_time,
pioa.dripping_speed, pioa.create_time, pioa.create_person, pioa.update_time,
pioa.update_person, pioa.ORDERING_DEPT_CODE, pioa.DOCTOR_CODE,
pioa.STOP_DOCTOR_CODE, pioa.NURSE_CODE, pioa.STOP_NURSE_CODE
3 from pat_inhos_order pioa;
```

```
1 insert into pat_inhos_order_group_bak202403
2 (seq_id, group_unique_code, order_group_no, inhos_code, orderbar,
package_bar, plan_time, order_sort_no, source_type, isprint, execute_status,
execute_date, print_date, execute_person, print_person, apply_time,
is_dispensed, remark, reason, execute_type, start_execute_user,
start_execute_date, start_check_user, end_execute_user, end_execute_date,
create_time, create_person, update_time, update_person)
3 select piog.seq_id, piog.group_unique_code, piog.order_group_no,
piog.inhos_code, piog.orderbar, piog.package_bar, piog.plan_time,
piog.order_sort_no, piog.source_type, piog.isprint, piog.execute_status,
piog.execute_date, piog.print_date, piog.execute_person, piog.print_person,
piog.apply_time, piog.is_dispensed, piog.remark, piog.reason,
piog.execute_type, piog.start_execute_user, piog.start_execute_date,
piog.start_check_user, piog.end_execute_user, piog.end_execute_date,
piog.create_time, piog.create_person, piog.update_time, piog.update_person
4 from pat_inhos_order_group piog;
```

2.2. 复制表结构

医嘱

查看创建表结构：

```
1 show create table pat_inhos_order;
```

备份创建表结构：

```
1 CREATE TABLE `pat_inhos_order` (
  `seq_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT '自增长流水号',
  `order_code` varchar(50) NOT NULL COMMENT '医嘱号',
  `inhos_code` varchar(20) NOT NULL COMMENT '病人住院号',
  `pat_code` varchar(20) NOT NULL COMMENT '病人标识号',
  `order_group_no` varchar(50) DEFAULT '' COMMENT '医嘱组号',
  `order_sub_no` varchar(16) DEFAULT '' COMMENT '医嘱子序号',
  `repeat_indicator` int(16) DEFAULT NULL COMMENT '长期医嘱标志(1-长期, 0-临时)',
  `order_class_code` varchar(30) DEFAULT NULL COMMENT '医嘱类型编号',
  `order_class` varchar(16) DEFAULT '' COMMENT '医嘱类型名称',
  `order_text` varchar(128) DEFAULT '' COMMENT '医嘱正文',
  `order_text_abbr` varchar(30) DEFAULT '' COMMENT '医嘱简称',
  `order_remark` varchar(200) DEFAULT NULL COMMENT '医嘱批注',
  `item_code` varchar(50) DEFAULT '' COMMENT '项目编号,如药品编号',
  `item_specification` varchar(50) DEFAULT NULL COMMENT '项目规格',
  `item_price` decimal(10,4) DEFAULT NULL COMMENT '项目价格',
  `total dosage` decimal(11,5) DEFAULT '0.00000' COMMENT '药品使用总量',
  `total dosage_units` varchar(16) DEFAULT '' COMMENT '药品使用问题单位',
  `dosage` decimal(11,5) DEFAULT '0.00000' COMMENT '药品一次使用剂量',
  `dosage_units` varchar(16) DEFAULT '' COMMENT '剂量单位',
  `administration_code` varchar(30) DEFAULT NULL COMMENT '用法编号',
  `administration` varchar(16) DEFAULT '' COMMENT '给药途径和方法',
  `start_date_time` datetime DEFAULT NULL COMMENT '本医嘱起始日期及时间',
  `stop_date_time` datetime DEFAULT NULL COMMENT '本医嘱停止日期及时间',
  `duration` int(11) DEFAULT NULL COMMENT '一次执行的持续时间',
  `duration_units` varchar(16) DEFAULT '' COMMENT '持续时间单位',
  `frequency_code` varchar(30) DEFAULT NULL COMMENT '频率编号',
  `frequency` varchar(16) DEFAULT '' COMMENT '执行频率描述',
```

```
29     `freq_counter` int(11) DEFAULT NULL COMMENT '频率次数',
30     `freq_interval` int(11) DEFAULT NULL COMMENT '频率间隔',
31     `freq_interval_unit` varchar(16) DEFAULT '' COMMENT '频率间隔单位',
32     `freq_detail` varchar(16) DEFAULT '' COMMENT '执行时间详细描述',
33     `is_self_prepare` varchar(1) DEFAULT NULL COMMENT '是否为自备药',
34     `is_emergent` varchar(1) DEFAULT NULL COMMENT '是否紧急',
35     `is_continue` varchar(1) DEFAULT '' COMMENT '医嘱是否有延续性 0-没有 1-有延续
性',
36     `need_skintest` varchar(1) DEFAULT NULL COMMENT '是否需要皮试',
37     `perform_result` varchar(16) DEFAULT '' COMMENT '执行结果',
38     `ordering_dept` varchar(16) DEFAULT '' COMMENT '开医嘱科室',
39     `doctor` varchar(50) DEFAULT '' COMMENT '开医嘱医生',
40     `stop_doctor` varchar(50) DEFAULT '' COMMENT '停医嘱医生',
41     `nurse` varchar(50) DEFAULT '' COMMENT '开医嘱校对护士',
42     `confirm_date_time` datetime DEFAULT NULL COMMENT '确认时间',
43     `stop_nurse` varchar(50) DEFAULT '' COMMENT '停医嘱校对护士',
44     `enter_date_time` datetime DEFAULT NULL COMMENT '开医嘱录入日期及时间',
45     `stop_order_date_time` datetime DEFAULT NULL COMMENT '停医嘱录入日期及时间',
46     `order_status` varchar(16) DEFAULT '' COMMENT '医嘱状态(新开、校对、执行、停止
等)',
47     `skintest_type` varchar(200) DEFAULT '' COMMENT '皮试类型',
48     `is_dispense` int(11) DEFAULT NULL COMMENT '是否静配 0-非静配 1-静配',
49     `order dosage` decimal(11,5) DEFAULT NULL COMMENT '医嘱用量(his计算好的)',
50     `firstday_times` int(11) DEFAULT NULL COMMENT '首日执行次数',
51     `lastday_times` int(11) DEFAULT NULL COMMENT '末日拆分次数',
52     `nurse_remark` varchar(100) DEFAULT '' COMMENT '医嘱护士批注',
53     `firstday_plan_time` varchar(50) DEFAULT NULL COMMENT '首日计划执行时间',
54     `endday_plan_time` varchar(50) DEFAULT NULL COMMENT '末日计划执行时间',
55     `dripping_speed` varchar(200) DEFAULT NULL COMMENT '滴速',
56     `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
57     `create_person` varchar(50) NOT NULL COMMENT '创建人',
58     `update_time` datetime DEFAULT NULL COMMENT '修改时间',
59     `update_person` varchar(50) DEFAULT '' COMMENT '修改人',
60     `ORDERING_DEPT_CODE` int(11) DEFAULT NULL,
61     `DOCTOR_CODE` varchar(4) DEFAULT NULL,
62     `STOP_DOCTOR_CODE` varchar(4) DEFAULT NULL,
63     `NURSE_CODE` int(11) DEFAULT NULL,
64     `STOP_NURSE_CODE` int(11) DEFAULT NULL,
65 PRIMARY KEY (`seq_id`),
66 UNIQUE KEY `uq_pat_inhos_order_1` (`order_code`),
67 KEY `idx_pat_inhos_order_2` (`pat_code`),
68 KEY `idx_pat_inhos_order_4` (`ordering_dept`, `enter_date_time`),
69 KEY `idx_pat_inhos_order_5` (`start_date_time`),
70 KEY `idx_pat_inhos_order_6` (`stop_date_time`),
71 KEY `enter_date_time` (`enter_date_time`),
72 KEY `create_time_idx` (`create_time`) USING BTREE,
73 KEY `idx_pat_inhos_order_7` (`ordering_dept`, `inhos_code`),
```

```
王鹏博 3065
74     KEY `idx_pat_inhos_order_3`(`order_group_no`, `inhos_code`),
75     KEY `idx_pat_inhos_order_8`(`inhos_code`)
76 ) ENGINE=InnoDB AUTO_INCREMENT=73447892 DEFAULT CHARSET=utf8 COMMENT='病历模块-
住院医嘱表';
```

医嘱批次

查看创建表结构：

```
1   show create table pat_inhos_order_group;
```

备份创建表结构：

```
1   CREATE TABLE `pat_inhos_order_group` (
2     `seq_id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '自增长编号',
3     `group_unique_code` varchar(60) NOT NULL COMMENT '批次唯一编号',
4     `order_group_no` varchar(50) DEFAULT '' COMMENT '医嘱批次编号',
5     `inhos_code` varchar(20) DEFAULT '' COMMENT '病人住院号',
6     `orderbar` varchar(40) DEFAULT '' COMMENT '医嘱条码',
7     `package_bar` varchar(50) DEFAULT NULL COMMENT '药箱条码',
8     `plan_time` datetime DEFAULT NULL COMMENT '计划执行时间',
9     `order_sort_no` int(11) DEFAULT NULL COMMENT '医嘱排序编号',
10    `source_type` varchar(20) DEFAULT '' COMMENT '数据来源',
11    `isprint` int(11) NOT NULL DEFAULT '0' COMMENT '是否已打印 0 否; 1 是',
12    `execute_status` int(11) NOT NULL DEFAULT '0' COMMENT '执行状态 0 - 未执行,
1- 执行中, 2 - 已执行, 3-停止, 4-作废',
13    `execute_date` datetime DEFAULT NULL COMMENT '执行时间',
14    `print_date` datetime DEFAULT NULL COMMENT '打印时间',
15    `execute_person` varchar(20) DEFAULT '' COMMENT '执行人',
16    `print_person` varchar(20) DEFAULT '' COMMENT '打印人',
17    `apply_time` datetime DEFAULT NULL COMMENT '开立时间',
18    `is_dispensed` int(11) NOT NULL DEFAULT '0' COMMENT '是否已配药 1-是, 0 -
否',
19    `remark` varchar(100) DEFAULT '' COMMENT '医嘱备注',
20    `reason` varchar(100) DEFAULT NULL COMMENT '手动或作废执行的原因',
```

```
21   `execute_type` int(11) DEFAULT '0' COMMENT '执行方式 0-无效, 1- 扫描执行, 2  
22 - 手动执行',  
23   `start_execute_user` varchar(20) DEFAULT NULL COMMENT '手动补录开始执行人',  
24   `start_execute_date` datetime DEFAULT NULL COMMENT '手动补录开始执行时间',  
25   `start_check_user` varchar(20) DEFAULT NULL COMMENT '手动补录开始核对人',  
26   `end_execute_user` varchar(20) DEFAULT NULL COMMENT '手动补录结束执行人',  
27   `end_execute_date` datetime DEFAULT NULL COMMENT '手动补录结束执行时间',  
28   `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
29   `create_person` varchar(50) NOT NULL DEFAULT '' COMMENT '创建人',  
30   `update_time` datetime DEFAULT NULL COMMENT '修改时间',  
31   `update_person` varchar(50) DEFAULT NULL COMMENT '修改人',  
32 PRIMARY KEY (`seq_id`),  
33 UNIQUE KEY `uq_pat_inhos_order_group_1`(`group_unique_code`),  
34 KEY `idx_pat_inhos_order_group_1`(`order_group_no`),  
35 KEY `idx_pat_inhos_order_group_2`(`orderbar`),  
36 KEY `idx_pat_inhos_order_group_3`(`execute_date`),  
37 KEY `idx_pat_inhos_order_group_4`(`plan_time`),  
38 KEY `idx_idx_pat_inhos_order_group_5`(`inhos_code`),  
39 KEY `idx_pat_inhos_order_group_6`(`inhos_code`, `plan_time`, `execute_status`)  
40 ) ENGINE=InnoDB AUTO_INCREMENT=113029111 DEFAULT CHARSET=utf8 COMMENT='医嘱批次  
表';
```

2.3. 创建表

医嘱-出院临时表

注意：使用MySQL CLI直接连接MySQL进行表创建！！！

注意：删除自增属性！！！并新增出院时间字段！！！

注意：删除自增属性！！！并新增出院时间字段！！！

注意：删除自增属性！！！并新增出院时间字段！！！

删除自增属性如下：

```

CREATE TABLE `pat_inhos_order_bak202403` (
    `seq_id` bigint unsigned NOT NULL AUTO_INCREMENT COMMENT '自增长流水号',
    -- ...省略其它字段
    `STOP_DOCTOR_CODE` varchar(4) DEFAULT NULL,
    `NURSE_CODE` int DEFAULT NULL,
    `STOP_NURSE_CODE` int DEFAULT NULL,
    PRIMARY KEY (`seq_id`),
    UNIQUE KEY `uq_pat_inhos_order_1`(`order_code`),
    KEY `idx_pat_inhos_order_2`(`pat_code`),
    KEY `idx_pat_inhos_order_3`(`order_group_no`),
    KEY `idx_pat_inhos_order_4`(`ordering_dept`, `enter_date_time`),
    KEY `idx_pat_inhos_order_5`(`start_date_time`),
    KEY `idx_pat_inhos_order_6`(`stop_date_time`),
    KEY `idx_pat_inhos_order_1`(`inhos_code`, `ordering_dept`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=5137819956 DEFAULT CHARSET=utf8mb3 COMMENT='病历模块-住院医嘱表';

```

这两个部分删除掉

新增出院时间字段：

```

`ordering_dept_code` int(11) DEFAULT NULL,
`doctor_code` varchar(4) DEFAULT NULL,
`stop_doctor_code` varchar(4) DEFAULT NULL,
`nurse_code` int(11) DEFAULT NULL,
`stop_nurse_code` int(11) DEFAULT NULL,
`out_date` datetime DEFAULT NULL COMMENT '出院日期',
PRIMARY KEY (`seq_id`),
UNIQUE KEY `uq_pat_inhos_order_out_1`(`order_code`),
KEY `idx_pat_inhos_order_out_2`(`pat_code`),
KEY `idx_pat_inhos_order_out_3`(`ordering_dept`, `enter_date_time`),

```

最终创建语句：

```

1 CREATE TABLE `pat_inhos_order_out_tmp` (
2     `seq_id` bigint(20) unsigned NOT NULL COMMENT '自增长流水号',
3     `order_code` varchar(50) NOT NULL COMMENT '医嘱号',
4     `inhos_code` varchar(20) NOT NULL COMMENT '病人住院号',
5     `pat_code` varchar(20) NOT NULL COMMENT '病人标识号',
6     `order_group_no` varchar(50) DEFAULT '' COMMENT '医嘱组号',
7     `order_sub_no` varchar(16) DEFAULT '' COMMENT '医嘱子序号',
8     `repeat_indicator` int(16) DEFAULT NULL COMMENT '长期医嘱标志(1-长期, 0-临时)',
9     `order_class_code` varchar(30) DEFAULT NULL COMMENT '医嘱类型编号',
10    `order_class` varchar(16) DEFAULT '' COMMENT '医嘱类型名称',
11    `order_text` varchar(128) DEFAULT '' COMMENT '医嘱正文',
12    `order_text_abbr` varchar(30) DEFAULT '' COMMENT '医嘱简称',
13    `order_remark` varchar(200) DEFAULT NULL COMMENT '医嘱批注',
14    `item_code` varchar(50) DEFAULT '' COMMENT '项目编号,如药品编号',
15    `item_specification` varchar(50) DEFAULT NULL COMMENT '项目规格',

```

```
16   `item_price` decimal(10,4) DEFAULT NULL COMMENT '项目价格',
17   `total dosage` decimal(11,5) DEFAULT '0.00000' COMMENT '药品使用总量',
18   `total dosage_units` varchar(16) DEFAULT '' COMMENT '药品使用问题单位',
19   `dosage` decimal(11,5) DEFAULT '0.00000' COMMENT '药品一次使用剂量',
20   `dosage_units` varchar(16) DEFAULT '' COMMENT '剂量单位',
21   `administration_code` varchar(30) DEFAULT NULL COMMENT '用法编号',
22   `administration` varchar(16) DEFAULT '' COMMENT '给药途径和方法',
23   `start_date_time` datetime DEFAULT NULL COMMENT '本医嘱起始日期及时间',
24   `stop_date_time` datetime DEFAULT NULL COMMENT '本医嘱停止日期及时间',
25   `duration` int(11) DEFAULT NULL COMMENT '一次执行的持续时间',
26   `duration_units` varchar(16) DEFAULT '' COMMENT '持续时间单位',
27   `frequency_code` varchar(30) DEFAULT NULL COMMENT '频率编号',
28   `frequency` varchar(16) DEFAULT '' COMMENT '执行频率描述',
29   `freq_counter` int(11) DEFAULT NULL COMMENT '频率次数',
30   `freq_interval` int(11) DEFAULT NULL COMMENT '频率间隔',
31   `freq_interval_unit` varchar(16) DEFAULT '' COMMENT '频率间隔单位',
32   `freq_detail` varchar(16) DEFAULT '' COMMENT '执行时间详细描述',
33   `is_self_prepare` varchar(1) DEFAULT NULL COMMENT '是否为自备药',
34   `is_emergent` varchar(1) DEFAULT NULL COMMENT '是否紧急',
35   `is_continue` varchar(1) DEFAULT '' COMMENT '医嘱是否有延续性 0-没有 1-有延续
性',
36   `need_skintest` varchar(1) DEFAULT NULL COMMENT '是否需要皮试',
37   `perform_result` varchar(16) DEFAULT '' COMMENT '执行结果',
38   `ordering_dept` varchar(16) DEFAULT '' COMMENT '开医嘱科室',
39   `doctor` varchar(50) DEFAULT '' COMMENT '开医嘱医生',
40   `stop_doctor` varchar(50) DEFAULT '' COMMENT '停医嘱医生',
41   `nurse` varchar(50) DEFAULT '' COMMENT '开医嘱校对护士',
42   `confirm_date_time` datetime DEFAULT NULL COMMENT '确认时间',
43   `stop_nurse` varchar(50) DEFAULT '' COMMENT '停医嘱校对护士',
44   `enter_date_time` datetime DEFAULT NULL COMMENT '开医嘱录入日期及时间',
45   `stop_order_date_time` datetime DEFAULT NULL COMMENT '停医嘱录入日期及时间',
46   `order_status` varchar(16) DEFAULT '' COMMENT '医嘱状态(新开、校对、执行、停止
等)',
47   `skintest_type` varchar(200) DEFAULT '' COMMENT '皮试类型',
48   `is_dispense` int(11) DEFAULT NULL COMMENT '是否静配 0-非静配 1-静配',
49   `order dosage` decimal(11,5) DEFAULT NULL COMMENT '医嘱用量(his计算好的)',
50   `firstday_times` int(11) DEFAULT NULL COMMENT '首日执行次数',
51   `lastday_times` int(11) DEFAULT NULL COMMENT '末日拆分次数',
52   `nurse_remark` varchar(100) DEFAULT '' COMMENT '医嘱护士批注',
53   `firstday_plan_time` varchar(50) DEFAULT NULL COMMENT '首日计划执行时间',
54   `endday_plan_time` varchar(50) DEFAULT NULL COMMENT '末日计划执行时间',
55   `dripping_speed` varchar(200) DEFAULT NULL COMMENT '滴速',
56   `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
57   `create_person` varchar(50) NOT NULL COMMENT '创建人',
58   `update_time` datetime DEFAULT NULL COMMENT '修改时间',
59   `update_person` varchar(50) DEFAULT '' COMMENT '修改人',
60   `ordering_dept_code` int(11) DEFAULT NULL,
```

```
王鹏博 3065
61   `doctor_code` varchar(4) DEFAULT NULL,
62   `stop_doctor_code` varchar(4) DEFAULT NULL,
63   `nurse_code` int(11) DEFAULT NULL,
64   `stop_nurse_code` int(11) DEFAULT NULL,
65   `out_date` datetime DEFAULT NULL COMMENT '出院日期',
66   PRIMARY KEY (`seq_id`),
67   UNIQUE KEY `uq_pat_inhos_order_out_1`(`order_code`),
68   KEY `idx_pat_inhos_order_out_2`(`pat_code`),
69   KEY `idx_pat_inhos_order_out_3`(`ordering_dept`, `enter_date_time`),
70   KEY `idx_pat_inhos_order_out_4`(`start_date_time`),
71   KEY `idx_pat_inhos_order_out_5`(`stop_date_time`),
72   KEY `idx_pat_inhos_order_out_6`(`enter_date_time`),
73   KEY `idx_pat_inhos_order_out_7`(`create_time`) USING BTREE,
74   KEY `idx_pat_inhos_order_out_8`(`ordering_dept`, `inhos_code`),
75   KEY `idx_pat_inhos_order_out_9`(`order_group_no`, `inhos_code`),
76   KEY `idx_pat_inhos_order_out_10`(`inhos_code`),
77   KEY `idx_pat_inhos_order_out_11`(`out_date`)
78 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='病历模块-住院医嘱表(出院患者)'
```

医嘱-出院表

注意：使用MySQL CLI连接ShardingSphere Proxy代理进行表创建！！！

注意：与医嘱-出院临时表结构一样，只是名称不一样！！！

注意：与医嘱-出院临时表结构一样，只是名称不一样！！！

注意：与医嘱-出院临时表结构一样，只是名称不一样！！！

结构同上述出院临时表，创建表：

```
1 use windranger_emr;
2
3 CREATE TABLE `pat_inhos_order_out` (
4   `seq_id` bigint(20) unsigned NOT NULL COMMENT '自增长流水号',
5   `order_code` varchar(50) NOT NULL COMMENT '医嘱号',
6   `inhos_code` varchar(20) NOT NULL COMMENT '病人住院号',
7   `pat_code` varchar(20) NOT NULL COMMENT '病人标识号',
8   `order_group_no` varchar(50) DEFAULT '' COMMENT '医嘱组号',
9   `order_sub_no` varchar(16) DEFAULT '' COMMENT '医嘱子序号',
10  `repeat_indicator` int(16) DEFAULT NULL COMMENT '长期医嘱标志(1-长期, 0-临时)',
11  `order_class_code` varchar(30) DEFAULT NULL COMMENT '医嘱类型编号',
```

```
12 `order_class` varchar(16) DEFAULT '' COMMENT '医嘱类型名称',
13 `order_text` varchar(128) DEFAULT '' COMMENT '医嘱正文',
14 `order_text_abbr` varchar(30) DEFAULT '' COMMENT '医嘱简称',
15 `order_remark` varchar(200) DEFAULT NULL COMMENT '医嘱批注',
16 `item_code` varchar(50) DEFAULT '' COMMENT '项目编号,如药品编号',
17 `item_specification` varchar(50) DEFAULT NULL COMMENT '项目规格',
18 `item_price` decimal(10,4) DEFAULT NULL COMMENT '项目价格',
19 `total dosage` decimal(11,5) DEFAULT '0.00000' COMMENT '药品使用总量',
20 `total dosage_units` varchar(16) DEFAULT '' COMMENT '药品使用问题单位',
21 `dosage` decimal(11,5) DEFAULT '0.00000' COMMENT '药品一次使用剂量',
22 `dosage_units` varchar(16) DEFAULT '' COMMENT '剂量单位',
23 `administration_code` varchar(30) DEFAULT NULL COMMENT '用法编号',
24 `administration` varchar(16) DEFAULT '' COMMENT '给药途径和方法',
25 `start_date_time` datetime DEFAULT NULL COMMENT '本医嘱起始日期及时间',
26 `stop_date_time` datetime DEFAULT NULL COMMENT '本医嘱停止日期及时间',
27 `duration` int(11) DEFAULT NULL COMMENT '一次执行的持续时间',
28 `duration_units` varchar(16) DEFAULT '' COMMENT '持续时间单位',
29 `frequency_code` varchar(30) DEFAULT NULL COMMENT '频率编号',
30 `frequency` varchar(16) DEFAULT '' COMMENT '执行频率描述',
31 `freq_counter` int(11) DEFAULT NULL COMMENT '频率次数',
32 `freq_interval` int(11) DEFAULT NULL COMMENT '频率间隔',
33 `freq_interval_unit` varchar(16) DEFAULT '' COMMENT '频率间隔单位',
34 `freq_detail` varchar(16) DEFAULT '' COMMENT '执行时间详细描述',
35 `is_self_prepare` varchar(1) DEFAULT NULL COMMENT '是否为自备药',
36 `is_emergent` varchar(1) DEFAULT NULL COMMENT '是否紧急',
37 `is_continue` varchar(1) DEFAULT '' COMMENT '医嘱是否有延续性 0-没有 1-有延续性',
38 `need_skintest` varchar(1) DEFAULT NULL COMMENT '是否需要皮试',
39 `perform_result` varchar(16) DEFAULT '' COMMENT '执行结果',
40 `ordering_dept` varchar(16) DEFAULT '' COMMENT '开医嘱科室',
41 `doctor` varchar(50) DEFAULT '' COMMENT '开医嘱医生',
42 `stop_doctor` varchar(50) DEFAULT '' COMMENT '停医嘱医生',
43 `nurse` varchar(50) DEFAULT '' COMMENT '开医嘱校对护士',
44 `confirm_date_time` datetime DEFAULT NULL COMMENT '确认时间',
45 `stop_nurse` varchar(50) DEFAULT '' COMMENT '停医嘱校对护士',
46 `enter_date_time` datetime DEFAULT NULL COMMENT '开医嘱录入日期及时间',
47 `stop_order_date_time` datetime DEFAULT NULL COMMENT '停医嘱录入日期及时间',
48 `order_status` varchar(16) DEFAULT '' COMMENT '医嘱状态(新开、校对、执行、停止等)',
49 `skintest_type` varchar(200) DEFAULT '' COMMENT '皮试类型',
50 `is_dispense` int(11) DEFAULT NULL COMMENT '是否静配 0-非静配 1-静配',
51 `order dosage` decimal(11,5) DEFAULT NULL COMMENT '医嘱用量(his计算好的)',
52 `firstday_times` int(11) DEFAULT NULL COMMENT '首日执行次数',
53 `lastday_times` int(11) DEFAULT NULL COMMENT '末日拆分次数',
54 `nurse_remark` varchar(100) DEFAULT '' COMMENT '医嘱护士批注',
55 `firstday_plan_time` varchar(50) DEFAULT NULL COMMENT '首日计划执行时间',
56 `endday_plan_time` varchar(50) DEFAULT NULL COMMENT '末日计划执行时间',
```

```
57 `dripping_speed` varchar(200) DEFAULT NULL COMMENT '滴速',  
58 `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
59 `create_person` varchar(50) NOT NULL COMMENT '创建人',  
60 `update_time` datetime DEFAULT NULL COMMENT '修改时间',  
61 `update_person` varchar(50) DEFAULT '' COMMENT '修改人',  
62 `ordering_dept_code` int(11) DEFAULT NULL,  
63 `doctor_code` varchar(4) DEFAULT NULL,  
64 `stop_doctor_code` varchar(4) DEFAULT NULL,  
65 `nurse_code` int(11) DEFAULT NULL,  
66 `stop_nurse_code` int(11) DEFAULT NULL,  
67 `out_date` datetime NOT NULL comment '出院日期'  
68 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='病历模块-住院医嘱表(出院患者)';
```

医嘱-在院表

使用MySQL CLI直接连接MySQL进行表创建！！！

利用2.2.复制的表结构创建在院表。即原始的医嘱表用做在院表~

这里之所以称之为在院表，指的是除了出院患者医嘱数据外的所有数据均存于此表~

```
1 CREATE TABLE `pat_inhos_order_in` (  
2   `seq_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT '自增长流水号',  
3   `order_code` varchar(50) NOT NULL COMMENT '医嘱号',  
4   `inhos_code` varchar(20) NOT NULL COMMENT '病人住院号',  
5   `pat_code` varchar(20) NOT NULL COMMENT '病人标识号',  
6   `order_group_no` varchar(50) DEFAULT '' COMMENT '医嘱组号',  
7   `order_sub_no` varchar(16) DEFAULT '' COMMENT '医嘱子序号',  
8   `repeat_indicator` int(16) DEFAULT NULL COMMENT '长期医嘱标志(1-长期, 0-临时)',  
9   `order_class_code` varchar(30) DEFAULT NULL COMMENT '医嘱类型编号',  
10  `order_class` varchar(16) DEFAULT '' COMMENT '医嘱类型名称',  
11  `order_text` varchar(128) DEFAULT '' COMMENT '医嘱正文',  
12  `order_text_abbr` varchar(30) DEFAULT '' COMMENT '医嘱简称',  
13  `order_remark` varchar(200) DEFAULT NULL COMMENT '医嘱批注',  
14  `item_code` varchar(50) DEFAULT '' COMMENT '项目编号,如药品编号',  
15  `item_specification` varchar(50) DEFAULT NULL COMMENT '项目规格',  
16  `item_price` decimal(10,4) DEFAULT NULL COMMENT '项目价格',  
17  `total dosage` decimal(11,5) DEFAULT '0.00000' COMMENT '药品使用总量',  
18  `total dosage_units` varchar(16) DEFAULT '' COMMENT '药品使用问题单位',  
19  `dosage` decimal(11,5) DEFAULT '0.00000' COMMENT '药品一次使用剂量',  
20  `dosage_units` varchar(16) DEFAULT '' COMMENT '剂量单位',  
21  `administration_code` varchar(30) DEFAULT NULL COMMENT '用法编号',
```

```
22 `administration` varchar(16) DEFAULT '' COMMENT '给药途径和方法',
23 `start_date_time` datetime DEFAULT NULL COMMENT '本医嘱起始日期及时间',
24 `stop_date_time` datetime DEFAULT NULL COMMENT '本医嘱停止日期及时间',
25 `duration` int(11) DEFAULT NULL COMMENT '一次执行的持续时间',
26 `duration_units` varchar(16) DEFAULT '' COMMENT '持续时间单位',
27 `frequency_code` varchar(30) DEFAULT NULL COMMENT '频率编号',
28 `frequency` varchar(16) DEFAULT '' COMMENT '执行频率描述',
29 `freq_counter` int(11) DEFAULT NULL COMMENT '频率次数',
30 `freq_interval` int(11) DEFAULT NULL COMMENT '频率间隔',
31 `freq_interval_unit` varchar(16) DEFAULT '' COMMENT '频率间隔单位',
32 `freq_detail` varchar(16) DEFAULT '' COMMENT '执行时间详细描述',
33 `is_self_prepare` varchar(1) DEFAULT NULL COMMENT '是否为自备药',
34 `is_emergent` varchar(1) DEFAULT NULL COMMENT '是否紧急',
35 `is_continue` varchar(1) DEFAULT '' COMMENT '医嘱是否有延续性 0-没有 1-有延续
性',
36 `need_skintest` varchar(1) DEFAULT NULL COMMENT '是否需要皮试',
37 `perform_result` varchar(16) DEFAULT '' COMMENT '执行结果',
38 `ordering_dept` varchar(16) DEFAULT '' COMMENT '开医嘱科室',
39 `doctor` varchar(50) DEFAULT '' COMMENT '开医嘱医生',
40 `stop_doctor` varchar(50) DEFAULT '' COMMENT '停医嘱医生',
41 `nurse` varchar(50) DEFAULT '' COMMENT '开医嘱校对护士',
42 `confirm_date_time` datetime DEFAULT NULL COMMENT '确认时间',
43 `stop_nurse` varchar(50) DEFAULT '' COMMENT '停医嘱校对护士',
44 `enter_date_time` datetime DEFAULT NULL COMMENT '开医嘱录入日期及时间',
45 `stop_order_date_time` datetime DEFAULT NULL COMMENT '停医嘱录入日期及时间',
46 `order_status` varchar(16) DEFAULT '' COMMENT '医嘱状态(新开、校对、执行、停止
等)',
47 `skintest_type` varchar(200) DEFAULT '' COMMENT '皮试类型',
48 `is_dispense` int(11) DEFAULT NULL COMMENT '是否静配 0-非静配 1-静配',
49 `order dosage` decimal(11,5) DEFAULT NULL COMMENT '医嘱用量(his计算好的)',
50 `firstday_times` int(11) DEFAULT NULL COMMENT '首日执行次数',
51 `lastday_times` int(11) DEFAULT NULL COMMENT '末日拆分次数',
52 `nurse_remark` varchar(100) DEFAULT '' COMMENT '医嘱护士批注',
53 `firstday_plan_time` varchar(50) DEFAULT NULL COMMENT '首日计划执行时间',
54 `endday_plan_time` varchar(50) DEFAULT NULL COMMENT '末日计划执行时间',
55 `dripping_speed` varchar(200) DEFAULT NULL COMMENT '滴速',
56 `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
57 `create_person` varchar(50) NOT NULL COMMENT '创建人',
58 `update_time` datetime DEFAULT NULL COMMENT '修改时间',
59 `update_person` varchar(50) DEFAULT '' COMMENT '修改人',
60 `ORDERING_DEPT_CODE` int(11) DEFAULT NULL,
61 `DOCTOR_CODE` varchar(4) DEFAULT NULL,
62 `STOP_DOCTOR_CODE` varchar(4) DEFAULT NULL,
63 `NURSE_CODE` int(11) DEFAULT NULL,
64 `STOP_NURSE_CODE` int(11) DEFAULT NULL,
65 PRIMARY KEY (`seq_id`),
66 UNIQUE KEY `uq_pat_inhos_order_1` (`order_code`),
```

```
王鹏博 3065
67     KEY `idx_pat_inhos_order_2`(`pat_code`),
68     KEY `idx_pat_inhos_order_4`(`ordering_dept`, `enter_date_time`),
69     KEY `idx_pat_inhos_order_5`(`start_date_time`),
70     KEY `idx_pat_inhos_order_6`(`stop_date_time`),
71     KEY `enter_date_time`(`enter_date_time`),
72     KEY `create_time_idx`(`create_time`) USING BTREE,
73     KEY `idx_pat_inhos_order_7`(`ordering_dept`, `inhos_code`),
74     KEY `idx_pat_inhos_order_3`(`order_group_no`, `inhos_code`),
75     KEY `idx_pat_inhos_order_8`(`inhos_code`)
76 ) ENGINE=InnoDB AUTO_INCREMENT=73447892 DEFAULT CHARSET=utf8 COMMENT='病历模块-住院医嘱表';
```

医嘱批次-出院临时表

注意：使用MySQL CLI直接连接MySQL进行表创建！！！

注意：删除自增属性！！！并新增出院时间字段！！！

注意：删除自增属性！！！并新增出院时间字段！！！

注意：删除自增属性！！！并新增出院时间字段！！！

```
1 use windranger_emr;
2
3 CREATE TABLE `pat_inhos_order_group_out_tmp` (
4     `seq_id` bigint(20) unsigned NOT NULL COMMENT '自增长流水号',
5     `group_unique_code` varchar(60) NOT NULL COMMENT '批次唯一编号',
6     `order_group_no` varchar(50) DEFAULT '' COMMENT '医嘱批次编号',
7     `inhos_code` varchar(20) DEFAULT '' COMMENT '病人住院号',
8     `orderbar` varchar(40) DEFAULT '' COMMENT '医嘱条码',
9     `package_bar` varchar(50) DEFAULT NULL COMMENT '药箱条码',
10    `plan_time` datetime DEFAULT NULL COMMENT '计划执行时间',
11    `order_sort_no` int(11) DEFAULT NULL COMMENT '医嘱排序编号',
12    `source_type` varchar(20) DEFAULT '' COMMENT '数据来源',
13    `isprint` int(11) NOT NULL DEFAULT '0' COMMENT '是否已打印 0 否; 1 是',
14    `execute_status` int(11) NOT NULL DEFAULT '0' COMMENT '执行状态 0 - 未执行,
15    1- 执行中, 2 - 已执行, 3-停止, 4-作废',
16    `execute_date` datetime DEFAULT NULL COMMENT '执行时间',
17    `print_date` datetime DEFAULT NULL COMMENT '打印时间',
18    `execute_person` varchar(20) DEFAULT '' COMMENT '执行人',
19    `print_person` varchar(20) DEFAULT '' COMMENT '打印人',
20    `apply_time` datetime DEFAULT NULL COMMENT '开立时间',
```

```
20      `is_dispensed` int(11) NOT NULL DEFAULT '0' COMMENT '是否已配药 1-是， 0 -  
否',  
21      `remark` varchar(100) DEFAULT '' COMMENT '医嘱备注',  
22      `reason` varchar(100) DEFAULT NULL COMMENT '手动或作废执行的原因',  
23      `execute_type` int(11) DEFAULT '0' COMMENT '执行方式 0-无效, 1- 扫描执行, 2  
- 手动执行',  
24      `start_execute_user` varchar(20) DEFAULT NULL COMMENT '手动补录开始执行人',  
25      `start_execute_date` datetime DEFAULT NULL COMMENT '手动补录开始执行时间',  
26      `start_check_user` varchar(20) DEFAULT NULL COMMENT '手动补录开始核对人',  
27      `end_execute_user` varchar(20) DEFAULT NULL COMMENT '手动补录结束执行人',  
28      `end_execute_date` datetime DEFAULT NULL COMMENT '手动补录结束执行时间',  
29      `out_date` datetime NOT NULL COMMENT '出院日期',  
30      `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
31      `create_person` varchar(50) NOT NULL DEFAULT '' COMMENT '创建人',  
32      `update_time` datetime DEFAULT NULL COMMENT '修改时间',  
33      `update_person` varchar(50) DEFAULT NULL COMMENT '修改人',  
34      PRIMARY KEY (`seq_id`),  
35      UNIQUE KEY `uq_order_group_out_1` (`group_unique_code`),  
36      KEY `idx_order_group_out_1` (`order_group_no`),  
37      KEY `idx_order_group_out_2` (`orderbar`),  
38      KEY `idx_order_group_out_3` (`execute_date`),  
39      KEY `idx_order_group_out_4` (`plan_time`),  
40      KEY `idx_order_group_out_5` (`inhos_code`),  
41      KEY `idx_order_group_out_6` (`inhos_code`, `plan_time`, `execute_status`)  
42 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='医嘱批次表(出院患者)';
```

医嘱批次-出院表

注意：使用MySQL CLI连接ShardingSphere Proxy代理进行表创建！！！

注意：与医嘱批次-出院临时表结构一样，只是名称不一样！！！

注意：与医嘱批次-出院临时表结构一样，只是名称不一样！！！

注意：与医嘱批次-出院临时表结构一样，只是名称不一样！！！

查询原始表结构：

```
1 show create table pat_inhos_order_group;
```

创建表：

```
1  use windranger_emr;
2
3  CREATE TABLE `pat_inhos_order_group_out` (
4      `seq_id` bigint(20) unsigned NOT NULL COMMENT '自增长流水号',
5      `group_unique_code` varchar(60) NOT NULL COMMENT '批次唯一编号',
6      `order_group_no` varchar(50) DEFAULT '' COMMENT '医嘱批次编号',
7      `inhos_code` varchar(20) DEFAULT '' COMMENT '病人住院号',
8      `orderbar` varchar(40) DEFAULT '' COMMENT '医嘱条码',
9      `package_bar` varchar(50) DEFAULT NULL COMMENT '药箱条码',
10     `plan_time` datetime DEFAULT NULL COMMENT '计划执行时间',
11     `order_sort_no` int(11) DEFAULT NULL COMMENT '医嘱排序编号',
12     `source_type` varchar(20) DEFAULT '' COMMENT '数据来源',
13     `isprint` int(11) NOT NULL DEFAULT '0' COMMENT '是否已打印 0 否; 1 是',
14     `execute_status` int(11) NOT NULL DEFAULT '0' COMMENT '执行状态 0 - 未执行,
15     1- 执行中, 2 - 已执行, 3-停止, 4-作废',
16     `execute_date` datetime DEFAULT NULL COMMENT '执行时间',
17     `print_date` datetime DEFAULT NULL COMMENT '打印时间',
18     `execute_person` varchar(20) DEFAULT '' COMMENT '执行人',
19     `print_person` varchar(20) DEFAULT '' COMMENT '打印人',
20     `apply_time` datetime DEFAULT NULL COMMENT '开立时间',
21     `is_dispensed` int(11) NOT NULL DEFAULT '0' COMMENT '是否已配药 1-是, 0 -
否',
22     `remark` varchar(100) DEFAULT '' COMMENT '医嘱备注',
23     `reason` varchar(100) DEFAULT NULL COMMENT '手动或作废执行的原因',
24     `execute_type` int(11) DEFAULT '0' COMMENT '执行方式 0-无效, 1- 扫描执行, 2
- 手动执行',
25     `start_execute_user` varchar(20) DEFAULT NULL COMMENT '手动补录开始执行人',
26     `start_execute_date` datetime DEFAULT NULL COMMENT '手动补录开始执行时间',
27     `start_check_user` varchar(20) DEFAULT NULL COMMENT '手动补录开始核对人',
28     `end_execute_user` varchar(20) DEFAULT NULL COMMENT '手动补录结束执行人',
29     `end_execute_date` datetime DEFAULT NULL COMMENT '手动补录结束执行时间',
30     `out_date` datetime NOT NULL COMMENT '出院日期',
31     `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
32     `create_person` varchar(50) NOT NULL DEFAULT '' COMMENT '创建人',
33     `update_time` datetime DEFAULT NULL COMMENT '修改时间',
34     `update_person` varchar(50) DEFAULT NULL COMMENT '修改人',
35     PRIMARY KEY (`seq_id`),
36     UNIQUE KEY `uq_order_group_out_1` (`group_unique_code`),
37     KEY `idx_order_group_out_1` (`order_group_no`),
38     KEY `idx_order_group_out_2` (`orderbar`),
39     KEY `idx_order_group_out_3` (`execute_date`),
40     KEY `idx_order_group_out_4` (`plan_time`),
41     KEY `idx_order_group_out_5` (`inhos_code`),
42     KEY `idx_order_group_out_6` (`inhos_code`, `plan_time`, `execute_status`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='医嘱批次表(出院患者);
```

医嘱批次-在院表

使用MySQL CLI直接连接MySQL进行表创建！！！

利用2.2.复制的表结构创建在院表。即原始的医嘱表用做在院表~

这里之所以称之为在院表，顾名思义，就是除了出院患者医嘱数据外的所有数据均存于此表~

```
1  CREATE TABLE `pat_inhos_order_group_in` (
2      `seq_id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '自增长编号',
3      `group_unique_code` varchar(60) NOT NULL COMMENT '批次唯一编号',
4      `order_group_no` varchar(50) DEFAULT '' COMMENT '医嘱批次编号',
5      `inhos_code` varchar(20) DEFAULT '' COMMENT '病人住院号',
6      `orderbar` varchar(40) DEFAULT '' COMMENT '医嘱条码',
7      `package_bar` varchar(50) DEFAULT NULL COMMENT '药箱条码',
8      `plan_time` datetime DEFAULT NULL COMMENT '计划执行时间',
9      `order_sort_no` int(11) DEFAULT NULL COMMENT '医嘱排序编号',
10     `source_type` varchar(20) DEFAULT '' COMMENT '数据来源',
11     `isprint` int(11) NOT NULL DEFAULT '0' COMMENT '是否已打印 0 否; 1 是',
12     `execute_status` int(11) NOT NULL DEFAULT '0' COMMENT '执行状态 0 - 未执行,
13     执行中, 2 - 已执行, 3-停止, 4-作废',
14     `execute_date` datetime DEFAULT NULL COMMENT '执行时间',
15     `print_date` datetime DEFAULT NULL COMMENT '打印时间',
16     `execute_person` varchar(20) DEFAULT '' COMMENT '执行人',
17     `print_person` varchar(20) DEFAULT '' COMMENT '打印人',
18     `apply_time` datetime DEFAULT NULL COMMENT '开立时间',
19     `is_dispensed` int(11) NOT NULL DEFAULT '0' COMMENT '是否已配药 1-是, 0 -
20     否',
21     `remark` varchar(100) DEFAULT '' COMMENT ' 医嘱备注',
22     `reason` varchar(100) DEFAULT NULL COMMENT '手动或作废执行的原因',
23     `execute_type` int(11) DEFAULT '0' COMMENT '执行方式 0-无效, 1- 扫描执行, 2
24     - 手动执行',
25     `start_execute_user` varchar(20) DEFAULT NULL COMMENT '手动补录开始执行人',
26     `start_execute_date` datetime DEFAULT NULL COMMENT '手动补录开始执行时间',
27     `start_check_user` varchar(20) DEFAULT NULL COMMENT '手动补录开始核对人',
28     `end_execute_user` varchar(20) DEFAULT NULL COMMENT '手动补录结束执行人',
29     `end_execute_date` datetime DEFAULT NULL COMMENT '手动补录结束执行时间',
30     `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
31     `create_person` varchar(50) NOT NULL DEFAULT '' COMMENT '创建人',
32     `update_time` datetime DEFAULT NULL COMMENT '修改时间',
33     `update_person` varchar(50) DEFAULT NULL COMMENT '修改人',
34     PRIMARY KEY (`seq_id`),
35     UNIQUE KEY `uq_pat_inhos_order_group_1` (`group_unique_code`),
```

```
33     KEY `idx_pat_inhos_order_group_1`(`order_group_no`),
34     KEY `idx_pat_inhos_order_group_2`(`orderbar`),
35     KEY `idx_pat_inhos_order_group_3`(`execute_date`),
36     KEY `idx_pat_inhos_order_group_4`(`plan_time`),
37     KEY `idx_idx_pat_inhos_order_group_5`(`inhos_code`),
38     KEY `idx_pat_inhos_order_group_6`(`inhos_code`, `plan_time`, `execute_status`)
39 ) ENGINE=InnoDB AUTO_INCREMENT=113029111 DEFAULT CHARSET=utf8 COMMENT='医嘱批次表';
```

2.4. 数据迁移

使用MySQL CLI直接连接MySQL进行表数据迁移！！！

医嘱-在院

1. 统计

```
1 select count(*)
2 from pat_inhos_order pioa
3 left join pat_inhos_record pir on pioa.inhos_code = pir.inhos_code
4 where pir.inhos_code is null or pir.status != 0 or pir.out_date is null;
```

总数：637455 耗时：3min

2. 迁移

注意：在院表没有out_date字段！

```
1 insert into pat_inhos_order_in (seq_id, order_code, inhos_code, pat_code,
order_group_no, order_sub_no, repeat_indicator, order_class_code,
order_class, order_text, order_text_abbr, order_remark, item_code,
item_specification, item_price, total dosage, total dosage_units, dosage,
```

```

dosage_units, administration_code, administration, start_date_time,
stop_date_time, duration, duration_units, frequency_code, frequency,
freq_counter, freq_interval, freq_interval_unit, freq_detail,
is_self_prepare, is_emergent, is_continue, need_skintest, perform_result,
ordering_dept, doctor, stop_doctor, nurse, confirm_date_time, stop_nurse,
enter_date_time, stop_order_date_time, order_status, skintest_type,
is_dispense, order dosage, firstday_times, lastday_times, nurse_remark,
firstday_plan_time, endday_plan_time, dripping_speed, create_time,
create_person, update_time, update_person, ORDERING_DEPT_CODE, DOCTOR_CODE,
STOP_DOCTOR_CODE, NURSE_CODE, STOP_NURSE_CODE)
2 select pioa.seq_id, pioa.order_code, pioa.inhos_code, pioa.pat_code,
pioa.order_group_no, pioa.order_sub_no, pioa.repeat_indicator,
pioa.order_class_code, pioa.order_class, pioa.order_text,
pioa.order_text_abbr, pioa.order_remark, pioa.item_code,
pioa.item_specification, pioa.item_price, pioa.total dosage,
pioa.total dosage_units, pioa.dosage, pioa.dosage_units,
pioa.administration_code, pioa.administration, pioa.start_date_time,
pioa.stop_date_time, pioa.duration, pioa.duration_units, pioa.frequency_code,
pioa.frequency, pioa.freq_counter, pioa.freq_interval,
pioa.freq_interval_unit, pioa.freq_detail, pioa.is_self_prepare,
pioa.is_emergent, pioa.is_continue, pioa.need_skintest, pioa.perform_result,
pioa.ordering_dept, pioa.doctor, pioa.stop_doctor, pioa.nurse,
pioa.confirm_date_time, pioa.stop_nurse, pioa.enter_date_time,
pioa.stop_order_date_time, pioa.order_status, pioa.skintest_type,
pioa.is_dispense, pioa.order dosage, pioa.firstday_times, pioa.lastday_times,
pioa.nurse_remark, pioa.firstday_plan_time, pioa.endday_plan_time,
pioa.dripping_speed, pioa.create_time, pioa.create_person, pioa.update_time,
pioa.update_person, pioa.ORDERING_DEPT_CODE, pioa.DOCTOR_CODE,
pioa.STOP_DOCTOR_CODE, pioa.NURSE_CODE, pioa.STOP_NURSE_CODE
3 from pat_inhos_order pioa
4 left join pat_inhos_record pir on pioa.inhos_code = pir.inhos_code
5 where pir.inhos_code is null or pir.status != 0 or pir.out_date is null;

```

总数：637455 耗时：5min

医嘱-出院

1.统计

```

1 select count(*)
2 from pat_inhos_order pioa
3 left join pat_inhos_record pir on pioa.inhos_code = pir.inhos_code

```

```
4 where pir.inhos_code is not null and pir.status = 0 and pir.out_date is not null;
```

总数：19653695 耗时：4min

2.迁移

注意：出院临时表有out_date字段！

```
1 insert into pat_inhos_order_out_tmp (seq_id, order_code, inhos_code,
pat_code, order_group_no, order_sub_no, repeat_indicator, order_class_code,
order_class, order_text, order_text_abbr, order_remark, item_code,
item_specification, item_price, total dosage, total dosage_units, dosage,
dosage_units, administration_code, administration, start_date_time,
stop_date_time, duration, duration_units, frequency_code, frequency,
freq_counter, freq_interval, freq_interval_unit, freq_detail,
is_self_prepare, is_emergent, is_continue, need_skintest, perform_result,
ordering_dept, doctor, stop_doctor, nurse, confirm_date_time, stop_nurse,
enter_date_time, stop_order_date_time, order_status, skintest_type,
is_dispense, order_dosage, firstday_times, lastday_times, nurse_remark,
firstday_plan_time, endday_plan_time, dripping_speed, out_date, create_time,
create_person, update_time, update_person, ORDERING_DEPT_CODE, DOCTOR_CODE,
STOP_DOCTOR_CODE, NURSE_CODE, STOP_NURSE_CODE)
2 select pioa.seq_id, pioa.order_code, pioa.inhos_code, pioa.pat_code,
pioa.order_group_no, pioa.order_sub_no, pioa.repeat_indicator,
pioa.order_class_code, pioa.order_class, pioa.order_text,
pioa.order_text_abbr, pioa.order_remark, pioa.item_code,
pioa.item_specification, pioa.item_price, pioa.total dosage,
pioa.total dosage_units, pioa.dosage, pioa.dosage_units,
pioa.administration_code, pioa.administration, pioa.start_date_time,
pioa.stop_date_time, pioa.duration, pioa.duration_units, pioa.frequency_code,
pioa.frequency, pioa.freq_counter, pioa.freq_interval,
pioa.freq_interval_unit, pioa.freq_detail, pioa.is_self_prepare,
pioa.is_emergent, pioa.is_continue, pioa.need_skintest, pioa.perform_result,
pioa.ordering_dept, pioa.doctor, pioa.stop_doctor, pioa.nurse,
pioa.confirm_date_time, pioa.stop_nurse, pioa.enter_date_time,
pioa.stop_order_date_time, pioa.order_status, pioa.skintest_type,
pioa.is_dispense, pioa.order_dosage, pioa.firstday_times, pioa.lastday_times,
pioa.nurse_remark, pioa.firstday_plan_time, pioa.endday_plan_time,
pioa.dripping_speed, pir.out_date, pioa.create_time, pioa.create_person,
```

```
王鹏博 3065 pioa.update_time, pioa.update_person, pioa.ORDERING_DEPT_CODE,
王鹏博 3065 pioa.DOCTOR_CODE, pioa.STOP_DOCTOR_CODE, pioa.NURSE_CODE, pioa.STOP_NURSE_CODE
3   from pat_inhos_order pioa
4   left join pat_inhos_record pir on pioa.inhos_code = pir.inhos_code
5   where pir.inhos_code is not null and pir.status = 0 and pir.out_date is not
null;
```

总数：19653695 耗时：

医嘱-验证

迁移完成之后，两个表的数据总量之和应该等于pat_inhos_order_all表的数据总量。

```
1 select count(*) from pat_inhos_order; # 20291150
2 select count(*) from pat_inhos_order_in; # 637455
3 select count(*) from pat_inhos_order_out_tmp; # 19653695
```

医嘱批次-在院

1.统计

```
1 select count(*)
2   from pat_inhos_order_group piog
3   left join pat_inhos_record pir on piog.inhos_code = pir.inhos_code
4   where pir.inhos_code is null or pir.status != 0 or pir.out_date is null;
```

总数：1972883 耗时：7min

2.迁移

注意：在院表没有out_date字段！

```
1 insert into pat_inhos_order_group_in
2     (seq_id, group_unique_code, order_group_no, inhos_code, orderbar,
package_bar, plan_time, order_sort_no, source_type, isprint, execute_status,
execute_date, print_date, execute_person, print_person, apply_time,
is_dispensed, remark, reason, execute_type, start_execute_user,
start_execute_date, start_check_user, end_execute_user, end_execute_date,
create_time, create_person, update_time, update_person)
3 select piog.seq_id, piog.group_unique_code, piog.order_group_no,
piog.inhos_code, piog.orderbar, piog.package_bar, piog.plan_time,
piog.order_sort_no, piog.source_type, piog.isprint, piog.execute_status,
piog.execute_date, piog.print_date, piog.execute_person, piog.print_person,
piog.apply_time, piog.is_dispensed, piog.remark, piog.reason,
piog.execute_type, piog.start_execute_user, piog.start_execute_date,
piog.start_check_user, piog.end_execute_user, piog.end_execute_date,
piog.create_time, piog.create_person, piog.update_time, piog.update_person
4 from pat_inhos_order_group piog
5 left join pat_inhos_record pir on piog.inhos_code = pir.inhos_code
6 where pir.inhos_code is null or pir.status != 0 or pir.out_date is null;
```

总数：1972883 耗时：8min

医嘱批次-出院

1.统计

```
1 select count(*)
2   from pat_inhos_order_group piog
3  left join pat_inhos_record pir on piog.inhos_code = pir.inhos_code
4 where pir.inhos_code is not null and pir.status = 0 and pir.out_date is not
null;
```

总数：33690258 耗时：5min

2.迁移

注意：出院临时表有out_date字段！

```
1 insert into pat_inhos_order_group_out_tmp
2     (seq_id, group_unique_code, order_group_no, inhos_code, orderbar,
package_bar, plan_time, order_sort_no, source_type, isprint, execute_status,
execute_date, print_date, execute_person, print_person, apply_time,
is_dispensed, remark, reason, execute_type, start_execute_user,
start_execute_date, start_check_user, end_execute_user, end_execute_date,
out_date, create_time, create_person, update_time, update_person)
3 select piog.seq_id, piog.group_unique_code, piog.order_group_no,
piog.inhos_code, piog.orderbar, piog.package_bar, piog.plan_time,
piog.order_sort_no, piog.source_type, piog.isprint, piog.execute_status,
piog.execute_date, piog.print_date, piog.execute_person, piog.print_person,
piog.apply_time, piog.is_dispensed, piog.remark, piog.reason,
piog.execute_type, piog.start_execute_user, piog.start_execute_date,
piog.start_check_user, piog.end_execute_user, piog.end_execute_date,
pir.out_date, piog.create_time, piog.create_person, piog.update_time,
piog.update_person
4 from pat_inhos_order_group piog
5 left join pat_inhos_record pir on piog.inhos_code = pir.inhos_code
6 where pir.inhos_code is not null and pir.status = 0 and pir.out_date is not
null;
```

总数：33690258 耗时：3h 12min

医嘱批次-验证

迁移完成之后，两个表的数据总量之和应该等于pat_inhos_order_group_all表的数据总量。

```
1 select count(*) from pat_inhos_order_group; # 36071028
2 select count(*) from pat_inhos_order_group_in; # 2380770
3 select count(*) from pat_inhos_order_group_out_tmp; # 33690258
```

2.5. 出院临时表 数据迁移

ShardingSphere Proxy必须是Cluster模式！！！

注意检查server.xml配置：

```
1 mode:
2 type: Cluster
```

```
3 repository:
4   type: ZooKeeper
5   props:
6     namespace: governance_ds
7     server-lists: 10.2.3.66:2181
8     retryIntervalMilliseconds: 500
9     timeToLiveSeconds: 60
10    maxRetries: 3
11    operationTimeoutMilliseconds: 500
```

注册数据来源存储单元：

```
1 REGISTER MIGRATION SOURCE STORAGE UNIT windranger_emr_source (
2   URL="jdbc:mysql://10.2.3.173:3306/windranger_emr?
characterEncoding=utf8&serverTimezone=GMT%2B8&allowMultiQueries=true",
3   USER="user",
4   PASSWORD="Lachesis-mh_1024",
5   PROPERTIES("minPoolSize"="1", "maxPoolSize"="20", "idleTimeout"="60000")
6 );
```

执行数据迁移：

```
1 MIGRATE TABLE windranger_emr_source.pat_inhos_order_out_tmp INTO
windranger_emr.pat_inhos_order_out;
2 MIGRATE TABLE windranger_emr_source.pat_inhos_order_group_out_tmp INTO
windranger_emr.pat_inhos_order_group_out;
```

因为ShardingSphere Proxy存在并发迁移问题，所以只能医嘱迁移完成之后，再迁移医嘱批次！！！

查看迁移进度：

```
1 SHOW MIGRATION LIST;
2 SHOW MIGRATION STATUS {JOB_ID};
```

```

mysql> SHOW MIGRATION LIST;
+-----+-----+-----+-----+-----+
| id   | tables          | job_item_count | active | create_time      | stop_time |
+-----+-----+-----+-----+-----+
| j0102p0000ca9d5cc364f25ee50b70c061a0913822 | windranger_emr_source.pat_inhos_order_group_out_tmp | 1       | true  | 2024-02-28 14:43:23 | NULL      |
+-----+-----+-----+-----+-----+
1 row in set (0.13 sec)

mysql> SHOW MIGRATION STATUS j0102p0000ca9d5cc364f25ee50b70c061a0913822;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| item | data_source    | tables          | status        | active | processed_records_count | inventory_finished_percentage | incremental_idle_seconds | error_message |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0    | windranger_emr_source | windranger_emr_source.pat_inhos_order_group_out_tmp | EXECUTE_INVENTORY_TASK | true   | 9000                  | 100                   | 0                     | null           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)

```

迁移完成判断标准：上述结果中processed_records_count等于xxx_out_tmp表的记录数时，说明数据已经迁移完成。

医嘱（出院患者）总数：耗时：

医嘱批次（出院患者）总数：耗时：

2.6. 数据校验

医嘱校验

因为前面已经将程序和配置文件准备好了，再此只需要执行即可。

在tidb-community-toolkit的根目录下执行如下命令：

```

1 $ ./sync_diff_inspector --config=./config_pat_order_out.toml
2 A total of 1 tables need to be compared
3
4 Progress [>-----] 0%
5 0/0
6 Comparing the table structure of ``windranger_emr``.`pat_inhos_order_out_tmp`` ...
... failure
7
8 Progress [=====>] 100%
9 0/0
10 The structure of `windranger_emr`.`pat_inhos_order_out_tmp` is not equal
11 The rest of tables are all equal.
12
13 A total of 1 tables have been compared, 0 tables finished, 1 tables failed, 0
tables skipped.
14 The patch file has been generated in

```

```
15           'output_order_out_sharding/fix-on-mysql2/'  
16 You can view the comparision details through  
'./output_order_out_sharding/sync_diff.log'
```

这里我们重点关注summary.txt文件，其说明见：

```
[root@localhost tidb-community-toolkit]# cd output_order_out_sharding/  
[root@localhost output_order_out_sharding]# ll  
total 156  
drwxr-xr-x 2 root root    78 Feb 28 13:34 checkpoint  
drwxr-xr-x 3 root root    46 Feb 28 13:29 fix-on-mysql2 ← 进入比对结果目录  
-rw-r--r-- 1 root root   925 Feb 28 13:34 summary.txt ← 如果数据不一致就会存在修复SQL  
-rw-r----- 1 root root 152807 Feb 28 13:34 sync_diff.log ← 当然这里数据完全一致的情况下，此文件目录为空  
[root@localhost output_order_out_sharding]# cat summary.txt 重点查看结果汇总  
Summary
```

Source Database

```
host = "10.2.3.173"  
port = 3306  
user = "user"
```

Target Databases

```
host = "10.2.3.173"  
port = 3306  
user = "user"
```

Comparison Result

succeed表示比较完成

The table structure and data in following tables are equivalent 属于正常显现
The following tables contains inconsistent data

TABLE	RESULT	STRUCTURE EQUALITY	DATA DIFF ROWS	UPCOUNT	DOWNCOUNT
`windranger_emr`.`pat_inhos_order_out_tmp`	succeed	false	+0/-0	19653695	19653695

Time Cost: 5m3.631300255s
Average Speed: 14.711922MB/s

医嘱批次

同上！

可能遇到的问题

如果执行报错如下：

```
[root@localhost tidb-community-toolkit]# ./sync_diff_inspector --config=./config_pat_order_out.toml
A total of 1 tables need to be compared

Progress [>-----] 0% 0/0
Comparing the table structure of ``windranger_emr``.`pat_inhos_order_out_tmp`` ... failure
Comparing the table data of ``windranger_emr``.`pat_inhos_order_out_tmp` ... failure

Progress [=====>-----] 16% 20/119
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33744->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33750->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33634->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33738->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33736->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33764->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33754->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33656->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33760->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33726->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33680->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33742->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33658->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33700->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33674->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33612->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33672->10.2.3.157:3306: read: connection reset by peer
[mysql] 2024/03/06 12:34:03 packets.go:37: read tcp 10.2.3.170:33756->10.2.3.157:3306: read: connection reset by peer
Comparing the table data of ``windranger_emr``.`pat_inhos_order_out_tmp` ... failure

Progress [=====>-----] 100% 0/0
Error in comparison process:
Error 1203 (42000): User user already has more than 'max_user_connections' active connections error occurred in `windranger_emr`.`pat_inhos_order_out_tmp`
You can view the comparison details through './output_order_out_sharding/sync_diff.log'
[root@localhost tidb-community-toolkit]#
```

解决方案：可适当调小对应配置中的线程数。

```
[root@localhost tidb-community-toolkit]# cat config_pat_order_out.toml
#####
# Global config #####
check-thread-count = 8          # 检查数据的线程数量
export-fix-sql = true           # 如果开启后，若表数据存在不一致，则输出用于修复的SQL语句
check-struct-only = false        # 只对比表结构而不对比数据

#####
# Datasource Config #####
[data-sources.mysql1]           # 上游MySQL数据库配置（源端）
    host = "10.2.3.157"
    port = 3306
    user = "user"
    password = "Lachesis-mh_1024"
    route-rules = ["rule1"]
[data-sources.mysql2]           # 下游MySQL数据库配置（目标端）
    host = "10.2.3.157"
    port = 3306
    user = "user"
    password = "Lachesis-mh_1024"

#####
# Routes #####
# 如果需要对比大量的不同库名或者表名的数据，或者用于校验上游多个分表与下游总表的数据，可以通过table-rule来设置映射关系
[routes.rule1]
schema-pattern = "windranger_emr"      # 匹配数据源的库名，支持通配符 "*" 和 "?"
table-pattern = "pat_inhos_order_out_202*" # 匹配数据源的表名，支持通配符 "*" 和 "?"
target-schema = "windranger_emr"         # 目标库名
target-table = "pat_inhos_order_out_tmp" # 目标表名

#####
# Task config #####
[task]
output-dir = "./output_order_out_sharding"
source-instances = ["mysql1"]           # 上游数据库，内容是data-sources声明的唯一标识id。分库分表场景下支持多个上游数据库，如：["mysql10", "mysql20"]
target-instance = "mysql2"               # 下游数据库，内容是data-sources声明的唯一标识id
target-check-tables = ["windranger_emr.pat_inhos_order_out_tmp"] # 需要比对的下游数据库的表，每个表需要包含数据库名和表名，两者由.隔开
```

3.善后

即历史数据迁移完成并校验完成，且数据一致之后，进行如下善后工作。

3.1.关闭binlog

注释掉上述修改/usr/my.cnf配置，之前的修改如下：

```
1 [mysqld]
2 server_id = 1
3 log-bin=/var/lib/mysql/master-bin.log
4 max_binlog_size = 1G
5 binlog-format=ROW
6 expire_logs_days = 6
```

注释之后重启MySQL！重启失败可找DBA宝哥联系解决！

4. 思考

- 既然出院表的数据均来自在院表，那么seq_id的自增属性是不是可以去掉？但是主键属性依然在，这样做好处有两点：
 - 保证seq_id的全局唯一性
 - 保证出院数据的来源唯一性（以免误操作数据未经在院表，直接插入出院表中）

4. 迁移速度慢的优化思路： MySQL千万级数据库数据插入insert速度加速调优

Original Confluence page attachments

Name	Size	Created by	Created on	Labels	Comment
 image-2024-3-15_16-47-... 65.31KB	65.31 KB	未知用户 (pengbo.wang)	2024-03-15T16:46:56.000+08:00		
	124.56 KB				

 image-2024-3-15_16-46-... 124.56KB	王鹏博 3065	未知用户 (pengbo.wang)	2024-03-15T16:46:18.000+08:00	王鹏博 3065	王鹏博 3065	王鹏博 3065
 image-2024-3-6_13-50-... 185.53KB	王鹏博 3065	185.53 KB	王鹏博	2024-03-06T13:45:17.000+08:00	王鹏博 3065	王鹏博 3065
 image-2024-3-6_13-50-4.png 791.90KB	王鹏博 3065	791.90 KB	王鹏博	2024-03-06T13:44:33.000+08:00	王鹏博 3065	王鹏博 3065
 image-2024-3-5_14-53-... 42.99KB	王鹏博 3065	42.99 KB	王鹏博	2024-03-05T14:48:25.000+08:00	王鹏博 3065	王鹏博 3065
 shardingsphere-infra-expr-... 16.03KB	王鹏博 3065	16.03 KB	王鹏博	2024-03-05T13:41:34.000+08:00	王鹏博 3065	王鹏博 3065
 mysql-connector-... 983.30KB	王鹏博 3065	983.30 KB	王鹏博	2024-03-05T13:41:31.000+08:00	王鹏博 3065	王鹏博 3065
 image-2024-3-5_13-46-8.png 16.03KB	王鹏博 3065	16.03 KB	王鹏博	2024-03-05T13:40:46.000+08:00	王鹏博 3065	王鹏博 3065
 image-2024-3-5_13-46-8-... 983.30KB	王鹏博 3065	983.30 KB	王鹏博	2024-03-05T13:40:46.000+08:00	王鹏博 3065	王鹏博 3065
 apache-shardingsph... 93.21MB	王鹏博 3065	93.21 MB	王鹏博	2024-03-05T13:28:00.000+08:00	王鹏博 3065	王鹏博 3065
 image-2024-2-28_14-50-... 137.10KB	王鹏博 3065	137.10 KB	王鹏博	2024-02-28T14:45:	王鹏博 3065	王鹏博 3065

				59.000+08: 00		
	 image-2024-2-28_14-46-... 81.61KB	81.61 KB	王鹏博	2024-02-28T14:41: 55.000+08: 00		
	 image-2024-2-28_14-27-... 186.78KB	186.78 KB	王鹏博	2024-02-28T14:22: 48.000+08: 00		
	 image-2024-2-28_13-53-... 182.35KB	182.35 KB	王鹏博	2024-02-28T13:48: 24.000+08: 00		
	 image-2024-2-20_14-23-... 128.71KB	128.71 KB	王鹏博	2024-02-20T14:19: 40.000+08: 00		
	 image-2024-2-20_14-22-... 128.81KB	128.81 KB	王鹏博	2024-02-20T14:18: 34.000+08: 00		
	 image-2024-2-20_14-19-... 124.04KB	124.04 KB	王鹏博	2024-02-20T14:16: 14.000+08: 00		
	 image-2024-2-20_14-19-... 22.55KB	22.55 KB	王鹏博	2024-02-20T14:15: 54.000+08: 00		
	 image-2024-2-20_14-7-... 124.59KB	124.59 KB	王鹏博	2024-02-20T14:03: 44.000+08: 00		
	 image-2024-2-20_13-57-... 117.96KB	117.96 KB	王鹏博	2024-02-20T13:53: 37.000+08: 00		

