# 概述

参考：

<https://mp.weixin.qq.com/s?__biz=MzIzOTA2NjEzNQ==&mid=2454780189&idx=1&sn=2013c2f62cbdffa7f274f6ccf60cbd09&chksm=fe8b9465c9fc1d7373226dfeb4bf7cb27cece721140819d30118af9b8dcddce12e6c365490a5&mpshare=1&scene=24&srcid=0315yBplX63k706DextFWkT7&sharer_sharetime=1615768697494&sharer_shareid=33f795d236f19ac7c128b2e279563f84#rd>

**什么是闪回？**

对已经写入数据库中的数据（已提交的事务），进行回滚，恢复到指定时间点之前的状态。

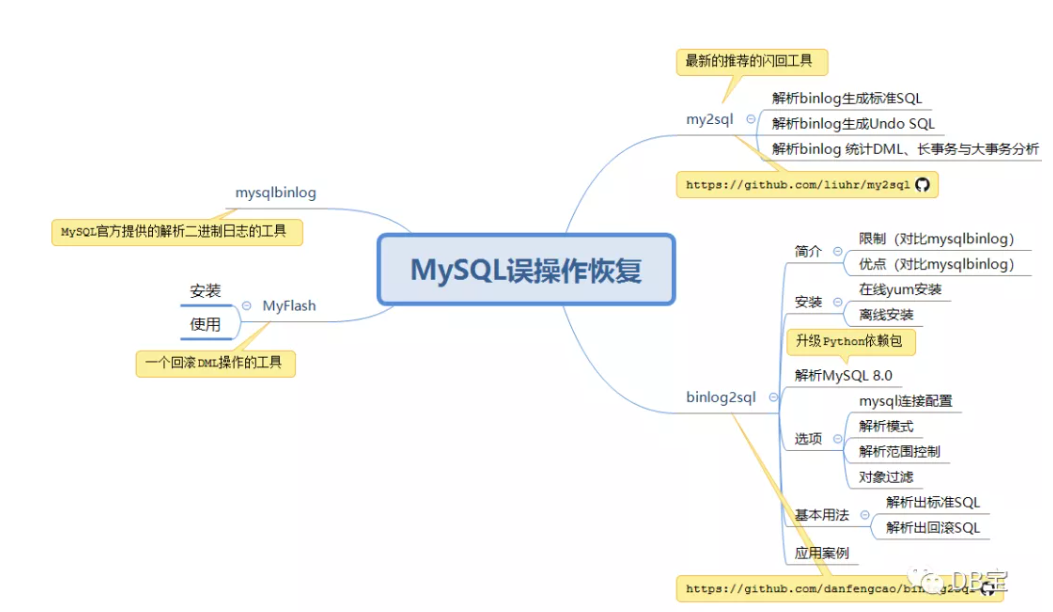
**闪回有什么作用？**

由于人为操作或其他原因导致数据错写、错删，需要将错误的数据恢复到错误之前的状态。

**闪回有什么条件？**

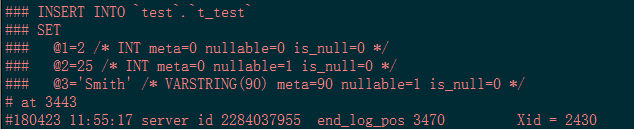
开启ROW格式binlog，闪回时间不宜过长。

可以用于MySQL误操作闪回的工具包括my2sql、binlog2sql和MyFlash等工具，其中，个人感觉my2sql最好用。

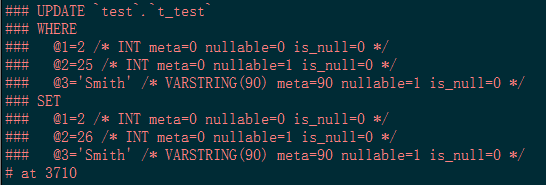


# 原理

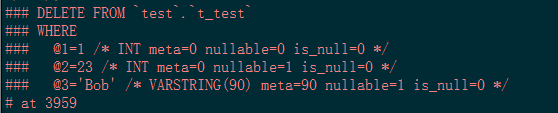
插入操作：



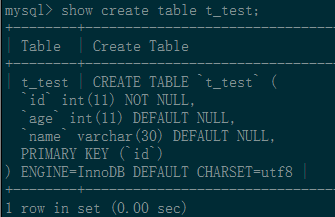
更新操作：



删除操作：



表结构：



ROW格式binlog：

不记录执行的sql语句的上下文相关的信息，仅需要记录那一条记录被修改成什么了。具备幂等性，可以重复执行（有主键约束的前提）。

闪回：

通过ROW格式binlog的逆操作进行回滚。

# 工具

## my2sql

### 概述

my2sql是使用go语言开发的MySQL binlog解析工具，通过解析MySQL binlog，可以生成原始SQL、回滚SQL、去除主键的INSERT SQL等，也可以生成DML统计信息。类似工具有binlog2sql、MyFlash、my2fback等，本工具基于my2fback、binlog\_rollback工具二次开发而来。

my2sql的GitHub地址：https://github.com/liuhr/my2sql

### 特点

#### 优点

功能丰富，不仅支持回滚操作，还有其他实用功能。

基于golang实现，速度快，全量解析1.1Gbinlog只需要1分30秒左右，当前其他类似开源工具一般要几十分钟。

#### 限制

使用回滚/闪回功能时，binlog格式必须为row,且binlog\_row\_image=full， DML统计以及大事务分析不受影响

只能回滚DML，不能回滚DDL

支持指定-tl时区来解释binlog中time/datetime字段的内容。开始时间-start-datetime与结束时间-stop-datetime也会使用此指定的时区，但注意此开始与结束时间针对的是binlog event header中保存的unix timestamp。结果中的额外的datetime时间信息都是binlog event header中的unix timestamp

此工具是伪装成从库拉取binlog，需要连接数据库的用户有SELECT, REPLICATION SLAVE, REPLICATION CLIENT权限

MySQL8.0版本需要在配置文件中加入default\_authentication\_plugin=mysql\_native\_password，用户密码认证必须是mysql\_native\_password才能解析

### 用途

数据快速回滚(闪回)

主从切换后新master丢数据的修复

从binlog生成标准SQL，带来的衍生功能

生成DML统计信息，可以找到哪些表更新的比较频繁

IO高TPS高， 查出哪些表在频繁更新

找出某个时间点数据库是否有大事务或者长事务

主从延迟，分析主库执行的SQL语句

除了支持常规数据类型，对大部分工具不支持的数据类型做了支持，比如json、blob、text、emoji等数据类型sql生成。

### 参数

-U 优先使用unique key作为where条件，默认false

-add-extraInfo 是否把database/table/datetime/binlogposition…信息以注释的方式加入生成的每条sql前，默认false

datetime=2020-07-16\_10:44:09 database=orchestrator table=cluster\_domain\_name binlog=mysql-bin.011519 startpos=15552 stoppos=15773  
 UPDATE `orchestrator`.`cluster\_domain\_name` SET `last\_registered`='2020-07-16 10:44:09' WHERE `cluster\_name`='192.168.1.1:3306'

-big-trx-row-limit n  
 transaction with affected rows greater or equal to this value is considerated as big transaction  
 找出影响了n行数据的事务，默认500条

-databases、-tables库及表条件过滤, 以逗号分隔

-sql 要解析的sql类型，可选参数insert、update、delete，默认全部解析

-doNotAddPrifixDb  
 Prefix table name witch database name in sql,ex: insert into db1.tb1 (x1, x1) values (y1, y1)  
 默认生成insert into db1.tb1 (x1, x1) values (y1, y1)类sql，也可以生成不带库名的sql

-file-per-table 为每个表生成一个sql文件

-full-columns  
 For update sql, include unchanged columns. for update and delete, use all columns to build where condition.  
 default false, this is, use changed columns to build set part, use primary/unique key to build where condition  
 生成的sql是否带全列信息，默认false

-ignorePrimaryKeyForInsert 生成的insert语句是否去掉主键，默认false

-output-dir 将生成的结果存放到制定目录

-output-toScreen 将生成的结果打印到屏幕，默认写到文件

-threads  线程数，默认8个

-work-type

2sql表示生成原始sql，rollback表示生成回滚sql，stats表示只统计DML、事务信息

### 使用

#### 解析出标准SQL

##### 根据时间点解析出标准SQL

/my2sql -user root -password xxxx -host 127.0.0.1 -port 3306 -work-type 2sql -start-file mysql-bin.011259 -start-datetime "2020-07-16 10:20:00" -stop-datetime "2020-07-16 11:00:00" -output-dir ./tmpdir

##### 根据pos点解析出标准SQL

/my2sql -user root -password xxxx -host 127.0.0.1 -port 3306 -work-type 2sql -start-file mysql-bin.011259 -start-pos 4 -stop-file mysql-bin.011259 -stop-pos 583918266 -output-dir ./tmpdir

#### 解析出回滚SQL

##### 根据时间点解析出回滚SQL

/my2sql -user root -password xxxx -host 127.0.0.1 -port 3306 -work-type rollback -start-file mysql-bin.011259 -start-datetime "2020-07-16 10:20:00" -stop-datetime "2020-07-16 11:00:00" -output-dir ./tmpdir

##### 根据pos点解析出回滚SQL

./my2sql -user root -password xxxx -host 127.0.0.1 -port 3306 -work-type rollback -start-file mysql-bin.011259 -start-pos 4 -stop-file mysql-bin.011259 -stop-pos 583918266 -output-dir ./tmpdir

#### 统计DML以及大事务

统计时间范围各个表的DML操作数量，统计一个事务大于500条、时间大于300秒的事务

./my2sql -user root -password xxxx -host 127.0.0.1 -port 3306 -work-type stats -start-file mysql-bin.011259 -start-datetime "2020-07-16 10:20:00" -stop-datetime "2020-07-16 11:00:00" -big-trx-row-limit 500 -long-trx-seconds 300 -output-dir ./tmpdir

统计一段pos点范围各个表的DML操作数量，统计一个事务大于500条、时间大于300秒的事务

./my2sql -user root -password xxxx -host 127.0.0.1 -port 3306 -work-type stats -start-file mysql-bin.011259 -start-pos 4 -stop-file mysql-bin.011259 -stop-pos 583918266 -big-trx-row-limit 500 -long-trx-seconds 300 -output-dir ./tmpdir

#### 从某一个pos点解析出标准SQL，并且持续打印到屏幕

./my2sql -user root -password xxxx -host 127.0.0.1 -port 3306 -work-type 2sql -start-file mysql-bin.011259 -start-pos 4 -output-toScreen

### 案例演示

## binlog2sql

## MyFlash

# 对比

binlog2sql当前是业界使用最广泛的MySQL回滚工具，下面对my2sql和binlog2sql做个性能对比。



# 缺点

要求binlog格式必须为ROW格式。

闪回的binlog不宜太多，否则效率较低。

无法闪回DDL操作。

要求闪回的表带主键。

# 分布式数据库实践

## TDSQL

## GoldenDB

### 概述

对drop表进行备份，在用户或者运维人员误操作的情况下，能够通过闪回功能恢复原表，减少损失。

### 设计思路

drop：在RDB上新建表的dict\_resyclebin，将drop表的操作转成drop\_into\_recyclebin，db上的表放入回收站中，元数据中将表信息从dictionary\_info移入dict\_resyclebin中，删除索引信息，并记录版本号version（回收站中）。

flashback：根据proxy报上来的集群号、表名以及版本号在dict\_recyclebin中找到对应表，将其移回dictionary\_info中，并将frm、pal、sql、bk语句信息返回给proxy。

purge：根据proxy报上来的purge类型（清空某集群下所有的表、清空一张表的所有副本、清楚一张表），在回收站中进行对应的操作。

Proxy查询回收站中的表：支持两种查询：

1. proxy只报集群号，MDS下发该集群下所有库表以及版本号
2. Proxy报集群号、库表名，MDS下发对应的表以及所有版本号

重分布中的drop操作：在原有的drop语句最后加上“purge”，代表真删除，与闪回的drop区分。

### 流程梳理

#### 交互流程

#### 内部流程

## TiDB

## OceanBase