



链滴

MySQL 日志

作者: [Hefery](#)

原文链接: <https://ld246.com/article/1641520667152>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

二进制日志binary-log

二进制日志：记录了所有对MySQL数据库成功执行的修改事件，包括增删改查事件和对表结构的修事件

作用：基于时间点的备份和恢复；主从复制

二进制日志的格式

- 基于段的格式 `binlog_format=STATEMENT`

优点：日志记录量相对较小，节约磁盘及网络I/O

缺点：可能造成MySQL复制的主备服务器数据不一致

- 基于行的日志格式 `binlog_format=ROW`

优点：Row格式可以避免MySQL复制中出现的主从不一致问题,使MySQL主从复制更加安全；对每一数据的修改比基于段的复制高效

缺点：记录日志量较大.

`binlog_row_image=[FULL | MINIMAL | NOBLOB]`

- 混合日志格式`binlog_format=MIXED`

特点：根据SQL语句由系统决在基于段和基于行的日志格式中选择；数据量的大小由所执行的SQL语决定

开启：my.ini中MySQLId组设置是关于二进制日志的：`log-bin`

查看：

```
# SHOW BINARY LOGS
show variables like 'binlog_format'
MYSQLBINLOG D:/mysql/log/binlog.001
```

设置：

```
set session binlog_format=STATEMENT | ROW
flush logs
```

```
log-bin [=base_name]
binlog_format=[ROW | STATEMENT | MIXED] # 二进制日志格式
binlog_row_image=[FULL | MINIMAL | NOBLOB] # Row格式设置
binlog_rows_query_log_events=[ON | OFF]
log_slave_updates=[ON | OFF]
sync_binlog=[1 | 0]
```

删除：

```
RESET MASTER # 删除所有二进制文件
PURGE MASTER LOGS TO "binlog.001" # 删除指定二进制文件
```

暂停和恢复：

```
SET sql_log_bin = 0 # 暂停
SET sql_log_bin = 1 # 恢复
```

使用二进制日志恢复数据库:

```
MYSQLBINLOG --stop-date="2021-08-30 15:27:48" D:/mysql/log/binlog.001 | mysql --uuser -ppass
```

错误日志error-log

错误日志: 记录MySQL服务的启动、运行或停止MySQL服务时出现的问题

作用: 分析排除MySQL运行错误; 记录未经授权的访问

启动: my.ini中MySQLId组设置是关于错误日志的: log_error=\$mysql/sql_log/mysql-error.log

查看:

```
SHOW VARIABLES LIKE "log_error" # 查看存储路径及文件名
```

删除:

服务端: MYSQLADMIN -u root -P flush-logs

客户端: flush logs

通用查询general-log

通用查询日志: 记录建立的客户端连接和执行的语句

启动: my.ini中MySQLId组设置是关于查询日志的: log

查看: D:\MySQL-5.7.10-win32\data目录下的myPC.log

删除: 物理删除 .err 日志文件, 执行 MYSQLADMIN -u root -P flush-logs

慢查询slow-query-log

慢查询: 记录所有执行时间超过 long_query_time的所有查询或不使用索引的查询

启动

```
# 开启慢查询
```

```
set global slow_query_log=on;
```

配置

slow_query_log=[ON | OFF]:是否开启慢查询日志

slow_query_log_file =\$mysql/sql_log/slowlog.log: 慢查询日志存储路径

long_query_time: 慢查询阈值, 当查询时间多于设定的阈值时, 记录日志

log_queries_not_using_indexes=[ON | OFF]: 未使用索引的查询也被记录到慢查询日志中

log_slow_admin_statements=[ON | OFF]

log_slow_slave_statements=[ON | OFF]

查看

```
SHOW VARIABLES LIKE '%slow_query_log%' # 查看存储路径及文件名
```

重做日志redo-log

作用

- 确保事务的持久性
- 防止在发生故障的时间点，尚有脏页未写入磁盘，在重启mysql服务的时候，根据redo log进行重做，从而达到事务的持久性这一特性

事务开始之后就产生redo log，redo log的落盘并不是随着事务的提交才写入的，而是在事务的执行过程中，便开始写入redo log文件中。当对应事务的脏页写入到磁盘之后，redo log的使命也就完成了重做日志占用的空间就可以重用（被覆盖）

innodb_log_group_home_dir 指定日志文件组所在的路径
innodb_log_files_in_group 指定重做日志文件组中文件的数量
innodb_log_file_size 重做日志文件的大小
innodb_mirrored_log_groups 指定了日志镜像文件组的数量

回滚日志undo-log

保存了事务发生之前的数据的一个版本，可以用于回滚，同时可以提供多版本并发控制下的读（MVCC），也即非锁定读

逻辑格式的日志，在执行undo的时候，仅仅是将数据从逻辑上恢复至事务之前的状态，而不是从物理页面上操作实现的

事务开始之前，将当前是版本生成undo log，undo也会产生redo来保证undo log的可靠性。当事务提交之后，undo log并不能立马被删除，而是放入待清理的链表，由purge线程判断是否由其他事在使用undo段中表的上一个事务之前的版本信息，决定是否可以清理undo log的日志空间

innodb_undo_directory = /data/undospace/ # undo独立表空间的存放目录
innodb_undo_logs = 128 # 回滚段为128KB
innodb_undo_tablespaces = 4 # 指定有4个undo log文件

中继日志relay-log

用于主从复制，临时存储从主库同步的二进制日志，用来给slave库恢复

relay_log=filename
relay_log_purge=[ON | OFF]