

# MySQL数据库常用的41个脚本，速来下载！

原创 chenjuchao IT小Chen 2025年03月25日 12:50 北京



## 1.连接信息查询

```
1 show processlist;
2 select * from information_schema.processlist limit 10;
3 select user,SUBSTRING_INDEX(host,':',1) as ip,db,count(*) as num from i
```

## 2.查询非Sleep会话

```
1 select id,USER,HOST,DB,COMMAND,TIME,STATE from information_schema.proce
2 select id,USER,HOST,DB,COMMAND,TIME,STATE,info from information_schema.
```

## 3.通过ID查询对应的SQL语句

```
1 select info from information_schema.processlist where id=XXX\G;
```

## 4.运行进程查询

```
1 select now();
2
```

```
select host,user,Command,Time,State,info from information_schema.proces
```

## 5.批量杀会话

```
1 select concat('kill ',id,';'),host,user,command,time,state,info from in
2 where command !='Sleep' and user not in
3 ('repl','system user','event_scheduler')
4 order by time desc limit 10;
```

## 6.锁信息查询

启动锁监控

```
1 update performance_schema.setup_instruments set enabled='YES' where name
```

查询

```
1 SELECT * FROM INFORMATION_SCHEMA.INNODB_LOCKS;
2 SELECT * FROM INFORMATION_SCHEMA.INNODB_LOCK_WAITS;
3 show status like '%key_read%';
4 show status like 'Qcache%';
5
6 select table_schema,table_name,table_type from information_schema.table
7 where table_schema not in ('information_schema','mysql','performance_sc
8 and table_name not in
9 (
10 select table_name
11 from (
12 select table_name,index_name
13 from information_schema.statistics
14 where table_schema not in ('information_schema','mysql','performance_sc
15 group by table_name,index_name) tab_ind_cols
16 group by table_name
17 );
```

## 7.检查会话阻塞

检查

```
1 show processlist;
2 show full processlist;
3 select id,user,host,db,command,time,state,info from information_schema.
```

### 查看事务

```
1 select * from information_schema.innodb_trx\G;
```

### 查看正在锁的事务

```
1 select * from information_schema.innodb_locks;
```

### 查看等待锁的事务

```
1 select * from information_schema.innodb_lock_waits;
```

### 检查 metadata lock

```
1 select * from performance_schema.setup_instruments where name like '%lo
2 select * from performance_schema.metadata_locks;
```

### 查看锁源头

```
1 select * from performance_schema.metadata_locks\G;
```

### 查看锁源头最近一次执行的SQL

```
1 select * from performance_schema.events_statements_current where thread_
```

### 查询thread\_id和processlist\_id对应关系

```
1 select name,thread_id,processlist_id ,thread_os_id from performance_sche
```

## 分析锁源头

```

1  select
2      locked_schema,
3      locked_table,
4      locked_type,
5      waiting_processlist_id,
6      waiting_age,
7      waiting_query,
8      waiting_state,
9      blocking_processlist_id,
10     blocking_age,
11     substring_index(sql_text,"transaction_begin;" ,-1) as blocking_query,
12     sql_kill_blocking_connection
13  from
14      (
15          select
16              b.owner_thread_id as granted_thread_id,
17              a.object_schema as locked_schema,
18              a.object_name as locked_table,
19              "metadata lock" as locked_type,
20              c.processlist_id as waiting_processlist_id,
21              c.processlist_time as waiting_age,
22              c.processlist_info as waiting_query,
23              c.processlist_state as waiting_state,
24              d.processlist_id as blocking_processlist_id,
25              d.processlist_time as blocking_age,
26              d.processlist_info as blocking_query,
27              concat('kill ', d.processlist_id) as sql_kill_blocking_connection
28          from
29              performance_schema.metadata_locks a
30          join performance_schema.metadata_locks b on a.object_schema =
31              and a.object_name = b.object_name
32              and a.lock_status = 'pending'
33              and b.lock_status = 'granted'
34              and a.owner_thread_id <> b.owner_thread_id
35              and a.lock_type = 'exclusive'
36          join performance_schema.threads c on a.owner_thread_id = c.thread_id
37          join performance_schema.threads d on b.owner_thread_id = d.thread_id
38      ) t1,
39      (
40          select
41              thread_id,
42              group_concat( case when event_name = 'statement/sql/begin'
43                  from
44                      performance_schema.events_statements_history

```

```
45         group by thread_id
46     ) t2
47 where
48     t1.granted_thread_id = t2.thread_id \G;
```

## 8.存储引擎信息

```
1 show engine innodb status\G;
```

## 9.数据库大小查询

按库

```
1 select table_schema as DB_NAME,concat(round(sum(DATA_LENGTH/1024/1024),2),'
```

按表

```
1 select TABLE_NAME,DATA_LENGTH/1024/1024 as TAB_SIZE,INDEX_LENGTH/1024/10
```

全部

```
1 select concat(round(sum(DATA_LENGTH/1024/1024),2),'MB') as DB_DATA_SIZE,
```

## 10.查询表字符集

```
1 SELECT
2 T.TABLE_SCHEMA,T.TABLE_NAME,T.ENGINE,CCSA.character_set_name
3 FROM information_schema.`TABLES` T, information_schema.`COLLATION_CHARA
4 WHERE CCSA.collation_name = T.table_collation AND
5 T.table_schema in ('xxx') ORDER BY 1,2;
```

## 11.查询行格式为Compact，并且列数>20的表

```
1 select
```

```

2  concat ('alter table ',a.TABLE_SCHEMA,'.',a.TABLE_NAME,' row_format=DYNAMIC
3  from
4  information_schema.tables a,(select TABLE_SCHEMA,TABLE_NAME,count(*) fr
5  where
6  a.TABLE_SCHEMA=b.TABLE_SCHEMA and
7  a.TABLE_NAME=b.TABLE_NAME and
8  ROW_FORMAT = 'Compact';

```

## 12.页

页压缩的表数据文件，在某些文件系统下，cp后，会导致表放大，占用更多的磁盘空间

查询哪些表配置了页压缩属性

```

1  SELECT TABLE_NAME, TABLE_SCHEMA, CREATE_OPTIONS FROM INFORMATION_SCHEMA.

```

拼接出表空间格式名称，例如:cjc/t1

```

1  SELECT concat(TABLE_SCHEMA,"/",TABLE_NAME) FROM INFORMATION_SCHEMA.TABLES

```

计算cp复制额外需要的空间

```

1  SELECT NAME, FILE_SIZE, ALLOCATED_SIZE,(FILE_SIZE-ALLOCATED_SIZE)/1024/
2  (SELECT concat(TABLE_SCHEMA,"/",TABLE_NAME) FROM INFORMATION_SCHEMA.TAB

```

总大小

```

1  SELECT SUM(FILE_SIZE-ALLOCATED_SIZE)/1024/1024/1024 "GB" FROM INFORMATION_SCHEMA.
2  (SELECT concat(TABLE_SCHEMA,"/",TABLE_NAME) FROM INFORMATION_SCHEMA.TAB

```

## 13.检查注释信息

如果存在注释乱码的问题，可能会影响到MySQL升级。

```

1  SELECT table_schema,table_name,column_name,hex(column_comment),column_c

```

```
2 select TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME, COLUMN_COMMENT, LENGTH(COLUMN_COMMENT) from INFORMATION_SCHEMA.COLUMNS where TABLE_SCHEMA = 'test'
3 select TABLE_SCHEMA, TABLE_NAME, TABLE_COMMENT, length(TABLE_COMMENT) from INFORMATION_SCHEMA.TABLES where TABLE_SCHEMA = 'test'
```

## 14.检

查询比较耗时

## 5.7

```
1 SELECT NAME FROM INFORMATION_SCHEMA.INNODB_SYS_TABLESPACES WHERE
2 NAME NOT LIKE '%#%' AND NAME NOT LIKE '%@%' AND NAME NOT LIKE '%FTS_000%'
3 NAME NOT IN (SELECT CONCAT(TABLE_SCHEMA, '/', TABLE_NAME) NAME FROM INFORMATION_SCHEMA.TABLES)
```

## 15.运行参数查询

```
1 select * from performance_schema.global_status where variable_name in
2 (
3 'Bytes_received',
4 'Bytes_sent',
5 'Connections',
6 'Locked_connects',
7 'Max_used_connections',
8 'Aborted_clients',
9 'Aborted_connects',
10 'Threads_cached',
11 'Threads_connected',
12 'Threads_created',
13 'Threads_running',
14 'Key_blocks_not_flushed',
15 'Key_blocks_unused',
16 'Key_blocks_used',
17 'Table_locks_immediate',
18 'Table_locks_waited',
19 'Key_read_requests',
20 'Key_reads',
21 'Qcache_free_blocks',
22 'Qcache_free_memory',
23 'Qcache_hits',
24 'Qcache_inserts',
25 'Qcache_lowmem_prunes',
26 'Qcache_not_cached',
27 'Qcache_queries_in_cache',
```

```
28 'Qcache_total_blocks',
29 'Created_tmp_disk_tables',
30 'Created_tmp_files',
31 'Created_tmp_tables',
32 'Slow_launch_threads',
33 'Slow_queries'
34 );
```

## 16.全局参数查询

```
1 select * from performance_schema.global_variables where variable_name
2 (
3 'skip_name_resolve',
4 'skip_external_locking',
5 'lower_case_table_names',
6 'event_scheduler',
7 'max_connections',
8 'max_user_connections',
9 'wait_timeout',
10 'interactive_timeout',
11 'max_allowed_packet',
12 'binlog_format',
13 'expire_logs_days',
14 'innodb_buffer_pool_size',
15 'innodb_buffer_pool_instances',
16 'innodb_flush_log_at_trx_commit',
17 'innodb_lock_wait_timeout',
18 'gtid_mode',
19 'enforce_gtid_consistency',
20 'auto_increment_increment',
21 'auto_increment_offset'
22 );
```

## 17.用户信息查询

```
1 select user,host,authentication_string,password_expired,password_last_ch
```

## 18.对象信息查询

```
1 select table_schema,table_type,count(*) from information_schema.tables
```



```
2 select table_schema,table_name,count(*) from information_schema.partiti
```

## 19.统计信息自查询

```
1 select table_schema,count(*)
2 from information_schema.statistics
3 where table_schema not in ('information_schema','mysql','performance_sc
4 group by table_schema;
```

## 20.视图信息查询

```
1 select table_schema,table_name,definer from information_schema.views whe
```

## 21.触发器信息查询

```
1 select trigger_schema,trigger_name,definer from information_schema.trigg
```

## 22.存储过程信息查询

```
1 select routine_schema,routine_name,routine_type,definer from information
```

## 23.event查询

```
1 select event_schema,event_name,definer from information_schema.events wh
```

## 24.表信息查询

```
1 select table_schema,table_name,table_type,table_rows from information_sc
```

## 25.分区表

```
1 select table_schema,table_name,count(*) from information_schema.partiti
2 select table_schema,table_name,max(partition_name) from information_sch
```

## 26.events查询

```
1 show variables like 'event_scheduler';
2 select event_schema,event_name,definer from information_schema.events w
```

## 27.数据库存在没有主键业务表

```
1 SELECT
2 t1.table_schema,
3 t1.table_name
4 FROM
5 information_schema.TABLES t1
6 LEFT OUTER JOIN information_schema.TABLE_CONSTRAINTS t2 ON t1.table_sch
7 AND t1.table_name = t2.TABLE_NAME
8 AND t2.CONSTRAINT_NAME IN ('PRIMARY')
9 WHERE
10 t2.table_name IS NULL
11 AND t1.table_type = 'BASE TABLE'
12 AND t1.TABLE_SCHEMA NOT IN (
13 'information_schema',
14 'performance_schema',
15 'mysql',
16 'sys'
17 );
```

## 28.查询主键自增SQL

```
1 SELECT
2 t.TABLE_NAME,
3 c.COLUMN_NAME,
4 ts.AUTO_INCREMENT
5 FROM
6 INFORMATION_SCHEMA.TABLE_CONSTRAINTS AS t,
7 information_schema.TABLES AS ts,
8 information_schema.KEY_COLUMN_USAGE AS c
9 WHERE
```

```
10 t.TABLE_NAME = ts.TABLE_NAME
11 AND ts.TABLE_NAME = c.TABLE_NAME
12 AND t.TABLE_SCHEMA = 'XXX'
13 AND t.CONSTRAINT_TYPE = 'PRIMARY KEY'
14 ORDER BY ts.`AUTO_INCREMENT` DESC;
```

## 29.导出到CSV

```
1 select .....
2 INTO OUTFILE '/home/mysql/xxxxx.csv' FIELDS TERMINATED BY ',' ENCLOSED
3 from tab where ...;
```

- 1 secure-file-priv=''和secure-file-priv=时，无限制，任何目录都可以导出
- 2 secure-file-priv=NULL时，无权限，任何目录都不能导出
- 3 secure-file-priv=指定路径 时，指定路径有导出权限，其他目录不能导出

## 30.LOAD 导入

```
1 LOAD DATA INFILE '/db/mysqldata/3306/file/t1a.csv'
2 INTO TABLE t2
3 FIELDS TERMINATED BY ','
4 ENCLOSED BY '"'
5 LINES TERMINATED BY '\n'
6 IGNORE 1 LINES;
```

### 普通用户

```
1 LOAD DATA LOCAL INFILE '/db/mysqldata/3306/file/t1a.csv'
2 INTO TABLE t2
3 FIELDS TERMINATED BY ','
4 ENCLOSED BY '"'
5 LINES TERMINATED BY '\n'
6 IGNORE 1 LINES;
```

## 31.mysqlDump常用语句

mysqldump备份参考语句：

- 1 帮助信息

```
2 mysqldump --help
```

```
1 全库备份
2 mysqldump -uroot -p --all-databases --hex-blob --single-transaction --s
3 8.0.33 基于 GTID 全备
4 mysqldump -uroot -p --all-databases --hex-blob --source-data=2 --single
5 指定数据库备份
6 mysqldump -uroot -p cjc --hex-blob --single-transaction --set-gtid-purc
7 指定表备份
8 mysqldump -uroot -p cjc t2 t3 --hex-blob --single-transaction --set-gti
9 只备份表结构
10 mysqldump -uroot -p cjc t3 --no-data --hex-blob --single-transaction --
11 只备份表数据
12 mysqldump -uroot -p cjc t3 --no-create-info --hex-blob --single-transac
13 用户备份
14 mysqlpump -udumper -p --set-gtid-purged=OFF --exclude-databases=% --use
15 指定库备份：
16 ---导出单个数据库，不指定 -databases参数，不会生成create database 和use dbname
17 mysqldump -u用户名 -p 数据库名 --extended-insert=TRUE --hex-blob --quick
```

## 32.tps,qps查询

```
1 cat mysql_tps_qps_detail.sh
2 #!/bin/bash
3 function check_tps_qps {
4 date +%F_%T >> mysql_tps_qps_detail.log
5 mysqladmin -uroot -p'*****' extended-status -i1 -c2|awk 'BEGIN{local_
6 $2 ~ /Queries$/ {q=$4-lq;lq=$4;}
7 $2 ~ /Com_commit$/ {c=$4-lc;lc=$4;}
8 $2 ~ /Com_rollback$/ {r=$4-lr;lr=$4;}
9 $2 ~ /Innodb_rows_deleted$/ {deleted=$4-ldeleted;ldeleted=$4;}
10 $2 ~ /Innodb_rows_inserted$/ {inserted=$4-linserted;linserted=$4;}
11 $2 ~ /Innodb_rows_read$/ {read=$4-lread;lread=$4;}
12 $2 ~ /Innodb_rows_updated$/ {updated=$4-lupdated;lupdated=$4;}
13 $2 ~ /Threads_connected$/ {tc=$4;}
14 $2 ~ /Threads_running$/ {tr=$4;
15 if(local_switch==0)
16 {local_switch=1; count=0}
17 else {
18 if(count>10)
19 {count=0;print "-----"
20 else{
21 count+=1;
```

```

22 printf "%-6d %-8d %-7d %-8d %-6d %-8d %-7d %-8d %-10d %d \n", q,c,r,c+
23 } } }' >> mysql_tps_qps_detail.log
24 }
25 for i in {1..10}
26 do
27 check_tps_qps
28 done

```

### 33.查询 binlog

```

1 show binary logs;
2 show master status;
3 show binlog events;
4 show binlog events in 'mysql-bin.00000x';
5 show binlog events in 'mysql-bin.00000x' from 100; ---pos >=100
6 show binlog events in 'mysql-bin.00000x' from 100 limit 10; ---pos >=100
7 show binlog events in 'mysql-bin.00000x' from 100 limit 10,3; ---pos >=100

```

### 34.删除 binlog

```

1 删除所有binlog
2 purge binary logs;
3 ###删除指定名称之前的所有文件(不包括此文件)
4 purge binary logs to 'mysql-bin.000023';
5 ###删除指定时间之前的所有文件
6 purge binary logs before '2023-11-11 12:00:00';

```

### 35.mysqlbinlog

```

1 mysqlbinlog --no-defaults --base64-output=decode-rows --verbose -vvv my
2 mysqlbinlog --no-defaults --base64-output=decode-rows --verbose -vvv my
3 mysqlbinlog --no-defaults --base64-output=decode-rows --verbose -vvv my

```

其中：grep -C 中的C表示前后

B是前面 A是后面 C是前后

### 16.高可用查询

```
1 show slave status\G;
2 show slave hosts\G;
3 show master status\G;
4 show variables like '%semi%';
5 show global status like '%semi%';
```

### 37.主从配置

主从:

```
1 create user repl@'IP' identified with mysql_native_password by '*****'
2 grant replication slave on *.* to repl@'IP';
```

```
1 #MySQL 8.0
2 CHANGE REPLICATION SOURCE TO
3 SOURCE_HOST = 'IP',
4 SOURCE_PORT = 3306,
5 SOURCE_USER = 'repl',
6 SOURCE_PASSWORD = '*****',
7 SOURCE_AUTO_POSITION = 1,
8 SOURCE_CONNECT_RETRY=10;
9
10 CHANGE REPLICATION SOURCE TO
11 SOURCE_HOST='source_host_name',
12 SOURCE_USER='replication_user_name',
13 SOURCE_PASSWORD='replication_password',
14 SOURCE_LOG_FILE='recorded_log_file_name',
15 SOURCE_LOG_POS=recorded_log_position,
16 SOURCE_CONNECT_RETRY=10;
17
18 #MySQL 5.7
19 CHANGE MASTER TO
20 MASTER_HOST='IP',
21 MASTER_USER='repl',
22 MASTER_PASSWORD='*****',
23 MASTER_PORT=3306,
24 MASTER_AUTO_POSITION=1,
25 MASTER_CONNECT_RETRY=10;
26
27 CHANGE MASTER TO
28 MASTER_HOST='IP',
29 MASTER_USER='repl',
```

```
30 MASTER_PASSWORD='*****',
31 MASTER_PORT=3306,
32 MASTER_LOG_FILE='mysql-bin.000001',
33 MASTER_LOG_POS=154,
34 MASTER_CONNECT_RETRY=10;
```

### 38.跳过单个事务

也就是将执行失败的事务，通过begin;commit;的方式替换为一个空事务。

```
1 set gtid_next='*****-3788-11ef-a068-*****:83';
2 begin;
3 commit;
4 set gtid_next=automatic;
5 start slave;
```

### 39.统计某类文件大小

```
1 ls mysql-relay-bin.* | xargs du -s | awk '{sum += $1} END {print sum/10
2 ls mysql-bin.* | xargs du -s | awk '{sum += $1} END {print sum/1024"M"}'
```

### 40.输出执行结果

```
1 mysql -uroot -p --tee=/cjc/dbtmpfile/20231111/mysql_20231111.log
```

### 41.设置过滤

```
1 show processlist;
2 show full processlist;
```

#### 过滤信息

```
1 pager grep -v Sleep
2 pager less
```

#### 取消过滤

```
1 pager
```

###chenjuchao 20250325###

欢迎关注我的公众号《IT小Chen》



IT小Chen

陈举超，Oracle ACE-Pro、墨天伦MVP、OCMU 用户组成员、ACDU成员、ITPUB专家博...  
444篇原创内容

公众号



chenjuchao

稀罕作者

数据库 151    MySQL 28

数据库 · 目录

上一篇

16-达梦数据库物理恢复(完全恢复)

下一篇

17-达梦数据库物理恢复(不完全恢复)



个人观点，仅供参考