



产品

解决方案

文档

客户案例

合作伙伴

服务与支持

学习与社区

搜索

语言



免费试用

TiDB 社区智慧合集 | TiDB 相关 SQL 脚本大全

← 查看全部博客

作者：PingCAP

案例实践

2024-02-22

非常感谢各位 TiDBer 在之前 【TiDBer 唠嗑茶话会 48】非正式 TiDB 相关 SQL 脚本征集大赛! (<https://asktug.com/t/topic/996635>)里提供的各种常用脚本。

在这篇文章中，我们整理了社区同学提供的一系列 TiDB 相关 SQL 脚本，希望大家在 TiDB 的使用过程中提供一些帮助和参考。这些脚本涵盖了常见场景下的 SQL 操作，欢迎各位 TiDBer 持续补充更新~

未来，我们也将整理更多 TiDB 相关实用指南，帮助大家更好地了解、运用 TiDB，敬请期待！

TiDB 相关 SQL 脚本大全

1 缓存表

贡献者：@ShawnYan

```
alter table xxx cache|nocache;
```

2 TSO 时间转换

贡献者：@我是咖啡哥

- 方法一：使用函数 TIDB_PARSE_TSO

```
SELECT TIDB_PARSE_TSO(437447897305317376);
+-----+
| TIDB_PARSE_TSO(437447897305317376) |
+-----+
| 2022-11-18 08:28:17.704000 |
+-----+
1 row in set (0.25 sec)
```

- 方法二：使用 pd-ctl

```
~$ tiup ctl:v6.4.0 pd -i -u http://pdip:2379
Starting component `ctl`: /Users/xxx/.tiup/components/ctl/v6.4.0/ctl pd -i -u http://pdip:2379
» tso 437447897305317376
system: 2022-11-18 08:28:17.704 +0800 CST
logic: 0
```

3 读取历史数据

贡献者：@我是咖啡哥

- 使用 AS OF TIMESTAMP 语法读取历史数据，可以通过以下三种方式使用 AS OF TIMESTAMP 语法：

SELECT ... FROM ... AS OF TIMESTAMP

START TRANSACTION READ ONLY AS OF TIMESTAMP

SET TRANSACTION READ ONLY AS OF TIMESTAMP

```
select * from t as of timestamp '2021-05-26 16:45:26';
start transaction read only as of timestamp '2021-05-26 16:45:26';
set transaction read only as of timestamp '2021-05-26 16:45:26';
```

- 通过系统变量 tidb_read_staleness 读取历史数据

从 5 秒前至现在的时间范围内选择一个尽可能新的时间戳

```
set @@tidb_read_staleness="-5";
```

- 通过系统变量 tidb_snapshot 读取历史数据

设置一个特殊的环境变量，这个是一个 session scope 的变量，其意义为读取这个时间之前的最新的一个版本

```
set @@tidb_snapshot="2016-10-08 16:45:26";
```

清空这个变量后，即可读取最新版本数据

```
set @@tidb_snapshot="";
```

4 查询 tikv_gc_life_time 和 tikv_gc_safe_point 默认时长

贡献者：@TiDBer_m6V1BaIM

```
select VARIABLE_NAME, VARIABLE_VALUE from mysql.tidb where VARIABLE_NAME like "tikv_gc%";
```

5 搜索某个用户的 TopN 慢查询

贡献者：@fanruinet

```
select query_time,query,user
from information_schema.slow_query
where is_internal=false -- 排除 TiDB 内部的慢查询 SQL
and user = "user1" -- 查找的用户名
order by query_time desc
limit 2;
```

6 统计间隔 5 分钟的数据

贡献者：@forever

```
SELECT concat(date_format(create_time,'%Y-%m-%d %H:'),floor(date_format(create_time,'%i')/5)),count(*)
FROM jcxx
GROUP BY 1;
```

7 反解析 digest 成 SQL 文本

贡献者：@hey-hoho

```
select tidb_decode_sql_digests('["xxxxx"]');
```

8 不涉及分区表用下面的方式查看表的使用情况

贡献者：@xfworld

```
select TABLE_SCHEMA, TABLE_NAME, TABLE_ROWS,
(DATA_LENGTH+INDEX_LENGTH)/1024/1024/1024 as table_size from tables order by table_size
desc limit 20;
```

9 partition 表提供了分区表和非分区表的资源使用情况

贡献者：@xfworld

```
select TABLE_SCHEMA, TABLE_NAME, PARTITION_NAME, TABLE_ROWS,
(DATA_LENGTH+INDEX_LENGTH)/1024/1024/1024 as table_size from
information_schema.PARTITIONS order by table_size desc limit 20;
```

10 查询分析器中看配置文件参数

贡献者：@Kongdom

```
show config
```

SHOW CONFIG 语句用于展示 TiDB 各个组件当前正在应用的配置，请注意，配置与系统变量作用于不同维度，请不要混淆，如果希望获取系统变量信息，请使用 SHOW VARIABLES (<https://docs.pingcap.com/zh/tidb/stable/sql-statement-show-variables>) 语法。

11 查找读流量排名前 10 的热点 region

贡献者：@BraveChen

```
SELECT DISTINCT region_id
FROM INFORMATION_SCHEMA.tikv_region_status
WHERE READ_BYTES > ?
ORDER BY READ_BYTES DESC
LIMIT 10
```


12 查看参数和变量的脚本

贡献者：@buddyyuan

```
#!/bin/bash
case $1 in
-pd)
mysql -uroot -h127.0.0.1 -P4000 -p"" -e "SHOW CONFIG WHERE type ='pd' and name like '%$2%'"
;;
-tidb)
mysql -uroot -h127.0.0.1 -P4000 -p"" -e "SHOW CONFIG WHERE type ='tidb' and name like '%$2%'"
;;
-tikv)
mysql -uroot -h127.0.0.1 -P4000 -p"" -e "SHOW CONFIG WHERE type ='tikv' and name like '%$2%'"
;;
-tiflash)
mysql -uroot -h127.0.0.1 -P4000 -p"" -e "SHOW CONFIG WHERE type ='tiflash' and name like '%$2%'"
;;
-var)
mysql -uroot -h127.0.0.1 -P4000 -p"" -e "show variables like '%$2%';"
;;
-h)
echo "-pd      show pd parameters"
echo "-tidb    show tidb parameters"
echo "-tikv    show tikv parameters"
echo "-tiflash  show tiflash parameters"
echo "-var     show itidb variables"
;;
esac
```

还能用 grep 在过滤一次

```
[root@vm172-16-201-125 ~]# sh showparammeter.sh -tikv memory-pool-quota | grep -i "210:29160"
tikv      192.16.201.210:29160      server.grpc-memory-pool-quota      9223372036854775807B
```

13 查找重复记录

贡献者：@ealam_ 小羽

```
select *
from 表
where 重复字段 in
(
select 重复字段
from 表
group by 重复字段
having count(*)>1
)
```

14 查询耗时最高的慢 sql

贡献者：@caiyc

```
select query sql_text,
sum_query_time,
mnt as executions,
avg_query_time,
avg_proc_time,
avg_wait_time,
max_query_time,
avg_backoff_time,
Cop_proc_addr,
digest,
(case
when avg_proc_time = 0 then
'point_get or commit'
when (avg_proc_time > avg_wait_time and
avg_proc_time > avg_backoff_time) then
'coprocessor_process'
when (avg_backoff_time > avg_wait_time and
avg_proc_time < avg_backoff_time) then
'backoff'
else
'coprocessor_wait'
end) as type
from (select substr(query, 1, 100) query,
count(*) mnt,
avg(query_time) avg_query_time,
avg(process_time) avg_proc_time,
avg(wait_time) avg_wait_time,
max(query_time) max_query_time,
sum(query_time) sum_query_time,
digest,
Cop_proc_addr,
avg(backoff_time) avg_backoff_time
from information_schema.cluster_slow_query
where time >= '2022-07-14 17:00:00'
and time <= '2022-07-15 17:10:00'
and DB = 'web'
group by substr(query, 1, 100)) t
order by max_query_time desc limit 20;
```

15 日常维护用的最多的 SQL

贡献者：@tracy0984

```
select * from information_schema.cluster_processlist;
-- kill id;
```

16 恢复数据（适用于 drop 与 truncate）

贡献者：@凌云 Cloud

```
FLASHBACK TABLE target_table_name[TO new_table_name]
```

17 批量修改库名

贡献者：@TiDBer_dog

```
./bat_rename.sh lihongbao/ dev2_kelun dev2_sinodemo 路径./leo_backup
```

18 高并发的场景下获取 sql

贡献者：@jiawei

```
select * from information_schema.processlist where info is not null
```

19 查看 schema 下的表都有哪些

贡献者：@Ming

```
show tables in schema;
```

20 查看表 leader

贡献者：@TiDBer_wTKU9jv6

```
select count(1),tss.ADDRESS from INFORMATION_SCHEMA.TIKV_REGION_PEERS trp,INFORMATION_SCHEMA.TIKV_REG
```

21 shell 的调皮加速脚本

贡献者：@gcworkerishungry

```
alias ctidb="mysql -u root -ptidb -Dcktest -h S001 -P4000"
alias dtidb="tiup cluster display tidb-test"
alias etidb="tiup cluster edit-config tidb-test"
alias ptidb="tiup cluster prune tidb-test"
alias rtidb="tiup cluster restart tidb-test"
```

22 恢复数据到新的数据库

贡献者：@TiDBer_ 徐川

```
./loader -h 192.168.180.3 -u root -p q1w2 -P 4000 -t 32 -d leo_backup/
```

23 开启 tiflash

贡献者：@TiDBer_pFFcXLgY

```
alter table xxx set tiflash replica 1
```

24 表 region 分布语句

贡献者：@秋枫之舞


```
select
  trs.db_name,
  trs.table_name,
  trs.index_name,
  trp.store_id,
  count(*),
  sum(approximate_keys)
from
  information_schema.tikv_region_status trs,
  information_schema.tikv_store_status tss,
  information_schema.tikv_region_peers trp
where
  trs.db_name = 'prd01'
  and trs.table_name = 'tab_name'
  and trp.is_leader = 1
  and trp.store_id = tss.store_id
  and trs.region_id = trp.region_id
group by
  trs.db_name,
  trs.table_name,
  trs.index_name,
  trp.store_id
order by
  trs.index_name;
```

25 查看列的元数据

贡献者：@张雨齐0720

```
show stats_histograms where db_name like 'test' and table_name like 'test1' ;
```

26 表的存储位置（store、peer 信息）

贡献者：@bert

```
SELECT distinct a.TIDB_TABLE_ID, b.DB_NAME, b.TABLE_NAME, b.REGION_ID, b.APPROXIMATE_SIZE
, c.PEER_ID, c.STORE_ID, c.IS_LEADER, c.STATUS, d.ADDRESS
, d.STORE_STATE_NAME, d.VERSION, d.CAPACITY, d.AVAILABLE, d.LABEL
FROM INFORMATION_SCHEMA.TABLES a
INNER JOIN INFORMATION_SCHEMA.TIKV_REGION_STATUS b
INNER JOIN INFORMATION_SCHEMA.TIKV_REGION_PEERS c
INNER JOIN INFORMATION_SCHEMA.TIKV_STORE_STATUS d
WHERE a.TIDB_TABLE_ID = b.TABLE_ID
AND b.REGION_ID = c.REGION_ID
AND c.STORE_ID = d.STORE_ID
AND a.TABLE_SCHEMA = 'test'
AND a.TABLE_NAME = 't';
```

27 将集群升级到指定版本（在线升级）

贡献者：@TiDBer_ 杨龟干外公

```
tiup cluster upgrade
例如升级到 v4.0.0 版本：
tiup cluster upgrade tidb-test v4.0.0
```

28 查询表大小

贡献者：@我是咖啡哥

```
SELECT
  t.TABLE\_NAME,
  t.TABLE\_ROWS,
  t.TABLE\_TYPE,
  round(t.DATA\_LENGTH/1024/1024/1024,2) data\_GB,
  round(t.INDEX\_LENGTH/1024/1024/1024,2) index\_GB,
  t.CREATE\_OPTIONS,
  t.TABLE\_COMMENT
FROM
  INFORMATION\_SCHEMA.`TABLES` t
WHERE
  table\_schema = 'test'
  and t.table\_type='BASE TABLE'
order by t.TABLE\_ROWS desc;
SELECT CONCAT(table\_schema, '.', table\_name) AS 'Table Name', table\_rows AS 'Number of Rows', CONCAT
```

29 统计信息

贡献者：@我是咖啡哥

- 查看表的元数据

```
show stats\_meta where db\_name like '%sbtest%';
```

- 查看表的健康状态

```
show stats\_healthy;
```

Healthy 字段，一般小于等于 60 的表需要做 analyze

```
show stats\_healthy where table\_name = 'xxx';  
show stats\_healthy where db\_name=' ' and table\_name='orders';  
\_name like 'sbtest' and table\_name like 'sbtest1' ;
```

- 查看直方图信息

```
show stats\_buckets where db\_name=' ' and table\_name=' ';
```

- 查看 analyze 状态

```
show analyze status;
```

- 分析表、分区

```
analyze table sbtest1;  
ANALYZE TABLE xxx PARTITION P202204;
```

30 执行计划

贡献者：@我是咖啡哥

绑定执行计划

- 默认是 session 级别

```
create binding for select \* from t using select \* from t use index()  
create binding for SELECT \* FROM t1 INNER JOIN t2 ON t1.id = t2.t1\_id WHERE t1.int\_col = ? using  
explain SELECT \* FROM t1 INNER JOIN t2 ON t1.id = t2.t1\_id WHERE t1.int\_col = 1;  
show bindings for SELECT \* FROM t1 INNER JOIN t2 ON t1.id = t2.t1\_id WHERE t1.int\_col = 1;  
show global bindings;  
show session bindings;  
SELECT @@SESSION.last\_plan\_from\_binding;
```

- 使用 explain format = ‘verbose’ 语句查看 SQL 的执行计划

```
explain format = 'verbose';  
drop binding for sql;
```

31 查看 regions

贡献者：@我是咖啡哥

```
SHOW TABLE t\_its\_unload\_priority\_intermediate\_info regions;  
SHOW TABLE t\_its\_unload\_priority\_intermediate\_info INDEX IDX\_UPII\_GROUP\_BY\_COMPOSITE regions
```

32 统计读写热点表

贡献者：@我是咖啡哥


```
use INFORMATION\__SCHEMA;
SELECT
db\_name,
table\_name,
index\_name,
type,
sum( flow\_bytes ),
count( 1 ),
group\_concat( h.region\_id ),
count( DISTINCT p.store\_id ),
group\_concat( p.store\_id )
FROM
INFORMATION\__SCHEMA.tidb\_hot\_regions h
JOIN INFORMATION\__SCHEMA.tikv\_region\_peers p ON h.region\_id = p.region\_id
AND p.is\_leader = 1
GROUP BY
db\_name,
table\_name,
index\_name,
type;
SELECT
p.store\_id,
sum(flow\_bytes ),
count(1)
FROM
INFORMATION\__SCHEMA.tidb\_hot\_regions h
JOIN INFORMATION\__SCHEMA.tikv\_region\_peers p ON h.region\_id = p.region\_id
AND p.is\_leader = 1
GROUP BY
p.store\_id
ORDER BY
2 DESC;
select tidb\_decode\_plan();
```

33 TiFlash

贡献者：@我是咖啡哥

```
ALTER TABLE t\_test\_time\_type SET TIFLASH REPLICA 1;
SELECT \* FROM information\_schema.tiflash\_replica;
select \* from information\_schema.CLUSTER\_HARDWARE where type='tiflash' and DEVICE\_TYPE='disk' and
```

34 admin 命令

贡献者：@我是咖啡哥

```
admin show ddl jobs;
ADMIN CHECK TABLE t_test;
admin show slow
ADMIN SHOW TELEMETRY;
```

35 修改隔离参数

贡献者：@我是咖啡哥

● session 级别修改
Engine 隔离：默认：[“tikv”，“tidb”，“tiflash”] 由于 TiDB Dashboard 等组件需要读取一些存储于 TiDB 内存表区的系统表，因此建议实例级别 engine 配置中始终加入“tidb”engine。

```
set session tidb\_isolation\_read\_engines = 'tiflash,tidb';
```

```
set @@session.tidb\_isolation\_read\_engines = "tiflash,tidb";
```

● 手工 Hint

```
select /\*+ read\_from\_storage(tiflash\[table\_name]) \*/ ... from table\_name;
select /\*+ read\_from\_storage(tiflash\[alias\_a,alias\_b]) \*/ ... from table\_name\_1 as alias\_a,
set @@tidb\_allow\_mpp=1;
show config where name like '%oom%' and type='tidb';
admin show ddl;
```

36 排错-查看日志

贡献者：@我是咖啡哥

```
SELECT \* FROM INFORMATION\__SCHEMA.CLUSTER\_LOG t
WHERE time > '2022-08-09 00:00:00' AND time < '2022-08-10 00:00:00'
AND TYPE in ('tikv')
AND `LEVEL` = 'ERROR'
ORDER BY time desc;
```

37 查询所有节点所在 OS 的 CPU 当前使用率

贡献者：@人如其名


```
SELECT
  b.time,
  a.hostname,
  a.ip,
  a.types,
  b.cpu_used_percent
FROM
  (
    SELECT
      GROUP_CONCAT(TYPE) AS TYPES,
      SUBSTRING_INDEX(instance, ':', 1) AS ip,
      value AS hostname
    FROM
      information_schema.cluster_systeminfo
    WHERE
      name = 'kernel.hostname'
    GROUP BY
      ip,
      hostname
  ) a,
  (
    SELECT
      time,
      SUBSTRING_INDEX(instance, ':', 1) AS ip,
      (100 - value) AS cpu_used_percent
    FROM
      metrics_schema.node_cpu_usage
    WHERE
      MODE = 'idle'
      AND time = NOW()
  ) b
WHERE
  a.ip = b.ip
```

输出示例：

time	hostname	ip	types	cpu_us
2023-01-10 22:40:15.000000	localhost.localdomain	192.168.31.201	tidb,pd,tikv,tiflash	11.438
1 row in set (0.04 sec)				

说明：我这里所有类型组件只创建了有一个而且都在一个 os 上，所以只显示了一行。

38 清理 tidb 大量数据的脚本，实现删除百万级别以上的数据，而且不影响 tidb 正常使用

贡献者：@xingzhenxiang

```
date1=`date --date "7 days ago" +"%Y-%m-%d"`
delete_db_sql="delete from mysql_table where create_date_time< '$date1' limit 10000"
i=0
while ((++i)); do
a=`/bin/mysql -uroot -p123456 -A mysql_database -h127.0.0.1 --comments -e "${delete_db_sql}" -vvv|grep
if(($a<1)); then
break 1
fi
sleep 1
printf "%-4d" ${i})
```

🌟感谢以上 TiDBer 们贡献的 SQL 脚本～记得点赞收藏，可以随时在你的个人收藏夹里查看到~

未来我们将继续发布更多来自社区的精品内容，希望能为大家提供更多有价值的信息和经验。如果您有任何感兴趣的话题，可以在下方留言，我们会整理相关的资料与大家分享哦！

下载 TiDB 社区版

咨询 TiDB 企业版

免费试用 TiDB Cloud

适用于中国出海企业和开发者

SQL

TiDB

相关博客

高性能批处理

TiDB 7.5.0 LTS 高性能数据批处理方案

TiDB 7.5 LTS 版本正式 GA 了一个 IMPORT INTO 的功能，该功能将原本 tidb-lightning 的物理导入能力集成到 TiDB 计算节点上，使用一条 SQL 语句就可以完成大批量数据的导入，大幅简化了超大规模数据写入时的复杂度。

运维监控

通过 Prometheus 编写 TiDB 巡检脚本（脚本已开源，内附链接）

本文作者来自神州数码钛合金战队，介绍了利用 TiDB 的 Prometheus 来获取集群相关的各项数据，并且把告警值改低，以排查集群的隐患，提前规划，避免出现异常。

TiDB | TiKV | 容量弹性扩展

一篇文章彻底搞懂 TiDB 集群各种容量计算方式

本文讨论了 TiKV 的容量计算细节，希望阅读文章后大家能对 TiDB 集群的各种空间计算有清晰的认识。

查看全部博客 →

关于我们

- 公司概况
- 发展历程
- 新闻中心
- 市场活动
- 加入我们

隐私政策

- Cookie 政策
- 安全合规
- 版本支持策略
- 漏洞管理策略
- 网站使用条款

资源中心

- 社区
- TiDB 文档
- TiDB 6.x in Action
- 快速上手指南
- 社区问答–AskTUG
- 博客
- GitHub
- PingCAP Education
- 商标下载及使用

联系我们

- 商务咨询
- 400–6790–886
- 010–58400041
- info@pingcap.com

前台总机

010–53326356

媒体合作

pr@pingcap.com

PingCAP 公司

PingCAP 是业界领先的企业级开源分布式数据库企业，提供包括开源分布式数据库产品、解决方案与咨询、技术支持与培训认证服务，致力于为全球行业用户提供稳定高效、安全可靠、开放兼容的新型数据服务平台，解放企业生产力，加速企业数字化转型升级。



联系我们

友情链接 ▾