GreatSQL TPC-H 性能测试报告

GreatSQL TPC-H 性能测试报告 (2024 年 2 月28日)

GreatSQL 社区

【文档声明】

GreatSQL 社区提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。您应当通过 GreatSQL 社区网站或 GreatSQL 社区提供的其他授权通道下载、获取本文档,且仅能用于自身的合法合规的业务活动。本文档的内容视为 GreatSQL 社区的保密信息,您应当严格遵守保密义务;未经 GreatSQL 社区事先书面同意,您不得向任何第三方披露本手册内容或提供给任何第三方使用。

未经 GreatSQL 社区事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部,不得以任何方式或途径进行替换和宣传。

由于产品版本升级、调整或其他原因,本文档内容有可能变更。GreatSQL 社区保留在没有任何通知或者提示下对本文档的内容进行修改的权利,并在 GreatSQL 社区授权通道中不定期发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过 GreatSQL 社区授权渠道下载、获取最新版的用户文档。

本文档仅作为用户使用 GreatSQL 社区产品及服务的参考性指引。GreatSQL 社区在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但 GreatSQL 社区在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,GreatSQL 社区不承担任何法律责任。在任何情况下,GreatSQL社区均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使 GreatSQL 社区已被告知该等损失的可能性)。

GreatSQL 社区文档中所有内容,包括但不限于图片、架构设计、页面布局、文字描述,均由 GreatSQL 社区和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业 秘密等。非经 GreatSQL 社区和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开替 换、改变、散布、发行或公开发表 GreatSQL 社区网站、产品程序或内容。此外,未经 GreatSQL 社区 事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制 GreatSQL 社区的 名称(包括但不限于单独为或以组合形式包含"GreatSQL 社区"、"GreatSQL"等 GreatSQL 社区 和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名 称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别 GreatSQL 社区和/或其关联公司)。

如若发现本文档存在任何错误,请与 GreatSQL 社区取得直接联系。

GreatSQL社区官网: https://greatsql.cn。

1. 概述

本次测试针对GreatSQL数据库基于标准 TPC-H 场景的测试。

TPC-H(商业智能计算测试)是美国交易处理效能委员会(TPC,TransactionProcessing Performance Council)组织制定的用来模拟决策支持类应用的一个测试集。目前,学术界和工业界普遍采用 TPC-H 来评价决策支持技术方面应用的性能。这种商业测试可以全方位评测系统的整体商业计算综合能力,对厂商的要求更高,同时也具有普遍的商业实用意义,目前在银行信贷分析和信用卡分析、电信运营分析、税收分析、烟草行业决策分析中都有广泛的应用,TPC-H 查询包含八张数据表和22 条复杂 SQL 查询,大多数查询包含多表联接(JOIN)、子查询和聚合查询等。

GreatSQL数据库是一款**开源免费**数据库,可在普通硬件上满足金融级应用场景,具有**高可用、高性能、高兼容、高安全**等特性,可作为MySQL或Percona Server for MySQL的理想可选替换。

2. 测试环境信息

操作系统	OS: CentOS Linux release 7.9.2009 (Core) 内核: 3.10.0-1160.el7.x86_64
CPU	Intel(R) Xeon(R) Gold 6238 CPU @ 2.10GHz * 4
内存	251G

磁盘	INTEL SSDPE2KE032T8	
数据库	GreatSQL 8.0.32-25, Release 25, Revision 79f57097e3f	

服务器详细信息



1. 操作系统

\$ cat /etc/os-release NAME="CentOS Linux" VERSION="7 (Core)" ID="centos" ID LIKE="rhel fedora" VERSION ID="7" PRETTY NAME="CentOS Linux 7 (Core)" ANSI COLOR="0;31" CPE_NAME="cpe:/o:centos:centos:7" HOME_URL="https://www.centos.org/" BUG_REPORT_URL="https://bugs.centos.org/" CENTOS MANTISBT PROJECT="CentOS-7" CENTOS_MANTISBT_PROJECT_VERSION="7"

REDHAT SUPPORT PRODUCT="centos" REDHAT SUPPORT PRODUCT VERSION="7"

2. CPU

\$ lscpu

Architecture: x86 64

CPU op-mode(s): 32-bit, 64-bit

Byte Order: Little Endian

176 CPU(s):

On-line CPU(s) list: 0-175

Thread(s) per core: 2 Core(s) per socket: 22

Socket(s): 4

NUMA node(s): 4

Vendor ID: GenuineIntel

CPU family: 6 Model: 85

Model name: Intel(R) Xeon(R) Gold 6238 CPU @ 2.10GHz

Stepping: 7

CPU MHz: 1000.012

CPU max MHz: 3700.0000 CPU min MHz: 1000.0000

BogoMIPS: 4200.00

Virtualization: VT-x

L1d cache: 32K

L1i cache: 32K

L2 cache: 1024K

L3 cache: 30976K

NUMA node0 CPU(s):

0,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60,64,68,72,76,80,84,88,92,96,100,104,108,1 12,116,120,124,128,132,136,140,144,148,152,156,160,164,168,172

NUMA node1 CPU(s):

1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61,65,69,73,77,81,85,89,93,97,101,105,109,1 13,117,121,125,129,133,137,141,145,149,153,157,161,165,169,173

NUMA node2 CPU(s):

2,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62,66,70,74,78,82,86,90,94,98,102,106,110, 114,118,122,126,130,134,138,142,146,150,154,158,162,166,170,174

NUMA node3 CPU(s):

3,7,11,15,19,23,27,31,35,39,43,47,51,55,59,63,67,71,75,79,83,87,91,95,99,103,107,111, 115,119,123,127,131,135,139,143,147,151,155,159,163,167,171,175

Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch epb cat_l3 cdp_l3 invpcid_single intel_ppin intel_pt ssbd mba ibrs ibpb stibp ibrs_enhanced tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm cqm mpx rdt_a avx512f avx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local dtherm ida arat pln pts pku ospke avx512_vnni md_clear spec_ctrl intel_stibp flush_l1d arch_capabilities

3. 内存

\$ free -ht

total used free shared buff/cache available

Mem: 251G 167G 22G 7.2M 61G 82G

Swap: 4.0G 1.1G 2.9G Total: 255G 168G 24G

4. 磁盘

磁盘设备型号

S nyme list

Node SN Model Namespace Usage Format

FW Rev

/dev/nvme0n1 PHLN018200FD3P2BGN INTEL SSDPE2KE032T8 1

3.20 TB / 3.20 TB 512 B + 0 B VDV10152

磁盘挂载参数、文件系统

\$ df -hT | grep ssd

/dev/nvme0n1 xfs 3.0T 1.5T 1.5T 49% /ssd2

NVMe SSD设备简单测速

\$ dd oflag=direct if=/dev/zero of=./zero bs=1M count=20480

20480+0 records in

20480+0 records out

21474836480 bytes (21 GB) copied, 8.69131 s, 2.5 GB/s

🍟 提示:在下面运行TPC-H测试时,设置了Rapid引擎最大可使用的内存及线程数。

3. 测试表结构和数据量

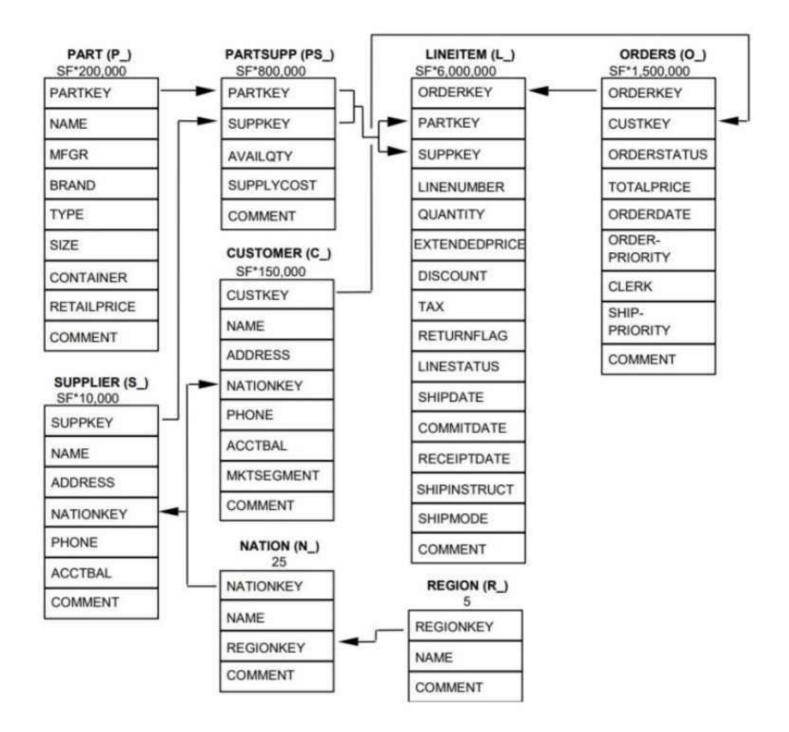
各表数据量对比:

表名	TPC-H SF100数据量	TPC-H SF300数据量	备注
region	5	5	地区信息
nation	25	25	国家表
supplier	1000000	3000000	供应商信息
part	20000000	60000000	零件表
customer	15000000	45000000	消费者表
partsupp	80000000	240000000	配件供应表
orders	150000000	450000000	订单表
lineitem	600037902	1799989091	订单明细表

Rapid引擎表空间压缩率:

库名	InnoDB表空间文件总大小	Rapid引擎表空间总大小	压缩率
TPC-H SF100	184570593436	28728373248	6.42
TPC-H SF300	591644573888	74334864443	7.96

各表结构关系如下图所示:



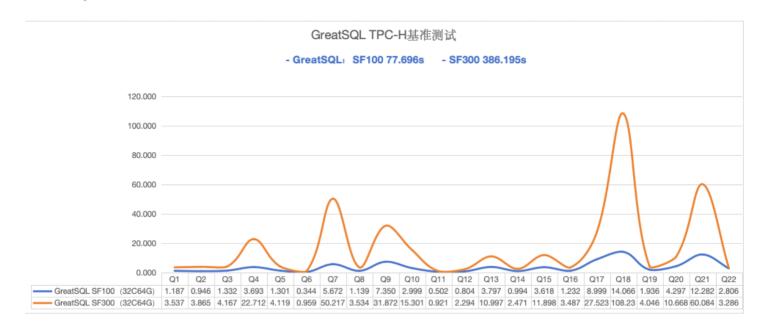
4. 测试结果

GreatSQL 8.0.32-25中,采用全新的Rapid存储引擎,使得其在TPC-H性能测试中表现大大优于此前的其他版本,也大大优于MySQL社区版、Percona Server MySQL、MariaDB等数据库。

在TPC-H SF100场景下,运行完全部22个TPC-H查询SQL总耗时为**79.28秒**。在TPC-H SF300场景下,运行完全部22个TPC-H查询SQL总耗时为**386.195秒**。

每条SQL详细耗时如下:

	GreatSQL TPC-H SF100 (32C64G)耗时(秒)	GreatSQL TPC-H SF300 (32C64G)耗时(秒)
Q1	1.184	3.537
Q2	0.924	3.865
Q3	1.324	4.167
Q4	3.678	22.712
Q5	1.287	4.119
Q6	0.344	0.959
Q7	5.48	50.217
Q8	1.13	3.534
Q9	7.311	31.872
Q10	2.885	15.301
Q11	0.477	0.921
Q12	0.799	2.294
Q13	3.758	10.997
Q14	0.966	2.471
Q15	2.831	11.898
Q16	1.194	3.487
Q17	8.537	27.523
Q18	13.007	108.237
Q19	1.892	4.046
Q20	4.21	10.668
Q21	11.965	60.084
Q22	2.513	3.286
总耗时	77.696	386.195



5. 测试步骤

5.1 安装 GreatSQL

请参考GreatSQL手册内容:安装指南,完成GreatSQL安装。

5.2 生成 TPC-H 测试数据

请参考GreatSQL手册内容: TPC-H性能测试,完成TPC-H工具编译安装。

运行 TPC-H dbgen 工具,生成数据文件,一共会生成 8 个表对应的 tbl 数据文件,例如:

```
1 $ ./dbgen -vf -s 100
2 ...
3
4 $ ls -l *tbl
5 -rw-r--r-- 1 root root 2463490271 Sep 26 09:20 customer.tbl
6 -rw-r--r-- 1 root root 79579694556 Sep 26 09:20 lineitem.tbl
7 -rw-r--r-- 1 root root 2224 Sep 26 09:20 nation.tbl
8 -rw-r--r-- 1 root root 17793116301 Sep 26 09:20 orders.tbl
9 -rw-r--r-- 1 root root 12209211160 Sep 26 09:20 partsupp.tbl
10 -rw-r--r-- 1 root root 2453234158 Sep 26 09:20 part.tbl
11 -rw-r--r-- 1 root root 389 Sep 26 09:20 region.tbl
12 -rw-r--r-- 1 root root 142869803 Sep 26 09:20 supplier.tbl
```

也可以参考 duckdb_dbgen.py 脚本做法,利用duckdb并行生成测试数据。

5.3 创建 TPC-H 测试数据库表并导入数据

参考GreatSQL社区提供的TPC-H数据库表初始化脚本: tpch-create-table.sql,完成TPC-H测试数据库表创建。

```
1 $ mysql -f < tpch-create-table.sql</pre>
2 $ mysqlshow tpch100
3 Database: tpch100
4 +----+
5 | Tables |
6 +----+
7 | customer |
8 | lineitem |
9 | nation |
10 | orders |
11 | part |
12 | partsupp |
13 | region |
14 | revenue0 |
15 | supplier |
16 +----+
```

利用GreatSQL的 parallel load data特性并行导入TPC-H测试数据。

需要先修改GreatSQL选项 secure_file_priv 设置,指向上述 workdir 所在目录,重启GreatSQL 使之生效。

参考GreatSQL社区提供的并发导入脚本:load-data-parallel.sh,完成数据导入。



提示:运行LOAD DATA导入数据时,可能会在 tmpdir 产生临时文件,因此要保证 tmpdir 有足够的剩余可用磁盘空间。

5.4 确认Rapid引擎设置,并加载数据到secondary engine

数据导入完成后,在开始运行TPC-H测试前,需要先将测试数据加载到secondary engine引擎中。

先执行下面命令,动态修改Rapid引擎最大可使用内存,其余相关选项均为默认值:

```
1 greatsql> SET GLOBAL rapid_memory_limit = 68719476736;
2 greatsql> SET GLOBAL rapid_worker_threads = 32;
```

之后,执行以下命令加载测试数据到secondary engine:

```
1 greatsql> alter table customer secondary_load;
2 alter table lineitem secondary_load;
3 alter table nation secondary_load;
4 alter table orders secondary_load;
5 alter table part secondary_load;
6 alter table partsupp secondary_load;
7 alter table region secondary_load;
8 alter table supplier secondary_load;
```

这个过程需要一定时间,请耐心等待。

5.5 执行 TPC-H 测试

参考GreatSQL社区提供的TPC-H性能测试脚本,完成测试,并记录各个SQL的耗时。

该测试脚本大概工作模式如下:

- 1. 先执行22个查询SQL,进行数据预热,每条SQL各执行2次。
- 2. 再分别执行22个查询SQL,每个SQL各执行3次。
- 3. 每次执行SQL都会记录其起止时间,及其耗时,如下面例所示:

```
1 [2023-09-27 01:38:45] BEGIN RUN TPC-H Q1 1 times
2 [2023-09-27 01:38:46] TPC-H Q1 END, COST: 1.301s
3
```

```
4
5 [2023-09-27 01:38:46] BEGIN RUN TPC-H Q1 2 times
6 [2023-09-27 01:38:47] TPC-H Q1 END, COST: 0.787s
```

上述结果中的 COST: 1.301s ,即为本SQL的运行耗时: 1.301秒。

6. 附录

6.1 创建测试表DDL

```
1 -- DROP DATABASE IF EXISTS tpch;
 2 -- CREATE DATABASE IF NOT EXISTS tpch DEFAULT CHARACTER SET latin1;
3 -- USE tpch;
5 drop table if exists nation;
6 create table nation ( n_nationkey integer not null,
7
                                                char(25) not null,
                                   n_name
8
                                   n_regionkey integer not null,
9
                                   n_comment
                                                varchar(152),
                                   primary key(n_nationkey),
10
                                   key nation_fk1 (n_regionkey) )
11
   secondary_engine = rapid;
12
13 drop table if exists region;
14 create table region ( r_regionkey integer not null,
                                              char(25) not null,
15
                                   r_name
16
                                   r_comment
                                                varchar(152),
                                   primary key(r_regionkey) ) secondary_engine =
17
   rapid;
18
19 drop table if exists part;
20 create table part (p_partkey
                                      integer not null,
                                                varchar(55) not null,
                                   p_name
21
                                   p_mfgr
22
                                                char(25) not null,
23
                                   p_brand
                                                char(10) not null,
24
                                   p_type
                                                varchar(25) not null,
                                                 integer not null,
25
                                   p_size
                                   p_container char(10) not null,
26
                                   p_retailprice decimal(15,2) not null,
27
                                               varchar(23) not null,
28
                                   p_comment
29
                                   primary key(p_partkey) ) secondary_engine =
   rapid;
```

```
30
31 drop table if exists supplier;
32 create table supplier ( s_suppkey
                                          integer not null,
                                                  char(25) not null,
33
                                    s_name
                                                  varchar(40) not null,
                                    s_address
34
35
                                    s_nationkey
                                                  integer not null,
                                                  char(15) not null,
36
                                    s_phone
                                    s_acctbal
                                                  decimal(15,2) not null,
37
38
                                    s_comment
                                                  varchar(101) not null,
39
                                    primary key(s_suppkey),
                                    key supplier_fk1 (s_nationkey) )
40
   secondary_engine = rapid;
41
42 drop table if exists partsupp;
43 create table partsupp (ps_partkey
                                           integer not null,
44
                                    ps_suppkey
                                                   integer not null,
                                                   integer not null,
45
                                    ps_availqty
46
                                    ps_supplycost decimal(15,2) not null,
47
                                    ps_comment
                                                   varchar(199) not null,
                                    primary key(ps_partkey,ps_suppkey),
48
49
                                    key partsupp_fk1 (ps_suppkey),
                                    key partsupp_fk2 (ps_partkey) )
50
   secondary_engine = rapid;
51
52
53 drop table if exists customer;
54 create table customer ( c_custkey
                                          integer not null,
                                                  varchar(25) not null,
55
                                    c_name
                                                  varchar(40) not null,
56
                                    c_address
                                    c_nationkey
                                                  integer not null,
57
58
                                    c_phone
                                                  char(15) not null,
                                    c_acctbal
                                                  decimal(15,2)
59
                                                                  not null,
                                    c_mktsegment char(10) not null,
60
                                    c_comment
                                                  varchar(117) not null,
61
                                    primary key(c_custkey),
62
63
                                    key customer_fk1 (c_nationkey) )
   secondary_engine = rapid;
64
65 drop table if exists orders;
66 create table orders ( o_orderkey
                                            integer not null,
                                                     integer not null,
67
                                    o_custkey
                                    o_orderstatus
                                                     char(1) not null,
68
                                    o_totalprice
                                                     decimal(15,2) not null,
69
                                    o_orderdate
                                                     date not null,
70
71
                                    o_orderpriority char(15) not null,
72
                                    o clerk
                                                     char(15) not null,
                                    o_shippriority
73
                                                     integer not null,
```

```
74
                                    o_comment
                                                     varchar(79) not null,
75
                                    primary key(o_orderkey),
                                    key orders_fk1 (o_custkey) ) secondary_engine
76
   = rapid;
77
78 drop table if exists lineitem;
79 create table lineitem ( l_orderkey
                                          integer not null,
                                    l_partkey
                                                  integer not null,
80
81
                                    l suppkey
                                                  integer not null,
82
                                    l_linenumber integer not null,
83
                                    l_quantity
                                                  decimal(15,2) not null,
                                    l_extendedprice decimal(15,2) not null,
84
                                    l_discount decimal(15,2) not null,
85
                                    l tax
                                                  decimal(15,2) not null,
86
                                    l_returnflag char(1) not null,
87
88
                                    l_linestatus char(1) not null,
                                    l_shipdate
                                                  date not null,
89
90
                                    l_commitdate date not null,
                                    l_receiptdate date not null,
91
                                    l_shipinstruct char(25) not null,
92
93
                                    l_shipmode
                                                  char(10) not null,
                                                   varchar(44) not null,
                                    l_comment
94
                                    primary key(l_orderkey,l_linenumber),
95
96
                                    key lineitem_fk1 (l_orderkey) ,
97
                                    key lineitem_fk2 (l_partkey,l_suppkey) )
   secondary_engine = rapid;
```

6.2 22条TPC-H测试SQL

```
1 -- tpch_queries_1.sql
 2 SELECT /*+ SET_VAR(use_secondary_engine=1)
   SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q1 */
 3
       l_returnflag,
 4
       l_linestatus,
       sum(l_quantity) AS sum_qty,
 5
       sum(l_extendedprice) AS sum_base_price,
 6
 7
       sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
       sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
 8
 9
       avg(l_quantity) AS avg_qty,
       avg(l_extendedprice) AS avg_price,
10
11
       avg(l_discount) AS avg_disc,
       count(*) AS count_order
12
13 FROM
       lineitem
14
15 WHERE
```

```
16
       l_shipdate <= CAST('1998-09-02' AS date)</pre>
17 GROUP BY
       l_returnflag,
18
       l_linestatus
19
20 ORDER BY
21
       l_returnflag,
       l_linestatus;
22
23
24
25
26
27 -- tpch_queries_2.sql
28 SELECT /*+ SET_VAR(use_secondary_engine=1)
   SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q2 */
29
       s_acctbal,
30
       s_name,
31
       n_name,
32
       p_partkey,
33
       p_mfgr,
34
       s_address,
35
       s_phone,
       s_comment
36
37 FROM
38
       part,
39
       supplier,
40
       partsupp,
       nation,
41
42
       region
43 WHERE
       p_partkey = ps_partkey
44
45
       AND s_suppkey = ps_suppkey
       AND p_{size} = 15
46
47
       AND p_type LIKE '%BRASS'
48
       AND s_nationkey = n_nationkey
49
       AND n_regionkey = r_regionkey
50
       AND r_name = 'EUROPE'
       AND ps_supplycost = (
51
            SELECT
52
                min(ps_supplycost)
53
            FROM
54
55
                partsupp,
                supplier,
56
                nation,
57
                region
58
59
            WHERE
60
                p_partkey = ps_partkey
61
                AND s_suppkey = ps_suppkey
```

```
62
                 AND s_nationkey = n_nationkey
 63
                 AND n_regionkey = r_regionkey
                 AND r_name = 'EUROPE')
 64
 65 ORDER BY
        s_acctbal DESC,
 66
 67
        n_name,
 68
        s_name,
 69
        p_partkey
 70 LIMIT 100;
 71
 72
 73
 74
 75 -- tpch_queries_3.sql
 76 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q3 */
77
        l_orderkey,
 78
        sum(l_extendedprice * (1 - l_discount)) AS revenue,
 79
        o_orderdate,
        o_shippriority
 80
 81 FROM
 82
        customer,
 83
        orders,
        lineitem
 84
 85 WHERE
        c_mktsegment = 'BUILDING'
 86
        AND c_custkey = o_custkey
 87
 88
        AND l_orderkey = o_orderkey
        AND o_orderdate < CAST('1995-03-15' AS date)
 89
        AND l_shipdate > CAST('1995-03-15' AS date)
 90
 91 GROUP BY
        l_orderkey,
 92
        o_orderdate,
 93
        o_shippriority
 94
 95 ORDER BY
 96
        revenue DESC,
        o_orderdate
 97
    LIMIT 10;
 98
 99
100
101
102
103 -- tpch_queries_4.sql
104 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q4 */
105
        o_orderpriority,
        count(*) AS order_count
106
```

```
107 FROM
108
        orders
109 WHERE
        o_orderdate >= CAST('1993-07-01' AS date)
110
        AND o_orderdate < CAST('1993-10-01' AS date)
111
        AND EXISTS (
112
            SELECT
113
114
                 *
115
             FROM
                 lineitem
116
            WHERE
117
                 l_orderkey = o_orderkey
118
                 AND l_commitdate < l_receiptdate)
119
120 GROUP BY
        o_orderpriority
121
122 ORDER BY
        o_orderpriority;
123
124
125
126
127
128 -- tpch_queries_5.sql
129 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q5 */
130
        n_name,
        sum(l_extendedprice * (1 - l_discount)) AS revenue
131
132 FROM
133
        customer,
134
        orders,
135
        lineitem,
136
        supplier,
137
        nation,
138
        region
139 WHERE
140
        c_custkey = o_custkey
141
        AND l_orderkey = o_orderkey
142
        AND l_{suppkey} = s_{suppkey}
        AND c_nationkey = s_nationkey
143
        AND s_nationkey = n_nationkey
144
        AND n_regionkey = r_regionkey
145
        AND r_name = 'ASIA'
146
        AND o_orderdate >= CAST('1994-01-01' AS date)
147
        AND o_orderdate < CAST('1995-01-01' AS date)
148
149 GROUP BY
150
        n_name
151 ORDER BY
152
        revenue DESC;
```

```
153
154
155
156
157 -- tpch_queries_6.sql
158 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET VAR(secondary engine cost threshold=0) */ /*+ Q6 */
        sum(l_extendedprice * l_discount) AS revenue
159
160 FROM
161
        lineitem
162 WHERE
        l_shipdate >= CAST('1994-01-01' AS date)
163
164
        AND l_shipdate < CAST('1995-01-01' AS date)
        AND l_discount BETWEEN 0.05
165
        AND 0.07
166
167
        AND l_quantity < 24;
168
169
170
171
172 -- tpch_queries_7.sql
173 SELECT /*+ SET VAR(use secondary engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q7 */
174
        supp_nation,
        cust_nation,
175
176
        l_year,
        sum(volume) AS revenue
177
178 FROM (
        SELECT
179
            n1.n_name AS supp_nation,
180
181
            n2.n_name AS cust_nation,
            extract(year FROM l_shipdate) AS l_year,
182
            l_extendedprice * (1 - l_discount) AS volume
183
184
        FROM
185
            supplier,
186
            lineitem,
187
            orders,
188
            customer,
            nation n1,
189
            nation n2
190
        WHERE
191
            s_suppkey = l_suppkey
192
193
            AND o_orderkey = l_orderkey
            AND c_custkey = o_custkey
194
195
            AND s_nationkey = n1.n_nationkey
            AND c_nationkey = n2.n_nationkey
196
            AND ((n1.n_name = 'FRANCE'
197
```

```
198
                     AND n2.n_name = 'GERMANY')
199
                 OR (n1.n_name = 'GERMANY'
                     AND n2.n_name = 'FRANCE'))
200
            AND l_shipdate BETWEEN CAST('1995-01-01' AS date)
201
            AND CAST('1996-12-31' AS date)) AS shipping
202
203 GROUP BY
204
        supp_nation,
205
        cust_nation,
206
        l_year
207 ORDER BY
208
        supp_nation,
        cust_nation,
209
        l_year;
210
211
212
213
214
215 -- tpch_queries_8.sql
216 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q8 */
217
        o_year,
        sum(
218
            CASE WHEN nation = 'BRAZIL' THEN
219
220
                 volume
            ELSE
221
222
223
             END) / sum(volume) AS mkt_share
224 FROM (
        SELECT
225
             extract(year FROM o_orderdate) AS o_year,
226
            l_extendedprice * (1 - l_discount) AS volume,
227
            n2.n name AS nation
228
        FROM
229
230
            part,
231
             supplier,
232
            lineitem,
233
            orders,
234
            customer,
            nation n1,
235
236
            nation n2,
237
             region
238
        WHERE
239
            p_partkey = l_partkey
240
            AND s_{suppkey} = l_{suppkey}
241
            AND l_orderkey = o_orderkey
242
            AND o_custkey = c_custkey
            AND c_nationkey = n1.n_nationkey
243
```

```
244
            AND n1.n_regionkey = r_regionkey
245
            AND r_name = 'AMERICA'
            AND s_nationkey = n2.n_nationkey
246
            AND o_orderdate BETWEEN CAST('1995-01-01' AS date)
247
            AND CAST('1996-12-31' AS date)
248
249
            AND p_type = 'ECONOMY ANODIZED STEEL') AS all_nations
250 GROUP BY
251
        o_year
252 ORDER BY
253
        o_year;
254
255
256
257
258 -- tpch_queries_9.sql
259 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q9 */
260
        nation,
261
        o_year,
262
        sum(amount) AS sum_profit
263 FROM (
        SELECT
264
265
            n_name AS nation,
266
            extract(year FROM o_orderdate) AS o_year,
267
            l_extendedprice * (1 - l_discount) - ps_supplycost * l_quantity AS
    amount
268
        FROM
269
            part,
270
            supplier,
271
            lineitem,
272
            partsupp,
273
            orders,
            nation
274
        WHERE
275
276
            s\_suppkey = l\_suppkey
277
            AND ps_suppkey = l_suppkey
278
            AND ps_partkey = l_partkey
            AND p_partkey = l_partkey
279
            AND o orderkey = l orderkey
280
            AND s_nationkey = n_nationkey
281
            AND p_name LIKE '%green%') AS profit
282
283 GROUP BY
284
        nation,
285
        o_year
286 ORDER BY
287
        nation,
288
        o_year DESC;
```

```
289
290
291
292
293 -- tpch_queries_10.sql
294 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q10 */
295
        c_custkey,
296
        c_name,
297
        sum(l_extendedprice * (1 - l_discount)) AS revenue,
298
        c_acctbal,
299
        n_name,
        c_address,
300
301
        c_phone,
        c_comment
302
303 FROM
304
        customer,
305
        orders,
306
        lineitem,
        nation
307
308 WHERE
        c_custkey = o_custkey
309
        AND l_orderkey = o_orderkey
310
311
        AND o orderdate >= CAST('1993-10-01' AS date)
        AND o_orderdate < CAST('1994-01-01' AS date)
312
        AND l_returnflag = 'R'
313
314
        AND c_nationkey = n_nationkey
315 GROUP BY
316
        c_custkey,
317
        c_name,
318
        c_acctbal,
319
        c_phone,
320
        n_name,
        c_address,
321
322
        c_comment
323 ORDER BY
        revenue DESC
324
325 LIMIT 20;
326
327
328
329
330 -- tpch_queries_11.sql
331 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q11 */
332
        ps_partkey,
333
        sum(ps_supplycost * ps_availqty) AS value
```

```
334 FROM
        partsupp,
335
336
        supplier,
        nation
337
338 WHERE
339
        ps_suppkey = s_suppkey
        AND s_nationkey = n_nationkey
340
        AND n_name = 'GERMANY'
341
342 GROUP BY
        ps_partkey
343
344 HAVING
345
        sum(ps_supplycost * ps_availqty) > (
            SELECT
346
                 sum(ps_supplycost * ps_availqty) * 0.0001000000
347
            FROM
348
349
                partsupp,
350
                 supplier,
351
                nation
352
            WHERE
353
                ps_suppkey = s_suppkey
354
                AND s_nationkey = n_nationkey
                AND n_name = 'GERMANY')
355
356 ORDER BY
357
        value DESC;
358
359
360
361
362 -- tpch_queries_12.sql
363 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q12 */
364
        l_shipmode,
        sum(
365
            CASE WHEN o_orderpriority = '1-URGENT'
366
367
                OR o_orderpriority = '2-HIGH' THEN
368
                1
            ELSE
369
370
            END) AS high_line_count,
371
372
        sum(
            CASE WHEN o_orderpriority <> '1-URGENT'
373
374
                AND o_orderpriority <> '2-HIGH' THEN
375
                 1
            ELSE
376
377
378
            END) AS low_line_count
379 FROM
```

```
380
        orders,
381
        lineitem
382 WHERE
        o_orderkey = l_orderkey
383
        AND l_shipmode IN ('MAIL', 'SHIP')
384
        AND l_commitdate < l_receiptdate
385
386
        AND l_shipdate < l_commitdate
        AND l_receiptdate >= CAST('1994-01-01' AS date)
387
388
        AND l_receiptdate < CAST('1995-01-01' AS date)
389 GROUP BY
390
        l_shipmode
391 ORDER BY
        l_shipmode;
392
393
394
395
396
397 -- tpch_queries_13.sql
398 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q13 */
399
        c_count,
        count(*) AS custdist
400
401 FROM (
402
        SELECT
403
            c_custkey,
404
            count(o_orderkey)
        FROM
405
406
            customer
        LEFT OUTER JOIN orders ON c_custkey = o_custkey
407
        AND o_comment NOT LIKE '%special%requests%'
408
409 GROUP BY
        c_custkey) AS c_orders (c_custkey,
410
411
            c_count)
412 GROUP BY
413
        c_count
414 ORDER BY
415
        custdist DESC,
416
        c_count DESC;
417
418
419
420
421 -- tpch_queries_14.sql
422 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q14 */
423
        100.00 * sum(
424
            CASE WHEN p_type LIKE 'PROMO%' THEN
```

```
425
                 l_extendedprice * (1 - l_discount)
            ELSE
426
427
             END) / sum(l_extendedprice * (1 - l_discount)) AS promo_revenue
428
429 FROM
430
        lineitem,
431
        part
432 WHERE
433
        l_partkey = p_partkey
434
        AND l_shipdate >= date '1995-09-01'
        AND l_shipdate < CAST('1995-10-01' AS date);
435
436
437
438
439
440 -- tpch_queries_15.sql
441 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q15 */
442
        s_suppkey,
443
        s_name,
444
        s_address,
        s_phone,
445
        total_revenue
446
447 FROM
448
        supplier,
449
        (
450
            SELECT
451
                 l_suppkey AS supplier_no,
                 sum(l_extendedprice * (1 - l_discount)) AS total_revenue
452
            FROM
453
454
                lineitem
            WHFRF
455
                 l_shipdate >= CAST('1996-01-01' AS date)
456
                AND l_shipdate < CAST('1996-04-01' AS date)
457
458
            GROUP BY
459
                 supplier_no) revenue0
460 WHERE
        s_suppkey = supplier_no
461
        AND total_revenue = (
462
            SELECT
463
464
                max(total_revenue)
            FROM (
465
466
                 SELECT
                     l_suppkey AS supplier_no,
467
468
                     sum(l_extendedprice * (1 - l_discount)) AS total_revenue
469
                 FROM
                     lineitem
470
```

```
471
                 WHERE
                     l_shipdate >= CAST('1996-01-01' AS date)
472
                     AND l_shipdate < CAST('1996-04-01' AS date)
473
474
                 GROUP BY
                     supplier_no) revenue1)
475
476 ORDER BY
477
        s_suppkey;
478
479
480
481
482 -- tpch_queries_16.sql
483 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q16 */
484
        p_brand,
485
        p_type,
486
        p_size,
487
        count(DISTINCT ps_suppkey) AS supplier_cnt
488 FROM
489
        partsupp,
490
        part
491 WHERE
492
        p_partkey = ps_partkey
493
        AND p_brand <> 'Brand#45'
        AND p_type NOT LIKE 'MEDIUM POLISHED%'
494
        AND p_size IN (49, 14, 23, 45, 19, 3, 36, 9)
495
496
        AND ps_suppkey NOT IN (
            SELECT
497
498
                 s_suppkey
            FROM
499
500
                 supplier
501
            WHERE
                 s_comment LIKE '%Customer%Complaints%')
502
503 GROUP BY
504
        p_brand,
505
        p_type,
506
        p_size
507 ORDER BY
        supplier_cnt DESC,
508
509
        p_brand,
510
        p_type,
511
        p_size;
512
513
514
515
516 -- tpch_queries_17.sql
```

```
517 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q17 */
        sum(l_extendedprice) / 7.0 AS avg_yearly
518
519 FROM
520
        lineitem,
521
        part
522 WHERE
523
        p_partkey = l_partkey
524
        AND p_brand = 'Brand#23'
525
        AND p_container = 'MED BOX'
        AND l_quantity < (
526
            SELECT
527
                 0.2 * avg(l_quantity)
528
            FROM
529
530
                lineitem
            WHERE
531
532
                l_partkey = p_partkey);
533
534
535
536
537 -- tpch queries 18.sql
538 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q18 */
539
        c_name,
540
        c_custkey,
541
        o_orderkey,
542
        o_orderdate,
543
        o_totalprice,
        sum(l_quantity)
544
545 FROM
546
        customer,
547
        orders,
548
        lineitem
549 WHERE
550
        o_orderkey IN (
551
            SELECT
552
                l_orderkey
            FROM
553
554
                lineitem
            GROUP BY
555
556
                l_orderkey
            HAVING
557
558
                 sum(l_quantity) > 300)
559
        AND c_custkey = o_custkey
560
        AND o_orderkey = l_orderkey
561 GROUP BY
```

```
562
        c_name,
563
        c_custkey,
564
        o_orderkey,
        o_orderdate,
565
566
        o_totalprice
567 ORDER BY
        o_totalprice DESC,
568
569
        o_orderdate
570 LIMIT 100;
571
572
573
574
575 -- tpch_queries_19.sql
576 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q19 */
        sum(l_extendedprice * (1 - l_discount)) AS revenue
577
578 FROM
579
        lineitem,
580
        part
581 WHERE (p_partkey = l_partkey
        AND p_brand = 'Brand#12'
582
        AND p_container IN ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
583
584
        AND l_quantity >= 1
585
        AND l_quantity <= 1 + 10
        AND p_size BETWEEN 1 AND 5
586
        AND l_shipmode IN ('AIR', 'AIR REG')
587
        AND l shipinstruct = 'DELIVER IN PERSON')
588
        OR (p_partkey = l_partkey
589
            AND p_brand = 'Brand#23'
590
            AND p_container IN ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
591
            AND l_quantity >= 10
592
            AND l_quantity <= 10 + 10
593
            AND p_size BETWEEN 1 AND 10
594
595
            AND l_shipmode IN ('AIR', 'AIR REG')
596
            AND l_shipinstruct = 'DELIVER IN PERSON')
597
        OR (p_partkey = l_partkey
            AND p_brand = 'Brand#34'
598
            AND p container IN ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
599
            AND l_quantity >= 20
600
            AND l quantity <= 20 + 10
601
            AND p_size BETWEEN 1 AND 15
602
            AND l_shipmode IN ('AIR', 'AIR REG')
603
            AND l_shipinstruct = 'DELIVER IN PERSON');
604
605
606
607
```

```
608
609 -- tpch_queries_20.sql
610 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q20 */
611
        s_name,
612
        s_address
613 FROM
614
        supplier,
615
        nation
616 WHERE
617
        s_suppkey IN (
            SELECT
618
619
                ps_suppkey
620
            FROM
621
                partsupp
622
            WHERE
623
                ps_partkey IN (
624
                     SELECT
625
                         p_partkey
626
                     FROM
627
                         part
                     WHERE
628
                         p_name LIKE 'forest%')
629
630
                     AND ps_availqty > (
631
                         SELECT
632
                             0.5 * sum(l_quantity)
633
                         FROM
                             lineitem
634
                         WHERE
635
                             l_partkey = ps_partkey
636
637
                             AND l_suppkey = ps_suppkey
                             AND l_shipdate >= CAST('1994-01-01' AS date)
638
                             AND l_shipdate < CAST('1995-01-01' AS date)))
639
640
                AND s_nationkey = n_nationkey
641
                AND n_n = 'CANADA'
642
            ORDER BY
643
                s_name;
644
645
646
647
648 -- tpch_queries_21.sql
649 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q21 */
650
        s_name,
651
        count(*) AS numwait
652 FROM
```

```
653
        supplier,
        lineitem l1,
654
        orders,
655
        nation
656
657 WHERE
        s_suppkey = l1.l_suppkey
658
659
        AND o_orderkey = l1.l_orderkey
        AND o_orderstatus = 'F'
660
661
        AND l1.l_receiptdate > l1.l_commitdate
662
        AND EXISTS (
663
            SELECT
664
            FROM
665
                 lineitem l2
666
            WHERE
667
668
                 l2.l_orderkey = l1.l_orderkey
                AND l2.l_suppkey <> l1.l_suppkey)
669
670
        AND NOT EXISTS (
671
            SELECT
672
673
            FROM
                 lineitem 13
674
            WHERE
675
                 l3.l_orderkey = l1.l_orderkey
676
                 AND l3.l_suppkey <> l1.l_suppkey
677
                 AND l3.l_receiptdate > l3.l_commitdate)
678
        AND s_nationkey = n_nationkey
679
        AND n_name = 'SAUDI ARABIA'
680
681 GROUP BY
682
        s_name
683 ORDER BY
684
        numwait DESC,
685
        s_name
686 LIMIT 100;
687
688
689
690
691 -- tpch_queries_22.sql
692 SELECT /*+ SET_VAR(use_secondary_engine=1)
    SET_VAR(secondary_engine_cost_threshold=0) */ /*+ Q22 */
693
        cntrycode,
        count(*) AS numcust,
694
        sum(c_acctbal) AS totacctbal
695
696 FROM (
697
        SELECT
698
             substring(c_phone FROM 1 FOR 2) AS cntrycode,
```

```
699
            c_acctbal
        FROM
700
701
            customer
702
        WHERE
             substring(c_phone FROM 1 FOR 2) IN ('13', '31', '23', '29', '30',
703
    '18', '17')
704
            AND c_acctbal > (
705
                 SELECT
706
                     avg(c_acctbal)
707
                 FROM
708
                     customer
                WHERE
709
                     c_acctbal > 0.00
710
                     AND substring(c_phone FROM 1 FOR 2) IN ('13', '31', '23',
711
    '29', '30', '18', '17'))
                AND NOT EXISTS (
712
                     SELECT
713
714
715
                     FROM
716
                         orders
                     WHERE
717
718
                         o_custkey = c_custkey)) AS custsale
719 GROUP BY
720
        cntrycode
721 ORDER BY
        cntrycode;
722
```

6.3 参考资料

- TPC-H官网 http://www.tpc.org/tpch
- GreatSQL安装指南 https://greatsql.cn/docs/8032-25/user-manual/4-install-guide/0-install-guide.html
- TPC-H性能测试指南 https://greatsql.cn/docs/8032-25/user-manual/10-optimze/3-2-benchmark-tpch.html
- TPC-H测试建表DDL及查询SQL https://gitee.com/GreatSQL/GreatSQL-Doc/tree/master/tpch/3.0.1
- duckdb_dbgen.py脚本 https://gitee.com/GreatSQL/GreatSQL-Doc/blob/master/tpch/3.0.1/duckdb_dbgen.py