

TPC-H Benchmark Set

TPC-H Benchmark

TPC-H is an ad-hoc and decision support benchmark. Some of queries are available in the current Tajo. You can download the TPC-H data generator from [here](#).

DDL for TPC-H datasets

TPC-H benchmark provides 8 table datasets. The below DDL statements is for them.

```
create external table supplier (  
    S_SUPPKEY bigint,  
    S_NAME text,  
    S_ADDRESS text,  
    S_NATIONKEY bigint,  
    S_PHONE text,  
    S_ACCTBAL double,  
    S_COMMENT text)  
using text with ('text.delimiter'='|') location 'hdfs://x/y';  
  
create external table lineitem (  
    L_ORDERKEY bigint,  
    L_PARTKEY bigint,  
    L_SUPPKEY bigint,  
    L_LINENUMBER bigint,  
    L_QUANTITY double,  
    L_EXTENDEDPRICE double,  
    L_DISCOUNT double,  
    L_TAX double,  
    L_RETURNFLAG text,  
    L_LINESTATUS text,  
    L_SHIPDATE date,  
    L_COMMITDATE date,  
    L_RECEIPTDATE date,  
    L_SHIPINSTRUCT text,  
    L_SHIPMODE text,  
    L_COMMENT text)  
using text with ('text.delimiter'='|') location 'hdfs://x/y';  
  
create external table part (  
    P_PARTKEY bigint,  
    P_NAME text,  
    P_MFGR text,  
    P_BRAND text,  
    P_TYPE text,  
    P_SIZE integer,  
    P_CONTAINER text,  
    P_RETAILPRICE double,  
    P_COMMENT text)  
using text with ('text.delimiter'='|') location 'hdfs://x/y';  
  
create external table partsupp (  
    PS_PARTKEY bigint,  
    PS_SUPPKEY bigint,  
    PS_AVAILQTY int,  
    PS_SUPPLYCOST double,  
    PS_COMMENT text)  
using text with ('text.delimiter'='|') location 'hdfs://x/y';  
  
create external table customer (  
    C_CUSTKEY bigint,  
    C_NAME text,  
    C_ADDRESS text,  
    C_NATIONKEY bigint,  
    C_PHONE text,  
    C_ACCTBAL double,  
    C_MKTSEGMENT text,
```

```

        C_COMMENT text)
using text with ('text.delimiter'='|') location 'hdfs://x/y';

create external table orders (
    O_ORDERKEY bigint,
    O_CUSTKEY bigint,
    O_ORDERSTATUS text,
    O_TOTALPRICE double,
    O_ORDERDATE date,
    O_ORDERPRIORITY text,
    O_CLERK text,
    O_SHIPPRIORITY int,
    O_COMMENT text)
using text with ('text.delimiter'='|') location 'hdfs://x/y';

create external table nation (
    N_NATIONKEY bigint,
    N_NAME text,
    N_REGIONKEY bigint,
    N_COMMENT text)
using text with ('text.delimiter'='|') location 'hdfs://x/y';

create external table region (
    R_REGIONKEY bigint,
    R_NAME text,
    R_COMMENT text)
using text with ('text.delimiter'='|') location 'hdfs://x/y';

```

TPC-H Queries

Q1

```

select
    l_returnflag,
    l_linestatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(l_extendedprice*(1-l_discount)) as sum_disc_price,
    sum(l_extendedprice*(1-l_discount)*(1+l_tax)) as sum_charge,
    avg(l_quantity) as avg_qty,
    avg(l_extendedprice) as avg_price,
    avg(l_discount) as avg_disc,
    count(*) as count_order
from
    lineitem
where
    l_shipdate <= '1998-09-01'::date
group by
    l_returnflag, l_linestatus
order by
    l_returnflag, l_linestatus

```

Q2

Tajo does not support scalar subquery yet. So, you should use multiple queries as follows:

```

create table nation_region as select n_regionkey, r_regionkey, n_nationkey, n_name, r_name from region join
nation on n_regionkey = r_regionkey where r_name = 'EUROPE';

create table r2_1 as select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_comment,
ps_supplycost from nation_region join supplier on s_nationkey = n_nationkey join partsupp on s_suppkey =
ps_suppkey join part on p_partkey = ps_partkey where p_size = 15 and p_type like '%BRASS';

create table r2_2 as select p_partkey, min(ps_supplycost) as min_ps_supplycost from r2_1 group by p_partkey;

select s_acctbal, s_name, n_name, r2_1.p_partkey, p_mfgr, s_address, s_phone, s_comment from r2_1 join r2_2 on
r2_1.p_partkey = r2_2.p_partkey where ps_supplycost = min_ps_supplycost order by s_acctbal, n_name, s_name,
r2_1.p_partkey;

```

Q3

```

select
    l_orderkey,
    sum(l_extendedprice*(1-l_discount)) as revenue,
    o_orderdate,
    o_shippriority
from
    customer as c
    join orders as o
        on c.c_mktsegment = 'BUILDING' and c.c_custkey = o.o_custkey
    join lineitem as l
        on l.l_orderkey = o.o_orderkey
where
    o_orderdate < '1995-03-15'::date
    and l_shipdate > '1995-03-15'::date
group by
    l_orderkey, o_orderdate, o_shippriority
order by
    revenue desc, o_orderdate;

```

Q5

```

select
    n_name,
    sum(l_extendedprice * (1 - l_discount)) as revenue
from
    customer,
    orders,
    lineitem,
    supplier,
    nation,
    region
where
    c_custkey = o_custkey
    and l_orderkey = o_orderkey
    and l_suppkey = s_suppkey
    and c_nationkey = s_nationkey
    and s_nationkey = n_nationkey
    and n_regionkey = r_regionkey
    and r_name = 'ASIA'
    and o_orderdate >= '1994-01-01'::date
    and o_orderdate < '1995-01-01'::date
group by
    n_name
order by
    revenue desc;

```

Q6

```

select
    sum(l_extendedprice*l_discount) as revenue
from
    lineitem
where
    l_shipdate >= '1994-01-01'::date
    and l_shipdate < '1995-01-01'::date
    and l_discount >= 0.05
    and l_discount <= 0.07
    and l_quantity < 24;

```

Q10

```

select
    c_custkey,
    c_name,
    sum(l_extendedprice * (1 - l_discount)) as revenue,
    c_acctbal,
    n_name,
    c_address,
    c_phone,
    c_comment
from
    customer as c
    join nation as n
        on c.c_nationkey = n.n_nationkey
    join orders as o
        on c.c_custkey = o.o_custkey and o.o_orderdate >= '1993-10-01'::date and o.o_orderdate < '1994-01-01'::date
    join lineitem as l
        on l.l_orderkey = o.o_orderkey and l.l_returnflag = 'R'
group by
    c_custkey, c_name, c_acctbal, c_phone, n_name, c_address, c_comment
order by
    revenue desc

```

Q12

```

select
    l_shipmode,
    sum(case when o_orderpriority = '1-URGENT' or o_orderpriority = '2-HIGH' then 1 else 0 end) as
high_line_count,
    sum(case when o_orderpriority <> '1-URGENT' and o_orderpriority <> '2-HIGH' then 1 else 0 end) as
low_line_count
from
    orders,
    lineitem
where
    o_orderkey = l_orderkey
    and (l_shipmode = 'MAIL' or l_shipmode = 'SHIP')
    and l_commitdate < l_receiptdate
    and l_shipdate < l_commitdate
    and l_receiptdate >= '1994-01-01'::date
    and l_receiptdate < '1995-01-01'::date
group by
    l_shipmode
order by
    l_shipmode

```

Q14

```

select
    100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice*(1-l_discount) else 0 end) / sum
(l_extendedprice * (1 - l_discount)) as promo_revenue
from
    lineitem,
    part
where
    l_partkey = p_partkey
    and l_shipdate >= '1995-09-01'::date
    and l_shipdate < '1995-10-01'::date

```

Q19

```

select
    sum(l_extendedprice * (1 - l_discount) ) as revenue
from
    lineitem,
    part
where
    (
        p_partkey = l_partkey
        and p_brand = 'Brand#12'
        and p_container in ( 'SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
        and l_quantity >= 1 and l_quantity <= 1 + 10
        and p_size between 1 and 5
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON'
    )
    or
    (
        p_partkey = l_partkey
        and p_brand = 'Brand#23'
        and p_container in ( 'MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
        and l_quantity >= 10 and l_quantity <= 10 + 10
        and p_size between 1 and 10
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON'
    )
    or
    (
        p_partkey = l_partkey
        and p_brand = 'Brand#34'
        and p_container in ( 'LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
        and l_quantity >= 20 and l_quantity <= 20 + 10
        and p_size between 1 and 15
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON'
    );

```