

# TPCH Test Case

---

## Test Case

## Test Case

Due to the limited availability of real-time ETL datasets within the industry, such as NetMax, which predominantly feature single-table scenarios, it becomes difficult to reflect the performance in realistic settings. Therefore, similar to approaches adopted in some Incremental View Maintenance (IVM) studies, we select the standard TPC-H benchmark dataset as base data. We then generate data processing tasks by constructing corresponding views or view groups for queries Q1–Q22 to complete our tests.

Considering that standard TPC-H queries include dynamically changing parameters, the constructed views aim to encompass complete data as much as possible (excluding queries like Q13 and Q20 for which designing compliant views was infeasible; we simplify these by removing the WHERE conditions to complete view definitions). This ensures that even as query parameters change, the relevant TPC-H queries can still retrieve correct results from the views. This approach closely mirrors the requirements in real-world scenarios.

Q1

```
1  select
2      l_returnflag,
3      l_linestatus,
4      l_shipdate,
5      sum(l_quantity) as sum_qty,
6      sum(l_extendedprice) as sum_base_price,
7      sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
8      sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
9      sum(l_extendedprice) as sum_price,
10     sum(l_discount) as sum_disc,
11     count(l_quantity) as count_qty,
12     count(l_extendedprice) as count_price,
13     count(l_discount) as count_disc,
14     count(*) as count_order
15  from
16      lineitem
17  group by
18      l_returnflag,
19      l_linestatus,
20      l_shipdate
21  distributed by (l_returnflag, l_linestatus);
```

Q2

```
1  SELECT s_acctbal,
2      s_name,
3      n_name,
4      p_partkey,
5      p_mfgr,
6      s_address,
7      s_phone,
8      s_comment,
9      ps_supplycost,
10     p_size,
11     p_type,
12     r_name
13  FROM part,
14     supplier,
15     partsupp,
16     nation,
17     region
18  WHERE
19     p_partkey = ps_partkey
20     AND s_suppkey = ps_suppkey
21     AND s_nationkey = n_nationkey
22     AND n_regionkey = r_regionkey;
23
24  create incremental materialized view q2_mv2 as
25  SELECT
26     min(ps_supplycost),
27     ps_partkey as t2_partkey,
28     r_name
29  FROM
30     partsupp,
31     supplier,
32     nation,
33     region
34  WHERE
35     s_suppkey = ps_suppkey
36     AND s_nationkey = n_nationkey
37     AND n_regionkey = r_regionkey
38  GROUP BY
39     r_name,
40     ps_partkey;
```

Q3

```
1  select
2      l_orderkey,
3      sum(l_extendedprice * (1 - l_discount)) as revenue,
4      o_orderdate,
5      o_shippriority,
6      l_shipdate,
7      c_mktsegment
8  from
9      customer,
10     orders,
11     lineitem
12  where
13      c_custkey = o_custkey
14      and l_orderkey = o_orderkey
15  group by
16      l_orderkey,
17      o_orderdate,
18      o_shippriority,
19      o_orderdate,
20      l_shipdate,
21      c_mktsegment;
```

Q4

```
1  select
2      o_orderpriority,
3      count(*) as order_count
4  from
5      orders,
6      lineitem
7  where
8      l_orderkey = o_orderkey
9      and l_commitdate < l_receiptdate
10 group by
11     o_orderpriority,
12     o_orderdate;
```

Q5

```
1  select
2      n_name,
3      sum(l_extendedprice * (1 - l_discount)) as revenue,
4      r_name,
5      o_orderdate
6  from
7      customer,
8      orders,
9      lineitem,
10     supplier,
11     nation,
12     region
13  where
14      c_custkey = o_custkey
15      and l_orderkey = o_orderkey
16      and l_suppkey = s_suppkey
17      and c_nationkey = s_nationkey
18      and s_nationkey = n_nationkey
19      and n_regionkey = r_regionkey
20  group by
21      n_name,
22      r_name,
23      o_orderdate;
```

Q6

```
1  select
2      sum(l_extendedprice * l_discount) as revenue,
3      l_shipdate,
4      l_discount,
5      l_quantity
6  from
7      lineitem
8  group by
9      l_shipdate,
10     l_discount,
11     l_quantity;
```

Q7

```
1  select
2      n1.n_name as supp_nation,
3      n2.n_name as cust_nation,
4      extract(year from l_shipdate) as l_year,
5      sum(l_extendedprice * (1 - l_discount)) as revenue
6  from
7      supplier,
8      lineitem,
9      orders,
10     customer,
11     nation n1,
12     nation n2
13  where
14      s_suppkey = l_suppkey
15      and o_orderkey = l_orderkey
16      and c_custkey = o_custkey
17      and s_nationkey = n1.n_nationkey
18      and c_nationkey = n2.n_nationkey
19      --and l_shipdate between date '1995-01-01' and date '1996-12-31'
20  group by
21      supp_nation,
22      cust_nation,
23      l_year;
```

Q8

```
1  select
2      extract(year from o_orderdate) as o_year,
3      sum(l_extendedprice * (1 - l_discount)) as mkt_share,
4      r_name,
5      p_type,
6      n2.n_name as nation
7  from
8      part,
9      supplier,
10     lineitem,
11     orders,
12     customer,
13     nation n1,
14     nation n2,
15     region
16  where
17     p_partkey = l_partkey
18     and s_suppkey = l_suppkey
19     and l_orderkey = o_orderkey
20     and o_custkey = c_custkey
21     and c_nationkey = n1.n_nationkey
22     and n1.n_regionkey = r_regionkey
23     and s_nationkey = n2.n_nationkey
24     --and o_orderdate between date '1995-01-01' and date '1996-12-31'
25  group by
26     o_year,
27     r_name,
28     p_type,
29     n2.n_name;
```

Q9

```
1  select
2      n_name as nation,
3      extract(year from o_orderdate) as o_year,
4      sum(l_extendedprice * (1 - l_discount) - ps_supplycost * l_quantity) a
5  s sum_profit,
6      p_name
7  from
8      part,
9      supplier,
10     lineitem,
11     partsupp,
12     orders,
13     nation
14  where
15     s_suppkey = l_suppkey
16     and ps_suppkey = l_suppkey
17     and ps_partkey = l_partkey
18     and p_partkey = l_partkey
19     and o_orderkey = l_orderkey
20     and s_nationkey = n_nationkey
21  group by
22     nation,
23     o_year,
24     p_name;
```

Q10



```
1  select
2      c_custkey,
3      c_name,
4      sum(l_extendedprice * (1 - l_discount)) as revenue,
5      c_acctbal,
6      n_name,
7      c_address,
8      c_phone,
9      c_comment,
10     o_orderdate
11  from
12     customer,
13     orders,
14     lineitem,
15     nation
16  where
17     c_custkey = o_custkey
18     and l_orderkey = o_orderkey
19     and l_returnflag = 'R'
20     and c_nationkey = n_nationkey
21  group by
22     c_custkey,
23     c_name,
24     c_acctbal,
25     c_phone,
26     n_name,
27     c_address,
28     c_comment,
29     o_orderdate;
```

Q11

```
1  select
2      ps_partkey,
3      sum(ps_supplycost * ps_availqty) as value,
4      n_name
5  from
6      partsupp,
7      supplier,
8      nation
9  where
10     ps_suppkey = s_suppkey
11     and s_nationkey = n_nationkey
12  group by
13     ps_partkey,
14     n_name;
```

Q12

```
1  select
2      l_shipmode,
3      sum(case
4          when o_orderpriority = '1-URGENT'
5              or o_orderpriority = '2-HIGH'
6              then 1
7              else 0
8      end) as high_line_count,
9      sum(case
10         when o_orderpriority <> '1-URGENT'
11             and o_orderpriority <> '2-HIGH'
12             then 1
13             else 0
14     end) as low_line_count,
15     l_receiptdate
16  from
17     orders,
18     lineitem
19  where
20     o_orderkey = l_orderkey
21     and l_commitdate < l_receiptdate
22     and l_shipdate < l_commitdate
23  group by
24     l_shipmode,
25     l_receiptdate;
```

Q13

Plain Text

```
1  select
2      c_count,
3      count(*) as custdist
4  from
5      (
6          select
7              c_custkey,
8              count(o_orderkey)
9          from
10             customer left outer join orders on
11                 c_custkey = o_custkey
12                 --and o_comment
13          group by
14              c_custkey
15      ) as c_orders (c_custkey, c_count)
16  group by
17      c_count;
```

Q14

Plain Text

```
1  select
2      sum(case
3          when p_type like 'PROMO%'
4              then l_extendedprice * (1 - l_discount)
5          else 0
6      end) as s1,
7      sum(l_extendedprice * (1 - l_discount)) as s2,
8      l_shipdate
9  from
10     lineitem,
11     part
12  where
13     l_partkey = p_partkey
14  group by
15     l_shipdate;
```

Q15

```
1  select
2      l_suppkey as supplier_no,
3      sum(l_extendedprice * (1 - l_discount)) as total_revenue,
4      l_shipdate
5  from
6      lineitem
7  group by
8      l_suppkey,
9      l_shipdate;
```

Q16

```
1  select
2      p_brand,
3      p_type,
4      p_size,
5      count(distinct ps_suppkey) as supplier_cnt
6  from
7      partsupp,
8      part,
9      supplier
10 where
11     p_partkey = ps_partkey
12     and ps_suppkey = s_suppkey
13     and s_comment not like '%Customer%Complaints%'
14 group by
15     p_brand,
16     p_type,
17     p_size;
```

Q17

```
1  select
2      sum(l_extendedprice/7.0) mv_sum,
3      p_brand,
4      p_container
5  from
6      lineitem,
7      part,
8      (SELECT
9          l_partkey avg_l_partkey,
10         avg(l_quantity) AS avg_quantity
11        FROM
12         lineitem
13        GROUP BY
14         l_partkey) t
15  where
16      p_partkey = l_partkey
17      and avg_l_partkey = l_partkey and l_quantity < 0.2*avg_quantity
18  group by
19      p_brand,
20      p_container;
```

Q18

```
1  select
2      c_name,
3      c_custkey,
4      o_orderkey,
5      o_orderdate,
6      o_totalprice,
7      sum(l_quantity) sum,
8      q18_mv1.sum inner_sum
9  from
10     customer,
11     orders,
12     lineitem l,
13     (select
14         l_orderkey,
15         sum(l_quantity) sum
16     from
17         lineitem
18     group by
19         l_orderkey) q18_mv1
20  where
21     o_orderkey = q18_mv1.l_orderkey
22     and c_custkey = o_custkey
23     and o_orderkey = l.l_orderkey
24     -- and q18_mv1.sum > 313
25  group by
26     c_name,
27     c_custkey,
28     o_orderkey,
29     o_orderdate,
30     o_totalprice,
31     q18_mv1.sum;
```

Q19

```
1  select
2      p_brand,
3      l_quantity,
4      p_container,
5      p_size,
6      sum(l_extendedprice* (1 - l_discount)) as revenue
7  from
8      lineitem,
9      part
10 where
11     (
12         p_partkey = l_partkey
13         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PK
14     G')
15         and p_size between 1 and 5
16         and l_shipmode in ('AIR', 'AIR REG')
17         and l_shipinstruct = 'DELIVER IN PERSON'
18     )
19     or
20     (
21         p_partkey = l_partkey
22         and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED
23     PACK')
24         and p_size between 1 and 10
25         and l_shipmode in ('AIR', 'AIR REG')
26         and l_shipinstruct = 'DELIVER IN PERSON'
27     )
28     or
29     (
30         p_partkey = l_partkey
31         and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PK
32     G')
33         and p_size between 1 and 15
34         and l_shipmode in ('AIR', 'AIR REG')
35         and l_shipinstruct = 'DELIVER IN PERSON'
36     )
37 group by
38     p_brand,
39     l_quantity,
40     p_container,
41     p_size;
```

```

1  select
2      l_partkey agg_partkey,
3      l_suppkey agg_suppkey,
4      sum(l_quantity) AS agg_quantity
5  from
6      dbo.lineitem
7  group by
8      l_partkey,
9      l_suppkey;

```

Q21

```

1  SELECT s_name, n_name, COUNT(*) AS numwait
2  FROM supplier
3  JOIN lineitem l1 ON s_suppkey = l1.l_suppkey
4  JOIN orders ON o_orderkey = l1.l_orderkey
5  JOIN nation ON s_nationkey = n_nationkey
6  -- 处理多供应商条件 (原EXISTS)
7  JOIN (
8      SELECT l_orderkey, COUNT(DISTINCT l_suppkey)
9      FROM lineitem
10     GROUP BY l_orderkey
11     HAVING COUNT(DISTINCT l_suppkey) >= 2
12 ) multi_orders ON l1.l_orderkey = multi_orders.l_orderkey
13 -- 处理无其他供应商延迟条件 (原NOT EXISTS)
14 LEFT JOIN (
15     SELECT l_main.l_orderkey, l_main.l_suppkey
16     FROM lineitem l_main
17     JOIN lineitem l_other
18     ON l_main.l_orderkey = l_other.l_orderkey
19     AND l_main.l_suppkey <> l_other.l_suppkey
20     AND l_other.l_receiptdate > l_other.l_commitdate
21     GROUP BY l_main.l_orderkey, l_main.l_suppkey
22 ) delayed_other
23 ON l1.l_orderkey = delayed_other.l_orderkey
24 AND l1.l_suppkey = delayed_other.l_suppkey
25 WHERE o_orderstatus = 'F'
26     AND l1.l_receiptdate > l1.l_commitdate
27     AND delayed_other.l_orderkey IS NULL -- 反连接实现NOT EXISTS
28 GROUP BY s_name, n_name;

```

Q22



```

1 SELECT c_acctbal, substring(c_phone FROM 1 FOR 2) AS cntrycode
2 FROM customer LEFT JOIN orders ON o_custkey = c_custkey WHERE o_custkey IS
NULL

```

	Streaming View	SPY	DBX (异步)	DBY (同步)
Q1	12.50	5.45	0.50	0.34
Q2	9.83	2.32	0.47	✗ min / max
Q3	9.50	4.00	0.43	0.12
Q4	12.24	4.41	0.44	0.24
Q5	10.52	3.90	0.50	Not O 0.18 O 0.24
Q6	9.83	4.84	0.54	0.33
Q7	9.23	3.47	0.52	✗ self join
Q8	8.59	3.12	0.46	✗ self join
Q9	4.58	2.51	0.43	Not O 0.05 O 0.11
Q10	11.53	5.61	0.44	0.22
Q11	13.9	6.00	0.46	0.92
Q12	12.00	6.18	0.44	0.29
Q13	12.50	6.98	✗ left join	✗ left join
Q14	9.67	5.36	0.46	0.32
Q15	8.82	6.32	0.42	0.12
Q16	12.24	6.98	0.47	✗ count(distinct)
Q17	6.06	5.40	0.47	✗ agg & nest
Q18	7.57	1.41	0.48	✗ nest

Q19	10.71	6.00	0.43	0.31
Q20	11.30	5.95	✗ deep parameter	0.34
Q21	5.30	1.66	✗ left join	✗ left join
Q22	12.50	1.84	✗ left join	✗ left join

	Q1	Q2	Q8	Q10	Q12	Q18
Streaming View 1W	34.02	61.54	91.60	36.42	31.32	58.28
Streaming View 2W	61.16	101.4	160.00	64.42	56.76	117.6
Streaming View 4W	127.00	194.20	244.60	122.2	111.20	235
SPY-O 1W	54.48	576.52	215.28	48.88	37.14	185.6
SPY-O 2W	106.5	614.6	480.6	89.8	63.3	234
SPY-O 4W	305.8	623.2	560	147.54	116.32	241
DBX (<1W)	400.56	324.64	399.36	380.48	373.12	423.68
DBY (<1W)	775.04	✗	✗	568.72	726.24	✗

	Q1	Q2	Q8	Q10	Q12	Q18
Streaming View	2.83G	3.34G	3.03G	3.56G	3.43G	3.33G
SPY	3.55G	17.6G	16.7G	7.75G	5.80G	8.09G
DBX	1.36G	1.42G	1.44G	1.41G	1.46G	1.42G

DBY	1.02G	×	×	0.74G	0.64G	×
-----	-------	---	---	-------	-------	---

  

	Q1	Q2	Q8	Q10	Q12	Q18
Streaming View(1G)	12.76	10.00	10.32	11.76	13.04	8.20
Streaming View(10G)	12.50	9.83	8.59	11.53	12.00	7.57
Streaming View(100G)	12.11	8.95	7.31	11.53	11.76	6.51
Streaming View(1T)	12.00	8.63	5.61	9.67	11.76	5.57
SPY(1G)	6.42	2.42	3.27	6.06	6.18	1.62
SPY(10G)	5.45	2.32	3.12	5.61	6.18	1.41
SPY(100G)	5.41	2.22	3.11	0.38	5.88	1.25
SPY(1T)	×	×	×	×	×	×
DBX(1G)	0.53	0.47	0.46	0.46	0.46	0.49
DBX(10G)	0.50	0.47	0.46	0.44	0.44	0.48
DBX(100G)	0.46	0.45	0.41	0.42	0.40	0.42
DBX(1T)	0.44	0.36	0.35	0.34	0.32	0.31
DBY (1G)	0.33	×	×	0.27	0.29	×
DBY(10G)	0.34	×	×	0.22	0.27	×
DBY(100G)	0.38	×	×	0.19	0.26	×
DBY(1T)	0.31	×	×	0.12	0.16	×