25 March 2017

High Volume In-Memory Data Unification

Performance Results for UniConnect Platform powered by Intel® Xeon® Processor E7 Family

Contents

Authors
Vicky Soni
Prathmesh Datye
Sumit Kulkarni

Executive Summary

Enterprises today must put all their data to good use if they are to survive in an increasingly digitised world. However, this necessitates the management of large volumes of data across disparate systems. Enterprises must ensure when choosing a big data platform that analytical queries can be processed with speed, stability and efficiency.

To demonstrate this capability, Percipient tested its UniConnect platform on the Intel Xeon E7-8800/4800v4 series of processors. This testing exercise demonstrated that, just on a **single node installation**, the UniConnect platform was able to produce average query CPU times ranging from under 1 minute for 1 GB, to 10 hours for 3000 GB of data. The query CPU time for unifying 2.3 billion rows of Hive and CSV data was 30 minutes.

Background

The *UniConnect 2.2* platform is designed to integrate highly diverse data with maximum efficiency. The platform, comprised of proprietary and open source elements, offers in memory parallel processing and SQL-based capabilities. These allow organisations to implement flexible, high speed and scalable data analytics and data applications while overcoming universal data challenges such as cost and complexity.

The *Intel Xeon E7-8800/4800v4* series with Intel NVMe SSDs is highly optimized for business processing and analytics workloads such as OLTP, ERP, Data Analysis etc. This product family offers the required compute and advanced caching capabilities required to run the UniConnect platform. This combined stack comprising of Percipient's UniConnect Platform with Intel Xeon E7 v4 and Intel NVMe SSDs based hardware provides an integrated appliance for ease of deployment.

In order to determine the best performance that such a stack is able to produce, Percipient conducted a performance test in March 2017. Percipient used a specific indicator, the TPC Benchmark™H (TPC-H), to measure the performance of queries run on the UniConnect platform powered by the Intel Xeon processor. The UniConnect Testing Driver was used to generate performance results.

The TPC is a non-profit corporation founded to define transaction processing and database benchmarks and to disseminate objective, verifiable TPC performance data to the industry. Typically, the TPC produces benchmarks that measure transaction processing (TP) and database (DB) performance in terms of how many transactions a given system and database can perform per unit of time, e.g., transactions per second or transactions per minute.

The TPC Benchmark™H (TPC-H) is a decision support benchmark. It consists of a suite of business oriented ad-hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance. This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity, and give answers to critical business questions. For more details about TPC-H refer to:http://www.tpc.org/tpch/

Test Environment

Specifications for the test environment were as follows:

Server
Server Vendor- GenuineIntel
OS- CentOS – x86_64
Core(s) per socket - 16
Sockets – 4
Processor – Intel® Xeon® CPU E7-4850 @2.10GHz
Total RAM – 263.83 GB
Available Memory – 260.55 GB
Free Memory –257.96 GB
Hard Disk size (/home) – 1.4TB
UniConnect Platform (Single Node Installation)
JVM – Xmx200G (ALLOCATED MEMORY FOR jvm IS 200GB)
Query Max Memory- 200GB
Query Max Memory per Node -110GB

Hadoop Ecosystem
Apache Hadoop- 2.7.3
Apache Hive- 2.1.1
Apache Derby – 10.13.1.1

Dataset Sizes

The testing process was executed on datasets of the following sizes:

TPC-H Catalog:

- 1GB
- 100GB
- 1000GB
- 3000GB

Hive Catalog:

- 300GB
- 1000GB

Performance Results

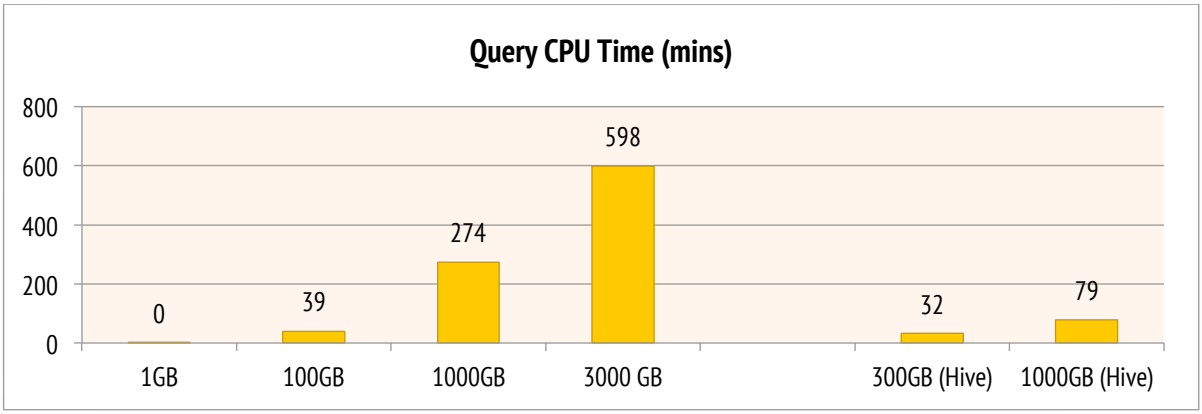
The UniConnect Testing Driver measures the **wall time**, the **CPU time** used by the query, and the total **process CPU time** for UniConnect processes. For each query, the driver reports the median, mean and standard deviation of the query runs. The difference between process and query CPU times is the query overhead, which is normally from garbage collections.

1. **Wall Time**: This is the real world amount of time taken from the start of a task to its completion. It is equivalent to timing your job with a stopwatch and the measured time can be affected by anything else that the system happens to be doing at the time.
2. **Query CPU Time**: This is the actual time to execute a query on the data source
3. **Process CPU Time**: This measures the total amount of time the CPU spent running your query or anything requested by your application. This includes kernel time.

Note that while wall (clock) times can be expected to exceed CPU times in a single processor environment, this is not the case for multi-threading processes. In this test, given the 4 socket, 16 cores per socket environment, and use of parallel execution, the Query CPU times shown significantly exceed the Wall times.

Based on the above attributes, the UniConnect + Xeon performances were measured for a total of 21 TPC-H, Hive, and Unified (CSV + Hive) queries. Performances were also measured for datasets of varying sizes. Each query was executed 3 times, from which a mean time was derived, and these were then averaged across the 21 queries.

Mean times averaged for 21 queries (mins)			
	Wall time	Query CPU Time	Process CPU Time
1GB	0.06	0.44	0.77
100GB	1.02	39.01	50.26
1000GB	4.46	274.17	303.45
3000 GB	9.35	598.39	657.10
Mean times averaged for 21 queries (mins)			
	Wall time	**Query CPU Time**	**Process CPU Time**
300GB (Hive)	2.44	31.87	55.58
1000GB (Hive)	4.38	78.80	115.19



In addition, non-TPC-H performance measurements were taken, using the UniConnect UI, for queries based on the unification of CSV and Hive data. The benchmarking environment remained the same.

Times for CSV + Hive (300GB) unified queries (mins)		
	Wall time	Query CPU time
Query 1: 2.26 billion rows	4.58	37.97
Query 2: 2.3 billion rows	6.11	29.73
Query 3: 495 million rows	3.72	41.52

The results achieved demonstrate that, just on a **single node installation**, the UniConnect platform is able to produce average query CPU times ranging from under 1 minute for 1 GB, to 10 hours for 3000 GB, of data. The query CPU time for unifying 2.3 billion rows of Hive and CSV data was 30 minutes. The above times can be further reduced by adding nodes as required.

Appendix

Benchmarking Queries

The following are the TPC-H standard queries used for the above benchmarking.

Query_ID: TPCH_1

```
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 l
where
    l.shipdate <= date '1998-12-01' - interval '65' day
group by
    l.returnflag,
    l.linestatus
order by
    l.returnflag,
    l.linestatus
```

Query_Id: TPCH_2

```
select
    s.acctbal,
    s.name,
    n.name,
    p.partkey,
    p.mfgr,
    s.address,
    s.phone,
    s.comment
from
    part p,
    supplier s ,
    partsupp ps,
```

```
        nation n,
        region r
    where
        p.partkey = ps.partkey
        and s.suppkey = ps.suppkey
        and p.size = 20
        and p.type like '%COPPER'
        and s.nationkey = n.nationkey
        and n.regionkey = r.regionkey
        and r.name = 'ASIA'
        and ps.supplycost = (
            select
                min(ps.supplycost)
            from
                partsupp ps,
                supplier s ,
                nation n,
                region r
            where
                p.partkey = ps.partkey
                and s.suppkey = ps.suppkey
                and s.nationkey = n.nationkey
                and n.regionkey = r.regionkey
                and r.name = 'ASIA'
        )
    order by
        s.acctbal desc,
        n.name,
        s.name,
        p.partkey
```

Query_ID: TPCH_3

```
    select
        l.orderkey,
        sum(l.extendedprice * (1 - l.discount)) as revenue,
        o.orderdate,
        o.shippriority
    from
        customer c ,
        orders o,
        lineitem l
    where
        c.mktsegment = 'MACHINERY'
        and c.custkey = o.custkey
        and l.orderkey = o.orderkey
        and o.orderdate < date '1995-03-18'
```

```
        and l.shipdate > date '1995-03-18'
    group by
        l.orderkey,
        o.orderdate,
        o.shippriority
    order by
        revenue desc,
        o.orderdate
```

Query_Id: TPCH_4

```
    select
        o.orderpriority,
        count(*) as order_count
    from
        orders o
    where
        o.orderdate >= date '1995-08-01'
        and o.orderdate < date '1995-08-01' + interval '3' month
        and exists (
            select
                *
            from
                lineitem l
            where
                l.orderkey = o.orderkey
                and l.commitdate < l.receiptdate
        )
    group by
        o.orderpriority
    order by
        o.orderpriority
```

Query_Id: TPCH_5

```
    Select
        n.name,
        sum(l.extendedprice * (1 - l.discount)) as revenue
    from
        customer c ,
        orders o,
        lineitem l,
        supplier s ,
        nation n,
        region r
    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 = 'AFRICA'
        and o.orderdate >= date '1996-01-01'
        and o.orderdate < date '1996-01-01' + interval '1' year
    group by
        n.name
    order by
        revenue desc
```

Query_Id: TPCH_6

```
    Select
        sum(l.extendedprice * l.discount) as revenue
    from
        lineitem l
    where
        l.shipdate >= date '1996-01-01'
        and l.shipdate < date '1996-01-01' + interval '1' year
        and l.discount between 0.09 - 0.01 and 0.09 + 0.01
        and l.quantity < 24
```

Query_Id: TPCH_7

```
    Select
        supp_nation,
        cust_nation,
        l_year,
        sum(volume) as revenue
    from
        (
            select
                n1.name as supp_nation,
                n2.name as cust_nation,
                extract(year from l.shipdate) as l_year,
                l.extendedprice * (1 - l.discount) as volume
            from
                supplier s ,
                lineitem l,
                orders o,
                customer c ,
                nation n1,
                nation n2
```

```
        where
                s.suppkey = l.suppkey
                and o.orderkey = l.orderkey
                and c.custkey = o.custkey
                and s.nationkey = n1.nationkey
                and c.nationkey = n2.nationkey
                and (
                        (n1.name = 'UNITED KINGDOM' and n2.name = 'ARGENTINA')
                        or (n1.name = 'ARGENTINA' and n2.name = 'UNITED KINGDOM')
                )
                and l.shipdate between date '1995-01-01' and date '1996-12-31'
    ) as shipping
group by
    supp_nation,
    cust_nation,
    l_year
order by
    supp_nation,
    cust_nation,
    l_year
```

Query_Id: TPCH_8

```
Select
    o_year,
    sum(case
        when nation = 'ARGENTINA' then volume
        else 0
    end) / sum(volume) as mkt_share
from
    (
        select
            extract(year from o.orderdate) as o_year,
            l.extendedprice * (1 - l.discount) as volume,
            n2.name as nation
        from
            part p,
            supplier s ,
            lineitem l,
            orders o,
            customer c ,
            nation n1,
            nation n2,
            region r
        where
            p.partkey = l.partkey
```

```
                and s.suppkey = l.suppkey
                and l.orderkey = o.orderkey
                and o.custkey = c.custkey
                and c.nationkey = n1.nationkey
                and n1.regionkey = r.regionkey
                and r.name = 'AMERICA'
                and s.nationkey = n2.nationkey
                and o.orderdate between date '1995-01-01' and date '1996-12-31'
                and p.type = 'LARGE POLISHED BRASS'
        ) as all_nation
group by
        o_year
order by
        o_year
```

Query_Id: TPCH_9

```
Select
        nation,
        o_year,
        sum(amount) as sum_profit
from
        (
                select
                        n.name as nation,
                        extract(year from o.orderdate) as o_year,
                        l.extendedprice * (1 - l.discount) - ps.supplycost * l.quantity as amount
                from
                        part p,
                        supplier s ,
                        lineitem l,
                        partsupp ps,
                        orders o,
                        nation n
                where
                        s.suppkey = l.suppkey
                        and ps.suppkey = l.suppkey
                        and ps.partkey = l.partkey
                        and p.partkey = l.partkey
                        and o.orderkey = l.orderkey
                        and s.nationkey = n.nationkey
                        and p.name like '%navy%'
        ) as profit
group by
        nation,
        o_year
order by
```

```
    nation,
    o_year desc
```

Query_Id: TPCH_10

```
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 c ,
    orders o,
    lineitem l,
    nation n
where
    c.custkey = o.custkey
    and l.orderkey = o.orderkey
    and o.orderdate >= date '1994-08-01'
    and o.orderdate < date '1994-08-01' + interval '3' month
    and l.returnflag = 'R'
    and c.nationkey = n.nationkey
group by
    c.custkey,
    c.name,
    c.acctbal,
    c.phone,
    n.name,
    c.address,
    c.comment
order by
    revenue desc
```

Query_ID: TPCH_11

```
Select
    ps.partkey,
    sum(ps.supplycost * ps.availqty) as value
from
    partsupp ps,
    supplier s ,
    nation n
```

```
where
        ps.suppkey = s.suppkey
        and s.nationkey = n.nationkey
        and n.name = 'ARGENTINA'
group by
        ps.partkey having
                sum(ps.supplycost * ps.availqty) > (
                        select
                                sum(ps.supplycost * ps.availqty) * 0.0001000000
                        from
                                partsupp ps,
                                supplier s ,
                                nation n
                        where
                                ps.suppkey = s.suppkey
                                and s.nationkey = n.nationkey
                                and n.name = 'ARGENTINA'
                )
order by
        value desc
```

Query_ID: TPCH_12

```
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 o,
        lineitem l
where
        o.orderkey = l.orderkey
        and l.shipmode in ('AIR', 'TRUCK')
        and l.commitdate < l.receiptdate
        and l.shipdate < l.commitdate
        and l.receiptdate >= date '1995-01-01'
        and l.receiptdate < date '1995-01-01' + interval '1' year
```

```
group by
    l.shipmode
order by
    l.shipmode
```

Query_ID: TPCH_13

```
Select
    c_count,
    count(*) as custdist
from
    (
        select
            c.custkey,
            count(o.orderkey)
        from
            customer c  left outer join orders o on
                c.custkey = o.custkey
                and o.comment not like '%special%accounts%'
        group by
            c.custkey
    ) as c_orders (c_custkey, c_count)
group by
    c_count
order by
    custdist desc,
    c_count desc
```

Query_ID: TPCH_14

```
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 promorevenue
from
    lineitem l,
    part p
where
    l.partkey = p.partkey
    and l.shipdate >= date '1995-09-01'
    and l.shipdate < date '1995-09-01' + interval '1' month
```

Query_ID: TPCH_15

```
with revenue0 as (select
        l.suppkey supplier_no,
        sum(l.extendedprice * (1 - l.discount)) total_revenue
    from
        lineitem l
    where
        l.shipdate >= date '1995-03-01'
        and l.shipdate < date '1995-03-01' + interval '3' month
    group by
        l.suppkey)

select
    s.suppkey,
    s.name,
    s.address,
    s.phone,
    total_revenue
from
    supplier s ,
     revenue0
where
    s.suppkey = supplier_no
    and total_revenue = (
        select
            max(total_revenue)
        from
            revenue0
    )
order by
    s.suppkey
```

Query_ID: TPCH_16

```
Select
    p.brand,
    p.type,
    p.size,
    count(distinct ps.suppkey) as supplier_cnt
from
    partsupp ps,
    part p
where
    p.partkey = ps.partkey
    and p.brand <> 'Brand#33'
    and p.type not like 'STANDARD ANODIZED%'
    and p.size in (41, 39, 47, 12, 6, 33, 23, 18)
```

```
        and ps.suppkey not in (
            select
                s.suppkey
            from
                supplier s
            where
                s.comment like '%Customer%Complaints%'
        )
    group by
        p.brand,
        p.type,
        p.size
    order by
        supplier_cnt desc,
        p.brand,
        p.type,
        p.size
```

Query_ID: TPCH_17

```
    Select
        sum(l.extendedprice) / 7.0 as avg_yearly
    from
        lineitem l,
        part p
    where
        p.partkey = l.partkey
        and p.brand = 'Brand#31'
        and p.container = 'WRAP CASE'
        and l.quantity < (
            select
                0.2 * avg(l.quantity)
            from
                lineitem l
            where
                l.partkey = p.partkey
```

Query_ID: TPCH_18

```
    Select
        c.name,
        c.custkey,
        o.orderkey,
        o.orderdate,
        o.totalprice,
        sum(l.quantity)
    from
```

```
        customer c ,
        orders o,
        lineitem l
    where
        o.orderkey in (
            select
                l.orderkey
            from
                lineitem l
            group by
                l.orderkey having
                        sum(l.quantity) > 312
        )
        and c.custkey = o.custkey
        and o.orderkey = l.orderkey
    group by
        c.name,
        c.custkey,
        o.orderkey,
        o.orderdate,
        o.totalprice
    order by
        o.totalprice desc,
        o.orderdate
```

Query_ID: TPCH_19

```
    Select
        sum(l.extendedprice* (1 - l.discount)) as revenue
    from
        lineitem l,
        part p
    where
        (
            p.partkey = l.partkey
            and p.brand = 'Brand#54'
            and p.container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
            and l.quantity >= 4 and l.quantity <= 4 + 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#51'
            and p.container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
```

```
        and l.quantity >= 12 and l.quantity <= 12 + 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#14'
        and p.container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
        and l.quantity >= 23 and l.quantity <= 23 + 10
        and p.size between 1 and 15
        and l.shipmode in ('AIR', 'AIR REG')
        and l.shipinstruct = 'DELIVER IN PERSON'
    )
```

Query_ID: TPCH_20

```
Select
    s.name,
    s.address
from
    supplier s ,
    nation n
where
    s.suppkey in (
        select
            ps.suppkey
        from
            partsupp ps
        where
            ps.partkey in (
                select
                    p.partkey
                from
                    part p
                where
                    p.name like 'rose%'
            )
            and ps.availqty > (
                select
                    0.5 * sum(l.quantity)
                from
                    lineitem l
                where
                    l.partkey = ps.partkey
```

```
                                and l.suppkey = ps.suppkey
                                and l.shipdate >= date '1997-01-01'
                                and l.shipdate < date '1997-01-01' + interval '1' year
                )
        )
        and s.nationkey = n.nationkey
        and n.name = 'EGYPT'
order by
        s.name
```

Query_ID: TPCH_21

```
Select
        s.name,
        count(*) as numwait
from
        supplier s ,
        lineitem l1,
        orders o,
        nation n
where
        s.suppkey = l1.suppkey
        and o.orderkey = l1.orderkey
        and o.orderstatus = 'F'
        and l1.receiptdate > l1.commitdate
        and exists (
                select
                        *
                from
                        lineitem l2
                where
                        l2.orderkey = l1.orderkey
                        and l2.suppkey <> l1.suppkey
        )
        and not exists (
                select
                        *
                from
                        lineitem l3
                where
                        l3.orderkey = l1.orderkey
                        and l3.suppkey <> l1.suppkey
                        and l3.receiptdate > l3.commitdate
        )
        and s.nationkey = n.nationkey
        and n.name = 'UNITED KINGDOM'
group by
```

```
        s.name
    order by
        numwait desc,
        s.name
```

Query_ID: TPCH_22

```
    Select
        cntrycode,
        count(*) as numcust,
        sum(acctbal) as totacctbal
    from
        (
            select
                substring(c.phone from 1 for 2) as cntrycode,
                c.acctbal
            from
                customer c
            where
                substring(c.phone from 1 for 2) in
                        ('14', '11', '26', '21', '18', '20', '12')
                and c.acctbal > (
                        select
                                avg(c.acctbal)
                        from
                                customer c
                        where
                                c.acctbal > 0.00
                                and substring(c.phone from 1 for 2) in
                                        ('14', '11', '26', '21', '18', '20', '12')
                )
                and not exists (
                        select
                                *
                        from
                                orders o
                        where
                                o.custkey = c.custkey
                )
        ) as custsale
    group by
        cntrycode
    order by
        cntrycode
```

The following are non-TPC-H unified data queries

Query 1

```
select
	l.orderkey,
	sum(l.extendedprice * (1 - l.discount)) as revenue,
	o.orderdate,
	o.shippriority
from
	csv.public.customer c ,
	hive.default.orders o,
	hive.default.lineitem l
where
	c.mktsegment = 'MACHINERY'
	and c.custkey = o.custkey
	and l.orderkey = o.orderkey
	and o.orderdate < date '1995-03-18'
	and l.shipdate > date '1995-03-18'
group by
	l.orderkey,
	o.orderdate,
	o.shippriority
order by
	revenue desc,
	o.orderdate
```

Query 2

```
select
	supp_nation,
	cust_nation,
	l_year,
	sum(volume) as revenue
from
	(
		select
			n1.name as supp_nation,
			n2.name as cust_nation,
			extract(year from l.shipdate) as l_year,
			l.extendedprice * (1 - l.discount) as volume
		from
			hive.default.supplier s ,
			hive.default.lineitem l,
			hive.default.orders o,
			csv.public.customer c ,
```

```
                hive.default.nation n1,
                hive.default.nation n2
        where
                s.suppkey = l.suppkey
                and o.orderkey = l.orderkey
                and cast(c.custkey as bigint) = o.custkey
                and s.nationkey = n1.nationkey
                and c.nationkey = n2.nationkey
                and (
                        (n1.name = 'UNITED KINGDOM' and n2.name = 'ARGENTINA')
                        or (n1.name = 'ARGENTINA' and n2.name = 'UNITED KINGDOM')
                )
                and l.shipdate between date '1995-01-01' and date '1996-12-31'
    ) as shipping
group by
    supp_nation,
    cust_nation,
    l_year
order by
    supp_nation,
    cust_nation,
    l_year
```

Query 3

```
select
    c_count,
    count(*) as custdist
from
    (
        select
                cast(c.custkey as bigint),
                count(o.orderkey)
        from
                csv.public.customer c  left outer join hive.default.orders o on
                        cast(c.custkey as bigint) = o.custkey
                        and o.comment not like '%special%accounts%'
        group by
                cast(c.custkey as bigint)
    ) as c_orders (c_custkey, c_count)
group by
    c_count
order by
    custdist desc,
    c_count desc
```

END-OF-REPORT