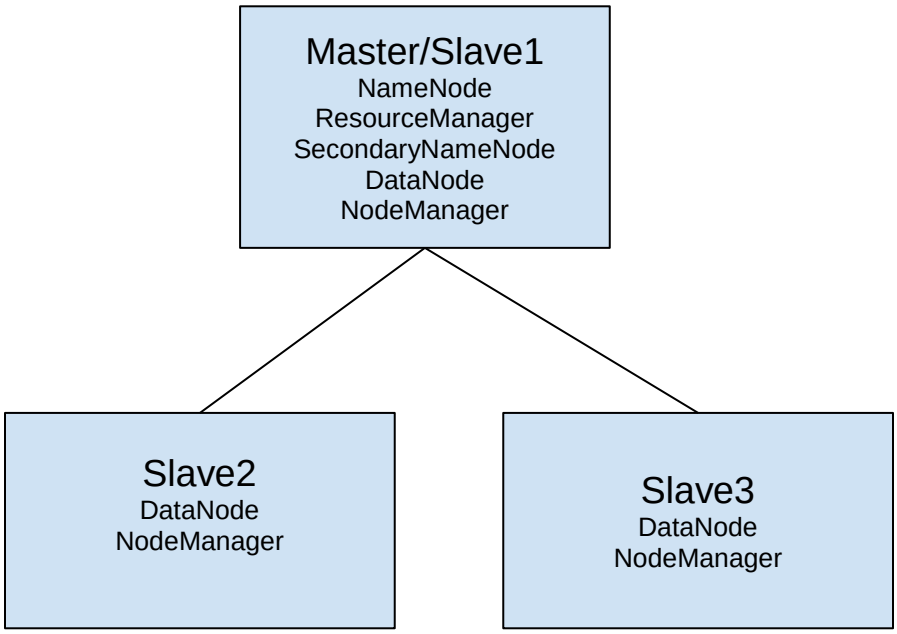
Background

This document explains the TPCH report of CarbonData (1.5.2 version) and Parquet on Spark 2.3.2 execution engine.

Hardware

CPU: Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz - 48 CPU
Memory: 378 GB DDR4 RAM
Hard Disk: 12 x 4 TB SATA 7200 RPM HDD



Configurations

Carbon Properties

```
# Number of cores used while loading per executor, Used 12 cores while loading the data.
carbon.number.of.cores.while.loading = 12

# Unsafe working memory per executor used during data loading.
carbon.unsafe.working.memory.in.mb=5120

# Disabling the push-down filter.
carbon.push.rowfilters.for.vector=false
```

Spark Conf

```
# Yarn overhead memory to facilitate offheap memory used by Spark and Carbon
spark.yarn.executor.memoryOverhead=20480
```

Data Loading Spark Configurations

```
# Number of cores used per executor
executor-cores 20

# Number of executors used in cluster
num-executors 3

# Total executor memory used per executor
executor-memory 140G

# Driver memory
driver-memory 25G
```

Query Spark Configurations

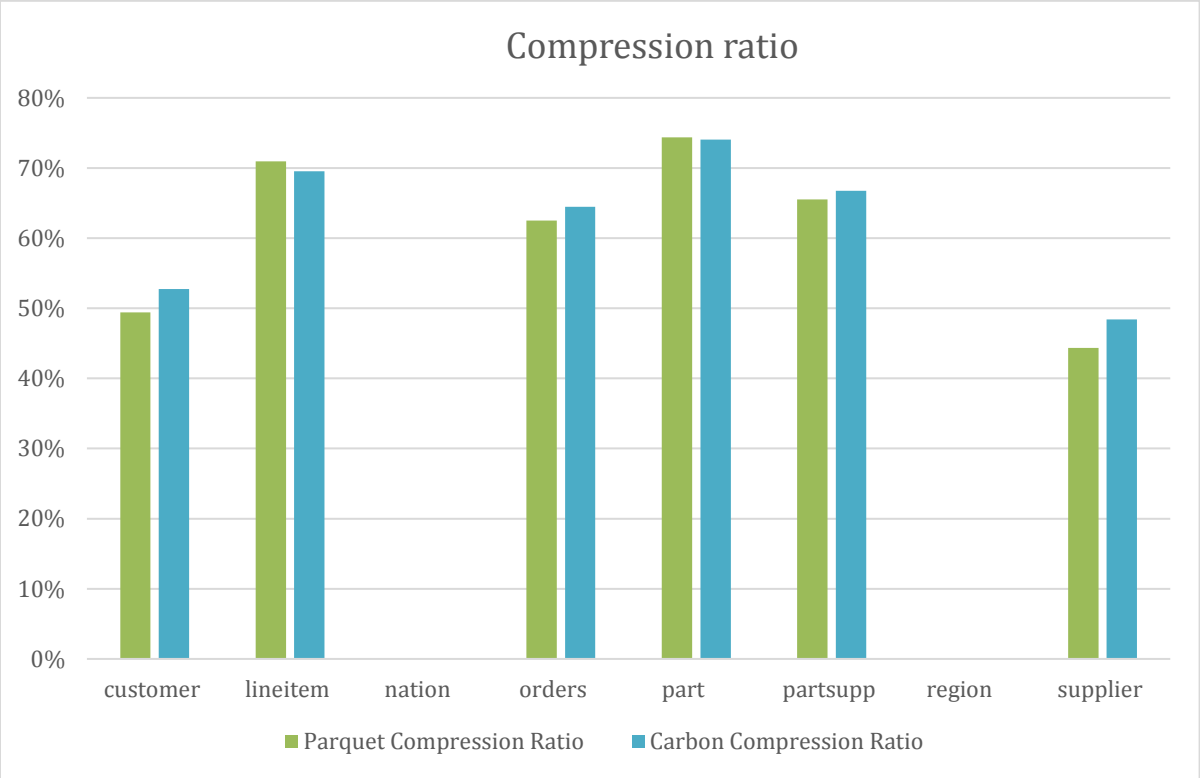
```
# Number of cores used per executor
executor-cores 5

# Number of executors used in cluster
num-executors 18

# Total executor memory used per executor
executor-memory 25G

# Driver memory
driver-memory 15G
```

Compression Ratio

The following chart depicts the compression ratio between Carbon and Parquet.

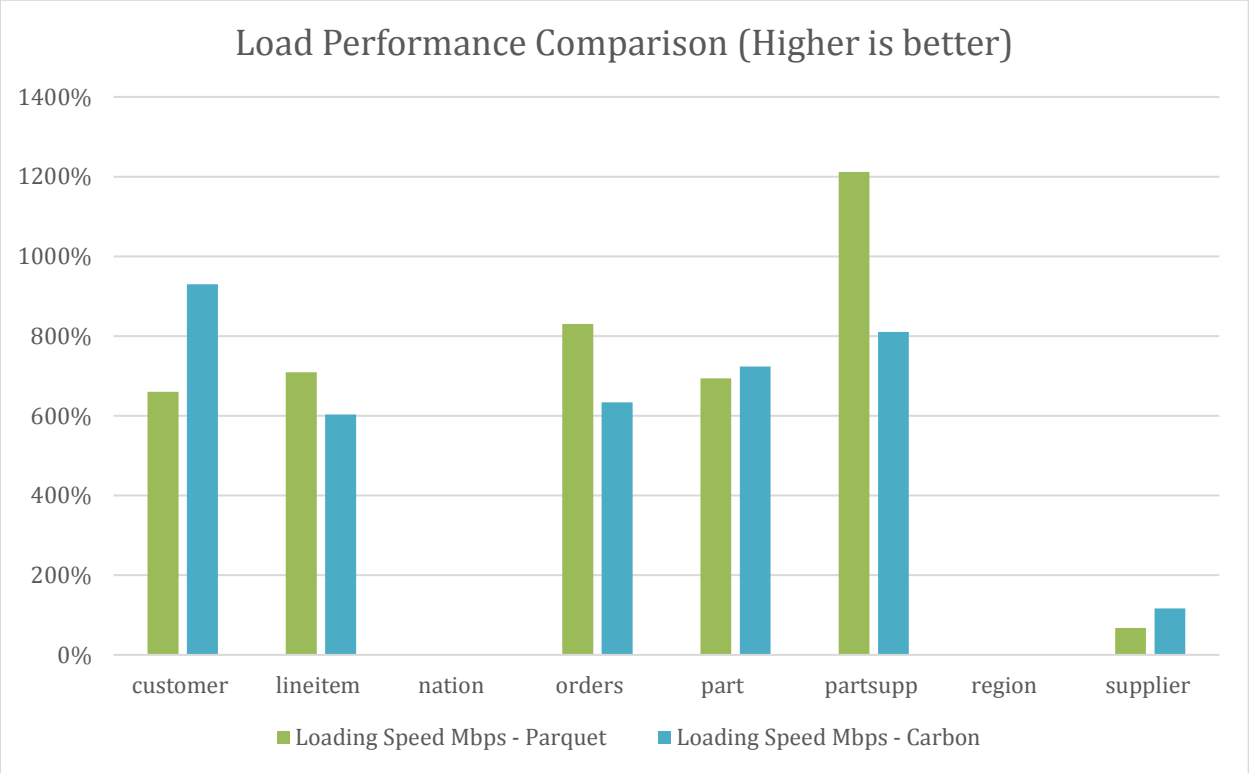
Formula

```
Compression Ratio = ((O - S)/S) * 100
O = Raw Data Size
S = Store size (Parquet or Carbon)
```



Loading Performance

The following chart depicts the loading performance between carbon and parquet. It is shown in MB per second per each node. (Higher the value, better the loading performance)

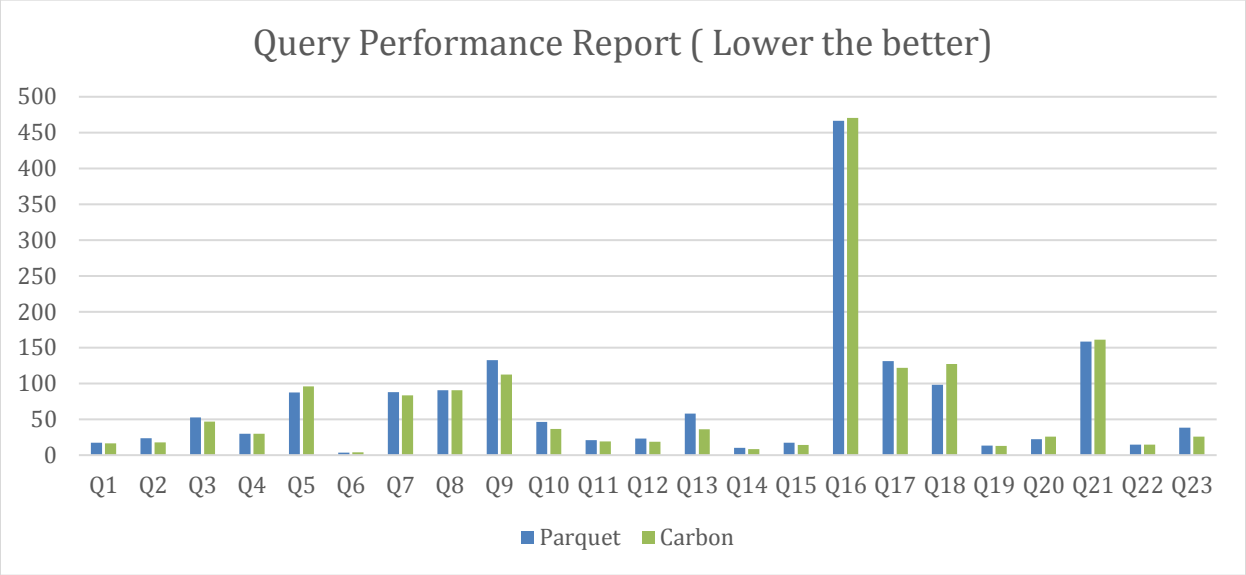


Query Performance

The following chart depicts the performance of Carbon and Parquet. To have a fair comparison we have loaded Carbon with no sort option and Parquet loaded directly.

How it is tested

Executed each query three times and taken best out of it in both Parquet and Carbon.



Queries	Parquet	CarbonData
Q1	17.217	16.77
Q2	31.184	20.54
Q3	55.334	52.749
Q4	31.423	31.668
Q5	88.642	90.322
Q6	4.903	4.322
Q7	90.891	92.185
Q8	100.928	93.316
Q9	135.525	123.084
Q10	49.596	38.276
Q11	20.312	21.361
Q12	22.43	26.175
Q13	51.691	36.9
Q14	10.884	9.304
Q15	21.91	18.227
Q16	453.201	454
Q17	131.208	124.595
Q18	103.995	121.199
Q19	14.004	14.015

Q20	24.898	23.94
Q21	166.414	158.301
Q22	15.959	13.889
Q23(Full Scan Query)	36.54	25.591

Scripts and data

Data Size : 500 GB (Generated using https://github.com/electrum/tpch-dbgen)

Parquet Create Table Script

```
CREATE TABLE LINEITEM (L_ORDERKEY BIGINT,L_PARTKEY BIGINT,L_SUPPKEY BIGINT, L_LINENUMBER INTEGER,L_QUANTITY double,L_EXTENDEDPRICE double,L_DISCOUNT double,L_TAX double,L_RETURNFLAG string,L_LINESTATUS string,L_SHIPDATE DATE,L_COMMITDATE DATE,L_RECEIPTDATE DATE,L_SHIPINSTRUCT string,L_SHIPMODE string,L_COMMENT string) using parquet options('parquet.compression'='snappy');


CREATE TABLE IF NOT EXISTS customer ( c_custkey BIGINT, c_name STRING, c_address STRING, c_nationkey INT, c_phone STRING, c_acctbal double, c_mktsegment STRING, c_comment STRING) using parquet options('parquet.compression'='snappy');


CREATE TABLE IF NOT EXISTS nation ( n_nationkey INT, n_name STRING, n_regionkey INT, n_comment STRING) using parquet options('parquet.compression'='snappy');


CREATE TABLE IF NOT EXISTS orders ( o_orderkey BIGINT, o_custkey BIGINT, o_orderstatus STRING, o_totalprice double, o_orderdate date, o_orderpriority STRING, o_clerk STRING, o_shippriority INT, o_comment STRING) using parquet options('parquet.compression'='snappy');


CREATE TABLE IF NOT EXISTS part ( p_partkey BIGINT, p_name STRING, p_mfgr STRING, p_brand STRING, p_type STRING, p_size INT, p_container STRING, p_retailprice double, p_comment STRING ) using parquet options('parquet.compression'='snappy');


CREATE TABLE IF NOT EXISTS partsupp ( ps_partkey BIGINT, ps_suppkey BIGINT, ps_availqty INT, ps_supplycost double, ps_comment STRING ) using parquet options('parquet.compression'='snappy');


CREATE TABLE IF NOT EXISTS region (r_regionkey INT, r_name STRING, r_comment STRING ) using parquet options('parquet.compression'='snappy');


CREATE TABLE IF NOT EXISTS supplier ( s_suppkey BIGINT, s_name STRING, s_address STRING, s_nationkey INT, s_phone STRING, s_acctbal double, s_comment STRING ) using parquet options('parquet.compression'='snappy');
```

Carbon Create Table Scripts

```
CREATE TABLE LINEITEM (L_ORDERKEY BIGINT,L_PARTKEY BIGINT,L_SUPPKEY
BIGINT, L_LINENUMBER INTEGER,L_QUANTITY double,L_EXTENDEDPRICE
double,L_DISCOUNT double,L_TAX double,L_RETURNFLAG string,L_LINESTATUS
string,L_SHIPDATE DATE,L_COMMITDATE DATE,L_RECEIPTDATE
DATE,L_SHIPINSTRUCT string,L_SHIPMODE string,L_COMMENT string) STORED BY
'carbondata'
TBLPROPERTIES('SORT_COLUMNS'='','table_blocklet_size'='90','table_blocksize'='120');

CREATE TABLE IF NOT EXISTS customer ( c_custkey BIGINT, c_name STRING, c_address
STRING, c_nationkey INT, c_phone STRING, c_acctbal double, c_mktsegment STRING,
c_comment STRING) STORED BY 'carbondata'
TBLPROPERTIES('SORT_COLUMNS'='','table_blocklet_size'='90','table_blocksize'='120');

CREATE TABLE IF NOT EXISTS nation ( n_nationkey INT, n_name STRING, n_regionkey
INT, n_comment STRING) STORED BY 'carbondata'
TBLPROPERTIES('SORT_COLUMNS'='','table_blocklet_size'='90','table_blocksize'='120');

CREATE TABLE IF NOT EXISTS orders ( o_orderkey BIGINT, o_custkey BIGINT,
o_orderstatus STRING, o_totalprice double, o_orderdate date, o_orderpriority STRING,
o_clerk STRING, o_shippriority INT, o_comment STRING) STORED BY 'carbondata'
TBLPROPERTIES('SORT_COLUMNS'='','table_blocklet_size'='90','table_blocksize'='120');

CREATE TABLE IF NOT EXISTS part ( p_partkey BIGINT, p_name STRING, p_mfgr
STRING, p_brand STRING, p_type STRING, p_size INT, p_container STRING, p_retailprice
double, p_comment STRING ) STORED BY 'carbondata'
TBLPROPERTIES('SORT_COLUMNS'='','table_blocklet_size'='90','table_blocksize'='120');

CREATE TABLE IF NOT EXISTS partsupp ( ps_partkey BIGINT, ps_suppkey BIGINT,
ps_availqty INT, ps_supplycost double, ps_comment STRING )STORED BY 'carbondata'
TBLPROPERTIES('SORT_COLUMNS'='','table_blocklet_size'='90','table_blocksize'='120');

CREATE TABLE IF NOT EXISTS region (r_regionkey INT, r_name STRING, r_comment
STRING )STORED BY 'carbondata'
TBLPROPERTIES('SORT_COLUMNS'='','table_blocklet_size'='90','table_blocksize'='120');

CREATE TABLE IF NOT EXISTS supplier ( s_suppkey BIGINT, s_name STRING, s_address
STRING, s_nationkey INT, s_phone STRING, s_acctbal double, s_comment STRING )
STORED BY 'carbondata'
TBLPROPERTIES('SORT_COLUMNS'='','table_blocklet_size'='90','table_blocksize'='120');
```

TPCH Queries

```
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 <=date( '1998-09-02') group by l_returnflag, l_linestatus order by
l_returnflag, l_linestatus;

select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_comment from
part, supplier, partsupp, nation, region where p_partkey = ps_partkey and s_suppkey =
ps_suppkey and p_size = 15 and p_type like '%BRASS' and s_nationkey = n_nationkey and
n_regionkey = r_regionkey and r_name = 'EUROPE' and ps_supplycost = ( select
```

min(ps_supplycost) from partsupp, supplier,nation, region where p_partkey = ps_partkey and s_suppkey = ps_suppkey and s_nationkey = n_nationkey and n_regionkey = r_regionkey and r_name = 'EUROPE') order by s_acctbal desc, n_name, s_name, p_partkey limit 100;

select l_orderkey, sum(l_extendedprice * (1 - l_discount)) as revenue, o_orderdate, o_shippriority from customer, orders, lineitem where c_mktsegment = 'BUILDING' and c_custkey = o_custkey and l_orderkey = o_orderkey and o_orderdate < date('1995-03-15') and l_shipdate > date('1995-03-15') group by l_orderkey, o_orderdate, o_shippriority order by revenue desc, o_orderdate limit 10;

select o_orderpriority, count(*) as order_count from orders where o_orderdate >= date('1993-07-01') and o_orderdate < date('1993-10-01') and exists (select * from lineitem where l_orderkey = o_orderkey and l_commitdate < l_receiptdate) group by o_orderpriority order by o_orderpriority;

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 >=date('1994-01-01') and o_orderdate < date('1995-01-01') group by n_name order by revenue desc;

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

select supp_nation, cust_nation, l_year, sum(volume) as revenue from (select n1.n_name as supp_nation, n2.n_name as cust_nation, year(l_shipdate) as l_year, l_extendedprice * (1 - l_discount) as volume from supplier,lineitem,orders,customer,nation n1,nation n2 where s_suppkey = l_suppkey and o_orderkey = l_orderkey and c_custkey = o_custkey and s_nationkey = n1.n_nationkey and c_nationkey = n2.n_nationkey and ((n1.n_name = 'FRANCE' and n2.n_name = 'GERMANY') or (n1.n_name = 'GERMANY' and n2.n_name = 'FRANCE')) 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;

select o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(volume) as mkt_share from (select year(o_orderdate) as o_year, l_extendedprice * (1-l_discount) as volume, n2.n_name as nation from part,supplier,lineitem,orders,customer,nation n1,nation n2,region where p_partkey = l_partkey and s_suppkey = l_suppkey and l_orderkey = o_orderkey and o_custkey = c_custkey and c_nationkey = n1.n_nationkey and n1.n_regionkey = r_regionkey and r_name = 'AMERICA' and s_nationkey = n2.n_nationkey and o_orderdate between date('1995-01-01') and date('1996-12-31') and p_type = 'ECONOMY ANODIZED STEEL') as all_nations group by o_year order by o_year;

select nation, o_year, sum(amount) as sum_profit from (select n_name as nation, year(o_orderdate) as o_year, l_extendedprice * (1 - l_discount) - ps_supplycost * l_quantity as amount from part, supplier, lineitem, partsupp, orders, nation 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 '%green%') as profit group by nation, o_year order by nation, o_year desc;

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, orders, lineitem, nation where c_custkey = o_custkey and l_orderkey = o_orderkey and o_orderdate >= date('1993-10-01') and o_orderdate < date('1994-01-01') 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 limit 20;

select ps_partkey, sum(ps_supplycost * ps_availqty) as value from partsupp, supplier, nation

```
where ps_suppkey = s_suppkey and s_nationkey = n_nationkey and n_name = 'GERMANY'
group by ps_partkey having sum(ps_supplycost * ps_availqty) > ( select sum(ps_supplycost *
ps_availqty) * 0.0001000000 s from partsupp, supplier, nation where ps_suppkey =
s_suppkey and s_nationkey = n_nationkey and n_name = 'GERMANY' ) order by value desc;

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 in ('MAIL', 'SHIP') and l_commitdate <
l_receiptdate and l_shipdate < l_commitdate and l_receiptdate >= date('1994-01-01') and
l_receiptdate < date('1995-01-01') group by l_shipmode order by l_shipmode;

select c_count, count(*) as custdist from (select c_custkey, count(o_orderkey) as c_count
from customer left outer join orders on ( c_custkey = o_custkey and o_comment not like
'%special%requests%' ) group by c_custkey ) as c_orders group by c_count order by custdist
desc, c_count desc;

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 >= date('1995-09-01') and l_shipdate <
date('1995-10-01');

select s_suppkey, s_name, s_address, s_phone, total_revenue from supplier,revenue where
s_suppkey = supplier_no and total_revenue = ( select max(total_revenue) from revenue )
order by s_suppkey;

select p_brand, p_type, p_size, count(distinct ps_suppkey) as supplier_cnt from partsupp,
part where p_partkey = ps_partkey and p_brand <> 'Brand#45' and p_type not like 'MEDIUM
POLISHED%' and p_size in (49, 14, 23, 45, 19, 3, 36, 9) and ps_suppkey not in ( select
s_suppkey from supplier 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;
select sum(l_extendedprice) / 7.0 as avg_yearly from lineitem,part where p_partkey =
l_partkey and p_brand = 'Brand#23' and p_container = 'MED BOX' and l_quantity < ( select
0.2 * avg(l_quantity) from lineitem where l_partkey = p_partkey );

select c_name, c_custkey, o_orderkey, o_orderdate, o_totalprice, sum(l_quantity) from
customer, orders, lineitem where o_orderkey in ( select l_orderkey from lineitem group by
l_orderkey having sum(l_quantity) > 300 ) 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;

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' );

select s_name, s_address from supplier, nation where s_suppkey in ( select ps_suppkey from
partsupp where ps_partkey in ( select p_partkey from part where p_name like 'forest%' ) and
ps_availqty > ( select 0.5 * sum(l_quantity) from lineitem where l_partkey = ps_partkey and
l_suppkey = ps_suppkey and l_shipdate >= date('1994-01-01') and l_shipdate < date('1995-
01-01') ) ) and s_nationkey = n_nationkey and n_name = 'CANADA' order by s_name;
```

```
select s_name, count(*) as numwait from supplier, lineitem l1, orders, nation where
s_suppkey = l1.l_suppkey and o_orderkey = l1.l_orderkey and o_orderstatus = 'F' and
l1.l_receiptdate > l1.l_commitdate and exists ( select * from lineitem l2 where l2.l_orderkey =
l1.l_orderkey and l2.l_suppkey <> l1.l_suppkey ) and not exists ( select * from lineitem l3
where l3.l_orderkey = l1.l_orderkey and l3.l_suppkey <> l1.l_suppkey and l3.l_receiptdate >
l3.l_commitdate ) and s_nationkey = n_nationkey and n_name = 'SAUDI ARABIA' group by
s_name order by numwait desc, s_name;

select cntrycode, count(*) as numcust, sum(c_acctbal) as totacctbal from ( select
substring(c_phone,1 ,2) as cntrycode, c_acctbal from customer where
substring(c_phone ,1,2) in ('13','31','23','29','30','18','17') and c_acctbal > ( select
avg(c_acctbal) from customer where c_acctbal > 0.00 and substring(c_phone,1,2) in ('13',
'31', '23', '29', '30', '18', '17') ) and not exists ( select * from orders where o_custkey =
c_custkey ) ) as custsale group by cntrycode order by cntrycode;

select count(l_shipdate), count(l_shipinstruct), count(l_orderkey), count(l_suppkey),
count(l_quantity), count(l_partkey), count(l_receiptdate), count(l_commitdate),
count(l_comment), count(l_discount), count(l_linenumber), count(L_RETURNFLAG),
count(L_LINESTATUS), count(l_shipmode) from lineitem;
```