TPC-DS Benchmark Set

TPC-DS Benchmark

The TPC Benchmark DS (TPC-DS) is a decision support benchmark that models several generally applicable aspects of a decision support system, including queries and data maintenance. Although the underlying business model of TPC-DS is a retail product supplier, the database schema, data population, queries, data maintenance model and implementation rules have been designed to be broadly representative of modern decision support systems. (http://www.tpc.org/tpcds/)

This benchmark illustrates decision support systems that:

- Examine large volumes of data
- Give answers to real-world business questions
- Execute queries of various operational requirements and complexities (e.g., ad-hoc, reporting, iterative OLAP, data mining)
- Are characterized by high CPU and IO load
- Are periodically synchronized with source OLTP databases through database maintenance functions

Official scale factors of datasets are 100GB, 300GB, 1TB, 3TB, 10TB, 30TB and 100TB.

DDL for TPC-DS datasets

The TPC-DS schema models the sales and sales returns process for an organization that employs three primary sales channels: store, catalogs, and the Internet. The schema includes 7 fact tables and 17 dimension tables.

- **Fact tables**: store_sales, store_returns, catalog_sales, catalog_returns, web_sales, web_returns, inventory
- **Dimension tables**: store, call_center, catalog_page, web_site, web_page, warehouse, customer, customer_address, customer_demographics, date_dim, household_demographics, item, income_band, promotion, reason, ship_mode, time_dim

The following DDL statements are for them in alphabetical order.

```sql
create external table call_center (
    cc_call_center_sk bigint,
    cc_call_center_id text,
    cc_rec_start_date date,
    cc_rec_end_date date,
    cc_closed_date_sk bigint,
    cc_open_date_sk bigint,
    cc_name text,
    cc_class text,
    cc_employees bigint,
    cc_sq_ft bigint,
    cc_hours text,
    cc_manager text,
    cc_mkt_id bigint,
    cc_mkt_class text,
    cc_mkt_desc text,
    cc_market_manager text,
    cc_division bigint,
    cc_division_name text,
    cc_company bigint,
    cc_company_name text,
    cc_street_number text,
    cc_street_name text,
    ...
);```
cc_street_type text,
cc_suite_number text,
cc_city text,
cc_county text,
cc_state text,
cc_zip text,
cc_country text,
cc_gmt_offset double,
cc_tax_percentage double
)
using text with ('text.delimiter'='|') location 'hdfs://x/y/call_center';

create external table catalog_page (  
cp_catalog_page_sk bigint,
cp_catalog_page_id text,
cp_start_date_sk bigint,
cp_end_date_sk bigint,
cp_department text,
cp_catalog_number bigint,
cp_catalog_page_number bigint,
cp_description text,
cp_type text  
)
using text with ('text.delimiter'='|') location 'hdfs://x/y/catalog_page';

create external table catalog_returns (  
  cr_returned_date_sk bigint,
cr_returned_time_sk bigint,
cr_item_sk bigint,
cr_refunded_customer_sk bigint,
cr_refunded_cdemo_sk bigint,
cr_refunded_hdemo_sk bigint,
cr_refunded_addr_sk bigint,
cr_calling_customer_sk bigint,
cr_returning_customer_sk bigint,
cr_returning_cdemo_sk bigint,
cr_returning_hdemo_sk bigint,
cr_returning_addr_sk bigint,
cr_call_center_sk bigint,
cr_catalog_page_sk bigint,
cr_ship_mode_sk bigint,
cr_warehouse_sk bigint,
cr_reason_sk bigint,
cr_order_number bigint,
cr_return_quantity bigint,
cr_return_amount double,
cr_return_tax double,
cr_return_amt_inc_tax double,
cr_fee double,
cr_return_ship_cost double,
cr_refunded_cash double,
cr_reversed_charge double,
cr_store_credit double,
cr_net_loss double
using text with ('text.delimiter'='|') location 'hdfs://x/y/catalog_returns';

cREATE EXTERNAL TABLE CATALOG_SALES (  
c_sold_date_sk bigint,  
c_sold_time_sk bigint,  
c_ship_date_sk bigint,  
c_bill_customer_sk bigint,  
c_bill_cdemo_sk bigint,  
c_bill_hdemo_sk bigint,  
c_bill_addr_sk bigint,  
c_ship_customer_sk bigint,  
c_ship_cdemo_sk bigint,  
c_ship_hdemo_sk bigint,  
c_ship_addr_sk bigint,  
c_call_center_sk bigint,  
c_catalog_page_sk bigint,  
c_ship_mode_sk bigint,  
c_warehouse_sk bigint,  
c_item_sk bigint,  
c_promo_sk bigint,  
c_order_number bigint,  
c_quantity bigint,  
c_wholesale_cost double,  
c_list_price double,  
c_sales_price double,  
c_ext_discount_amt double,  
c_ext_sales_price double,  
c_ext_wholesale_cost double,  
c_ext_list_price double,  
c_ext_tax double,  
c_coupon_amt double,  
c_ext_ship_cost double,  
c_net_paid double,  
c_net_paid_inc_tax double,  
c_net_paid_inc_ship double,  
c_net_paid_inc_ship_tax double,  
c_net_profit double
)
using text with ('text.delimiter'='|') location 'hdfs://x/y/catalog_sales';

CREATE EXTERNAL TABLE CUSTOMER (  
customer_sk bigint,  
customer_id text,  
c_current_cdemo_sk bigint,  
c_current_hdemo_sk bigint,  
c_current_addr_sk bigint,  
c_first_shipto_date_sk bigint,  
c_first_sales_date_sk bigint,  
salutation text,  
c_first_name text,  
last_name text,
create external table customer (  
c_customer_flag text,  
c_birth_day bigint,  
c_birth_month bigint,  
c_birth_year bigint,  
c_birth_country text,  
c_login text,  
c_email_address text,  
c_last_review_date_sk bigint  
)  
using text with ('text.delimiter'='|') location 'hdfs://x/y/customer';

create external table customer_address (  
ca_address_sk bigint,  
ca_address_id text,  
ca_street_number text,  
ca_street_name text,  
ca_street_type text,  
ca_suite_number text,  
ca_city text,  
ca_county text,  
ca_state text,  
ca_zip text,  
ca_country text,  
ca_gmt_offset double,  
ca_location_type text  
)  
using text with ('text.delimiter'='|') location 'hdfs://x/y/customer_address';

create external table customer_demographics (  
cd_demo_sk bigint,  
cd_gender text,  
cd_marital_status text,  
cd_education_status text,  
cd_purchase_estimate bigint,  
cd_credit_rating text,  
cd_dep_count bigint,  
cd_dep_employed_count bigint,  
cd_dep_college_count bigint  
)  
using text with ('text.delimiter'='|') location 'hdfs://x/y/customer_demographics';

create external table date_dim (  
d_date_sk bigint,  
d_date_id text,  
d_date date,  
d_month_seq bigint,  
d_week_seq bigint,  
d_quarter_seq bigint,  
d_year bigint,  
d_dow bigint,  
d_moy bigint,
create external table date_dim (  
d_dom bigint,  
d_qoy bigint,  
d_fy_year bigint,  
d_fy_quarter_seq bigint,  
d_fy_week_seq bigint,  
d_day_name text,  
d_quarter_name text,  
d_holiday text,  
d_weekend text,  
d_following_holiday text,  
d_first_dom bigint,  
d_last_dom bigint,  
d_same_day_ly bigint,  
d_same_day_lq bigint,  
d_current_day text,  
d_current_week text,  
d_current_month text,  
d_current_quarter text,  
d_current_year text  
)  
using text with ('text.delimiter'='|') location 'hdfs://x/y/date_dim';

create external table household_demographics (  
hd_demo_sk bigint,  
hd_income_band_sk bigint,  
hd_buy_potential text,  
hd_dep_count bigint,  
hd_vehicle_count bigint  
)  
using text with ('text.delimiter'='|') location 'hdfs://x/y/household_demographics';

create external table income_band (  
ib_income_band_sk bigint,  
ib_lower_bound bigint,  
ib_upper_bound bigint  
)  
using text with ('text.delimiter'='|') location 'hdfs://x/y/income_band';

create external table inventory (  
inv_date_sk bigint,  
inv_item_sk bigint,  
inv_warehouse_sk bigint,  
inv_quantity_on_hand bigint  
)  
using text with ('text.delimiter'='|') location 'hdfs://x/y/inventory';

create external table item (  
i_item_sk bigint,  
i_item_id text,  
i_rec_start_date date,  
i_rec_end_date date,  
i_item_desc text,
create external table item (  
i_current_price double,  
i_wholesale_cost double,  
i_brand_id bigint,  
i_brand text,  
i_class_id bigint,  
i_class text,  
i_category_id bigint,  
i_category text,  
i_manufact_id bigint,  
i_manufact text,  
i_size text,  
i_formulation text,  
i_color text,  
i_units text,  
i_container text,  
i_manager_id bigint,  
i_product_name text  )  
using text with ('text.delimiter'='|') location 'hdfs://x/y/item';

create external table promotion (  
p_promo_sk bigint,  
p_promo_id text,  
p_start_date_sk bigint,  
p_end_date_sk bigint,  
p_item_sk bigint,  
p_cost double,  
p_response_target bigint,  
p_promo_name text,  
p_channel_dmail text,  
p_channel_email text,  
p_channel_catalog text,  
p_channel_tv text,  
p_channel_radio text,  
p_channel_press text,  
p_channel_event text,  
p_channel_demo text,  
p_channel_details text,  
p_purpose text,  
p_discount_active text  )  
using text with ('text.delimiter'='|') location 'hdfs://x/y/promotion';

create external table reason (  
r_reason_sk bigint,  
r_reason_id text,  
r_reason_desc text  )  
using text with ('text.delimiter'='|') location 'hdfs://x/y/reason';

create external table ship_mode (  
sm_ship_mode_sk bigint,  
sm_ship_mode_id text,  
...
create external table store (
    s_store_sk bigint,
    s_store_id text,
    s_rec_start_date date,
    s_rec_end_date date,
    s_closed_date_sk bigint,
    s_store_name text,
    s_number_employees bigint,
    s_floor_space bigint,
    s_hours text,
    s_manager text,
    s_market_id bigint,
    s_geography_class text,
    s_market_desc text,
    s_market_manager text,
    s_division_id bigint,
    s_division_name text,
    s_company_id bigint,
    s_company_name text,
    s_street_number text,
    s_street_name text,
    s_street_type text,
    s_suite_number text,
    s_city text,
    s_county text,
    s_state text,
    s_zip text,
    s_country text,
    s_gmt_offset double,
    s_tax_precentage double
) 
using text with ('text.delimiter'='|') location 'hdfs://x/y/store';

create external table store_returns (sr Returned_date_sk bigint,
    sr_return_time_sk bigint,
    sr_item_sk bigint,
    sr_customer_sk bigint,
    sr_cdemo_sk bigint,
    sr_hdemo_sk bigint,
    sr_addr_sk bigint,
    sr_store_sk bigint,
    sr_reason_sk bigint,
    sr_ticket_number bigint,
    sr_return_quantity bigint,
    sr_return_amt double,
```sql
create external table store_returns (  sr_return_tax double,  sr_return_amt_inc_tax double,  sr_fee double,  sr_return_ship_cost double,  sr_refunded_cash double,  sr_reversed_charge double,  sr_store_credit double,  sr_net_loss double )  using text with ('text.delimiter'='|') location 'hdfs://x/y/store_returns';
create external table store_sales (  ss_sold_date_sk bigint,  ss_sold_time_sk bigint,  ss_item_sk bigint,  ss_customer_sk bigint,  ss_cdemo_sk bigint,  ss_hdemo_sk bigint,  ss_addr_sk bigint,  ss_store_sk bigint,  ss_promo_sk bigint,  ss_ticket_number bigint,  ss_quantity bigint,  ss_wholesale_cost double,  ss_list_price double,  ss_sales_price double,  ss_ext_discount_amt double,  ss_ext_sales_price double,  ss_ext_wholesale_cost double,  ss_ext_list_price double,  ss_ext_tax double,  ss_coupon_amt double,  ss_net_paid double,  ss_net_paid_inc_tax double,  ss_net_profit double )  using text with ('text.delimiter'='|') location 'hdfs://x/y/store_sales';
create external table time_dim (  t_time_sk bigint,  t_time_id text,  t_time bigint,  t_hour bigint,  t_minute bigint,  t_second bigint,  t_am_pm text,  t_shift text,  t_sub_shift text,  t_meal_time text )  using text with ('text.delimiter'='|') location 'hdfs://x/y/time_dim';
create external table warehouse (  )
```
create external table web_warehouse (
    w_warehouse_sk bigint,
    w_warehouse_id text,
    w_warehouse_name text,
    w_warehouse_sq_ft bigint,
    w_street_number text,
    w_street_name text,
    w_street_type text,
    w_suite_number text,
    w_city text,
    w_county text,
    w_state text,
    w_zip text,
    w_country text,
    w_gmt_offset double
) using text with ('text.delimiter'='|') location 'hdfs://x/y/warehouse';

create external table web_page (  
    wp_web_page_sk bigint,
    wp_web_page_id text,
    wp_rec_start_date date,
    wp_rec_end_date date,
    wp_creation_date_sk bigint,
    wp_access_date_sk bigint,
    wp_autogen_flag text,
    wp_customer_sk bigint,
    wp_url text,
    wp_type text,
    wp_char_count bigint,
    wp_link_count bigint,
    wp_image_count bigint,
    wp_max_ad_count bigint
) using text with ('text.delimiter'='|') location 'hdfs://x/y/web_page';

create external table web_returns (  
    wr_returned_date_sk bigint,
    wr_returned_time_sk bigint,
    wr_item_sk bigint,
    wr_refunded_customer_sk bigint,
    wr_refunded_cdemo_sk bigint,
    wr_refunded_hdemo_sk bigint,
    wr_refunded_addr_sk bigint,
    wr_returning_customer_sk bigint,
    wr_returning_cdemo_sk bigint,
    wr_returning_hdemo_sk bigint,
    wr_returning_addr_sk bigint,
    wr_web_page_sk bigint,
    wr_reason_sk bigint,
    wr_order_number bigint,
    wr_return_quantity bigint,
    wr_return_amt double,
    wr_return_tax double,
create external table web_returns (  
  wr_return_amt_inc_tax double,  
  wr_fee double,  
  wr_return_ship_cost double,  
  wr_refunded_cash double,  
  wr_reversed_charge double,  
  wr_account_credit double,  
  wr_net_loss double  
)  
using text with ("text.delimiter"="|") location 'hdfs://x/y/web_returns';

create external table web_sales (  
  ws_sold_date_sk bigint,  
  ws_sold_time_sk bigint,  
  ws_ship_date_sk bigint,  
  ws_item_sk bigint,  
  ws_bill_customer_sk bigint,  
  ws_bill_cdemo_sk bigint,  
  ws_bill_hdemo_sk bigint,  
  ws_bill_addr_sk bigint,  
  ws_ship_customer_sk bigint,  
  ws_ship_cdemo_sk bigint,  
  ws_ship_hdemo_sk bigint,  
  ws_ship_addr_sk bigint,  
  ws_web_page_sk bigint,  
  ws_web_site_sk bigint,  
  ws_ship_mode_sk bigint,  
  ws_warehouse_sk bigint,  
  ws_promo_sk bigint,  
  ws_order_number bigint,  
  ws_quantity bigint,  
  ws_wholesale_cost double,  
  ws_list_price double,  
  ws_sales_price double,  
  ws_ext_discount_amt double,  
  ws_ext_sales_price double,  
  ws_ext_wholesale_cost double,  
  ws_ext_list_price double,  
  ws_ext_tax double,  
  ws_coupon_amt double,  
  ws_ext_ship_cost double,  
  ws_net_paid double,  
  ws_net_paid_inc_tax double,  
  ws_net_paid_inc_ship double,  
  ws_net_paid_inc_ship_tax double,  
  ws_net_profit double  
)  
using text with ("text.delimiter"="|") location 'hdfs://x/y/web_sales';

create external table web_site (  
  web_site_sk bigint,  
  web_site_id text,  
  web_rec_start_date date,  
  web_rec_end_date date,
TPC-DS provides 99 queries. Since Tajo does not support some features like WITH clause, ROLLUP operation, and STDDEV_SAMP function, I omit queries containing them. The following are 26 queries tested on Tajo.

Q3

```
select d_year, i_brand_id brand_id, i_brand brand, sum(ss_ext_sales_price) sum_agg
from date_dim, store_sales, item
where d_date_sk = ss_sold_date_sk and ss_item_sk = i_item_sk and
i_manufact_id = 436 and d_moy=12
group by d_year, i_brand, i_brand_id
order by d_year, sum_agg desc, brand_id
limit 100;
```
Q7

```sql
select i_item_id, avg(ss_quantity) agg1, avg(ss_list_price) agg2, avg(ss_coupon_amt) agg3, avg(ss_sales_price) agg4
from store_sales, customer_demographics, date_dim, item, promotion
where ss_sold_date_sk = d_date_sk and ss_item_sk = i_item_sk and
  ss_cdemo_sk = cd_demo_sk and ss_promo_sk = p_promo_sk and
  cd_gender = 'F' and cd_marital_status = 'W' and cd_education_status = 'Primary' and
  (p_channel_email = 'N' or p_channel_event = 'N') and d_year = 1998
  group by i_item_id
order by i_item_id
limit 100;
```

Q15

```sql
select ca_zip, sum(cs_sales_price)
from catalog_sales, customer, customer_address, date_dim
where cs_bill_customer_sk = c_customer_sk and c_current_addr_sk = ca_address_sk and
  (substr(ca_zip,1,5) in ('85669','86197','88274','83405','86475',
   '85392','85460','80348','81792')
   or ca_state in ('CA','WA','GA') or cs_sales_price > 500) and
  cs_sold_date_sk = d_date_sk and d_qoy = 2 and d_year = 2000
  group by ca_zip
order by ca_zip
limit 100;
```

Q19

```sql
select i_brand_id brand_id, i_brand brand, i_manufact_id, i_manufact, sum(ss_ext_sales_price) ext_price
from date_dim, store_sales, item, customer, customer_address, store
where d_date_sk = ss_sold_date_sk and ss_item_sk = i_item_sk and
  i_manager_id=7 and d_moy=11 and d_year=1999 and
  ss_customer_sk = c_customer_sk and c_current_addr_sk = ca_address_sk
and substr(ca_zip,1,5) <> substr(s_zip,1,5) and
  ss_store_sk = s_store_sk
  group by i_brand, i_brand_id, i_manufact_id, i_manufact
order by ext_price desc, i_brand, i_brand_id, i_manufact_id, i_manufact
limit 100;
```
Q21 (modified: '+ 30 days' --> '+ 30')

```sql
select *
from (select w_warehouse_name, i_item_id,
        sum(case when (cast(d_date as date) < cast ('1998-04-08' as date)) then inv_quantity_on_hand else 0 end) as inv_before,
        sum(case when (cast(d_date as date) >= cast ('1998-04-08' as date)) then inv_quantity_on_hand else 0 end) as inv_after
        from inventory, warehouse, item, date_dim
        where i_current_price between 0.99 and 1.49 and i_item_sk = inv_item_sk and inv_warehouse_sk = w_warehouse_sk and
        inv_date_sk = d_date_sk and d_date between (cast ('1998-04-08' as date) - 30) and (cast ('1998-04-08' as date) + 30)
        group by w_warehouse_name, i_item_id) x
where (case when inv_before > 0 then inv_after / inv_before else null end) between 2.0/3.0 and 3.0/2.0
order by w_warehouse_name, i_item_id
limit 100;
```

Q25

```sql
select i_item_id, i_item_desc, s_store_id, s_store_name,
        sum(ss_net_profit) as store_sales_profit, sum(sr_net_loss) as store_returns_loss, sum(cs_net_profit) as catalog_sales_profit
from store_sales, store_returns, catalog_sales, date_dim d1, date_dim d2, date_dim d3, store, item
where d1.d_moy = 4 and d1.d_year = 2000 and d1.d_date_sk = ss_sold_date_sk
and
        i_item_sk = ss_item_sk and s_store_sk = ss_store_sk and
        ss_customer_sk = sr_customer_sk and ss_item_sk = sr_item_sk and
        ss_ticket_number = sr_ticket_number and
        sr_returned_date_sk = d2.d_date_sk and d2.d_moy between 4 and 10 and d2.d_year = 2000 and
        sr_customer_sk = cs_bill_customer_sk and sr_item_sk = cs_item_sk and
        cs_sold_date_sk = d3.d_date_sk and d3.d_moy between 4 and 10 and d3.d_year = 2000
        group by i_item_id, i_item_desc, s_store_id, s_store_name
order by i_item_id, i_item_desc, s_store_id, s_store_name
limit 100;
```
Q26

select i_item_id, avg(cs_quantity) agg1, avg(cs_list_price) agg2, avg(cs_coupon_amt) agg3, avg(cs_sales_price) agg4
from catalog_sales, customer_demographics, date_dim, item, promotion
where cs_sold_date_sk = d_date_sk and cs_item_sk = i_item_sk and
cs_bill_cdemo_sk = cd_demo_sk and
    cs_promo_sk = p_promo_sk and cd_gender = 'F' and cd_marital_status = 'W' and cd_education_status = 'Primary' and
    (p_channel_email = 'N' or p_channel_event = 'N') and d_year = 1998
group by i_item_id
order by i_item_id
limit 100;

Q29

select i_item_id, i_item_desc, s_store_id, s_store_name,
    sum(ss_quantity) as store_sales_quantity,
    sum(sr_return_quantity) as store_returns_quantity,
    sum(cs_quantity) as catalog_sales_quantity
from store_sales, store_returns, catalog_sales, date_dim d1, date_dim d2, date_dim d3, store, item
where d1.d_moy = 4 and d1.d_year = 1999 and d1.d_date_sk = ss_sold_date_sk
and i_item_sk = ss_item_sk and
    s_store_sk = ss_store_sk and ss_customer_sk = sr_customer_sk and
    ss_item_sk = sr_item_sk and
    ss_ticket_number = sr_ticket_number and sr_returned_date_sk = d2.d_date_sk and d2.d_moy between 4 and 4 + 3 and
    d2.d_year = 1999 and sr_customer_sk = cs_bill_customer_sk and
    sr_item_sk = cs_item_sk and
    cs_sold_date_sk = d3.d_date_sk and d3.d_year in (1999,1999+1,1999+2)
group by i_item_id, i_item_desc, s_store_id, s_store_name
order by i_item_id, i_item_desc, s_store_id, s_store_name
limit 100;
Q34

```
select c_last_name, c_first_name, c_salutation, c_preferred_cust_flag, ss_ticket_number, cnt
from (select ss_ticket_number, ss_customer_sk, count(*) cnt
    from store_sales, date_dim, store, household_demographics
    where store_sales.ss_sold_date_sk = date_dim.d_date_sk and
    store_sales.ss_store_sk = store.s_store_sk and
    store_sales.ss_hdemo_sk = household_demographics.hd_demo_sk and
    (date_dim.d_dom between 1 and 3 or date_dim.d_dom between 25 and 28) and
    (household_demographics.hd_buy_potential = '10000' or household_demographics.hd_buy_potential = 'unknown') and
    household_demographics.hd_vehicle_count > 0 and
    (case when household_demographics.hd_vehicle_count > 0 then household_demographics.hd_dep_count/
    household_demographics.hd_vehicle_count
    else null end) > 1.2 and
    date_dim.d_year in (1998, 1998+1, 1998+2) and
    store.s_county in ('Williamson County', 'Williamson County', 'Williamson County', 'Williamson County', 'Williamson County', 'Williamson County', 'Williamson County')
group by ss_ticket_number, ss_customer_sk) dn, customer
where ss_customer_sk = c_customer_sk and cnt between 15 and 20
order by c_last_name, c_first_name, c_salutation, c_preferred_cust_flag
desc;
```

Q37 (modified: '+ 60 days' --> '+ 60')

```
select i_item_id, i_item_desc, i_current_price
from item, inventory, date_dim, catalog_sales
where i_current_price between 22 and 22 + 30 and inv_item_sk = i_item_sk
and d_date_sk=inv_date_sk and
    d_date between cast('2001-06-02' as date) and (cast('2001-06-02' as date) + 60) and
    i_manufact_id in (678, 964, 918, 849) and inv_quantity_on_hand between 100 and 500 and cs_item_sk = i_item_sk
group by i_item_id, i_item_desc, i_current_price
order by i_item_id
limit 100;
```
Q40 (modified: ‘30 days’ --> ‘30’)  
select w_state, i_item_id,  
    sum(case when (cast(d_date as date) < cast ('1998-04-08' as date))  
    then cs_sales_price - coalesce(cr_refunded_cash,0) else 0  
end) as sales_before,  
    sum(case when (cast(d_date as date) >= cast ('1998-04-08' as date))  
    then cs_sales_price - coalesce(cr_refunded_cash,0) else 0  
end) as sales_after  
from catalog_sales left outer join catalog_returns on (cs_order_number =  
    cr_order_number and cs_item_sk = cr_item_sk),  
    warehouse, item, date_dim  
where i_current_price between 0.99 and 1.49 and i_item_sk = cs_item_sk and  
    cs_warehouse_sk = w_warehouse_sk and cs_sold_date_sk = d_date_sk and  
    d_date between (cast ('1998-04-08' as date) - 30) and (cast ('1998-04-08' as date) + 30)  
group by w_state,i_item_id  
order by w_state,i_item_id  
limit 100;

Q42  
select d_year, i_category_id, i_category, sum(ss_ext_sales_price)  
from date_dim, store_sales, item  
where d_date_sk = ss_sold_date_sk and ss_item_sk = i_item_sk and  
    i_manager_id = 1 and d_moy=12 and d_year=1998  
group by d_year, i_category_id, i_category  
order by sum(ss_ext_sales_price) desc, d_year, i_category_id, i_category  
limit 100;
Q43

```sql
select  s_store_name, s_store_id,
        sum(case when (d_day_name='Sunday') then ss_sales_price else null end) sun_sales,
        sum(case when (d_day_name='Monday') then ss_sales_price else null end) mon_sales,
        sum(case when (d_day_name='Tuesday') then ss_sales_price else null end) tue_sales,
        sum(case when (d_day_name='Wednesday') then ss_sales_price else null end) wed_sales,
        sum(case when (d_day_name='Thursday') then ss_sales_price else null end) thu_sales,
        sum(case when (d_day_name='Friday') then ss_sales_price else null end) fri_sales,
        sum(case when (d_day_name='Saturday') then ss_sales_price else null end) sat_sales
from date_dim, store_sales, store
where d_date_sk = ss_sold_date_sk and s_store_sk = ss_store_sk and
  s_gmt_offset = -5 and d_year = 1998
group by s_store_name, s_store_id
order by s_store_name, s_store_id, sun_sales, mon_sales, tue_sales, wed_sales, thu_sales, fri_sales, sat_sales
limit 100;
```
Q50

```
select s_store_name, s_company_id, s_street_number, s_street_name, s_street_type, s_suite_number, s_city, s_county, s_state, s_zip,
       sum(case when (sr_returned_date_sk - ss_sold_date_sk <= 30 ) then 1 else 0 end) as "30 days",
       sum(case when (sr_returned_date_sk - ss_sold_date_sk > 30) and (sr_returned_date_sk - ss_sold_date_sk <= 60) then 1 else 0 end ) as "31-60 days",
       sum(case when (sr_returned_date_sk - ss_sold_date_sk > 60) and (sr_returned_date_sk - ss_sold_date_sk <= 90) then 1 else 0 end) as "61-90 days",
       sum(case when (sr_returned_date_sk - ss_sold_date_sk > 90) and (sr_returned_date_sk - ss_sold_date_sk <= 120) then 1 else 0 end) as "91-120 days",
       sum(case when (sr_returned_date_sk - ss_sold_date_sk > 120) then 1 else 0 end) as ">120 days"
from store_sales, store_returns, store, date_dim d1, date_dim d2
where d2.d_year = 2000 and d2.d_moy  = 9 and ss_ticket_number = sr_ticket_number and ss_item_sk = sr_item_sk and ss_sold_date_sk = d1.d_date_sk and sr_returned_date_sk = d2.d_date_sk and ss_customer_sk = sr_customer_sk and ss_store_sk = s_store_sk
       group by s_store_name, s_company_id, s_street_number, s_street_name, s_street_type, s_suite_number, s_city, s_county, s_state, s_zip
order by s_store_name, s_company_id, s_street_number, s_street_name, s_street_type, s_suite_number, s_city, s_county, s_state, s_zip
limit 100;
```}

Q52

```
select d_year, i_brand_id brand_id ,i_brand brand, sum(ss_ext_sales_price) ext_price
from date_dim dt, store_sales, item
where d_date_sk = store_sales.ss_sold_date_sk and store_sales.ss_item_sk = i_item_sk and i_manager_id = 1 and d_moy=12 and d_year=1998
       group by d_year, i_brand, i_brand_id
order by d_year, ext_price desc, brand_id
limit 100;
```
Q55

```sql
select i_brand_id brand_id, i_brand brand, sum(ss_ext_sales_price)
  ext_price
from date_dim, store_sales, item
where d_date_sk = ss_sold_date_sk and ss_item_sk = i_item_sk and
  i_manager_id=36 and d_moy=12 and d_year=2001
group by i_brand, i_brand_id
order by ext_price desc, i_brand_id
limit 100;
```

Q62

```sql
select substr(w_warehouse_name,1,20), sm_type, web_name,
  sum(case when (ws_ship_date_sk - ws_sold_date_sk <= 30 ) then 1
        else 0 end) as "30 days",
  sum(case when (ws_ship_date_sk - ws_sold_date_sk > 30) and
          (ws_ship_date_sk - ws_sold_date_sk <= 60) then 1 else
        0 end) as "31-60 days",
  sum(case when (ws_ship_date_sk - ws_sold_date_sk > 60) and
          (ws_ship_date_sk - ws_sold_date_sk <= 90) then 1 else
        0 end) as "61-90 days",
  sum(case when (ws_ship_date_sk - ws_sold_date_sk > 90) and
          (ws_ship_date_sk - ws_sold_date_sk <= 120) then 1
        else 0 end) as "91-120 days",
  sum(case when (ws_ship_date_sk - ws_sold_date_sk  > 120) then 1
        else 0 end) as ">120 days"
from web_sales, warehouse, ship_mode, web_site, date_dim
where d_month_seq between 1212 and 1212 + 11 and ws_ship_date_sk =
  d_date_sk and ws_warehouse_sk   = w_warehouse_sk and
  ws_ship_mode_sk = sm_ship_mode_sk and ws_web_site_sk = web_site_sk
group by substr(w_warehouse_name,1,20), sm_type, web_name
order by substr(w_warehouse_name,1,20), sm_type, web_name
limit 100;
```
Q65

```sql
select s_store_name, i_item_desc, sc.revenue, i_current_price, i_wholesale_cost, i_brand
from store, item,
    (select ss_store_sk, avg(revenue) as ave
     from (select  ss_store_sk, ss_item_sk, sum(ss_sales_price) as revenue
          from store_sales, date_dim
          where ss_sold_date_sk = d_date_sk and d_month_seq between 1212 and 1212+11
          group by ss_store_sk, ss_item_sk) sa
     group by ss_store_sk) sb,
    (select  ss_store_sk, ss_item_sk, sum(ss_sales_price) as revenue
     from store_sales, date_dim
     where ss_sold_date_sk = d_date_sk and d_month_seq between 1212 and 1212+11
     group by ss_store_sk, ss_item_sk) sc
where sb.ss_store_sk = sc.ss_store_sk and sc.revenue <= 0.1 * sb.ave and s_store_sk = sc.ss_store_sk and i_item_sk = sc.ss_item_sk
order by s_store_name, i_item_desc
limit 100;
```

Q68

```sql
select c_last_name ,c_first_name, ca_city, bought_city, ss_ticket_number, extended_price, extended_tax, list_price
from (select ss_ticket_number, ss_customer_sk, ca_city bought_city, sum(ss_ext_sales_price) extended_price, sum(ss_ext_list_price) list_price, sum(ss_ext_tax) extended_tax
     from store_sales, date_dim, store, household_demographics, customer_address
     where store_sales.ss_sold_date_sk = date_dim.d_date_sk and store_sales.ss_store_sk = store.s_store_sk and store_sales.ss_hdemo_sk = household_demographics.hd_demo_sk and store_sales.ss_addr_sk = customer_address.ca_address_sk and date_dim.d_dom between 1 and 2 and (household_demographics.hd_dep_count = 5 or household_demographics.hd_vehicle_count= 3) and date_dim.d_year in (1999,1999+1,1999+2) and store.s_city in ('Midway','Fairview')
group by ss_ticket_number, ss_customer_sk, ss_addr_sk,ca_city) dn,
customer, customer_address current_addr
where ss_customer_sk = c_customer_sk and customer.c_current_addr_sk = current_addr.ca_address_sk and current_addr.ca_city <> bought_city
order by c_last_name, ss_ticket_number
limit 100;
```
select channel, col_name, d_year, d_qoy, i_category, count(*) sales_cnt,
SUM(ext_sales_price) sales_amt
from (select 'store' as channel, 'ss_addr_sk' col_name, d_year, d_qoy,
i_category, ss_ext_sales_price ext_sales_price
from store_sales, item, date_dim
where ss_addr_sk IS NULL and ss_sold_date_sk=d_date_sk and
ss_item_sk=i_item_sk
union all
select 'web' as channel, 'ws_web_page_sk' col_name, d_year, d_qoy,
i_category, ws_ext_sales_price ext_sales_price
from web_sales, item, date_dim
where ws_web_page_sk IS NULL and ws_sold_date_sk=d_date_sk and
ws_item_sk=i_item_sk
union all
select 'catalog' as channel, 'cs_warehouse_sk' col_name, d_year,
d_qoy, i_category, cs_ext_sales_price ext_sales_price
from catalog_sales, item, date_dim
where cs_warehouse_sk IS NULL and cs_sold_date_sk=d_date_sk and
cs_item_sk=i_item_sk) foo
group by channel, col_name, d_year, d_qoy, i_category
order by channel, col_name, d_year, d_qoy, i_category
limit 100;

select c_last_name, c_first_name, substr(s_city,1,30), ss_ticket_number,
amt, profit
from (select ss_ticket_number, ss_customer_sk, store.s_city, sum
(ss_coupon_amt) amt, sum(ss_net_profit) profit
from store_sales,date_dim,store,household_demographics
where store_sales.ss_sold_date_sk = date_dim.d_date_sk and
store_sales.ss_store_sk = store.s_store_sk and
store_sales.ss_hdemo_sk = household_demographics.hd_demo_sk and
(household_demographics.hd_dep_count = 8 or
household_demographics.hd_vehicle_count > 0) and
date_dim.d_dow = 1 and date_dim.d_year in (1998,1998+1,1998+2)
and store.s_number_employees between 200 and 295
group by ss_ticket_number,ss_customer_sk,ss_addr_sk,store.s_city) ms,
customer
where ss_customer_sk = c_customer_sk
order by c_last_name,c_first_name,substr(s_city,1,30), profit
limit 100;
Q84

```sql
select c_customer_id as customer_id, c_last_name || ', ' || c_first_name as customername
from customer, customer_address, customer_demographics,
household_demographics, income_band, store_returns
where ca_city = 'Hopewell' and c_current_addr_sk = ca_address_sk and
  ib_lower_bound >= 32287 and ib_upper_bound <= 32287 + 50000 and
  ib_income_band_sk = hd_income_band_sk and
  cd_demo_sk = c_current_cdemo_sk and hd_demo_sk = c_current_hdemo_sk
and sr_cdemo_sk = cd_demo_sk
order by c_customer_id
limit 100;
```

Q91

```sql
select cc_call_center_id Call_Center, cc_name Call_Center_Name, cc_manager Manager, sum(cr_net_loss) Returns_Loss
from call_center, catalog_returns, date_dim, customer, customer_address,
customer_demographics, household_demographics
where cr_call_center_sk = cc_call_center_sk and cr_returned_date_sk =
  d_date_sk and cr_returning_customer_sk = c_customer_sk and
  cd_demo_sk = c_current_cdemo_sk and hd_demo_sk = c_current_hdemo_sk and
  ca_address_sk = c_current_addr_sk and d_year = 1999 and d_moy = 11 and
  ((cd_marital_status = 'M' and cd_education_status = 'Unknown') or
   (cd_marital_status = 'W' and cd_education_status = 'Advanced Degree')) and
  hd_buy_potential like '0-500%' and ca_gmt_offset = -7
group by cc_call_center_id,cc_name,cc_manager,cd_marital_status,
cd_education_status
order by sum(cr_net_loss) desc;
```

Q93

```sql
select ss_customer_sk ,sum(act_sales) sumsales
from (select ss_item_sk, ss_ticket_number, ss_customer_sk,
case when sr_return_quantity is not null then (ss_quantity-
  sr_return_quantity)*ss_sales_price
   else
   (ss_quantity*ss_sales_price) end act_sales
  from store_sales left outer join store_returns on (sr_item_sk = ss_item_sk and sr_ticket_number = ss_ticket_number), reason
  where sr_reason_sk = r_reason_sk and r_reason_desc = 'Did not like
  the warranty') t
 group by ss_customer_sk
order by sumsales, ss_customer_sk
limit 100;
```
SELECT substr(w_warehouse_name,1,20), sm_type, cc_name,
    SUM(CASE WHEN (cs_ship_date_sk - cs_sold_date_sk) <= 30 THEN 1 ELSE 0 END) AS "30 days",
    SUM(CASE WHEN (cs_ship_date_sk - cs_sold_date_sk) > 30 AND (cs_ship_date_sk - cs_sold_date_sk) <= 60 THEN 1 ELSE 0 END) AS "31-60 days",
    SUM(CASE WHEN (cs_ship_date_sk - cs_sold_date_sk) > 60 AND (cs_ship_date_sk - cs_sold_date_sk) <= 90 THEN 1 ELSE 0 END) AS "61-90 days",
    SUM(CASE WHEN (cs_ship_date_sk - cs_sold_date_sk) > 90 AND (cs_ship_date_sk - cs_sold_date_sk) <= 120 THEN 1 ELSE 0 END) AS "91-120 days",
    SUM(CASE WHEN (cs_ship_date_sk - cs_sold_date_sk) > 120 THEN 1 ELSE 0 END) AS ">120 days"
FROM catalog_sales, warehouse, ship_mode, call_center, date_dim
WHERE d_month_seq BETWEEN 1212 AND 1212 + 11 AND cs_ship_date_sk = d_date_sk AND cs_warehouse_sk = w_warehouse_sk AND cs_ship_mode_sk = sm_ship_mode_sk AND cs_call_center_sk = cc_call_center_sk
GROUP BY substr(w_warehouse_name,1,20), sm_type, cc_name
ORDER BY substr(w_warehouse_name,1,20), sm_type, cc_name
LIMIT 100;

Reference

- [https://github.com/cloudera/impala-tpcds-kit](https://github.com/cloudera/impala-tpcds-kit)
- [https://github.com/hortonworks/hive-testbench](https://github.com/hortonworks/hive-testbench)