CWI

Database Architectures

Verifying TPC-DS Results

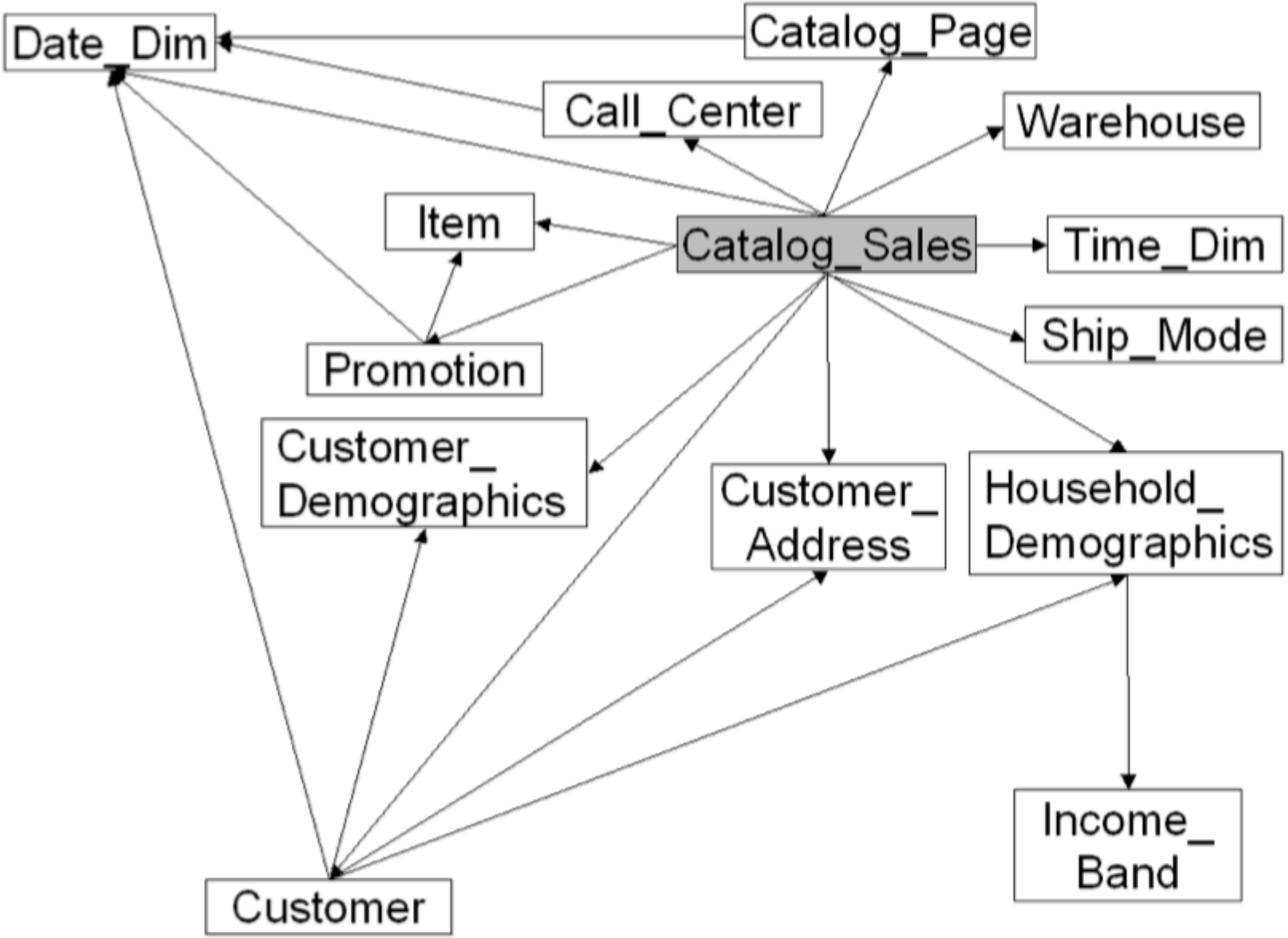
Hannes Mühleisen, 2018-10-29

TPC-DS?

- “Decision Support”
- VLDB ’06, Othayoth & Pöss: The Making of TPC-DS
    - “TPC-H is not representative”
- *One* official result, 2018
    - “Transwarp Data Hub v5.1” ?!
    - 12 years later? Why?

TPC-DS Design

- Multiple snowflake schemas with *shared dimensions*
  - 24 tables with an average of 18 columns
  - More representative skewed database content + NULLs
  - Sub-linear scaling of non-fact tables
- Ad-hoc, reporting, iterative and extraction queries
  - ~~99~~ **103** distinct SQL 99 queries with random substitutions (ROLLUP & PARTITION)
  - Substitutes also aggr funcs *and column names*
- ETL-like data maintenance (ignored)


Date_Dim
Catalog_Page
Call_Center
Warehouse
Item
Catalog_Sales
Time_Dim
Promotion
Ship_Mode
Customer_
Demographics
Customer_
Address
Household_
Demographics
Income_
Band
Customer

TPC-DS Q1

```
with customer_total_return as
(select sr_customer_sk as ctr_customer_sk
,sr_store_sk as ctr_store_sk
,sum([AGG_FIELD]) as ctr_total_return
from store_returns
,date_dim
where sr_returned_date_sk = d_date_sk
and d_year = [YEAR]
group by sr_customer_sk
,sr_store_sk)
select c_customer_id
from customer_total_return ctr1
,store
,customer
where ctr1.ctr_total_return > (select avg(ctr_total_return)*1.2
from customer_total_return ctr2
where ctr1.ctr_store_sk = ctr2.ctr_store_sk)
and s_store_sk = ctr1.ctr_store_sk
and s_state = '[STATE]'
and ctr1.ctr_customer_sk = c_customer_sk
order by c_customer_id
LIMIT 100;
```

TPC-DS Q4

```
with year_total as (
 select c_customer_id customer_id
       ,c_first_name customer_first_name
       ,c_last_name customer_last_name
       ,c_preferred_cust_flag customer_preferred_cust_flag
       ,c_birth_country customer_birth_country
       ,c_login customer_login
       ,c_email_address customer_email_address
       ,d_year dyear
       ,sum(((ss_ext_list_price-ss_ext_wholesale_cost-ss_ext_discount_amt)+ss_ext_sales_price)/2) year_total
       ,'s' sale_type
 from customer
     ,store_sales
     ,date_dim
 where c_customer_sk = ss_customer_sk
   and ss_sold_date_sk = d_date_sk
 group by c_customer_id
         ,c_first_name
         ,c_last_name
         ,c_preferred_cust_flag
         ,c_birth_country
         ,c_login
         ,c_email_address
         ,d_year
 union all
 select c_customer_id customer_id
       ,c_first_name customer_first_name
       ,c_last_name customer_last_name
       ,c_preferred_cust_flag customer_preferred_cust_flag
       ,c_birth_country customer_birth_country
       ,c_login customer_login
       ,c_email_address customer_email_address
       ,d_year dyear
       ,sum((((cs_ext_list_price-cs_ext_wholesale_cost-cs_ext_discount_amt)+cs_ext_sales_price)/2) ) year_total
       ,'c' sale_type
 from customer
     ,catalog_sales
     ,date_dim
 where c_customer_sk = cs_bill_customer_sk
   and cs_sold_date_sk = d_date_sk
 group by c_customer_id
         ,c_first_name
         ,c_last_name
         ,c_preferred_cust_flag
         ,c_birth_country
         ,c_login
         ,c_email_address
         ,d_year
union all
 select c_customer_id customer_id
       ,c_first_name customer_first_name
       ,c_last_name customer_last_name
       ,c_preferred_cust_flag customer_preferred_cust_flag
       ,c_birth_country customer_birth_country
       ,c_login customer_login
       ,c_email_address customer_email_address
       ,d_year dyear
       ,sum((((ws_ext_list_price-ws_ext_wholesale_cost-ws_ext_discount_amt)+ws_ext_sales_price)/2) ) year_total
       ,'w' sale_type
 from customer
     ,web_sales
     ,date_dim
 where c_customer_sk = ws_bill_customer_sk
   and ws_sold_date_sk = d_date_sk
 group by c_customer_id
         ,c_first_name
         ,c_last_name
         ,c_preferred_cust_flag
         ,c_birth_country
         ,c_login
         ,c_email_address
         ,d_year
         )
 select
                 t_s_secyear.customer_id
                ,t_s_secyear.customer_first_name
                ,t_s_secyear.customer_last_name
                ,[SELECTONE]
 from year_total t_s_firstyear
     ,year_total t_s_secyear
     ,year_total t_c_firstyear
     ,year_total t_c_secyear
     ,year_total t_w_firstyear
     ,year_total t_w_secyear
 where t_s_secyear.customer_id = t_s_firstyear.customer_id
   and t_s_firstyear.customer_id = t_c_secyear.customer_id
   and t_s_firstyear.customer_id = t_c_firstyear.customer_id
   and t_s_firstyear.customer_id = t_w_firstyear.customer_id
   and t_s_firstyear.customer_id = t_w_secyear.customer_id
   and t_s_firstyear.sale_type = 's'
   and t_c_firstyear.sale_type = 'c'
   and t_w_firstyear.sale_type = 'w'
   and t_s_secyear.sale_type = 's'
   and t_c_secyear.sale_type = 'c'
   and t_w_secyear.sale_type = 'w'
   and t_s_firstyear.dyear =  [YEAR]
   and t_s_secyear.dyear = [YEAR]+1
   and t_c_firstyear.dyear =  [YEAR]
   and t_c_secyear.dyear =  [YEAR]+1
   and t_w_firstyear.dyear = [YEAR]
   and t_w_secyear.dyear = [YEAR]+1
   and t_s_firstyear.year_total > 0
   and t_c_firstyear.year_total > 0
   and t_w_firstyear.year_total > 0
   and case when t_c_firstyear.year_total > 0 then t_c_secyear.year_total / t_c_firstyear.year_total else null end
            > case when t_s_firstyear.year_total > 0 then t_s_secyear.year_total / t_s_firstyear.year_total else null end
   and case when t_c_firstyear.year_total > 0 then t_c_secyear.year_total / t_c_firstyear.year_total else null end
            > case when t_w_firstyear.year_total > 0 then t_w_secyear.year_total / t_w_firstyear.year_total else null end
 order by t_s_secyear.customer_id
         ,t_s_secyear.customer_first_name
         ,t_s_secyear.customer_last_name
         ,[SELECTONE]
limit 100;
```

“Qualification”

- Reference results “provided” for SF 1
  - *Completely borked* format, every single one had different encoding (grrr)
- NULL first/last difference, hence 2 sets of results for many queries
- Query placeholder values only in spec PDF (grrr)

B.1 query1.tpl

Find customers who have returned items more than 20% more often than the average customer returns for a store in a given state for a given year.

Qualification Substitution Parameters:

- YEAR.01=2000
- STATE.01=TN
- AGG_FIELD.01 = SR_RETURN_AMT

[Q 34 NULLs last]

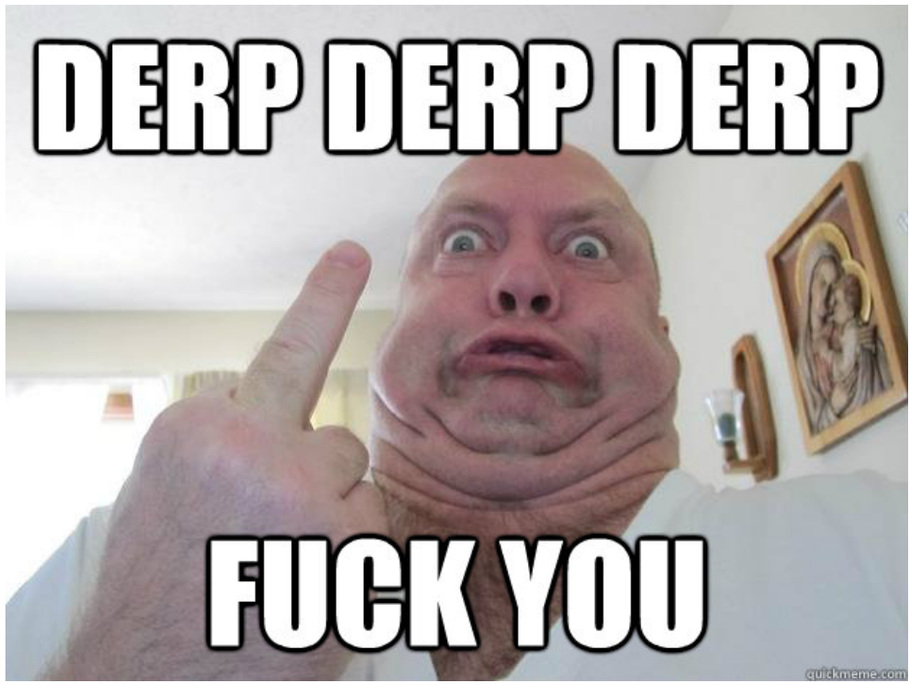
```
SQL*Plus: Release 12.1.0.2.0 Production on Wed Jun 8 16:25:41 2016

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> Connected.
SQL> SQL> SQL> SQL>
C_LAST_NAME          C_FIRST_NAME     C_SALUTATI C SS_TICKET_NUMBER  CNT
-------------------- ---------------- ---------- - --------------- ----------
Abbott               Harriet          Ms.        N          195008   15
Acosta               Victor           Mr.        N          156807   15
Adams                Jerry            Sir        N          216653   15
…
Woody                Philip           Mr.        N          211173   16
Wright               Amanda                      Y          105395   15
Yates                Kimberly         Ms.        Y          220849   15
                     Grace                       147095   16
                     Joshua           Sir        N          108357   15
                     Judy             Dr.                   234350   16
                     Roy                         200033   15
                        Mr.                      Y          214003   15
                        Ms.                       54536   16
                             148695   16
                             151297   15
                             Y         16278   15
                             Y        146557   16
                             N         55640   15
                             N         58614   15
                             N         96514   15
                             N        220211   15

451 rows selected.

Elapsed: 00:00:03.36
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
```



Challenge: Reproduce Results

- What are the correct answers to the qualification queries on SF1?
- Very useful for testing!
  - check correctness before bragging about performance

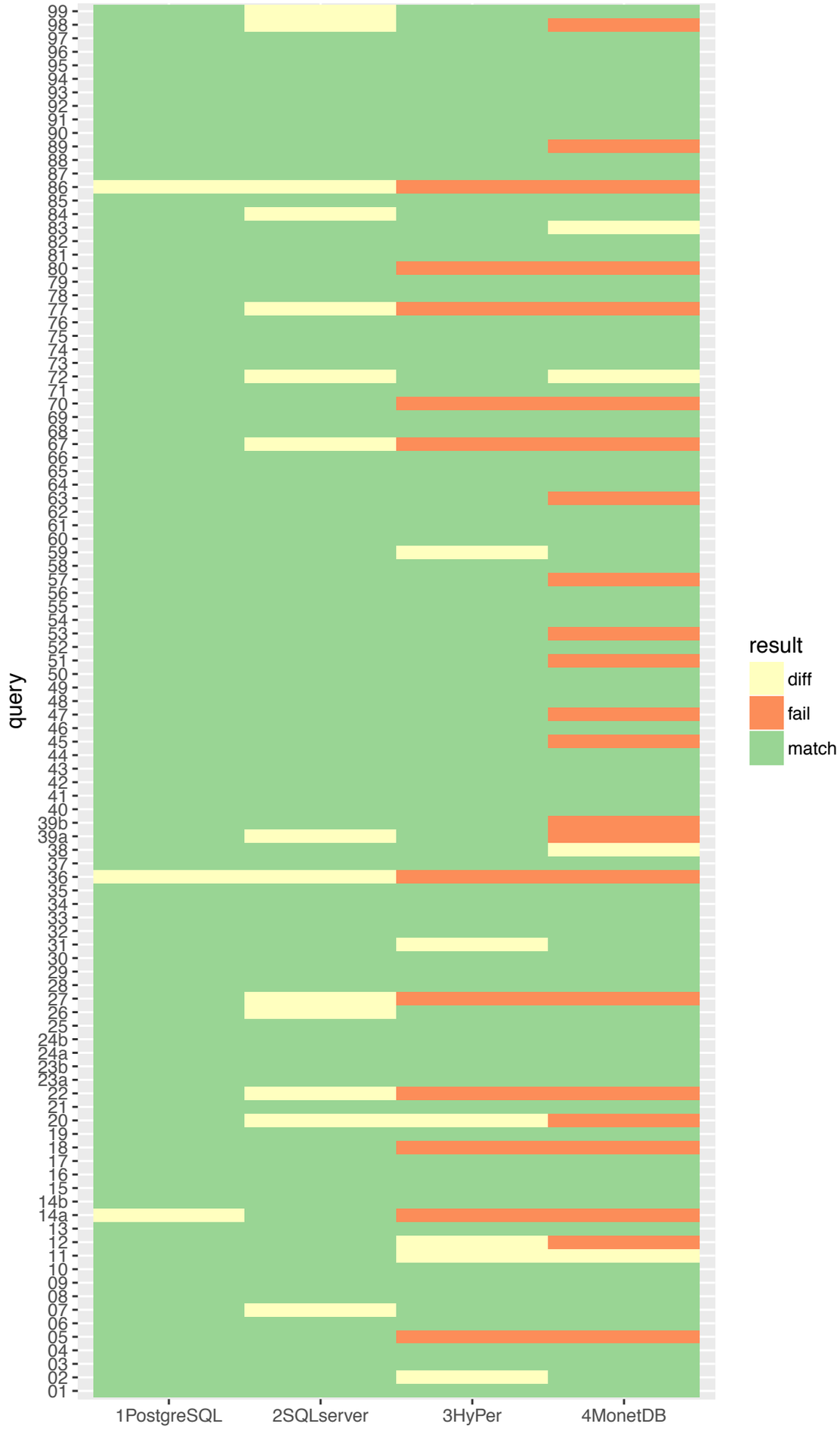
- Steps:
  - Clean up reference results (manually)
  - Create tool to check reference results (numeric diffs)
  - Create qualification queries from PDF (manually)
  - Run queries on bunch of systems to see whether they agree

Systems tested

- Postgres 11.0
- HyPer 20182.18.1009.2120
- SQL Server on Docker
- MonetDB 11.31.11
- SQLite 3.24.0
  - (dropped because could not run enough queries)

Results

- Reference results for Queries 21, 34, 66, 71, 73, 79 and 98 were wrong (missing/additional rows, values, spacing)
  - Determined by consensus of 2 or more systems
- Queries 78 and 83 are probably broken (int div/rounding)
- Postgres/SQL server can run all queries
- HyPer misses ROLLUP, thus failing ~ 10 queries
- MonetDB misses ROLLUP and PARTITION, thus failing ~ 20 queries
  - Analytics branch can’t do PARTITION queries yet (checked 2018-10-29)
- All but three queries have reproduced results
  - Need additional systems to double-check reference results


query
99
98
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39b
39a
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24b
24a
23b
23a
22
21
20
19
18
17
16
15
14b
14a
13
12
11
10
09
08
07
06
05
04
03
02
01
1PostgreSQL
2SQLserver
3HyPer
4MonetDB
result
diff
fail
match

https://github.com/cwida/tpcds-kit

- Cleaned and checked reference (results NULL first+last)
- Qualification queries
- Data generator
  - Copy of SF1 at https://homepages.cwi.nl/~hannes/tpcds-sf1-xz/
- Load scripts for various DBs
- Result diff tool (1% numeric drift and spaces allowed)
- Plotting script (R)
- Some more cleanup and documentation to come

Next steps

- Cleanup repo and document
- Verify last three inconsistent results
  - Which system? Oracle? Vectorwise? Redshift
- Write paper for TPCTC?
- Actually run a performance comparison?
- Create test cases for systems / file bug reports
- Questions?