Converting TPC-H Query Templates to use DSQgen for Easy Extensibility

John M. Stephens, Gradientsystems
Meikel Poess, Oracle Corporation

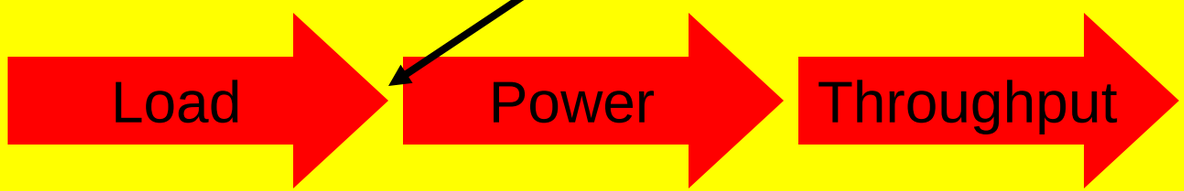
TPC-H

- TPC-H has been a very successful benchmark for the TPC
  - 147+ publications[1]
  - 10+ hardware systems[1]
  - 7+ database systems[1]
- TPC-H's tools (dbgen/qgen) are 15 years old
- In order to add queries or modify queries code changes to qgen are necessary

1 as of June 2009

How Ad-Hoc Queries are Implemented

- TPC-H defines query templates instead of queries
- Qgen substitutes scalar variables randomly during benchmark runtime with random seed
- Seed is determined at the end of the load time (second granularity)



Query Template Example: TPC-H's Query 6

```
SELECT  SUM (l_extendedprice*l_discount)
FROM    lineitem
WHERE   l_shipdate>=date'[DATE]'
   AND  l_shipdate<date'[DATE]'+interval'1'year
   AND  l_discount between [DISCOUNT] - 0.01
                   and [DISCOUNT] + 0.01
   AND  l_quantity < [QUANTITY];
```

Query Template Example: TPC-H's Query 6

```
SELECT  SUM (l_extendedprice*l_discount)
FROM    lineitem
WHERE   l_shipdate>=date'10-1-1996'
   AND  l_shipdate<date'10-1-1996'+interval'1'year
   AND  l_discount between 0.05 - 0.01
                   and 0.05 + 0.01
   AND  l_quantity < 300;
```

Current Query Generator Qgen

- Data relationships are hard-coded in qgen
- Substitution parameters are hard-coded in qgen
  → query modifications require code changes
  → additional queries require code changs
  → testing, bug fixing etc.

DSQgen

- Originally developed for TPC-DS
- Query templates are defined in an extendable query language
- The definitions of substitution tags are included in query template
- Previous publications
  - Meikel Poess, John M. Stephens: Generating Thousand Benchmark Queries in Seconds. VLDB 2004: 1045-1053
  - Meikel Poess: Controlled SQL query evolution for decision support benchmarks. WOSP 2007: 38-41

DSQgen's Template Language

- A template consists of two parts:
  - substitution tag definitions
  - SQL Text
- Substitution tag definition can be:
  - random number between an lower and upper bound
  - list of items
  - unique list of items

Substitution Types

- Random Number Substitution
  - order_quantity = random (1, 10, uniform);
- Random String Substitution
  - color=TEXT({"brown",6},{"black",3},{"grey",1} ,{"pink",1});
- List Operators LIST,ULIST
  - colors=LIST(TEXT({"brown",6},{"black",3} ,{"grey",1} ,{"pink",1}),2);
  - colors=ULIST(TEXT({"brown",6},{"black",3} ,{"grey",1} ,{"pink",1}),2);

DSQgen's Template Language

- Built-In Functions
  - _SCALE
  - _SEED
  - _QUERY
  - _TEMPLATE
  - _STREAM
  - _LIMITA,_LIMITB,_LIMITC
  - _LIMIT

TPC-H Queries can be Divided into 5 Major Types

- **Type 1**: randomly selects one or more numbers from a dense interval.
- **Type 2**: randomly selects one or more strings from a list of possible items.
- **Type 3**: randomly selects a date.
- **Type 4**: selects the scale factor of the database being queried
- **Type 5**: selects the number of rows to be returned by the top most SQL statement.

Example: Query 16

- SELECT p_brand ,p_type ,p_size ,cou
  ps_suppkey) as supplier_cnt
  FROM partsupp, part
  WHERE p_partkey = ps_partke
    AND p_brand <> ':1'
    AND p_type not like ':2%'
    AND p_size in (:3, :4, :5, :6, :7, :8, :9, :10)
    AND ps_suppkey not in (SEL
                FROM supp
                WHERE s_
                    '%Custo
  _brand ,p_type, p_size
  upplier_cnt desc, p_brand, p_type,
  p_size;

(p_brand) is substituted as Brand#MN, where M and N are two single character strings representing two numbers randomly and independently selected within [1 .. 5];

(p_type) is made of three syllables:

1) STANDARD,SMALL,MEDIUM,LARGE,ECONOMY,PROMOT
2) ANODIZED,BURNISHED,PLATED,POLISHED,BRUSHED
3) TIN,NICKEL,BRASS,STEEL,COPPER

(p_size) are eight randomly selected as a set of different values of [1...50];

Query 16 in DSQGEN Syntax

```
DEFINE PBRAND_A = RANDOM(1,5,uniform);
DEFINE PBRAND_B = RANDOM(1,5,uniform);
DEFINE PTYPE_A=TEXT({"STANDARD",1},{“SMALL",1},{“MEDIUM",1},{“LARGE",1}
                    ,{“ECONOMY",1},{“PROMO",1});
DEFINE PTYPE_B=TEXT({“ANODIZED",1},{“BURNISHED",1},{“PLATED",1}
                    ,{“POLISHED",1},{“BRUSHED”,1});
DEFINE PTYPE_C=TEXT({“TIN",1},{“NICKEL",1},{“BRASS",1},{“STEEL",1},{“COPPER",1});
DEFINE SIZE = ULIST(RANDOM(1,50,uniform),8);

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#[PBRAND_A][PBRAND_B]'
  AND p_type not like '[PTYPE_A] [PTYPE_B] [PTYPE_C]%'
  AND p_size in ([SIZE.1],[SIZE.2],[SIZE.3],[SIZE.4]
                ,[SIZE.5],[SIZE.6],[SIZE.7],[SIZE.8])
  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;
```

Modified Query 11 of TPC-H

```
DEFINE NK = random (0,31, uniform);
DEFINE AGG= text({“sum”,1},{“min”,1}
                 ,{“max”,1});

SELECT ps_partkey
       ,[AGG](ps_supplycost * ps_availqty)
as value
FROM partsupp,supplier
WHERE ps_suppkey = s_suppkey
  AND s_nationkey = [NK]
GROUP BY ps_partkey;
```

Modified Query 3

```
DEFINE SHIPDATE = random(1,31,uniform);
DEFINE LIMIT=10;
DEFINE COL=text({“l_quantity”,1},{“l_discount”,1}
                ,{“l_extendedprice”,1},{“l_tax”,1});

[_LIMITA] select [_LIMITB] l_orderkey
          ,sum([COL]), o_orderdate, o_shippriority
FROM customer, orders, lineitem
WHERE c_custkey = o_custkey
  AND l_orderkey = o_orderkey
  AND o_orderdate < date '1995-03-[SHIPDAY]'
  AND l_shipdate > date '1995-03-[SHIPDAY]'
GROUP BY l_orderkey, o_orderdate, o_shippriority
ORDER BY [COL] desc, o_orderdate
[_LIMITC];
```

Summary

- We demonstrated that
  - all existing 22 TPC-H queries can be converted to use DSQgen
  - Conversion has no impact on the viability or comparability of existing TPC-H results
  - TPC-H queries can be enriched without code changes
  - New queries can be easily added without any code changes