

Towards a Unified Query Plan Representation

Jinsheng Ba^{1,2}, Manuel Rigger²

¹ETH Zurich, ²National University of Singapore



Query Plan

Query Plan

t0

c0	c1
2	null



EXPLAIN

```
SELECT *  
FROM t0  
WHERE c0 < 5;
```

QUERY PLAN

Seq Scan on t0 (cost=0.00..38.25 rows=753 width=8)
Filter: (c0 < 5)

PostgreSQL query plan

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t0	NULL	ALL	NULL	NULL	NULL	NULL	1	100.00	Using where

MySQL query plan

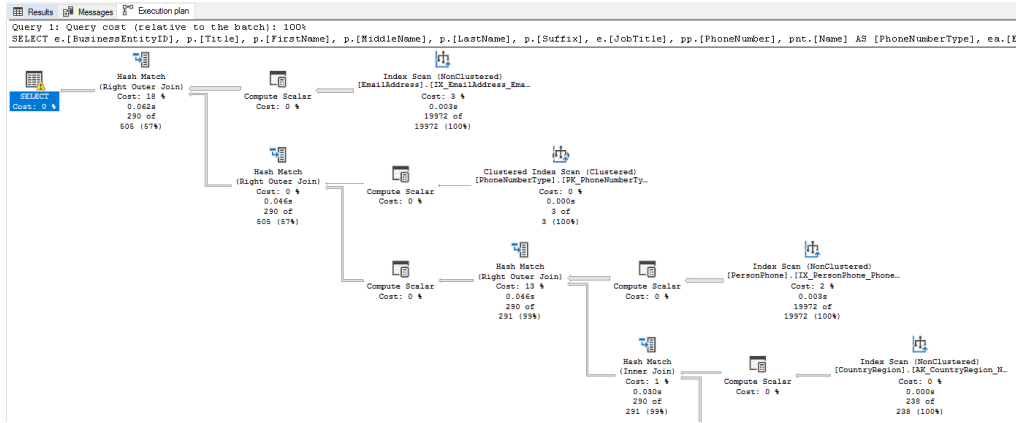
id	estRows	task	access object	operator info
TableReader_7	3323.33	root		data:Selection_6
└─Selection_6	3323.33	cop[tikv]		lt(testdb.t0.c0, 5)
└─┬─TableFullScan_5	10000.00	cop[tikv]	table:t0	keep order:false, stats:pseudo

TiDB query plan

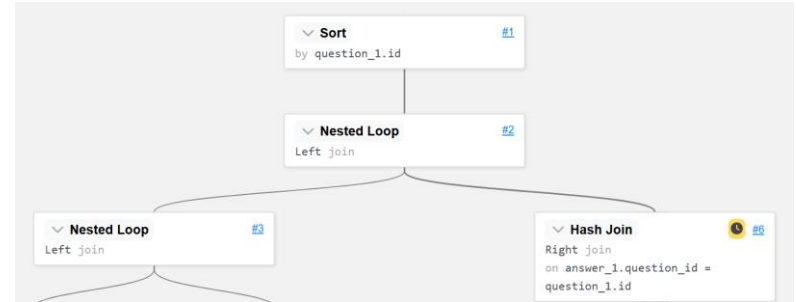
c0	c1
2	

Each DBMS has its own query plan representation!

Query Plan Applications: Visualization



SQLServer



PostgreSQL

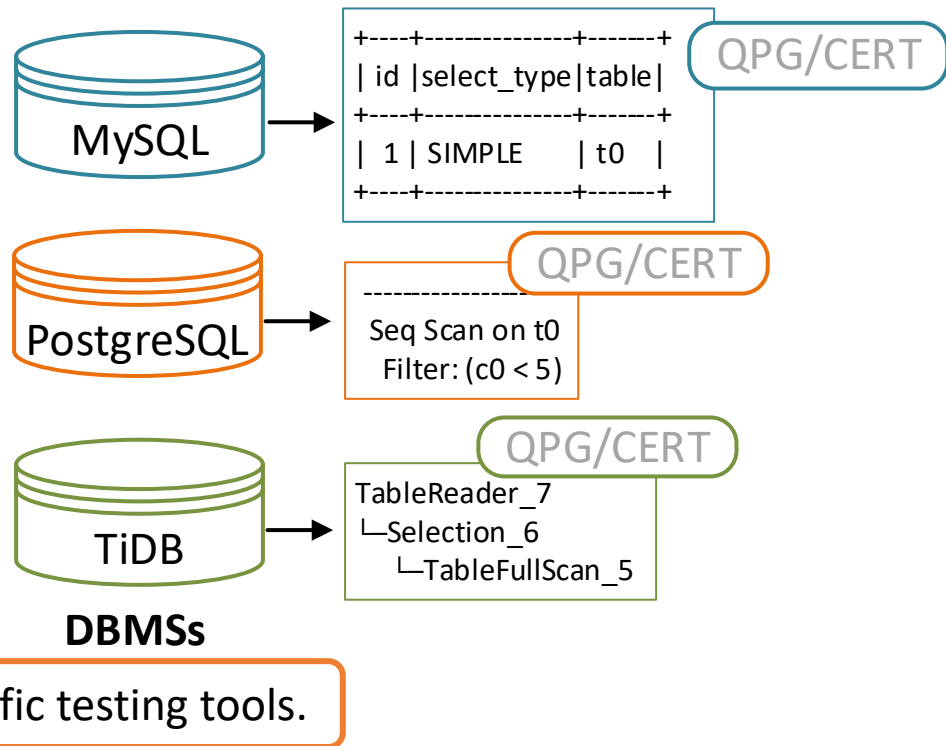
DBMS-specific query plan visualization tools.

Query Plan Applications: Testing

Two testing methods:

QPG [1]: Decide how to generate next input base on query plans.

CERT [2]: Decide whether there is a cardinality issue by examining query plans.



[1] Jinsheng Ba & Manuel Rigger. (2023). Testing Database Engines via Query Plan Guidance. In Proceedings of International Conference on Software Engineering (ICSE).

[2] Jinsheng Ba & Manuel Rigger. (2024). Finding Performance Issues in Database Engines via Cardinality Estimation Testing. In Proceedings of International Conference on Software Engineering (ICSE).

Problem

Building applications on query plans is challenging!

- Non-trivial effort
 - More than 1000 DBMSs in the world [1]
- Query plans are not well documented [2]



14.1. Using EXPLAIN

14.1.1. EXPLAIN Basics

14.1.2. EXPLAIN ANALYZE

14.1.3. Caveats

PostgreSQL devises a *query plan* for each query it receives. Choosing the right plan to match the query structure and the properties of the data is absolutely critical for good performance, so the system includes a complex *planner* that tries to choose good plans. You can use the `EXPLAIN` command to see what query plan the planner creates for any query. Plan-reading is an art that requires some experience to master, but this section attempts to cover the basics.

[1] <https://dbdb.io/>

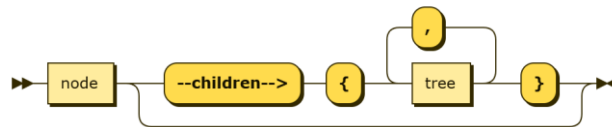
[2] <https://www.postgresql.org/docs/current/using-explain.html>

Contributions

1. A study of query plan representations

9 real-world DBMSs

2. A proposal for a unified query plan representation



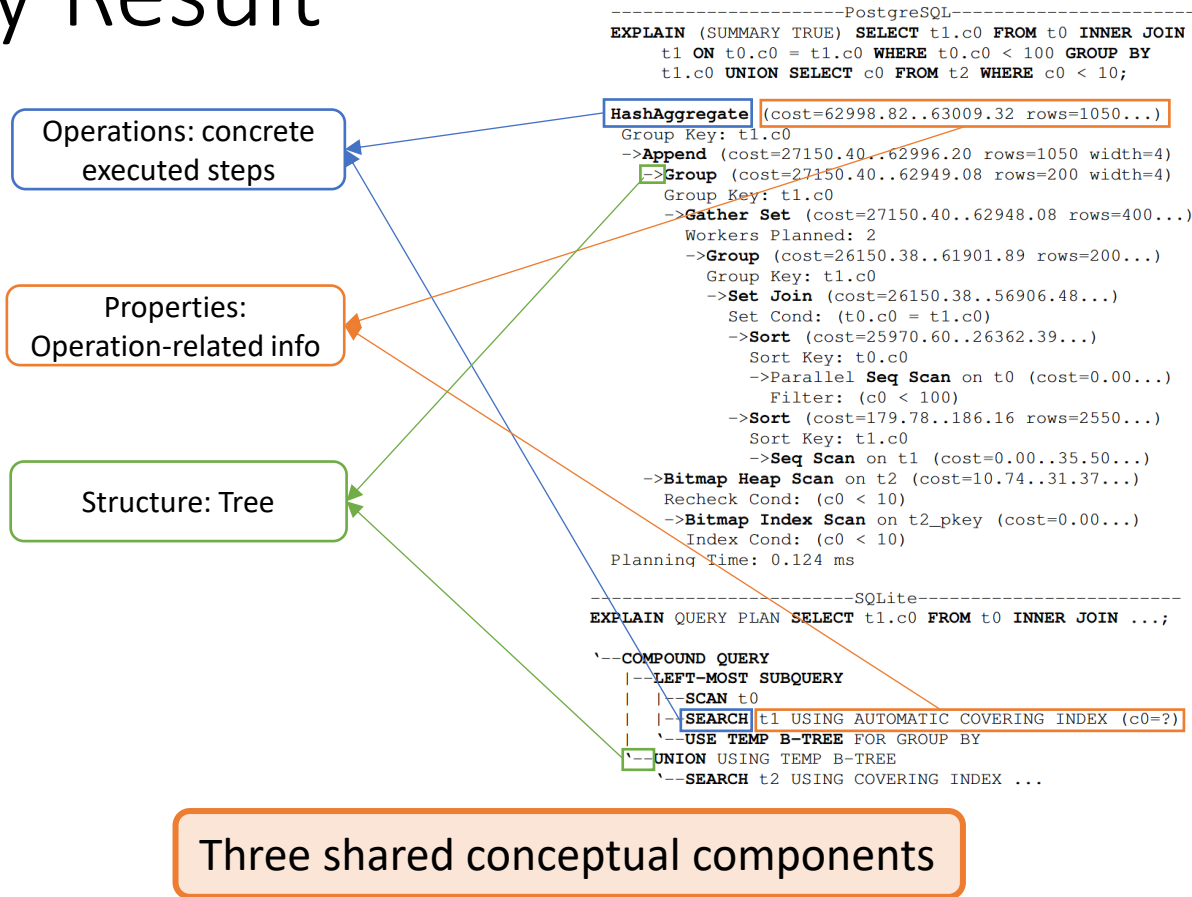
3. Applications

Testing, Visualization, Benchmarking

| Study Targets

DBMS	Version	Data Model	Release	Rank
InfluxDB	2.7.0	Time-series	2013	28
MongoDB	6.0.5	Document	2009	5
MySQL	8.0.32	Relational	1995	2
Neo4j	5.6.0	Graph	2007	22
PostgreSQL	14.7	Relational	1989	4
SQL Server	16.0.4015.1	Relational	1989	3
SQLite	3.41.2	Relational	1990	10
SparkSQL	3.3.2	Relational	2014	37
TiDB	6.5.1	Relational	2016	84

Study Result



Study Result

Hash join,
merge join.....

THE NUMBER OF OPERATIONS AND PROPERTIES IN QUERY PLAN REPRESENTATIONS. RA: RELATIONAL ALGEBRA.

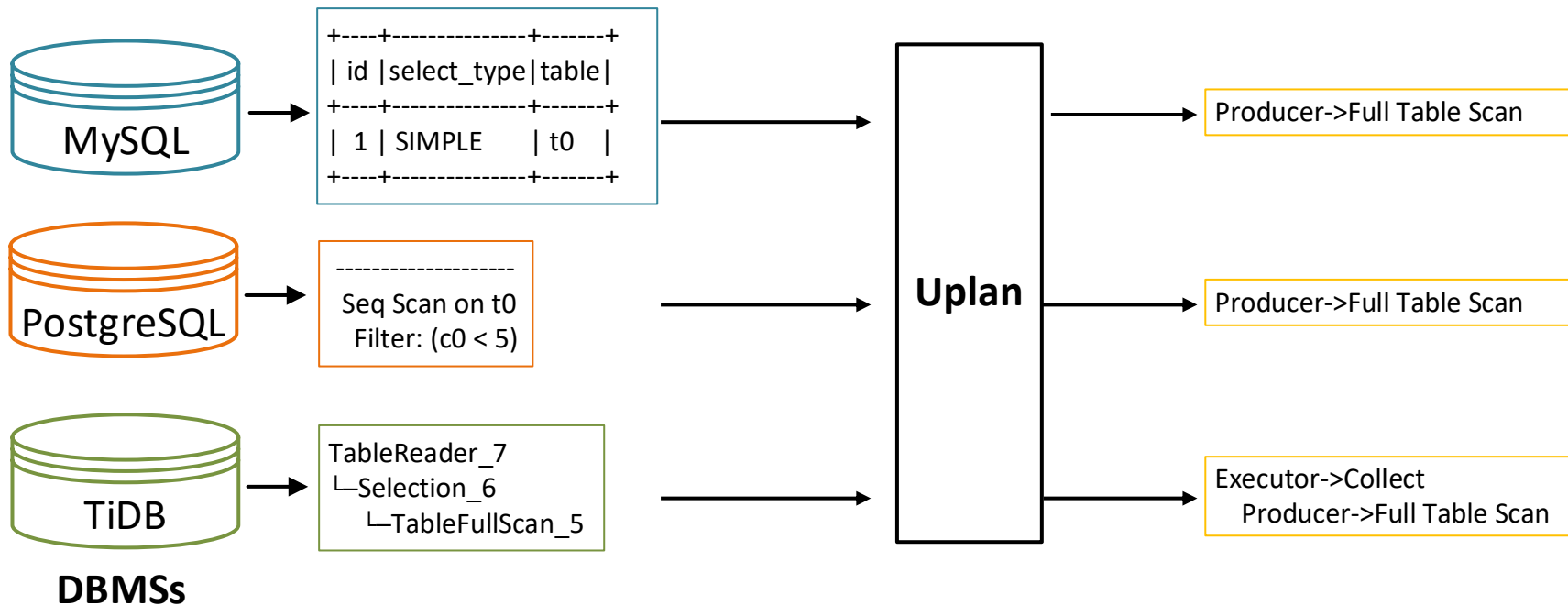
DBMS RA Operators	Operations								Properties				
	Producer σ	Combinator $\cup, \cap, -$	Join $\bowtie, \times$	Folder γ	Projector Π	Executor	Consumer	Sum	Cardinality	Cost	Configuration	Status	Sum
InfluxDB	0	0	0	0	0	0	0	0	5	0	0	1	6
MongoDB	14	9	0	5	3	10	3	44	16	5	18	12	51
MySQL	15	3	2	1	0	2	0	23	3	6	3	10	22
Neo4j	18	11	43	6	3	17	13	111	3	3	12	7	25
PostgreSQL	18	8	3	3	0	9	1	42	8	17	42	40	107
SQL Server	15	3	3	3	0	16	19	59	4	4	7	3	18
SQLite	3	6	3	0	0	5	0	17	0	0	3	0	3
SparkSQL	7	1	2	6	0	43	18	77	11	11	0	0	22
TiDB	19	6	7	5	1	13	5	56	2	5	4	1	12
Avg:	12	5	7	3	1	13	7	48	6	6	10	8	30

Full scan, index
scan.....

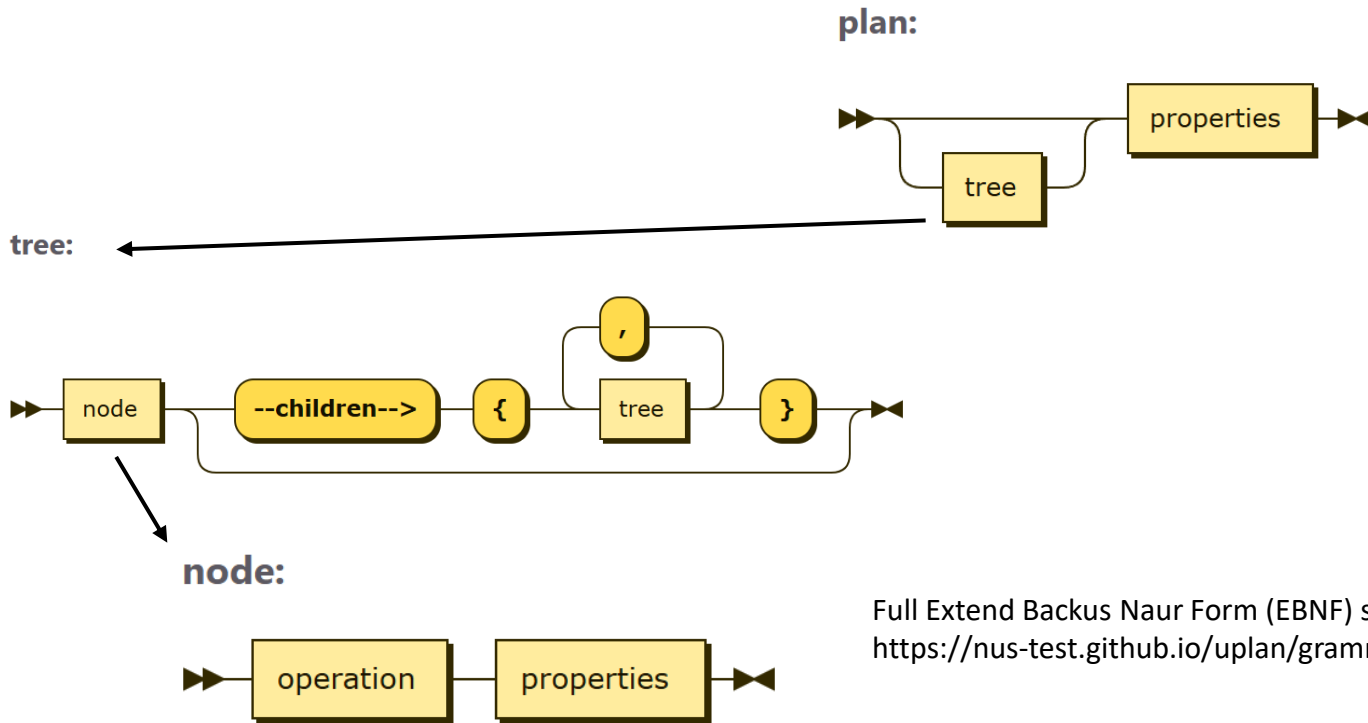
Memory,
time.....

The commonalities across DBMS-specific query plans make a unified representation possible.

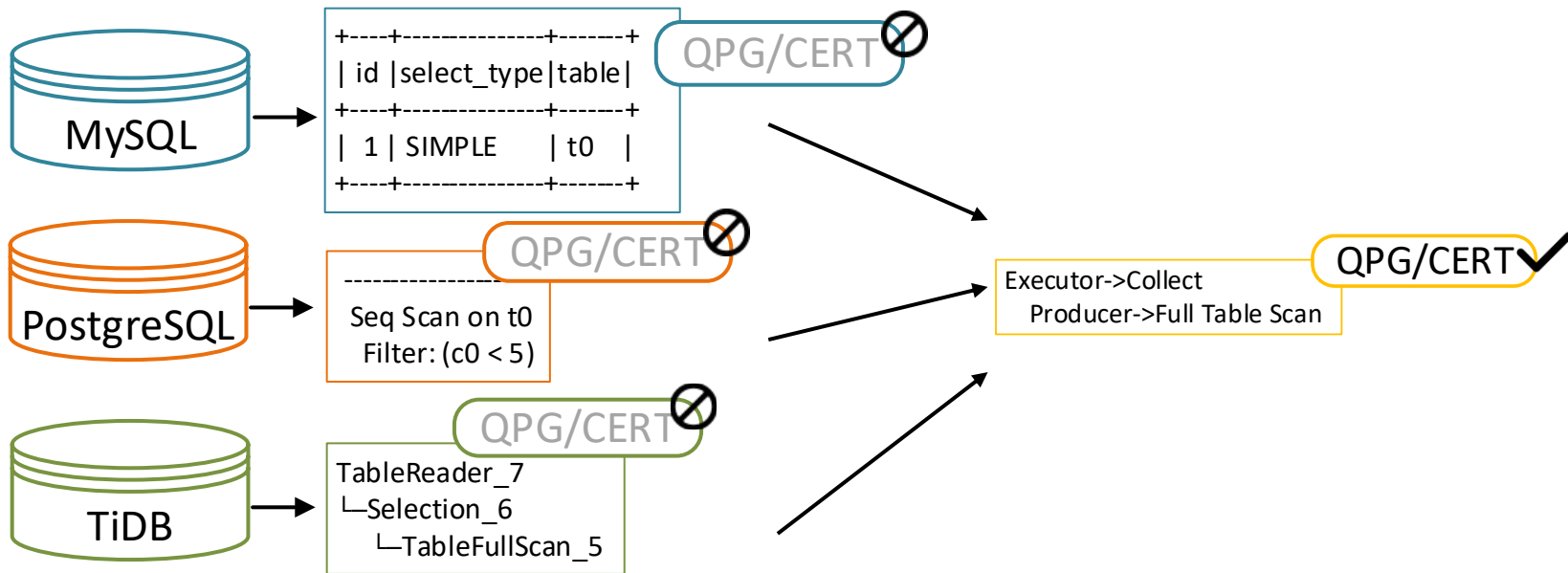
Unified Query Plan Representation (Uplan)



Unified Query Plan Representation (Uplan)



Application: Testing



DBMSs

UPlan enables large-scale adoption for testing methods QPG and CERT in a DBMS-agnostic implementation way.

Application: Benchmarking

PostgreSQL: six table scanning

TiDB: four table scanning

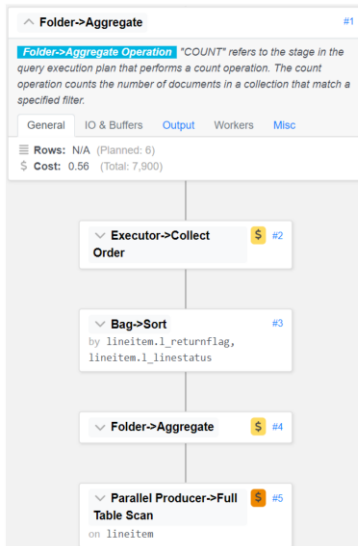
Comparing the unified query plan representation provides actionable insights.

```
SELECT ... FROM PARTSUPP, SUPPLIER, NATION WHERE ...  
HAVING ... > (SELECT ... FROM PARTSUPP, SUPPLIER,  
              NATION WHERE ...) ...;
```

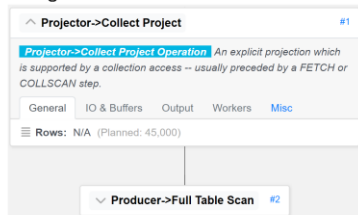
```
-----  
PostgreSQL:                                TiDB:  
Bag->Sort                                  Projector->Project  
Folder->Aggregate                          Bag->Sort  
Join->Hash Join                            Folder->Aggregate Hash  
  Producer->Full Table                    Projector->Project  
  name object: partsupp                  Join->Index Hash  
  Executor->Hash Row                     Join->Index Hash  
  Join->Hash                              Executor->Collect  
    Producer->Full Table                  Producer->Full Table  
    name object: supplier                 name object: nation  
    Executor->Hash Row                   Executor->Collect Order  
    Producer->Full Table                  Producer->Index-only...  
    name object: nation                   name object: supplier  
Folder->Aggregate                          Executor->Collect Order  
Join->Hash Join                            Producer->Index-only...  
  Producer->Full Table                    name object: partsupp  
  name object: partsupp                  Producer->Id Scan  
  Executor->Hash Row                      name object: partsupp  
  Join->Hash Join  
    Producer->Full Table  
    name object: supplier  
    Executor->Hash Row  
    Producer->Full Table  
    name object: nation
```

Application: Visualization

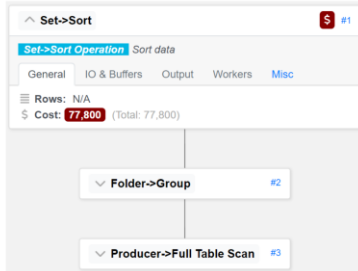
PostgreSQL



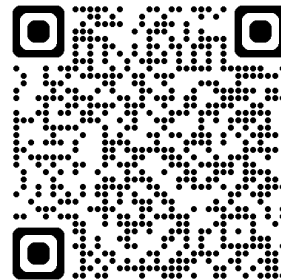
MongoDB



MySQL



Demo



<https://nus-test.github.io/uplan/visualization.html>

Existing DBMS-specific visualization tools could support more DBMSs if they supported our unified query plan representation.

Conclusion

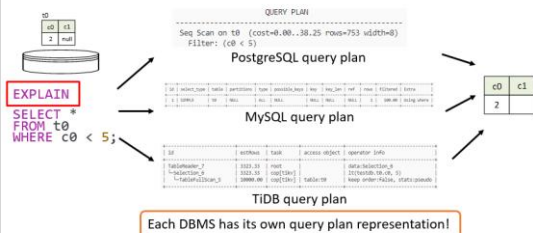


Website

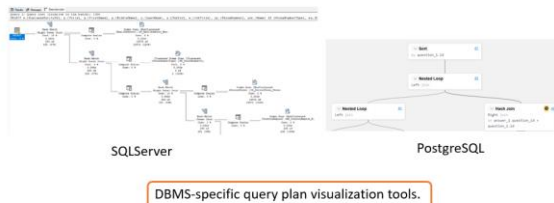


Author

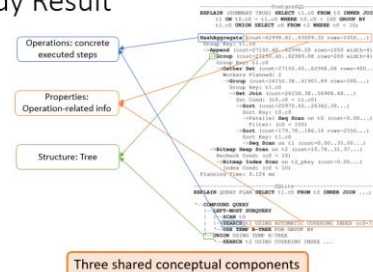
Query Plan



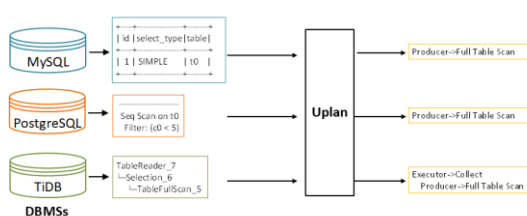
Query Plan Applications: Visualization



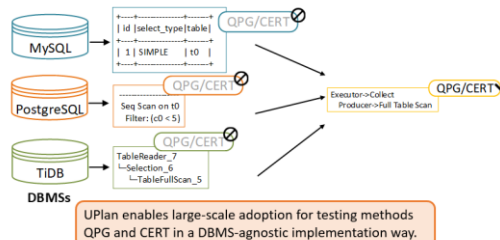
Study Result



Unified Query Plan Representation (Uplan)



Application: Testing



Application: Visualization

