Databend

为Databend添加SQLancer支持

Databend & 开源之夏

汇报人：韩艺松

目录
CONTENT

1 项目背景

2 测试方法讲解

3 实现方案及细节

4 总结与展望

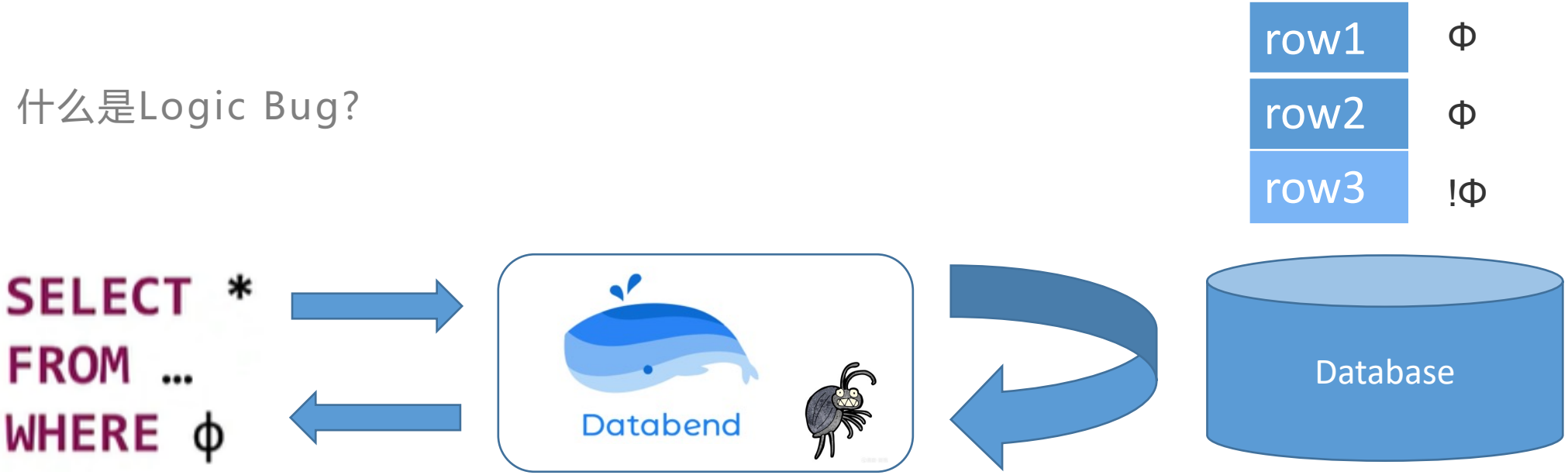
一、项目背景

为Databend实现SQLancer的意义

Databend & 开源之夏

Logic Bugs

什么是Logic Bug?



在DBMS中Logic Bug是导致DBMS返回不正确结果集的错误。

Logic Bugs

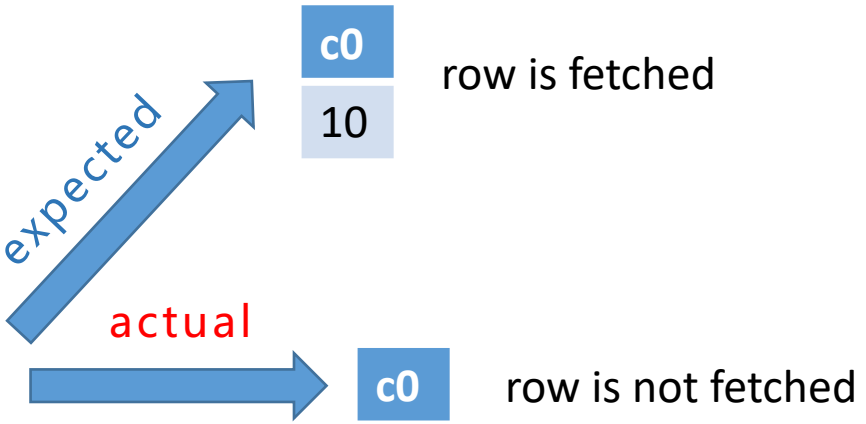
DBMS作为底层基础设施，大家普遍期望它是充分测试并可靠的。就跟任何一个大型软件一样，DBMS也不可避免地回存在Bug。而其中最危险也最让人难以捉摸的，莫过于Logic Bug了。

致命的Logic Bug

例如MySQL的Double negation Bug

```
CREATE TABLE t0 ( c0 INT );
INSERT INTO t0 ( c0 ) VALUES (10) ;

SELECT * FROM t0 WHERE 123 != ( NOT ( NOT 123) );
```



MySQL 似乎优化掉了双重否定，这对于布尔类型是正确的，但不适用于其他数据类型。

像这样返回错误结果的Logic Bug，很可能造成业务数据不一致。如果配合上Delete语句，甚至有可能造成所有业务数据的丢失，是非常危险的一类Bug。

隐蔽的Logic Bug

bug: `only_null` for nullable column returns true when column is empty · Issue #8000 · datafuselabs/databend (github.com)

SELECT (false and NULL NOT IN (0.1, 0.2, 0.3,0.4)) ::BIGINT; -- 结果返回0

SELECT (false and NULL NOT IN (0.1, 0.2, 0.3,0.4)) ::BIGINT FROM t1,t0;

ERROR 1105 (HY000): Code: 1010, displayText = Can't cast column from nullable data into non-nullable type (while in processor thread 0).

SELECT (false and NULL NOT IN (0.1, 0.2, 0.3)) ::BIGINT FROM t1,t0; -- 返回：empty set

DBMS领域的Test Oracles

在数据库测试理论中，Test Oracle是一种用于判断测试用例是否执行成功的机制。最简单的Test Oracle可以是单元测试中的assert语句，其他数据库也可以作为Test Oracle。

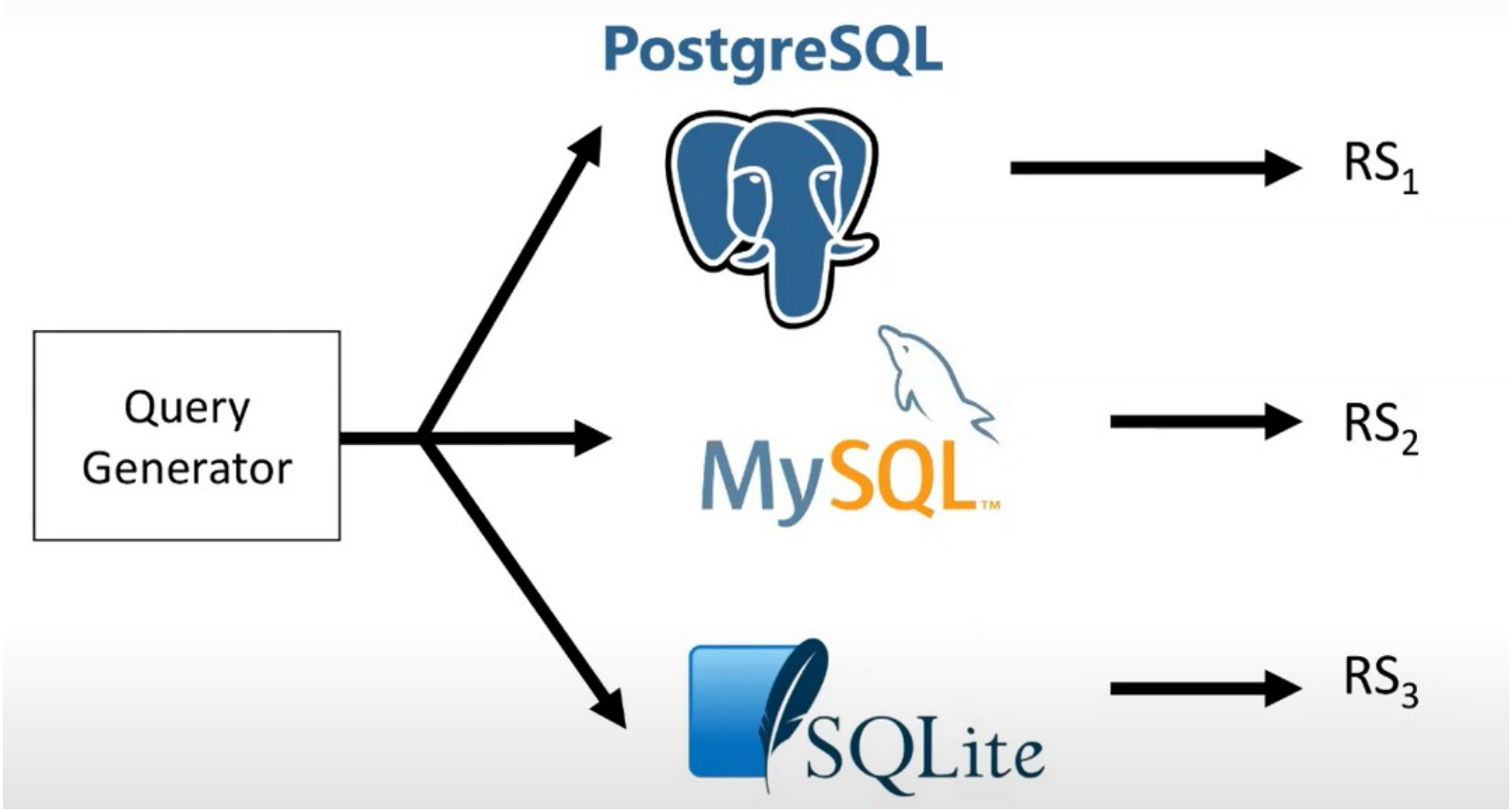
1. Differential Testing

2. Fuzzing

3. Manuel Rigger博士提出的：PQS、NoREC、TLP

Differential Testing

把相同的SQL发送给多个不同的数据库各自执行，如果结果不一致，则很有可能其中一个存在bug。这种方式被称之为Differential Testing。



$RS_1 = RS_2 = RS_3$?

Differential Testing

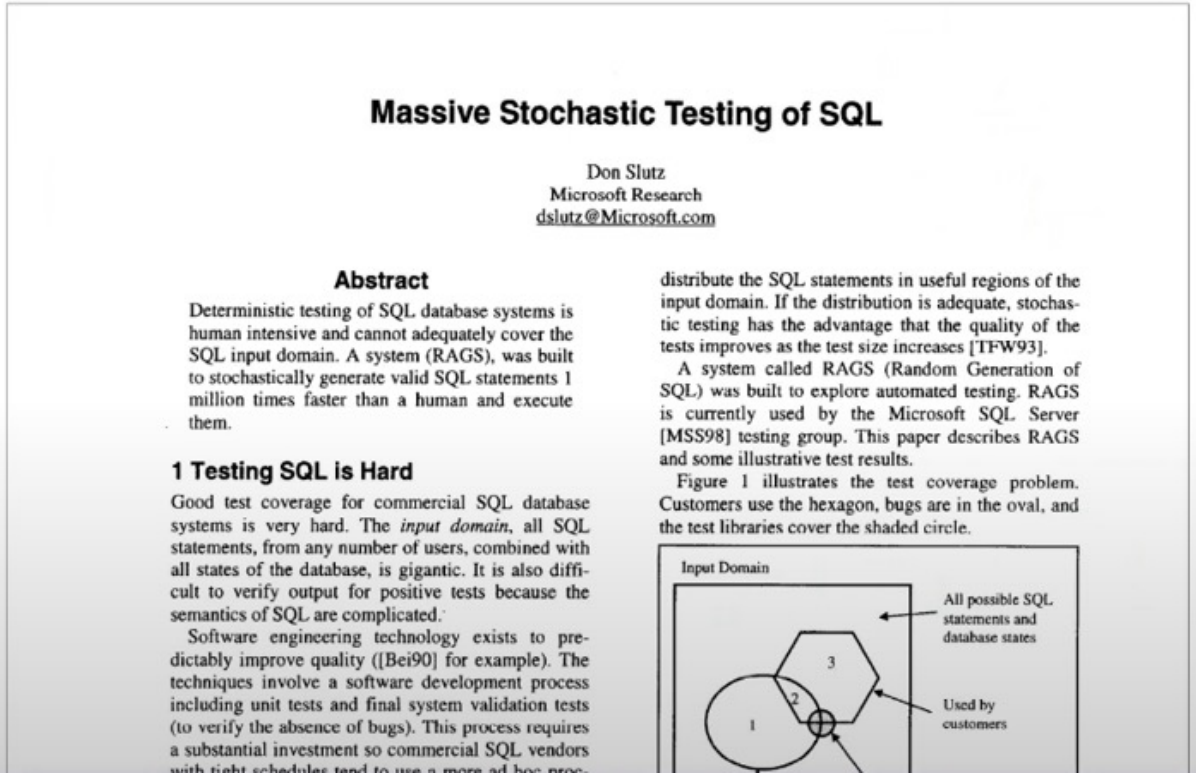
但是它的缺点也很明显，它不满足测试理论中的可靠性和完备性。比如：

1、如果2个数据库存在相同的逻辑bug，它是检测不出来的。

2、虽然大多数据库都遵循SQL标准，但是各种数据库的实现都有自己的方言和 feature，并且在标准不明确的地方不同的数据库也有各自不同的实现，导致很多时候并不通用，这样交叉验证的方式局限性很大。

这些缺点都造成了Differential Testing测试范围的狭小以及效果不够理想。

"[Differential testing] proved to be extremely useful, but only for the **small set of common SQL**"



Massive Stochastic Testing of SQL

Don Slutz
Microsoft Research
dslutz@Microsoft.com

Abstract

Deterministic testing of SQL database systems is human intensive and cannot adequately cover the SQL input domain. A system (RAGS), was built to stochastically generate valid SQL statements 1 million times faster than a human and execute them.

1 Testing SQL is Hard

Good test coverage for commercial SQL database systems is very hard. The *input domain*, all SQL statements, from any number of users, combined with all states of the database, is gigantic. It is also difficult to verify output for positive tests because the semantics of SQL are complicated.

Software engineering technology exists to predictably improve quality ([Bei90] for example). The techniques involve a software development process including unit tests and final system validation tests (to verify the absence of bugs). This process requires a substantial investment so commercial SQL vendors

distribute the SQL statements in useful regions of the input domain. If the distribution is adequate, stochastic testing has the advantage that the quality of the tests improves as the test size increases [TFW93].

A system called RAGS (Random Generation of SQL) was built to explore automated testing. RAGS is currently used by the Microsoft SQL Server [MSS98] testing group. This paper describes RAGS and some illustrative test results.

Figure 1 illustrates the test coverage problem. Customers use the hexagon, bugs are in the oval, and the test libraries cover the shaded circle.

Fuzzing

Fuzzing cannot find logic bugs

SEGMENTATION FAULT

了解Logic Bug的危险性和隐蔽性后，可以知道固化的用例
和手工的测试很难以有效地发现问题。

目前Databend的逻辑测试方式主要是从成熟的开源数据库上
迁移test case然后使用sqllogictest脚本跑测试样例
（Differential Testing），它存在一些不足：

1、测试局限性大，不够深入，容易遗漏Bug。
2、维护比较费力气，获得Bug效率低。

SQLancer

这时我们如果引入有效的随机化全自动测试，就能高效地发现这些逻辑bug。

SQLancer (Synthesized Query Lancer) 是一种自动测试DBMS的工具，以便在DBMS中发现logic bugs 。目前为MySQL、Postgresql、SQLite等元老级数据库抓出了400+BUG。

SQLancer可以对sqllogictest的不足进行补充，它具备Fuzzing（模糊测试具备领域感知能力）+生成Test oracle进行验证的能力（PQS、NoREC、TLP），全程高效自动化，可以为特定的数据库定制更有针对性的测试。

SQLancer的效果

在NoREC与TLP的初步实现后，短时间就触发Databend 20多个Bug，说明了SQLancer自动化测试的高效性。

[SQLacner for Databend 测试使用说明]

不禁回忆起在完善TLP的那个夜晚，几十分钟就触发了4个bug。

454 Open ✓ 2,434 Closed Author Label Projects M

- bug: Code: 4000, displayText = block pruning failure, task 4274595 panicked. C-bug sqlancer
  #7464 opened 17 minutes ago by hanyisong 1 of 2 tasks
- bug: the content of the result sets mismatch C-bug sqlancer
  #7463 opened 26 minutes ago by hanyisong 1 of 2 tasks
- feat: materialized views C-feature good first issue
  #7462 opened 27 minutes ago by BohuTANG
- bug: Code: 4000, displayText = unexpected end of file (failed to fill whole buffer) (while in processor thread 15). C-bug sqlancer
  #7461 opened 33 minutes ago by hanyisong 1 of 2 tasks
- bug: expression evaluation error C-bug sqlancer
  #7460 opened 42 minutes ago by hanyisong 1 of 2 tasks
- bug: where clause error C-bug sqlancer
  #7457 opened 7 hours ago by hanyisong 1 of 2 tasks

二、测试方法讲解

Manuel Rigger博士提出的NoREC、TLP、PQS

Databend & 开源之夏

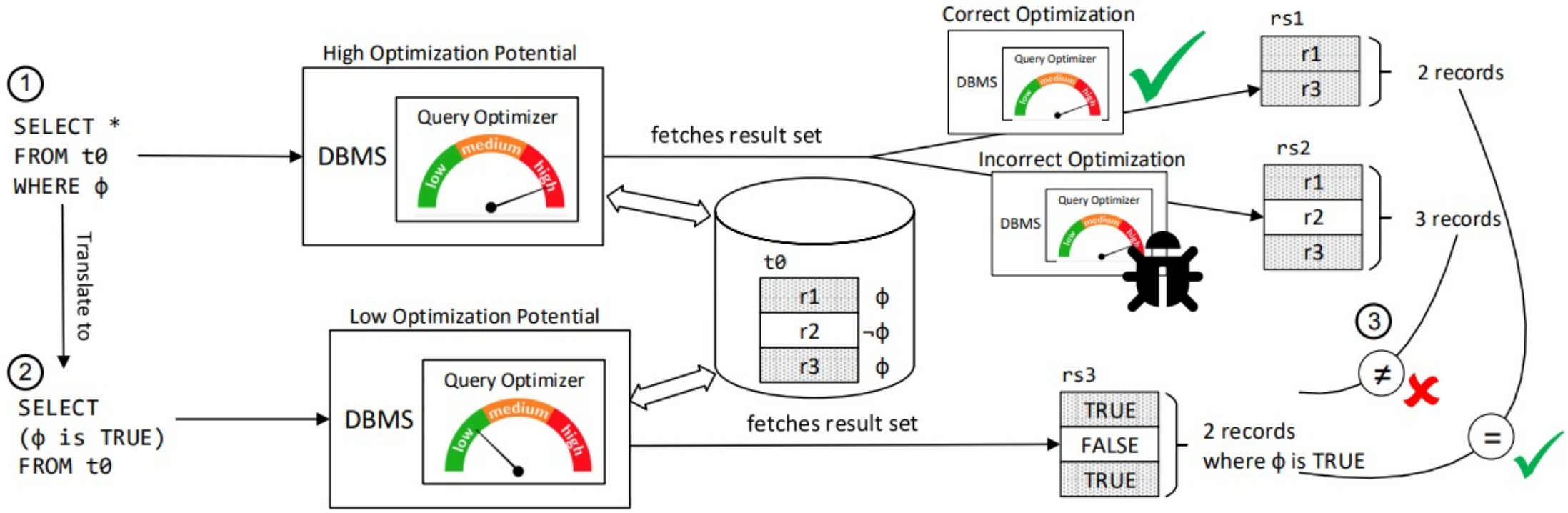
Testing Approaches

Approach	Description
Pivoted Query Synthesis (PQS)	PQS 随机选择一行，称为pivot row,，为其生成保证获取该行的query。 如果该行不包含在结果集中，则检测到错误。
Non-optimizing Reference Engine Construction (NoREC)	NoREC旨在发现优化错误。它将可能由 DBMS 优化的query转换为几乎没有任何优化适用的query，并比较两个结果集。 结果集之间的不匹配表明 DBMS 中存在错误。
Ternary Logic Partitioning (TLP)	TLP 将一个query分成三个分区query，其结果组合后并与原始查询的结果集进行比较。 若结果集不匹配则表示 DBMS 中的错误。 与 NoREC 和 PQS 相比，它可以检测聚合函数等高级功能中的错误。

NoREC测试方法

全称：Non-Optimizing Reference Engine Construction

Paper: https://arxiv.org/pdf/2007.08292.pdf



Bug Example

样例来源：
https://github.com/datafuselabs/databend/issues/7863

```
DROP DATABASE IF EXISTS databend1;
CREATE DATABASE databend1;
USE databend1;
CREATE TABLE t0(c0 BOOLEAN NULL DEFAULT(true));
CREATE TABLE t1(c0 DOUBLE NULL DEFAULT(0.1), c1 VARCHAR NULL);
INSERT INTO t1(c0) VALUES (0.2);
```

```
SELECT * FROM t1 LEFT JOIN t0 ON true WHERE (NOT -1);
```

t1.c0	t1.c1	t0.c0
0.2	NULL	NULL

```
SELECT SUM(count) FROM (
      SELECT (((NOT -1) IS NOT NULL AND (NOT -1)) ::BIGINT) as count
      FROM t1 LEFT JOIN t0 ON true
) as res;
```

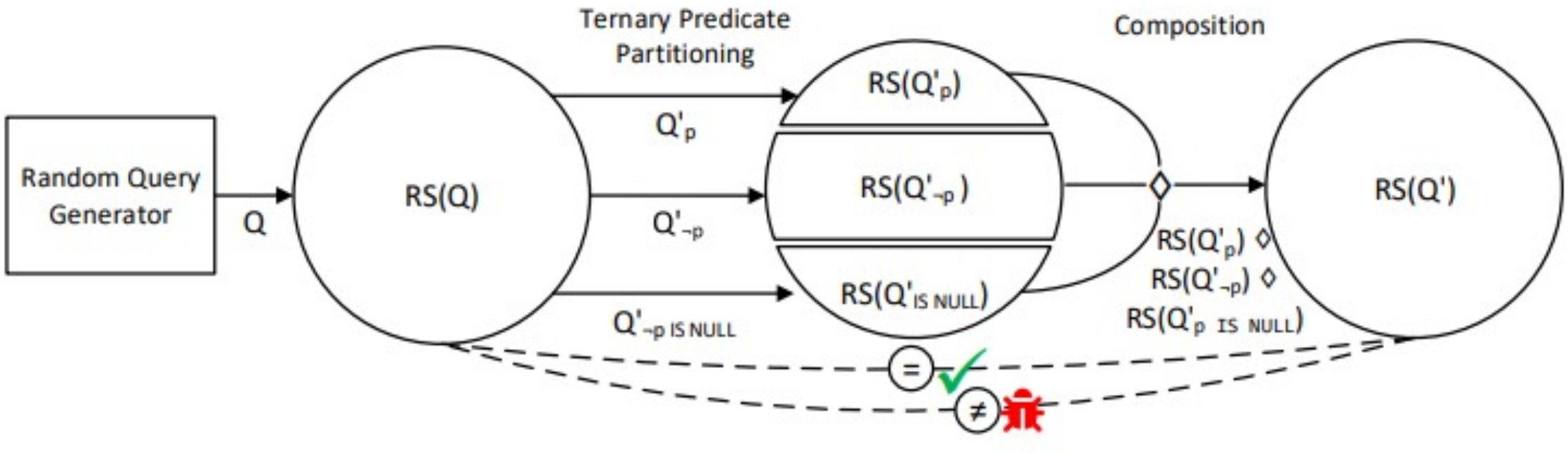
sum(count)
0

第一条query返回了一行，
而oracle为0行，所以是个Bug。

TLP测试方法

全称：Ternary Logic Partitioning

Paper: https://www.manuelrigger.at/preprints/TLP.pdf



TLP测试方法

Oracle	Q	Q'_{ptern}	$\diamond(Q'_p, Q'_{\neg p}, Q'_{p\ NULL})$

WHERE	`SELECT <columns>`
	`FROM <tables> [<joins>]`
		`SELECT <columns>`
		`FROM <tables> [<joins>]`
		`WHERE` p_{tern}
			$Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL}$

WHERE Extended	`SELECT <columns>`
	`FROM <tables> [<joins>]`
	`WHERE` p_{exist}
		`SELECT <columns>`
		`FROM <tables> [<joins>]`
		`WHERE` p_{exist} `AND` p_{tern}
			$Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL}$

GROUP BY	`SELECT <columns>`
	`FROM <tables> <joins>`
	`GROUP BY <columns>`
		`SELECT <columns>`
		`FROM <tables> <joins>`
		`WHERE` p_{tern}
		`GROUP BY <columns>`
			$Q'_p \cup Q'_{\neg p} \cup Q'_{p\ NULL}$

HAVING	`SELECT <columns>`
	`FROM <tables> <joins>`
	`[WHERE ...]`
	`[GROUP BY ...]`
		`SELECT <columns>`
		`FROM <tables> <joins>`
		`[WHERE ...]`
		`[GROUP BY ...]`
		`HAVING` p_{tern}
			$Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL}$

DISTINCT	`SELECT DISTINCT`
	`<columns>`
	`FROM <tables> <joins>`
		`SELECT [DISTINCT]`
		`<columns>`
		`FROM <tables> <joins>`
		`WHERE` p_{tern}
			$Q'_p \cup Q'_{\neg p} \cup Q'_{p\ NULL}$

TLP测试方法

Aggregate (MIN)	`SELECT MIN(<e>) FROM <tables> [<joins>]`	`SELECT MIN(<e>) FROM <tables> [<joins>] WHERE` p_{tern}	$MIN(Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL})$
Aggregate (MAX)	`SELECT MAX(<e>) FROM <tables> [<joins>]`	`SELECT MAX(<e>) FROM <tables> [<joins>] WHERE` p_{tern}	$MAX(Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL})$
Aggregate (SUM)	`SELECT SUM(<e>) FROM <tables> [<joins>]`	`SELECT SUM(<e>) FROM <tables> [<joins>] WHERE` p_{tern}	$SUM(Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL})$
Aggregate (COUNT)	`SELECT COUNT(<e>) FROM <tables> [<joins>]`	`SELECT COUNT(<e>) FROM <tables> [<joins>] WHERE` p_{tern}	$SUM(Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL})$
Aggregate (AVG)	`SELECT AVG(<e>) FROM <tables> [<joins>]`	`SELECT SUM(<e>) as s, COUNT(<e>) as c FROM <tables> [<joins>] WHERE` p_{tern}	$\frac{SUM(s(Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL}))}{SUM(c(Q'_p \uplus Q'_{\neg p} \uplus Q'_{p\ NULL}))}$

Bug Example


```
DROP DATABASE IF EXISTS databend1;
CREATE DATABASE databend1;
USE databend1;
CREATE TABLE t0(c0 BOOLEAN NULL DEFAULT(true));
INSERT INTO t0(c0) VALUES (false), (false), (true);
```

样例来源：
https://github.com/datafuselabs/databend/issues/7360

第一条query与oracle返回结果不一致，所以是个Bug。

```
SELECT t0.c0 FROM t0;
```

t0.c0
0
0
1

≠

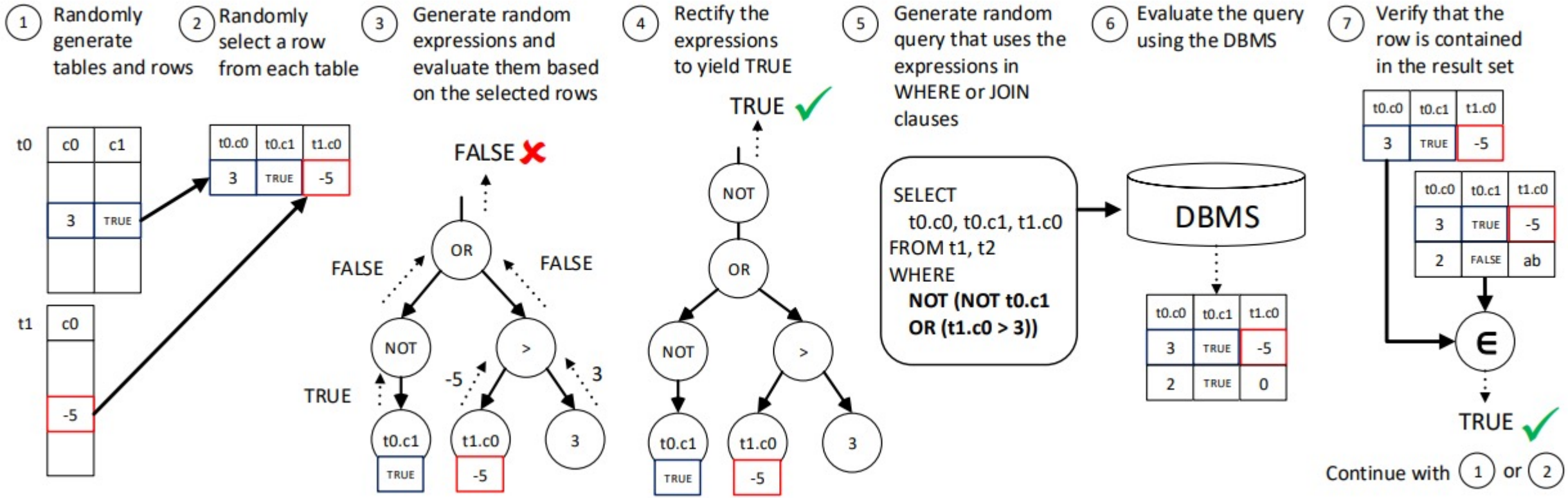
t0.c0
0
0
1
0
0
1

```
SELECT t0.c0 FROM t0 WHERE (NULL BETWEEN NULL AND NULL)
UNION ALL
SELECT t0.c0 FROM t0 WHERE (NOT (NULL BETWEEN NULL AND NULL))
UNION ALL
SELECT t0.c0 FROM t0 WHERE ((NULL BETWEEN NULL AND NULL) IS NULL);
```

PQS测试方法

全称：Pivoted Query Synthesis

Paper: https://arxiv.org/pdf/2001.04174.pdf



Comparison

Property	PQS	NoREC	TLP
WHERE	✓	✓	✓
Aggregates, HAVING, ...	✗	✗	✓
Ground truth	✓	✗	✗

PQS基于一个事实，而TLP、NoREC基于变形查询局限在没有一个基本事实。

三、实现细节

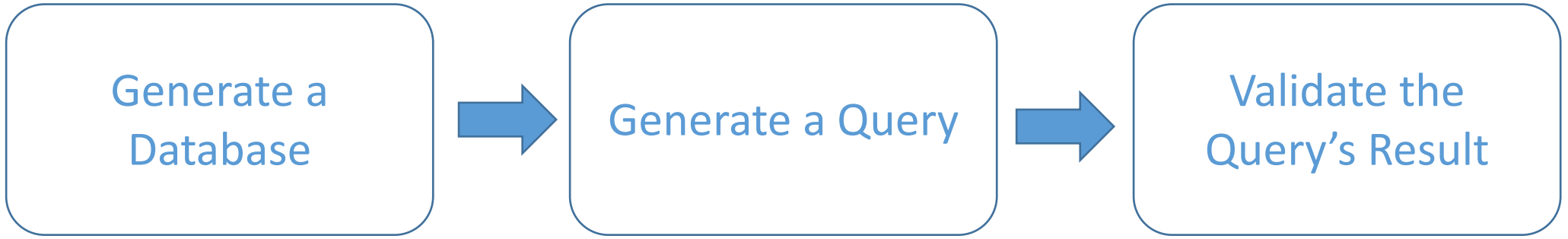
遇到的一些问题及解决方案

Databend & 开源之夏

自动化测试核心挑战

1. Effective test case

2. Test oracle



大概流程

1. provider类创建database。
2. provider类调用generateAndTestDatabase方法：
2.1 generateDatabase方法执行SQL生成一些table和view，然后再执行SQL获得table与column信息构造成DatabendSchema，最后将DatabendSchema注入到globalState。
2.2 接着执行oracle.check()进行验证。

四、总结与展望

Databend & 开源之夏

总结

通过对PQS、TLP、NoREC的探索，我发现它们都有个共同点：基于理论从而只需在DBMS上进行测试。

其中，TLP、NoREC通过计算不同的SQL语句获得的结果集，检测不同SQL获得结果集是否存在差异就如同sqlaner赋予了数据库自我反省的能力。

而PQS的oracle则基于事实去检测数据库是否符合事实。

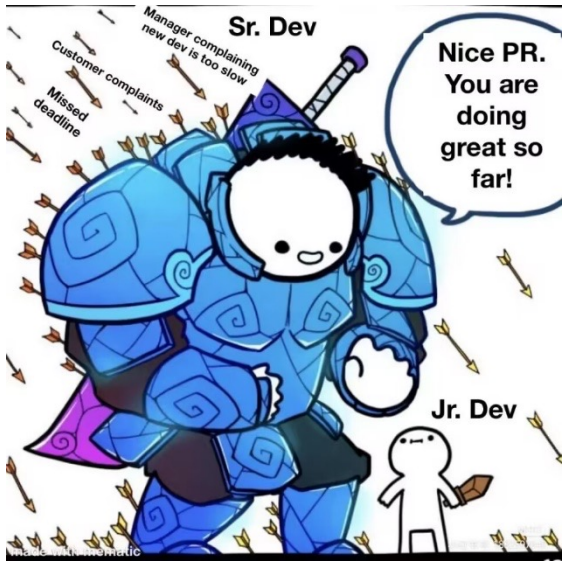
展望

自从看到SQLancer为Databend带来惊喜的测试效果后，我忍不住就画了个饼，
继续深入挖掘NoREC、TLP、PQS的潜在价值：

1、为随机化的生成器添加更多Databend的特性和方言，使之更有领域探索能力。
2、PQS据SQLancer官方说是最强大的技术，值得实现一下。
3、随着Databend的SQL语句优化工作的推动，NoREC能体现更大的价值。
4、TLP的效果是最好的而且只是分区查询的一种，值得研究一下其他的分区方案。

感谢

感谢许志清导师的鼓励与信任，还有Databend社区大牛们的帮助，开源之夏给我提供的了解开源和成长的机会，最后特别感谢SQLancer的创建者Manuel Rigger很热情地给我解答了许多疑问，并且他与我们分享了为Databend开发SQLancer实现的兴奋之情！




推文

Manuel Rigger
@RiggerManuel

Thanks to Yisong Han and @Datafuse_Labs for contributing testing support for Databend to @sqlancer_dbms! Hope it will help with preventing bugs before they reach any users. weekly.databend.rs/2022-09-28-dat...

翻译推文

Databend Automated Testing with SQLancer

Databend Automated Testing with SQLancer is one of the Databend community's projects in the Open Source Promotion Plan 2022. @hanyisong helped us with this important work, which has now been merged into sqlancer/sqlancer repository.

SQLancer (Synthesized Query Lancer) is a tool to automatically test Database Management Systems (DBMS) in order to find logic bugs in their implementation.

Learn more at https://github.com/sqlancer/sqlancer/pull/568

发表回复

Databend

感谢您的观看

THANK YOU FOR WATCHING

Databend & 开源之夏

汇报人：韩艺松

xx/xx/2022