



Bug hunting in Postgres

Robins Tharakan (he/him)

Postgres Database Engineer
Amazon Web Services



My postgres involvement

- pgAdmin – minor features
- psql – autocomplete support
- Minor Engine core changes
- Recently, aggressively fuzzing Postgres
 - and extensions

Agenda

Postgres Community – Interacting with & Reporting bugs

What is Fuzz-Testing?

Results overview OR Why should you be interested?

What is SQLSmith?

What is SQLReduce?

Demo

Basics of the components of a simple bug-report?




Postgres Community: Open-Source development

Interacting with the Postgres developer community

- Learn about Contributing to Postgres
 - Fantastic Talk by Claire Giordiano
 - How to Contribute to an Open Source Project like Postgres
 - PgConf New York (NYC 2022)
 - <https://youtu.be/Q8ST09rlFhs>
- Bug Reporting Guidelines
 - <https://www.postgresql.org/docs/devel/bug-reporting.html>
- Bug Reporting Wiki
 - https://wiki.postgresql.org/wiki/Guide_to_reporting_problems

Postgres Buildfarm


PostgreSQL BuildFarm

[Home](#)
[Status](#)
[Failures](#)
[Members](#)
[Register](#)
[Typedefs](#)
[GitHub](#)
[Email lists](#)

PostgreSQL BuildFarm Status

Shown here is the latest status of each farm member for each branch it has reported on in the last 30 days.

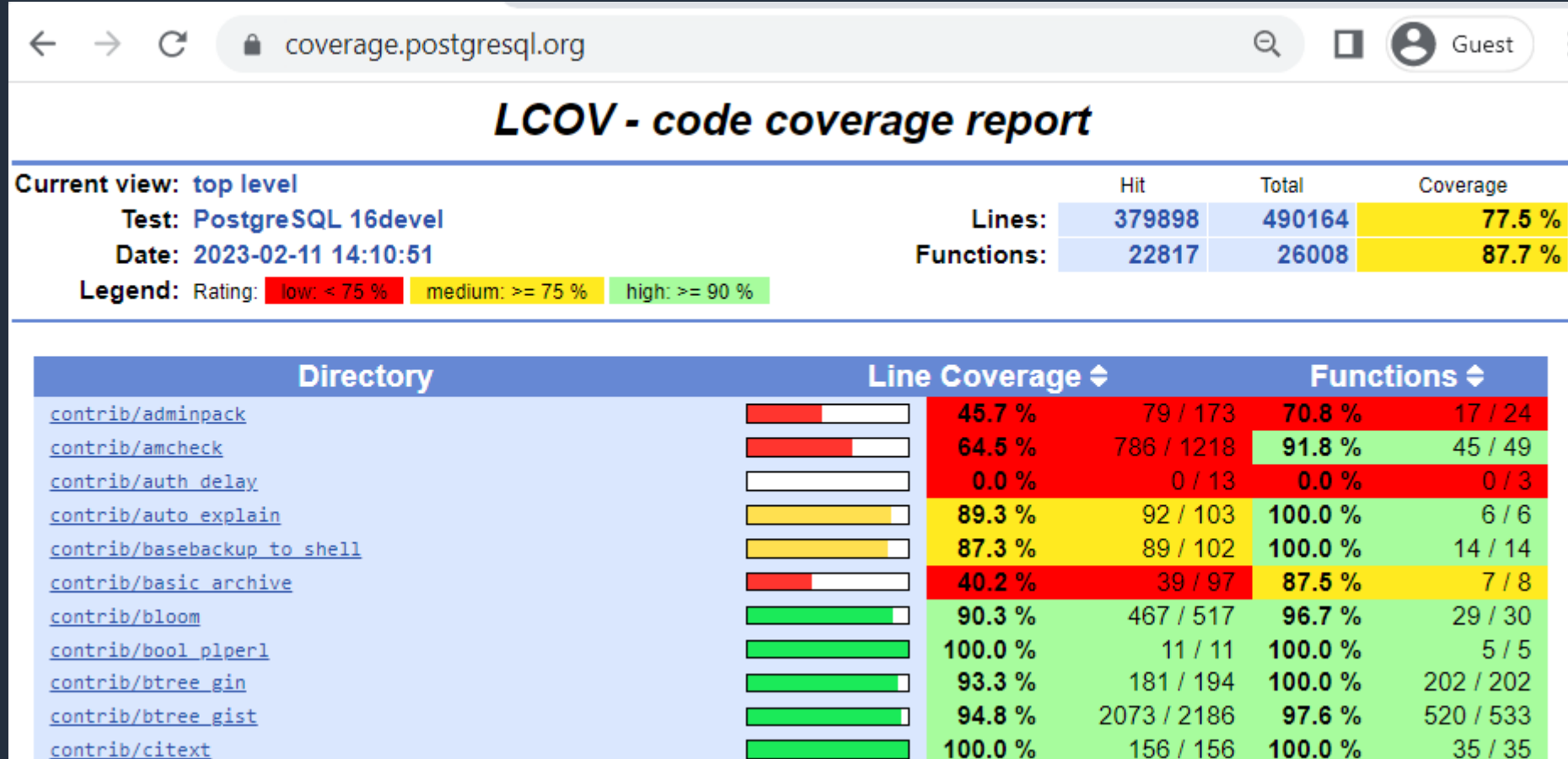
Use the farm member link for history of that member on the relevant branch.

Legend
 = cassert
 = debug
 = gssapi
 = krb5
 = llvm
 = nls
 = openssl

 = pam
 = perl
 = python
 = tap-tests
 = tcl
 = thread-safety
 = vpath
 = xml

Branch: HEAD		
Alias	System	Status
branta	Ubuntu 20.04.4 LTS (Focal Fossa) gcc 10.3.0 s390x (z15)	00:11 ago OK [44e56ba] Config
pipit	Red Hat Enterprise Linux 8.6 (Ootpa) gcc 8.5.0 s390x (z15)	01:05 ago OK [44e56ba] Config
komodoensis	Debian Sid gcc 7 x86_64	01:55 ago OK [44e56ba] Config
alabio	Debian GNU/Linux 11 (bullseye) gcc 10.2.1 x86_64	02:23 ago OK [44e56ba] Config
dragonet	Debian Sid clang clang 3.9 x86_64	02:27 ago OK [44e56ba] Config
desmoxytes	Debian Sid gcc gcc 10 x86_64	02:27 ago OK [44e56ba] Config
idiacanthus	Debian Sid clang gcc 10 x86_64	02:27 ago OK [44e56ba] Config
hamerkop	Windows Server 2016 Visual C++ Visual Studio 2017 AMD64	03:23 ago OK [44e56ba] Config
cottonmouth	Ubuntu 22.04.1 LTS (Jammy Jellyfish) gcc 12.1.0 s390x (z15)	04:08 ago OK [44e56ba] Config

Postgres code coverage



Bugs mailing list

pgsql-bugs

If you find a bug, please use the [bug reporting form](#).

Search the Archives

(enter a message-id to go directly to that message)

[Advanced Search](#)

View Archives

2023

[February 2023](#)

[View Archives](#)

[Download mbox](#)

[January 2023](#)

[View Archives](#)

[Download mbox](#)

2022

[December 2022](#)

[View Archives](#)

[Download mbox](#)

[November 2022](#)

[View Archives](#)

[Download mbox](#)

[October 2022](#)

[View Archives](#)

[Download mbox](#)

[September 2022](#)

[View Archives](#)

[Download mbox](#)

[August 2022](#)

[View Archives](#)

[Download mbox](#)

Postgres Community – Contributions

- Community has been very receptive
- But most open-source interactions demand quality
- Contributions
 - Document improvements
 - Ideas / Improvements
 - Bug-Reports
 - SQL Repros
 - Patches etc..

Postgres Community Bug Report

BUG #17788: Incorrect memory access when parsing interval

From: PG Bug reporting form <noreply(at)postgresql(dot)org>
 To: pgsql-bugs(at)lists(dot)postgresql(dot)org
 Cc: exclusion(at)gmail(dot)com
 Subject: BUG #17788: Incorrect memory access when parsing empty string as sql_standard
 Date: 2023-02-12 10:00:01
 Message-ID: 17788-dabac9f98f7eafd5@postgresql.org
 Views: [Raw Message](#) | [Whole Thread](#) | [Download mbox](#) | [Resend email](#)
 Thread: 2023-02-12 10:00:01 from PG Bug reporting form <noreply(at)postgresql(dot)org>
 Lists: [pgsql-bugs](#)

The following bug has been logged on the website:

Bug reference: 17788
 Logged by: Alexander Lakhin
 Email address: exclusion(at)gmail(dot)com
 PostgreSQL version: 15.2
 Operating system: Ubuntu 22.04
 Description:

When executing under valgrind:
 SET IntervalStyle TO sql_standard;
 SELECT '':interval;

The following error is detected:
 ==00:00:00.03 574 1155861== Use of uninitialised value of size 8



pgsql: Avoid dereferencing an undefined pointer in DecodeInterval()

From: Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
 To: pgsql-committers(at)lists(dot)postgresql(dot)org
 Subject: pgsql: Avoid dereferencing an undefined pointer in DecodeInterval().
 Date: 2023-02-12 17:51:02
 Message-ID: E1pRGVC-000mYP-7e@gemulon.postgresql.org
 Views: [Raw Message](#) | [Whole Thread](#) | [Download mbox](#) | [Resend email](#)
 Thread: 2023-02-12 17:51:02 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
 Lists: [pgsql-committers](#)

Avoid dereferencing an undefined pointer in DecodeInterval().

Commit e39f99046 moved some code up closer to the start of DecodeInterval(), without noticing that it had been implicitly relying on previous checks to reject the case of empty input. Given empty input, we'd now dereference a pointer that hadn't been set, possibly leading to a core dump. (But if we fail to provoke a SIGSEGV, nothing bad happens, and the expected syntax error is thrown a bit later.)

Per bug #17788 from Alexander Lakhin. Back-patch to v15 where the fault was introduced.

Discussion: <https://postgr.es/m/17788-dabac9f98f7eafd5@postgresql.org>

Branch

 REL_15_STABLE

Details

<https://git.postgresql.org/pg/commitdiff/0ef65d0f55e5cec81fe98aba7c907dfc1b93923f>

Modified Files

 src/backend/utils/adt/datetime.c | 2 +-
 src/test/regress/expected/interval.out | 5 ++++
 src/test/regress/sql/interval.sql | 3 +++
 3 files changed, 9 insertions(+), 1 deletion(-)

Fuzz Testing

... is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs...

...(and then wish for some fun 😊 !)

What is SQLSmith?

**“I love the smell of coredumps
in the morning”**

(... from SQLSmith README)

What is SQLSmith?

- Open source fuzz-testing tool
 - Written by Andreas Seltenreich (and others)
- Random SQL query generator
- Works with multiple databases – Postgres / MonetDB etc.
 - For e.g. It was a quick effort to add basic Redshift support
- Allows query / error logging to a database (very helpful!)
- Efficient / allows concurrent runs

Why should I be interested 😊 ?

SQLSmith Scorecard

SQLSmith

- 100+ bugs found
- Projects
 - Postgres
 - Sqlite3
 - MonetDB
 - YugaByte
 - TimescaleDB
- Extensions
 - Orafce etc.

This Project

- 15+ bugs found
- Projects
 - Postgres
 - Redshift
- Extensions
 - postgis*
 - Orafce
 - Mysql_fdw
 - Rdkit
 - pg_repack
 - etc.

Community Feedback

"I'd hoped that some of these failures shared a root cause, but nope they were really four different bugs :-(. I've now pushed fixes for all four, and I hope you'll turn your fuzzer back on."

This is very valuable testing."

regards, tom lane

1 UPDATE SQL
2900 lines
120,000 characters

Single random SQL

Demo Time !

SQLSmith intro

Lets trigger SQLSmith, where we:

- Point to the target DB
- Generate 1 query
 - But don't print that SQL
- Do NOT run the SQL

```
BugHunt@PgConfIndia> sqlsmith --version
SQLSmith v1.4-97-gb1db8ac
```

```
BugHunt@PgConfIndia> sqlsmith --max-queries=1 --target="$TCONNSTR" --dry-run >/dev/null
SQLSmith v1.4-97-gb1db8ac
Loading types...done.
Loading tables...done.
Loading columns and constraints...100...200...300...400...500...600...700...800...900...1000...1100...done.
Loading operators...done.
Loading routines...done.
Loading routine parameters...1000...2000...3000...done.
Loading aggregates...done.
Loading aggregate parameters...done.
Generating indexes...done.
```

SQLSmith Demo - 2

Now let's see that 1 SQL it generated

- This SQL changes each time
- We aren't yet running on the database
- Can be very short or very long

```
BugHunt@PgConfIndia> sqlsmith --max-queries=1 --target="$TCOM
SQLsmith v1.4-97-gb1db8ac
Loading types...done.
Loading tables...done.
Loading columns and constraints...100...200...300...400...500
Loading operators...done.
Loading routines...done.
Loading routine parameters...1000...2000...3000...done.
Loading aggregates...done.
Loading aggregate parameters...done.
Generating indexes...done.

select
  public.cashmultirange() as c0
from
  pg_catalog.pg_description as sample_0 tablesample system (1
    left join public.test_timestz as sample_1 tablesample sy
    on (pg_catalog.inet_server_addr() >= cast(nullif((sele
      ,
      (select i from public.test_inet limit 1 offset 1)
      ) as inet))
  inner join (select
    ref_0.latest_end_lsn as c0,
    ref_0.last_msg_send_time as c1,
    ref_0.pid as c2
```

SQLSmith Demo - 3

Now let's see a summary report

- Again, 1 SQL only
- See the 'e' line
- See ERROR line
- Interesting to see the SQL Components
- See bad / ok per component

```

BugHunt@PgConfIndia> sqlsmith --max-queries=1 --verbose --target="$TC
SQLsmith v1.4-97-gblldb8ac
Loading types...done.
Loading tables...done.
Loading columns and constraints...100...200...300...400...500...600..
Loading operators...done.
Loading routines...done.
Loading routine parameters...1000...2000...3000...done.
Loading aggregates...done.
Loading aggregate parameters...done.
Generating indexes...done.
e
queries: 1
AST stats (avg): height = 14 nodes = 114
1      ERROR:  operator does not exist: anymultirange &> int4range
error rate: 1
impedance report:
  case_expr: 1/0 (bad/ok)
  funcall: 1/0 (bad/ok)
  atomic_subselect: 1/0 (bad/ok)
  const_expr: 1/0 (bad/ok)
  column_reference: 1/0 (bad/ok)
  nullif: 1/0 (bad/ok)
  truth_value: 1/0 (bad/ok)
  null_predicate: 1/0 (bad/ok)
  bool_term: 1/0 (bad/ok)
  comparison_op: 1/0 (bad/ok)
  table_or_query_name: 1/0 (bad/ok)
  table_sample: 1/0 (bad/ok)
  table_subquery: 1/0 (bad/ok)
  expr_join_cond: 1/0 (bad/ok)
  joined_table: 1/0 (bad/ok)
  from_clause: 1/0 (bad/ok)
  select_list: 1/0 (bad/ok)
  query_spec: 1/0 (bad/ok)

```

SQLSmith Demo - 4

Now let's time query generation

- TLDR – 600ms per 100 SQLs
 - Or 6ms per 1 SQL query generation

```
BugHunt@PgConfIndia> time sqlsmith --max-queries=1 --
SQLsmith v1.4-97-gb1db8ac
Loading types...done.
Loading tables...done.
Loading columns and constraints...100...200...300...4
Loading operators...done.
Loading routines...done.
Loading routine parameters...1000...2000...3000...dor
Loading aggregates...done.
Loading aggregate parameters...done.
Generating indexes...done.
25
```

```
real    0m16.473s
user    0m0.603s
sys     0m0.104s
```

```
BugHunt@PgConfIndia> time sqlsmith --max-queries=100
SQLsmith v1.4-97-gb1db8ac
Loading types...done.
Loading tables...done.
Loading columns and constraints...100...200...300...4
Loading operators...done.
Loading routines...done.
Loading routine parameters...1000...2000...3000...dor
Loading aggregates...done.
Loading aggregate parameters...done.
Generating indexes...done.
7370
```

```
real    0m17.014s
user    0m0.734s
sys     0m0.088s
```

SQLSmith Demo - 5

Now let's see a summary report of 100 SQLs

```

Loading aggregates...done.
Loading aggregate parameters...done.
Generating indexes...done.
.ee..e...ee.e.eSe.eeeee.e.e.ee.eee.eS...eeeeee.ee.ee...ee.eSSe.eee.ee.e.SeS.eee.
.e...eS.eee.e.eSe.eee..
queries: 100
AST stats (avg): height = 8.49 nodes = 80.42
4      ERROR:  argument declared anymultirange is not a multirange type but ty
3      ERROR:  operator does not exist: point = point
3      ERROR:  syntax error at or near "4"
3      ERROR:  syntax error at or near "6"
2      ERROR:  argument declared anyrange is not a range type but type anyrang
2      ERROR:  cannot accept a value of type table_am_handler
2      ERROR:  cannot cast type unknown to anyenum
2      ERROR:  column "col" does not exist
2      ERROR:  function pg_catalog.tsmultirange(tsrage[]) does not exist
2      ERROR:  function pg_catalog.tstzmultirange(tstzrange[]) does not exist
2      ERROR:  operator does not exist: cstring = cstring
1      ERROR:  cannot accept a value of type event_trigger
1      ERROR:  cannot accept a value of type trigger

```

SQLSmith Demo - 6

Lets log that to a database!

```
BugHunt@PgConfIndia> sqlsmith --max-queries=1 --target="$TCONNSTR" --log-to="$LCONNSTR"  
SQLsmith v1.4-97-gb1db8ac  
Loading types...done.  
Loading tables...done.  
Loading columns and constraints...100...200...300...400...500...600...700...800...900...  
Loading operators...done.  
Loading routines...done.  
Loading routine parameters...1000...2000...3000...done.
```

SQLSmith Demo - 7

Let's look inside the Logging Database

```
BugHunt@PgConfIndia> psql "$LCONNSTR"
psql (16devel, server 13.8)
Type "help" for help.

sqlsmithlatest=> \x
Expanded display is on.
sqlsmithlatest=> select * from error where length(query) <200 and query not ilike '%ALTER%' order by t desc limit 1;
-[ RECORD 1 ]-----
id          | 612748
msg         | ERROR:  PL/Tcl functions cannot return type language_handler+
query       | select                                     +
            |   public.float8multirange() as c0         +
            | from                                       +
            |   public.prem1 as ref_0                  +
            | where pg_catalog.pltcl_call_handler() is not NULL +
            | limit 90
target      |
sqlstate    | 0A000
t           | 2023-02-23 22:44:17.176887+00
errid       | 2491769780
```


SQLSmith Demo - 8

Lets assume that ErrID = 2354535915 is super interesting

```
sqlsmithlatest=> select msg, sqlstate, t, errid, left(query,200) from error
re errid= 2354535915;
-[ RECORD 1 ]-----
msg          | server closed the connection unexpectedly          +
              |               This probably means the server terminated abnormally+
              |               before or while processing the request.      +
              |
sqlstate     | 08000
t            | 2023-02-10 09:19:32.321505+00
errid        | 2354535915
left         | WITH
              | jennifer_0 AS (select
              |     public.textmultirange() as c0,
              |     pg_catalog.jsonb_build_object() as c1
              |   from
              |     (select
              |       29 as c0,
              |       sample_0.customer_id as c1,
```

SQLSmith Demo - 9

Fetch the full SQL from SQLSmith database

```
BugHunt@PgConfIndia> psql -Atq "$LCONNSTR" -c "select query from error_archive
where errid= 2354535915;" > reduce.sql

BugHunt@PgConfIndia> wc -l reduce.sql
66 reduce.sql

BugHunt@PgConfIndia> head reduce.sql
WITH
jennifer_0 AS (select
    public.textmultirange() as c0,
    pg_catalog.jsonb_build_object() as c1
from
    (select
        29 as c0,
        sample_0.customer_id as c1,
        54 as c2,
        (select seqno from public.bt_i4_heap limit 1 offset 37)
```

SQLSmith Demo - 10

Have prepped the database for a given state

- Note that the Postgres cluster name is the git commit (easier to track)
- The git commit is specifically chosen (you'll soon see why)

```
BugHunt@PgConfIndia> ps xf | grep postgres
2141013 ?          SNs      0:00 /home/ubuntu/proj/sqlsmithdata/temp/bin/postgres -D /home/ubuntu/pro
2141019 ?          SNs      0:01 \_ postgres: f8ba1bf4e4@master@sqith: checkpointer
2141020 ?          SNs      0:02 \_ postgres: f8ba1bf4e4@master@sqith: background writer
2141027 ?          SNs      0:05 \_ postgres: f8ba1bf4e4@master@sqith: walwriter
2141028 ?          SNs      0:00 \_ postgres: f8ba1bf4e4@master@sqith: autovacuum launcher
2141029 ?          SNs      0:00 \_ postgres: f8ba1bf4e4@master@sqith: autoprewarm leader
2141030 ?          SNs      0:00 \_ postgres: f8ba1bf4e4@master@sqith: logical replication launcher
```

SQLSmith Demo - 11

Let's crash postgres!

```
BugHunt@PgConfIndia> psql "$TCONNSTR" -f reduce.sql
psql:reduce.sql:66: server closed the connection unexpectedly
        This probably means the server terminated abnormally
        before or while processing the request.
psql:reduce.sql:66: error: connection to server was lost
```

Postgres Error Logs details it well

```
2023-02-23 22:59:52.731 UTC [2141019] LOG:  checkpoint starting: time
2023-02-23 23:02:24.777 UTC [2141019] LOG:  checkpoint complete: wrote 1515 buffers (
write=152.031 s, sync=0.001 s, total=152.046 s; sync files=239, longest=0.001 s, ave
kB; lsn=6/AE7A63C8, redo lsn=6/AE4234D8
TRAP: failed Assert("outer_rel->rows > 0"), File: "indxpath.c", Line: 1909, PID: 2184
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(Exceptiona
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(+0x4ef67a)
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(+0x4edb5e)
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(+0x4ee261)
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(+0x4ee4cb)
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(create_ind
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(+0x4d3f89)
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(+0x4d3af6)
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(+0x4d373c)
postgres: f8balbf4e4@master@sqith: ubuntu postgres 127.0.0.1(44834) SELECT(make one r
```

SQLReduce

Christoph Berg

- <https://github.com/credativ/sqlreduce>
- **Input**
 - Accepts an SQL
- **Output**
 - Optionally accepts a desired output (or by default await an engine crash)
- **Action**
 - Iteratively shorten SQL while achieving desired output
- **Result**



Returns the shortest SQL

SQLSmith Demo - 12

SQLReduce basic working

Input query: `select '1'::date from pg_class where 1=1;`

Regenerated: `SELECT CAST('1' AS date) FROM pg_class WHERE 1 = 1`

Query returns: ▼ 22007

`SELECT FROM pg_class WHERE 1 = 1` ✗ no error

`SELECT CAST('1' AS date) WHERE 1 = 1` ▼

`SELECT WHERE 1 = 1` ✗ no error

`SELECT CAST('1' AS date)` ▼

`SELECT` ✗ no error

`SELECT NULL` ✗ no error

`SELECT '1'` ✗ no error

`SELECT CAST(NULL AS date)` ✗ no error

Minimal query yielding the same error:

`SELECT CAST('1' AS date)`

Pretty-printed minimal query:

`SELECT CAST('1' AS date)`

Seen: 9 items, 212 Bytes

Iterations: 8

Runtime: 0.040 s, 223.2 q/s

SQLReduce Demo – 12a

SQLSmith Demo - 13

Remove local table / data dependency, if possible

```
t=# BEGIN;
BEGIN
t=*# CREATE TABLE t();
CREATE TABLE
t=*# SELECT
t-*# FROM t AS sample_1
t-*#         INNER JOIN (pg_catalog.pg_user AS ref_1
t(*#         RIGHT JOIN pg_catalog.pg_conversion AS ref_2 ON NULL)
t-*#         ON ref_2.conname = ref_1.passwd
t-*#         OR (SELECT calls
t(*#         FROM pg_catalog.pg_stat_xact_user_functions) <= 52;
server closed the connection unexpectedly
        This probably means the server terminated abnormally
        before or while processing the request.
The connection to the server was lost. Attempting reset: Succeeded.
t=# █
```


Capture Backtrace, if possible

```
BugHunt@PgConfIndia> less /home/ubuntu/proj/sqlsmithdata/review_coredump/review_070623/coredump_9e8b694d81-master-20230210T091930Z-postgres-127.0.0-core-1

== Coredump filestamp ==
-rw----- 1 ubuntu ubuntu 1.1G Feb 10 09:19 /home/ubuntu/proj/sqlsmithdata/oldcores/core-postgres-1676020770-sig6-1

=== Coredump MD5 ===
95c0f9e537f86611f09fb95f9b7d2df6   /home/ubuntu/proj/sqlsmithdata/oldcores/core-postgres-1676020770-sig6-1

=== Coredump Epoch Timestamp ===
1676020770 => Fri Feb 10 09:19:30 UTC 2023

=== Coredump stat ===
File: /home/ubuntu/proj/sqlsmithdata/oldcores/core-postgres-1676020770-sig6-1
Size: 1074868224      Blocks: 1915128      IO Block: 4096    regular file
Device: 10302h/66306d Inode: 137          Links: 1
Access: (0600/-rw-----) Uid: ( 1000/   ubuntu)   Gid: ( 1000/   ubuntu)
Access: 2023-02-10 11:49:15.756629573 +0000
Modify: 2023-02-10 09:19:32.179308158 +0000
Change: 2023-02-10 11:49:05.464737872 +0000
Birth: -

=== Backtrace - PID 2918599 - 9e8b694d81@master ===
Reading symbols from /home/ubuntu/proj/sqlsmithdata/temp/bin/postgres...
[New LWP 2918599]
[thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by `postgres: 9e8b694d81@master@sqith: u28 postgres 127.0.0'.
Program terminated with signal SIGABRT, Aborted.
#0 __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
#0 __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
#1 0x00007fb4b671a859 in __GI_abort () at abort.c:79
#2 0x0000564136170ff3 in ExceptionalCondition (conditionName=0x564136324ff0 "out of memory") at elog.c:125
#3 0x0000564135e2767a in get_loop_count (root=0x564138eb8520, cur_relid=3, over_commit_mode=<error: maximum recursion depth exceeded while calling Python C API function '<>'>) at relutils.c:125
#4 0x0000564135e25b5e in build_index_paths (root=0x564138eb8520, rel=0x564142b8520, indexrel=0x564142b8520, pathlist=<error: maximum recursion depth exceeded while calling Python C API function '<>'>, npaths=<error: maximum recursion depth exceeded while calling Python C API function '<>'>) at relutils.c:125
```

SQLSmith Demo – 15

So going back, how did I get to **f8ba1bf4e4**?

```
BugHunt@PgConfIndia> ./blamever2.sh 36 39

SQL
===
rollback;
begin;
create table t();
SELECT
FROM t AS sample_1
      INNER JOIN (pg_catalog.pg_user AS ref_1
                  RIGHT JOIN pg_catalog.pg_conversion AS ref_2 ON NULL)
      ON ref_2.conname = ref_1.passwd
      OR (SELECT calls
          FROM pg_catalog.pg_stat_xact_user_functions) <= 52

Checking (f8ba1bf4e4~36) - 8538519db1 - Crash
Checking (f8ba1bf4e4~37) - 5840c20272 - Success
Checking (f8ba1bf4e4~38) - faff8f8e47 - Success
Checking (f8ba1bf4e4~39) - 1b6f632a35 - Success
```

What else?

- SQLancer
- SQL-Reduce etc....



Thank you!

Robins Tharakan

<https://www.thatguyfromdelhi.com/>

