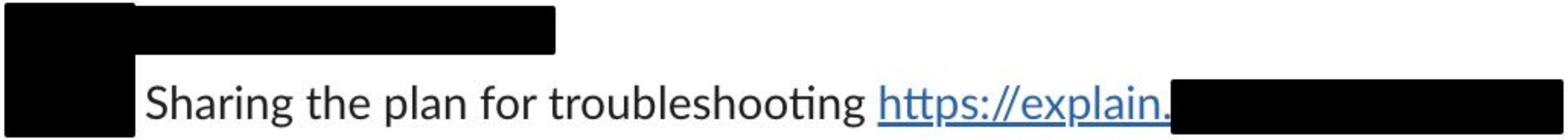
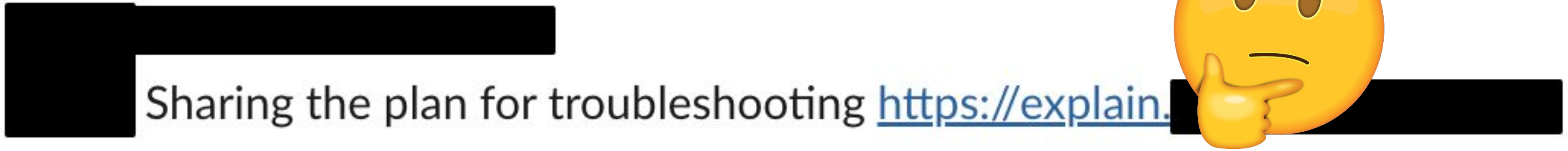
From EXPLAIN to Exposed: A Plan Gone Sideways

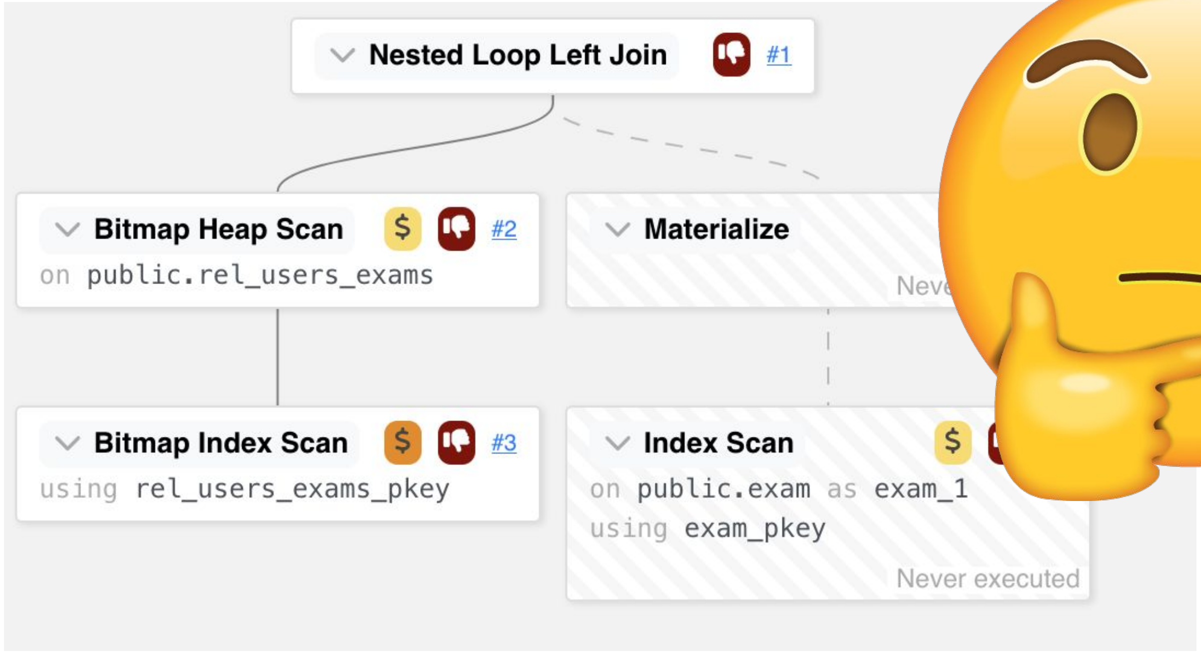
slack



[redacted] [redacted]

Sharing the plan for troubleshooting https://explain.[redacted] slack


[redacted]
Sharing the plan for troubleshooting https://explain.[redacted]



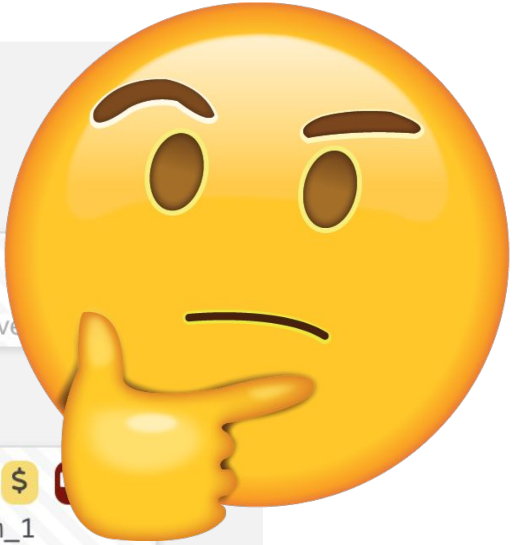
Nested Loop Left Join #1

Bitmap Heap Scan $ #2
on public.rel_users_exams

Materialize
Neve

Bitmap Index Scan $ #3
using rel_users_exams_pkey

Index Scan $
on public.exam as exam_1
using exam_pkey
Never executed



```
Nested Loop Left Join  (cost=11.95..28.52 rows=5 width=157) (actual
  Output: rel_users_exams.user_username, rel_users_exams.exam_id,
  Inner Unique: true
  Join Filter: (exam_1.id = rel_users_exams.exam_id)
  Buffers: sha
  ->  Bitmap H
        Output
        Rechec
        Buffer
        ->  Bi


  ->  Material
        Output
        ->  Ir


Planning Time:
Execution Time
```

```
SELECT rel_users_exams.user_username
            rel_users_exams.exam_id AS
            rel_users_exams.started_at
            rel_users_exams.finished_at
            exam_1.id AS exam_1_id,
            exam_1.title AS exam_1_title
            exam_1.date_from AS exam_1_da
            exam_1.date_to AS exam_1_date_
            exam_1.created AS exam_1_create
            exam_1.created_by_ AS exam_1_crea
            exam_1.duration AS exam_1_duratio
            exam_1.success_threshold AS exam_
            exam_1.published AS exam_1_publis
FROM rel_users_exams LEFT OUTER
JOIN exam AS exam_1
    ON exam_1.id = rel_users_exams.exam_id
WHERE 1 = rel_users_exams.exam_id;
```

EXPLAIN YOURSELF
amc HD
PREMIERE

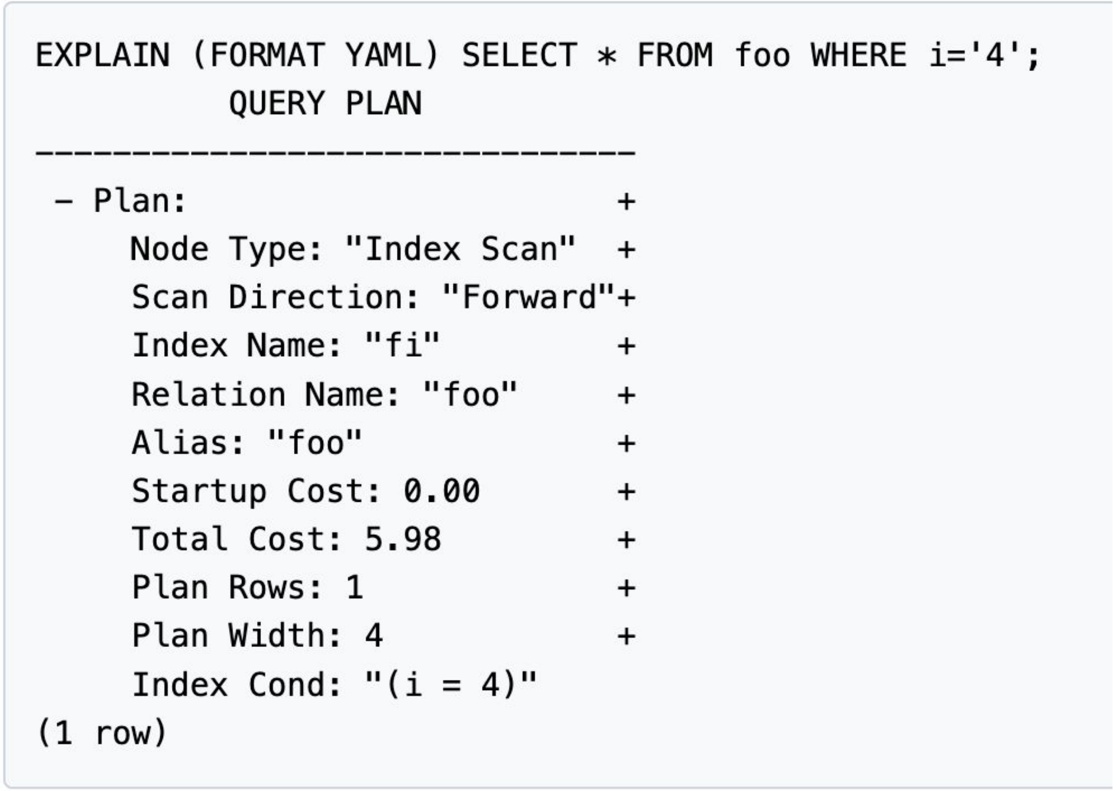
What is EXPLAIN anyway ?

What is EXPLAIN anyway ?

EXPLAIN

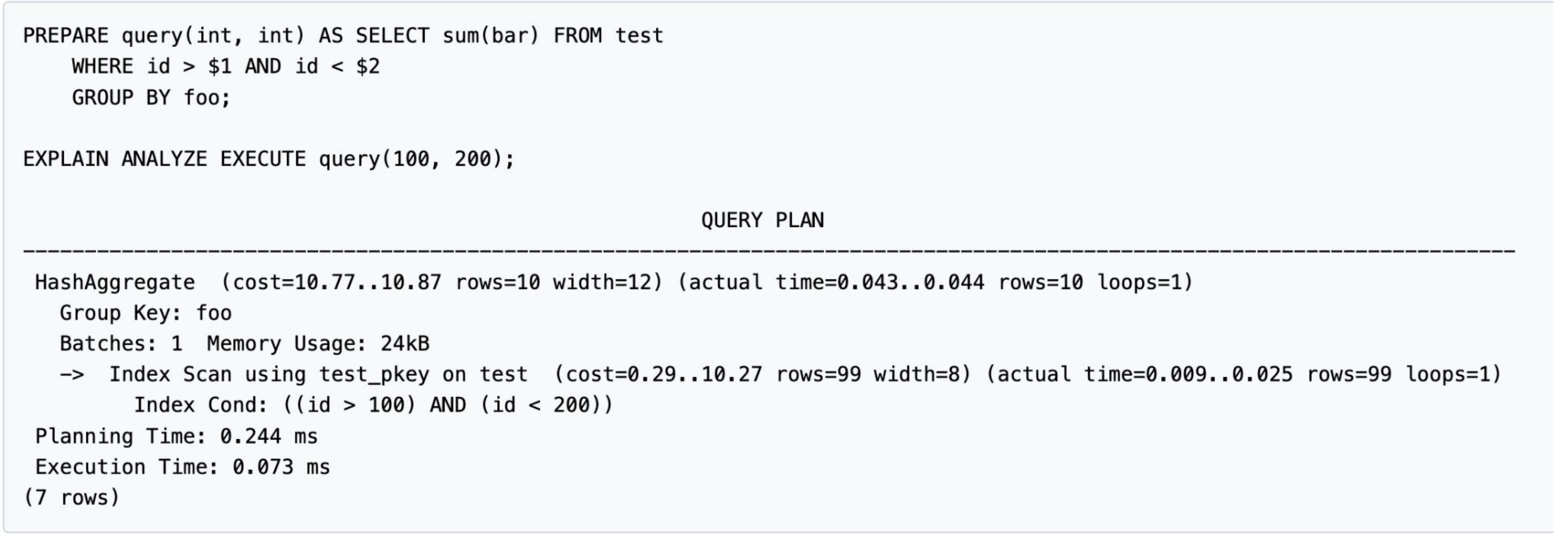
EXPLAIN — show the execution plan of a statement

What is EXPLAIN anyway ?



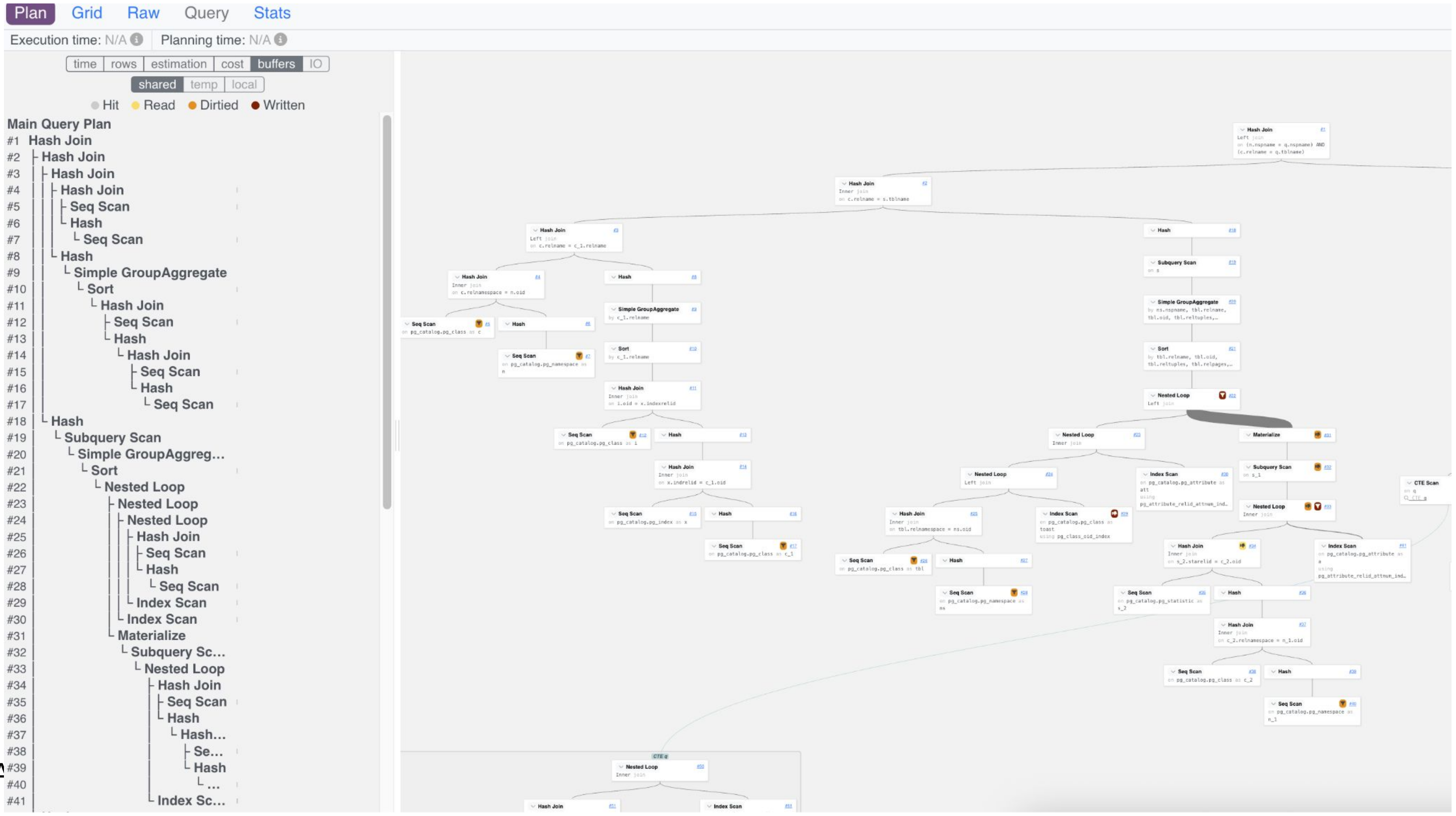
```
EXPLAIN (FORMAT YAML) SELECT * FROM foo WHERE i='4';
          QUERY PLAN
-------------------------------
 - Plan:                      +
     Node Type: "Index Scan"  +
     Scan Direction: "Forward"+
     Index Name: "fi"         +
     Relation Name: "foo"     +
     Alias: "foo"             +
     Startup Cost: 0.00       +
     Total Cost: 5.98         +
     Plan Rows: 1             +
     Plan Width: 4            +
     Index Cond: "(i = 4)"
(1 row)
```

What is EXPLAIN anyway ?



```
PREPARE query(int, int) AS SELECT sum(bar) FROM test
    WHERE id > $1 AND id < $2
    GROUP BY foo;

EXPLAIN ANALYZE EXECUTE query(100, 200);

                                                       QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------
 HashAggregate  (cost=10.77..10.87 rows=10 width=12) (actual time=0.043..0.044 rows=10 loops=1)
   Group Key: foo
   Batches: 1  Memory Usage: 24kB
   ->  Index Scan using test_pkey on test  (cost=0.29..10.27 rows=99 width=8) (actual time=0.009..0.025 rows=99 loops=1)
         Index Cond: ((id > 100) AND (id < 200))
 Planning Time: 0.244 ms
 Execution Time: 0.073 ms
(7 rows)
```

The online EXPLAINer game



Plan Grid Raw Query Stats

Execution time: N/A   Planning time: N/A

time | rows | estimation | cost | buffers | IO

shared | temp | local

● Hit ● Read ● Dirtied ● Written

```
Main Query Plan
#1  Hash Join
#2  ├ Hash Join
#3  │ ├ Hash Join
#4  │ │ ├ Hash Join
#5  │ │ │ ├ Seq Scan
#6  │ │ │ └ Hash
#7  │ │ │   └ Seq Scan
#8  │ │ └ Hash
#9  │ │   └ Simple GroupAggregate
#10 │ │     └ Sort
#11 │ │       └ Hash Join
#12 │ │         ├ Seq Scan
#13 │ │         └ Hash
#14 │ │           └ Hash Join
#15 │ │             ├ Seq Scan
#16 │ │             └ Hash
#17 │ │               └ Seq Scan
#18 │ └ Hash
#19 │   └ Subquery Scan
#20 │     └ Simple GroupAggreg...
#21 │       └ Sort
#22 │         └ Nested Loop
#23 │           ├ Nested Loop
#24 │           │ ├ Nested Loop
#25 │           │ │ ├ Hash Join
#26 │           │ │ │ ├ Seq Scan
#27 │           │ │ │ └ Hash
#28 │           │ │ │   └ Seq Scan
#29 │           │ │ └ Index Scan
#30 │           │ └ Index Scan
#31 │           └ Materialize
#32 │             └ Subquery Sc...
#33 │               └ Nested Loop
#34 │                 ├ Hash Join
#35 │                 │ ├ Seq Scan
#36 │                 │ └ Hash
#37 │                 │   └ Hash...
#38 │                 │     ├ Se...
#39 │                 │     └ Hash
#40 │                 │       └ ...
#41 │                 └ Index Sc...
```

The online EXPLAINer game

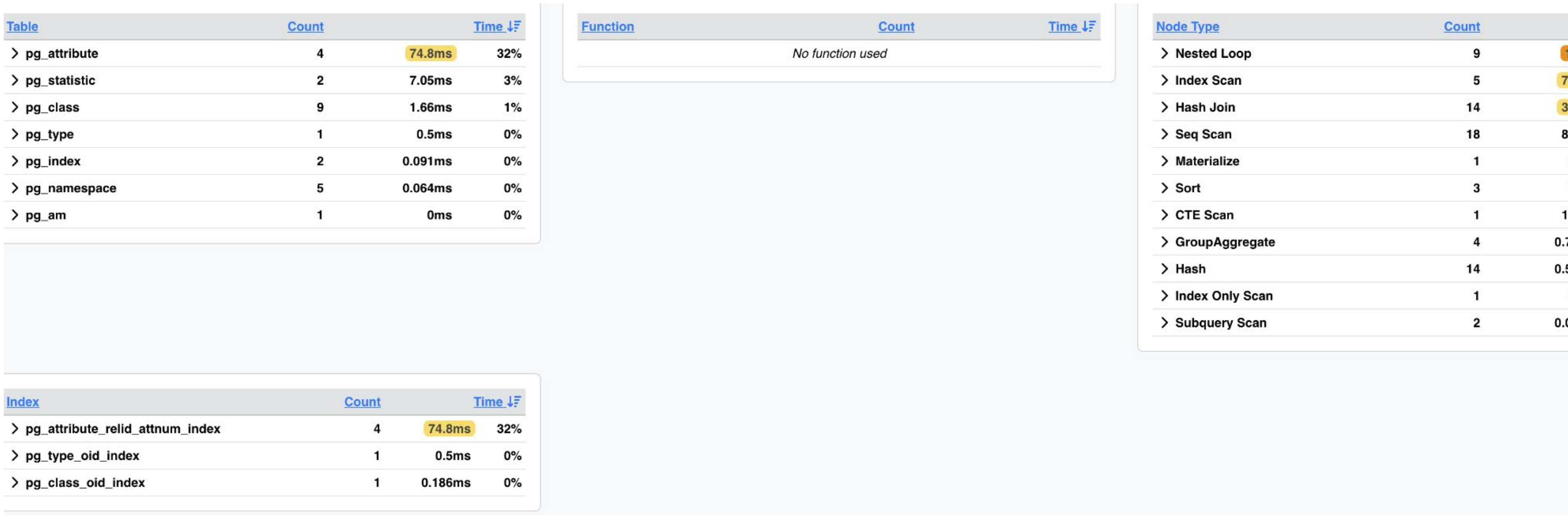


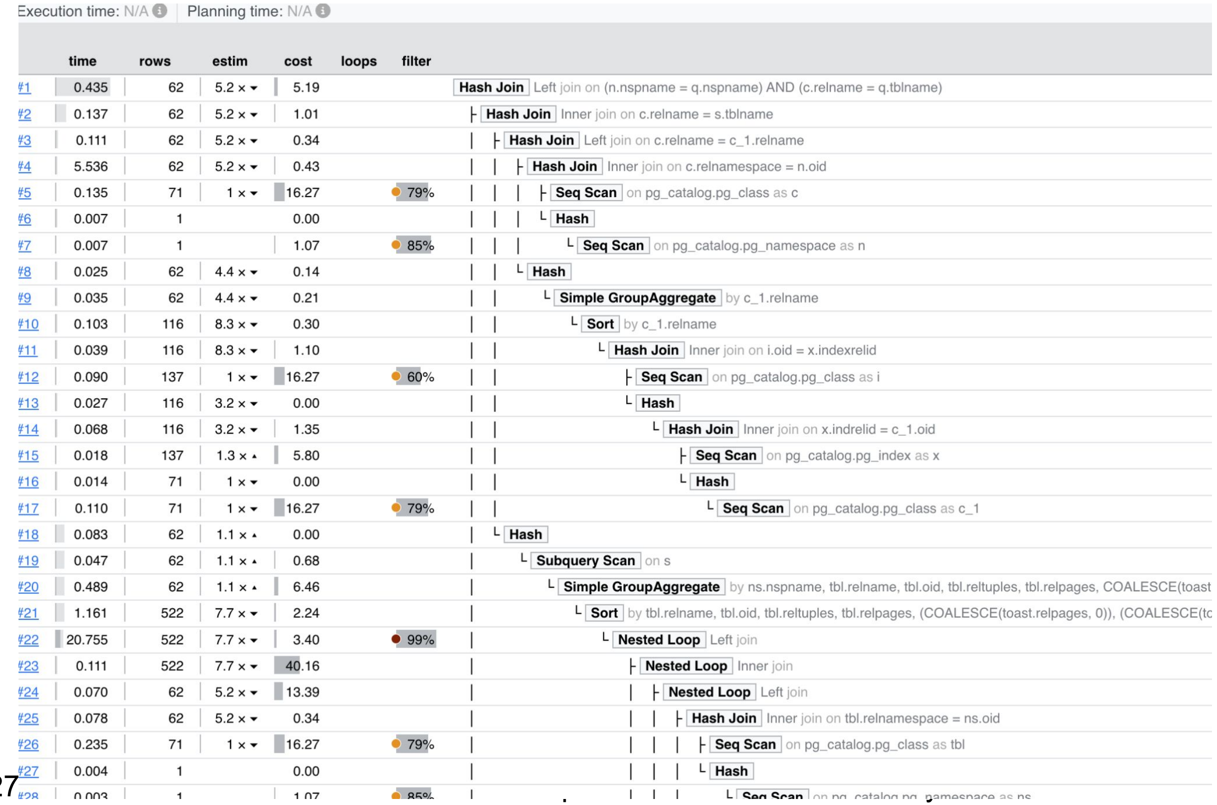
Table	Count	Time	
pg_attribute	4	74.8ms	32%
pg_statistic	2	7.05ms	3%
pg_class	9	1.66ms	1%
pg_type	1	0.5ms	0%
pg_index	2	0.091ms	0%
pg_namespace	5	0.064ms	0%
pg_am	1	0ms	0%

Function	Count	Time
No function used		

Node Type	Count	
Nested Loop	9	1
Index Scan	5	7
Hash Join	14	3
Seq Scan	18	8
Materialize	1	
Sort	3	
CTE Scan	1	1
GroupAggregate	4	0.7
Hash	14	0.5
Index Only Scan	1	
Subquery Scan	2	0.0

Index	Count	Time	
pg_attribute_relid_attnum_index	4	74.8ms	32%
pg_type_oid_index	1	0.5ms	0%
pg_class_oid_index	1	0.186ms	0%

The online EXPLAINer game



Execution time: N/A | Planning time: N/A

	time	rows	estim	cost	loops	filter	
#1	0.435	62	5.2 × ▾	5.19			Hash Join Left join on (n.nspname = q.nspname) AND (c.relname = q.tblname)
#2	0.137	62	5.2 × ▾	1.01			├ Hash Join Inner join on c.relname = s.tblname
#3	0.111	62	5.2 × ▾	0.34			│ ├ Hash Join Left join on c.relname = c_1.relname
#4	5.536	62	5.2 × ▾	0.43			│ │ ├ Hash Join Inner join on c.relnamespace = n.oid
#5	0.135	71	1 × ▾	16.27		● 79%	│ │ │ ├ Seq Scan on pg_catalog.pg_class as c
#6	0.007	1		0.00			│ │ │ └ Hash
#7	0.007	1		1.07		● 85%	│ │ │ └ Seq Scan on pg_catalog.pg_namespace as n
#8	0.025	62	4.4 × ▾	0.14			│ │ └ Hash
#9	0.035	62	4.4 × ▾	0.21			│ │ └ Simple GroupAggregate by c_1.relname
#10	0.103	116	8.3 × ▾	0.30			│ │ └ Sort by c_1.relname
#11	0.039	116	8.3 × ▾	1.10			│ │ └ Hash Join Inner join on i.oid = x.indexrelid
#12	0.090	137	1 × ▾	16.27		● 60%	│ │ ├ Seq Scan on pg_catalog.pg_class as i
#13	0.027	116	3.2 × ▾	0.00			│ │ └ Hash
#14	0.068	116	3.2 × ▾	1.35			│ │ └ Hash Join Inner join on x.indrelid = c_1.oid
#15	0.018	137	1.3 × ▴	5.80			│ │ ├ Seq Scan on pg_catalog.pg_index as x
#16	0.014	71	1 × ▾	0.00			│ │ └ Hash
#17	0.110	71	1 × ▾	16.27		● 79%	│ │ └ Seq Scan on pg_catalog.pg_class as c_1
#18	0.083	62	1.1 × ▴	0.00			│ └ Hash
#19	0.047	62	1.1 × ▴	0.68			│ └ Subquery Scan on s
#20	0.489	62	1.1 × ▴	6.46			│ └ Simple GroupAggregate by ns.nspname, tbl.relname, tbl.oid, tbl.reltuples, tbl.relpages, COALESCE(toast
#21	1.161	522	7.7 × ▾	2.24			│ └ Sort by tbl.relname, tbl.oid, tbl.reltuples, tbl.relpages, (COALESCE(toast.relpages, 0)), (COALESCE(tc
#22	20.755	522	7.7 × ▾	3.40		● 99%	│ └ Nested Loop Left join
#23	0.111	522	7.7 × ▾	40.16			│ ├ Nested Loop Inner join
#24	0.070	62	5.2 × ▾	13.39			│ │ ├ Nested Loop Left join
#25	0.078	62	5.2 × ▾	0.34			│ │ │ ├ Hash Join Inner join on tbl.relnamespace = ns.oid
#26	0.235	71	1 × ▾	16.27		● 79%	│ │ │ │ ├ Seq Scan on pg_catalog.pg_class as tbl
#27	0.004	1		0.00			│ │ │ │ └ Hash
#28	0.003	1		1.07		● 85%	│ │ │ │ └ Seq Scan on pg_catalog.pg_namespace as ns

The online EXPLAINer game

explain.dalibo.com PostgreSQL execution plan visualizer

Visualizing and understanding PostgreSQL EXPLAIN plans made easy.

The online EXPLAINer game

explain.depesz.com
PostgreSQL's explain analyze made readable

explain.dalibo.com PostgreSQL execution plan visualizer

Visualizing and understanding PostgreSQL EXPLAIN plans made easy.

The online EXPLAINer game

explain.depesz.com

PostgreSQL's explain analyze made readable

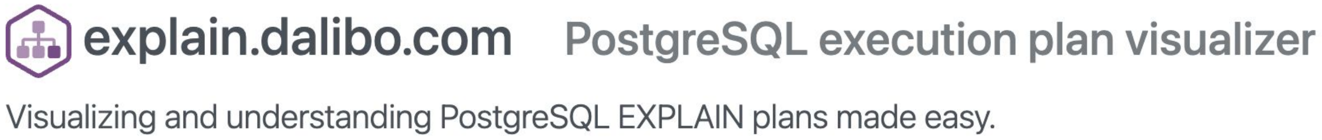
explain.dalibo.com PostgreSQL execution plan visualizer

Visualizing and understanding PostgreSQL EXPLAIN plans made easy.

Postgres EXPLAIN Visualizer (Pev)

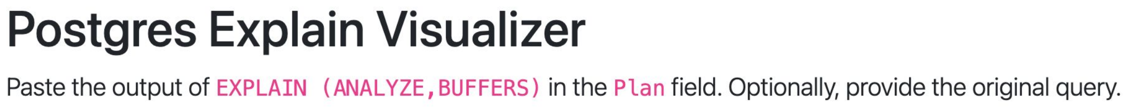
The online EXPLAINer game

explain.depesz.com
PostgreSQL's explain analyze made readable


explain.dalibo.com PostgreSQL execution plan visualizer

Visualizing and understanding PostgreSQL EXPLAIN plans made easy.

Postgres EXPLAIN Visualizer (Pev)


Postgres Explain Visualizer

Paste the output of `EXPLAIN (ANALYZE,BUFFERS)` in the `Plan` field. Optionally, provide the original query.

The online EXPLAINer game

pesz.com

plain analyze made readable

explain.dalibo.com Po alizer

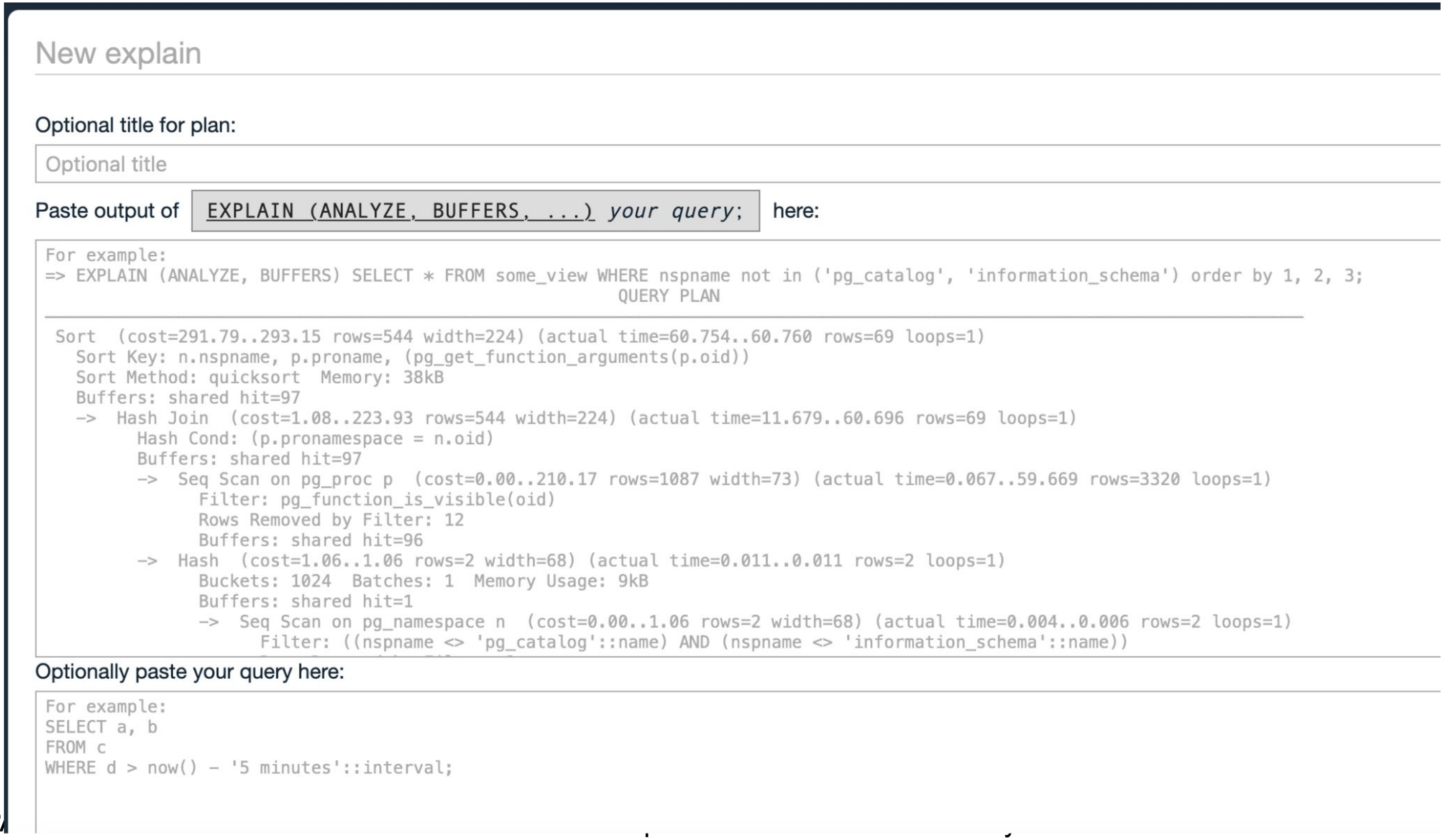
Visualizing and understanding PostgreSQL

Postgres ualizer (Pev)

Postg

Paste the output of `EXPLAIN (ANALYZE,BUFFERS)` in the `Plan` field. Optionally, provide the original query.

It’s a paste



New explain

Optional title for plan:

Optional title

Paste output of `EXPLAIN (ANALYZE, BUFFERS, ...) your query;` here:

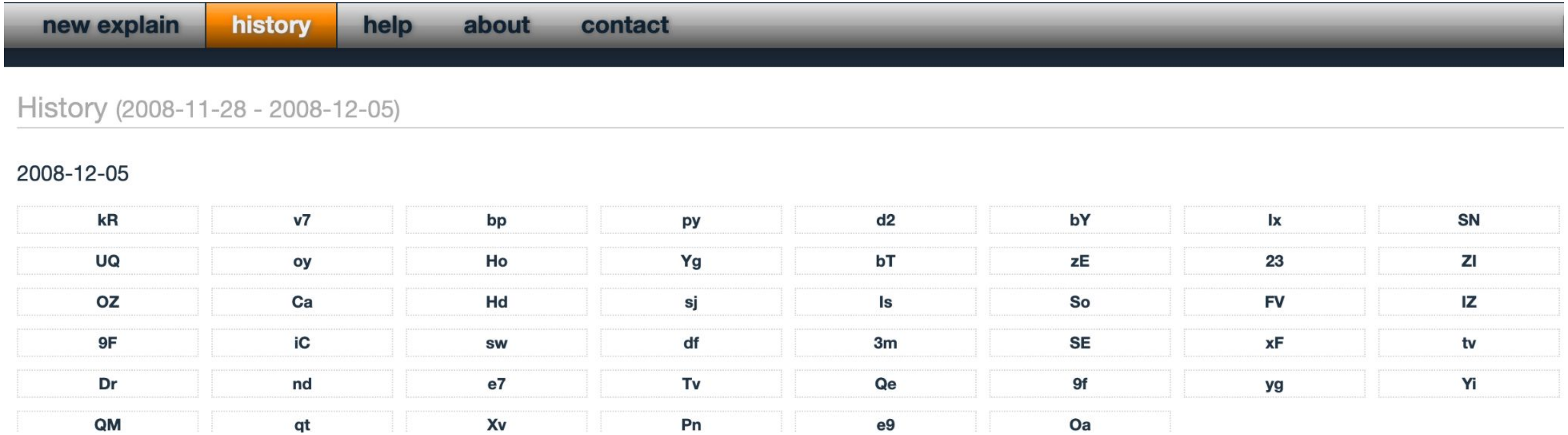
```
For example:
=> EXPLAIN (ANALYZE, BUFFERS) SELECT * FROM some_view WHERE nspname not in ('pg_catalog', 'information_schema') order by 1, 2, 3;
                                                        QUERY PLAN
─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
 Sort  (cost=291.79..293.15 rows=544 width=224) (actual time=60.754..60.760 rows=69 loops=1)
   Sort Key: n.nspname, p.proname, (pg_get_function_arguments(p.oid))
   Sort Method: quicksort  Memory: 38kB
   Buffers: shared hit=97
   ->  Hash Join  (cost=1.08..223.93 rows=544 width=224) (actual time=11.679..60.696 rows=69 loops=1)
         Hash Cond: (p.pronamespace = n.oid)
         Buffers: shared hit=97
         ->  Seq Scan on pg_proc p  (cost=0.00..210.17 rows=1087 width=73) (actual time=0.067..59.669 rows=3320 loops=1)
               Filter: pg_function_is_visible(oid)
               Rows Removed by Filter: 12
               Buffers: shared hit=96
         ->  Hash  (cost=1.06..1.06 rows=2 width=68) (actual time=0.011..0.011 rows=2 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               Buffers: shared hit=1
               ->  Seq Scan on pg_namespace n  (cost=0.00..1.06 rows=2 width=68) (actual time=0.004..0.006 rows=2 loops=1)
                     Filter: ((nspname <> 'pg_catalog'::name) AND (nspname <> 'information_schema'::name))
```

Optionally paste your query here:

```
For example:
SELECT a, b
FROM c
WHERE d > now() - '5 minutes'::interval;
```

It’s a paste, with history 🤔

It’s a paste, with history 🤔 and guessable IDs 🤔🤔

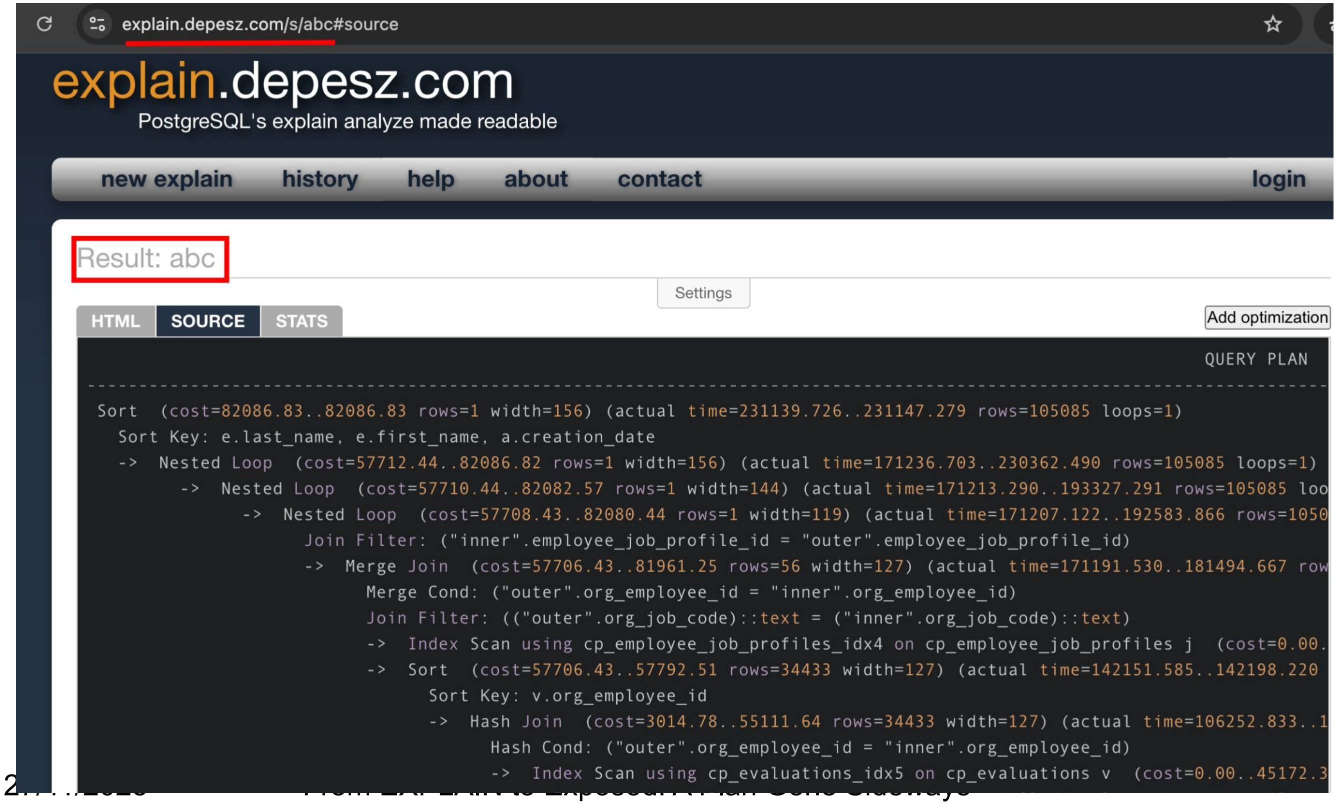


History (2008-11-28 - 2008-12-05)

2008-12-05

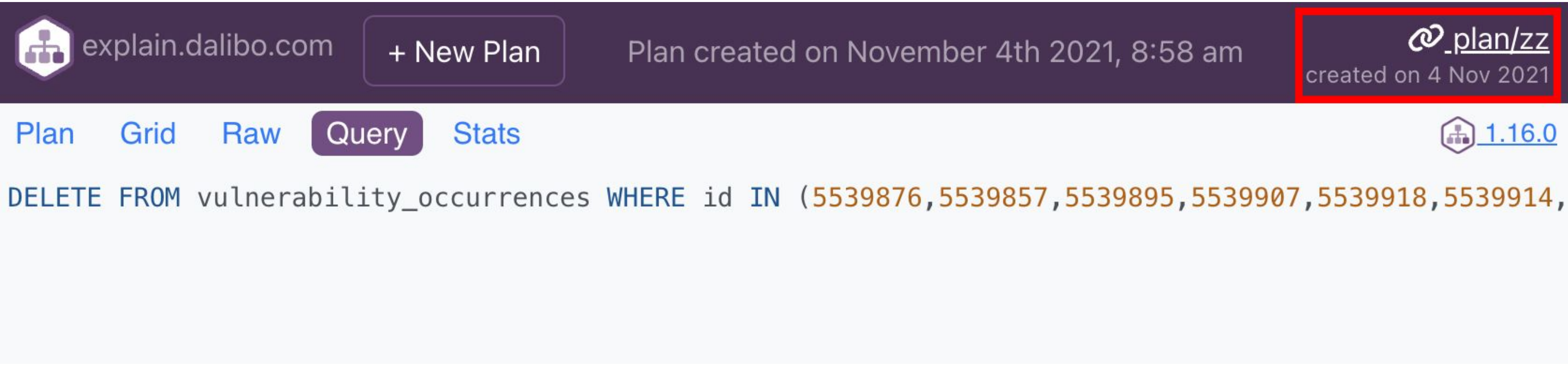
kR	v7	bp	py	d2	bY	Ix	SN
UQ	oy	Ho	Yg	bT	zE	23	ZI
OZ	Ca	Hd	sj	Is	So	FV	IZ
9F	iC	sw	df	3m	SE	xF	tv
Dr	nd	e7	Tv	Qe	9f	yg	Yi
QM	qt	Xv	Pn	e9	Oa		

It’s a paste, with history 🤔 and guessable IDs 🤔🤔



```
                                                                                        QUERY PLAN
----------------------------------------------------------------------------------------------------------------------------
 Sort  (cost=82086.83..82086.83 rows=1 width=156) (actual time=231139.726..231147.279 rows=105085 loops=1)
   Sort Key: e.last_name, e.first_name, a.creation_date
   ->  Nested Loop  (cost=57712.44..82086.82 rows=1 width=156) (actual time=171236.703..230362.490 rows=105085 loops=1)
         ->  Nested Loop  (cost=57710.44..82082.57 rows=1 width=144) (actual time=171213.290..193327.291 rows=105085 loo
               ->  Nested Loop  (cost=57708.43..82080.44 rows=1 width=119) (actual time=171207.122..192583.866 rows=1050
                     Join Filter: ("inner".employee_job_profile_id = "outer".employee_job_profile_id)
                     ->  Merge Join  (cost=57706.43..81961.25 rows=56 width=127) (actual time=171191.530..181494.667 row
                           Merge Cond: ("outer".org_employee_id = "inner".org_employee_id)
                           Join Filter: (("outer".org_job_code)::text = ("inner".org_job_code)::text)
                           ->  Index Scan using cp_employee_job_profiles_idx4 on cp_employee_job_profiles j  (cost=0.00.
                           ->  Sort  (cost=57706.43..57792.51 rows=34433 width=127) (actual time=142151.585..142198.220
                                 Sort Key: v.org_employee_id
                                 ->  Hash Join  (cost=3014.78..55111.64 rows=34433 width=127) (actual time=106252.833..1
                                       Hash Cond: ("outer".org_employee_id = "inner".org_employee_id)
                                       ->  Index Scan using cp_evaluations_idx5 on cp_evaluations v  (cost=0.00..45172.3
```

It’s a paste, with history 🤔 and guessable IDs 🤔🤔

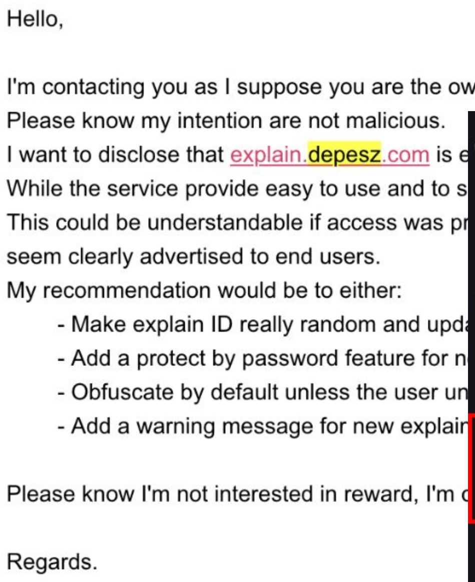


Plan created on November 4th 2021, 8:58 am

plan/zz
created on 4 Nov 2021

```
DELETE FROM vulnerability_occurrences WHERE id IN (5539876,5539857,5539895,5539907,5539918,5539914,
```

It’s a paste, with history 🤔 and guessable IDs 🤔🤔

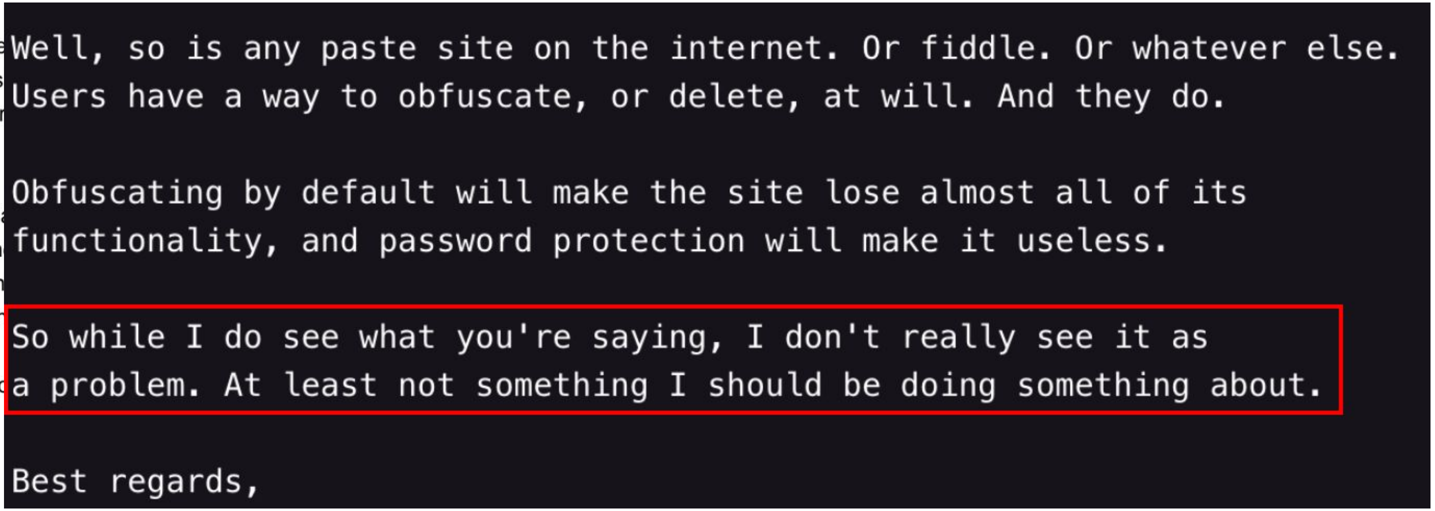


Hello,

I'm contacting you as I suppose you are the owner of https://explain.depesz.com/.
Please know my intention are not malicious.
I want to disclose that explain.depesz.com is e
While the service provide easy to use and to s
This could be understandable if access was pr
seem clearly advertised to end users.
My recommendation would be to either:
- Make explain ID really random and upda
- Add a protect by password feature for n
- Obfuscate by default unless the user un
- Add a warning message for new explain

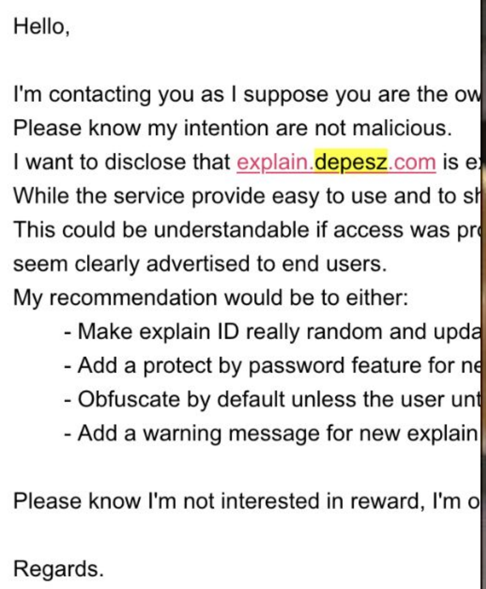
Please know I'm not interested in reward, I'm c

Regards.



```
Well, so is any paste site on the internet. Or fiddle. Or whatever else.
Users have a way to obfuscate, or delete, at will. And they do.

Obfuscating by default will make the site lose almost all of its
functionality, and password protection will make it useless.

So while I do see what you're saying, I don't really see it as
a problem. At least not something I should be doing something about.

Best regards,
```

It’s a paste, with history 🤔 and guessable IDs 🤔🤔



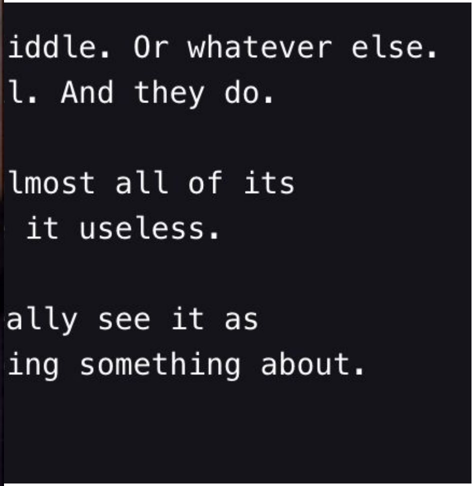
Hello,

I'm contacting you as I suppose you are the ow
Please know my intention are not malicious.
I want to disclose that explain.depesz.com is e
While the service provide easy to use and to sh
This could be understandable if access was pr
seem clearly advertised to end users.
My recommendation would be to either:

- Make explain ID really random and upda
- Add a protect by password feature for ne
- Obfuscate by default unless the user unt
- Add a warning message for new explain

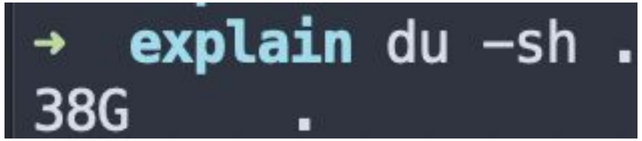
Please know I'm not interested in reward, I'm o

Regards.



```
iddle. Or whatever else.
l. And they do.

lmost all of its
 it useless.

ally see it as
ing something about.
```

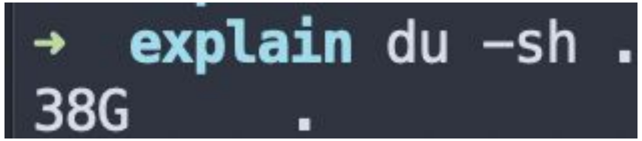
It’s a paste, with history 🤔 and guessable IDs 🤔🤔



```
→  explain du -sh .
38G     .
```

It’s a paste, with history 🤔 and guessable IDs 🤔🤔

Semgrep



```
→  explain du -sh .
38G     .
```

It’s a paste, with history 🤔 and guessable IDs 🤔🤔

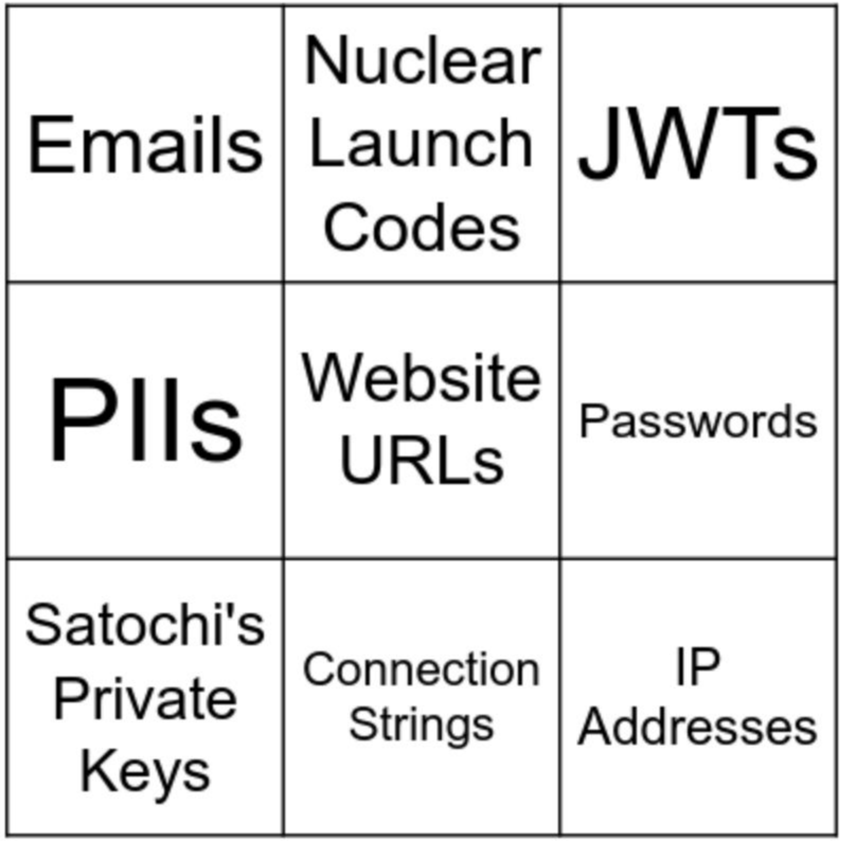
Semgrep

```
→  explain du -sh .
38G     .
```

It’s a paste, with history 🤔 and guessable IDs 🤔🤔

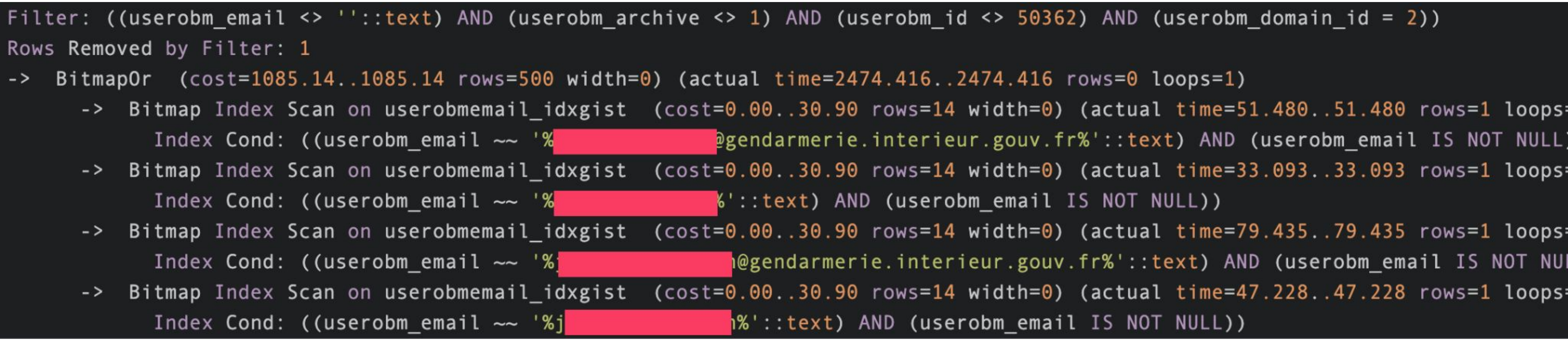
It’s a paste, with history 🤔 and guessable IDs 🤔🤔

Bingo Card



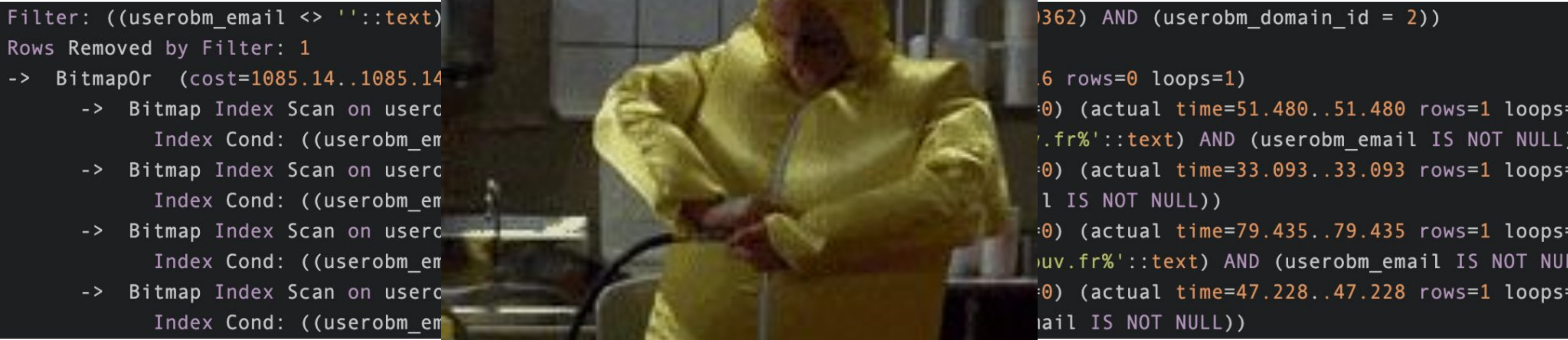
Emails	Nuclear Launch Codes	JWTs
PIIs	Website URLs	Passwords
Satochi's Private Keys	Connection Strings	IP Addresses

Emails / PII



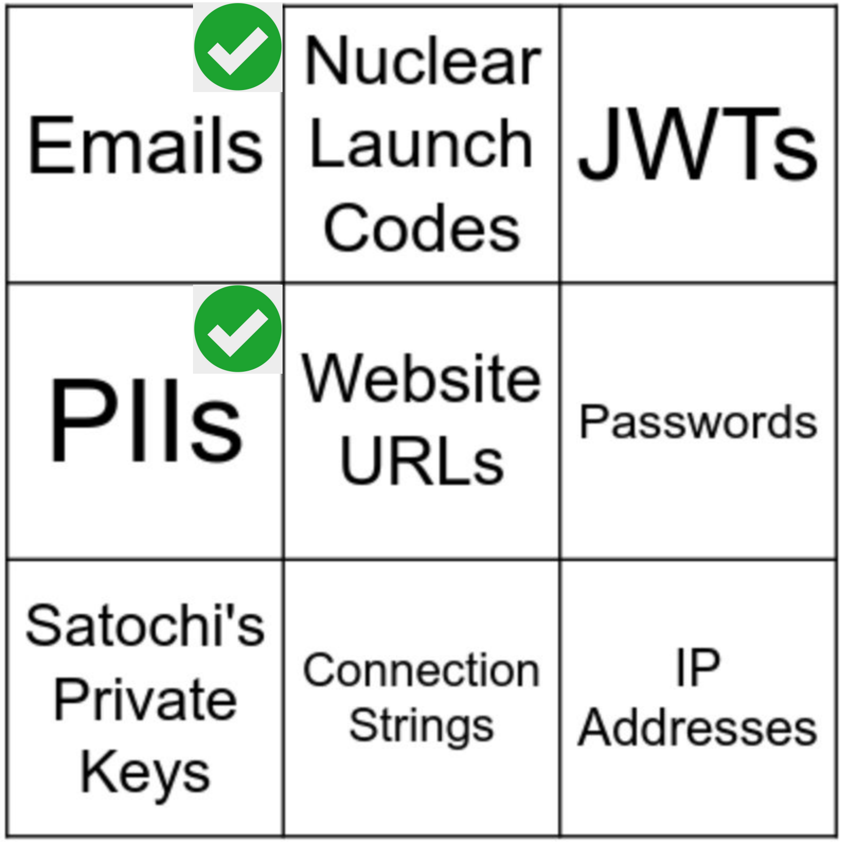
```
Filter: ((userobm_email <> ''::text) AND (userobm_archive <> 1) AND (userobm_id <> 50362) AND (userobm_domain_id = 2))
Rows Removed by Filter: 1
->  BitmapOr  (cost=1085.14..1085.14 rows=500 width=0) (actual time=2474.416..2474.416 rows=0 loops=1)
      ->  Bitmap Index Scan on userobmemail_idxgist  (cost=0.00..30.90 rows=14 width=0) (actual time=51.480..51.480 rows=1 loops
            Index Cond: ((userobm_email ~~ '%[redacted]@gendarmerie.interieur.gouv.fr%'::text) AND (userobm_email IS NOT NULL
      ->  Bitmap Index Scan on userobmemail_idxgist  (cost=0.00..30.90 rows=14 width=0) (actual time=33.093..33.093 rows=1 loops
            Index Cond: ((userobm_email ~~ '%[redacted]%'::text) AND (userobm_email IS NOT NULL))
      ->  Bitmap Index Scan on userobmemail_idxgist  (cost=0.00..30.90 rows=14 width=0) (actual time=79.435..79.435 rows=1 loops
            Index Cond: ((userobm_email ~~ '%[redacted]@gendarmerie.interieur.gouv.fr%'::text) AND (userobm_email IS NOT NU
      ->  Bitmap Index Scan on userobmemail_idxgist  (cost=0.00..30.90 rows=14 width=0) (actual time=47.228..47.228 rows=1 loops
            Index Cond: ((userobm_email ~~ '%j[redacted]%'::text) AND (userobm_email IS NOT NULL))
```

Emails / PII



```
Filter: ((userobm_email <> ''::text)                      362) AND (userobm_domain_id = 2))
Rows Removed by Filter: 1
->  BitmapOr  (cost=1085.14..1085.14                      6 rows=0 loops=1)
      ->  Bitmap Index Scan on userc                      0) (actual time=51.480..51.480 rows=1 loops
            Index Cond: ((userobm_em                      .fr%'::text) AND (userobm_email IS NOT NULL
      ->  Bitmap Index Scan on userc                      0) (actual time=33.093..33.093 rows=1 loops
            Index Cond: ((userobm_em                      l IS NOT NULL))
      ->  Bitmap Index Scan on userc                      0) (actual time=79.435..79.435 rows=1 loops
            Index Cond: ((userobm_em                      uv.fr%'::text) AND (userobm_email IS NOT NU
      ->  Bitmap Index Scan on userc                      0) (actual time=47.228..47.228 rows=1 loops
            Index Cond: ((userobm_em                      ail IS NOT NULL))
```

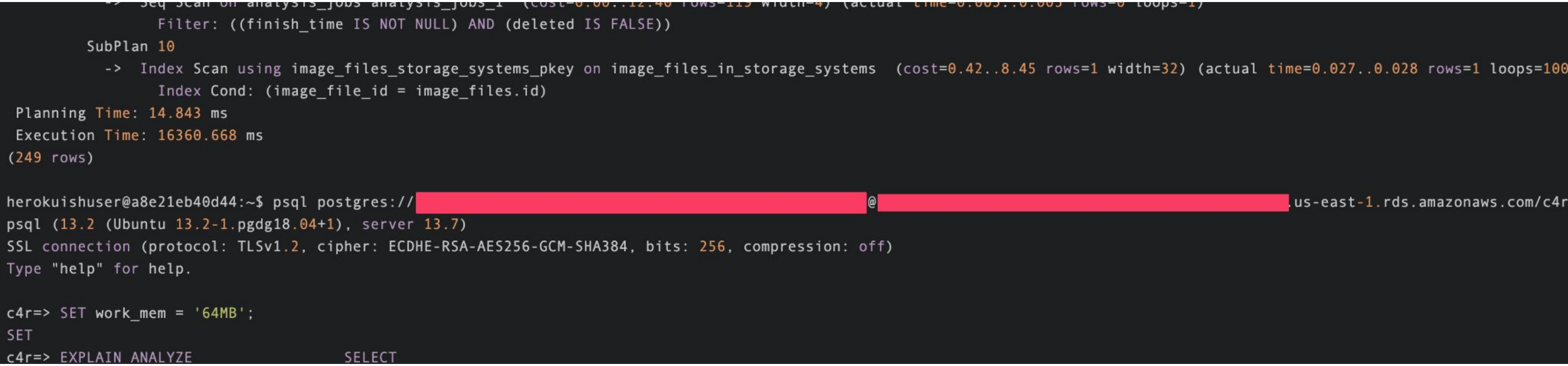
Emails / PII



Emails ✓	Nuclear Launch Codes	JWTs
PIIs ✓	Website URLs	Passwords
Satochi's Private Keys	Connection Strings	IP Addresses

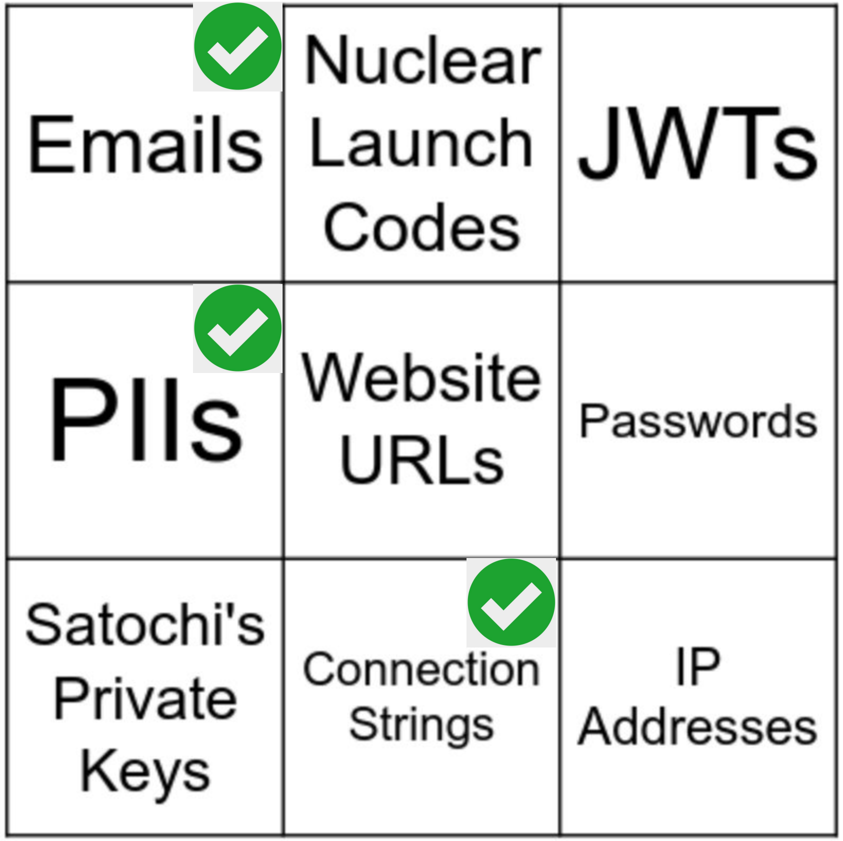
ConnectionString

Oups I pasted more than explain



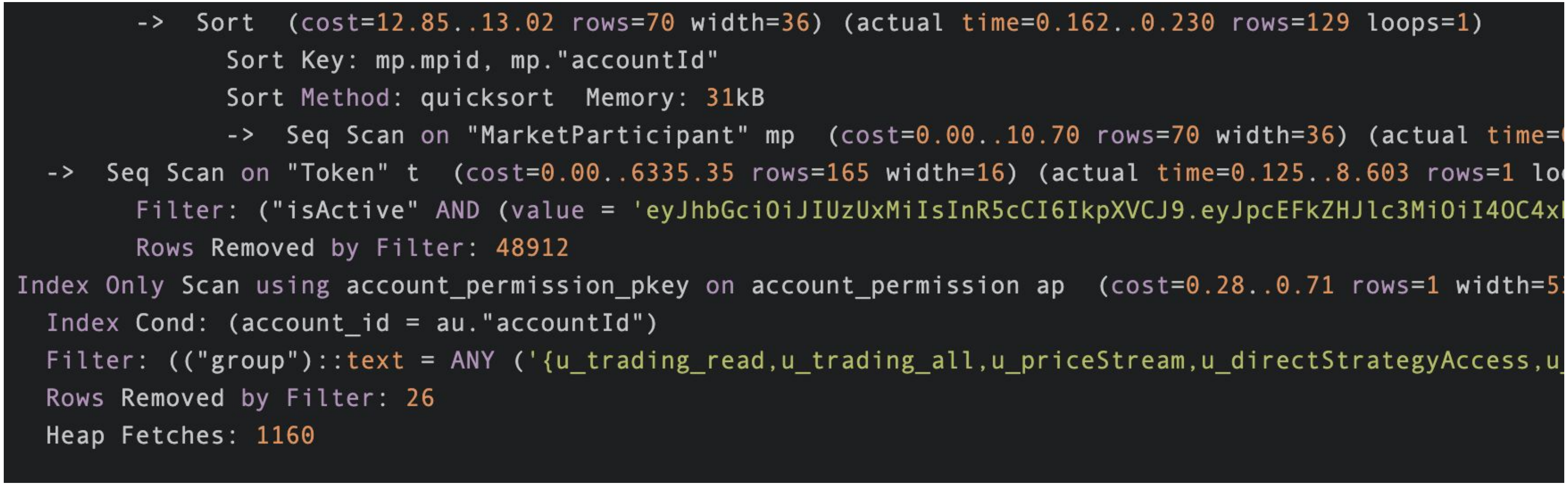
```
          ->  Seq Scan on analysis_jobs analysis_jobs_1  (cost=0.00..12.40 rows=119 width=4) (actual time=0.005..0.005 rows=0 loops=1)
                Filter: ((finish_time IS NOT NULL) AND (deleted IS FALSE))
        SubPlan 10
          ->  Index Scan using image_files_storage_systems_pkey on image_files_in_storage_systems  (cost=0.42..8.45 rows=1 width=32) (actual time=0.027..0.028 rows=1 loops=100
                Index Cond: (image_file_id = image_files.id)
 Planning Time: 14.843 ms
 Execution Time: 16360.668 ms
(249 rows)

herokuishuser@a8e21eb40d44:~$ psql postgres://[redacted]@[redacted].us-east-1.rds.amazonaws.com/c4r
psql (13.2 (Ubuntu 13.2-1.pgdg18.04+1), server 13.7)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

c4r=> SET work_mem = '64MB';
SET
c4r=> EXPLAIN ANALYZE                   SELECT
```

ConnectionString



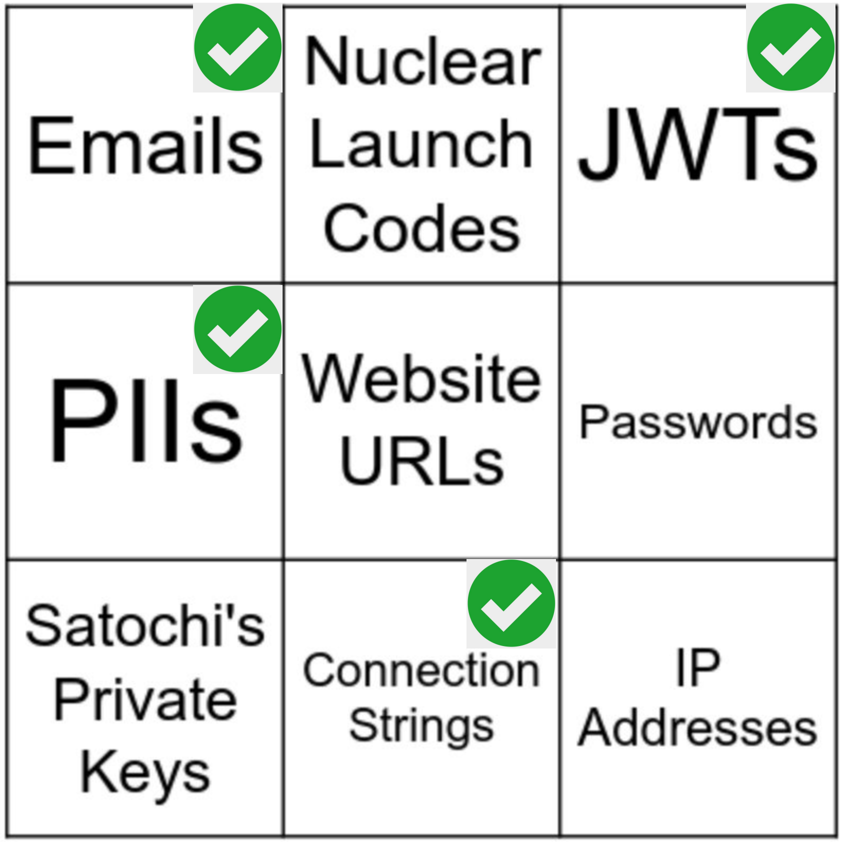
Emails ✓	Nuclear Launch Codes	JWTs
PIIs ✓	Website URLs	Passwords
Satochi's Private Keys	Connection Strings ✓	IP Addresses

JWT



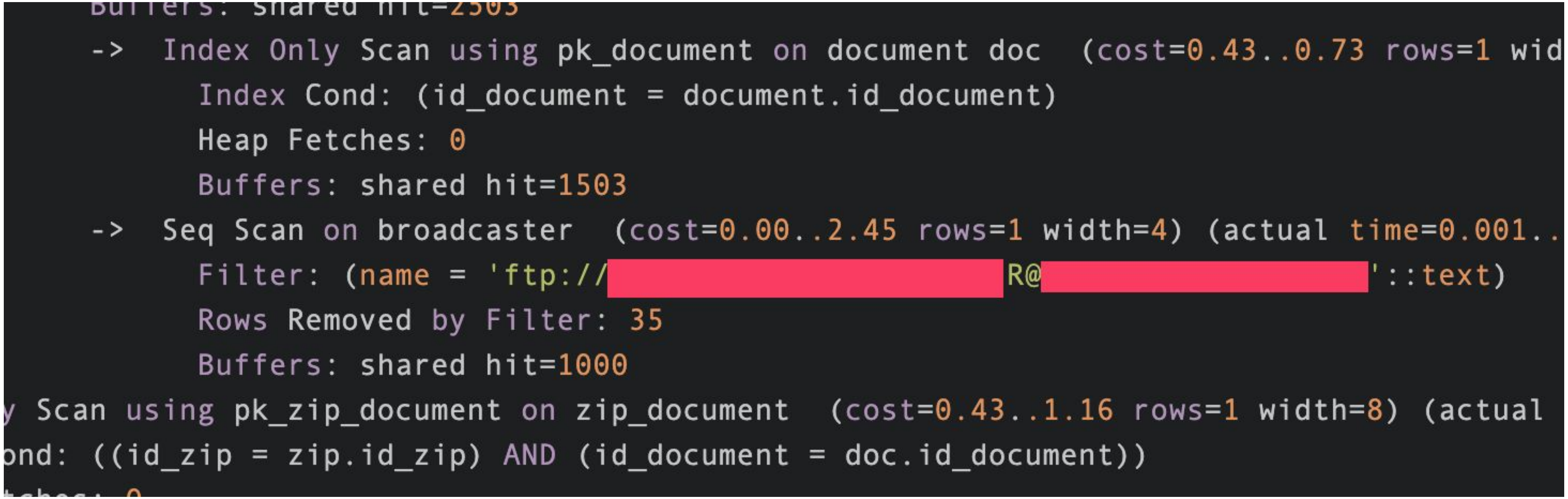
```
    ->  Sort  (cost=12.85..13.02 rows=70 width=36) (actual time=0.162..0.230 rows=129 loops=1)
          Sort Key: mp.mpid, mp."accountId"
          Sort Method: quicksort  Memory: 31kB
          ->  Seq Scan on "MarketParticipant" mp  (cost=0.00..10.70 rows=70 width=36) (actual time=
 ->  Seq Scan on "Token" t  (cost=0.00..6335.35 rows=165 width=16) (actual time=0.125..8.603 rows=1 lo
    Filter: ("isActive" AND (value = 'eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJpcEFkZHJlc3MiOiI4OC4x
    Rows Removed by Filter: 48912
Index Only Scan using account_permission_pkey on account_permission ap  (cost=0.28..0.71 rows=1 width=5
 Index Cond: (account_id = au."accountId")
 Filter: (("group")::text = ANY ('{u_trading_read,u_trading_all,u_priceStream,u_directStrategyAccess,u_
 Rows Removed by Filter: 26
 Heap Fetches: 1160
```

JWT



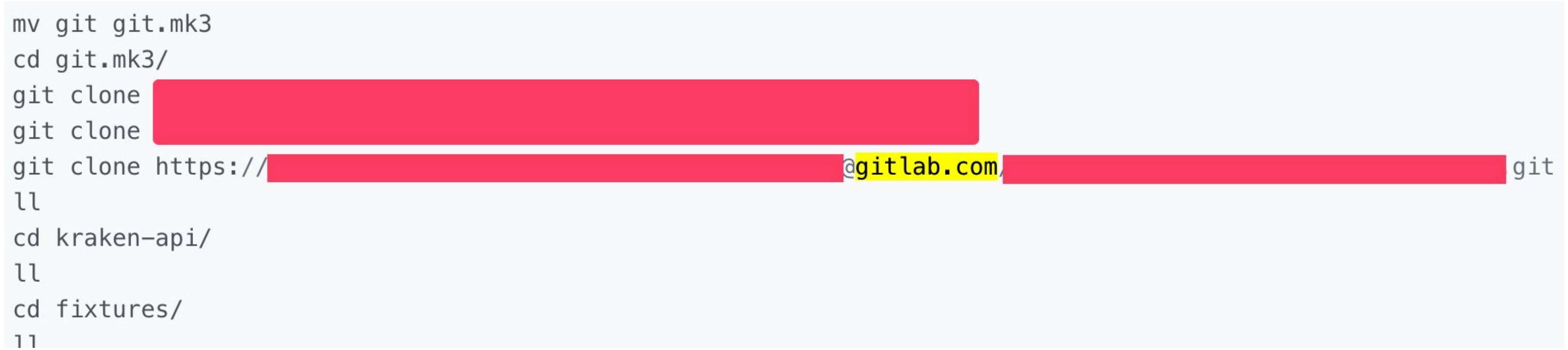
Emails ✓	Nuclear Launch Codes	JWTs ✓
PIIs ✓	Website URLs	Passwords
Satochi's Private Keys	Connection Strings ✓	IP Addresses

Website URLs / Passwords



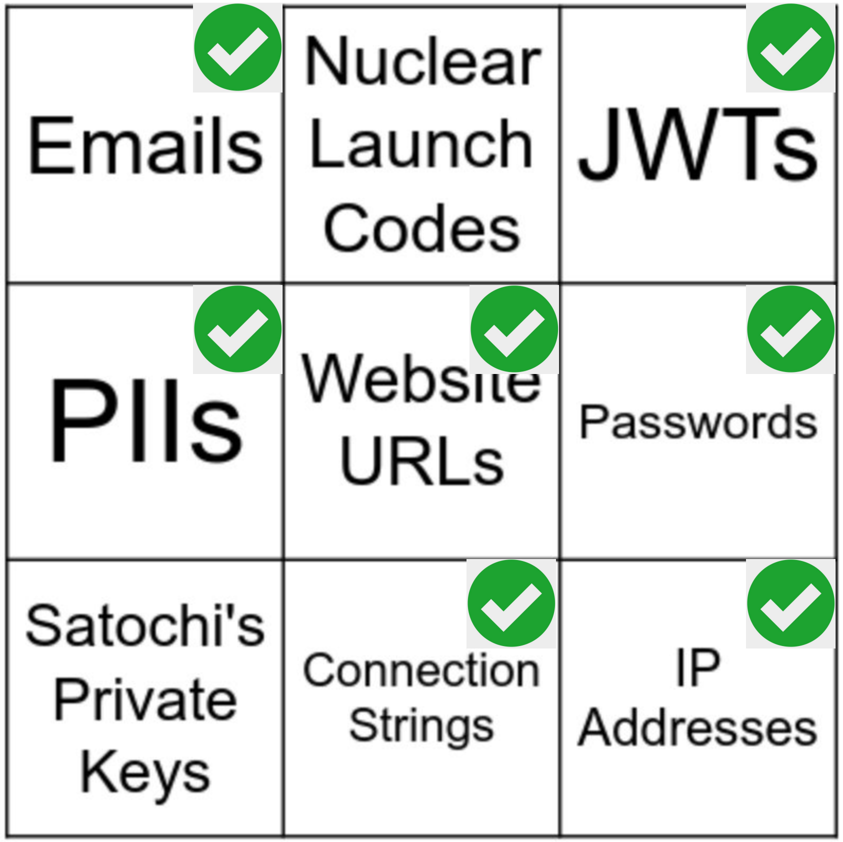
```
    Buffers: shared hit=2503
    ->  Index Only Scan using pk_document on document doc  (cost=0.43..0.73 rows=1 wid
          Index Cond: (id_document = document.id_document)
          Heap Fetches: 0
          Buffers: shared hit=1503
    ->  Seq Scan on broadcaster  (cost=0.00..2.45 rows=1 width=4) (actual time=0.001..
          Filter: (name = 'ftp://[redacted]R@[redacted]'::text)
          Rows Removed by Filter: 35
          Buffers: shared hit=1000
y Scan using pk_zip_document on zip_document  (cost=0.43..1.16 rows=1 width=8) (actual
ond: ((id_zip = zip.id_zip) AND (id_document = doc.id_document))
```

Website URLs / Passwords



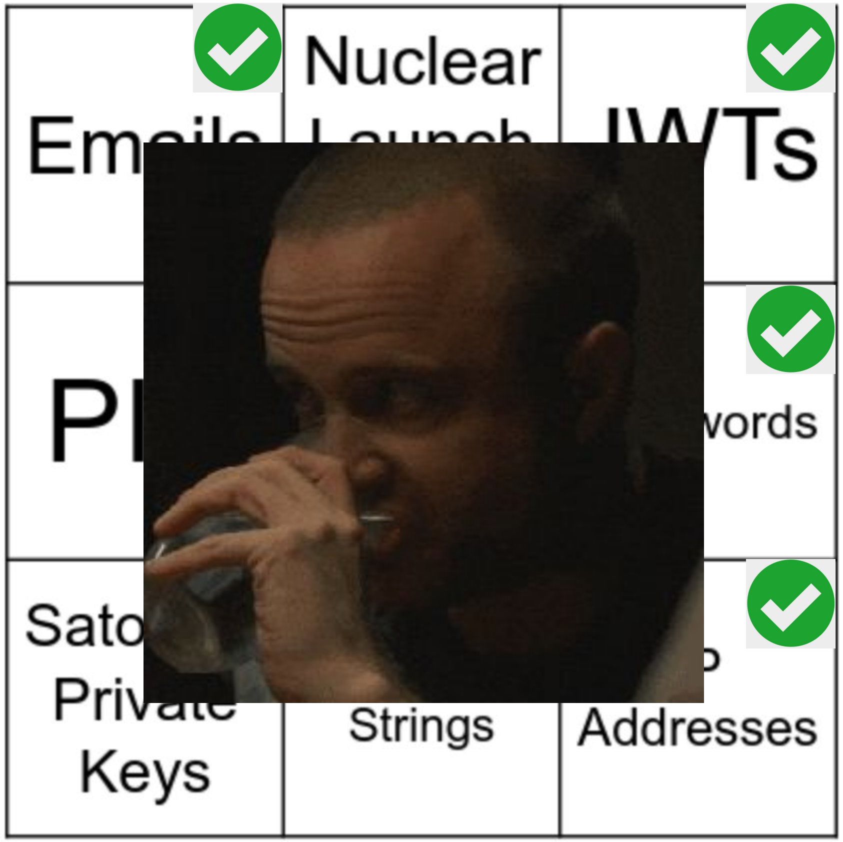
```
mv git git.mk3
cd git.mk3/
git clone [redacted]
git clone
git clone https://[redacted]@gitlab.com/[redacted] git
ll
cd kraken-api/
ll
cd fixtures/
ll
```

Website URLs / Passwords



Emails ✓	Nuclear Launch Codes	JWTs ✓
PIIs ✓	Website URLs ✓	Passwords ✓
Satochi's Private Keys	Connection Strings ✓	IP Addresses ✓

?

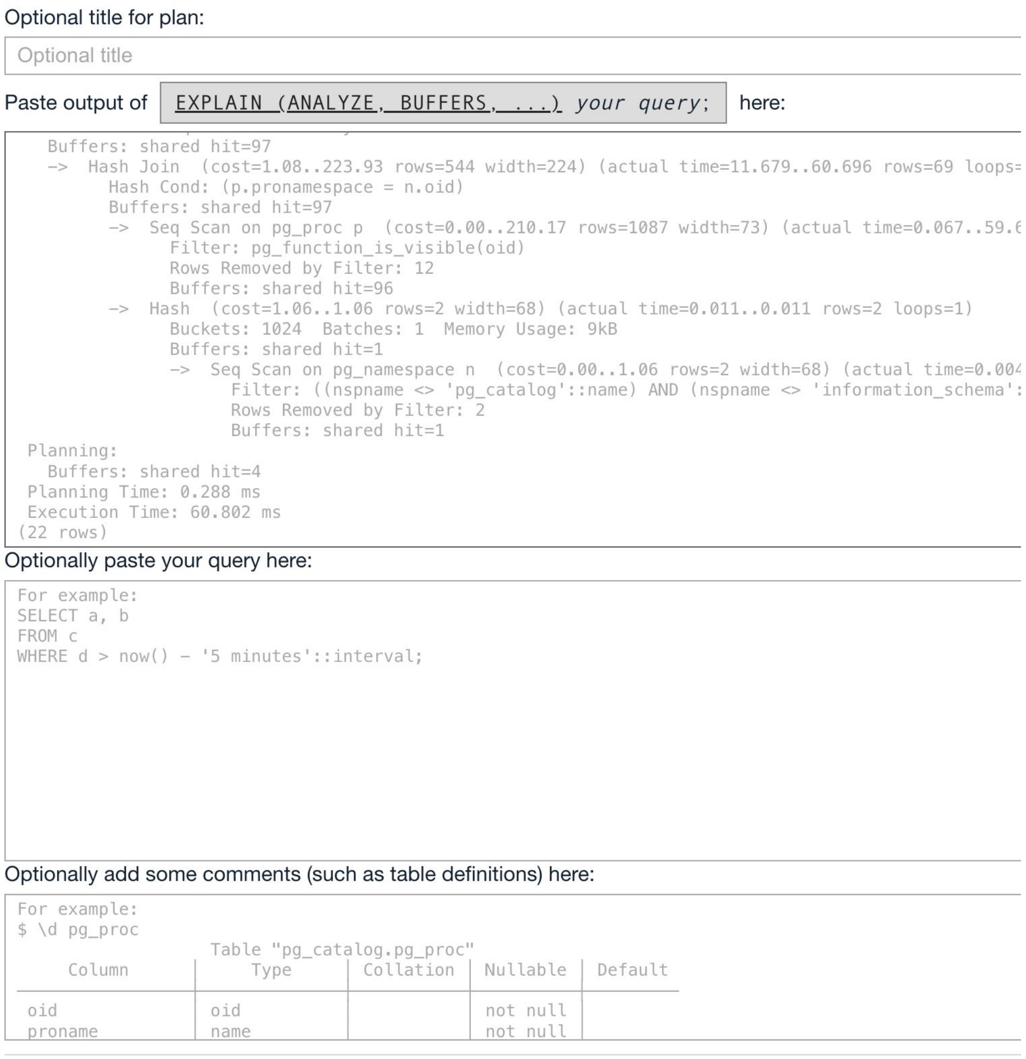

Em
Nuclear
JWTs
Pl
vords
Sato
Private
Keys
Strings
Addresses

depesz

explain.depesz.com

PostgreSQL's explain analyze made readable

depesz


Optional title for plan:

Optional title

Paste output of `EXPLAIN (ANALYZE, BUFFERS, ...) your query;` here:

```
   Buffers: shared hit=97
   ->  Hash Join  (cost=1.08..223.93 rows=544 width=224) (actual time=11.679..60.696 rows=69 loops=
         Hash Cond: (p.pronamespace = n.oid)
         Buffers: shared hit=97
         ->  Seq Scan on pg_proc p  (cost=0.00..210.17 rows=1087 width=73) (actual time=0.067..59.6
               Filter: pg_function_is_visible(oid)
               Rows Removed by Filter: 12
               Buffers: shared hit=96
         ->  Hash  (cost=1.06..1.06 rows=2 width=68) (actual time=0.011..0.011 rows=2 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               Buffers: shared hit=1
               ->  Seq Scan on pg_namespace n  (cost=0.00..1.06 rows=2 width=68) (actual time=0.004
                     Filter: ((nspname <> 'pg_catalog'::name) AND (nspname <> 'information_schema':
                     Rows Removed by Filter: 2
                     Buffers: shared hit=1
 Planning:
   Buffers: shared hit=4
 Planning Time: 0.288 ms
 Execution Time: 60.802 ms
(22 rows)
```

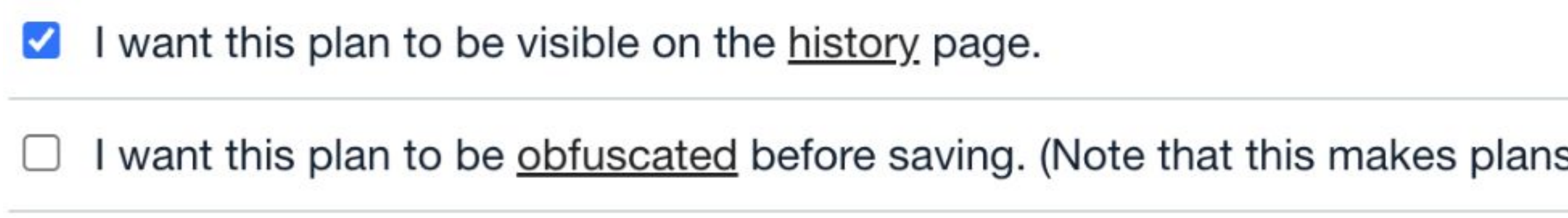
Optionally paste your query here:

```
For example:
SELECT a, b
FROM c
WHERE d > now() - '5 minutes'::interval;
```

Optionally add some comments (such as table definitions) here:

```
For example:
$ \d pg_proc
                   Table "pg_catalog.pg_proc"
     Column      |     Type     | Collation | Nullable | Default
-----------------+--------------+-----------+----------+---------
 oid             | oid          |           | not null |
 proname         | name         |           | not null |
```

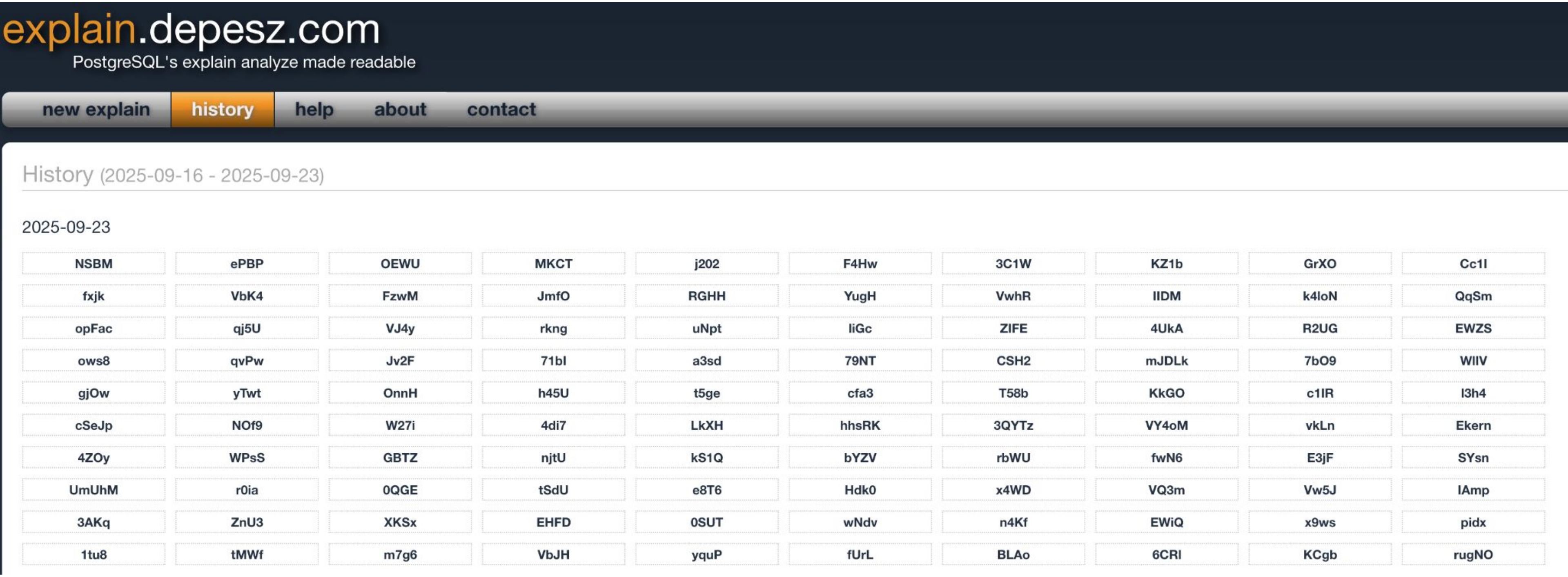
depesz



☑ I want this plan to be visible on the history page.

☐ I want this plan to be obfuscated before saving. (Note that this makes plans

depesz



History (2025-09-16 - 2025-09-23)

2025-09-23

NSBM	ePBP	OEWU	MKCT	j202	F4Hw	3C1W	KZ1b	GrXO	Cc1I
fxjk	VbK4	FzwM	JmfO	RGHH	YugH	VwhR	IIDM	k4loN	QqSm
opFac	qj5U	VJ4y	rkng	uNpt	liGc	ZIFE	4UkA	R2UG	EWZS
ows8	qvPw	Jv2F	71bl	a3sd	79NT	CSH2	mJDLk	7bO9	WIIV
gjOw	yTwt	OnnH	h45U	t5ge	cfa3	T58b	KkGO	c1IR	l3h4
cSeJp	NOf9	W27i	4di7	LkXH	hhsRK	3QYTz	VY4oM	vkLn	Ekern
4ZOy	WPsS	GBTZ	njtU	kS1Q	bYZV	rbWU	fwN6	E3jF	SYsn
UmUhM	r0ia	0QGE	tSdU	e8T6	Hdk0	x4WD	VQ3m	Vw5J	IAmp
3AKq	ZnU3	XKSx	EHFD	0SUT	wNdv	n4Kf	EWiQ	x9ws	pidx
1tu8	tMWf	m7g6	VbJH	yquP	fUrL	BLAo	6CRI	KCgb	rugNO

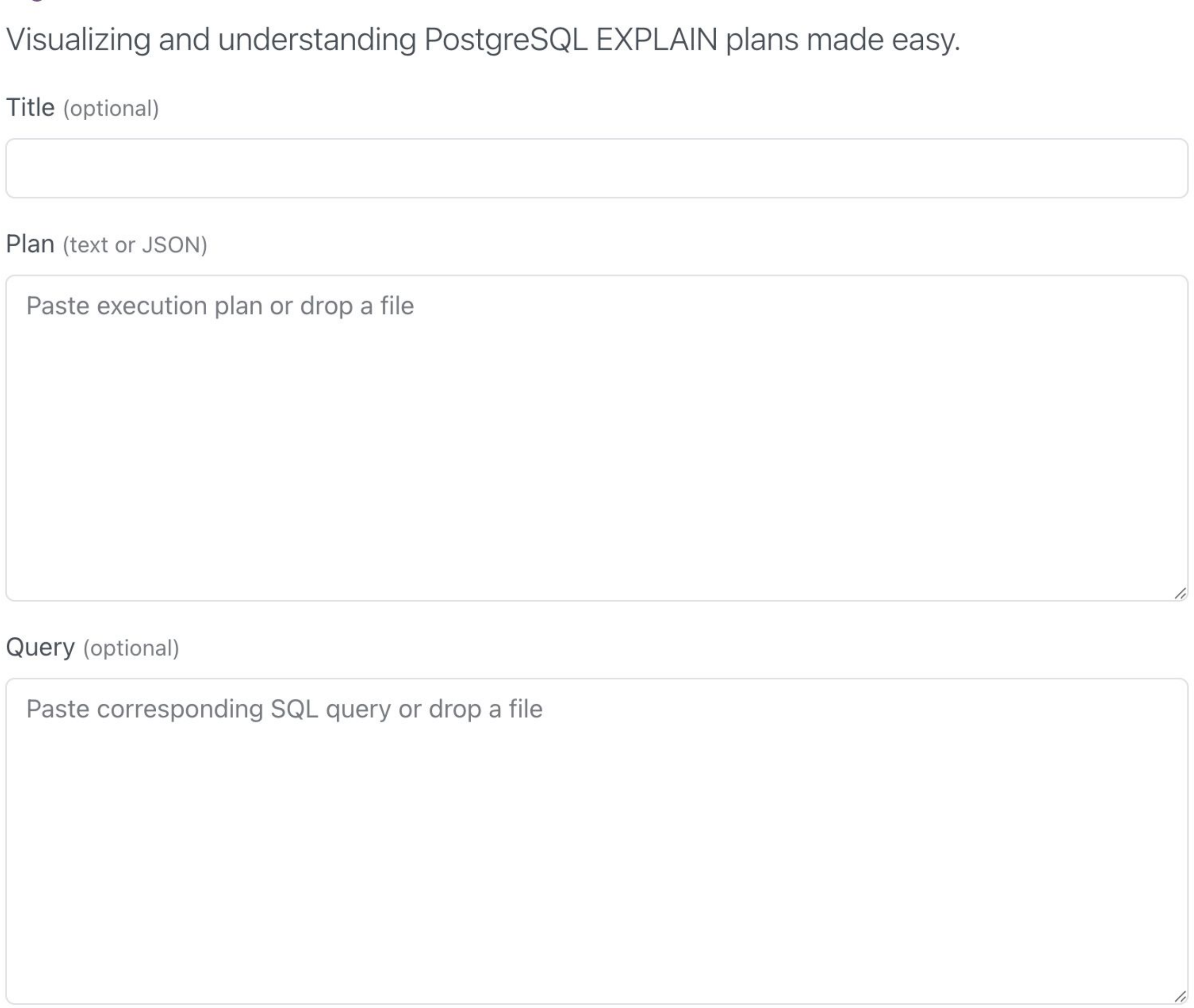
depesz

https://explain.depesz.com/s/EZB5

Dalibo

explain.dalibo.com

Dalibo



Visualizing and understanding PostgreSQL EXPLAIN plans made easy.

Title (optional)

Plan (text or JSON)

Paste execution plan or drop a file

Query (optional)

Paste corresponding SQL query or drop a file

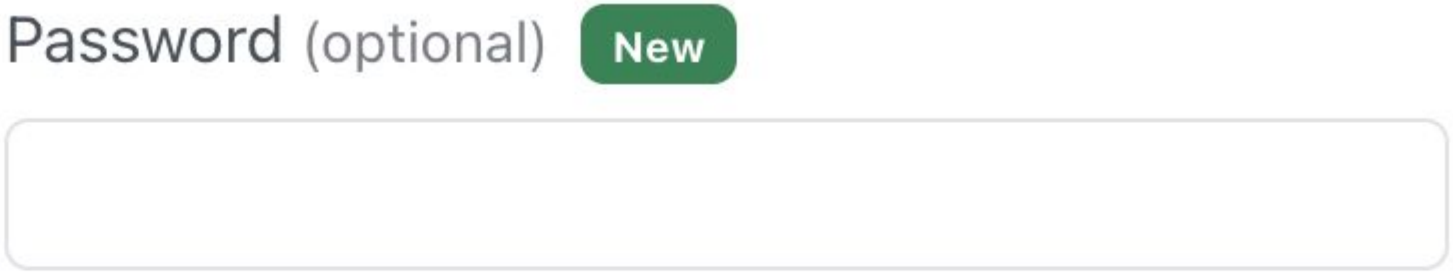
Dalibo

https://explain.dalibo.com/plan/aa

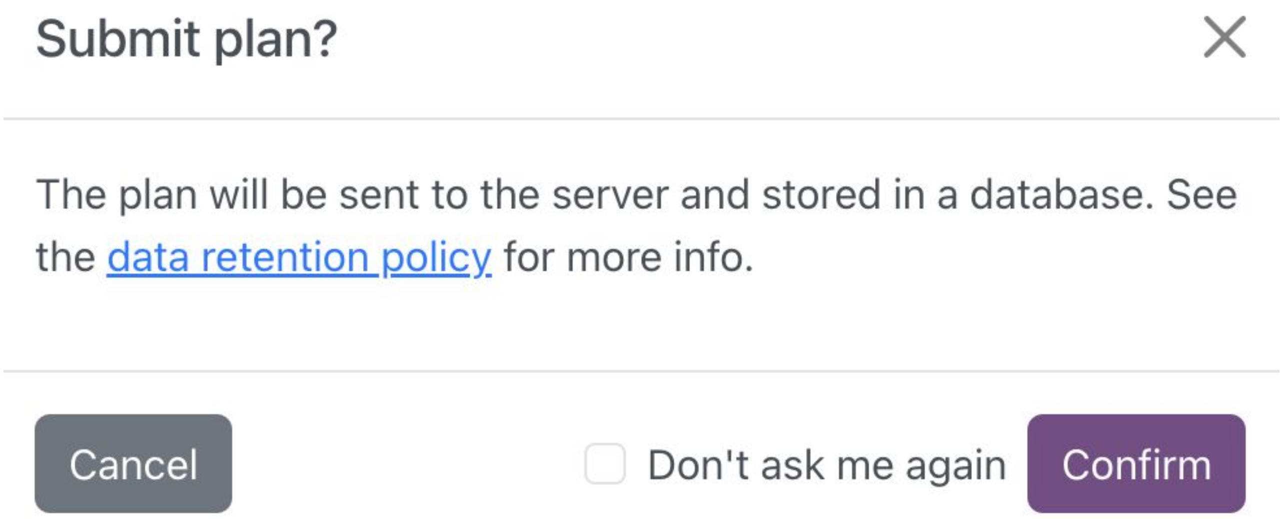
https://explain.dalibo.com/plan/faD

https://explain.dalibo.com/plan/87b216ccdgf491e0

Dalibo



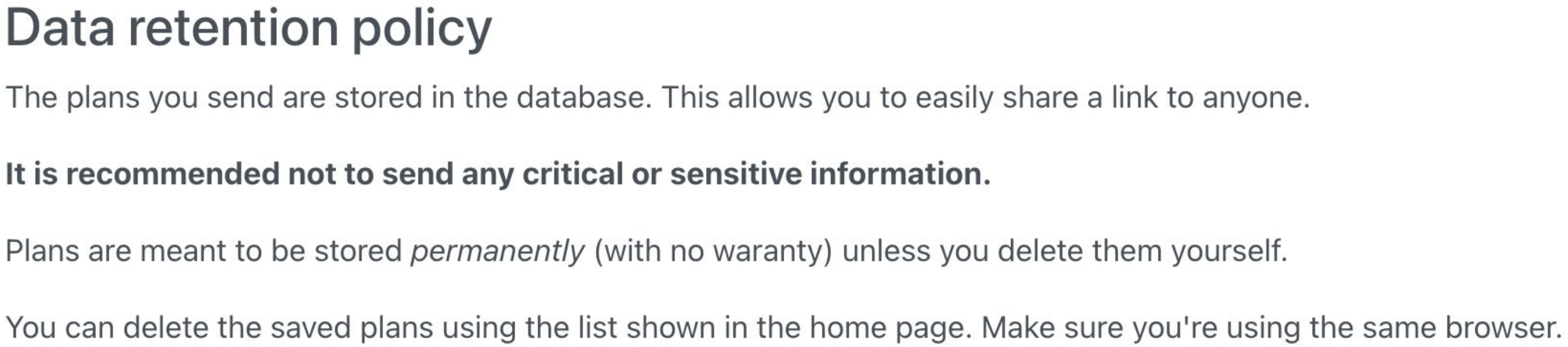
Dalibo



Submit plan?

The plan will be sent to the server and stored in a database. See the data retention policy for more info.

Dalibo


Data retention policy

The plans you send are stored in the database. This allows you to easily share a link to anyone.

It is recommended not to send any critical or sensitive information.

Plans are meant to be stored *permanently* (with no waranty) unless you delete them yourself.

You can delete the saved plans using the list shown in the home page. Make sure you're using the same browser.

Why / Where it leaks

- Literals in SQL Query appears in EXPLAIN output
- Fat fingers

Takeways

- EXPLAIN output is as sensitive as the SQL query used for the EXPLAIN
- People assume services are safe by default ((or don’t pay attention) or don’t care)

- End result: It’s not so bad (very few “hits” compared to the number of plans)

What can you do about it ?

- Check your Slack
- Educate
- Self Host

Stats (WIP)

- 100+ Emails
- 30+ IPs / Domains / URLs
- 4 verified secrets

Stats (WIP)

- 100+ Emails
- 30+ IPs / Do
- 4 verified se

Stats (WIP)

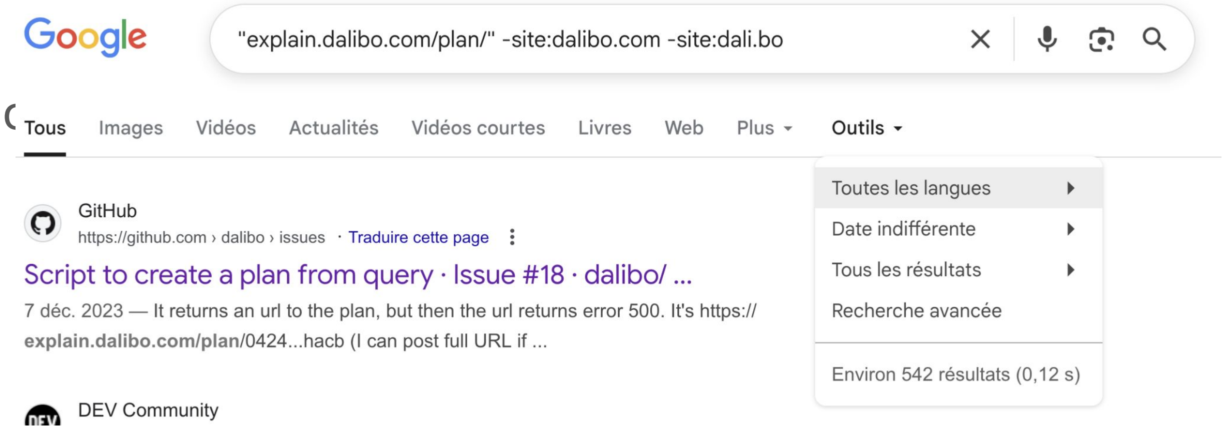
- 100+ Emails
- 30+ IPs / Do
- 4 verified se

Going further

- Active monitoring of new plans on depesz
- Dorks

Going further

- Active monitoring c
- Dorks


"explain.dalibo.com/plan/" -site:dalibo.com -site:dali.bo

Environ 542 résultats (0,12 s)

GitHub
https://github.com › dalibo › issues · Traduire cette page

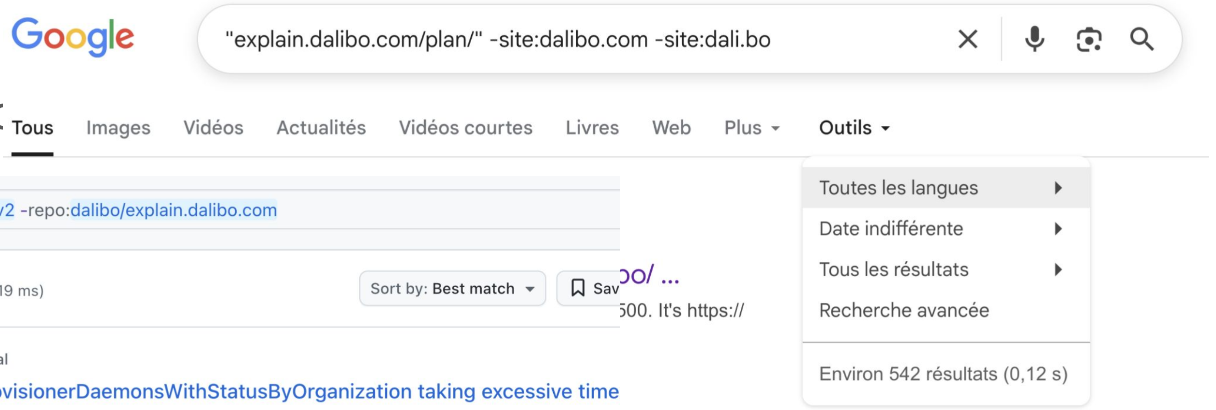
Script to create a plan from query · Issue #18 · dalibo/ ...

7 déc. 2023 — It returns an url to the plan, but then the url returns error 500. It's https://explain.dalibo.com/plan/0424...hacb (I can post full URL if ...

DEV Community

Going further

- Active monitoring (Dorks

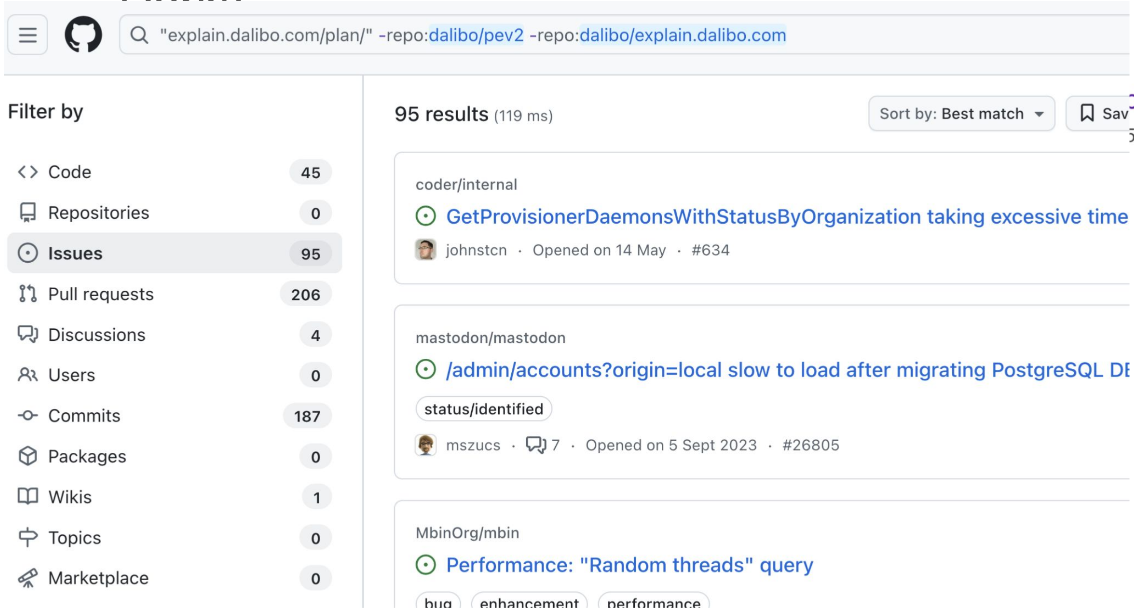


"explain.dalibo.com/plan/" -site:dalibo.com -site:dali.bo

Tous · Images · Vidéos · Actualités · Vidéos courtes · Livres · Web · Plus · Outils

Toutes les langues
Date indifférente
Tous les résultats
Recherche avancée

Environ 542 résultats (0,12 s)



"explain.dalibo.com/plan/" -repo:dalibo/pev2 -repo:dalibo/explain.dalibo.com

Filter by

Code	45
Repositories	0
Issues	95
Pull requests	206
Discussions	4
Users	0
Commits	187
Packages	0
Wikis	1
Topics	0
Marketplace	0

95 results (119 ms)

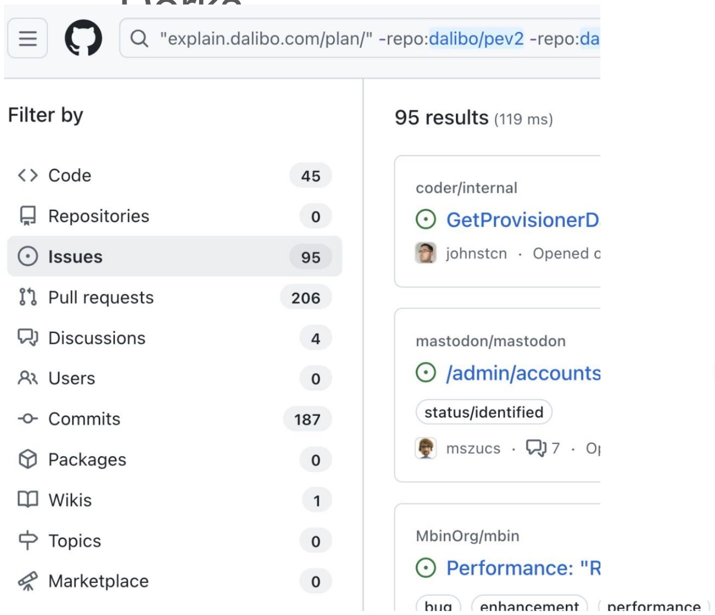
coder/internal
GetProvisionerDaemonsWithStatusByOrganization taking excessive time
johnstcn · Opened on 14 May · #634

mastodon/mastodon
/admin/accounts?origin=local slow to load after migrating PostgreSQL DE
status/identified
mszucs · 7 · Opened on 5 Sept 2023 · #26805

MbinOrg/mbin
Performance: "Random threads" query
bug enhancement performance

Going further

- Active monitoring c
  Dorks



"explain.dalibo.com/plan/" -repo:dalibo/pev2 -repo:da

Filter by

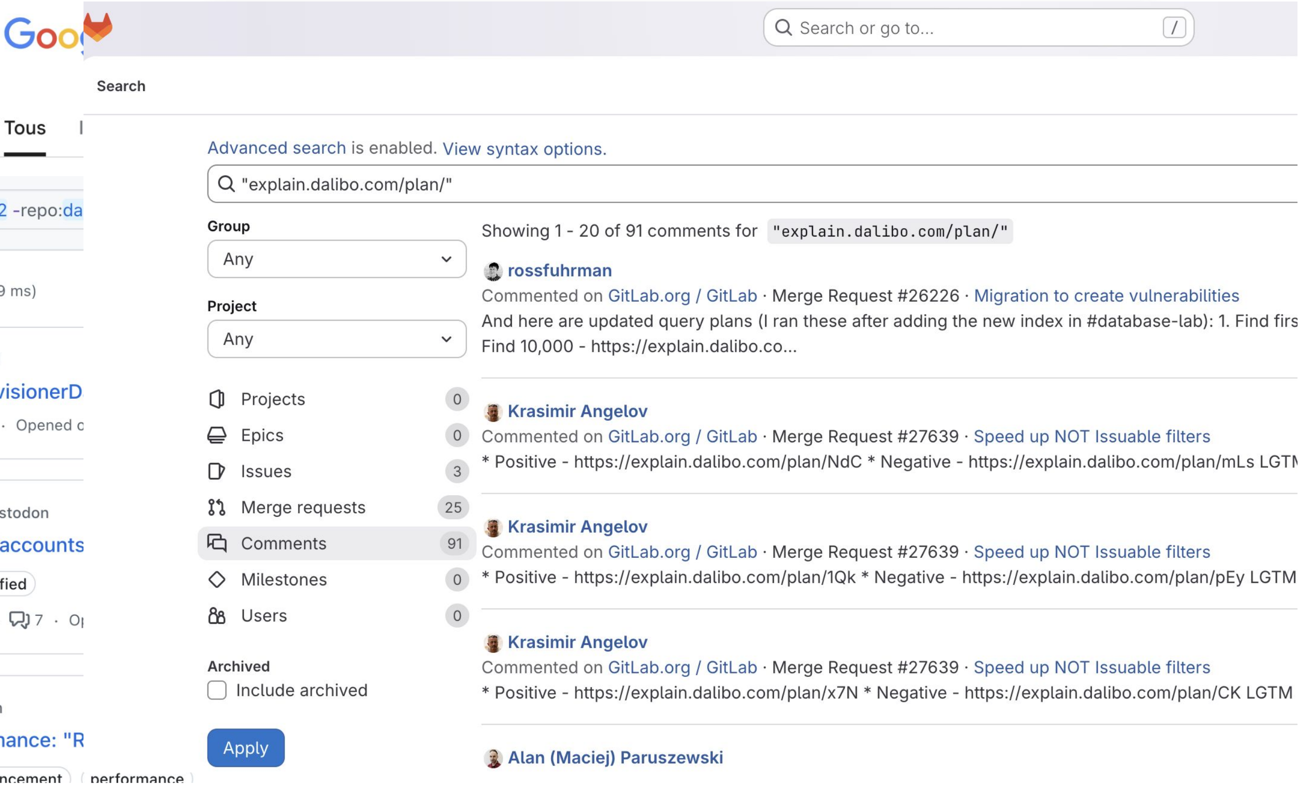
Code	45
Repositories	0
Issues	95
Pull requests	206
Discussions	4
Users	0
Commits	187
Packages	0
Wikis	1
Topics	0
Marketplace	0

95 results (119 ms)

coder/internal
GetProvisionerD
johnstcn · Opened c

mastodon/mastodon
/admin/accounts
status/identified
mszucs · 7 · O

MbinOrg/mbin
Performance: "R
bug enhancement performance



Advanced search is enabled. View syntax options.

"explain.dalibo.com/plan/"

Group: Any

Project: Any

Projects	0
Epics	0
Issues	3
Merge requests	25
Comments	91
Milestones	0
Users	0

Archived
Include archived

Apply

Showing 1 - 20 of 91 comments for "explain.dalibo.com/plan/"

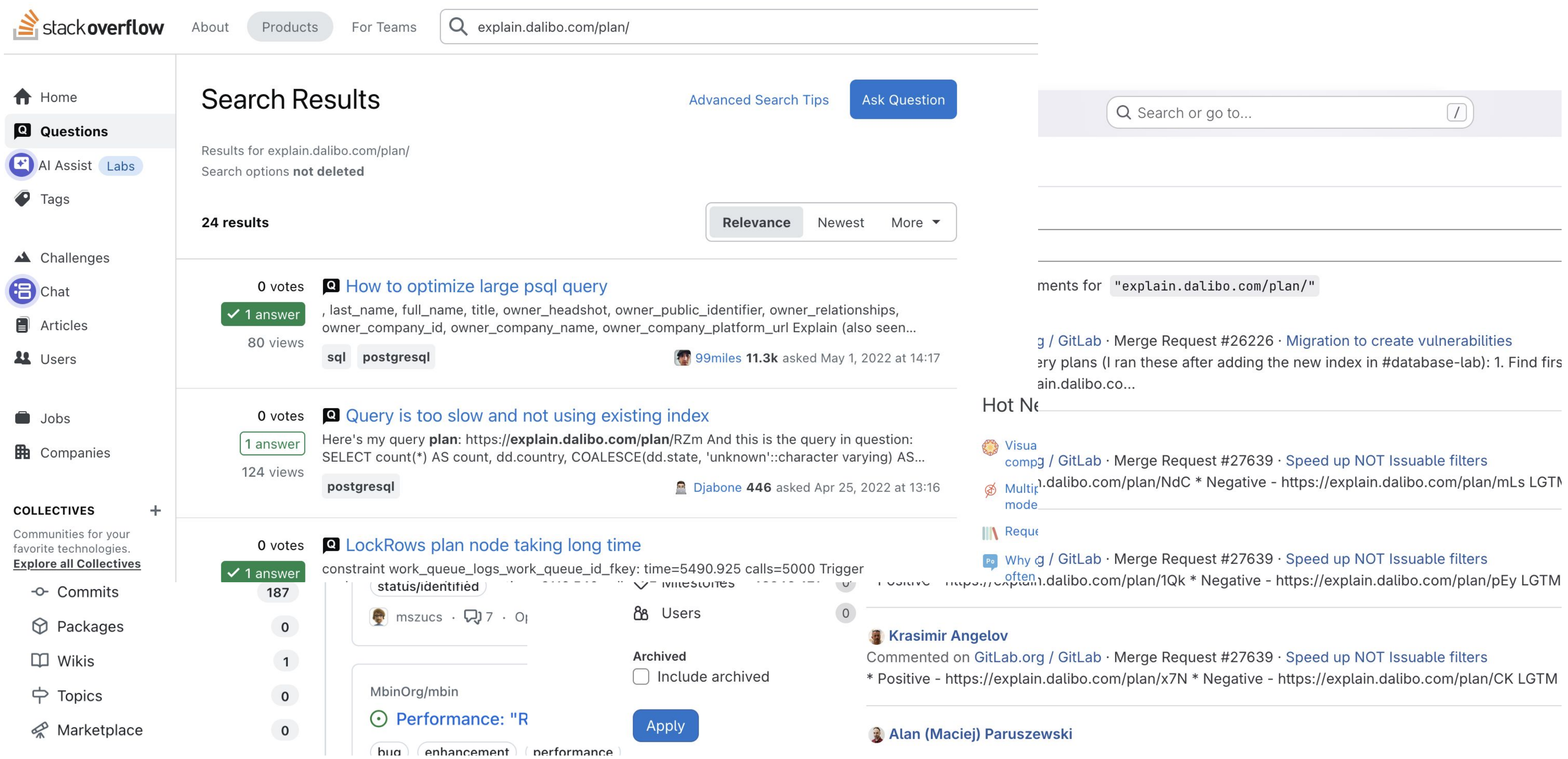
rossfuhrman
Commented on GitLab.org / GitLab · Merge Request #26226 · Migration to create vulnerabilities
And here are updated query plans (I ran these after adding the new index in #database-lab): 1. Find firs
Find 10,000 - https://explain.dalibo.co...

Krasimir Angelov
Commented on GitLab.org / GitLab · Merge Request #27639 · Speed up NOT Issuable filters
* Positive - https://explain.dalibo.com/plan/NdC * Negative - https://explain.dalibo.com/plan/mLs LGTM

Krasimir Angelov
Commented on GitLab.org / GitLab · Merge Request #27639 · Speed up NOT Issuable filters
* Positive - https://explain.dalibo.com/plan/1Qk * Negative - https://explain.dalibo.com/plan/pEy LGTM

Krasimir Angelov
Commented on GitLab.org / GitLab · Merge Request #27639 · Speed up NOT Issuable filters
* Positive - https://explain.dalibo.com/plan/x7N * Negative - https://explain.dalibo.com/plan/CK LGTM

Alan (Maciej) Paruszewski



Search Results

Results for explain.dalibo.com/plan/
Search options **not deleted**

24 results

0 votes · ✓ 1 answer · 80 views
How to optimize large psql query
, last_name, full_name, title, owner_headshot, owner_public_identifier, owner_relationships, owner_company_id, owner_company_name, owner_company_platform_url Explain (also seen...
sql postgresql — 99miles **11.3k** asked May 1, 2022 at 14:17

0 votes · 1 answer · 124 views
Query is too slow and not using existing index
Here's my query **plan**: https://**explain.dalibo.com/plan**/RZm And this is the query in question: SELECT count(*) AS count, dd.country, COALESCE(dd.state, 'unknown'::character varying) AS...
postgresql — Djabone **446** asked Apr 25, 2022 at 13:16

0 votes · ✓ 1 answer
LockRows plan node taking long time
constraint work_queue_logs_work_queue_id_fkey: time=5490.925 calls=5000 Trigger

status/identified
mszucs · 7 · O

MbinOrg/mbin
Performance: "R
bug enhancement performance

ments for "explain.dalibo.com/plan/"

g / GitLab · Merge Request #26226 · Migration to create vulnerabilities
ery plans (I ran these after adding the new index in #database-lab): 1. Find firs
ain.dalibo.co...

g / GitLab · Merge Request #27639 · Speed up NOT Issuable filters
n.dalibo.com/plan/NdC * Negative - https://explain.dalibo.com/plan/mLs LGTM

g / GitLab · Merge Request #27639 · Speed up NOT Issuable filters
n.dalibo.com/plan/1Qk * Negative - https://explain.dalibo.com/plan/pEy LGTM

Krasimir Angelov
Commented on GitLab.org / GitLab · Merge Request #27639 · Speed up NOT Issuable filters
* Positive - https://explain.dalibo.com/plan/x7N * Negative - https://explain.dalibo.com/plan/CK LGTM

Alan (Maciej) Paruszewski

That’s all folks

-