How to search every table & column in Postgres

(without learning the schema)

Sehrope Sarkuni
JackDB, Inc.
PGConf US 2015

Background

- PostgreSQL is a relational database
- Data is stored in tables
- Tables have columns
- We want to search for something

JackDB
User-centered database security

Problem

- I don’t know the tables
- I don’t know the columns

JackDB
User-centered database security

Problem (revised)

- ~~*I don’t know the tables*~~
- I don’t want to learn the tables
- ~~*I don’t know the columns*~~
- I don’t want to learn the columns

JackDB
User-centered database security

Solution: JSON! (row_to_json)

```
CREATE OR REPLACE FUNCTION gen_search_all () RETURNS text AS
$$
  SELECT string_agg(z.sql, CHR(10) || ' UNION ALL ' || CHR(10))
  FROM
     (SELECT 'SELECT ' || quote_literal(t.table_schema) || '::text AS table_schema'
          ||      ', ' || quote_literal(t.table_name) || '::text AS table_name'
          ||      ', row_to_json(t) AS data'
          || ' FROM ' || quote_ident(t.table_schema) || '.' || quote_ident(t.table_name) || ' t'
          || ' WHERE '
          || (SELECT string_agg(quote_ident(z.column_name) || '::text ~* $1', ' OR ')
                FROM information_schema.columns z
                WHERE z.table_schema = t.table_schema
                  AND z.table_name = t.table_name
                  AND (   z.data_type IN ('text', 'json', 'jsonb')
                       OR z.data_type ~* '^char')
              ) AS sql
      FROM information_schema.tables t
      WHERE t.table_schema = 'public'
        AND t.table_type = 'BASE TABLE'
      ORDER BY t.table_schema
             , t.table_name) z;
$$
LANGUAGE SQL;
```

JackDB
User-centered database security

Generated SQL

```
SELECT 'public'::text AS table_schema, 'actor'::text AS table_name,
row_to_json(t) AS data
 FROM public.actor t
 WHERE first_name::text ~* $1 OR last_name::text ~* $1
 UNION ALL
SELECT 'public'::text AS table_schema, 'address'::text AS
table_name, row_to_json(t) AS data
 FROM public.address t
 WHERE address::text ~* $1 OR address2::text ~* $1 OR district::text
~* $1 OR postal_code::text ~* $1 OR phone::text ~* $1
 UNION ALL
SELECT 'public'::text AS table_schema, 'category'::text AS
table_name, row_to_json(t) AS data
 FROM public.category t
 WHERE name::text ~* $1
 UNION ALL
[ ... truncated ... ]
```

JackDB
User-centered database security

Set Returning Wrapper

```
CREATE OR REPLACE FUNCTION search_all (p_text IN text)
RETURNS TABLE (table_schema text, table_name text, data json)
AS
$$
  DECLARE
    l_sql text;
  BEGIN
    l_sql := gen_search_all();
    RETURN QUERY EXECUTE l_sql USING p_text;
  END;
$$
LANGUAGE PLPGSQL;
```

JackDB
User-centered database security

SELECT * FROM search_all(‘john’)

```
-[ RECORD 1 ]
table_schema | public
table_name   | actor
data         | {"actor_id":
5,"first_name":"JOHNNY","last_name":"LOLLOBRIGIDA","last_update":"2006-02-15 09:34:33"}
[ ... truncated ... ]
-[ RECORD 4 ]
table_schema | public
table_name   | customer
data         | {"customer_id":2,"store_id":
1,"first_name":"PATRICIA","last_name":"JOHNSON","email":"patricia.johnson@example.org","
address_id":6,"activebool":true,"create_date":"2006-02-14","last_update":"2006-02-15
09:57:20","active":1}
-[ RECORD 5 ]
table_schema | public
table_name   | customer
data         | {"customer_id":214,"store_id":
1,"first_name":"KRISTIN","last_name":"JOHNSTON","email":"kristin.johnston@example.org","
address_id":218,"activebool":true,"create_date":"2006-02-14","last_update":"2006-02-15
09:57:20","active":1}
[ ... truncated ... ]
```

JackDB
User-centered database security

Expanding JSON in JackDB

JackDB Demo DB (Prod - R

https://app.jackdb.com/editor/E12345678900/

JackDB

JackDB Demo DB (Prod - Read/Write) ACTIVITY SETTINGS

Show sidebar | Editor 1 | Editor 2 | Editor 3 | Editor 4 | Editor 5 | Editor 6

Execute | Explain | Format | Import | Cancel | Save snippet | Load snippet

```
1 SELECT *
2 FROM search_all('john')
3
```

RETRIEVED 8 ROWS IN 0.081 SECONDS—EXECUTED 3 MINUTES AGO. DATA GRID SQL DOWNLOAD

	table_schema	table_name	data.active	data.activebool	data.actor_id	data.address_id	data.create_date	data.customer_id	data.email	data.first_name	data.last_name	data.last_update	data.store_id
1	public	actor	NULL	NULL	5	NULL	NULL	NULL	NULL	"JOHNNY"	"LOLLOBRIG…	"2006-02-15…	NULL
2	public	actor	NULL	NULL	40	NULL	NULL	NULL	NULL	"JOHNNY"	"CAGE"	"2006-02-15…	NULL
3	public	actor	NULL	NULL	192	NULL	NULL	NULL	NULL	"JOHN"	"SUVARI"	"2006-02-15…	NULL
4	public	customer	1	true	NULL	6	"2006-02-14"	2	"PATRIC…	"PATRICIA"	"JOHNSON"	"2006-02-15…	1
5	public	customer	1	true	NULL	218	"2006-02-14"	214	"KRISTI…	"KRISTIN"	"JOHNSTON"	"2006-02-15…	1
6	public	customer	1	true	NULL	305	"2006-02-14"	300	"JOHN.F…	"JOHN"	"FARNSWORT…	"2006-02-15…	1
7	public	customer	1	true	NULL	400	"2006-02-14"	395	"JOHNNY…	"JOHNNY"	"TURPIN"	"2006-02-15…	2
8	public	customer	1	true	NULL	577	"2006-02-14"	571	"JOHNNI…	"JOHNNIE"	"CHISHOLM"	"2006-02-15…	2

JackDB

User-centered database security

Conclusion

- We can use JSON to normalize columns
- We never have to learn a new schema
- We can search everything via one query
- This is probably a terrible idea

JackDB
User-centered database security

Thanks!

sehrope@jackdb.com

https://github.com/sehrope/pg-search-all

JackDB
User-centered database security