

# A look at the Elephants Trunk

# PostgreSQL 16

PGConf.NYC 2023  
New York, USA

Magnus Hagander  
*[magnus@hagander.net](mailto:magnus@hagander.net)*



# Magnus Hagander

- Redpill Linpro
  - Principal database consultant
- PostgreSQL
  - Core Team member
  - Committer
  - PostgreSQL Europe

# PostgreSQL 16

# Development schedule

- June 2022 - branch 15
- July 2022 - CF1
- September 2022 - CF2
- November 2022 - CF3
- January 2023 - CF4
- March 2023 - CF5
- Sep 2023 - Release!!

# New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

**Breaking changes**

# Windows

- Minimum version now Windows 10

# Removed ports

- HPUX
- Itanium



# promote\_trigger\_file

- Removed!
- Use *pg\_ctl promote*
- Or *pg\_promote()*

# vacuum\_defer\_cleanup\_age

- Removed!

# New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

# psql

- Use the extended query protocol

# psql

- Use the extended query protocol

```
postgres=# SELECT count(*) FROM pg_class WHERE
postgres-#         relkind=$1 AND relname NOT LIKE $2
postgres-# \bind 'r' 'pg_%'
postgres-# \g
count
-----
      4
(1 row)
```

# psql

- SHELL\_ERROR
- SHELL\_EXIT\_CODE

# psql

- SHELL\_ERROR
- SHELL\_EXIT\_CODE

```
postgres=# \! exit 2
postgres=# SELECT :SHELL_ERROR, :SHELL_EXIT_CODE;
 t          |          2
```

# SYSTEM\_USER

- What user was used to log in



# SYSTEM\_USER

- What user was used to log in

```
postgres=# SELECT CURRENT_USER, SYSTEM_USER;
 current_user | system_user
-----+-----
 mha          | peer:mha
(1 row)
```

# pg\_hba/pg\_ident

- Match users and databases with regexp

# pg\_hba/pg\_ident

- Match users and databases with regexp

```
host    /^dev    /^dev    10.0.0.0/24    scram-sha-256
host    sameuser /^test   10.0.0.0/24    scram-sha-256
```

# pg\_hba/pg\_ident

- Include files
  - Not just include-lists

# pg\_hba/pg\_ident

- Include files
  - Not just include-lists

```
include_if_exists pg_hba_override.conf
```

```
include pg_hba_test.conf
```

```
include_dir all_my_pghba
```

# Libpq auth

- `require_auth`
  - `require_auth=scram-sha-256`
  - `require_auth=!password`
- Configure `sslcertmode`
  - *allow*, *disable*, *require*
  - `sslcertmode=require`

# Libpq ssl

- Use system CA store
  - `sslrootcert=system`
  - Enforces `verify-full`

# Libpq load balancing

- Old behaviour:
  - `load_balance_hosts=disable`
  - Connect in order, based on *target\_session\_attrs*
- Load balanced:
  - `load_balance_hosts=random`



# Kerberos delegation

- Delegate client permissions downstream
  - postgres\_fdw
  - dblink
- Enable on server: *gss\_accept\_delegation*
- Enable in client (libpq): *gssdelegation=1*

# VACUUM

- Instrumentation

```
frozen: 47 pages from table (35.34% of total) had 10000 tuples frozen
```

# VACUUM

- Reload cost parameters during autovacuum
- Previously only between tables

# VACUUM

## Page level freezing

- Freeze individual pages much earlier
- More efficient than row-at-a-time
- Also triggers after FPI by heap pruning

# VACUUM

- Control memory use

```
postgres=# VACUUM (BUFFER_USAGE_LIMIT '10MB');  
VACUUM
```

- *vacuum\_buffer\_usage\_limit*

# Statistics

- Track last scan of relation
- Separate sequential and indexes

# Statistics

- Track last scan of relation
- Separate sequential and indexes

```
-[ RECORD 1 ]-----+-----  
reloid          | 16392  
.....  
seq_scan       | 13  
last_seq_scan  | 2023-03-05 17:06:23.993055+01  
...  
idx_scan       |  
last_idx_scan  |  
.....
```

# pg\_stat\_io

- More detailed i/o statistics
- Enhances what was in *pg\_stat\_bgwriter*
- Global but by types



# pg\_stat\_io

```
backend_type | standalone backend
io_object    | relation
io_context   | normal
reads        | 536
read_time    | 0
writes       | 1011
write_time   | 0
writebacks   | 0
writeback_time | 0
extends      | 663
extend_time  | 0
op_bytes     | 8192
hits         | 85491
evictions    | 0
reuses       |
```

# New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

# ANY\_VALUE

- SQL standard aggregate
- "Any value in group"

# ANY\_VALUE

- SQL standard aggregate
- "Any value in group"

```
postgres=# SELECT relkind, count(*), ANY_VALUE(relname)
```

```
postgres-# FROM pg_class GROUP BY relkind;
```

relkind	count	any_value
r	70	mytable
v	141	pg_shadow
i	164	pg_toast_1255_index
t	40	pg_toast_1255

# COPY DEFAULT

- Specify default values when loading

# COPY DEFAULT

- Specify default values when loading

```
postgres=# COPY foo FROM '/tmp/t.csv' WITH (FORMAT CSV, DEFAULT '\D');  
COPY 123
```

# Set STORAGE on creation

- Previously required create + alter

# Set STORAGE on creation

- Previously required create + alter

```
CREATE TABLE mytable (  
  a TEXT STORAGE EXTERNAL  
)
```



# Non-decimal literals

- For integer and numeric
- Hexadecimal, octal and binary
- Also allow underscores

# Non-decimal literals

- For integer and numeric
- Hexadecimal, octal and binary
- Also allow underscores

```
postgres=# SELECT 0x123, 0b110, 0o0664;  
          291 |          6 |          436
```

# Non-decimal literals

- For integer and numeric
- Hexadecimal, octal and binary
- Also allow underscores

```
postgres=# SELECT 0x123, 0b110, 0o0664;  
291 | 6 | 436
```

```
postgres=# SELECT 0x12_34, 0x1234, 0x_1234;  
4660 | 4660 | 4660
```

# JSON

- SQL standard JSON constructors
  - JSON\_ARRAY
  - JSON\_ARRAYAGG
  - JSON\_OBJECT
  - JSON\_OBJECTAGG

# JSON

- SQL standard JSON constructors
  - JSON\_ARRAY
  - JSON\_ARRAYAGG
  - JSON\_OBJECT
  - JSON\_OBJECTAGG

```
postgres=# SELECT JSON_OBJECT('foo' : 'bar');  
{"foo" : "bar"}
```

# JSON

- SQL standard JSON predicates
  - IS JSON
  - IS JSON ARRAY
  - IS JSON OBJECT
  - IS JSON SCALAR

# JSON

- SQL standard JSON predicates
  - IS JSON
  - IS JSON ARRAY
  - IS JSON OBJECT
  - IS JSON SCALAR

```
postgres=# select '{}'::jsonb IS JSON OBJECT;  
t
```

# New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance



# pg\_dump compression

- New syntax for specifying
  - *-Z gzip:9*
- LZ4 support
  - *-Z lz4:9*
- ZSTD support
  - *-Z zstd:9*

# Logical replication

## Parallel apply

- Parallel workers on subscription side
- Use multiple processes for the same subscription
- Enabled per subscription

# Logical replication

## Security

- Predefined role: *pg\_create\_subscription*
- Run replication actions as table owner
  - Set *run\_as\_owner=true* to run as subscription owner

# Logical replication

## Replicate from standby

- From physical standby
  - Logical decoding on standby
- Create publication on primary
- But connect replica to standby

# New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

# Many infrastructure

- No direct visibility
- Just runs faster
- E.g. faster memory management

# Explicit SIMD support

- Intel SSE2
- ARM NEON

# Relation extension

- Faster
  - Again



# Parallel aggregates

- `string_agg()`
- `array_agg()`

# Parallel hash full join

- Previously only LEFT
- Now FULL and RIGHT as well

# DISTINCT using Limit

- When possible, use Limit for DISTINCT
- Instead of Unique

# DISTINCT using Limit

- When possible, use Limit for DISTINCT
- Instead of Unique

```
postgres=# EXPLAIN SELECT DISTINCT relname FROM pg_class WHERE relname='test';
Limit (cost=0.27..4.29 rows=1 width=64)
-> Index Only Scan using pg_class_relname_nsp_index on pg_class (cost=0.27.
idth=64)
    Index Cond: (relname = 'test'::name)
```

# GROUP BY/DISTINCT

- Smarter sort orders for aggregates
  - E.g. use an index
- Remove redundant grouping
  - E.g. multiple keys when known identical

# Window functions

- Allowed to alter their own options
- Some functions don't care about ROWS or RANGE
- ROWS is faster than RANGE
- So allow functions to change it on the fly!
  - `row_number()`, `rank()`, `dense_rank()`
  - `percent_rank()`, `cume_dist()`, `ntile()`

**There's always more**

# There's always more

- Lots of smaller fixes
- Performance improvements
- etc, etc
- Can't mention them all!



# Please help!

- Download and test!
  - apt packages available
  - rpm/yum packages available
  - installers available
  - (etc)

# Thank you!

Magnus Hagander

magnus@hagander.net

@magnushagander

<https://www.hagander.net/talks/>

This material is licensed



