SQL START! 2025

# Monitoring PostgreSQL on Premises

Andrea Gnemmi
andrea.gnemmi@gmail.com
a.gnemmi@logifuture.com

# Sponsors & Organizers

[α SYS]

APRA
a vargroup company

Bi Factory
DATA KNOWLEDGE ADVISOR

Lodestar
LEADING INNOVATION

PLUSERVICE.NET
mycicero®
Società del Gruppo
mooney

Dev>Marche

UISS

UNIVERSITÀ
POLITECNICA
DELLE MARCHE

# Who am I?

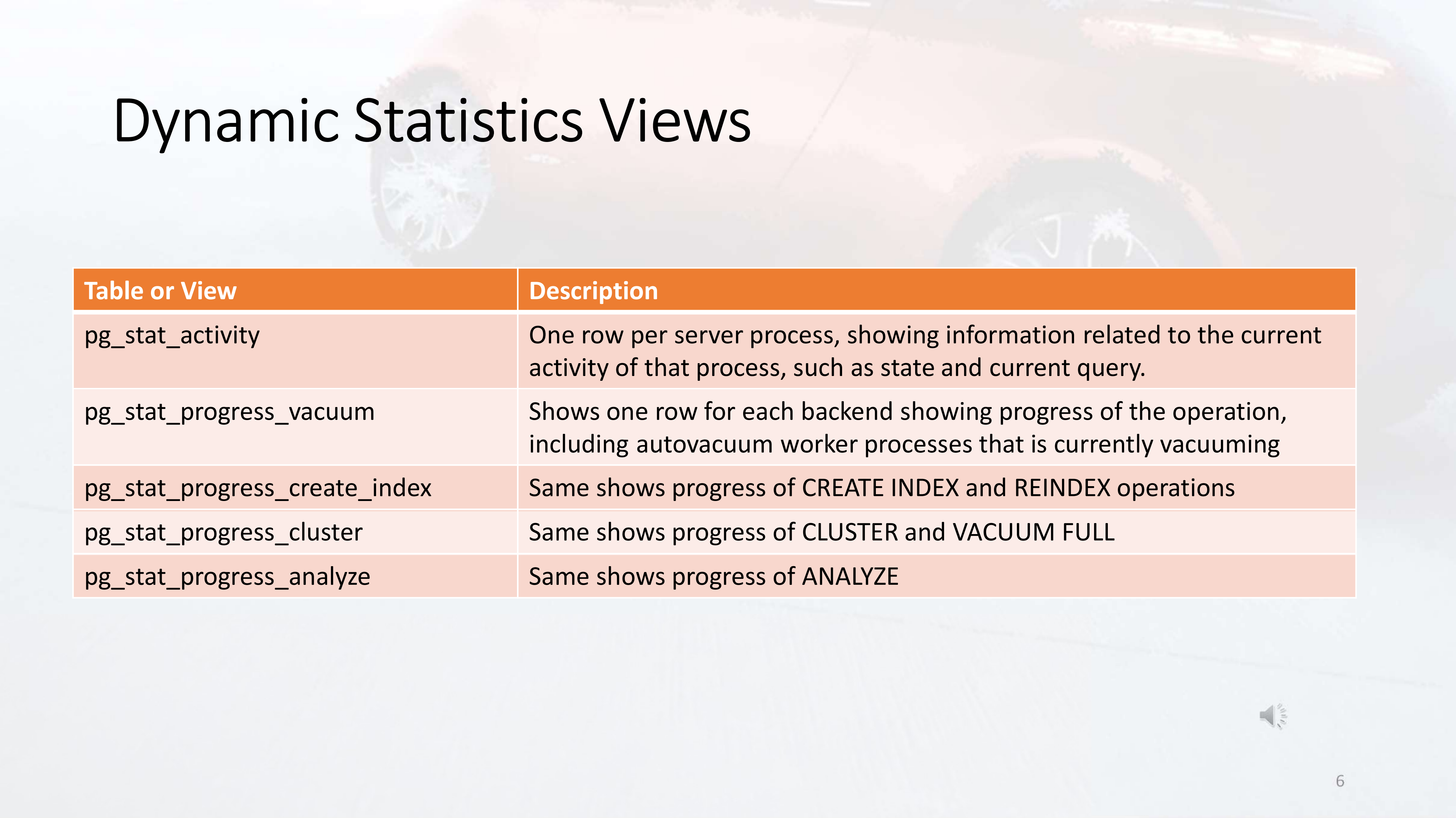
- andrea.gnemmi@gmail.com
- a.gnemmi@logifuture.com
- https://www.logifuture.com/
- https://www.mssqltips.com/sqlserverauthor/390/andrea-gnemmi/
- https://www.linkedin.com/in/andreagnemmi/

# Monitoring PostgreSQL on premises

- In this session we will review some of the monitoring tools available in PostgreSQL
- In particular we will see the cumulative statistics system view pg_stat_activity and the pg_stat_statement extension
- We will also check how to use the system view pg_locks and what offers the pgAdmin GUI Dashboard
- If we will have some time left we will take a look to the cumulative stats views used for monitoring VACUUM/AUTOVACUUM activities and I/O stats

# The cumulative statistics system

- the cumulative statistics system supports collection and reporting of information about server activity.
- Part of the system are dynamic statistics views such as the pg_stat_activity that tells you exactly what’s going on in your system and part are collected statistics views that gives you an idea of the system usage



# Dynamic Statistics Views

| Table or View | Description |
|---|---|
| pg_stat_activity | One row per server process, showing information related to the current activity of that process, such as state and current query. |
| pg_stat_progress_vacuum | Shows one row for each backend showing progress of the operation, including autovacuum worker processes that is currently vacuuming |
| pg_stat_progress_create_index | Same shows progress of CREATE INDEX and REINDEX operations |
| pg_stat_progress_cluster | Same shows progress of CLUSTER and VACUUM FULL |
| pg_stat_progress_analyze | Same shows progress of ANALYZE |

# Collected Statistics Views

| Table or View | Description |
|---|---|
| pg_stat_database | Show statistics about database, temporary file counts and sizes etc. |
| pg_stat_user_tables, pg_stat_sys_tables, pg_stat_all_tables | shows information about activities on a table like inserts, updates, deletes, vacuum, autovacuum etc. distinction between all tables, only current user tables and system ones. |
| pg_stat_user_indexes, pg_stat_sys_indexes, pg_stat_all_indexes | Same as above but for indexes. |
| pg_stat_io | From PostgreSQL 16 I/O statistics for every backend |
| pg_statio_user_tables, pg_statio_sys_tables, pg_statio_all_tables | I/O statistics specific for tables, again with distinction between all tables, only current user tables and system ones. |
| pg_statio_user_indexes, pg_statio_sys_indexes, pg_statio_all_indexes | Same as above but for indexes. |

# pg_stat_activity

- In order to monitor what is happening in real time in the db we can use the view pg_stat_activity
- The pg_stat_activity view will have one row per server process, showing information related to the current activity of that process.
- https://www.postgresql.org/docs/current/monitoring-stats.html#MONITORING-PG-STAT-ACTIVITY-VIEW
- Demo

# Locks in PostgreSQL

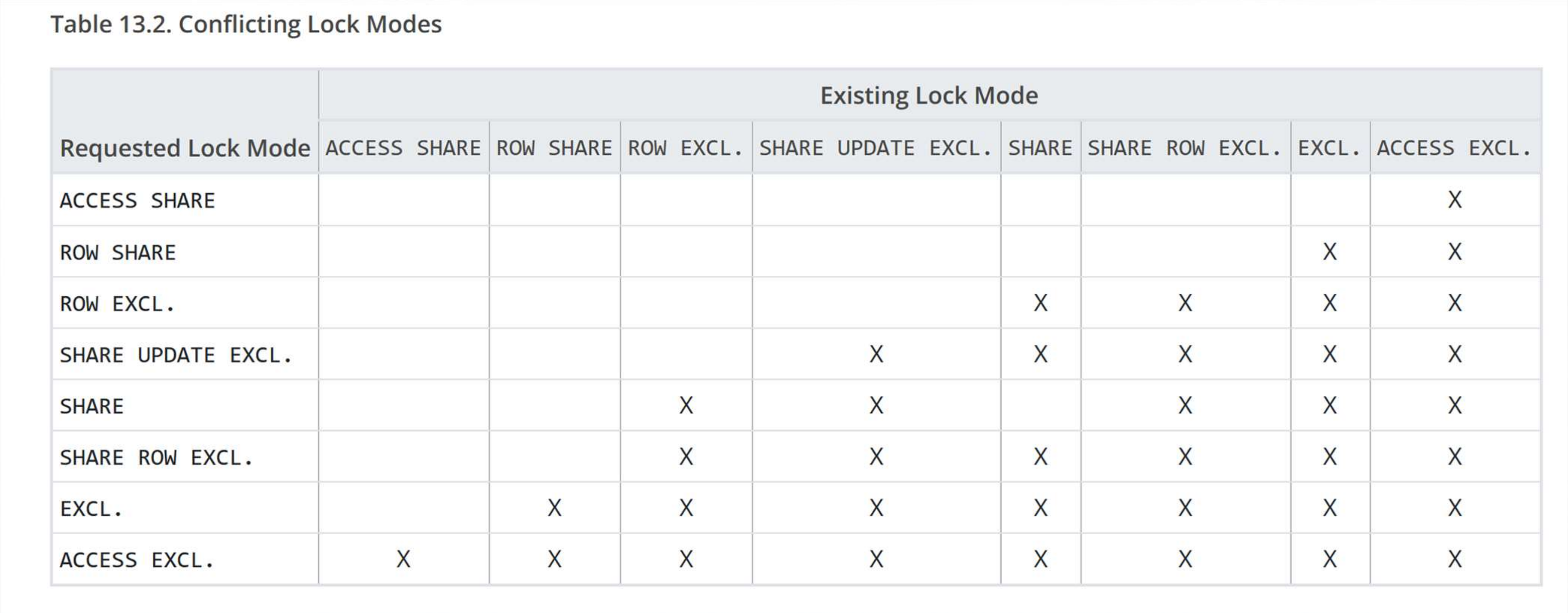


Table 13.2. Conflicting Lock Modes

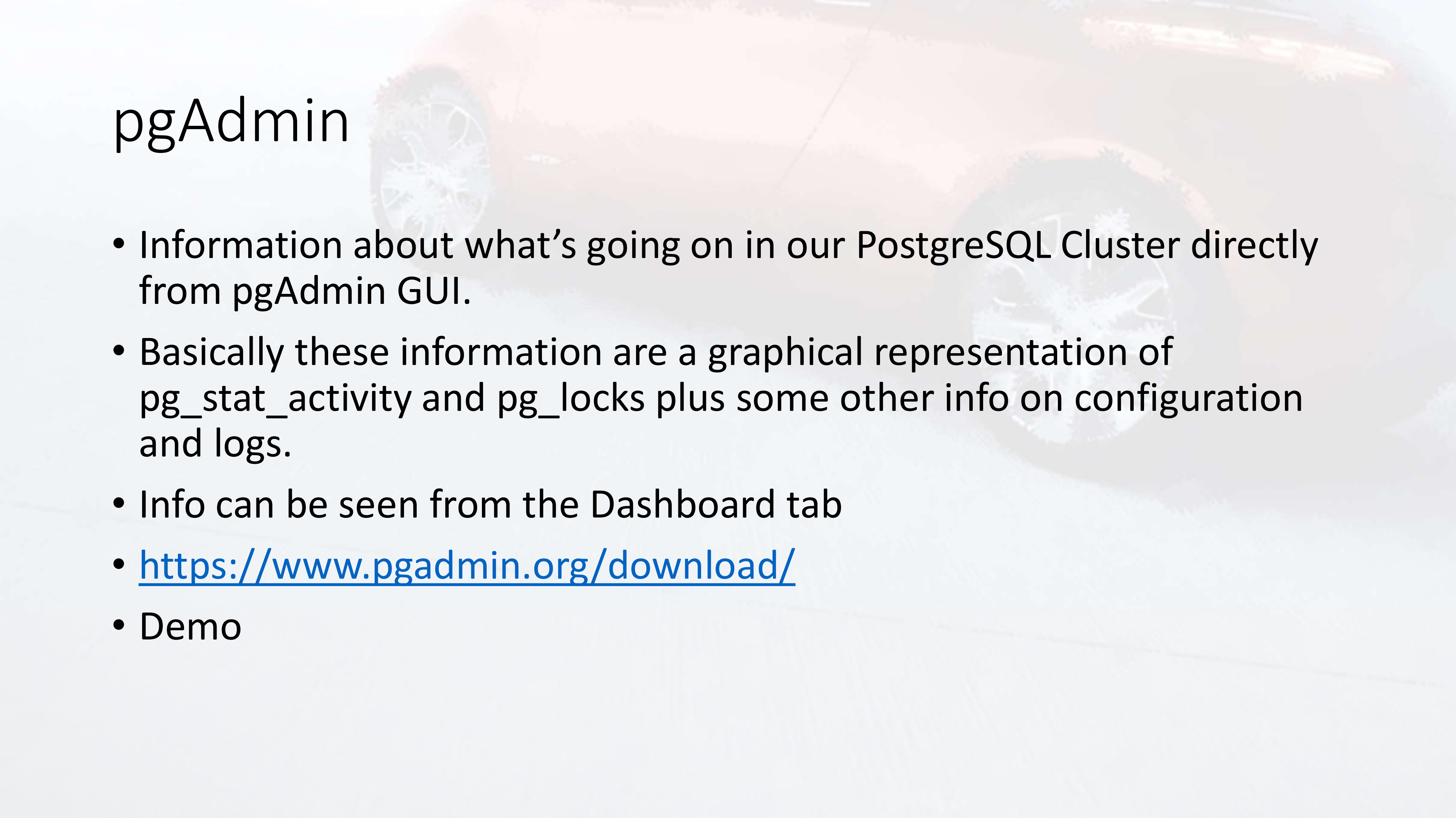
| | Existing Lock Mode | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Requested Lock Mode | ACCESS SHARE | ROW SHARE | ROW EXCL. | SHARE UPDATE EXCL. | SHARE | SHARE ROW EXCL. | EXCL. | ACCESS EXCL. |
| ACCESS SHARE | | | | | | | | X |
| ROW SHARE | | | | | | | X | X |
| ROW EXCL. | | | | | X | X | X | X |
| SHARE UPDATE EXCL. | | | | X | X | X | X | X |
| SHARE | | | X | X | | X | X | X |
| SHARE ROW EXCL. | | | X | X | X | X | X | X |
| EXCL. | | X | X | X | X | X | X | X |
| ACCESS EXCL. | X | X | X | X | X | X | X | X |

# View pg_locks

- As seen in the previous table there are locks also in PostgreSQL even if MVCC is implemented: https://www.interdb.jp/pg/pgsql05.html
- Locks can be monitored from the system view pg_locks
- Function pg_blocking_pids(pid) returns an array of integers with all the blocking pids. Pay attention that, as per official documentation:

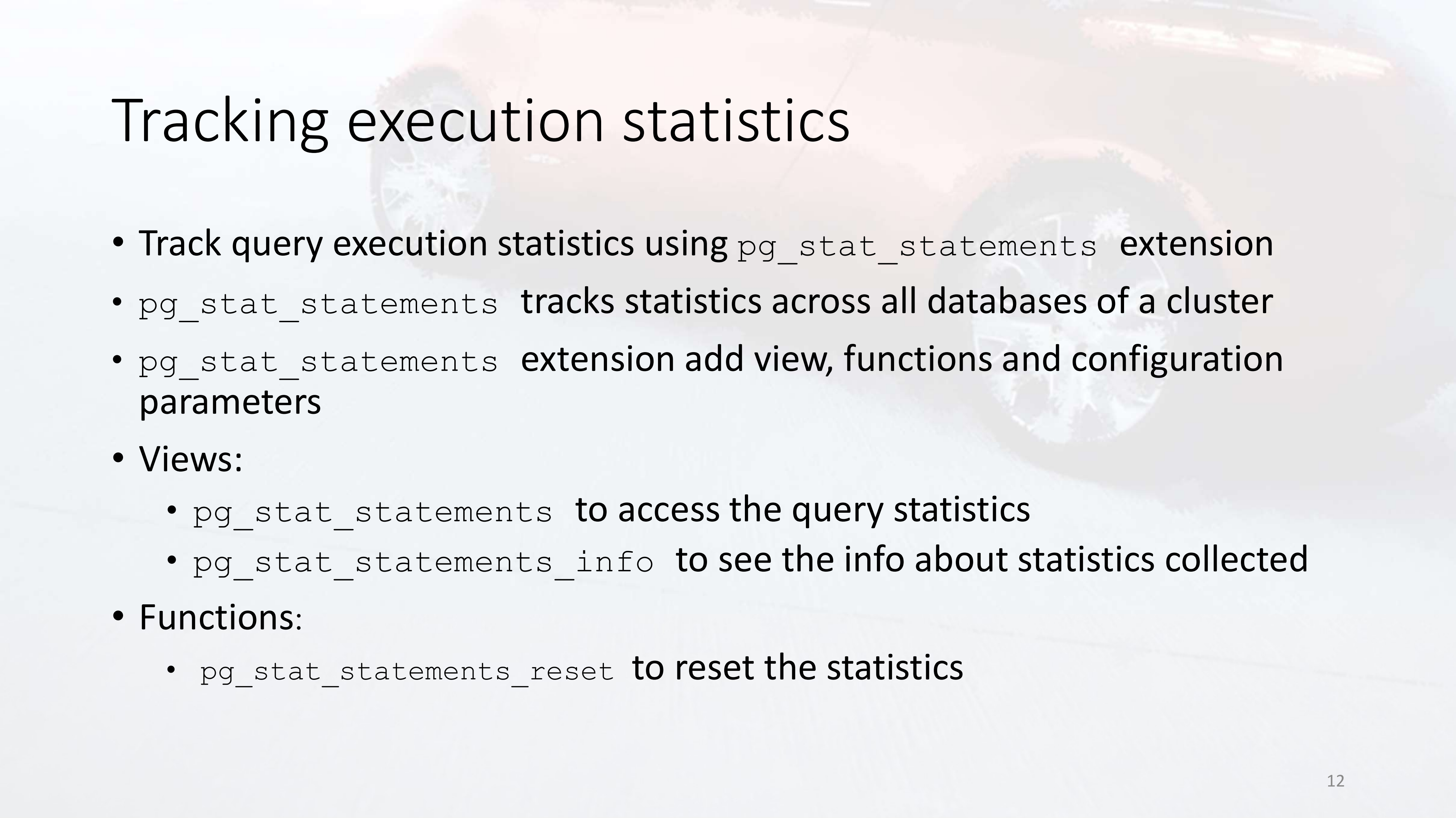
  *Frequent calls to this function could have some impact on database performance, because it needs exclusive access to the lock manager's shared state for a short time.*

- https://www.postgresql.org/docs/current/view-pg-locks.html
- Demo



# pgAdmin

- Information about what’s going on in our PostgreSQL Cluster directly from pgAdmin GUI.
- Basically these information are a graphical representation of pg_stat_activity and pg_locks plus some other info on configuration and logs.
- Info can be seen from the Dashboard tab
- https://www.pgadmin.org/download/
- Demo



# Tracking execution statistics

- Track query execution statistics using `pg_stat_statements` extension
- `pg_stat_statements` tracks statistics across all databases of a cluster
- `pg_stat_statements` extension add view, functions and configuration parameters
- Views:
  - `pg_stat_statements` to access the query statistics
  - `pg_stat_statements_info` to see the info about statistics collected
- Functions:
  - `pg_stat_statements_reset` to reset the statistics

# pg_stat_statements Setup

- Normally the extension package is already installed but maybe needed a:

  ```
  sudo yum install postgresql-contrib
  ```

- Add `pg_stat_statements` to `shared_preload_libraries` parameter in postgresql.conf (requires restart of PostgreSQL cluster)
- Configurable parameters:
  - `pg_stat_statements.max` - Maximum number of tracked statements, default is `5000`
  - `pg_stat_statements.track` – Which statements are counted, `top`, `all` or `none`. Default is `top`
  - `pg_stat_statements.track_utility` - Track commands other than `SELECT`, `INSERT`, `UPDATE` and `DELETE`. Default is `ON`
  - `pg_stat_statements.track_planning` – Track planning operations and duration. Default is `OFF`
  - `pg_stat_statements.save` - Whether save statement statistics across server shutdowns. Default is `ON`
- https://www.postgresql.org/docs/current/pgstatstatements.html
- Demo

# Track VACUUM/AUTOVACUUM activities

- In order to monitor VACUUM/AUTOVACUUM processes and the status of the tables/indexes to be vacuumed we can use the following cumulative stats views and extensions:
  - View pg_stat_progress_vacuum https://www.postgresql.org/docs/current/progress-reporting.html#VACUUM-PROGRESS-REPORTING
  - Catalog view pg_database in order to get available txid https://www.postgresql.org/docs/current/catalog-pg-database.html
  - Catalog view pg_class https://www.postgresql.org/docs/current/catalog-pg-class.html
  - Extension pgstattuple https://www.postgresql.org/docs/current/pgstattuple.html

# Pg_stat_io

- In order to dig deeper into I/O statistics, starting from PostgreSQL version 16, we can use the view pg_stat_io
- Attention: it works only if parameter track_io_timing is set to on, can have significant overhead
- https://www.postgresql.org/docs/current/monitoring-stats.html#MONITORING-PG-STAT-IO-VIEW
- https://tembo.io/blog/optimizing-memory-usage

# Summary

- In this session we have seen multiple ways to monitor some of the key topics in PostgreSQL
- It is by no means a complete list of the possibilities and I omitted all the monitoring tools products such as pganalyze and others
- Questions?
- https://posetteconf.com/2025/schedule/#livestream2

# Grazie!!!