EXPLAIN come visualizzare e leggere piani di esecuzione in PostgreSQL

Sponsor & Org

MINDMASH
KEEP THINKING

digitale

DATA SKILLS
UNDERSTANDING THE WORLD

M D

massive
dynamic *holding*

H

engage
LABS

daniela malvisi
COMMUNICATION

Lucient
ITALIA

Bi Factory
DATA KNOWLEDGE ADVISOR

Who am I?

- Senior DBA @ Logifuture
- “Author”

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

Agenda:

- EXPLAIN
- EXPLAIN ANALYZE
- Other options:
  - GENERIC PLAN
  - BUFFERS (only with EXPLAIN ANALYZE)
  - WAL
  - SETTINGS
  - FORMAT
- Using pgAdmin
- New features in PostgreSQL version 18
- Conclusions and references

EXPLAIN

- In order to obtain the execution plan created by the optimizer in PostgreSQL we have to use the statement EXPLAIN followed by options and the query itself or using other third-party tools in order to visualize it.
- Sections:
  - Access Method, Sequential or Index
  - Join Method, Hash, Nested Loop etc.
  - Join Type, Join Order
  - Sort and Aggregates

EXPLAIN

- In all this different steps we have numbers that are quoted, these are:
  - Estimated start-up cost
  - Estimated total cost
  - Estimated number of rows output by this plan node
  - Estimated average width (in bytes) of rows output by this plan node
- The costs are arbitrary numbers, based on a projected resource usage for a plan, that rely on table statistics, gathered with ANALYZE (or AUTOANALYZE that is automatically run together with AUTOVACUUM)
- System view pg_stats can be used to see table statistics
- Costs also rely on config parameters such as `seq_page_cost`, `random_page_cost`, in which we can specify values different than the default, in order to influence the optimizer to choose a plan instead of another (no plan hints in PostgreSQL!)

EXPLAIN - demo

```
postgres=# \c chinook
You are now connected to database "chinook" as user "postgres".
chinook=# explain (generic_plan) UPDATE pgbench_branches SET bbalance = bbalance + $1 WHERE bid = $2;
                                                QUERY PLAN
-----------------------------------------------------------------------------------------------------------
 Update on pgbench_branches  (cost=0.14..8.16 rows=0 width=0)
   ->  Index Scan using pgbench_branches_pkey on pgbench_branches  (cost=0.14..8.16 rows=1 width=10)
         Index Cond: (bid = $2)
(3 rows)

chinook=#
```

EXPLAIN ANALYZE

- In order to obtain the actual execution plan of the query we need to use EXPLAIN ANALYZE, in this case all the numbers are the real ones and not estimations
- Pay attention if the statement is an INSERT, UPDATE or DELETE because with ANALYZE it will be executed!
- In order to avoid the effects of executing such statements we can use a small trick, inserting the query in a BEGIN block and then issuing a ROLLBACK
- Option BUFFERS needs track_io_timing enabled in order to show time spent reading and writing data file blocks, local blocks and temporary file blocks (in milliseconds)
- Demo

Using pgAdmin

- EXPLAIN and EXPLAIN ANALYZE can be run from GUI pgAgent and visualized in a nicer way
- Using option FORMAT JSON we can visualize plans with pgAgent adding other options
- Demo

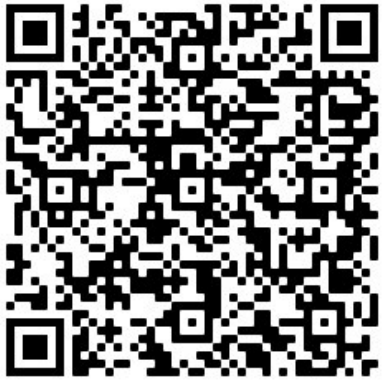
New Features in PostgreSQL version 18

- Option BUFFERS is now default in EXPLAIN ANALYZE (so you have it automatically also in pgAdmin)
- In verbose mode, CPU, WAL, and average-read statistics are now available.
- Index searches shows the efficiency of index scans (the lesser the better)
- When a planner setting is turned off it is now showed, still option settings is better (es. `set enable_seqscan = off;`)

Links and References

- Official documentation: https://www.postgresql.org/docs/current/sql-explain.html
- Table Statistics and how they are used: https://www.crunchydata.com/blog/hacking-the-postgres-statistics-tables-for-faster-queries
- Plan interpretation:
  - https://www.cybertec-postgresql.com/en/how-to-interpret-postgresql-explain-analyze-output/
  - https://www.depesz.com/2013/04/16/explaining-the-unexplainable/
- Settings for influencing query plan:
  https://www.postgresql.org/docs/current/runtime-config-query.html#RUNTIME-CONFIG-QUERY-ENABLE
  https://postgres.ai/blog/20251014-postgres-marathon-2-008
- Hash join explanation: https://www.highgo.ca/2025/10/22/understanding-the-execution-plan-of-a-hash-join/
- Browser based open source plan visualizers:
  - https://explain.dalibo.com/
  - https://explain.depesz.com/
- Licensed tool:
  - pgMustard: https://www.pgmustard.com/

Feedback!!
(don’t shoot the pianist!!!)



DATA SATURDAYS #77 PARMA 2025

Grazie!