

Open in app ↗

Medium

 SearchGeek Culture · [Follow publication](#) Member-only story

How To Read and Understand Explain Query Plans in Postgresql

MynotesoracleDBA · [Follow](#)

Published in Geek Culture

4 min read · Jan 29, 2023



Listen



Share

 More

Here we are going to understand the Explain Query Plan in Detail

In PostgreSQL, an explain query plan shows the execution steps that the database uses to run a query. The explain plan provides information on the type of operations performed, such as sequential scans, index scans, or nested loop joins, as well as the estimated cost of each operation. Here is how to read an explain plan in PostgreSQL:

- Identify the operation type: Each operation in the plan will have a label such as “Seq Scan”, “Index Scan”, “Nested Loop”, etc. These labels describe the type of operation being performed.
- Check the relation name: This is the name of the table or index being operated on.
- Analyze the cost and rows: The cost and rows columns give an estimate of the cost and number of rows the operation will process, respectively.
- Look at the filter conditions: The filter conditions column provides information on any filters applied to the data, such as WHERE clauses.

- **Observe the execution plan:** The explain plan provides a visual representation of the execution steps, with each operation shown as a node and arrows connecting the operations. The order of the operations is shown from left to right, with earlier operations feeding into later operations.
- By analyzing an explain plan, you can gain insight into how the database is processing your query, and identify any performance bottlenecks or areas for optimization.

```

google=# explain select * from office;
               QUERY PLAN
-----
Seq Scan on office (cost=0.00..6.00 rows=400 width=10)
(1 row)
google=#

```

start-up cost total cost output rows avg row size (bytes)

Explain plan in postgresql

In the image above:

- **start-up cost** is the estimated Postgres computational units to start up a node to start process the query.
- **total cost** is the estimated Postgres computational units to finish process the query and return results.
- **output rows** is the estimated number of rows returned.
- **width or average row size (in bytes)** is the estimated size of each output row.

The **start-up cost** and **total cost** are separated by .. simple. From the definition above, we can say the SELECT query costs 6 computational units and returns 400 rows with each weighs 10 bytes. The sequential scan means it iterates through all rows in the table.

Note: these stats are all estimated.

```
google=# explain analyze select * from office;
```

QUERY PLAN

```
Seq Scan on office (cost=0.00..6.00 rows=400 width=10) (actual time=0.014..0.199 rows=400 loops=1)
Planning time: 0.052 ms
Execution time: 0.406 ms
(3 rows)
```



startup
cost

total
cost

output
rows

how many
scans

Explain Analyze in postgresql

As you can see, the plan shows planning and execution time. It also added **actual time** for the sequential scan with a similar format as **cost: start-up time..total time**.

- **start-up time:** time is taken to start up the node and start the sequential scan.
- **total time:** time taken to finish the query and return output rows
- **rows:** number of rows returned.
- **loops:** how many seq scans was performed.

This query does 1 sequential scan that returns 400 rows and takes around 0.46 ms. Simple right!!

QUERY PLAN

```
Hash Join (cost=5.32..7.55 rows=10 width=25)
```

```
Hash Cond: (office.city_id = city.city_id)
```

```
-> Bitmap Heap Scan on office (cost=4.23..6.35 rows=10 width=10)
```

```
Recheck Cond: (office_id < '10'::numeric)
```

```
-> Bitmap Index Scan on office_pkey (cost=0.00..4.22 rows=10 width=0)
```

```
Index Cond: (office_id < '10'::numeric)
```

```
-> Hash (cost=1.04..1.04 rows=4 width=15)
```

```
-> Seq Scan on city (cost=0.00..1.04 rows=4 width=15)
```

```
(8 rows)
```

The query plan you provided is a representation of the steps that PostgreSQL will take to execute a query. Here is a breakdown of what it shows:

- **Hash Join:** The first operation is a Hash Join, which means that the database will use a hash table to join the data from the “office” table and the “city” table based on the “city_id” column.
- **Bitmap Heap Scan on Office:** The next operation is a Bitmap Heap Scan on the “office” table. This means that the database will use a bitmap to identify the rows from the table that match the condition “office_id < ‘10’”.
- **Bitmap Index Scan on Office_pkey:** The bitmap is then used to scan the index “office_pkey” to identify the matching rows.
- **Hash:** The next operation is a Hash, which is used to create a hash table of the data from the “city” table.
- **Seq Scan on City:** The final operation is a Sequential Scan on the “city” table, which means that the database will read the entire table to return the matching rows.

The cost and rows columns give an estimate of the resources (in terms of CPU and disk I/O) required to perform each operation and the number of rows the operation will process, respectively. By analyzing this plan, you can gain insight into how the database is processing your query and identify any performance bottlenecks or areas for optimization.

To Get more update follow here [Youtube](#)

[Postgres](#)[Postgresql Database](#)[Performance Tuning](#)[Explain Plan](#)[Follow](#)

Published in Geek Culture

33K Followers · Last published Sep 1, 2023

A new tech publication by Start it up (<https://medium.com/swlh>).

[Follow](#)

Written by MynotesoracleDBA

311 Followers · 14 Following

As a DBA we are going to write and discuss multiple database administration concepts. "Nothing Grows In Comfort Zone".

Responses (1)



Gvadakte

What are your thoughts?



Nontawat Numor

Dec 30, 2023



0.46

Must be 0.406 instead



[Reply](#)

More from MynotesoracleDBA and Geek Culture



In Nerd For Tech by MynotesoracleDBA

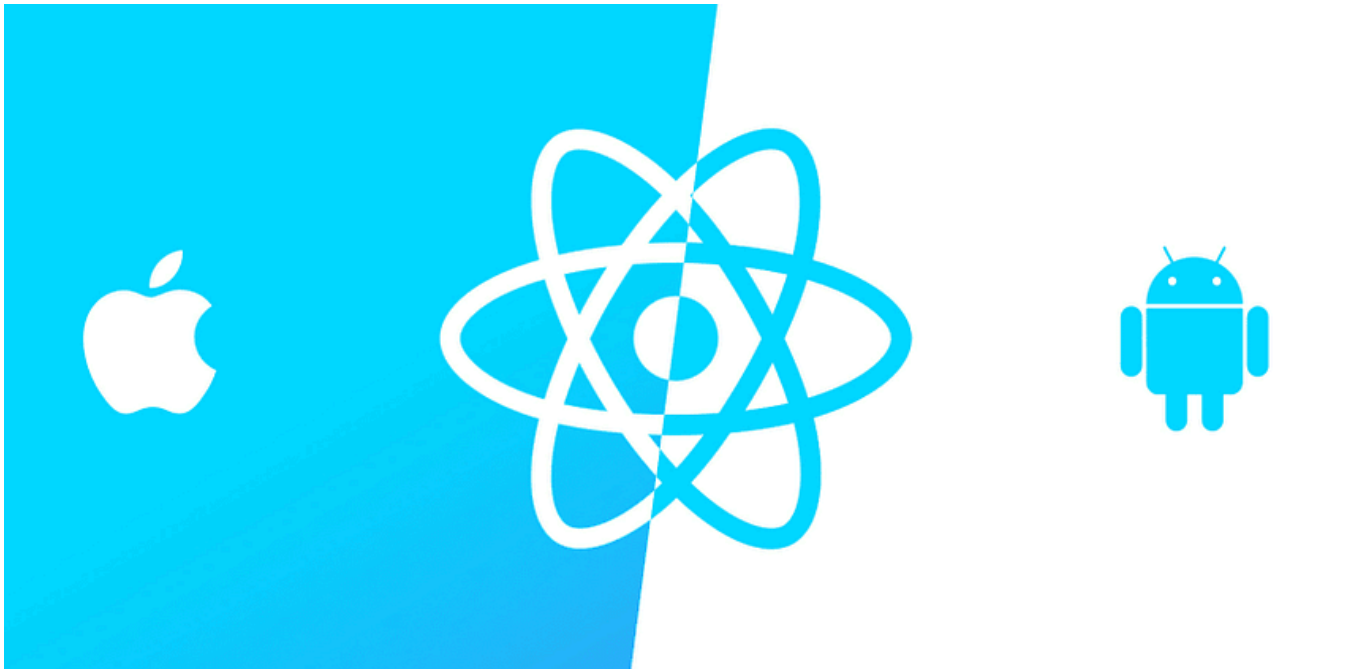
What Is A PGA In Oracle?

Here we are going to understand the components of PGA

Feb 6, 2022



5



In Geek Culture by Anshul Borawake

React Native Generate APK—Debug and Release APK

Generate Debug and Release APK in React Native; Windows, iOS and Linux

Apr 3, 2021



1.8K



16





In Geek Culture by Hasitha Subhashana

Circuit Breaker Pattern (Design Patterns for Microservices)

In a distributed system we have no idea how other components would fail. Network issues could occur, components could fail or a router or a...

Jun 12, 2021 🖱️ 906 💬 11



In Nerd For Tech by MynotesoracleDBA

How a SQL Statement is processed in Oracle?| Interview Q& A

Here we are going to understand the SQL statement processing steps in oracle

Dec 7, 2021  3[See all from MynotesoracleDBA](#)[See all from Geek Culture](#)

Recommended from Medium



In Databases by Sergey Egorenkov

Making SQL query 40x faster for 10 million rows table

Make your SQL query really fast using this approach

Mar 17  15

POSTGRESQL PERFORMANCE OPTIMIZATION



PostgreSQL

 In Towards Dev by Nakul Mitra

PostgreSQL Performance Optimization—Cleaning Dead Tuples & Reindexing

Performance optimization is crucial in PostgreSQL to ensure efficient query execution and minimal resource consumption.

Mar 28  1



```
3. nodeAPP 4. nodeTWO
b/postgresql/16/main/*

t patroni

/etc/patroni.yml list
21665717) -----+-----+-----+
Role      | State      | TL | Lag in MB |
-----+-----+-----+
Leader    | running    | 1  |           |
Replica   | streaming  | 1  | 0         |
Replica   | streaming  | 1  | 0         |
-----+-----+-----+
```

 Dickson Gathima

Building a Highly Available PostgreSQL Cluster with Patroni, etcd, and HAProxy

Achieving high availability in PostgreSQL requires the right combination of tools and architecture.

Mar 14 🖱 4



PostgreSQL



Dumindu Jayasekara

PostgreSQL: How data is stored in your drive

PostgreSQL is an open-source Relational Database Management System that uses and extends the SQL language to provide a powerful, robust...

Dec 28, 2024 🖱 26





Dimitris Moutsatsos

UUID vs. BIGINT: Choosing the Right Primary Key for Your PostgreSQL Cluster

Should I use UUID or BIGINT as the primary key?



Mar 1

17



In Dev Genius by Doran Gao

Efficient Large Table Cleanup in PostgreSQL

“We can only see a short distance ahead, but we can see plenty there that needs to be done.”
—Alan Turing



Oct 31, 2024

1

[See more recommendations](#)