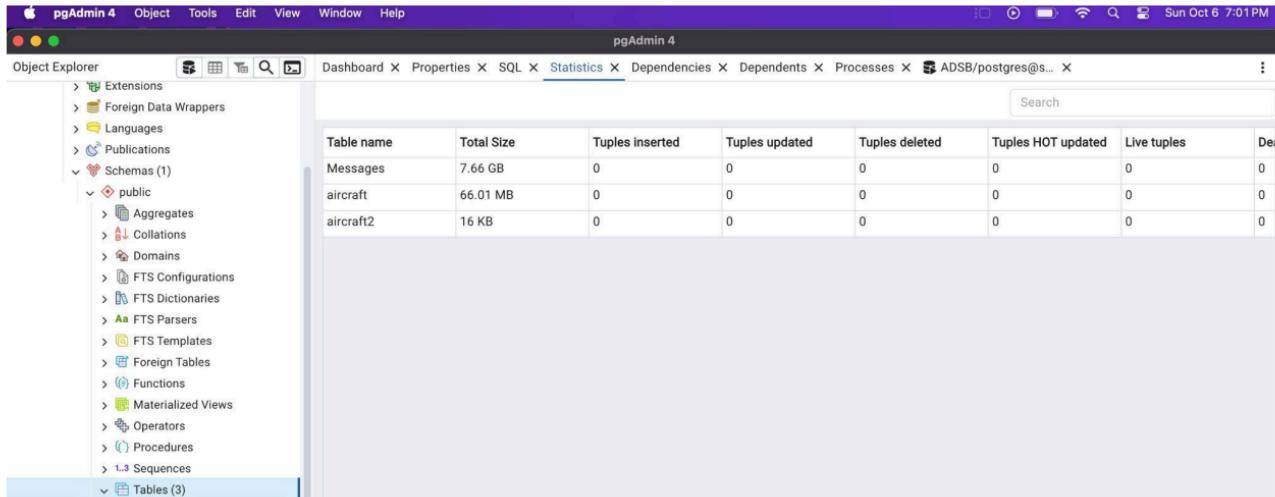


Table Statistics



The screenshot shows the pgAdmin 4 interface with the Statistics tab selected. The left sidebar displays various database objects like Extensions, Foreign Data Wrappers, Languages, Publications, Schemas, etc. The main pane shows a table with statistics for three tables: 'Messages', 'aircraft', and 'aircraft2'. The 'Messages' table has a total size of 7.66 GB and 0 tuples inserted, updated, deleted, or HOT updated. The 'aircraft' table has a total size of 66.01 MB and 0 tuples inserted, updated, deleted, or HOT updated. The 'aircraft2' table has a total size of 16 KB and 0 tuples inserted, updated, deleted, or HOT updated.

Table name	Total Size	Tuples inserted	Tuples updated	Tuples deleted	Tuples HOT updated	Live tuples	Dependencies
Messages	7.66 GB	0	0	0	0	0	0
aircraft	66.01 MB	0	0	0	0	0	0
aircraft2	16 KB	0	0	0	0	0	0

Please ignore aircraft2

Queries

1. We tried a query that returns a list of aircraft, their registration numbers, and how many messages are associated with each aircraft in the Messages table. The results are ordered by the number of messages, so the aircraft with the highest number of messages appears first.

Query

```
SELECT a."icao24", a."registration", COUNT(m."AircraftHex") AS "MessageCount"
FROM "aircraft" a
JOIN "Messages" m ON a."icao24" = m."AircraftHex"
GROUP BY a."icao24", a."registration"
ORDER BY "MessageCount" DESC;
```

- Original Query - took 1 minute 41 seconds to execute

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer, which displays the database structure under the 'nht2-server' node. In the center is the main workspace titled 'ADSB/postgres@nht2-server'. A query window is open with the following SQL code:

```
1 SELECT a."icao24", a."registration", COUNT(m."AircraftHex") AS "MessageCount"
2 FROM "aircraft" a
3 JOIN "Messages" m ON a."icao24" = m."AircraftHex"
4 GROUP BY a."icao24", a."registration"
5 ORDER BY "MessageCount" DESC;
```

Below the query window is a data output grid. The columns are 'icao24', 'registration', and 'MessageCount'. The data shows 50 rows of aircraft information, ordered by MessageCount in descending order. The last row of the data grid is as follows:

icao24	registration	MessageCount
344304	EC-LUB	807

At the bottom of the data grid, it says 'Total rows: 50 of 50' and 'Query complete 00:01:41.513 Ln 5, Col 10'.

- Ran the explain and analyze statement with Timing filter

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- Servers (1)
 - nht2-server
 - Databases (3)
 - ADSB

ADSB/postgres@nht2-server* x

Query Query History

```

1 v SELECT a."icao24", a."registration", COUNT(m."AircraftHex") AS "MessageCount"
2   FROM "aircraft" a
3   JOIN "Messages" m ON a."icao24" = m."AircraftHex"
4   GROUP BY a."icao24", a."registration"
5   ORDER BY "MessageCount" DESC;

```

Data Output Messages Explain X Notifications

Graphical Analysis Statistics

#	Node	Timings		Rows	Loops
		Exclusive	Inclusive		
1.	→ Sort (actual=96357.12..97722.869 rows=50 loops=1)	0.03 ms	97722.869 ms	50	1
2.	→ Aggregate (actual=96356.714..97722.839 rows=50 loo...	0.084 ms	97722.839 ms	50	1
3.	→ Gather Merge (actual=96356.703..97722.756 rows...	1373.275 ms	97722.756 ms	50	1
4.	→ Sort (actual=96349.396..96349.481 rows=17 l...	1.64 ms	96349.481 ms	17	3
5.	→ Aggregate (actual=96347.481..96347.84...	8.439 ms	96347.841 ms	17	3
6.	→ Hash Inner Join (actual=84198.804.... Hash Cond: ((a.icao24)::bpchar = m."Aircr...	14559.084 ms	96339.402 ms	17691	3
7.	→ Seq Scan on aircraft as a (actua...	170.366 ms	170.366 ms	173333	3
8.	→ Hash (actual=81609.952..8160...	63834.9 ms	81609.953 ms	20186615	3
9.	→ Seq Scan on Messages as ...	17775.053 ms	17775.053 ms	20186615	3

Total rows: 1 of 1 Query complete 00:01:43.864 Ln 1, Col 1

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- Servers (1)
 - nht2-server
 - Databases (3)
 - ADSB

ADSB/postgres@nht2-server* x

Query Query History

```

1 v SELECT a."icao24", a."registration", COUNT(m."AircraftHex") AS "MessageCount"
2   FROM "aircraft" a
3   JOIN "Messages" m ON a."icao24" = m."AircraftHex"
4   GROUP BY a."icao24", a."registration"
5   ORDER BY "MessageCount" DESC;

```

Data Output Messages Explain X Notifications

Graphical Analysis Statistics

Statistics per Node Type

Node type	Count	Time spent	% of query
Aggregate	2	8.523 ms	0.01%
Gather Merge	1	1373.275 ms	1.41%
Hash	1	63834.9 ms	65.33%
Hash Inner Join	1	14559.084 ms	14.9%
Seq Scan	2	17945.419 ms	18.37%
Sort	2	1.67 ms	0.01%

Statistics per Relation

Relation name	Scan count	Total time	% of query
Node type	Count	Sum of times	% of relation
Messages	1	17775.053 ms	18.19%
aircraft	1	170.366 ms	0.18%
Seq Scan	1	170.366 ms	100%

Total rows: 1 of 1 Query complete 00:01:43.864 Ln 1, Col 1

Based on the above results, it is clear that a major portion of the query is composed of the hash and sequential scan functions. Messages table has millions of rows and performing sequential scans over such massive data can slow down the query processing. Hence, we added an index to AircraftHex column in Messages table to quickly locate matching values rather than scanning the entire table.

```

CREATE INDEX idx_messages_hexident ON "Messages" ("AircraftHex");

```

Query returned successfully in 3 min 16 secs.

Re-ran the query and noticed that the execution time reduced to 57s from 1m 41s

```

SELECT a."icao24", a."registration", COUNT(m."AircraftHex") AS "MessageCount"
FROM "aircraft" a
JOIN "Messages" m ON a."icao24" = m."AircraftHex"
GROUP BY a."icao24", a."registration"
ORDER BY "MessageCount" DESC;

```

	icao24	registration	MessageCount
1	769101	9V-DHA	6317
2	484416	PH-BQM	6042
3	896523	A6-EVM	5499
4	406590	G-STBD	4548
5	394724	F-GRZE	3828
6	485789	PH-BHO	3426
7	478854	LN-RKS	3122
8	896451	A6-EQB	2678
9	485343	PH-BHI	2525
10	899124	B-16783	2080
11	899137	B-16786	2078
12	896454	A6-EQE	1360
13	485342	PH-BHH	1040
14	769102	9V-DHB	955
15	344304	EC-LUB	807
16	300279	I-LUXO	716

Successfully run. Total query runtime: 57 secs 432 msec. 50 rows affected.

Query with CTE

```

WITH MessageCounts AS (
    SELECT m."AircraftHex", COUNT(*) AS "MessageCount"
    FROM "Messages" m
    GROUP BY m."AircraftHex"
)
SELECT a."icao24", a."registration", mc."MessageCount"
FROM "aircraft" a
JOIN MessageCounts mc ON a."icao24" = mc."AircraftHex"

```

```
ORDER BY mc."MessageCount" DESC;
```

Further implemented Common Table Expression approach to avoid redundant iterations during the join operation for keeping track of the messages count. This, along with indexing, helped in reducing the query execution time to 15s.

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying a tree structure of servers, databases, and objects. The 'public' schema under the 'Tables' node is selected. The right pane has three tabs: 'Query', 'Query History', and 'Scratch Pad'. The 'Query' tab contains the following SQL code:

```
1 v WITH MessageCounts AS (
2   SELECT m."AircraftHex",
3     COUNT(*) AS "MessageCount"
4   FROM "Messages" m
5   GROUP BY m."AircraftHex"
6 )
7   SELECT a."icao24", a."registration", mc."MessageCount"
8   FROM "aircraft" a
9   JOIN MessageCounts mc
10  ON a."icao24" = mc."AircraftHex"
11  ORDER BY mc."MessageCount" DESC;
```

The 'Data Output' tab shows the results of the query, which is a table with columns 'icao24', 'registration', and 'MessageCount'. The data is as follows:

	icao24	registration	MessageCount
1	769101	9V-DHA	6317
2	484416	PH-BQM	6042
3	896523	A6-EVM	5499
4	406590	G-STBD	4548
5	394724	F-GRZE	3828
6	485789	PH-BHO	3426
7	478854	LN-RKS	3122
8	896451	A6-EQB	2678
9	485343	PH-BHI	2525
10	899124	B-16783	2080
11	899137	B-16786	2078
12	906554	AA-EPF	1360

Total rows: 50 of 50 Query complete 00:00:15.572 Ln 11, Col 9

Explain and Analyze results for query with index and CTE implementation. It can be observed that there is a shift from Sequential Scan to Index Scan in this latest query which reduces the processing times significantly. Also, CTE leads to pre-aggregation of data (aggregate node type comprises 53.38% of the query) which results in join operation being applied to a smaller and filtered dataset thereby making it faster compared to the original query's hash join operation.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Servers (1)
nht2-server
Databases (3)
ADS
Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1)
public Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (2)
Messages Columns

ADSB/postgres@nht2-server x ADSB/postgres@nht2-server x ADSB/postgres@nht2-server*

Query Query History

```
1 v WITH MessageCounts AS (
2   SELECT m."AircraftHex",
3         COUNT(*) AS "MessageCount"
4   FROM "Messages" m
5   GROUP BY m."AircraftHex"
6 )
7   SELECT a."icao24", a."registration", mc."MessageCount"
```

Data Output Messages Explain X Notifications

Graphical Analysis Statistics

#	Node	Timings	Rows	Loops	
		Exclusive	Inclusive	Actual	
1.	→ Sort (actual=14228.933..14231.62 rows=50 loops=1)	0.795 ms	14231.62 ms	50	1
2.	→ Hash Inner Join (actual=13796.009..14230.826 rows=50... Hash Cond: ((a.icao24)=b.pchar = mc."AircraftHex")	100.937 ms	14230.826 ms	50	1
3.	→ Seq Scan on aircraft as a (actual=0.564..338.257 ro... → Hash (actual=13788.96..13791.632 rows=9627 loo... Buckets: 16384 Batches: 1 Memory Usage: 570 kB	338.257 ms	338.257 ms	519999	1
4.	→ Subquery Scan (actual=835.925..13778.533 ro... → Aggregate (actual=835.923..13777.032 ro... → Gather Merge (actual=835.851..1376...	13.1 ms	13791.632 ms	9627	1
5.	→ Subquery Scan (actual=835.925..13778.533 ro... → Aggregate (actual=835.923..13777.032 ro... → Aggregate (actual=6.28..9182.4...	1.502 ms	13778.533 ms	9627	1
6.	→ Aggregate (actual=835.923..13777.032 ro... → Gather Merge (actual=835.851..1376...	8.191 ms	13777.032 ms	9627	1
7.	→ Index Only Scan using idx_...	4586.344 ms	13768.841 ms	16618	1
8.	→ Aggregate (actual=6.28..9182.4...	7587.232 ms	9182.498 ms	5539	3
9.	→ Index Only Scan using idx_...	1595.266 ms	1595.266 ms	20186615	3

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Servers (1)
nht2-server
Databases (3)
ADS
Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1)
public Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (2)
Messages Columns Constraints

ADSB/postgres@nht2-server x ADSB/postgres@nht2-server x ADSB/postgres@nht2-server*

Query Query History

```
1 v WITH MessageCounts AS (
2   SELECT m."AircraftHex",
3         COUNT(*) AS "MessageCount"
```

Data Output Messages Explain X Notifications

Graphical Analysis Statistics

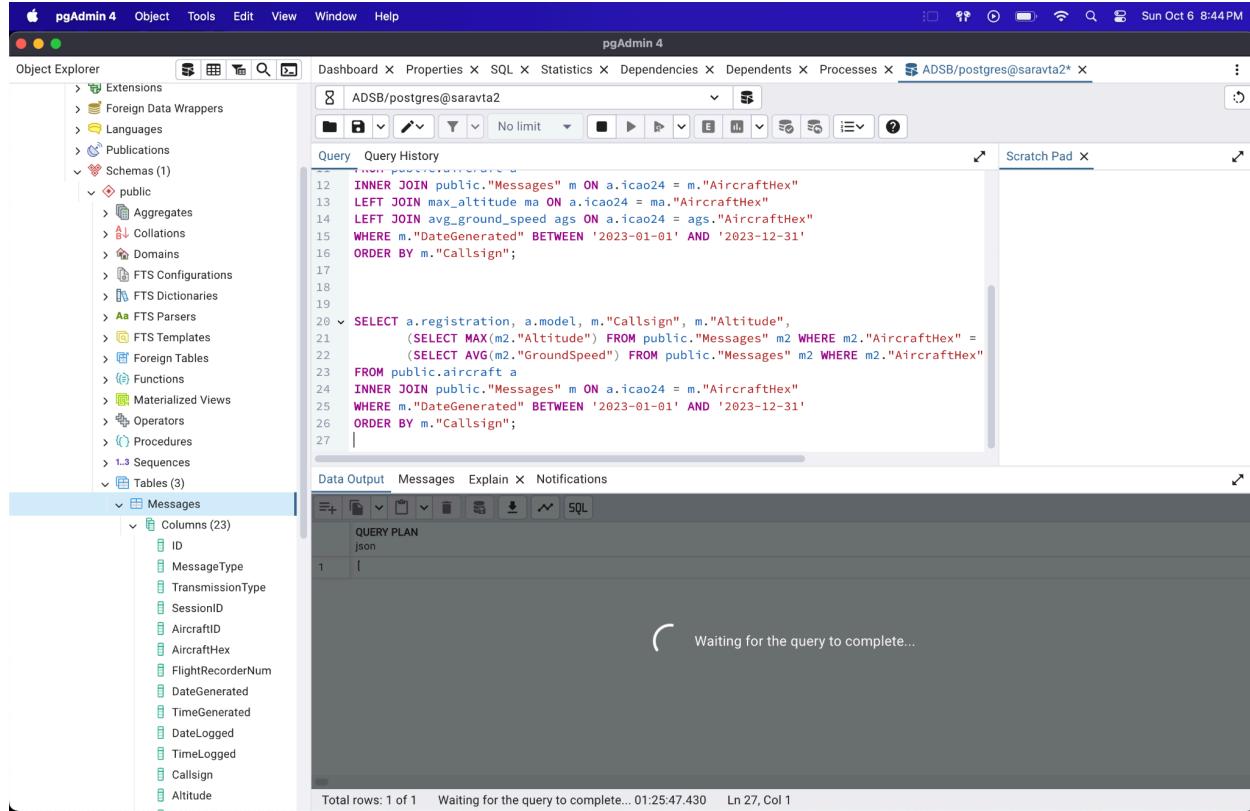
Statistics per Node Type

Node type	Count	Time spent	% of query
Aggregate	2	7595.423 ms	53.38%
Gather Merge	1	4586.344 ms	32.23%
Hash	1	13.1 ms	0.1%
Hash Inner Join	1	100.937 ms	0.71%
Index Only Scan	1	1595.266 ms	11.21%
Seq Scan	1	338.257 ms	2.38%
Sort	1	0.795 ms	0.01%
Subquery Scan	1	1.502 ms	0.02%

Statistics per Relation

Relation name	Scan count	Total time	% of query
Node type	Count	Sum of times	% of relation
Messages	1	1595.266 ms	11.21%
Index Only Scan	1	1595.266 ms	100%
aircraft	1	338.257 ms	2.38%
Seq Scan	1	338.257 ms	100%

2. With curiosity about just how efficient CTEs prove to be, we tried optimizing a query with 2 subqueries and 1 INNER JOIN with a CTE. This query is to return aircraft and message information for a specific time period and to return the maximum altitude and average ground speed of each aircraft.



The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure with Schemas (1), public, Tables (3), and Messages.
- Query Editor:** Contains the following SQL code:

```

-- CTE to get max altitude per aircraft
11 WITH max_altitude AS (
12     INNER JOIN public."Messages" m ON a.icao24 = m."AircraftHex"
13     LEFT JOIN max_altitude ma ON a.icao24 = ma."AircraftHex"
14     LEFT JOIN avg_ground_speed ags ON a.icao24 = ags."AircraftHex"
15     WHERE m."DateGenerated" BETWEEN '2023-01-01' AND '2023-12-31'
16     ORDER BY m."Callsign";
17
18
19
20     SELECT a.registration, a.model, m."Callsign", m."Altitude",
21             (SELECT MAX(m2."Altitude") FROM public."Messages" m2 WHERE m2."AircraftHex" =
22             (SELECT AVG(m2."GroundSpeed") FROM public."Messages" m2 WHERE m2."AircraftHex"
23             FROM public.aircraft a
24             INNER JOIN public."Messages" m ON a.icao24 = m."AircraftHex"
25             WHERE m."DateGenerated" BETWEEN '2023-01-01' AND '2023-12-31'
26             ORDER BY m."Callsign";
27

```
- Data Output:** Shows the results of the query, indicating "Waiting for the query to complete..."
- Bottom Status:** Total rows: 1 of 1, Waiting for the query to complete... 01:25:47.430, Ln 27, Col 1

With a CTE, we observed a large time reduction reducing the execution time to just 5m from 80+ mins.

Query Query History

```

1 v WITH max_altitude AS (
2     SELECT "AircraftHex", MAX("Altitude") AS max_alt
3     FROM public."Messages"
4     GROUP BY "AircraftHex"
5 ), avg_ground_speed AS (
6     SELECT "AircraftHex", AVG("GroundSpeed") AS avg_speed
7     FROM public."Messages"
8     GROUP BY "AircraftHex"
9 )
10 SELECT a.registration, a.model, m."Callsign", m."Altitude", ma.max_alt, ags.avg_speed
11 FROM public.aircraft a
12 INNER JOIN public."Messages" m ON a.icao24 = m."AircraftHex"
13 LEFT JOIN max_altitude ma ON a.icao24 = ma."AircraftHex"
14 LEFT JOIN avg_ground_speed ags ON a.icao24 = ags."AircraftHex"
15 WHERE m."DateGenerated" BETWEEN '2023-01-01' AND '2023-12-31'
16 ORDER BY m."Callsign";

```

Data Output Messages Explain X Notifications

	registration character varying	model character varying (93)	Callsign character varying (20)	Altitude integer	max_alt integer	avg_speed numeric
1	G-VIIL	BOEING 777-236	BAW24R	[null]	34875	405.4285714285714286
2	G-VIIL	BOEING 777-236	BAW24R	[null]	34875	405.4285714285714286
3	G-VIIL	BOEING 777-236	BAW24R	[null]	34875	405.4285714285714286
4	G-VIIL	BOEING 777-236	BAW24R	[null]	34875	405.4285714285714286
5	G-VIIL	BOEING 777-236	BAW24R	[null]	34875	405.4285714285714286
6	G-VIIL	BOEING 777-236	BAW24R	[null]	34875	405.4285714285714286
7	A6-XWA	A350-1041	ETD76T	[null]	25850	389.5833333333333333
8	ET-ASI	B787-8	ETH575	[null]	27225	423.0000000000000000
9	EC-NLP	A350-941	IBE6170	[null]	39000	546.1156462585034014
10	EC-NLP	A350-941	IBE6170	[null]	39000	546.1156462585034014
11	EC-NLP	A350-941	IBE6170	[null]	39000	546.1156462585034014

Total rows: 1000 of 31214

Query complete 00:05:01.651

Ln 3, Col 27

< Graphical Analysis Statistics >

Statistics per Node Type

Node type	Count	Time spent	% of query
Aggregate	2	214733.602 ms	60.91%
Gather Merge	1	552.427 ms	0.16%
Hash	3	42462.009 ms	12.05%
Hash Inner Join	1	4954.945 ms	1.41%
Hash Left Join	2	16.355 ms	0.01%
Seq Scan	4	89856.82 ms	25.49%
Sort	1	10.385 ms	0.01%
Subquery Scan	2	8.13 ms	0.01%

Statistics per Relation

Relation name	Scan count	Total time	% of query
Node type	Count	Sum of times	% of relation
Messages	3	89732.17 ms	25.45%
Seq Scan	3	89732.17 ms	100%
aircraft	1	124.65 ms	0.04%
Seq Scan	1	124.65 ms	100%

Data Output Messages Explain X Notifications

< Graphical Analysis Statistics >

#	Node	Timings		Rows	Loops
		Exclusive	Inclusive		
1.	→ Gather Merge (actual=352048.509..352594.67 rows=3121...	552.427 ms	352594.67 ms	31214	1
2.	→ Sort (actual=352039.633..352042.243 rows=10405 l...	10.385 ms	352042.243 ms	10405	3
3.	→ Hash Left Join (actual=348108.98..352031.859 r... Hash Cond: (a.icao24 = ags."AircraftHex")	5.612 ms	352031.859 ms	10405	3
4.	→ Hash Left Join (actual=202820.987..20674... Hash Cond: (a.icao24 = ma."AircraftHex")	10.743 ms	206740.096 ms	10405	3
5.	→ Hash Inner Join (actual=60286.746..64... Hash Cond: (a.icao24 = m."AircraftHex")	4954.945 ms	64200.15 ms	10405	3
6.	→ Seq Scan on aircraft as a (actual=...	124.65 ms	124.65 ms	173333	3
7.	→ Hash (actual=59120.554..59120.5... Buckets: 131072 Batches: 256 Memory	42454.155 ms	59120.555 ms	5356501	3
8.	→ Seq Scan on Messages as m ... Filter: ("DateGenerated" >= '2023-... Rows Removed by Filter: 1446822	16666.4 ms	16666.4 ms	5356501	3
9.	→ Hash (actual=142529.2..142529.203 r... Buckets: 16384 Batches: 1 Memory Usage: 5	4.067 ms	142529.203 ms	9630	3
10.	→ Subquery Scan (actual=142516.4...	6.724 ms	142525.137 ms	9630	3
11.	→ Aggregate (actual=142514.8... Buckets: Batches: Memory Usage:	106381.389 ms	142518.413 ms	9630	3
12.	→ Seq Scan on Messages ...	36137.024 ms	36137.024 ms	59474190	3
13.	→ Hash (actual=145286.15..145286.151 rows... Buckets: 16384 Batches: 1 Memory Usage: 607 k	3.787 ms	145286.151 ms	9630	3
14.	→ Subquery Scan (actual=145272.38..14...	1.406 ms	145282.365 ms	9630	3
15.	→ Aggregate (actual=145272.374..1... Buckets: Batches: Memory Usage: 1681	108352.213 ms	145280.959 ms	9630	3
16.	→ Seq Scan on Messages as M...	36928.746 ms	36928.746 ms	59474190	3