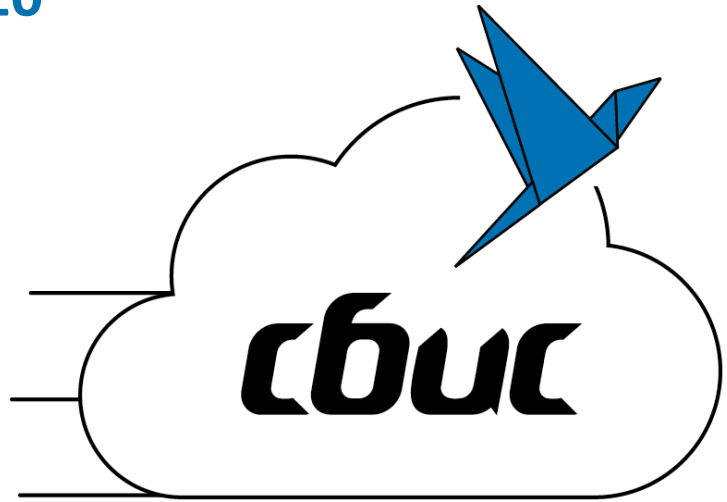
PGConf.Russia 2020



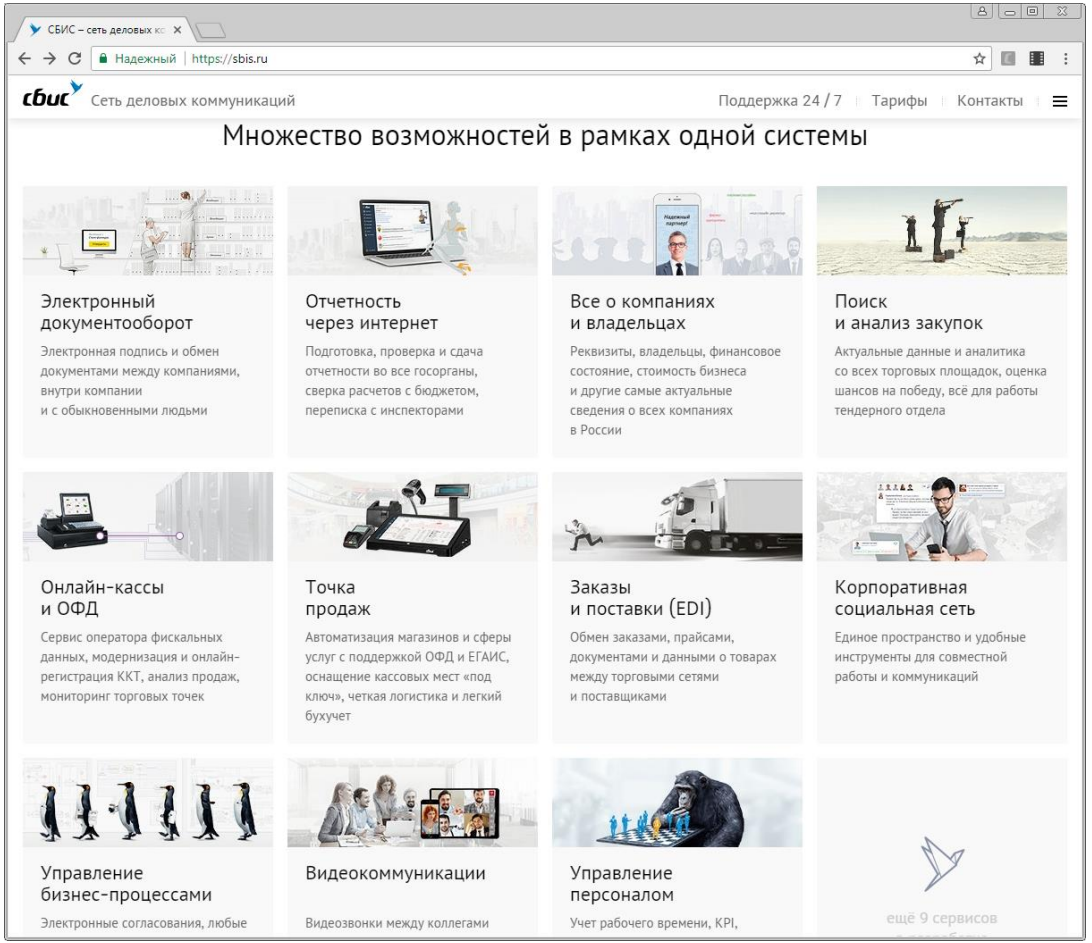
План + запрос = ♡

когда анализ запроса в радость

Кирилл Боровиков / Компания «Тензор», технический директор / explain.tensor.ru, sbis.ru

«Тензор» – это СБИС миллионы клиентов

- 100+ проектов
- 10+ центров разработки
- больше 1000 сотрудников



СБИС – data-centric application

Классика: «Почему тут выполнялось долго?»

- неэффективный алгоритм
- неактуальная статистика
- «затык» по ресурсам (процессор, диск, память)

«Нам нужен план!»


ПЛАН-ЗАКОН,
ВЫПОЛНЕНИЕ-ДОЛГ,
ПЕРЕВЫПОЛНЕНИЕ-ЧЕСТЬ!

План запроса

Дерево в текстовом виде

- каждый элемент – одна из операций

получение данных, построение битовых карт, обработка данных, операция над множествами, соединение, вложенный запрос, …

- выполнение плана – обход дерева

План запроса

```
Query Text: explain (analyze, buffers, costs off)
SELECT * FROM pg_class WHERE (oid, relname) = (
  SELECT oid, relname FROM pg_class WHERE relkind = 'r' LIMIT 1
);


Index Scan using pg_class_relname_nsp_index on pg_class (actual time=0.049..0.050 rows=1 loops=1)
  Index Cond: (relname = $1)
  Filter: (oid = $0)
  Buffers: shared hit=4
  InitPlan 1 (returns $0,$1)
    ->  Limit (actual time=0.019..0.020 rows=1 loops=1)
          Buffers: shared hit=1
          ->  Seq Scan on pg_class pg_class_1 (actual time=0.015..0.015 rows=1 loops=1)
                Filter: (relkind = 'r'::"char")
                Rows Removed by Filter: 5
                Buffers: shared hit=1
```

План запроса

План запроса

План текстом – ненаглядно:

- в узле – сумма по ресурсам поддерева
- время необходимо умножать на loops
- … так кто же «самое слабое звено»?

План запроса

План текстом – ненаглядно:

- в узле – сумма по ресурсам поддерева
- время необходимо умножать на loops
- … так кто же «самое слабое звено»?

«Понимание плана – это искусство, и чтобы овладеть им, нужен определённый опыт, …»

Создаем инструмент

explain.sbis.ru (https://pgconf.ru/2018/107220)

- автоматический разбор логов
- сводный паттерн-анализ
- наглядность планов
- непубличный сервис

Создаем инструмент

explain.~~sbis~~.ru → explain.tensor.ru

- публичный сервис
- наглядность планов
- структурные подсказки
- построчный профайлер запроса

Наглядность

explain.tensor.ru – оригинал плана

```
Seq Scan on pg_class  (cost=0.00..623.40 rows=7208
width=536) (actual time=0.009..1.304 rows=6609 loops=1)

  Buffers: shared hit=263

Planning Time: 0.108 ms

Execution Time: 1.800 ms
```

Наглядность

explain.tensor.ru – атрибуты узлов

#	node, ms	tree, ms	rows	ratio	RRbF	node	sh.ht
		3.393	1 440		5 169	итоговые результаты	263
0	3.393	3.393	1 440	1.1↑	5 169	Seq Scan on pg_class (cost=0.00..641.42 rows=1571 width=536) (actual time=0.018..3.393 rows=1440 loops=1) Filter: (relkind = 'r'::"char") Rows Removed by Filter: 5169 Buffers: shared hit=263	263

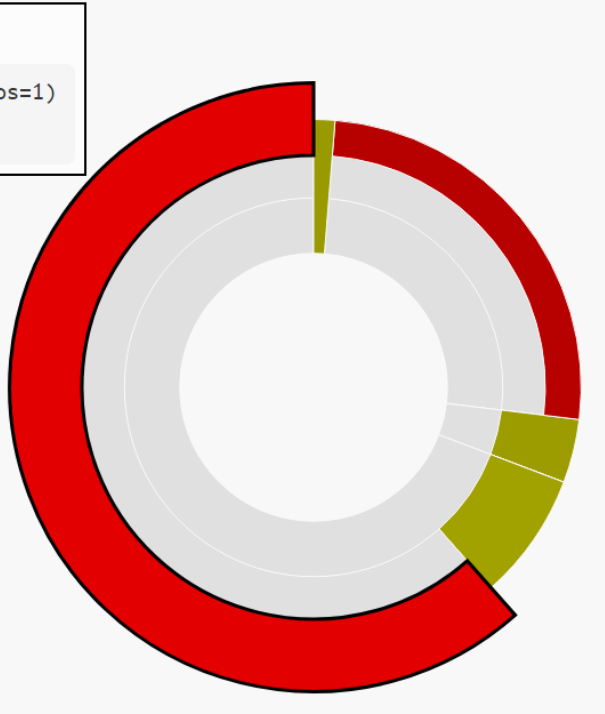
Наглядность

explain.tensor.ru – шаблон и pie chart

#6 **0.048**ms (61.5%), rows=101, loops=1

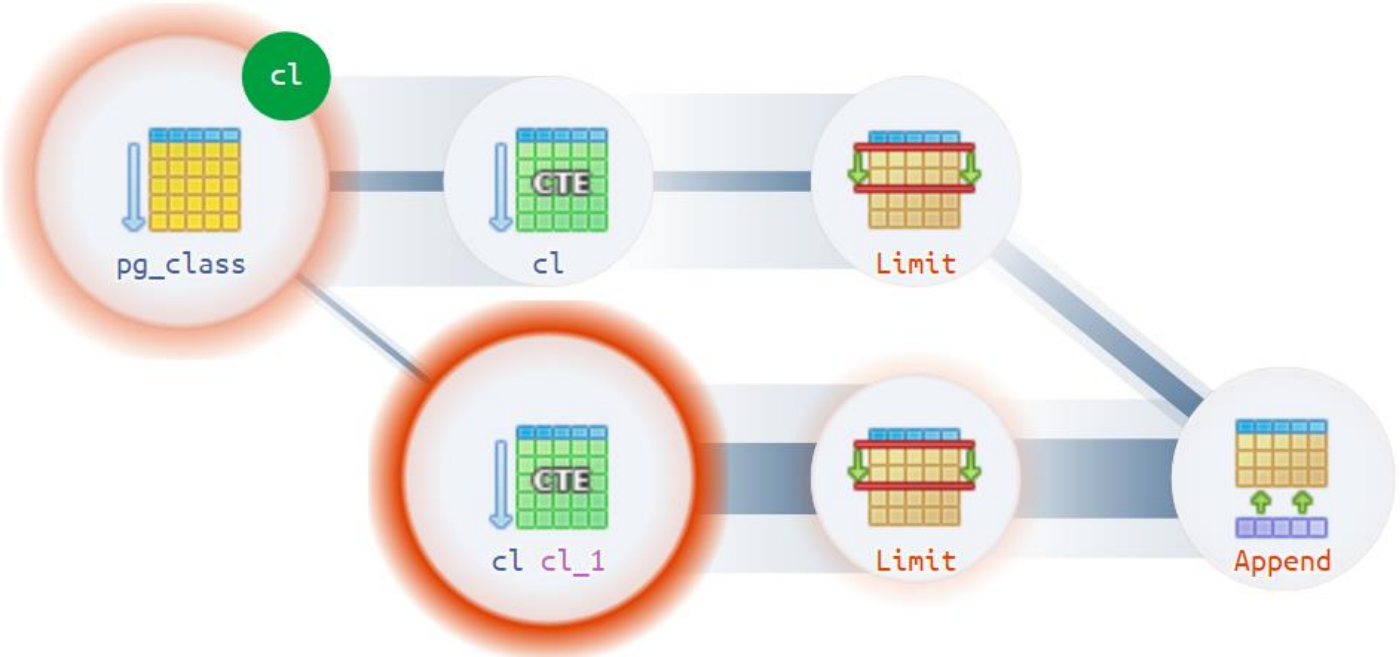
```
CTE Scan on cl cl_1  (cost=0.00..720.80 rows=7208 width=233) (actual time=0.000..0.053 rows=101 loops=1)
  Buffers: shared hit=2
```

#						
0	0.001	0.078	2	----		Append
1						CTE cl
2	0.020	0.020	101	71.4↑	->	Seq Scan on pg_class
3	0.003	0.018	1	----	->	Limit
4		0.015	1	7 208.0↑	->	CTE Scan on cl
5	0.006	0.059	1	----	->	Limit
6	0.048	0.053	101	71.4↑	->	CTE Scan on cl cl_1



Наглядность

explain.tensor.ru – диаграмма выполнения



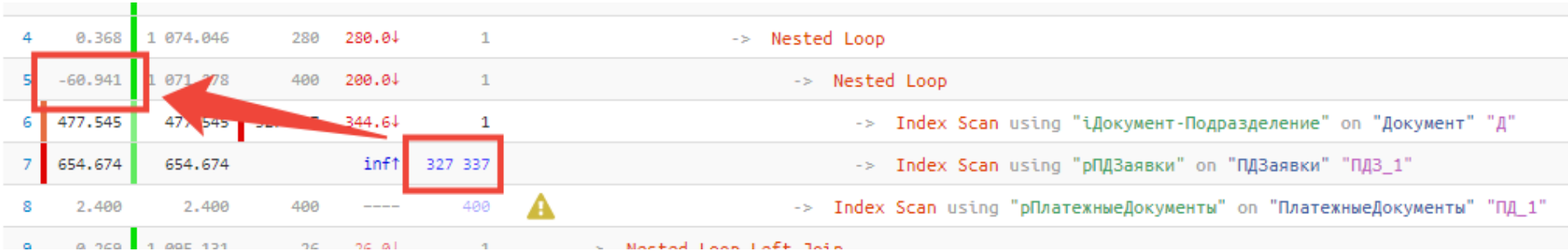

STRANGER
THINGS

Очень странные планы

Очень странные планы

Потерянные микросекунды

- … и даже [милли]секунды!



Очень странные планы

Потерянные микросекунды

```
SELECT 1::double precision - 0.999;
```

⇒ **0.001**

```
SELECT 1::double precision - 0.9999;
```

⇒ **9.999999999989e-05** <> 1e-04

Очень странные планы

Потерянные микросекунды

- по циклам узла – умножение проблем

  1µs (±0.5µs) × 1000 = 0.5ms .. 1.5ms

- по иерархии узлов – сложение проблем

  0.7µs (1µs) + 0.7µs (1µs) = 1.4µs (1µs)

  1.4µs (1µs) + 1.4µs (1µs) = 2.8µs (3µs)

Очень странные планы

Очень странные планы

Вложенные подзапросы – к какому узлу отнести?

#	node, ms	tree, ms	rows	ratio	RRbF	node	sh.ht
		6.448	1 468		5 269	итоговые результаты	4 593
0	3.181	6.448	1 468			Index Scan using pg_class_oid_index on pg_class (actual time=3.851..6.448 rows=1468 loops=1) Index Cond: (oid = ANY ($0)) Buffers: shared hit=4593	4 330
1						InitPlan 1 (returns $0)	
2	3.267	3.267	1 468		5 269	-> Seq Scan on pg_class pg_class_1 (actual time=0.013..3.267 rows=1468 loops=1) Filter: (relkind = 'r'::"char") Rows Removed by Filter: 5269 Buffers: shared hit=263	263

Очень странные планы

Вложенные подзапросы – а если мультирезультат?..

#	node, ms	tree, ms	rows	ratio	RRbF	node	sh.ht
		0.050	1		5	итоговые результаты	4
0	0.030	0.050	1			Index Scan using pg_class_relname_nsp_index on pg_class (actual time=0.049..0.050 rows=1 loops=1) Index Cond: (relname = $1) Filter: (oid = $0) Buffers: shared hit=4	3
1						InitPlan 1 (returns $0,$1)	
2	0.005	0.020	1			-> Limit (actual time=0.019..0.020 rows=1 loops=1) Buffers: shared hit=1	
3	0.015	0.015	1		5	-> Seq Scan on pg_class pg_class_1 (actual time=0.015..0.015 rows=1 loops=1) Filter: (relkind = 'r'::"char") Rows Removed by Filter: 5 Buffers: shared hit=1	1

Очень странные планы

Вложенные подзапросы – … или еще сложнее?

```
->  Subquery Scan on "*SELECT* 2"  (cost=0.58..2.21 rows=1 width=91) (actual time=0.045..15.747 rows=20 loops=1)
      Buffers: shared hit=47 read=21
      ->  Limit  (cost=0.58..2.20 rows=1 width=95) (actual time=0.045..15.739 rows=20 loops=1)
            Buffers: shared hit=47 read=21
            InitPlan 4 (returns $3,$4,$5)
              ->  CTE Scan on default_cursor  (cost=0.00..0.02 rows=1 width=28) (actual time=0.009..0.009 rows=1 loops=1)
            InitPlan 5 (returns $6)
              ->  CTE Scan on first_item first_item_1  (cost=0.00..0.00 rows=1 width=0) (actual time=0.001..0.001 rows=0 loops=1)
            ->  Index Scan using "iТемаАдресат-Тема-Акивность" on "ТемаАдресат"  (cost=0.56..3.79 rows=2 width=95) (actual time=
0.043..15.732 rows=20 loops=1)
                  Index Cond: (("Тема" = 'f718d8f4-04bc-42e2-b2e0-4e9964638f36'::uuid) AND (ROW((2147483647 - "ВсегоИсходящих"),
"ДатаВремя", "ПерсонаАдресат") > ROW($3, $4, $5)))
                  Filter: ("Отключен" IS NULL)
                  Rows Removed by Filter: 44
                  Buffers: shared hit=47 read=21
```

Очень странные планы

Очень странные планы

Повторные CTE Scan – «клоны» ресурсов

#	node, ms	tree, ms	rows	ratio	node	sh.ht
		0.078	2		итоговые результаты	3
0	0.001	0.078	2	----	Append (cost=623.40..633.75 rows=2 width=233) (actual time=0.018..0.078 rows=2 loops=1) Buffers: shared hit=3	
1					CTE cl	
2	0.020	0.020	101	71.4↑	-> Seq Scan on pg_class (cost=0.00..623.40 rows=7208 width=536) (actual time=0.010..0.020 rows=101 loops=1) Buffers: shared hit=3	3
3	0.003	0.018	1	----	-> Limit (cost=0.00..0.10 rows=1 width=233) (actual time=0.017..0.018 rows=1 loops=1) Buffers: shared hit=1	
4		0.015	1	7 208.0↑	-> CTE Scan on cl (cost=0.00..720.80 rows=7208 width=233) (actual time=0.015..0.015 rows=1 loops=1) Buffers: shared hit=1	
5	0.006	0.059	1	----	-> Limit (cost=10.00..10.10 rows=1 width=233) (actual time=0.059..0.059 rows=1 loops=1) Buffers: shared hit=2	
6	0.048	0.053	101	71.4↑	-> CTE Scan on cl cl_1 (cost=0.00..720.80 rows=7208 width=233) (actual time=0.000..0.053 rows=101 loops=1) Buffers: shared hit=2	

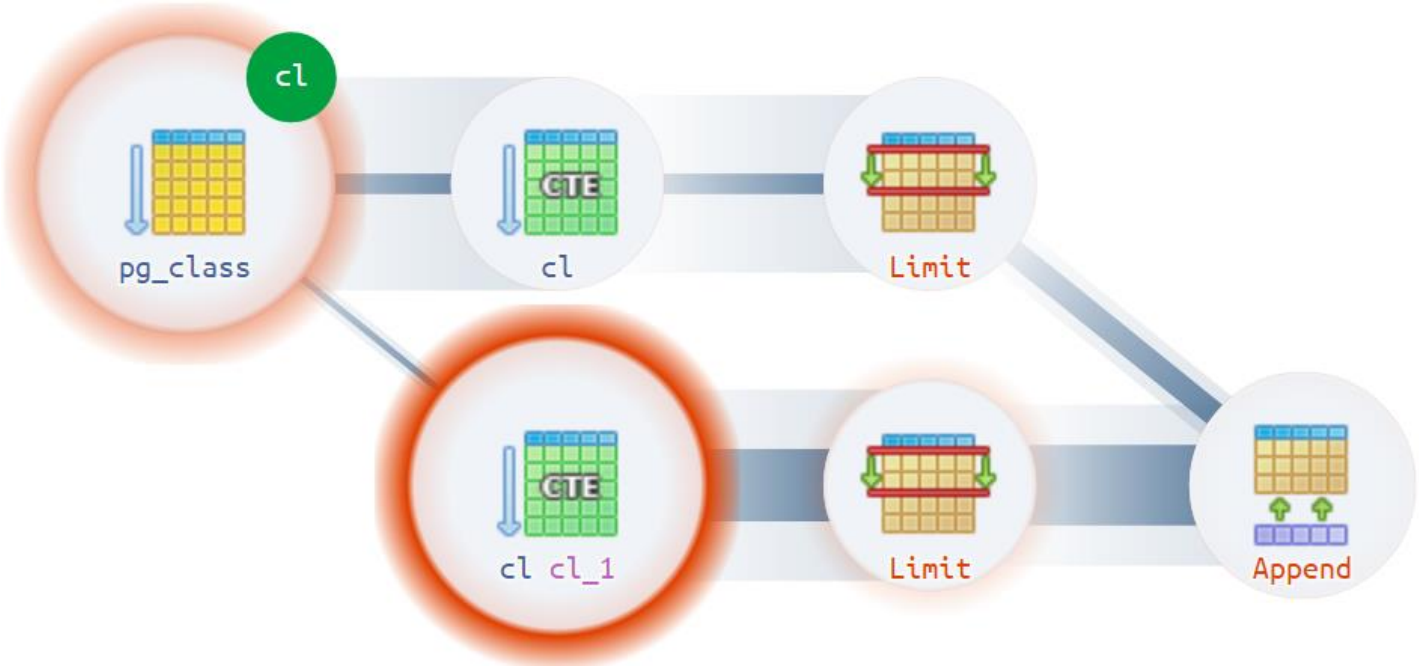
Очень странные планы

Повторные CTE Scan – «клоны» ресурсов

```
WITH cl AS (
  TABLE pg_class
)
  (TABLE cl LIMIT 1)
UNION ALL
  (TABLE cl LIMIT 1 OFFSET 100);
```

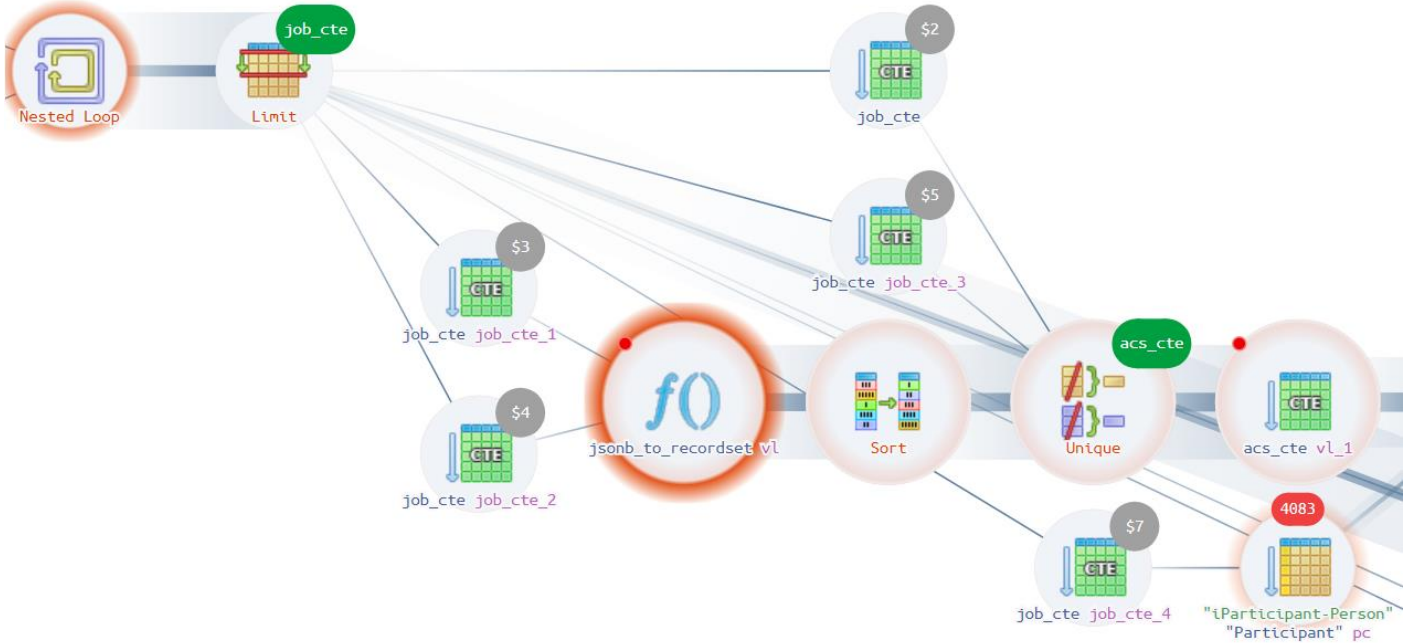
Очень странные планы

Повторные CTE Scan – простая схема

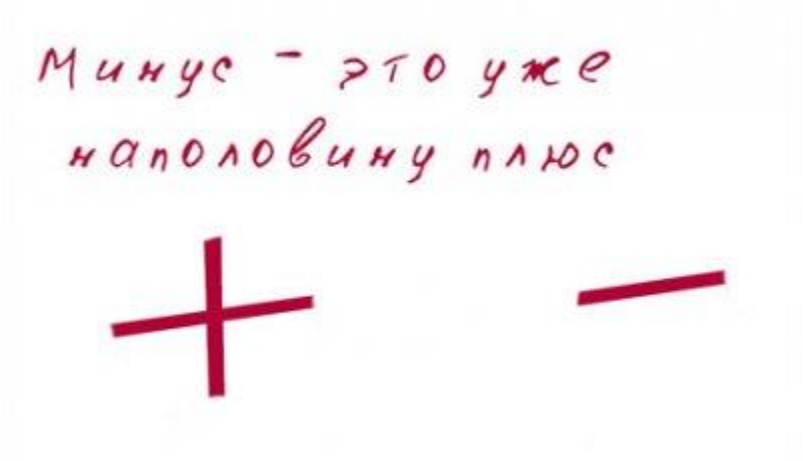


Очень странные планы

Повторные CTE Scan – совсем не простая схема



Очень странные планы



Очень странные планы

Недочитанные wCTE – «минусы» по ресурсам

#	node, ms	tree, ms	rows	ratio	node	sh.ht	sh.dr
		0.065	1		итоговые результаты	3	2
0	0.001	0.065	1		Limit (actual time=0.065..0.065 rows=1 loops=1) Buffers: shared hit=3 dirtied=2		
1					CTE ins		
2	0.093	0.101	10		-> Insert on tbl (actual time=0.063..0.101 rows=10 loops=1) Buffers: shared hit=30 dirtied=2	30	2
3	0.008	0.008	10		-> Result (actual time=0.003..0.008 rows=10 loops=1)		
4	-0.037	0.064	1		-> CTE Scan on ins (actual time=0.064..0.064 rows=1 loops=1) Buffers: shared hit=3 dirtied=2	-27	

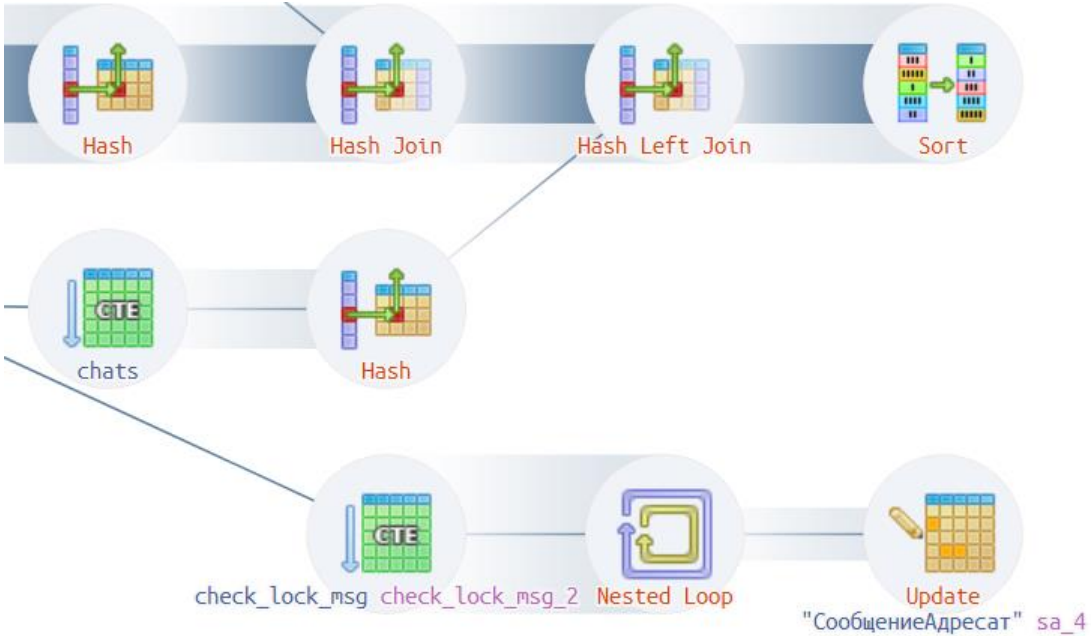
Очень странные планы

Недочитанные wCTE – «минусы» по ресурсам

```
WITH ins AS (
  INSERT INTO tbl(val)
  SELECT generate_series(1, 10)
  RETURNING *
)
TABLE ins LIMIT 1;
```

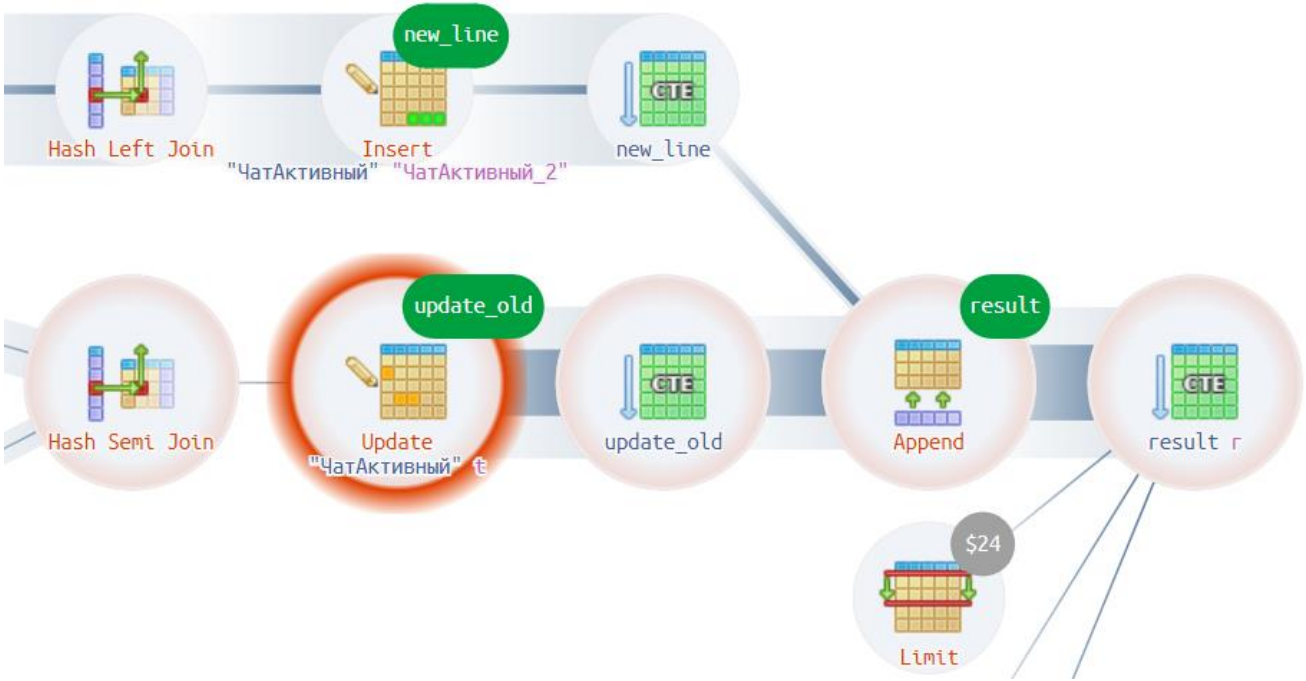
Очень странные планы

Недочитанные wCTE – сам себе «корень»



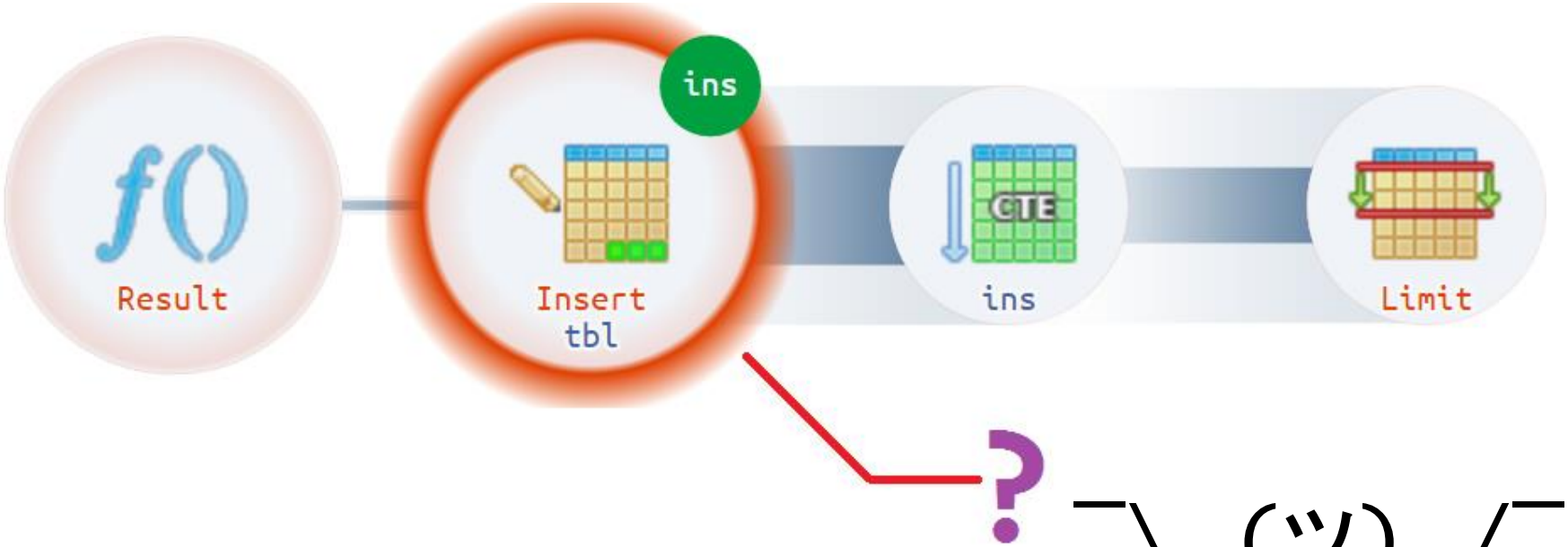
Очень странные планы

Недочитанные wCTE – прочитано все



Очень странные планы

Недочитанные wCTE – прочитано, но не все



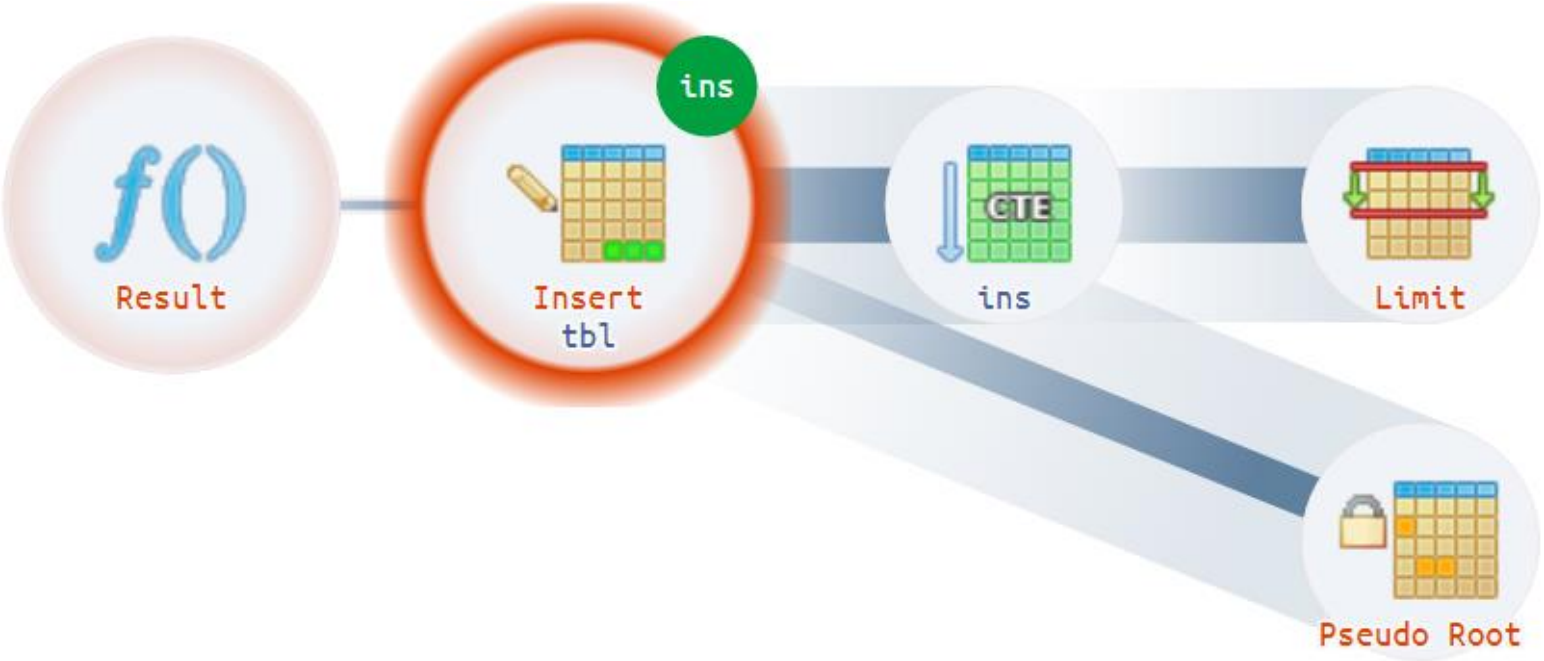
Очень странные планы

Недочитанные wCTE – исправляем анализ

#	node, ms	tree, ms	rows	ratio	node	sh.ht	sh.dr
		0.102	1		итоговые результаты	30	2
0	0.001	0.065	1		Limit (actual time=0.065..0.065 rows=1 loops=1) Buffers: shared hit=3 dirtied=2		
1					CTE ins		
2	0.093	0.101	10		-> Insert on tbl (actual time=0.063..0.101 rows=10 loops=1) Buffers: shared hit=30 dirtied=2	30	2
3	0.008	0.008	10		-> Result (actual time=0.003..0.008 rows=10 loops=1)		
4		0.064	1		-> CTE Scan on ins (actual time=0.064..0.064 rows=1 loops=1) Buffers: shared hit=3 dirtied=2		

Очень странные планы

Недочитанные wCTE – нам поможет «псевдокорень»!



Очень странные планы

Parallel / Gather – …

#	node, ms	tree, ms	rows	ratio	loops	node	sh.ht
		36.706	64			итоговые результаты	1 539
0	-55.959	36.706	64		1	Finalize GroupAggregate (actual time=36.644..36.706 rows=64 loops=1) Group Key: ((trunc((date_part('epoch'::text, (error_20191007.ts - ('2019-10-07'::date)::timestamp with time zone)) / '600'::double precision)))::integer) Buffers: shared hit=1539	- 556
1	62.740	92.665	119		1	-> Gather Merge (actual time=36.627..92.665 rows=119 loops=1) Workers Planned: 1 Workers Launched: 1 Buffers: shared hit=2095	
2	0.063	29.925	120		2	-> Sort (actual time=29.919..29.925 rows=60 loops=2) Sort Key: ((trunc((date_part('epoch'::text, (error_20191007.ts - ('2019-10-07'::date)::timestamp with time zone)) / '600'::double precision)))::integer) Sort Method: quicksort Memory: 28kB Worker 0: Sort Method: quicksort Memory: 27kB Buffers: shared hit=2095	7
3	6.073	29.862	120		2	-> Partial HashAggregate (actual time=29.848..29.862 rows=60 loops=2) Group Key: ((trunc((date_part('epoch'::text, (error_20191007.ts - ('2019-10-07'::date)::timestamp with time zone)) / '600'::double precision)))::integer) Buffers: shared hit=2088	
4	3.129	23.789	55 044		2	-> Parallel Append (actual time=2.236..23.789 rows=27522 loops=2) Buffers: shared hit=2088	
5	16.628	20.660	55 044		2	-> Parallel Bitmap Heap Scan on error_20191007 (actual time=2.235..20.660 rows=27522 loops=2) Recheck Cond: (errmsg = 3854) Filter: (dt = '2019-10-07'::date) Heap Blocks: exact=1180 Buffers: shared hit=2088	1 729
6	4.032	4.032	55 045		1	-> Bitmap Index Scan on error_20191007_errmsg_ts_idx (actual time=4.032..4.032 rows=55045 loops=1) Index Cond: (errmsg = 3854) Buffers: shared hit=359	359

Очень странные планы

Parallel / Gather – …

main process

```
-> Finalize *
     Buffers: 1500
```

```
   -> Gather *
        Buffers: 2000
```

```
      -> Partial *
           Buffers: 1500
```

worker

```
-> Partial *
     Buffers: 500
```

¯_(ツ)_/¯


Greetings, I am your Mentat Cyril.

Структурные подсказки

```
-> Limit
   -> Sort
      -> Index Only Scan using "iДокумент-ТипДокументаЛицо1" on "Документ" "Документ_1"
Рекомендации:
- создайте индекс, включающий поля, используемые для сортировки
```

Структурные подсказки

BitmapAnd

```
-> Bitmap Heap Scan on X

   -> BitmapAnd

      -> Bitmap Index Scan on X_idx1

      -> Bitmap Index Scan on X_idx2
```

- составной индекс по двум наборам полей

Структурные подсказки

BitmapOr

```
-> Bitmap Heap Scan on X

   -> BitmapOr

      -> Bitmap Index Scan on X_idx1

      -> Bitmap Index Scan on X_idx2
```

- UNION [ALL] между подзапросами

Структурные подсказки

Слишком много лишнего

```
-> Seq Scan | Index [Only] Scan [Backward]

   && 5 × rows < RRbF -- отфильтровано >80% прочитанного

   && loops × RRbF > 100
```

WHERE-индекс с условием фильтра

Структурные подсказки

Индексная «недофильтрация»

```
-> Index [Only] Scan [Backward]

   && 2 × rows <= RRbF -- отфильтровано >60% прочитанного
```

WHERE-индекс с условием фильтра

Структурные подсказки

Индексная «недосортировка»

```
-> Limit

   -> Sort

      -> Index [Only] Scan [Backward]
```

- расширить индекс полями сортировки

Структурные подсказки

«Редкая птица»

```
-> Seq Scan | Index [Only] Scan [Backward]

   && loops × (rows + RRbF) < (shared hit + shared read) × 8

      -- прочитано больше 1KB на каждую запись

   && shared hit + shared read > 64
```

- VACUUM на разреженной таблице

Структурные подсказки

CTE × CTE

```
-> CTE Scan

   && loops > 10

   && loops × (rows + RRbF) > 10000

      -- слишком большое декартово произведение CTE
```

- материализация в hstore/json-словарь

Структурные подсказки

«Что-то пошло не так...»

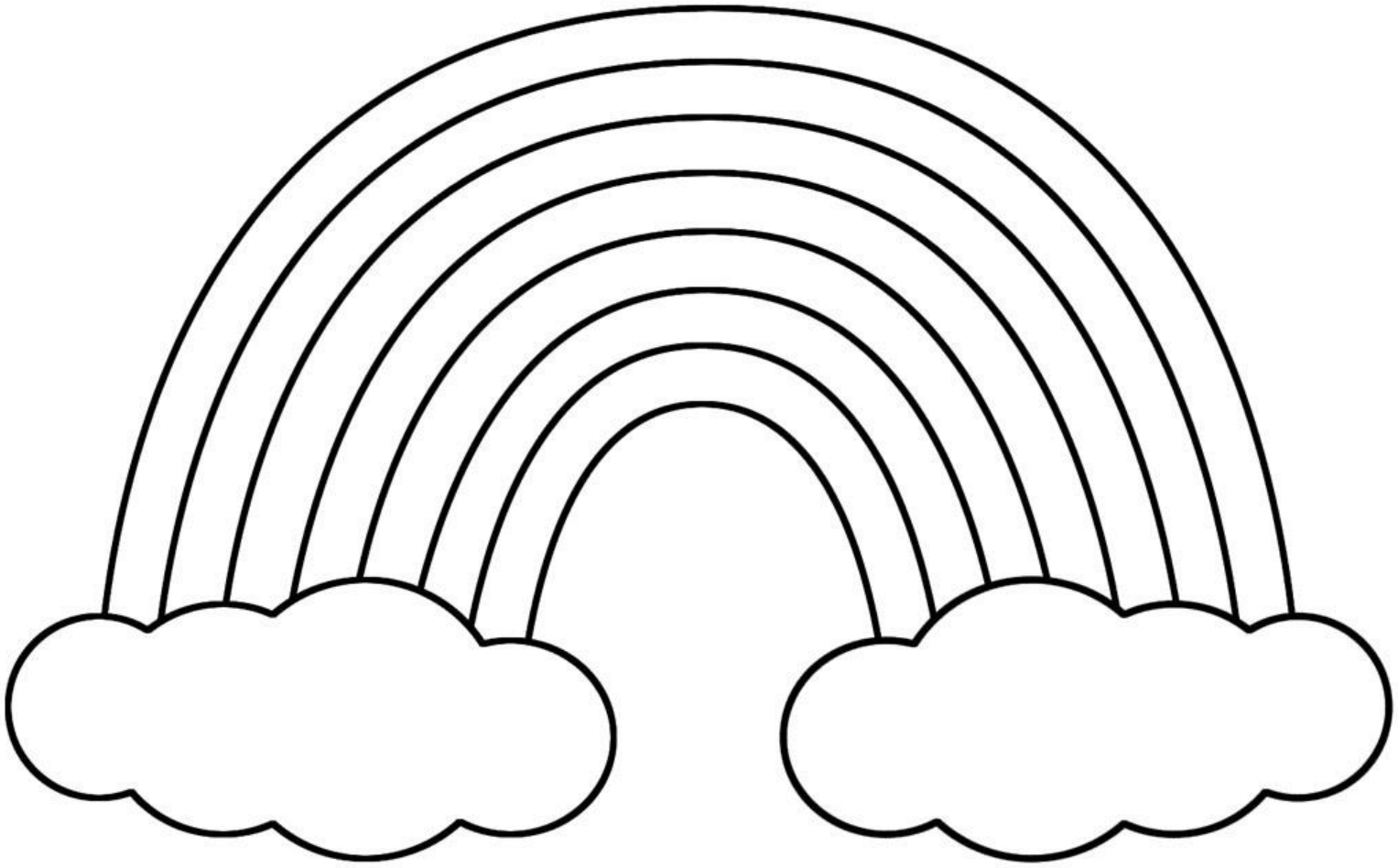
```
-> *

   && (shared hit / 8K) + (shared read / 1K) < time / 1000

      RAM hit = 64MB/s, HDD read = 8MB/s

   && time > 100ms -- читали мало, но слишком долго
```

- блокировка или «затык» на CPU/RAM/HDD



Query Profiler

```
Query Text: explain (analyze, buffers, costs off)
SELECT * FROM pg_class WHERE (oid, relname) = (
  SELECT oid, relname FROM pg_class WHERE relkind = 'r' LIMIT 1
);

Index Scan using pg_class_relname_nsp_index on pg_class (actual time=0.049..0.050 rows=1 loops=1)
  Index Cond: (relname = $1)
  Filter: (oid = $0)
  Buffers: shared hit=4
  InitPlan 1 (returns $0,$1)
    ->  Limit (actual time=0.019..0.020 rows=1 loops=1)
          Buffers: shared hit=1
          ->  Seq Scan on pg_class pg_class_1 (actual time=0.015..0.015 rows=1 loops=1)
                Filter: (relkind = 'r'::"char")
                Rows Removed by Filter: 5
                Buffers: shared hit=1
```

Query Profiler

```
EXPLAIN (ANALYZE, BUFFERS, COSTS off)
SELECT
  *
FROM
4.755ms  pg_class
WHERE
  (oid, relr
    SELECT
      oid
    , relname
    FROM
3.063ms    pg_class
    WHERE
      relkind = 'r'
  );
```

#1 **4.755**ms (43.6%), rows=6737, loops=1

Buffers узла (263): shared hit=263

```
Seq Scan on pg_class  (cost=0.00..599.85 rows=6737 width=540) (actual time=0.013..4.755 rows=6737 loops=1)
  Buffers: shared hit=263
```

Query Profiler

Query Profiler

Разбираем запрос

- NodeJS
- https://github.com/MGorkov/node-pgparser
- ← https://github.com/zhm/pg-query-parser
- ← https://github.com/lfittl/libpg_query

Query Profiler

Разбираем запрос – дерево

```
console.log(pgparser('select 1'))
```

```
[ { RawStmt:
     { stmt:
        { SelectStmt:
           { targetList:
              [ { ResTarget:
                   { val:
                      { A_Const:
                         { val:
                            { Integer:
                               { ival: 1 }
                            }, location: 7 } },
                  location: 7 } } ],
             op: 0 } } } } ]
```

Query Profiler

Собираем все обратно – раскраска

```
EXPLAIN (ANALYZE, BUFFERS, COSTS off)
SELECT
  *
FROM
  pg_class
WHERE
  (oid, relname) IN (
    SELECT
      oid
    , relname
    FROM
      pg_class
    WHERE
      relkind = 'r'
  );
```

Query Profiler

Query Profiler

Собираем все обратно – copy & paste

- просто скопировать текст:

```
SELECT 'const', $1::text;
```

Query Profiler

Собираем все обратно – copy & paste

- подставить значения параметров:

```
SELECT 'const', 'param'::text;
```

Query Profiler

Собираем все обратно – copy & paste

- собрать PREPARED на выполнение:

```
DEALLOCATE ALL; PREPARE q(text) AS

SELECT 'const', $1::text;

EXECUTE q('param'::text);
```

Query Profiler

Query Profiler

Совмещаем с планом

```
WITH cl AS (
  TABLE pg_class
)
  (
    TABLE cl LIMIT 1
  )
UNION ALL
  (
    TABLE cl LIMIT 1 OFFSET 100
  );
```

```
Append
  CTE cl
    ->  Seq Scan on pg_class
  ->  Limit
        ->  CTE Scan on cl
  ->  Limit
        ->  CTE Scan on cl cl_1
```

Query Profiler

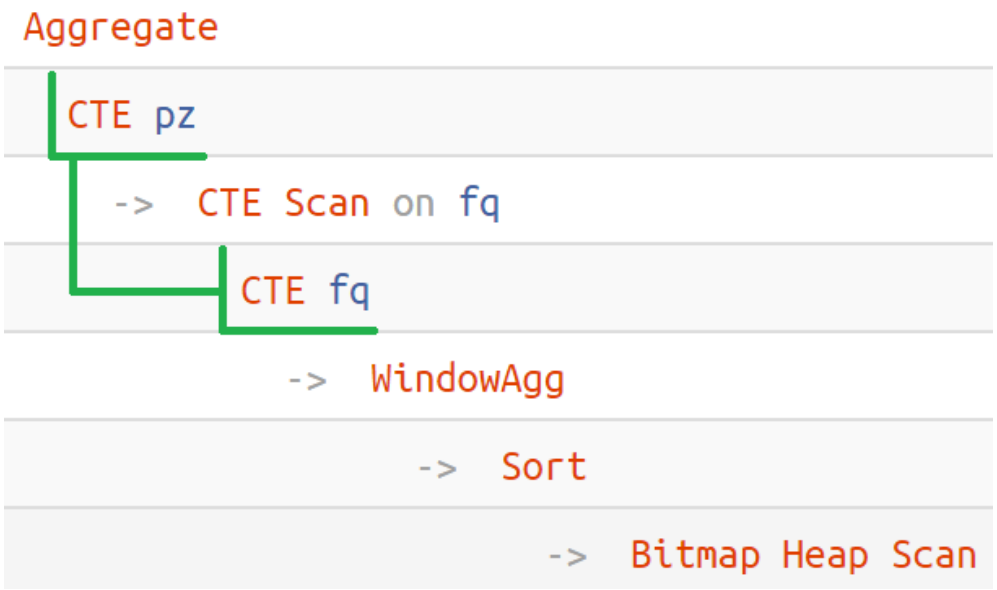
Совмещаем с планом – CTE-сегменты

```
WITH cl AS (
  TABLE pg_class
)
  (
    TABLE cl LIMIT 1
  )
UNION ALL
  (
    TABLE cl LIMIT 1 OFFSET 100
  );
```

```
Append
  CTE cl
    ->  Seq Scan on pg_class
  ->  Limit
        ->  CTE Scan on cl
  ->  Limit
        ->  CTE Scan on cl cl_1
```

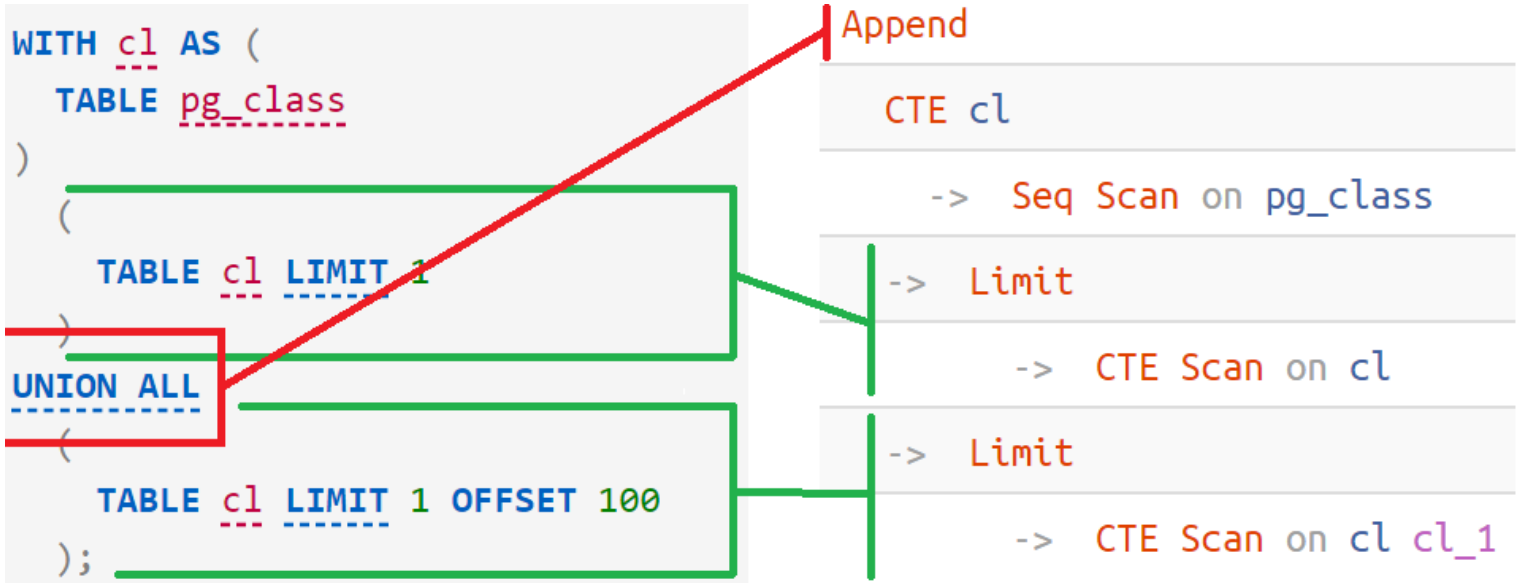
Query Profiler

Совмещаем с планом – CTE-сегменты *



Query Profiler

Совмещаем с планом – UNION-сегменты



Query Profiler

Совмещаем с планом – UNION-сегменты

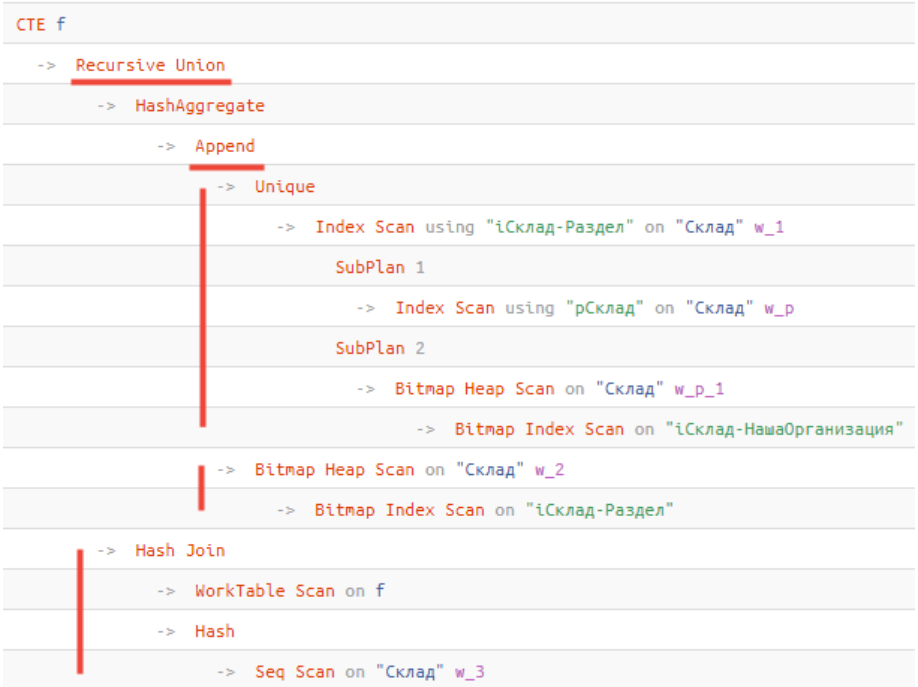
```
      WITH cl AS (
.025ms   TABLE pg_class
      )
        (
.002ms    TABLE c
        )
.002ms  UNION ALL
        (
.052ms    TABLE c
        );
```

#0 **0.002**ms (2.5%), rows=2, loops=1

```
Append  (cost=586.05..596.40 rows=2 width=233) (actual time=0.019..0.081 rows=2 loops=1)
  Buffers: shared hit=3
```

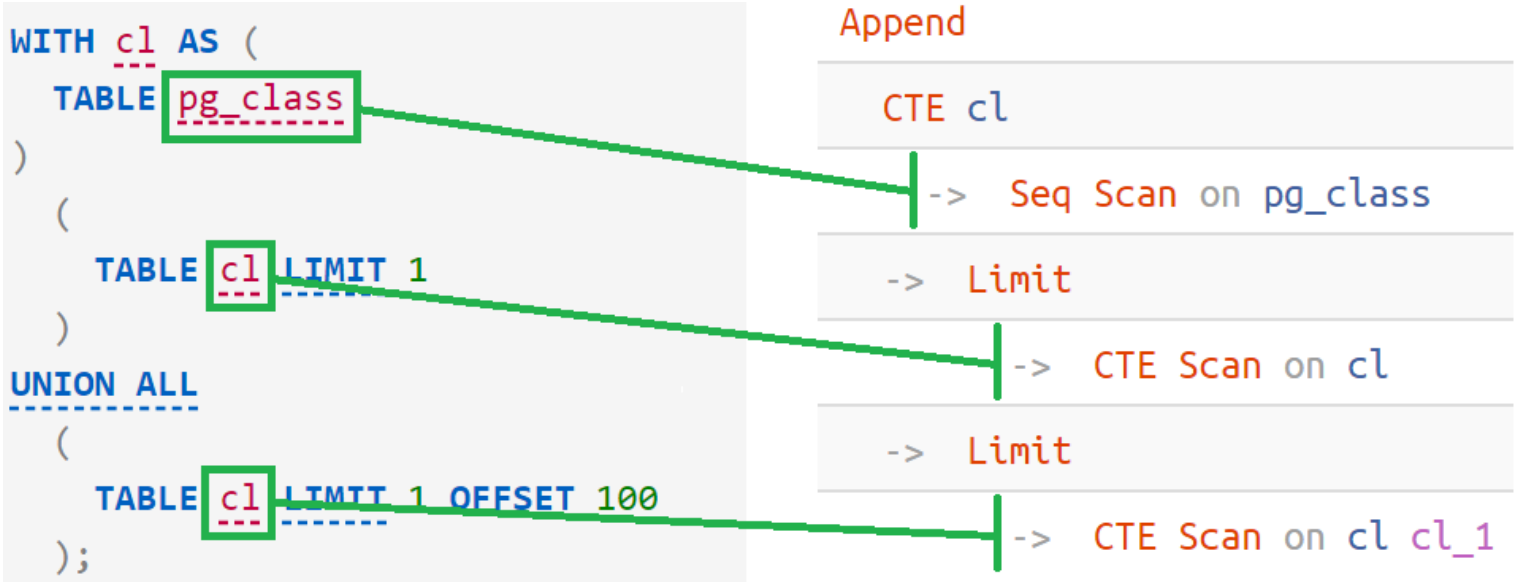
Query Profiler

Совмещаем с планом – UNION-сегменты *



Query Profiler

Совмещаем с планом – чтение-запись данных



Query Profiler

Совмещаем с планом – чтение-запись данных *

```
WITH cl AS (
  TABLE pg_class
)
  TABLE cl
UNION ALL
  TABLE cl; -- алиасов нет

Append
  CTE cl
    ->  Seq Scan on pg_class
  ->  CTE Scan on cl
  ->  CTE Scan on cl cl_1 -- «номерные» алиасы
```

Query Profiler

Совмещаем с планом – чтение-запись данных *

```
TABLE megatable; -- секционирование

[Merge] Append
  ->  Seq Scan on megatable_001
  ->  Seq Scan on megatable_002
  ->  Seq Scan on archive
```

Query Profiler

Совмещаем с планом – чтение-запись данных *

- Values Scan ⇒ … `FROM (VALUES …)`
- Result ⇒ нет `FROM`, или `One-Time Filter`
- Function Scan ⇒ … `FROM SRF()`
- Subquery Scan / InitPlan / SubPlan ⇒ `???`

Query Profiler

Совмещаем с планом – чтение-запись данных *

```
SELECT -- VALUES
  *
FROM
  (VALUES(1),(2)) x
, (VALUES(3),(4),(5)) y;

Nested Loop (... rows=6 loops=1)
  ->  Values Scan on "*VALUES*_1" (... rows=3 loops=1)
  ->  Materialize (... rows=2 loops=3)
        ->  Values Scan on "*VALUES*" (... rows=2 loops=1)
```

Query Profiler

Совмещаем с планом – обработка данных

```
WITH cl AS (
  TABLE pg_class
)
  (
    TABLE cl LIMIT 1
  )
UNION ALL
  (
    TABLE cl LIMIT 1 OFFSET 100
  );
```

```
Append
  CTE cl
    ->  Seq Scan on pg_class
  ->  Limit
        ->  CTE Scan on cl
  ->  Limit
        ->  CTE Scan on cl cl_1
```

Query Profiler

Совмещаем с планом – обработка данных

- Limit ⇒ **LIMIT**, Sort ⇒ **ORDER BY**
- *Aggregate ⇒ **GROUP BY**
- WindowAgg ⇒ **WINDOW**
- HashAggregate / Unique ⇒ **DISTINCT**

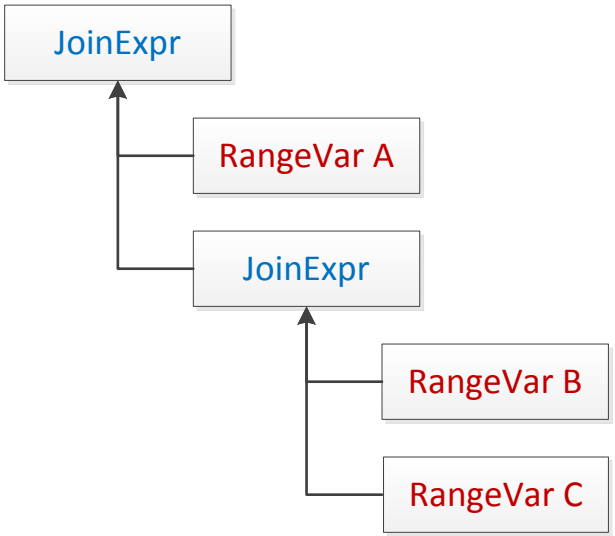
Query Profiler

Совмещаем с планом – JOIN

- JoinExpr : { larg , rarg }
- два потомка `*Loop` / `*Join`
- ⇒ совпала пара 0:larg/1:rarg || 0:rarg/1:larg
- ⇒ повторить, пока есть совпадения пар

Query Profiler

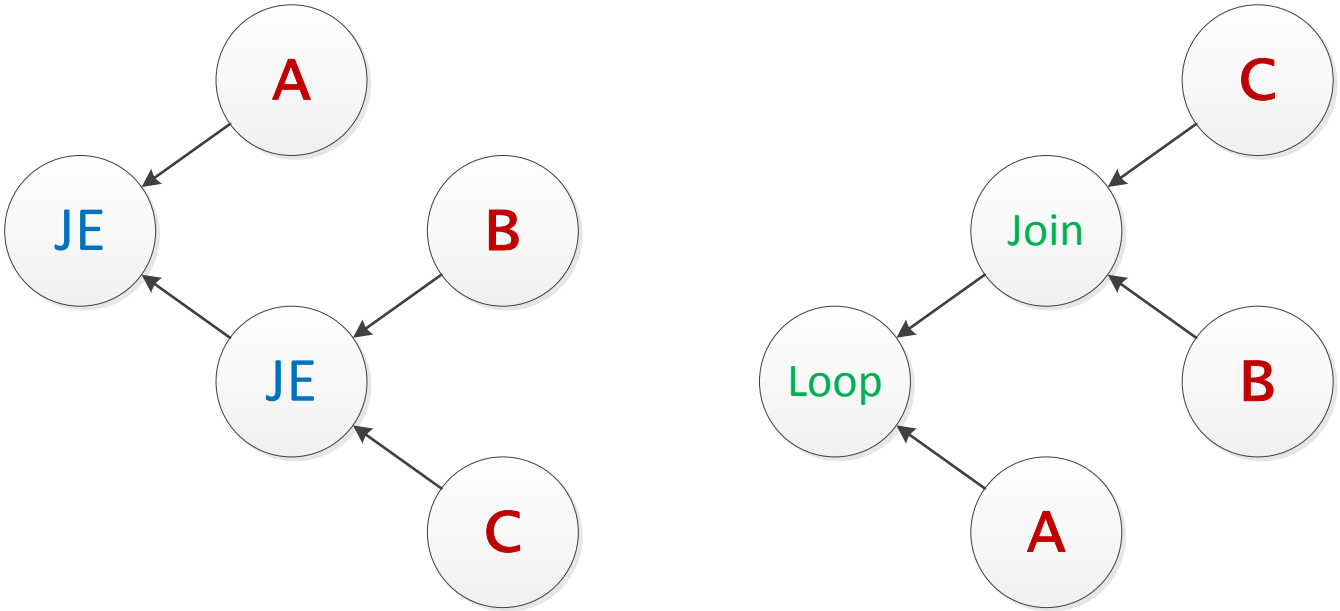
Совмещаем с планом – JOIN




```
-> *Loop
    -> *Join
        -> *Scan on C
        -> *Scan on B
    -> *Scan on A
```

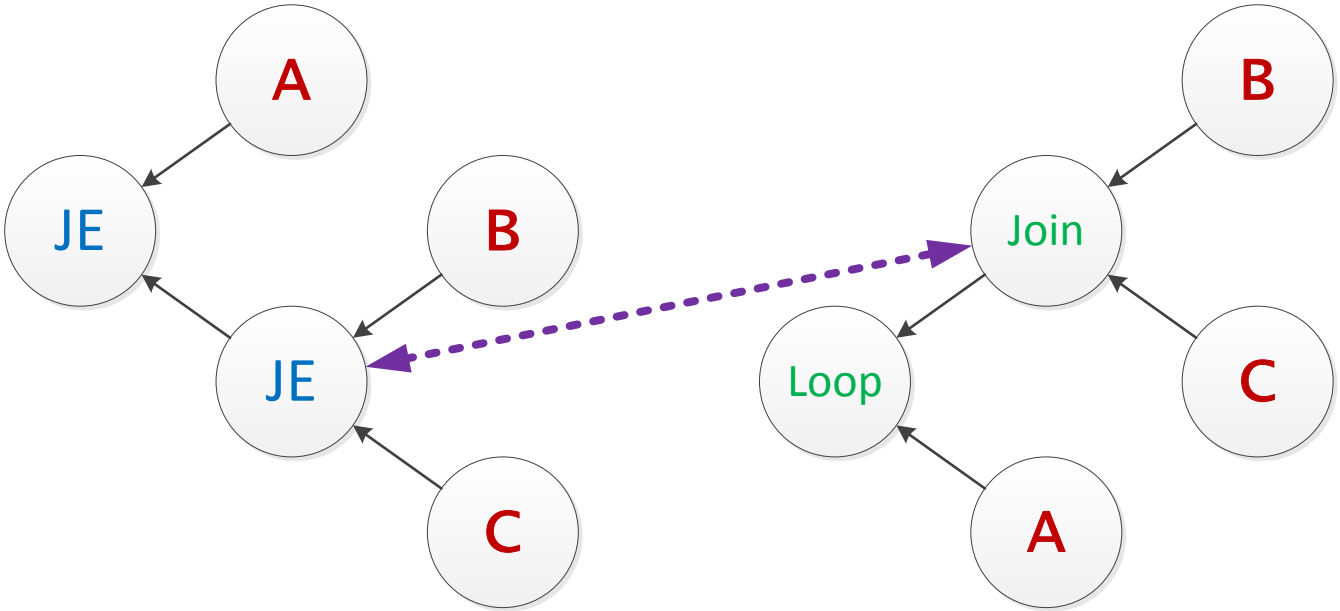
Query Profiler

Совмещаем с планом – JOIN



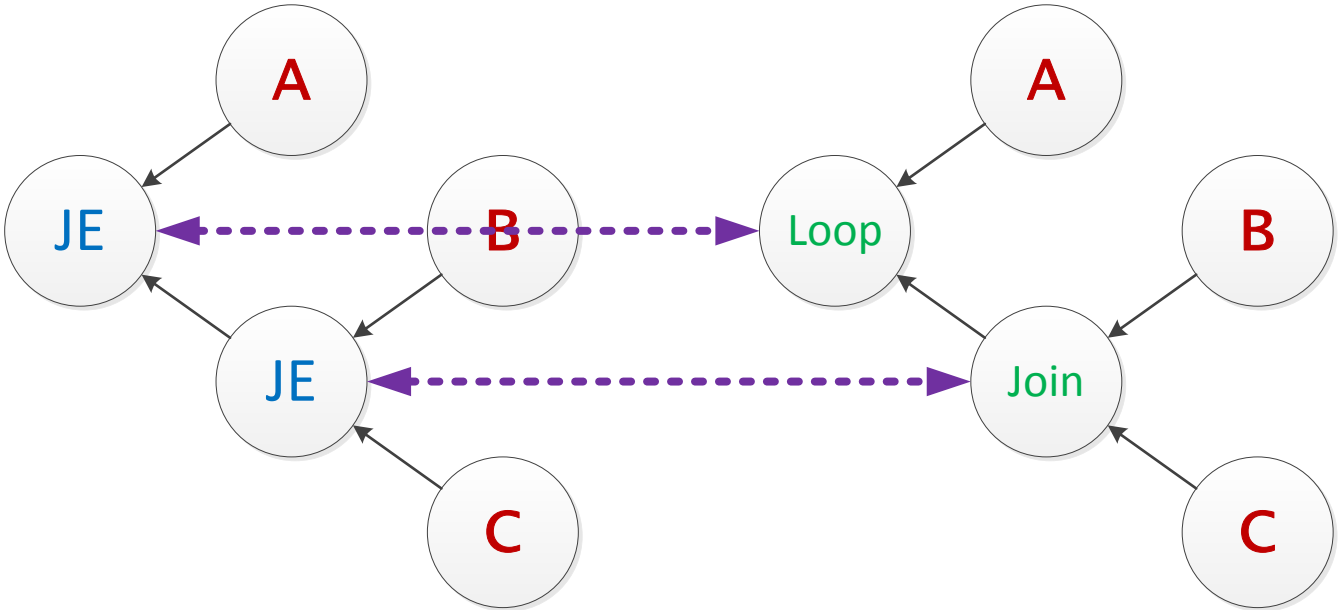
Query Profiler

Совмещаем с планом – JOIN



Query Profiler

Совмещаем с планом – JOIN



Query Profiler

Совмещаем с планом – JOIN

```
EXPLAIN (ANALYZE, BUFFERS, COSTS off)
SELECT
  *
FROM
  pg_i
JOIN
  pg_c
  ON
```

.776ms
3.890ms
4.998ms

#0 **3.890**ms (32.6%), rows=3890, loops=1

Hash Join (cost=505.03..1140.40 rows=3532 width=736) (actual time=3.095..11.937 rows=3890 loops=1)
Hash Cond: (cl.oid = idx.indexrelid)
Buffers: shared hit=405

```
Hash Join  (cost=505.03..1140.40 rows=3532 width=736) (actual time=3.095..11.937 rows=3890 loops=1)
  Hash Cond: (cl.oid = idx.indexrelid)
  Buffers: shared hit=405
  ->  Seq Scan on pg_class cl  (cost=0.00..616.70 rows=7094 width=540) (actual time=0.016..4.998 rows=6481 loops=
1)
        Buffers: shared hit=262
  ->  Hash  (cost=319.60..319.60 rows=3532 width=200) (actual time=3.048..3.049 rows=3890 loops=1)
        Buckets: 4096  Batches: 1  Memory Usage: 1058kB
        Buffers: shared hit=143
```

перейти к анализу

11.937

Query Profiler

Совмещаем с планом – JOIN *

```
A JOIN B JOIN C
```

```
-> Hash Join
     -> Hash
          -> Seq Scan on B
     -> Merge Join
          -> Index Scan on C
          -> Index Scan on A
```

```
(A JOIN B) JOIN C
```

```
A JOIN (B JOIN C)
```

```
A, B
```

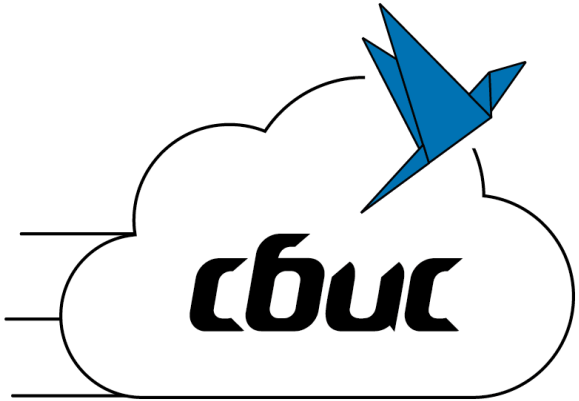
¯_(ツ)_/¯

Query Profiler

Совмещаем с планом – JOIN *

```
        FROM
.010ms    "ТочкаПродаж" ps
.013ms  LEFT JOIN
.007ms    "СтруктураПредприятия" sp
            USING("@Лицо")
        WHERE
          ps."@Лицо" = 79
.008ms UNION
        SELECT
          ps."@Лицо"
        , sp."Партнер" AND
          NOT ps."Категория"[13]
        , pz."Закрепление"
        , pz."Дата"
        , if(ps."@Лицо" = 79, -1, parent_branch."Level" + 1)
        FROM
.010ms    "ТочкаПродаж" ps
        LEFT JOIN
.007ms    "СтруктураПредприятия" sp
            USING("@Лицо")
.004ms  LEFT JOIN
.258ms    pz
            ON pz."Партнер" = ps."@Лицо"
.003ms  JOIN
.002ms    parent_branch
            ON pz."Партнер" = parent_branch."Закрепление"
```


31-60
61-90
301-330



Спасибо за внимание!

Боровиков Кирилл

kilor@tensor.ru / https://n.sbis.ru/explain

sbis.ru / tensor.ru