How Did We Live Without LATERAL?

Vik Fearing

Nordic PGDay — March 17, 2016
Helsinki, Finland

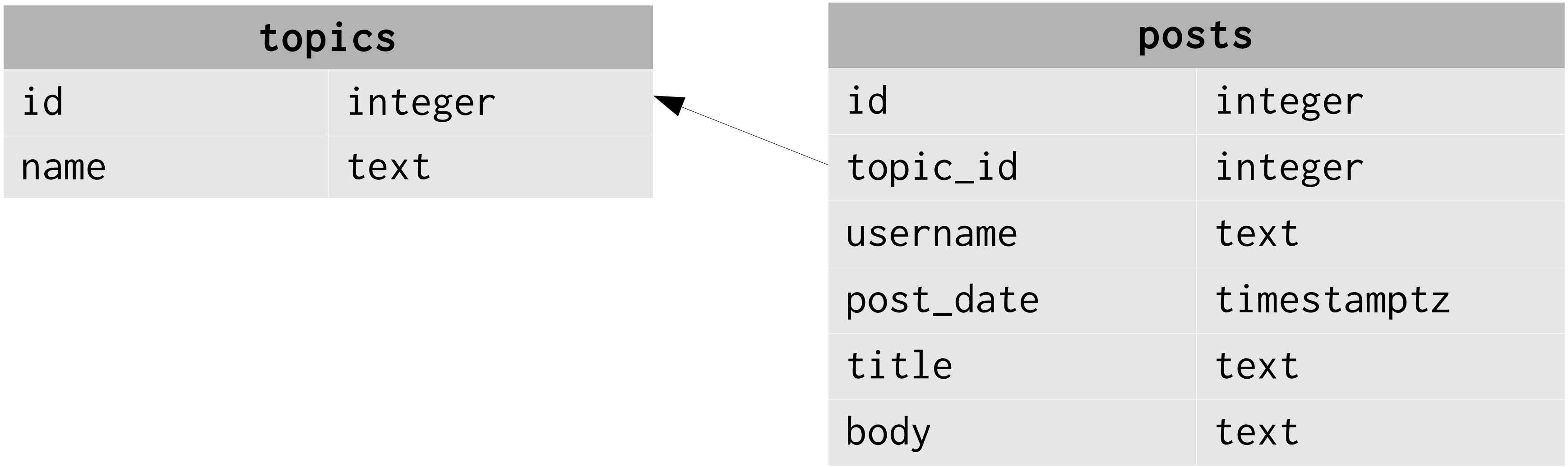
About Me

Vik Fearing

2ndQuadrant France

irc: xocolatl

twitter: @pg_xocolatl



topics	
id	integer
name	text

posts	
id	integer
topic_id	integer
username	text
post_date	timestamptz
title	text
body	text

- 20 topics
- Thousands of posts per topic
- Except “Topic 12” which has none

Goal:

Display all the topics in alphabetical order,
with information about the last five posts.

```
SELECT t.name,
       p.username,
       p.post_date,
       p.title
FROM topics t
LEFT JOIN
     (SELECT username, post_date, title
      FROM posts
      WHERE topic_id = t.id
      ORDER BY post_date DESC
      LIMIT 5) p ON true
ORDER BY t.name, p.post_date DESC;
```

```
ERROR:  invalid reference to FROM-clause
entry for table ”t”
LINE 9:       WHERE topic_id = t.id
                               ^
HINT:  There is an entry for table ”t”, but
it cannot be referenced from this part of
the query.
```

```
ERROR:  invalid reference to FROM-clause
entry for table ”t”
LINE 9:       WHERE topic_id = t.id
                               ^
HINT:  There is an entry for table ”t”, but
it cannot be referenced from this part of
the query.
```

Window Functions

```
SELECT topics.name,
       tmp.username,
       tmp.post_date,
       tmp.title
FROM topics
LEFT JOIN
  (SELECT *
   FROM
     (SELECT *,
             row_number() OVER
                 (PARTITION BY topic_id
                  ORDER BY post_date DESC) rownum
      FROM posts) tmpin
   WHERE rownum <= 5) tmp ON tmp.topic_id = topics.id
ORDER BY topics.name, tmp.post_date DESC;
```

```
SELECT topics.name,
       tmp.username,
       tmp.post_date,
       tmp.title
FROM topics
LEFT JOIN
  (SELECT *
   FROM
     (SELECT *,
             row_number() OVER
                 (PARTITION BY topic_id
                  ORDER BY post_date DESC) rownum
      FROM posts) tmpin
   WHERE rownum <= 5) tmp ON tmp.topic_id = topics.id
ORDER BY topics.name, tmp.post_date DESC;
```

```
SELECT topics.name,
       tmp.username,
       tmp.post_date,
       tmp.title
FROM topics
LEFT JOIN
  (SELECT *
   FROM
     (SELECT *,
             row_number() OVER
                 (PARTITION BY topic_id
                  ORDER BY post_date DESC) rownum
      FROM posts) tmpin
   WHERE rownum <= 5) tmp ON tmp.topic_id = topics.id
ORDER BY topics.name, tmp.post_date DESC;
```

```
SELECT topics.name,
       tmp.username,
       tmp.post_date,
       tmp.title
FROM topics
LEFT JOIN
  (SELECT *
   FROM
     (SELECT *,
             row_number() OVER
                 (PARTITION BY topic_id
                  ORDER BY post_date DESC) rownum
      FROM posts) tmpin
   WHERE rownum <= 5) tmp ON tmp.topic_id = topics.id
ORDER BY topics.name, tmp.post_date DESC;
```

```
Sort  (cost=20945.60..21067.05 rows=48583 width=31)
      (actual time=208.715..208.719 rows=96 loops=1)
  Sort Key: topics.name, tmpin.post_date
  Sort Method: quicksort  Memory: 32kB
  ->  Merge Left Join  (cost=0.56..15998.36 rows=48583 width=31)
                       (actual time=0.057..208.594 rows=96 loops=1)
        Merge Cond: (topics.id = tmpin.topic_id)
        ->  Index Scan using topics_pkey on topics
            (cost=0.14..12.44 rows=20 width=13)
            (actual time=0.009..0.024 rows=20 loops=1)
        ->  Materialize  (...) (...)
              ->  Subquery Scan on tmpin
                  (...rows=48583...) (...rows=95...)
                    Filter: (tmpin.rownum <= 5)
                    Rows Removed by Filter: 145654
                    ->  WindowAgg
                        (...rows=145749...) (...rows=145749...)
                          ->  Index Scan using ... on posts
                              (...rows=145749...) (...rows=145749...)
Total runtime: 208.809 ms
```

```
Sort  (cost=20945.60..21067.05 rows=48583 width=31)
      (actual time=208.715..208.719 rows=96 loops=1)
  Sort Key: topics.name, tmpin.post_date
  Sort Method: quicksort  Memory: 32kB
  ->  Merge Left Join  (cost=0.56..15998.36 rows=48583 width=31)
                       (actual time=0.057..208.594 rows=96 loops=1)
        Merge Cond: (topics.id = tmpin.topic_id)
        ->  Index Scan using topics_pkey on topics
            (cost=0.14..12.44 rows=20 width=13)
            (actual time=0.009..0.024 rows=20 loops=1)
        ->  Materialize  (...) (...)
              ->  Subquery Scan on tmpin
                  (...rows=48583...) (...rows=95...)
                    Filter: (tmpin.rownum <= 5)
                    Rows Removed by Filter: 145654
                    ->  WindowAgg
                        (...rows=145749...) (...rows=145749...)
                          ->  Index Scan using ... on posts
                              (...rows=145749...) (...rows=145749...)
Total runtime: 208.809 ms
```

```
Sort  (cost=20945.60..21067.05 rows=48583 width=31)
      (actual time=208.715..208.719 rows=96 loops=1)
  Sort Key: topics.name, tmpin.post_date
  Sort Method: quicksort  Memory: 32kB
  ->  Merge Left Join  (cost=0.56..15998.36 rows=48583 width=31)
                       (actual time=0.057..208.594 rows=96 loops=1)
        Merge Cond: (topics.id = tmpin.topic_id)
        ->  Index Scan using topics_pkey on topics
            (cost=0.14..12.44 rows=20 width=13)
            (actual time=0.009..0.024 rows=20 loops=1)
        ->  Materialize  (...) (...)
              ->  Subquery Scan on tmpin
                  (...rows=48583...) (...rows=95...)
                    Filter: (tmpin.rownum <= 5)
                    Rows Removed by Filter: 145654
                    ->  WindowAgg
                        (...rows=145749...) (...rows=145749...)
                          ->  Index Scan using ... on posts
                              (...rows=145749...) (...rows=145749...)
Total runtime: 208.809 ms
```

Stored Procedures

pl/pgsql

```
CREATE FUNCTION n_posts (topic integer, num integer)
RETURNS SETOF posts AS
$$
DECLARE
    empty posts;
BEGIN
    RETURN QUERY
        SELECT * FROM posts
        WHERE topic_id = $1
        ORDER BY post_date DESC
        LIMIT $2;
    IF NOT FOUND THEN
        RETURN NEXT empty;
    END IF;
END;
$$
LANGUAGE plpgsql;
```

```
CREATE FUNCTION n_posts (topic integer, num integer)
RETURNS SETOF posts AS
$$
DECLARE
    empty posts;
BEGIN
    RETURN QUERY
        SELECT * FROM posts
        WHERE topic_id = $1
        ORDER BY post_date DESC
        LIMIT $2;
    IF NOT FOUND THEN
        RETURN NEXT empty;
    END IF;
END;
$$
LANGUAGE plpgsql;
```

```
SELECT topics.name, (n_posts(id, 5)).*
FROM topics
ORDER BY topics.name;
```

```
SELECT topics.name, (n_posts(id, 5)).*
FROM topics
ORDER BY topics.name;
```

IT'S A TRAP!

```
SELECT topics.name, (n_posts(id, 5)).*
FROM topics
ORDER BY topics.name;
```

IT'S A TRAP!

```
Sort  (cost=1554.87..1604.87 rows=20000 width=13)
      (actual time=27.005..27.018 rows=96 loops=1)
  Sort Key: name
  Sort Method: quicksort  Memory: 32kB
  ->  Seq Scan on topics
      (cost=0.00..126.10 rows=20000 width=13)
      (actual time=1.071..26.764 rows=96 loops=1)
Total runtime: 27.053 ms
```

```
Sort  (cost=1554.87..1604.87 rows=20000 width=13)
      (actual time=27.005..27.018 rows=96 loops=1)
  Sort Key: name
  Sort Method: quicksort  Memory: 32kB
  ->  Seq Scan on topics
      (cost=0.00..126.10 rows=20000 width=13)
      (actual time=1.071..26.764 rows=96 loops=1)
Total runtime: 27.053 ms
```

Arrays

```
SELECT name, (unnest(posts)).*
FROM
  (SELECT *,
     (SELECT coalesce(array_agg(posts),
                      array[null::posts]) AS posts
      FROM
        (SELECT posts
         FROM posts
         WHERE topic_id = topics.id
         ORDER BY post_date DESC
         LIMIT 5) tmp)
   FROM topics
   OFFSET 0) AS tmp;
```

```
SELECT name, (unnest(posts)).*
FROM
  (SELECT *,
     (SELECT coalesce(array_agg(posts),
                      array[null::posts]) AS posts
      FROM
        (SELECT posts
         FROM posts
         WHERE topic_id = topics.id
         ORDER BY post_date DESC
         LIMIT 5) tmp)
   FROM topics
   OFFSET 0) AS tmp;
```

```
SELECT name, (unnest(posts)).*
FROM
  (SELECT *,
     (SELECT coalesce(array_agg(posts),
                      array[null::posts]) AS posts
      FROM
        (SELECT posts
         FROM posts
         WHERE topic_id = topics.id
         ORDER BY post_date DESC
         LIMIT 5) tmp)
   FROM topics
   OFFSET 0) AS tmp;
```

```
Subquery Scan on tmp
(cost=0.00..108.70 rows=2000 width=41)
(actual time=0.161..1.221 rows=96 loops=1)
  ->  Seq Scan on topics
      (cost=0.00..98.35 rows=20 width=13)
      (actual time=0.110..0.868 rows=20 loops=1)
        SubPlan 1
          ->  Aggregate
              (cost=4.85..4.86 rows=1 width=54)
              (actual time=0.039..0.039 rows=1 loops=20)
                ->  Limit  (...) (...)
                      ->  Index Scan ...
                            Index Cond: (topic_id = topics.id)
Total runtime: 1.325 ms
```

```
Subquery Scan on tmp
(cost=0.00..108.70 rows=2000 width=41)
(actual time=0.161..1.221 rows=96 loops=1)
  ->  Seq Scan on topics
      (cost=0.00..98.35 rows=20 width=13)
      (actual time=0.110..0.868 rows=20 loops=1)
        SubPlan 1
          ->  Aggregate
              (cost=4.85..4.86 rows=1 width=54)
              (actual time=0.039..0.039 rows=1 loops=20)
                ->  Limit  (...) (...)
                      ->  Index Scan ...
                            Index Cond: (topic_id = topics.id)
Total runtime: 1.325 ms
```

```
Subquery Scan on tmp
(cost=0.00..108.70 rows=2000 width=41)
(actual time=0.161..1.221 rows=96 loops=1)
  ->  Seq Scan on topics
      (cost=0.00..98.35 rows=20 width=13)
      (actual time=0.110..0.868 rows=20 loops=1)
        SubPlan 1
          ->  Aggregate
              (cost=4.85..4.86 rows=1 width=54)
              (actual time=0.039..0.039 rows=1 loops=20)
                ->  Limit  (...) (...)
                      ->  Index Scan ...
                            Index Cond: (topic_id = topics.id)
Total runtime: 1.325 ms
```

Recursive Queries

```
WITH RECURSIVE
  rp AS (SELECT topic_name as topic_name, (p).*, 1 AS rcount
           FROM (SELECT t.name as topic_name,
                        (SELECT p FROM posts p
                          WHERE p.topic_id = t.id
                          ORDER BY p.post_date DESC, p.id DESC
                          LIMIT 1) AS p
                   FROM topics t offset 0) s
                  WHERE (p).id IS NOT NULL
         UNION ALL
         SELECT topic_name, (p).*, s.rcount + 1
           FROM (SELECT rp.topic_name,
                         (SELECT p FROM posts p
                          WHERE p.topic_id = rp.topic_id
                            AND (p.post_date, p.id) < (rp.post_date, rp.id)
                          ORDER BY p.post_date DESC, p.id DESC
                          LIMIT 1) AS p,
                        rp.rcount
                   FROM rp
                  WHERE rp.rcount < 5 offset 0) s
          WHERE (p).id IS NOT NULL)
SELECT topic_name, username, post_date, title
FROM rp
ORDER BY topic_name, post_date DESC;
```

Who wants to maintain that?

```
CREATE INDEX posts_topic_id_post_date_id_idx
ON posts USING btree
(topic_id, post_date DESC, id DESC);
```

```
 Sort  (cost=1943.52..1945.25 rows=690 width=104)
       (actual time=3.490..3.503 rows=95 loops=1)
   Sort Key: rp.topic_name, rp.post_date
   Sort Method: quicksort  Memory: 32kB
   CTE rp
     ->  Recursive Union  (cost=0.00..1897.19 rows=690 width=68)
                          (actual time=0.077..2.280 rows=95 loops=1)
           ->  Subquery Scan on s  (cost=0.00..27.34 rows=20 width=41)
                                   (actual time=0.074..0.437 rows=19 loops=1)
                 Filter: ((s.p).id IS NOT NULL)
                 Rows Removed by Filter: 1
                 ->  Seq Scan on topics t  (cost=0.00..27.14 rows=20 width=13)
                                           (actual time=0.066..0.397 rows=20 loops=1)
                       SubPlan 1
                         ->  Limit  (cost=0.42..1.30 rows=1 width=66)
                                    (actual time=0.016..0.016 rows=1 loops=20)
                               ->  Index Scan using posts_topic_id_post_date_id_idx on posts p
                                   (cost=0.42..6729.50 rows=7671 width=66)
                                   (actual time=0.015..0.015 rows=1 loops=20)
                                     Index Cond: (topic_id = t.id)
           ->  Subquery Scan on s_1  (cost=0.00..185.60 rows=67 width=68)
                                     (actual time=0.032..0.349 rows=15 loops=5)
                 Filter: ((s_1.p).id IS NOT NULL)
                 ->  WorkTable Scan on rp rp_1  (cost=0.00..184.77 rows=67 width=52)
                                                (actual time=0.028..0.316 rows=15 loops=5)
                       Filter: (rcount < 5)
                       Rows Removed by Filter: 4
                       SubPlan 2
                         ->  Limit  (cost=0.42..2.69 rows=1 width=66)
                                    (actual time=0.017..0.017 rows=1 loops=76)
                               ->  Index Scan using posts_topic_id_post_date_id_idx on posts p_1
                                   (cost=0.42..5799.82 rows=2557 width=66)
                                   (actual time=0.016..0.016 rows=1 loops=76)
                                     Index Cond: ((topic_id = rp_1.topic_id) AND
                                                  (ROW(post_date, id) < ROW(rp_1.post_date,
rp_1.id)))
   ->  CTE Scan on rp  (cost=0.00..13.80 rows=690 width=104) (actual time=0.082..2.432 rows=95
loops=1)
 Total runtime: 3.698 ms
```

```
Sort  (cost=1943.52..1945.25 rows=690 width=104)
      (actual time=3.490..3.503 rows=95 loops=1)

...

Total runtime: 3.698 ms
```

LATERAL

```
SELECT t.name,
       p.username,
       p.post_date,
       p.title
FROM topics t
LEFT JOIN LATERAL
     (SELECT username, post_date, title
      FROM posts
      WHERE topic_id = t.id
      ORDER BY post_date DESC
      LIMIT 5) p ON true
ORDER BY t.name, p.post_date DESC;
```

```
SELECT t.name,
       p.username,
       p.post_date,
       p.title
FROM topics t
LEFT JOIN LATERAL
     (SELECT username, post_date, title
      FROM posts
      WHERE topic_id = t.id
      ORDER BY post_date DESC
      LIMIT 5) p ON true
ORDER BY t.name, p.post_date DESC;
```

```
SELECT t.name,
       p.username,
       p.post_date,
       p.title
FROM topics t
LEFT JOIN LATERAL
     (SELECT username, post_date, title
      FROM posts
      WHERE topic_id = t.id
      ORDER BY post_date DESC
      LIMIT 5) p ON true
ORDER BY t.name, p.post_date DESC;
```

```
Sort  (cost=102.17..102.42 rows=100 width=31)
      (actual time=0.601..0.610 rows=96 loops=1)
  Sort Key: t.name, posts.post_date
  Sort Method: quicksort  Memory: 32kB
  ->  Nested Loop Left Join
      (cost=0.42..98.85 rows=100 width=31)
      (actual time=0.044..0.403 rows=96 loops=1)
        ->  Seq Scan on topics t
            (cost=0.00..1.20 rows=20 width=13)
            (actual time=0.009..0.012 rows=20 loops=1)
        ->  Limit  (...rows=5...) (...rows=5...)
              ->  Index Scan using ... on posts
                  (...rows=7671...) (...rows=5...)
                    Index Cond: (topic_id = t.id)
Total runtime: 0.660 ms
```

```
Sort  (cost=102.17..102.42 rows=100 width=31)
      (actual time=0.601..0.610 rows=96 loops=1)
  Sort Key: t.name, posts.post_date
  Sort Method: quicksort  Memory: 32kB
  ->  Nested Loop Left Join
      (cost=0.42..98.85 rows=100 width=31)
      (actual time=0.044..0.403 rows=96 loops=1)
        ->  Seq Scan on topics t
            (cost=0.00..1.20 rows=20 width=13)
            (actual time=0.009..0.012 rows=20 loops=1)
        ->  Limit  (...rows=5...) (...rows=5...)
              ->  Index Scan using ... on posts
                  (...rows=7671...) (...rows=5...)
                    Index Cond: (topic_id = t.id)
Total runtime: 0.660 ms
```

```
Sort  (cost=102.17..102.42 rows=100 width=31)
      (actual time=0.601..0.610 rows=96 loops=1)
  Sort Key: t.name, posts.post_date
  Sort Method: quicksort  Memory: 32kB
  ->  Nested Loop Left Join
      (cost=0.42..98.85 rows=100 width=31)
      (actual time=0.044..0.403 rows=96 loops=1)
        ->  Seq Scan on topics t
            (cost=0.00..1.20 rows=20 width=13)
            (actual time=0.009..0.012 rows=20 loops=1)
        ->  Limit  (...rows=5...) (...rows=5...)
              ->  Index Scan using ... on posts
                  (...rows=7671...) (...rows=5...)
                    Index Cond: (topic_id = t.id)
Total runtime: 0.660 ms
```

```
Sort  (cost=102.17..102.42 rows=100 width=31)
      (actual time=0.601..0.610 rows=96 loops=1)
  Sort Key: t.name, posts.post_date
  Sort Method: quicksort  Memory: 32kB
  ->  Nested Loop Left Join
      (cost=0.42..98.85 rows=100 width=31)
      (actual time=0.044..0.403 rows=96 loops=1)
        ->  Seq Scan on topics t
            (cost=0.00..1.20 rows=20 width=13)
            (actual time=0.009..0.012 rows=20 loops=1)
        ->  Limit  (...rows=5...) (...rows=5...)
              ->  Index Scan using ... on posts
                  (...rows=7671...) (...rows=5...)
                    Index Cond: (topic_id = t.id)
Total runtime: 0.660 ms
```

unnest() for JSON

```
SELECT name, j->’vars’->g AS vars
FROM (VALUES (’hello’, ’{”vars”: [1, 2, 3]}’::json),
             (’world’, ’{”vars”: [4, 5]}’::json)) v(name, j),
     LATERAL generate_series(0, json_array_length(j->'vars')-1) g;
```

name	vars
hello	1
hello	2
hello	3
world	4
world	5

Table Permissions

```
SELECT c.oid::regclass, x.*
FROM pg_class c,
     aclexplode(c.relacl) x
WHERE c.oid = ’pg_class’::regclass;
```

oid	grantor	grantee	privilege_type	is_grantable
pg_class	10	0	SELECT	f

Questions?

About Me

Vik Fearing

2ndQuadrant France

irc: xocolatl

twitter: @pg_xocolatl