I Didn’t Know PostgreSQL Could Do That!

WARNING: CODE AHEAD (AKA SIT CLOSE)

Robert Treat

Principal Database Engineer



© 2025, Amazon Web Services, Inc. or its affiliates.

One Weird Trick

© 2025, Amazon Web Services, Inc. or its affiliates..

One Weird Trick

© 2025, Amazon Web Services, Inc. or its affiliates..

pg_stat_statements



© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# create extension pg_stat_statements;
CREATE EXTENSION
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# \d pg_stat_statements
                    View "public.pg_stat_statements"
       Column        |       Type       | Collation | Nullable | Default
---------------------+------------------+-----------+----------+---------
 userid              | oid              |           |          |
 dbid                | oid              |           |          |
 toplevel            | boolean          |           |          |
 queryid             | bigint           |           |          |
 query               | text             |           |          |
 plans               | bigint           |           |          |
 total_plan_time     | double precision |           |          |
 min_plan_time       | double precision |           |          |
 max_plan_time       | double precision |           |          |
 mean_plan_time      | double precision |           |          |
 stddev_plan_time    | double precision |           |          |
 calls               | bigint           |           |          |
 total_exec_time     | double precision |           |          |
 min_exec_time       | double precision |           |          |
 max_exec_time       | double precision |           |          |
 mean_exec_time      | double precision |           |          |
 stddev_exec_time    | double precision |           |          |
 rows                | bigint           |           |          |
 shared_blks_hit     | bigint           |           |          |
 shared_blks_read    | bigint           |           |          |
 shared_blks_dirtied | bigint           |           |          |
 shared_blks_written | bigint           |           |          |
 local_blks_hit      | bigint           |           |          |
 local_blks_read     | bigint           |           |          |
 local_blks_dirtied  | bigint           |           |          |
 local_blks_written  | bigint           |           |          |
 temp_blks_read      | bigint           |           |          |
 temp_blks_written   | bigint           |           |          |
 blk_read_time       | double precision |           |          |
 blk_write_time      | double precision |           |          |
 wal_records         | bigint           |           |          |
 wal_fpi             | bigint           |           |          |
 wal_bytes           | numeric          |           |          |
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
-[ RECORD 1 ]----+----------------------
userid           | 10
dbid             | 43566
queryid          | 2983775486487580928
calls            | 2
total_exec_time  | 0.25535800000000003
min_exec_time    | 0.116851
max_exec_time    | 0.13850700000000002
mean_exec_time   | 0.12767900000000001
stddev_exec_time | 0.010828000000000015
rows             | 0
query            | select * from actor where first_name = $1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# select userid, dbid, queryid, calls, total_exec_time, min_exec_time, max_exec_time, mean_exec_time,
stddev_exec_time, rows, query::char(50) from pg_stat_statements where dbid = 43566 and query ~ '^s';
 userid | dbid  |       queryid        | calls
|  total_exec_time   |   min_exec_time    |    max_exec_time    |   mean_exec_time    |   stddev_exec_time   | rows
|                  query
--------+-------+----------------------+-------+--------------------+--------------------+--------------------+-------------
-------+--------------------+------+------------------------------------------------
     10 | 43566 |  298377548648758092 |     2 | 0.2553580000000003 |          0.116851 |
0.13850700000000002 | 0.1276790000000001 | 0.01082800000000015 |  400 | select * from actor
     10 | 43566 | -8322612994292972818 |     4 |           1.757743 |          0.034259 |  1.601076000000002
|         0.43943575 |    0.670959248394556 |    1 | select * from actor where first_name = $1
     10 | 43566 |  6252506586341865369 |     5 |  2538.0079760000003 |  504.4303100000001
|  511.12813700000004 |        507.6015952 |   2.450660837350304 | 1000 | select * from actor_info
     10 | 43566 |  3547040367596174317 |     1 |           1.813802 |          1.813802 |           1.813802
|           1.813802 |                  0 |    1 | select count(*) from payment
     10 | 43566 | -4257403805458828492 |     2 |          40.521781 |          0.034998 |  40.48678299999996
|         20.2608905 |   20.22589249999997 |   18 | select * from pg_database
     10 | 43566 | -2570349626643393169 |     3 |          61.871451 |          7.063284 |  47.67332999999996
|          20.623817 |    19.12691143112982 |  119 | select * from pg_stat_statements where query ~ $1
     10 | 43566 | -8541103795978023600 |     1 |   78.8289509999999 |  78.8289509999999
|   78.8289509999999 |   78.8289509999999 |                  0 | 1921 | select * from pg_stat_statements
     10 | 43566 | -4297514348524569892 |     2 |          12.874234 |          6.232544 |           6.64169
|          6.437117 |   0.204572999999999 | 2125 | select userid, dbid, queryid, calls, total_exec_ti
```

© 2025, Amazon Web Services, Inc. or its affiliates.

Transactional DDL



© 2025, Amazon Web Services, Inc. or its affiliates.

Transactions

```
BEGIN;

<sql goes here>

ROLLBACK;
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
BEGIN;

DROP TABLE actor;

ROLLBACK;
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
BEGIN;

CREATE INDEX … ON …;

EXPLAIN ANALYZE …;

ROLLBACK;
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
BEGIN;

CREATE INDEX … ON …;

EXPLAIN ANALYZE …;

ROLLBACK;
```

(don’t do this)

© 2025, Amazon Web Services, Inc. or its affiliates.

```
BEGIN;

CREATE TABLE …;

CREATE TABLE … REFERENCES …;

COMMIT;
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
BEGIN;

ALTER TABLE actor RENAME
COLUMN ‘family_name’ to ‘last_name’;

SAVEPOINT “run-tests”;

INSERT INTO actor (first_name, last_name)
SELECT ‘Robert’,’Treat’;

ROLLBACK TO “run-tests”;

COMMIT;
```

© 2025, Amazon Web Services, Inc. or its affiliates.

Insert … Returning



© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# INSERT INTO actor (first_name, last_name)
         VALUES ('ROB','TREAT’)
         RETURNING *;


-[ RECORD 1 ]--------------------------
actor_id    | 202
first_name  | ROB
last_name   | TREAT
last_update | 2025-08-01 17:44:38.097142

insert 0 1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# UPDATE actor
         SET first_name = 'ROBERT’
         WHERE actor_id = 202
         RETURNING first_name, last_name, last_update;


-[ RECORD 1 ]--------------------------
first_name  | ROBERT
last_name   | TREAT
last_update | 2025-08-01 17:53:21.777389

update 1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# DELETE FROM actor
         WHERE last_name = 'TREAT’
         RETURNING *;


-[ RECORD 1 ]-------------------------
actor_id    | 202
first_name  | ROBERT
last_name   | TREAT
last_update | 2025-08-01 17:53:21.777389

delete 1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

CTE … Returning

© 2025, Amazon Web Services, Inc. or its affiliates.

Common Table Expressions



© 2025, Amazon Web Services, Inc. or its affiliates.

CTE

```
pagila=# WITH underemployed AS
            (SELECT * FROM actor
             WHERE actor_id NOT IN
                   (SELECT actor_id FROM film_actor)
            )
            SELECT * FROM underemployed;

-[ RECORD 1 ]--------------------
actor_id    | 42
first_name  | TOM
last_name   | MIRANDA
last_update | 2006-02-15 09:34:33
```

© 2025, Amazon Web Services, Inc. or its affiliates.

CTE

```
pagila=# WITH underemployed AS
            (SELECT * FROM actor
             WHERE actor_id NOT IN
                   (SELECT actor_id FROM film_actor)
            )
            SELECT * FROM underemployed;

-[ RECORD 1 ]--------------------
actor_id    | 42
first_name  | TOM
last_name   | MIRANDA
last_update | 2006-02-15 09:34:33
```

© 2025, Amazon Web Services, Inc. or its affiliates.

CTE

```
pagila=# WITH underemployed AS
            (SELECT * FROM actor
             WHERE actor_id NOT IN
                   (SELECT actor_id FROM film_actor)
            )
            SELECT * FROM underemployed;

-[ RECORD 1 ]--------------------
actor_id    | 42
first_name  | TOM
last_name   | MIRANDA
last_update | 2006-02-15 09:34:33
```

© 2025, Amazon Web Services, Inc. or its affiliates.

CTE … RETURNING

```
pagila=# WITH rip AS
         (DELETE FROM inventory
          WHERE inventory_id NOT IN
                (SELECT inventory_id FROM rental)
          RETURNING *
         )
         INSERT INTO giveaway SELECT * FROM rip
         RETURNING *;

inventory_id | film_id | store_id |     last_update
-------------|---------|----------|---------------------
           5 |       1 |        2 | 2006-02-15 10:09:17

INSERT 0 1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

CTE … RETURNING

```
pagila=# WITH rip AS
         (DELETE FROM inventory
          WHERE inventory_id NOT IN
                (SELECT inventory_id FROM rental)
          RETURNING *
         )
         INSERT INTO giveaway SELECT * FROM rip
         RETURNING *;

 inventory_id | film_id | store_id |     last_update
--------------|---------|----------|---------------------
            5 |       1 |        2 | 2006-02-15 10:09:17

INSERT 0 1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

CTE … RETURNING

```
pagila=# WITH rip AS
         (DELETE FROM inventory
          WHERE inventory_id NOT IN
                (SELECT inventory_id FROM rental)
          RETURNING *
         )
         INSERT INTO giveaway SELECT * FROM rip
         RETURNING *;

inventory_id | film_id | store_id |     last_update
-------------|---------|----------|---------------------
           5 |       1 |        2 | 2006-02-15 10:09:17

INSERT 0 1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

CTE … RETURNING

```
pagila=# WITH rip AS
         (DELETE FROM inventory
          WHERE inventory_id NOT IN
                (SELECT inventory_id FROM rental)
          RETURNING *
         )
         INSERT INTO giveaway SELECT * FROM rip
         RETURNING *;

inventory_id | film_id | store_id |     last_update
-------------|---------|----------|---------------------
           5 |       1 |        2 | 2006-02-15 10:09:17

INSERT 0 1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

CTE … RETURNING

```
pagila=# WITH rip AS
         (DELETE FROM inventory
          WHERE inventory_id NOT IN
                (SELECT inventory_id FROM rental)
          RETURNING *
         )
         INSERT INTO giveaway SELECT * FROM rip
         RETURNING *;

inventory_id | film_id | store_id |     last_update
-------------|---------|----------|---------------------
           5 |       1 |        2 | 2006-02-15 10:09:17

INSERT 0 1
```

© 2025, Amazon Web Services, Inc. or its affiliates.

Range Types



© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# \d rental
                    Table ”public.rental"
    Column    |            Type             | Nullable
--------------+-----------------------------+----------
 rental_id    | integer                     | not null
 rental_date  | timestamp without time zone | not null
 inventory_id | integer                     | not null
 customer_id  | smallint                    | not null
 return_date  | timestamp without time zone | not null
 staff_id     | smallint                    | not null
 last_update  | timestamp without time zone | not null
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# ALTER TABLE rental
         ADD COLUMN rental_period tsrange;

ALTER TABLE
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# UPDATE rental
         SET rental_period =
         tsrange(rental_date, return_date);


UPDATE 16044
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# select rental_id, rental_date, return_date, rental_period from rental limit 5;

 rental_id |     rental_date     |     return_date     |                 rental_period
-----------+---------------------+---------------------+---------------------------------------
         2 | 2005-05-24 22:54:33 | 2005-05-28 19:40:33 | ["2005-05-24 22:54:33","2005-05-28 19:40:33")
         3 | 2005-05-24 23:03:39 | 2005-06-01 22:12:39 | ["2005-05-24 23:03:39","2005-06-01 22:12:39")
         4 | 2005-05-24 23:04:41 | 2005-06-03 01:43:41 | ["2005-05-24 23:04:41","2005-06-03 01:43:41")
         5 | 2005-05-24 23:05:21 | 2005-06-02 04:33:21 | ["2005-05-24 23:05:21","2005-06-02 04:33:21")
         6 | 2005-05-24 23:08:07 | 2005-05-27 01:32:07 | ["2005-05-24 23:08:07","2005-05-27 01:32:07")
(5 rows)
```

© 2025, Amazon Web Services, Inc. or its affiliates.

How many movies were being rented on June 21st?

“contains”

```
pagila=# SELECT count(*)
         FROM rental
         WHERE rental_period @> '2005-06-21'::timestamp;
 count
-------
  1483
(1 row)
```

© 2025, Amazon Web Services, Inc. or its affiliates.

How many movies did we rent during the month of June?

“within”

```
pagila=# SELECT count(*)
         FROM rental
         WHERE rental_period &&
         tsrange('2005-06-01'::timestamp,'2005-06-30'::timestamp);
 count
-------
  3072
(1 row)
```

© 2025, Amazon Web Services, Inc. or its affiliates.

Running Totals

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# SELECT customer_id, payment_date::timestamp(0), amount,
   coalesce(
      sum(amount) over (
          partition by customer_id
          order by payment_date asc
          rows between unbounded preceding and 1 preceding
      ),0) AS prior_spend
FROM payment WHERE customer_id = 599;
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# SELECT customer_id, payment_date::timestamp(0), amount,
   coalesce(
      sum(amount) over (
          partition by customer_id
          order by payment_date asc
          rows between unbounded preceding and 1 preceding
      ),0) AS prior_spend
FROM payment WHERE customer_id = 599;

 customer_id |    payment_date     | amount | prior_spend
-------------+---------------------+--------+-------------
         599 | 2006-12-04 17:20:34 |   4.99 |           0
         599 | 2006-12-31 16:55:51 |   4.99 |        4.99
         599 | 2007-01-10 19:14:53 |   6.99 |        9.98
         599 | 2007-01-31 15:06:10 |   6.99 |       16.97
         599 | 2007-02-25 16:57:08 |   6.99 |       23.96
         599 | 2007-03-01 18:07:15 |   2.99 |       30.95
<break>
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# SELECT customer_id, payment_date::timestamp(0), amount,
   coalesce(
      sum(amount) over (
          partition by customer_id
          order by payment_date asc
          rows between unbounded preceding and 1 preceding
      ),0) AS prior_spend
FROM payment WHERE customer_id = 599;

 customer_id |    payment_date     | amount | prior_spend
-------------+---------------------+--------+-------------
         599 | 2006-12-04 17:20:34 |   4.99 |           0
         599 | 2006-12-31 16:55:51 |   4.99 |        4.99
         599 | 2007-01-10 19:14:53 |   6.99 |        9.98
         599 | 2007-01-31 15:06:10 |   6.99 |       16.97
         599 | 2007-02-25 16:57:08 |   6.99 |       23.96
         599 | 2007-03-01 18:07:15 |   2.99 |       30.95
<break>
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# SELECT customer_id, payment_date::timestamp(0), amount,
   coalesce(
      sum(amount) over (
          partition by customer_id
          order by payment_date asc
          rows between unbounded preceding and 1 preceding
      ),0) AS prior_spend
FROM payment WHERE customer_id = 599;

 customer_id |    payment_date     | amount | prior_spend
-------------+---------------------+--------+-------------
         599 | 2006-12-04 17:20:34 |   4.99 |           0
         599 | 2006-12-31 16:55:51 |   4.99 |        4.99
         599 | 2007-01-10 19:14:53 |   6.99 |        9.98
         599 | 2007-01-31 15:06:10 |   6.99 |       16.97
         599 | 2007-02-25 16:57:08 |   6.99 |       23.96
         599 | 2007-03-01 18:07:15 |   2.99 |       30.95
<break>
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# SELECT customer_id, payment_date::timestamp(0), amount,
   coalesce(
      sum(amount) over (
          partition by customer_id
          order by payment_date asc
          rows between unbounded preceding and 1 preceding
      ),0) AS prior_spend
FROM payment WHERE customer_id = 599;

 customer_id |    payment_date     | amount | prior_spend
-------------+---------------------+--------+-------------
         599 | 2006-12-04 17:20:34 |   4.99 |           0
         599 | 2006-12-31 16:55:51 |   4.99 |        4.99
         599 | 2007-01-10 19:14:53 |   6.99 |        9.98
         599 | 2007-01-31 15:06:10 |   6.99 |       16.97
         599 | 2007-02-25 16:57:08 |   6.99 |       23.96
         599 | 2007-03-01 18:07:15 |   2.99 |       30.95
<break>
```

© 2025, Amazon Web Services, Inc. or its affiliates.

Window Functions



© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# SELECT customer_id, payment_date::timestamp(0), amount,
   coalesce(
      sum(amount) over (
          partition by customer_id
          order by payment_date asc
          rows between unbounded preceding and 1 preceding
      ),0) AS prior_spend
FROM payment WHERE customer_id = 599;

 customer_id |    payment_date     | amount | prior_spend
-------------+---------------------+--------+-------------
         599 | 2006-12-04 17:20:34 |   4.99 |           0
         599 | 2006-12-31 16:55:51 |   1.99 |        4.99
         599 | 2007-01-10 19:14:53 |   6.99 |        6.98
         599 | 2007-01-31 15:06:10 |   4.99 |       13.97
         599 | 2007-02-25 16:57:08 |   6.99 |       18.96
         599 | 2007-03-01 18:07:15 |   3.99 |       25.95
<break>
```

© 2025, Amazon Web Services, Inc. or its affiliates.

customer_id	payment_date	amount	prior_spend
599	2006-12-04 17:20:34	4.99	0
599	2006-12-31 16:55:51	4.99	4.99
599	2007-01-10 19:14:53	6.99	9.98
599	2007-01-31 15:06:10	6.99	16.97
599	2007-02-25 16:57:08	6.99	23.96
599	2007-03-01 18:07:15	2.99	30.95
599	2007-03-07 15:33:30	2.99	33.94
599	2007-03-12 18:16:45	9.99	36.93
599	2007-03-16 02:09:29	0.99	46.92
599	2007-03-17 02:22:52	3.99	47.91
599	2007-03-21 12:16:07	1.99	51.90
599	2007-03-26 00:58:47	4.99	53.89
599	2007-04-10 17:12:38	1.99	58.88
599	2007-04-17 18:26:43	2.99	60.87
599	2007-04-20 06:05:44	2.99	63.86
599	2007-05-01 03:12:57	2.99	66.85
599	2007-05-15 05:52:04	2.99	69.84
599	2007-06-06 07:55:14	1.99	72.83
599	2007-06-11 02:58:37	8.99	74.82

(19 rows)

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# SELECT customer_id, payment_date::timestamp(0), amount,
   coalesce(
      sum(amount) over (
          partition by customer_id
          order by payment_date asc
          rows between unbounded preceding and 1 preceding
      ),0) AS prior_spend
FROM payment WHERE customer_id = 599;
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
Append  (cost=0.28..66.84 rows=24 width=16)
   ->  Index Scan using idx_fk_payment_p2007_01_customer_id on payment_p2007_01
payment_1  (cost=0.28..8.32 rows=2 width=16)
         Index Cond: (customer_id = 599)
   ->  Index Scan using idx_fk_payment_p2007_02_customer_id on payment_p2007_02
payment_2  (cost=0.28..8.38 rows=5 width=16)
         Index Cond: (customer_id = 599)
   ->  Index Scan using idx_fk_payment_p2007_03_customer_id on payment_p2007_03
payment_3  (cost=0.28..8.40 rows=6 width=16)
         Index Cond: (customer_id = 599)
   ->  Index Scan using idx_fk_payment_p2007_04_customer_id on payment_p2007_04
payment_4  (cost=0.28..8.38 rows=5 width=16)
         Index Cond: (customer_id = 599)
   ->  Index Scan using idx_fk_payment_p2007_05_customer_id on payment_p2007_05
payment_5  (cost=0.28..8.34 rows=3 width=16)
         Index Cond: (customer_id = 599)
   ->  Index Scan using idx_fk_payment_p2007_06_customer_id on payment_p2007_06
payment_6  (cost=0.28..8.29 rows=1 width=16)
         Index Cond: (customer_id = 599)
   ->  Seq Scan on payment_p2007_07_max payment_7  (cost=0.00..3.95 rows=1 width=16)
         Filter: (customer_id = 599)
   ->  Seq Scan on payment_p0000_default payment_8  (cost=0.00..12.65 rows=1 width=16)
         Filter: (customer_id = 599)
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# SELECT customer_id, payment_date::timestamp(0), amount,
   coalesce(
      sum(amount) over (
          partition by customer_id
          order by payment_date asc
          rows between unbounded preceding and 1 preceding
      ),0) AS prior_spend
FROM payment WHERE customer_id = 599;
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
 WindowAgg  (cost=67.35..67.81 rows=24 width=56)
   ->  Sort  (cost=67.33..67.39 rows=24 width=16)
         Sort Key: payment.payment_date
         ->  Append  (cost=0.28..66.78 rows=24 width=16)
               ->  Index Scan using idx_fk_payment_p2007_01_customer_id on payment_p2007_01
payment_1  (cost=0.28..8.31 rows=2 width=16)
                     Index Cond: (customer_id = 599)
               ->  Index Scan using idx_fk_payment_p2007_02_customer_id on payment_p2007_02
payment_2  (cost=0.28..8.37 rows=5 width=16)
                     Index Cond: (customer_id = 599)
               ->  Index Scan using idx_fk_payment_p2007_03_customer_id on payment_p2007_03
payment_3  (cost=0.28..8.39 rows=6 width=16)
                     Index Cond: (customer_id = 599)
               ->  Index Scan using idx_fk_payment_p2007_04_customer_id on payment_p2007_04
payment_4  (cost=0.28..8.37 rows=5 width=16)
                     Index Cond: (customer_id = 599)
               ->  Index Scan using idx_fk_payment_p2007_05_customer_id on payment_p2007_05
payment_5  (cost=0.28..8.33 rows=3 width=16)
                     Index Cond: (customer_id = 599)
               ->  Index Scan using idx_fk_payment_p2007_06_customer_id on payment_p2007_06
payment_6  (cost=0.28..8.29 rows=1 width=16)
                     Index Cond: (customer_id = 599)
               ->  Seq Scan on payment_p2007_07_max payment_7  (cost=0.00..3.95 rows=1 width=16)
                     Filter: (customer_id = 599)
               ->  Seq Scan on payment_p0000_default payment_8  (cost=0.00..12.65 rows=1 width=16)
                     Filter: (customer_id = 599)
```

© 2025, Amazon Web Services, Inc. or its affiliates.

Running Aggregate

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

GENERATE SERIES



© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

```
pagila=# WITH rented AS
         (SELECT day, sum(coalesce(amount,0)) as bucks
          FROM generate_series('2007-04-01'::date,
                               '2007-04-30'::timestamp,
                               '1 day'::interval
               ) day
          LEFT JOIN payment ON ( day = date_trunc('day',payment_date) )
          GROUP BY day
          ORDER BY 1 ASC
         )
         SELECT
                day::date,
                bucks,
                sum(bucks) OVER t as "total for month",
                (avg(bucks) OVER (ROWS BETWEEN 7 preceding
                                  AND current row))::numeric(6,2) as "7 day avg"
         FROM
              rented WINDOW t AS
                     (ROWS BETWEEN unbounded preceding AND current row);
```

© 2025, Amazon Web Services, Inc. or its affiliates.

day	bucks	total for month	7 day avg
2007-04-01	550.65	550.65	550.65
2007-04-02	561.65	1112.30	556.15
2007-04-03	584.70	1697.00	565.66
2007-04-04	489.82	2186.82	546.70
2007-04-05	694.51	2881.33	576.26
2007-04-06	460.94	3342.27	557.04
2007-04-07	598.69	3940.96	562.99
2007-04-08	543.76	4484.72	560.59
2007-04-09	405.02	4889.74	542.38
2007-04-10	483.85	5373.59	532.66
2007-04-11	508.82	5882.41	523.17
2007-04-12	457.98	6340.39	519.19
2007-04-13	500.87	6841.26	494.99
2007-04-14	498.87	7340.13	499.73
2007-04-15	415.98	7756.11	476.89
2007-04-16	439.94	8196.05	463.91
2007-04-17	458.90	8654.95	470.65
2007-04-18	533.82	9188.77	476.89
2007-04-19	493.83	9682.60	475.02
2007-04-20	388.00	10070.60	466.27
2007-04-21	568.73	10639.33	474.75
2007-04-22	468.80	11108.13	471.00
2007-04-23	550.71	11658.84	487.84
2007-04-24	456.97	12115.81	489.97
2007-04-25	537.71	12653.52	499.82
2007-04-26	380.06	13033.58	480.60
2007-04-27	453.02	13486.60	475.50
2007-04-28	474.91	13961.51	486.36
2007-04-29	493.86	14455.37	477.00
2007-04-30	434.93	14890.30	472.77

(30 rows)

© 2025, Amazon Web Services, Inc. or its affiliates.

day	bucks	total for month	7 day avg
<snip>			
2007-04-11	508.82	5882.41	523.17
2007-04-12	457.98	6340.39	519.19
2007-04-13	500.87	6841.26	494.99
2007-04-14	498.87	7340.13	499.73
2007-04-15	415.98	7756.11	476.89
2007-04-16	439.94	8196.05	463.91
2007-04-17	458.90	8654.95	470.65
2007-04-18	533.82	9188.77	476.89
2007-04-19	493.83	9682.60	475.02
2007-04-20	388.00	10070.60	466.27
<snip>			

© 2025, Amazon Web Services, Inc. or its affiliates.

Thank you!

@robtreat2 rob@xzilla.net https://pgtreats.info

© 2025, Amazon Web Services, Inc. or its affiliates..