pg_partitioner

Erik Jones

Introduction

- What?
  - A tool for automatically generating range based partition tables.
- Why?
  - Seriously. How many list threads have you seen on how to partition?

High Level Features

- Integer & date type range based partitioning
- Data migration from parent to child tables
- Custom insert triggers generated for parent
- Test mode (nothing commits)

High Level Features

- Duplicates indexes from parent on children.
- Duplicates all constraints from parent on children – foreign keys are optional.
- For referencing fkeys user has choice of how to handle.

Stages

- create (default)
  - creates child partitions with no constraints or index
- migrate
  - moves data from parent to child tables
- post
  - create indexes and constraints

Example

```
               ^
postgres=# create database pagila;
CREATE DATABASE
Time: 221.764 ms
postgres=# \c pagila
You are now connected to database "pagila".
pagila=# \d rental
                                    Table "public.rental"
    Column    |            Type             |                         Modifiers
--------------+-----------------------------+-----------------------------------------------------------
 rental_id    | integer                     | not null default nextval('rental_rental_id_seq'::regclass)
 rental_date  | timestamp without time zone | not null
 inventory_id | integer                     | not null
 customer_id  | smallint                    | not null
 return_date  | timestamp without time zone |
 staff_id     | smallint                    | not null
 last_update  | timestamp without time zone | not null default now()
Indexes:
    "rental_pkey" PRIMARY KEY, btree (rental_id)
    "idx_unq_rental_rental_date_inventory_id_customer_id" UNIQUE, btree (rental_date, inventory_id, customer_id)
    "idx_fk_inventory_id" btree (inventory_id)
Foreign-key constraints:
    "rental_customer_id_fkey" FOREIGN KEY (customer_id) REFERENCES customer(customer_id) ON UPDATE CASCADE ON DELETE RESTRICT
    "rental_inventory_id_fkey" FOREIGN KEY (inventory_id) REFERENCES inventory(inventory_id) ON UPDATE CASCADE ON DELETE RESTRICT
    "rental_staff_id_fkey" FOREIGN KEY (staff_id) REFERENCES staff(staff_id) ON UPDATE CASCADE ON DELETE RESTRICT
Triggers:
    last_updated BEFORE UPDATE ON rental FOR EACH ROW EXECUTE PROCEDURE last_updated()

pagila=#
```

Example

```
mage@oblivion:~
 2391 $ pg_partitioner -d pagila rental rental_date --stage all
('20050501', '20050601')
Creating public.rental_20050501_20050601
Creating public.rental_20050601_20050701
Creating public.rental_20050701_20050801
Creating public.rental_20050801_20050901
Creating public.rental_20050901_20051001
Creating public.rental_20051001_20051101
Creating public.rental_20051101_20051201
Creating public.rental_20051201_20060101
Creating public.rental_20060101_20060201
Creating public.rental_20060201_20060301

Found fkey payment_p2007_01_rental_id_fkey on public.payment_p2007_01(rental_id) referencing rental(rental_id)
Would you like to:
1. Drop it
2. Replace it with a trigger
3. Abort
2

Found fkey payment_p2007_02_rental_id_fkey on public.payment_p2007_02(rental_id) referencing rental(rental_id)
Would you like to:
1. Drop it
2. Replace it with a trigger
3. Abort
```

Example

```
1. Drop it
2. Replace it with a trigger
3. Abort
2

Found fkey payment_p2007_05_rental_id_fkey on public.payment_p2007_05(rental_id) referencing rental(rental_id)
Would you like to:
1. Drop it
2. Replace it with a trigger
3. Abort
2

Found fkey payment_p2007_06_rental_id_fkey on public.payment_p2007_06(rental_id) referencing rental(rental_id)
Would you like to:
1. Drop it
2. Replace it with a trigger
3. Abort
2

Found fkey payment_rental_id_fkey on public.payment(rental_id) referencing rental(rental_id)
Would you like to:
1. Drop it
2. Replace it with a trigger
3. Abort
2
Moved 16044 rows into partitions.
Commtting everything.
mage@oblivion:~
 2392 $ 2
```

Example

```
 public | customer                 | table | postgres
 public | film                     | table | postgres
 public | film_actor               | table | postgres
 public | film_category            | table | postgres
 public | inventory                | table | postgres
 public | language                 | table | postgres
 public | payment                  | table | postgres
 public | payment_p2007_01         | table | postgres
 public | payment_p2007_02         | table | postgres
 public | payment_p2007_03         | table | postgres
 public | payment_p2007_04         | table | postgres
 public | payment_p2007_05         | table | postgres
 public | payment_p2007_06         | table | postgres
 public | rental                   | table | postgres
 public | rental_20050501_20050601 | table | mage
 public | rental_20050601_20050701 | table | mage
 public | rental_20050701_20050801 | table | mage
 public | rental_20050801_20050901 | table | mage
 public | rental_20050901_20051001 | table | mage
 public | rental_20051001_20051101 | table | mage
 public | rental_20051101_20051201 | table | mage
 public | rental_20051201_20060101 | table | mage
 public | rental_20060101_20060201 | table | mage
 public | rental_20060201_20060301 | table | mage
 public | staff                    | table | postgres
 public | store                    | table | postgres
(31 rows)

pagila=#
```

Example

```
al_date < '2005-07-01 00:00:00'::timestamp without time zone)
Inherits: rental

pagila=# \d foo
Did not find any relation named "foo".
pagila=# \d rental
                                    Table "public.rental"
    Column    |            Type             |                          Modifiers
--------------+-----------------------------+-------------------------------------------------------------
 rental_id    | integer                     | not null default nextval('rental_rental_id_seq'::regclass)
 rental_date  | timestamp without time zone | not null
 inventory_id | integer                     | not null
 customer_id  | smallint                    | not null
 return_date  | timestamp without time zone |
 staff_id     | smallint                    | not null
 last_update  | timestamp without time zone | not null default now()
Indexes:
    "rental_pkey" PRIMARY KEY, btree (rental_id)
    "idx_unq_rental_rental_date_inventory_id_customer_id" UNIQUE, btree (rental_date, inventory_id, customer_id)
    "idx_fk_inventory_id" btree (inventory_id)
Foreign-key constraints:
    "rental_customer_id_fkey" FOREIGN KEY (customer_id) REFERENCES customer(customer_id) ON UPDATE CASCADE ON DELETE RESTRICT
    "rental_inventory_id_fkey" FOREIGN KEY (inventory_id) REFERENCES inventory(inventory_id) ON UPDATE CASCADE ON DELETE RESTRICT
    "rental_staff_id_fkey" FOREIGN KEY (staff_id) REFERENCES staff(staff_id) ON UPDATE CASCADE ON DELETE RESTRICT
Triggers:
    last_updated BEFORE UPDATE ON rental FOR EACH ROW EXECUTE PROCEDURE last_updated()
    rental_partition_trigger BEFORE INSERT OR UPDATE ON rental FOR EACH ROW EXECUTE PROCEDURE rental_ins_trig()

pagila=#
```

Example

```
 public | payment_p2007_05         | table | postgres
 public | payment_p2007_06         | table | postgres
 public | rental                   | table | postgres
 public | rental_20050501_20050601 | table | mage
 public | rental_20050601_20050701 | table | mage
 public | rental_20050701_20050801 | table | mage
 public | rental_20050801_20050901 | table | mage
 public | rental_20050901_20051001 | table | mage
 public | rental_20051001_20051101 | table | mage
pagila=# \d rental_20050801_20050901
                          Table "public.rental_20050801_20050901"
    Column    |            Type             |                          Modifiers
--------------+-----------------------------+-------------------------------------------------------------
 rental_id    | integer                     | not null default nextval('rental_rental_id_seq'::regclass)
 rental_date  | timestamp without time zone | not null
 inventory_id | integer                     | not null
 customer_id  | smallint                    | not null
 return_date  | timestamp without time zone |
 staff_id     | smallint                    | not null
 last_update  | timestamp without time zone | not null default now()
Indexes:
    "rental_20050801_20050901_pkey" PRIMARY KEY, btree (rental_id)
    "rental_20050801_20050901_idx_fk_inventory_id" btree (inventory_id)
Check constraints:
    "rental_20050801_20050901_rental_date_check" CHECK (rental_date >= '2005-08-01 00:00:00'::timestamp without time zone AND rent
al_date < '2005-09-01 00:00:00'::timestamp without time zone)
Inherits: rental

pagila=#
```

Trigger FunctionTemplate

```
CREATE OR REPLACE FUNCTION %(table_name)s_ins_func(rec %(table_name)s)
    RETURNS %(table_name)s AS $$
DECLARE
    partition varchar;
    name_parts varchar[];
    upper_dim integer;
    ins_sql varchar;
BEGIN
    FOR partition IN
        SELECT * FROM get_table_partitions('%(table_name)s')
    LOOP
        name_parts := string_to_array(partition, '_');
        upper_dim := array_upper(name_parts, 1);
        IF rec.%(part_column)s >= name_parts[upper_dim-1]::%(col_type)s
                AND rec.%(part_column)s < name_parts[upper_dim]::%(col_type)s THEN
            ins_sql := 'INSERT INTO %(table_name)s_' || name_parts[upper_dim-1] || '_' ||
                        name_parts[upper_dim] || ' (%(table_atts)s) VALUES (' || %(atts_vals)s || ');';
            EXECUTE ins_sql;
            RETURN NULL;
        END IF;
    END LOOP;
    RAISE WARNING 'No partition created for %(table_name)s to hold value %(col_type)s %%, leaving data in parent table.',
rec.%(part_column)s;
    RETURN rec;
END;
$$ language plpgsql;
```

TODO/Future

- partitioner catalog tables
- partition “policies”
- more (many more) tests
- some code cleanup/refactoring
- extend to “sharding”