Мамонов Антон

Пример

```
postgres=#  \c db21
Вы подключены к базе данных "db21" как пользователь "postgres".
db21=#
db21=# create sequence city_seq;
CREATE SEQUENCE
db21=# create table city_msk (id   int default nextval('city_seq'::regclass),city text,constraint check_city CHECK (city = 'MSK'));
CREATE TABLE
db21=# insert into city_msk (city) values ('MSK');
INSERT 0 1
db21=# explain (costs off) select * from city_msk where city = 'SPB';
            QUERY PLAN
-------------------------------
 Seq Scan on city_msk
   Filter: (city = 'SPB'::text)
(2 строки)

db21=#  set constraint_exclusion to on;
SET
db21=# explain (costs off) select * from city_msk where city = 'SPB';
        QUERY PLAN
-------------------------
 Result
   One-Time Filter: false
(2 строки)

db21=# reset constraint_exclusion;
RESET
db21=# select name, setting, enumvals from pg_settings where name = 'constraint_exclusion';
         name         |  setting  |      enumvals
----------------------+-----------+--------------------
 constraint_exclusion | partition | {partition,on,off}
(1 строка)

db21=# create table city_spb (id int default nextval('city_seq'::regclass) ,city text,constraint check_city CHECK (city = 'SPB'));
CREATE TABLE
db21=# insert into city_spb (city) values ('SPB');
INSERT 0 1
db21=# create view cities_v (id, city) as select id, city from city_msk union all select id, city from city_spb;
CREATE VIEW
db21=# explain (costs off) select * from cities_v where city = 'SPB';
            QUERY PLAN
```

```
db21=# create view cities_v (id, city) as select id, city from city_msk union all select id, city from city_spb;
CREATE VIEW
db21=# explain (costs off) select * from cities_v where city = 'SPB';
            QUERY PLAN
-------------------------------
 Seq Scan on city_spb
   Filter: (city = 'SPB'::text)
(2 строки)

db21=# explain (costs off) select * from cities_v where city = 'KEM';
        QUERY PLAN
-------------------------
 Result
   One-Time Filter: false
(2 строки)

db21=# create table cities (  id int default nextval('city_seq'::regclass),city text, ,constraint dummy_check check (city = '') no i
nherit);
ОШИБКА:  ошибка синтаксиса (примерное положение: ",")
СТРОКА 1: ... default nextval('city_seq'::regclass),city text, ,constrain...
                                                             ^
db21=# create table cities (  id int default nextval('city_seq'::regclass),city text ,constraint dummy_check check (city = '') no in
herit);
CREATE TABLE
db21=# alter table city_msk inherit cities;
ALTER TABLE
db21=# alter table city_spb inherit cities;
ALTER TABLE
db21=# create table city_kem() inherits (cities);
CREATE TABLE
db21=# insert into city_kem (city) values ('KEM');
INSERT 0 1
db21=# \d city_kem

[1]+  Остановлен    psql -Upostgres
postgres@docker:/home/user$ psql -U postgres
Пароль пользователя postgres:
psql (13.1 (Ubuntu 13.1-1.pgdg20.04+1))
Введите "help", чтобы получить справку.
```

```
psql (13.1 (Ubuntu 13.1-1.pgdg20.04+1))
Введите "help", чтобы получить справку.

postgres=# select * from cities;
ОШИБКА:  отношение "cities" не существует
СТРОКА 1: select * from cities;
                       ^
postgres=# \c db21
Вы подключены к базе данных "db21" как пользователь "postgres".
db21=# \c db21
Вы подключены к базе данных "db21" как пользователь "postgres".
db21=# select * from cities;
 id | city
----+------
  1 | MSK
  2 | SPB
  3 | KEM
(3 строки)

db21=# select * from ONLY cities;
 id | city
----+------
(0 строк)

db21=# explain (costs off) select * from cities where city = 'SPB';
             QUERY PLAN
------------------------------------
 Append
   ->  Seq Scan on city_spb cities_1
         Filter: (city = 'SPB'::text)
   ->  Seq Scan on city_kem cities_2
         Filter: (city = 'SPB'::text)
(5 строк)

db21=# alter table city_kem add constraint check_city check (city = 'KEM');
ALTER TABLE
db21=#  explain (costs off) delete from cities where city = 'SPB';
             QUERY PLAN
------------------------------------
 Delete on cities
```

```
db21=# alter table city_kem add constraint check_city check (city = 'KEM');
ALTER TABLE
db21=#  explain (costs off) delete from cities where city = 'SPB';
              QUERY PLAN
---------------------------------------
 Delete on cities
   Delete on city_spb cities_1
   ->  Seq Scan on city_spb cities_1
         Filter: (city = 'SPB'::text)
(4 строки)

db21=# create function city_ins () returns trigger as $$
db21$#          begin
db21$#             if new.city = 'MSK' then
db21$#                insert into city_msk select new.*;
db21$#             elsif new.city = 'SPB' then
db21$#                insert into city_spb select new.*;
db21$#             elsif new.city = 'KEM' then
db21$#                insert into city_kem select new.*;
db21$#             end if;
db21$#             return null;
db21$#          end; $$ language plpgsql;
CREATE FUNCTION
db21=# create trigger city_partition_ins
db21-#             before insert on cities
db21-#             for each row
db21-#             execute procedure city_ins();
CREATE TRIGGER
db21=# insert into cities (city) values ('MSK'), ('SPB'), ('KEM');
INSERT 0 0
db21=# select tableoid::regclass, * from cities;
 tableoid | id | city
----------+----+------
 city_msk |  1 | MSK
 city_msk |  4 | MSK
 city_spb |  2 | SPB
 city_spb |  5 | SPB
 city_kem |  3 | KEM
 city_kem |  6 | KEM
(6 строк)
```

Практика

Создание БД, таблиц и их заполнение.

```
postgres=# create database db21;
CREATE DATABASE
postgres=# \c db21
Вы подключены к базе данных "db21" как пользователь "postgres".
db21=# create table dates (ts timestamp);
CREATE TABLE
db21=# insert into dates select t.ts from generate_series ('2016-01-01'::timestamp
'2016-12-31'::timestamp ,'60 min'::interval) as t (ts);
INSERT 0 8761
db21=#
```

Создание дочерних таблиц

```
db21=# create table dates_2016q1 () inherits (dates);
CREATE TABLE
db21=# create table dates_2016q2 () inherits (dates);
CREATE TABLE
db21=# create table dates_2016q3 () inherits (dates);
CREATE TABLE
db21=# create table dates_2016q4 () inherits (dates);
CREATE TABLE
db21=# alter table dates_2016q1 add constraint check_ts check (ts >= '2016-01-01 00:
00:00'::timestamp and ts <  '2016-04-01 00:00:00'::timestamp);
ALTER TABLE
db21=# alter table dates_2016q2 add constraint check_ts check (ts >= '2016-04-01 00:
00:00'::timestamp and ts <  '2016-07-01 00:00:00'::timestamp);
ALTER TABLE
db21=# alter table dates_2016q3 add constraint check_ts check (ts >= '2016-07-01 00:
00:00'::timestamp and ts <  '2016-10-01 00:00:00'::timestamp);
ALTER TABLE
db21=# alter table dates_2016q4 add constraint check_ts check (ts >= '2016-10-01 00:
00:00'::timestamp and ts <  '2017-10-01 00:00:00'::timestamp);
ALTER TABLE
db21=#
```

Создание процедуры и функции

```
ALTER TABLE
db21=# create function dates_ins () returns trigger as $$
db21$# begin
db21$# if new.ts >= '2016-01-01 00:00:00'::timestamp and
db21$# new.ts <  '2016-04-01 00:00:00'::timestamp
db21$# then
db21$# insert into dates_2016q1 select new.*;
db21$# elsif new.ts >= '2016-04-01 00:00:00'::timestamp and
db21$# new.ts <  '2016-07-01 00:00:00'::timestamp
db21$# then
db21$# insert into dates_2016q2 select new.*;
db21$# elsif new.ts >= '2016-07-01 00:00:00'::timestamp and
db21$# new.ts <  '2016-10-01 00:00:00'::timestamp
db21$# then
db21$# insert into dates_2016q3 select new.*;
db21$# elsif new.ts >= '2016-10-01 00:00:00'::timestamp and
db21$# new.ts <  '2017-01-01 00:00:00'::timestamp
db21$# then
db21$# insert into dates_2016q4 select new.*;
db21$# end if;
db21$# return null;
db21$# end; $$ language plpgsql;
CREATE FUNCTION
db21=#
db21=# create trigger dates_ins_trg before insert on dates for each row execute
procedure dates_ins();
CREATE TRIGGER
db21=#
```

Проверка

Перенос данных

```
db21=# create trigger dates_ins_trg before insert on dates for each row execute
procedure dates_ins();
CREATE TRIGGER
db21=#
db21=#
db21=# insert into dates values ('2016-02-01');
INSERT 0 0
db21=# insert into dates values ('2016-05-01');
INSERT 0 0
db21=# insert into dates values ('2016-08-01');
INSERT 0 0
db21=# insert into dates values ('2016-11-01');
INSERT 0 0
db21=# select tableoid::regclass, count(*) from dates group by tableoid;
   tableoid    | count
---------------+-------
 dates_2016q1 |     1
 dates_2016q2 |     1
 dates_2016q4 |     1
 dates_2016q3 |     1
 dates        |  8761
(5 ёЄЁюъ)
```

```
db21=# with del as (delete from ONLY dates  where ts >= '2016-01-01 00:00:00'::timestamp
 and ts <  '2016-04-01 00:00:00'::timestamp returning *) insert into dates_2016q1 select
 * from del;
INSERT 0 2184
db21=#
db21=#
db21=# with del as (delete from ONLY dates where ts >= '2016-04-01 00:00:00'::timestamp
and ts <  '2016-07-01 00:00:00'::timestamp returning *) insert into dates_2016q2 select
* from del;
INSERT 0 2184
db21=# with del as (delete from ONLY dates where ts >= '2016-07-01 00:00:00'::timestamp
and ts <  '2016-10-01 00:00:00'::timestamp returning *) insert into dates_2016q3 select
* from del;
INSERT 0 2208
db21=# with del as (delete from ONLY dates where ts >= '2016-10-01 00:00:00'::timestamp
and ts <  '2017-01-01 00:00:00'::timestamp returning *) insert into dates_2016q4 select
* from del;
INSERT 0 2185
db21=#
db21=# select tableoid::regclass, count(*) from dates group by tableoid;
   tableoid    | count
---------------+-------
 dates_2016q1 |  2185
 dates_2016q2 |  2185
 dates_2016q4 |  2186
 dates_2016q3 |  2209
(4 ёЄЁюъш)


db21=# explain (costs off) select * from dates where ts < '2016-05-09'::timestamp;
                              QUERY PLAN
-----------------------------------------------------------------------
 Append
   ->  Seq Scan on dates dates_1
         Filter: (ts < '2016-05-09 00:00:00'::timestamp without time zone)
   ->  Seq Scan on dates_2016q1 dates_2
         Filter: (ts < '2016-05-09 00:00:00'::timestamp without time zone)
   ->  Seq Scan on dates_2016q2 dates_3
         Filter: (ts < '2016-05-09 00:00:00'::timestamp without time zone)
(7 ёЄЁюъ)
```