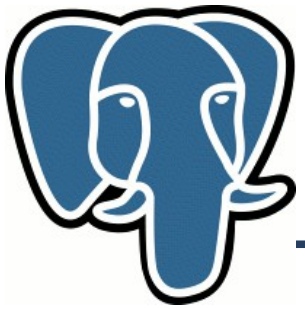
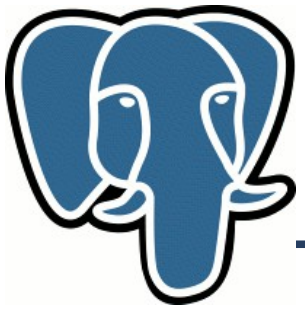


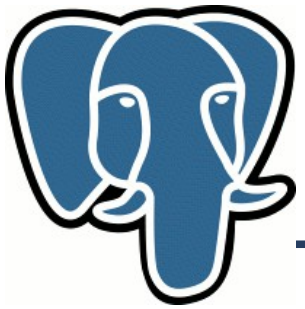
Расширяемость PostgreSQL

Федор Сигаев
Mail.Ru Group



Agenda

- Простой пример
- Что еще можно
- Проблемы

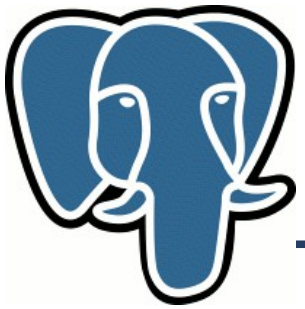


Расстояние круг - точка

```
# SELECT circle(point(1,2), 3);
  circle
-----------
 <(1,2),3>

# SELECT circle(point(1,2), 3) <-> point(20, 0);
ERROR:  operator does not exist: circle <-> point
LINE 1: select circle(point(1,2), 3) <-> point(20, 0);
                                     ^

HINT:  No operator matches the given name and argument type(s). You
might need to add explicit type casts.
```

А-а-а-а, что делать?

```
# select c <-> p from circle(point(1,2), 3) as c, point(1,1) as p;
```

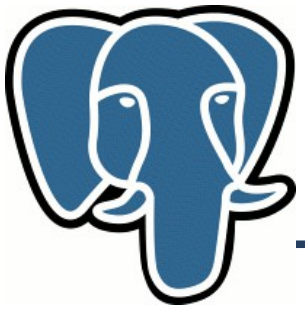


Расстояние: что есть?

```
# \do <->
```

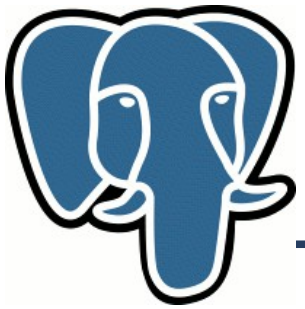
List of operators

Schema	Name	Left arg type	Right arg type	Result type	Description
pg_catalog	<->	box	box	double precision	distance between
pg_catalog	<->	circle	circle	double precision	distance between
pg_catalog	<->	circle	polygon	double precision	distance between
pg_catalog	<->	line	box	double precision	distance between
pg_catalog	<->	line	line	double precision	distance between
pg_catalog	<->	lseg	box	double precision	distance between
pg_catalog	<->	lseg	line	double precision	distance between
pg_catalog	<->	lseg	lseg	double precision	distance between
pg_catalog	<->	path	path	double precision	distance between
pg_catalog	<->	point	box	double precision	distance between
pg_catalog	<->	point	circle	double precision	distance between
pg_catalog	<->	point	line	double precision	distance between
pg_catalog	<->	point	lseg	double precision	distance between
pg_catalog	<->	point	path	double precision	distance between
pg_catalog	<->	point	point	double precision	distance between
pg_catalog	<->	polygon	polygon	double precision	distance between



Расстояние: грубая сила

- ```
  # SELECT @@ c <-> p FROM
        circle(point(1,2), 3) AS c, point(20,1) AS p;
       ?column?
  ------------------
  19.0262975904404
  ```
- Минус радиус
- ```
  # SELECT @@ c <-> p - radius(c) FROM
        circle(point(1,2), 3) AS c, point(20,1) AS p;
  ERROR:  operator does not exist: point - double
  precision
  LINE 1: SELECT @@ c <-> p - radius(c) FROM
  circle(point(1,2), 3) AS …
                               ^
  HINT:  No operator matches the given name and argument
  type(s). You might need to add explicit type casts.
  ```
- Ой

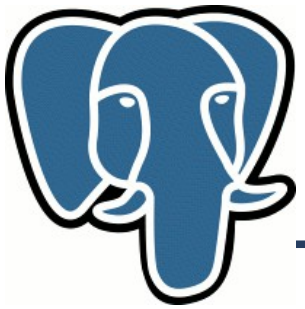


Расстояние: грубая сила

Оператор – имеет больший приоритет.
Точка минус число - ???

- # SELECT (@@ c <-> p) - radius(c) FROM
  circle(point(1,2), 3) AS c, point(20,1) AS p;
  ?column?
```
------------------
 16.0262975904404
```
- # SELECT (@@ c <-> p) - radius(c) FROM
  circle(point(1,2), 3) AS c, point(2,1) AS p;
  ?column?
```
------------------
 -1.5857864376269
```

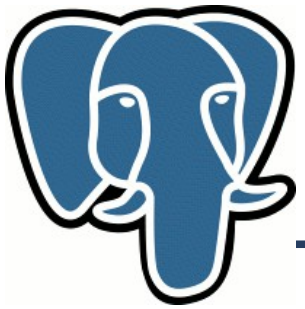
- Ой



Расстояние: грубая сила

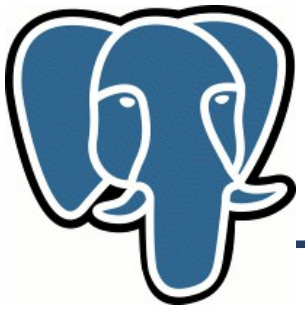
```
# SELECT
  CASE WHEN d.d < 0.0 THEN 0 ELSE d.d END
  FROM
   (
     SELECT (@@ c <-> p) - radius(c) AS d FROM
       circle(point(1,2), 3) AS c, point(2,1) AS p
   ) AS d;
 d
---
 0
```

Пошли корячить нашу ORM? Не, ну на ...



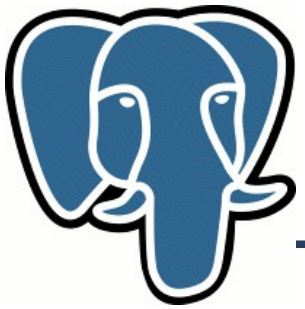
Расстояние: упрощение

```
CREATE OR REPLACE FUNCTION
   circle_point_distance(c circle, p point)
RETURNS float8 AS
$$
 SELECT
    CASE WHEN d.d < 0.0 THEN 0 ELSE d.d END  FROM
        (SELECT (@@ c <-> p) - radius(c) AS d) AS d;
$$
LANGUAGE SQL IMMUTABLE;
COMMENT ON FUNCTION circle_point_distance(circle, point)
   IS 'distance between';


# SELECT circle_point_distance(circle(point(1,2), 3), point(20,1));
 circle_point_distance
-----------------------
      16.026975904404
```



Расстояние: упрощение

```
# CREATE OPERATOR <-> (
    LEFTARG = circle,
    RIGHTARG = point,
    PROCEDURE = circle_point_distance
);

# COMMENT ON OPERATOR <-> (circle, point) IS
                'distance between';

#  SELECT circle(point(1,2), 3) <-> point(20,1);
     ?column?
------------------
 16.0262975904404
```

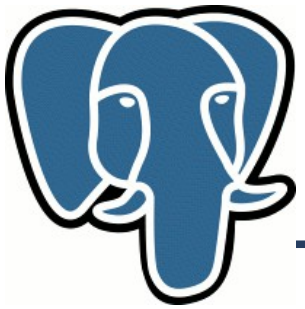


Расстояние: WOW!

```
# \do <->
```

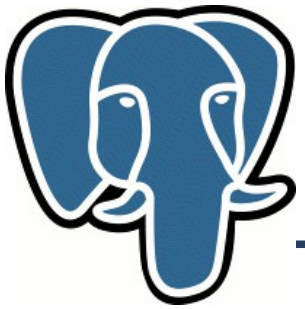
List of operators

Schema	Name	Left arg type	Right arg type	Result type	Description
pg_catalog	<->	box	box	double precision	distance between
pg_catalog	<->	circle	circle	double precision	distance between
pg_catalog	<->	circle	polygon	double precision	distance between
pg_catalog	<->	line	box	double precision	distance between
pg_catalog	<->	line	line	double precision	distance between
pg_catalog	<->	lseg	box	double precision	distance between
pg_catalog	<->	lseg	line	double precision	distance between
pg_catalog	<->	lseg	lseg	double precision	distance between
pg_catalog	<->	path	path	double precision	distance between
pg_catalog	<->	point	box	double precision	distance between
pg_catalog	<->	point	circle	double precision	distance between
pg_catalog	<->	point	line	double precision	distance between
pg_catalog	<->	point	lseg	double precision	distance between
pg_catalog	<->	point	path	double precision	distance between
pg_catalog	<->	point	point	double precision	distance between
pg_catalog	<->	polygon	polygon	double precision	distance between
public	<->	circle	point	double precision	distance between



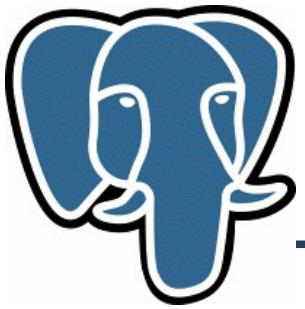
Расстояние: результаты

- Замели сложность под коврик. Все работает как настоящее
- Общности для можно сделать коммутатор
- Овраги!!! Надо делать побыстрее



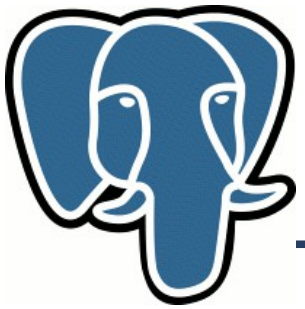
Расстояние: модуль

```
cpdistance.c
cpdistance--1.0.sql
cpdistance.control
Makefile
sql/cpdistance.sql
expected/cpdistance.out
```



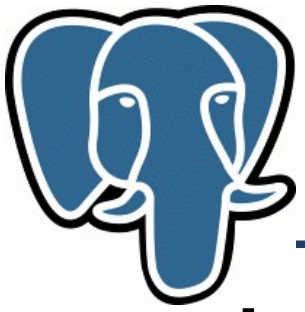
Скелет cpdistance.c

```
#include <postgres.h>
#include <utils/geo_decls.h>

PG_MODULE_MAGIC;

PG_FUNCTION_INFO_V1(circle_point_distance);
Datum   circle_point_distance(PG_FUNCTION_ARGS);
Datum
circle_point_distance(PG_FUNCTION_ARGS)
{
    ...
}
```



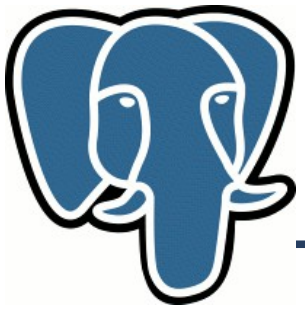
Мясо cpdistance.c

```
Datum
circle_point_distance(PG_FUNCTION_ARGS)
{
        CIRCLE  *c = PG_GETARG_CIRCLE_P(0);
        Point       *p = PG_GETARG_POINT_P(1);
        float8       distance;

        distance = DatumGetFloat8(
                        DirectFunctionCall2(
                                point_distance,
                                PointPGetDatum(p),
                                PointPGetDatum(&c->center)
                        )
                );

        if (FPle(distance, c->radius))
                distance = 0.0;
        else
                distance -= c->radius;

        PG_RETURN_FLOAT8(distance);
}
```



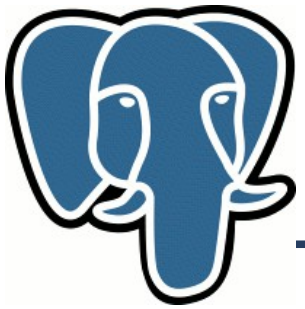
cpdistance—1.0.sql

```
-- complain if script is sourced in psql, rather than via
-- CREATE EXTENSION
\echo Use "CREATE EXTENSION cpdistance" to load this file. \quit

CREATE FUNCTION circle_point_distance(circle, point)
        RETURNS float8
        AS 'MODULE_PATHNAME', 'circle_point_distance'
        LANGUAGE C IMMUTABLE;

COMMENT ON FUNCTION circle_point_distance(circle, point) IS
             'distance between';

CREATE OPERATOR <-> (
        LEFTARG = circle,
        RIGHTARG = point,
        PROCEDURE = circle_point_distance
);

COMMENT ON OPERATOR <-> (circle, point) IS
               'distance between';
```



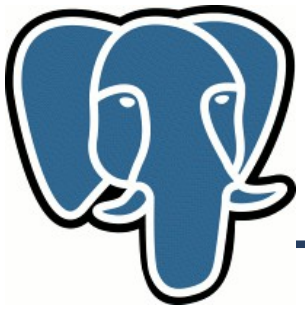
cpdistance.control

```
comment = 'Circle - point distance'
default_version = '1.0'
module_pathname = '$libdir/cpdistance'
relocatable = true
```



Makefile

```
EXTENSION = cpdistance
MODULES = cpdistance
DATA = cpdistance--1.0.sql
OBJS = cpdistance.o
REGRESS = cpdistance

ifdef USE_PGXS
PGXS := $(shell pg_config --pgxs)
include $(PGXS)
else
subdir = contrib/mchar
top_builddir = ../..
include $(top_builddir)/src/Makefile.global
include $(top_srcdir)/contrib/contrib-global.mk
endif
```



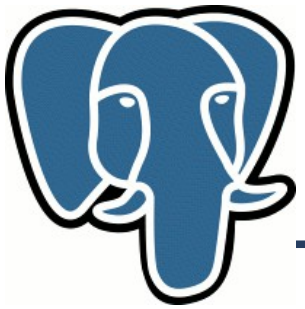
Тесты

sql/cpdistance.sql:

```
CREATE EXTENSION cpdistance;

SELECT circle(point(1,2), 3) <-> point(20,1);
```

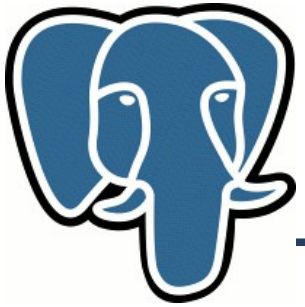
expected/cpdistance.out:

```
CREATE EXTENSION cpdistance;
SELECT circle(point(1,2), 3) <-> point(20,1);
     ?column?
------------------
 16.0262975904404
(1 row)
```



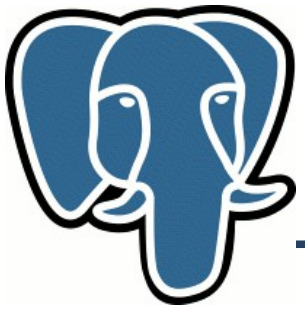
Компиляция и установка

```
% make USE_PGXS=1 all
cc -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement
-Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-
aliasing -fwrapv -g -O0 -fPIC -DPIC -I. -I./ -I/usr/local/pgsql/include/server
-I/usr/local/pgsql/include/internal   -c -o cpdistance.o cpdistance.c -MMD -MP
-MF .deps/cpdistance.Po
cc -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement
-Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-
aliasing -fwrapv -g -O0 -fPIC -DPIC -L/usr/local/pgsql/lib -Wl,--as-needed
-Wl,-R'/usr/local/pgsql/lib'  -shared -o cpdistance.so cpdistance.o

% su2 -c 'make USE_PGXS=1 install'
/usr/local/bin/gmkdir -p '/usr/local/pgsql/share/extension'
/usr/local/bin/gmkdir -p '/usr/local/pgsql/share/extension'
/usr/local/bin/gmkdir -p '/usr/local/pgsql/lib'
/usr/local/bin/ginstall -c -m 644 cpdistance.control
'/usr/local/pgsql/share/extension/'
/usr/local/bin/ginstall -c -m 644 cpdistance--1.0.sql
'/usr/local/pgsql/share/extension/'
/usr/local/bin/ginstall -c -m 755  cpdistance.so '/usr/local/pgsql/lib/'
```



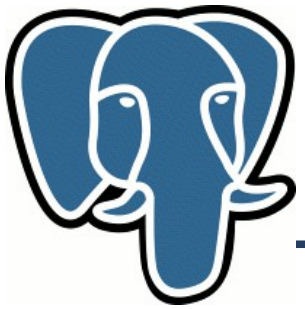
Тест

```
% make USE_PGXS=1 installcheck
/usr/local/pgsql/lib/pgxs/src/makefiles/../../src/test/regress/
pg_regress --inputdir=./ --psqldir='/usr/local/pgsql/bin'
--dbname=contrib_regression cpdistance
(using postmaster on Unix socket, default port)
============== dropping database "contrib_regression" ==============
DROP DATABASE
============== creating database "contrib_regression" ==============
CREATE DATABASE
ALTER DATABASE
============== running regression test queries        ==============
test cpdistance                   ... ok

=====================
 All 1 tests passed.
=====================
```



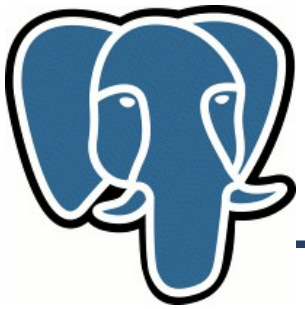
Установка

```
postgres=# CREATE EXTENSION cpdistance;
CREATE EXTENSION

postgres=# \dx
                    List of installed extensions
    Name    | Version |   Schema   |         Description
------------+---------+------------+------------------------------
 cpdistance | 1.0     | public     | Circle - point distance
 plpgsql    | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
```



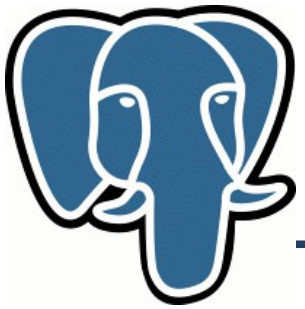
Глянуть

```
postgres=# \dx+
      Objects in extension "cpdistance"
               Object Description
-----------------------------------------------
 function circle_point_distance(circle,point)
 operator <->(circle,point)
(2 rows)

      Objects in extension "plpgsql"
            Object Description
-------------------------------------------
 function plpgsql_call_handler()
 function plpgsql_inline_handler(internal)
 function plpgsql_validator(oid)
 language plpgsql
```



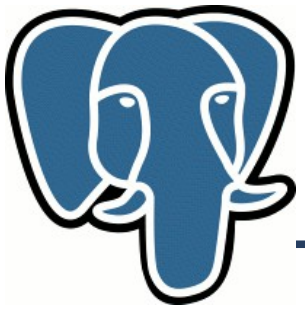
Что еще

- Типы, операции, функции, агрегаты, оконные агрегаты (полнотесктовый поиск первоначально был реализован как модуль! PostGIS и сейчас модуль)
- Языки (sql,pl/pgsql,pl/perl,pl/python, pl/tcl, pl/R, pl/java …,pl/v8). Pl/psql от EnterpriseDB!
  pl/sh!!!
- Встроенные индексы (Btree, Hash, GiST, GIN, SP-GiST, BRIN)
- Новые типы индексов (http://sigaev.ru/misc/bloom-0.4.tar.gz)
- Внешние таблицы (csv, oracle, mysql, informix, couchdb, redis, mongodb, twitter, www ...)
- Hooks



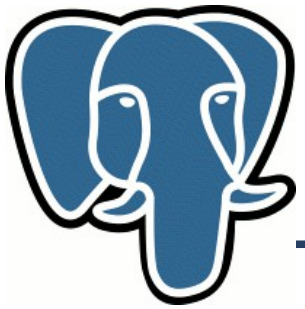
Но как?

```
% psql -E contrib_regression
...
contrib_regression=# \do <->
********* QUERY **********
SELECT n.nspname as "Schema",
  o.oprname AS "Name",
  CASE WHEN o.oprkind='l' THEN NULL ELSE pg_catalog.format_type(o.oprleft, NULL) END AS
                                                                          "Left arg type",
  CASE WHEN o.oprkind='r' THEN NULL ELSE pg_catalog.format_type(o.oprright, NULL) END AS
                                                                         "Right arg type",
  pg_catalog.format_type(o.oprresult, NULL) AS "Result type",
  coalesce(pg_catalog.obj_description(o.oid, 'pg_operator'),
           pg_catalog.obj_description(o.oprcode, 'pg_proc')) AS "Description"
FROM pg_catalog.pg_operator o
     LEFT JOIN pg_catalog.pg_namespace n ON n.oid = o.oprnamespace
WHERE o.oprname ~ '^(<->)$'
  AND pg_catalog.pg_operator_is_visible(o.oid)
ORDER BY 1, 2, 3, 4;
**************************
                                  List of operators
   Schema   | Name | Left arg type | Right arg type |   Result type    |   Description
------------+------+---------------+----------------+------------------+-----------------
...
 pg_catalog | <->  | box           | box            | double precision | distance between
...
```



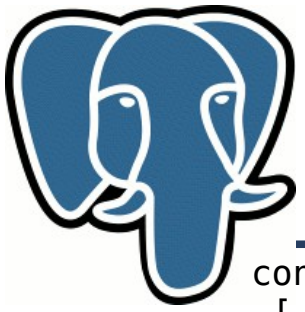
pg_operator

```
contrib_regression=# \d pg_operator
   Table "pg_catalog.pg_operator"
    Column    |  Type   | Modifiers
--------------+---------+-----------
 oprname      | name    | not null
 oprnamespace | oid     | not null
 oprowner     | oid     | not null
 oprkind      | "char"  | not null
 oprcanmerge  | boolean | not null
 oprcanhash   | boolean | not null
 oprleft      | oid     | not null
 oprright     | oid     | not null
 oprresult    | oid     | not null
 oprcom       | oid     | not null
 oprnegate    | oid     | not null
 oprcode      | regproc | not null
 oprrest      | regproc | not null
 oprjoin      | regproc | not null
Indexes:
    "pg_operator_oid_index" UNIQUE, btree (oid)
    "pg_operator_oprname_l_r_n_index" UNIQUE, btree
                (oprname, oprleft, oprright, oprnamespace)
```



pg_operator.<->

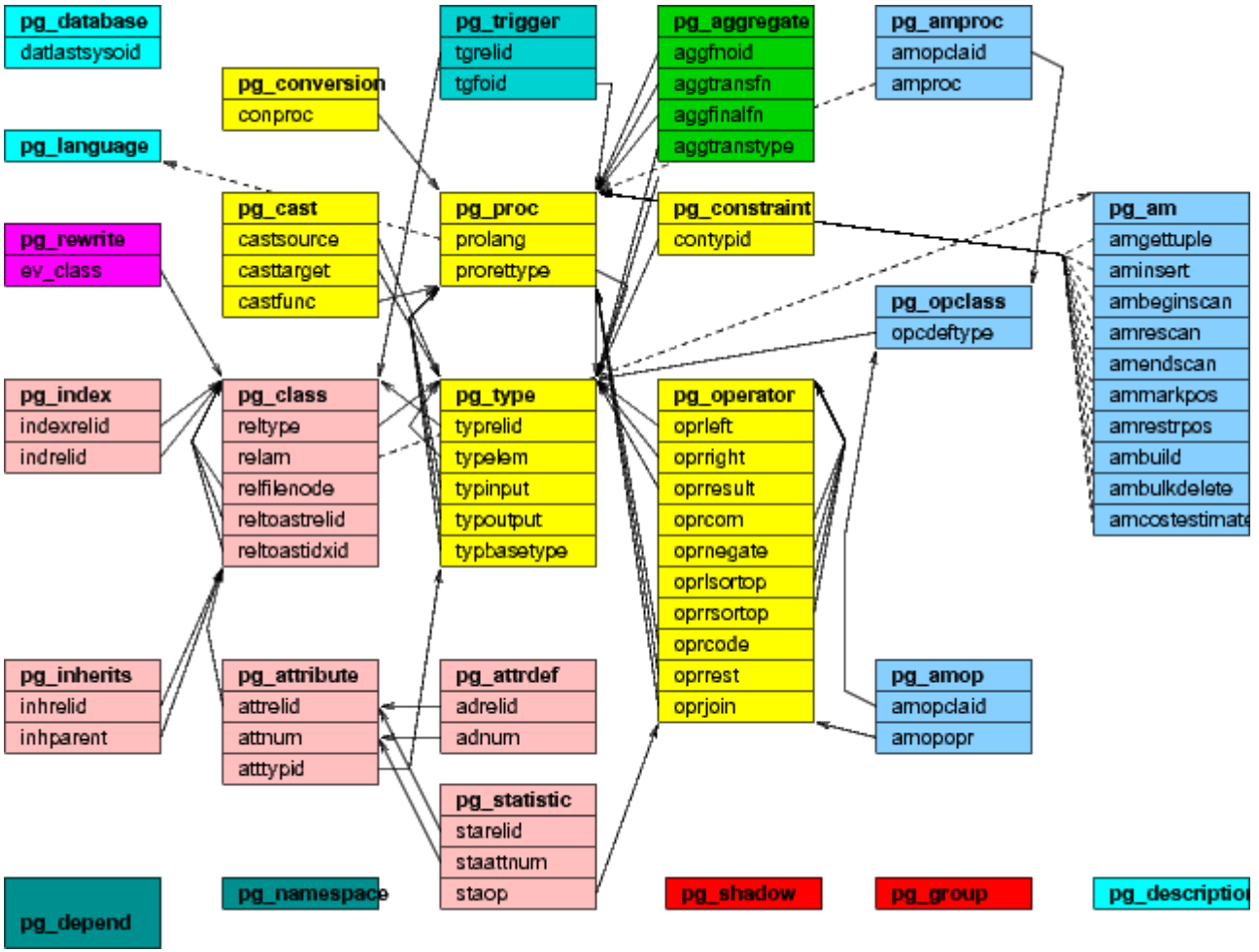
```
contrib_regression=# \x
Expanded display is on.
contrib_regression=# select o.oid, o.*
       from pg_operator o, pg_namespace n
       where o.oprname = '<->' and o.oprnamespace = n.oid and
          n.nspname = 'public';
-[ RECORD 1 ]+----------------------
oid          | 16390
oprname      | <->
oprnamespace | 2200
oprowner     | 16384
oprkind      | b
oprcanmerge  | f
oprcanhash   | f
oprleft      | 718
oprright     | 600
oprresult    | 701
oprcom       | 0
oprnegate    | 0
oprcode      | circle_point_distance
oprrest      | -
oprjoin      | -
```



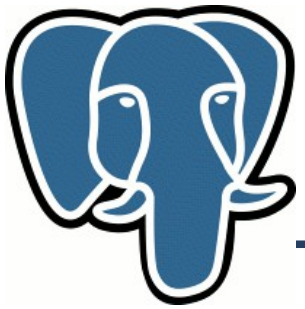
pg_proc.circle_point_distance

```
contrib_regression=# select * from pg_proc where proname = 'circle_point_distance';
-[ RECORD 1 ]---+----------------------
proname         | circle_point_distance
pronamespace    | 2200
proowner        | 16384
prolang         | 13
procost         | 1
prorows         | 0
provariadic     | 0
protransform    | -
proisagg        | f
proiswindow     | f
prosecdef       | f
proleakproof    | f
proisstrict     | f
proretset       | f
provolatile     | i
pronargs        | 2
pronargdefaults | 0
prorettype      | 701
proargtypes     | 718 600
proallargtypes  |
proargmodes     |
proargnames     |
proargdefaults  |
prosrc          | circle_point_distance
probin          | $libdir/cpdistance
proconfig       |
proacl          |
```

Барон Мюнхгаузен

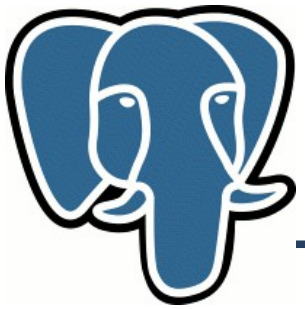


52 таблицы



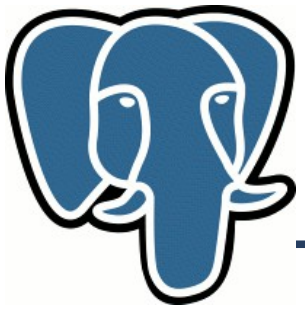
BTW: information schema

- ISO/IEC 9075-11
- http://www.postgresql.org/docs/devel/static/information-schema.html



Всё?

- Доступ к WAL (индексы, в процессе)
- Hooks
- Репликация ?!



Ссылки

http://www.postgresql.org/developer

http://www.postgresql.org/docs/devel/static

https://wiki.postgresql.org/wiki/Developer_FAQ

http://www.postgresql.org/docs/devel/static/git.html
http://git.postgresql.org/gitweb?p=postgresql.git

http://sigaev.ru/misc/cpdistance-1.0.tgz

Спасибо!

Задавайте вопросов, постараюсь найти ответов