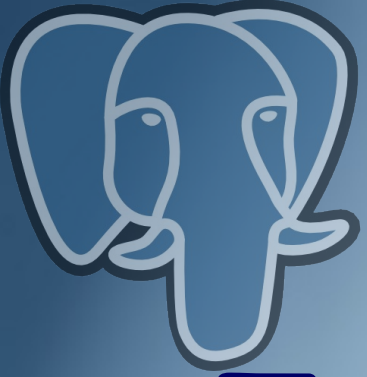


# Полнотекстовый поиск в PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

Олег Бартунов, Фёдор Сигаев
ГАИШ-МГУ,
PostgreSQL Global Development Group,
Major Developers
oleg@sai.msu.su

# Quick FTS primer

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
postgres=# \d apod Астрономическая Картинка Дня - APOD (1754)
        Table "public.apod"
  Column   |   Type    | Modifiers
-----------+-----------+-----------
 id        | integer   | not null
 title     | text      |
 body      | text      |
 sdate     | date      |
 keywords  | text      |
Indexes:
    "apod_pkey" PRIMARY KEY, btree (id)
```

**Полнотекстовый поиск в одну команду в PostgreSQL 8.3+**

```
postgres=# create index apod_title_idx on apod using gin(title);

postgres=# select title from apod where title @@ 'x-ray' limit 5;
          title
-------------------------------------
 The X-Ray Moon
 Vela Supernova Remnant in X-ray
 Tycho's Supernova Remnant in X-ray
 ASCA X-Ray Observatory
 Unexpected X-rays from Comet Hyakutake
(5 rows)

Time: 2.118 ms
```

# FTS in Databases

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

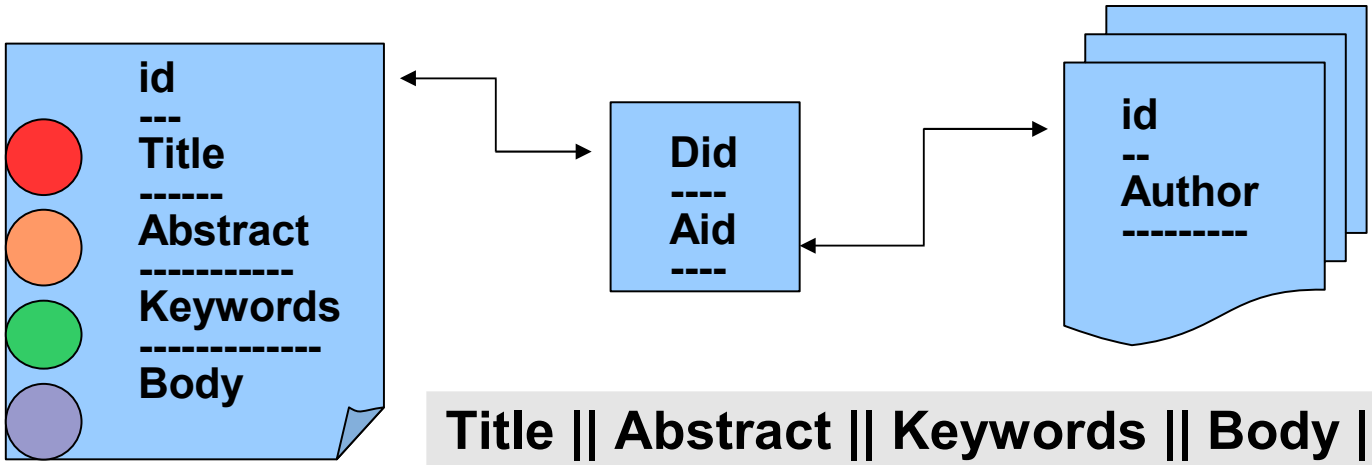
- Полнотекстовый поиск
  - найти документы удовлетворяющие запросу
  - отсортировать их в некотором порядке (opt.):
- Найти документы содержащие все слова из запроса и вернуть их отсортированными по похожести
- **Требования к FTS**
  - **полная интеграция с СУБД**
    - транзакционность
    - конкурентный доступ
    - восстановление после сбоев
    - online индекс
  - Конфигурируемость (парсеры, словари,...)
  - Масштабируемость

**Наиболее привычный вид поиска**

# Что такое *Документ* ?

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Произвольный текстовый атрибут
- Комбинация текстовых атрибутов из одной или разных таблиц
- Обязателен уникальный идентификатор



# Text Search Operators

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Традиционные операции текстового поиска ( TEXT op TEXT )
  - ~, ~*, LIKE, ILIKE

```
postgres=# select title  from apod where title ~* 'x-ray' limit 5;
                  title
-------------------------------------
 The X-Ray Moon
 Vela Supernova Remnant in X-ray
 Tycho's Supernova Remnant in X-ray
 ASCA X-Ray Observatory
 Unexpected X-rays from Comet Hyakutake
(5 rows)
```

```
postgres=# select title  from apod where title ilike '%x-ray%' limit 5;
```

# What's wrong ?

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Нет поддержки лингвистики
  - что есть слово ?
  - что индексировать ?
  - «нормализация» слов
  - стоп-слова  (noise-words)
- Нет релевантности
  - все документы одинаково «похожи»
- Медленно (нет индексной поддержки)
  - документы каждый раз сканируются

# FTS in PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **tsvector** – хранилище для документов, оптимизированное для поиска (*полнотекстовый индекс*)
  - отсортированный массив лексем
  - позиционная информация
  - структурная информация  (важность)
- **tsquery** – текстовый тип для запроса
- **FTS оператор**

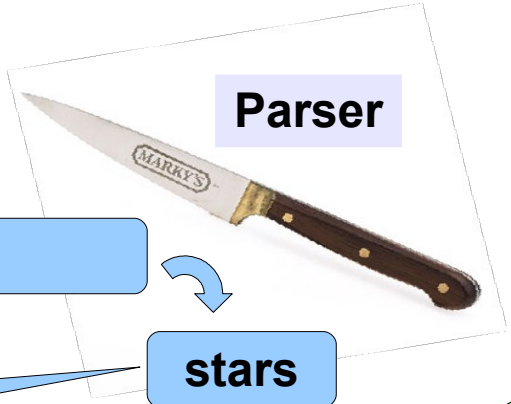
  tsquery @@ tsvector

# FTS in PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Где выигрыш ?**
  - документ обрабатывается при **индексировании** – не тратится время на обработку при поиске
  - документ разбивается на токены с помощью подключаемого парсера
  - токены превращаются в лексемы с помощью  подключаемых словарей
  - запоминаются позиционная информация и важность лексемы (ранжирование)
  - стоп-слова игнорируются

# Full-text index



# FTS in PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Query
  - обрабатывается при **поиске**
  - тоже разбивается на токены
  - токены превращаются в лексемы
  - убираются стоп-слова
  - можно ограничивать область поиска
  - может изменяться с помощью **query rewriting** «на ходу»

# FTS in PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Парсер разбивает текст на токены

Парсер

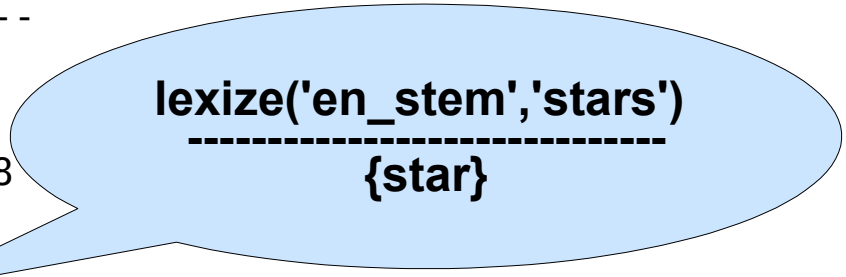
```
=# select * from token_type('default');
 tokid |    alias     |              description
-------+--------------+----------------------------------
     1 | lword        | Latin word
     2 | nlword       | Non-latin word
     3 | word         | Word
     4 | email        | Email
     5 | url          | URL
     6 | host         | Host
     7 | sfloat       | Scientific notation
     8 | version      | VERSION
     9 | part_hword   | Part of hyphenated word
    10 | nlpart_hword | Non-latin part of hyphenated word
    11 | lpart_hword  | Latin part of hyphenated word
    12 | blank        | Space symbols
    13 | tag          | HTML Tag
    14 | protocol     | Protocol head
    15 | hword        | Hyphenated word
    16 | lhword       | Latin hyphenated word
    17 | nlhword      | Non-latin hyphenated word
    18 | uri          | URI
    19 | file         | File or path name
    20 | float        | Decimal notation
    21 | int          | Signed integer
    22 | uint         | Unsigned integer
    23 | entity       | HTML Entity
(23 rows)
```

# FTS in PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Каждый токен  обрабатывается словарями

```
=# \dF+  russian_utf8
Configuration "pg_catalog.russian_utf8"
Parser name: "pg_catalog.default"
Locale: 'ru_RU.UTF-8' (default)
    Token     |       Dictionaries
--------------+--------------------------
 email        | pg_catalog.simple
 file         | pg_catalog.simple
 float        | pg_catalog.simple
 host         | pg_catalog.simple
 hword        | pg_catalog.ru_stem_utf8
 int          | pg_catalog.simple
 lhword       | pg_catalog.en_stem
 lpart_hword  | pg_catalog.en_stem
 lword        | pg_catalog.en_stem
 nlhword      | pg_catalog.ru_stem_utf8
 nlpart_hword | pg_catalog.ru_stem_utf8
 nlword       | pg_catalog.ru_stem_utf8
 part_hword   | pg_catalog.simple
 sfloat       | pg_catalog.simple
 uint         | pg_catalog.simple
 uri          | pg_catalog.simple
 url          | pg_catalog.simple
 version      | pg_catalog.simple
 word         | pg_catalog.ru_stem_utf8
```



# FTS in PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Слово передается от словаря к словарю пока оно не распознается.
- Если слово не распознано **всеми** словарями, то оно не индексируется.

```
=# \dF+ pg
                    Configuration "public.pg"
Parser name: "pg_catalog.default"
Locale: 'ru_RU.UTF-8' (default)
    Token     |                        Dictionaries
--------------+-------------------------------------------------------
 file         | pg_catalog.simple
 host         | pg_catalog.simple
 hword        | pg_catalog.simple
 int          | pg_catalog.simple
 lhword       | public.pg_dict,public.en_ispell,pg_catalog.en_stem
 lpart_hword  | public.pg_dict,public.en_ispell,pg_catalog.en_stem
 lword        | public.pg_dict,public.en_ispell,pg_catalog.en_stem
 nlhword      | pg_catalog.simple
 nlpart_hword | pg_catalog.simple
```

**lowercase**

**Стеммеры распознают все !**

**Правило: от «узкого» словаря к «широкому» !**

# FTS in PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Словарь** – это **программа**, которая принимает на вход токен и выдает массив лексем или NULL, если распознанно стоп-слово
- API позволяет писать словари под разные задачи
  - Укорачивать длинные цифры
  - Приводить все обозначения цветов в один вид
  - Приводить URL-и к каноническому виду
- Встроенные словари-заготовки (templates)  для
  - словарей ispell, myspell, huspell
  - snowball stemmer
  - thesaurus
  - synonym
  - simple

# FTS in PostgreSQL

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Набор функций для получения tsvector и tsquery
  - to_tsvector(fts_configuration, text)
  - to_tsquery(fts_configuration, text)

```
=# select to_tsvector('english', 'as supernovae stars');
      to_tsvector
-----------------------
 'star':3 'supernova':2
```

стоп-слово

position

```
=# select * from ts_debug('english', 'a supernovae stars');
 Alias |  Description  |   Token    |      Dicts list       |          Lexized token
-------+---------------+------------+-----------------------+---------------------------------
 lword | Latin word    | as         | {pg_catalog.en_stem} | pg_catalog.en_stem: {}
 blank | Space symbols |            |                       |
 lword | Latin word    | supernovae | {pg_catalog.en_stem} | pg_catalog.en_stem: {supernova}
 blank | Space symbols |            |                       |
 lword | Latin word    | stars      | {pg_catalog.en_stem} | pg_catalog.en_stem: {star}
(5 rows)
```

# FTS configuration

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- FTS конфигурация определяет
  - какой парсер используется для разбивания текста на токены
  - какие токены, какими словарями и в каком порядке обрабатываются
- Конфигурация задается с помощью SQL команд

```
{CREATE | ALTER | DROP} FULLTEXT {CONFIGURATION | DICTIONARY | PARSER}
```

- FTS конфигураций может быть много, поддерживаются схемы
- Информация о конфигурации доступна в psql

**\dF{,d,p}[+] [PATTERN]**

# FTS configuration

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
=# \dF
                                    List of fulltext configurations
   Schema   |        Name          |     Locale      | Default |        Description
ion
------------+----------------------+-----------------+---------+-----------------------
--------------------
 pg_catalog | danish_iso_8859_1    | da_DK.ISO8859-1 | Y       |
 pg_catalog | danish_utf_8         | da_DK.UTF-8     | Y       |
 pg_catalog | dutch_iso_8859_1     | nl_NL.ISO8859-1 | Y       |
 pg_catalog | dutch_utf_8          | nl_NL.UTF-8     | Y       |
 pg_catalog | english              | C               | Y       | English
 pg_catalog | finnish_iso_8859_1   | fi_FI.ISO8859-1 | Y       |
 pg_catalog | finnish_utf_8        | fi_FI.UTF-8     | Y       |
 pg_catalog | french_iso_8859_1    | fr_FR.ISO8859-1 | Y       |
 pg_catalog | french_utf_8         | fr_FR.UTF-8     | Y       |
 pg_catalog | german_iso_8859_1    | de_DE.ISO8859-1 | Y       |
 pg_catalog | german_utf_8         | de_DE.UTF-8     | Y       |
 pg_catalog | hungarian_iso_8859_1 |                 | Y       |
 pg_catalog | hungarian_utf_8      |                 | Y       |
 pg_catalog | italian_iso_8859_1   | it_IT.ISO8859-1 | Y       |
 pg_catalog | italian_utf_8        | it_IT.UTF-8     | Y       |
 pg_catalog | norwegian_iso_8859_1 | no_NO.ISO8859-1 | Y       |
 pg_catalog | norwegian_utf_8      | no_NO.UTF-8     | Y       |
 pg_catalog | portuguese_iso_8859_1| pt_PT.ISO8859-1 | Y       |
 pg_catalog | portuguese_utf_8     | pt_PT.UTF-8     | Y       |
 pg_catalog | russian_koi8         | ru_RU.KOI8-R    | Y       | Russian/KOI8-R
 pg_catalog | russian_utf8         | ru_RU.UTF-8     | Y       | Russian/UTF-8
 pg_catalog | russian_win1251      | ru_RU.CP1251    | Y       | Russian/WIN-1251
 pg_catalog | simple               |                 |         | Simple configuration
 pg_catalog | spanish_iso_8859_1   | es_ES.ISO8859-1 | Y       |
 pg_catalog | spanish_utf_8        | es_ES.UTF-8     | Y       |
 pg_catalog | swedish_iso_8859_1   | sv_SE.ISO8859-1 | Y       |
 pg_catalog | swedish_utf_8        | sv_SE.UTF-8     | Y       |
(27 rows)
```

**27 конфигураций для 10 языков**

# Pgweb example

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

**transaction !**

*$PGROOT/share/dicts_data*

```
BEGIN;
DROP FULLTEXT CONFIGURATION IF EXISTS public.pg;
DROP FULLTEXT DICTIONARY IF EXISTS pg_dict;
DROP FULLTEXT DICTIONARY IF EXISTS en_ispell;
CREATE FULLTEXT CONFIGURATION public.pg  LOCALE 'ru_RU.UTF-8' LIKE english WITH MAP;
ALTER  FULLTEXT CONFIGURATION public.pg SET AS DEFAULT;
CREATE FULLTEXT DICTIONARY pg_dict OPTION 'pg_dict.txt'  LIKE synonym;
CREATE FULLTEXT DICTIONARY en_ispell
OPTION 'DictFile="/usr/local/share/dicts/ispell/english-utf8.dict",
        AffFile="/usr/local/share/dicts/ispell/english-utf8.aff",
        StopFile="/usr/local/share/dicts/ispell/english-utf8.stop"'
LIKE ispell_template;
ALTER FULLTEXT DICTIONARY en_stem SET OPTION '/usr/local/share/dicts/ispell/english-utf8.stop';
ALTER FULLTEXT MAPPING ON pg FOR lword,lhword,lpart_hword
                      WITH pg_dict,en_ispell,en_stem;
DROP FULLTEXT MAPPING ON pg FOR email, url, sfloat, uri, float;
END;
```

**postgres postgresql**
**pgsql postgresql**
**postgre postgresql**

**don't index email, url,sfloat,uri,float**

# APOD example

**http://www.astronet.ru/db/apod.html**

- curl -O http://www.sai.msu.su/~megera/postgres/fts/apod.dump.gz
- zcat apod.dump.gz | psql postgres
- psql postgres

```
postgres=# \d apod
        Table "public.apod"
  Column  |   Type    | Modifiers
----------+-----------+-----------
 id       | integer   | not null
 title    | text      |
 body     | text      |
 sdate    | date      |
 keywords | text      |
```

```
postgres=# show tsearch_conf_name;
   tsearch_conf_name
-----------------------
 pg_catalog.russian_utf8
```

Default configuration for **ru_RU.UT-F8** locale

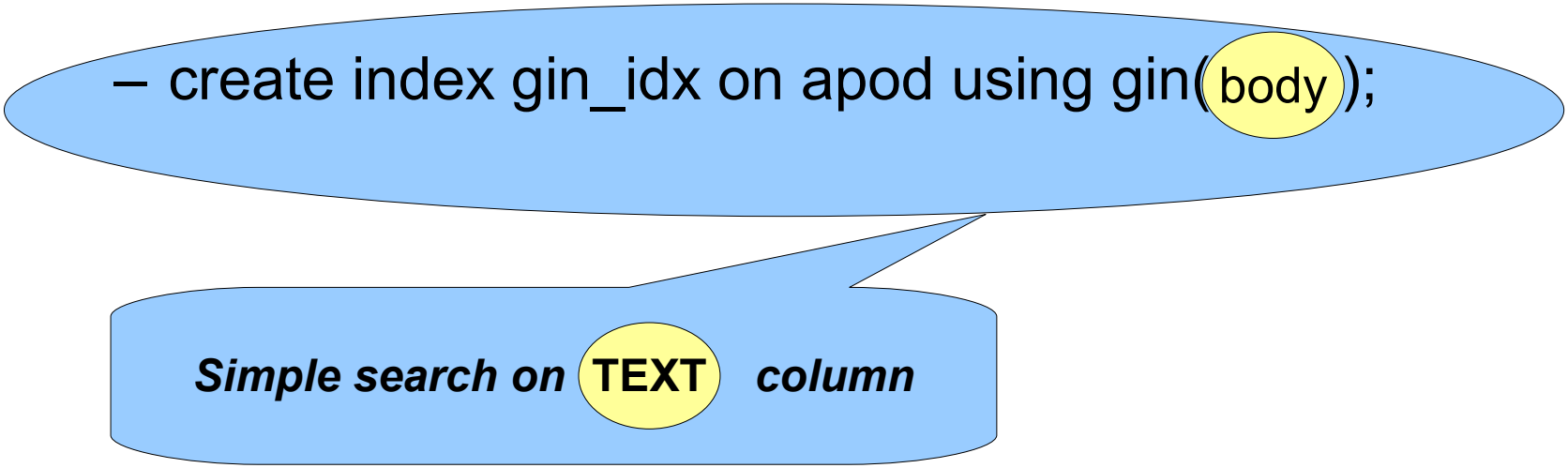
# APOD example: FTS configuration

```
postgres=# \dF+ pg_catalog.russian_utf8
Configuration "pg_catalog.russian_utf8"
Parser name: "pg_catalog.default"
Locale: 'ru_RU.UTF-8' (default)
    Token     |       Dictionaries
--------------+--------------------------
 email        | pg_catalog.simple
 file         | pg_catalog.simple
 float        | pg_catalog.simple
 host         | pg_catalog.simple
 hword        | pg_catalog.ru_stem_utf8
 int          | pg_catalog.simple
 lhword       | pg_catalog.en_stem
 lpart_hword  | pg_catalog.en_stem
 lword        | pg_catalog.en_stem
 nlhword      | pg_catalog.ru_stem_utf8
 nlpart_hword | pg_catalog.ru_stem_utf8
 nlword       | pg_catalog.ru_stem_utf8
 part_hword   | pg_catalog.simple
 sfloat       | pg_catalog.simple
 uint         | pg_catalog.simple
 uri          | pg_catalog.simple
 url          | pg_catalog.simple
 version      | pg_catalog.simple
 word         | pg_catalog.ru_stem_utf8
```

# FTS in PostgreSQL: Indices

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Для **ускорения** полнотекстового поиска (операторы @@, @@@ ) можно использовать индексы. Индексы нужны только для ускорения !
- RD-Tree - GiST (Generalized Search Tree)
  - create index gist_idx on apod using gist(fts);
- GiN ( Generalized Inverted Index) – обратный индекс
  - create index gin_idx on apod using gin(fts);



# APOD example: getting FTS index

```
postgres=# alter table apod add column fts tsvector;
postgres=# update apod  set fts=
                                setweight( coalesce( to_tsvector(title),''),'B')          ||
                                setweight( coalesce( to_tsvector(keywords),''),'A') ||
                                setweight( coalesce( to_tsvector(body),''),'D');
```

**NULL || nonNULL => NULL**

**if NULL then ''**

**A > B > D**

Важность

```
postgres=# create index apod_fts_idx on apod using gin(fts);
postgres=# vacuum analyze apod;

postgres=# select title from apod where fts  @@ plainto_tsquery('supernovae stars') limit 5;
                title
-------------------------------------
 Runaway Star
 Exploring The Universe With IUE 1978-1996
 Tycho Brahe Measures the Sky
 Unusual Spiral Galaxy M66
 COMPTEL Explores The Radioactive Sky
```

# APOD example: Search

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
postgres=# select title,rank_cd(fts, q) from apod,
to_tsquery('supernovae & x-ray') q
where fts  @@ q order by rank_cd desc limit 5;
                          title                           | rank_cd
----------------------------------------------------------+----------
 Supernova Remnant E0102-72 from Radio to X-Ray           | 1.59087
 An X-ray Hot Supernova in M81                            | 1.47733
 X-ray Hot Supernova Remnant in the SMC                   | 1.34823
 Tycho's Supernova Remnant in X-ray                       | 1.14318
 Supernova Remnant and Neutron Star                       | 1.08116
(5 rows)

Time: 1.965 ms
```

**rank_cd не нормирован, так как используется только локальная информация !**

**0 < rank/(rank+1) < 1**

**rank_cd('{0.1, 0.2, 0.4, 1.0 }',fts, q)**
**D C B A**

# APOD example: headline

```
postgres=# select headline(body,q,'StartSel=<,StopSel=>,MaxWords=10,MinWords=5'),
rank_cd(fts, q) from apod, to_tsquery('supernovae & x-ray') q where fts  @@
q order by rank_cd desc limit 5;
                                 headline                                  | rank_cd
---------------------------------------------------------------------------+---------
 <supernova> remnant  E0102-72, however, is giving astronomers a clue      | 1.59087
 <supernova> explosion. The picture was taken in  <X>-<rays>               | 1.47733
  <X>-<ray> glow is produced by  multi-million degree                      | 1.34823
  <X>-<rays> emitted by this shockwave made by a telescope                 | 1.14318
  <X>-<ray> glow. Pictured is the <supernova>                              | 1.08116
(5 rows)

Time: 39.298 ms
```

**Медленно** ! Правильно использовать **subselect**. Об этом подробнее в советах.

# APOD example

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Используя один индекс можно иметь разные поиски
  - поиск только в заголовках – поиск среди лексем, маркированных «важностью» 'b'.

```
=# select title,rank_cd(fts, q) from apod,
to_tsquery('supernovae:b & x-ray') q
where fts @@ q order by rank_cd desc limit 5;
                    title                     | rank_cd
----------------------------------------------+---------
 Supernova Remnant E0102-72 from Radio to X-Ray | 1.59087
 An X-ray Hot Supernova in M81                 | 1.47733
 X-ray Hot Supernova Remnant in the SMC        | 1.34823
 Tycho's Supernova Remnant in X-ray            | 1.14318
 Supernova Remnant and Neutron Star            | 1.08116
(5 rows)
```

**to_tsquery('supernovae:ab')** - поиск среди заголовков и ключевых слов

# FTS tips

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Серверная **locale** определяет FTS конфигурацию, используемую по умолчанию
  - show lc_ctype;
  - show lc_collate;
- GUC переменная **tsearch_conf_name** содержит название активной FTS конфигурации
  - show tsearch_conf_name;
- **search_path** определяет порядок просмотра схем при поиске FTS конфигурации
  - **pg_catalog всегда стоит первым, если не указан явно !!!**

```
show search_path;
pg_catalog,"$user",public
```

  - **set search_path=public,pg_catalog;** - гарантирует просмотр пользовательских FTS конфигураций перед встроенными системными

# FTS tips

- headline() функция медленная – используйте **subselect**

723 times

```
select id,headline(body,q),rank(fts,q) as rank
from apod, to_tsquery('stars') q
where fts @@ q order by rank desc limit 10;

Time: 723.634 ms
```

10 times !

```
select id,headline(body,q),rank from (
  select id,body,q, rank(fts,q) as rank from apod,
  to_tsquery('stars') q
  where fts @@ q
  order by rank desc limit 10
) as foo;

Time: 21.846 ms
```

```
=#select count(*)from apod where fts @@ to_tsquery('stars');
 count
-------
   790
```

# FTS tips

- **Нечеткий** поиск – используйте модуль pg_trgm
  - используется статистика по триграммам для нахождения наиболее похожего слова

```
=# select show_trgm('supyrnova');
                        show_trgm
-----------------------------------------------------
 {"  s","  su",nov,ova,pyr,rno,sup,upy,"va ",yrn}
```

```
=# select * into apod_words from stat('select fts from apod') order by ndoc desc,
     nentry desc,word;

=# \d apod_words
  Table "public.apod_words"
 Column |  Type   | Modifiers
--------+---------+-----------
 word   | text    |
 ndoc   | integer |
 nentry | integer |

=# create index trgm_idx on apod_words using gist(word gist_trgm_ops);

=# select word, similarity(word, 'supyrnova') AS sml
from apod_words where word % 'sypyrnova' order by sml desc, word;
   word    |   sml
-----------+----------
 supernova | 0.538462
```

собираем статистику по словам

# FTS tips – Query rewriting

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Изменение запроса **online**
  - расширение запроса
    - синонимы ( new york => Gottham, Big Apple, ...)
  - Сужение запроса
    - Курск => подводная лодка Курск
- Похоже на словарь тезаурус (синонимов), но не требует переиндексации

# FTS tips – Query rewriting

rewrite (tsquery, tsquery, tsquery)

rewrite (ARRAY[tsquery,tsquery,tsquery]) from aliases

rewrite (tsquery,'select tsquery,tsquery from aliases')

```
create table aliases( t tsquery primary key, s tsquery);
insert into aliases values(to_tsquery('supernovae'),
to_tsquery('supernovae|sn'));

apod=# select rewrite(to_tsquery('supernovae'),
'select * from aliases');
      rewrite
--------------------
 'supernova' | 'sn'
```

# FTS tips – Query rewriting

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
apod=# select title, coalesce(rank_cd(fts,q,1),2) as rank
from apod, to_tsquery('supernovae') q
where fts @@ q  order by rank desc limit 10;
```

| title | rank |
|---|---|
| The Mysterious Rings of Supernova 1987A | 0.669633 |
| Tycho's Supernova Remnant in X-ray | 0.598556 |
| Tycho's Supernova Remnant in X-ray | 0.598556 |
| Vela Supernova Remnant in Optical | 0.591655 |
| Vela Supernova Remnant in Optical | 0.591655 |
| Galactic Supernova Remnant IC 443 | 0.590201 |
| Vela Supernova Remnant in X-ray | 0.589028 |
| Supernova Remnant: Cooking Elements In The LMC | 0.585033 |
| Cas A Supernova Remnant in X-Rays | 0.583787 |
| Supernova Remnant N132D in X-Rays | **0.579241** |

**Low limit**

# FTS tips – Query rewriting

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
apod=# select id, title, coalesce(rank_cd(fts,q,1),2) as rank
from apod, rewrite(to_tsquery('supernovae'), 'select * from aliases') q
where fts @@ q  order by rank desc limit 10;
```

| id | title | rank |
|---|---|---|
| 1162701 | The Mysterious Rings of Supernova 1987A | 0.90054 |
| **1162717** | **New Shocks For Supernova 1987A** | **0.738432** |
| 1163673 | Echos of Supernova 1987A | 0.658021 |
| 1163593 | Shocked by Supernova 1987a | 0.621575 |
| 1163395 | Moving Echoes Around SN 1987A | 0.614411 |
| 1161721 | Tycho's Supernova Remnant in X-ray | 0.598556 |
| 1163201 | Tycho's Supernova Remnant in X-ray | 0.598556 |
| 1163133 | A Supernova Star-Field | 0.595041 |
| 1163611 | Vela Supernova Remnant in Optical | 0.591655 |
| 1161686 | Vela Supernova Remnant in Optical | 0.591655 |

```
apod=# select  title, coalesce(rank_cd(fts,q,1),2) as rank
from apod, to_tsquery('supernovae') q
where fts @@ q and id=1162717;
```

| title | rank |
|---|---|
| New Shocks For Supernova 1987A | **0.533312** |

**Old rank**

**new document**

# FTS tips – Partition your data

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Table inheritance + CE (constraint exclusion)
  - set constraint_exclusion=on;

```
psql test -c "create table a ( i int primary key);"
psql test -c "create table a1( check (i >=0    and i<=2000) ) inherits(a);"
psql test -c "create table a2( check (i >=2001 and i<=4000) ) inherits(a);"
psql test -c "create table a3( check (i >=4001 and i<=6000) ) inherits(a);"

#create index a_idx on a(i);
psql test -c "create index a1_idx on a1(i);"
psql test -c "create index a2_idx on a2(i);"
psql test -c "create index a3_idx on a3(i);"

for ((i=0;i<2000;i++)) do echo $i; done| psql test -c "copy a1 from stdin;"
for ((i=2001;i<4000;i++)) do echo $i; done| psql test -c "copy a2 from stdin;"
for ((i=4001;i<6000;i++)) do echo $i; done| psql test -c "copy a3 from stdin;"
```

```
=# \d a1
      Table "public.a1"
 Column |  Type   | Modifiers
--------+---------+-----------
 i      | integer | not null
Indexes:
    "a1_idx" btree (i)
Check constraints:
    "a1_i_check" CHECK (i >= 0 AND i <= 2000)
Inherits: a
```

# FTS tips – Partition your data

```
test=# explain select * from a where i = 10;
                              QUERY PLAN
-----------------------------------------------------------------------------
 Result  (cost=0.00..33.07 rows=4 width=4)
   ->  Append  (cost=0.00..33.07 rows=4 width=4)
         ->  Index Scan using a_pkey on a  (cost=0.00..8.27 rows=1 width=4)
               Index Cond: (i = 10)
         ->  Index Scan using a1_idx on a1 a  (cost=0.00..8.27 rows=1 width=4)
               Index Cond: (i = 10)
         ->  Index Scan using a2_idx on a2 a  (cost=0.00..8.27 rows=1 width=4)
               Index Cond: (i = 10)
         ->  Index Scan using a3_idx on a3 a  (cost=0.00..8.27 rows=1 width=4)
               Index Cond: (i = 10)
(10 rows)
D
```

**Все таблицы просматриваются**

```
test=# set constraint_exclusion=on;
SET
test=# explain select * from a where i = 10;
                              QUERY PLAN
-----------------------------------------------------------------------------
 Result  (cost=0.00..16.54 rows=2 width=4)
   ->  Append  (cost=0.00..16.54 rows=2 width=4)
         ->  Index Scan using a_pkey on a  (cost=0.00..8.27 rows=1 width=4)
               Index Cond: (i = 10)
         ->  Index Scan using a1_idx on a1 a  (cost=0.00..8.27 rows=1 width=4)
               Index Cond: (i = 10)
(6 rows)
```

**Только 1 таблица просматривается**

# FTS tips – Indices

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

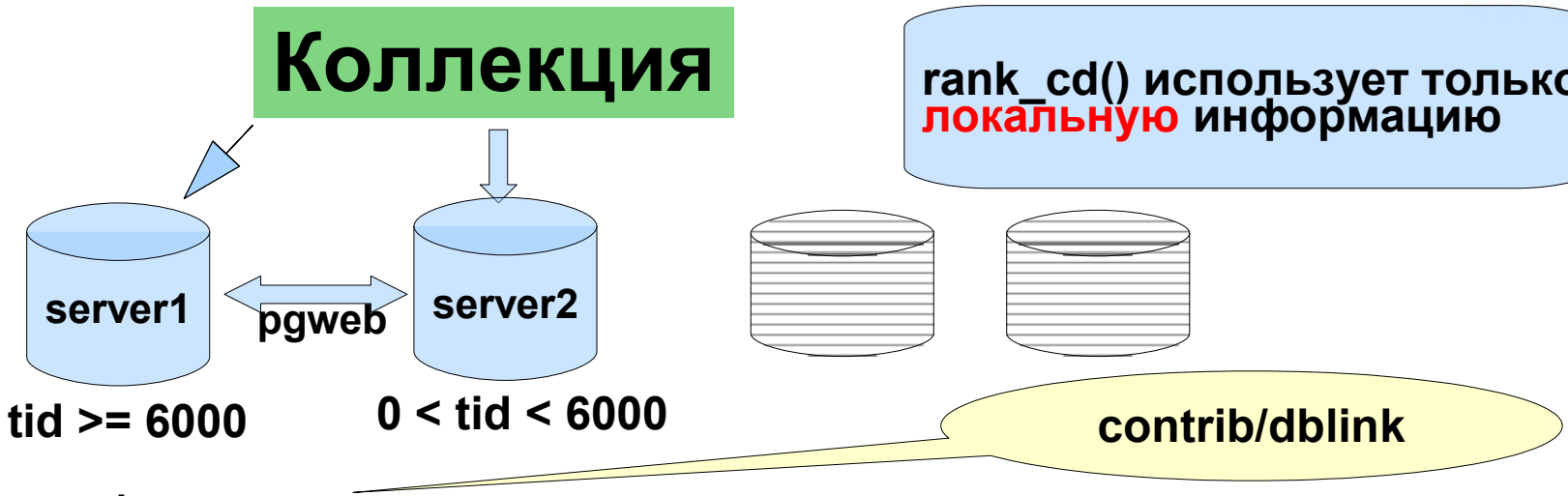
- Используйте индексы
  - GiST индекс для изменяющихся данных – текущие данные
    - быстро обновляется
    - не очень хорошо шкалируется
    - зависит от количества уникальных слова
  - GiN индекс для неменяющихся (архивных) таблиц
    - дольше обновляется ( при вставке документа из 1000 слов требуется сделать 1000 updates)
    - хорошо шкалируется
    - очень слабо зависит от числа уникальных слов
- Оба индекса конкурентны и поддерживают восстановление после сбоев

# FTS tips

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- GIN_FUZZY_SEARCH_LIMIT контролирует количество найденных документов, которое возвращает обратный индекс (GIN). Сам поиск быстрый, но чтение и передача документов ограничена диском.
  - По умолчанию GIN_FUZZY_SEARCH_LIMIT=0, т.е. выключен
  - Такие запросы, как правило, состоят из очень частотных слов.
  - В этом случае будет возвращен случайный набор найденных документов, но только если количество превысит GIN_FUZZY_SEARCH_LIMIT
  - Полезно установить его в разумных пределах (5000-20000), чтобы защититься от безумно большого количества найденных документов

# FTS tips - Distribute your data




```
select dblink_connect('pgweb','dbname=pgweb hostaddr='XXX.XXX.XXX.XXX');

select * from dblink('pgweb',
'select  tid, title, rank_cd(fts_index, q) as rank from pgweb,
to_tsquery(''table'') q
where q @@ fts_index and tid >= 6000 order by rank desc limit 10' )
as t1 (tid integer, title text, rank real)

union all

select tid, title,  rank_cd(fts_index, q) as rank from pgweb,
to_tsquery('table') q
where q @@ fts_index and tid < 6000 and tid > 0 order by rank desc limit 10

) as foo
order by rank desc limit 10;
```

# FTS features

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Полная интеграция с СУБД
- 27 встроенных конфигураций  для 10 европейских языков
- Поддержка пользовательских конфигураций
- Подключаемые словари ( ispell, snowball, thesaurus ), парсеры
- Поддержка multibyte (UTF-8)
- Поддержка ранжирования
- Два вида индексов – GiST, GIN
- Конкурентность и восстановление после сбоев
- Богатый язык запросов с возможностью изменения налету (query rewriting)

# Ссылки

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Документация**
  - http://www.sai.msu.su/~megera/postgres/fts/doc - FTS in PostgreSQL
  - http://www.sai.msu.su/~megera/wiki/tsearch2 - tsearch2 Wiki
  - http://www.sai.msu.su/~megera/postgres/gist/tsearch/V2 - Домашняя страница tsearch2
  - http://www.sai.msu.su/~megera/postgres/talks/ - статьи о PostgreSQL. Последняя версия этой презентации
  - http://www.sai.msu.su/~megera/postgres/talks/fts_pgsql_intro.html Ведение в полнотекстовый поиск в PostgreSQL (русский)
- **Данные**
  - http://www.sai.msu.su/~megera/postgres/fts/apod.dump.gz - дамп APOD (http://www.astronet.ru/db/apod.html)
- **Где работает**
  - http://search.postgresql.org - поиск по кластеру postgresql.org
- **Благодарности**
  - Российский Фонд Фундаментальных Исследований (РФФИ)
  - EnterprizeDB PostgreSQL Development Fund, Mannheim University, jfg:networks, Georgia Public Library Service, Рамблер

# Об авторах


- Major developers of PostgreSQL
- Новые типы данных и индексы: GiST, GIN
- Разработчики tsearch2, ltree, pg_trgm, hstore,intarray,..
- Контакты для предложений : oleg@sai.msu.su

# FTS Limitations

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Lexeme length < 2K
- Tsvector (lexemes+pos.) length < 1Mb
- #lexemes < 4^32
- 0 < #positions < 16383
- #positions per lexeme < 256
- #nodes (lexemes+ops.) in tsquery < 32768

**PostgreSQL mailing list archive:**

total **57,491,343** lexemes in **461020** msgs, **910989** unique lexemes