# Возможности текстового поиска в PostgreSQL

Смирнов Денис

# Текстовый поиск в pg

- Точный по подстроке
  - xxx_pattern_ops
  - триграммы, биграммы
- Нечеткий поиск
  - триграммы (%, <->)
  - полнотекстовый (@@)
  - фразовый (<->)
  - расстояние Левенштейна (seq scan only)
  - фонетические алгоритмы (Metaphone, Soundex)

# xxx_pattern_ops

- b-tree index
- классы операторов {text, varchar, bpchar}
- подстрока из начала/конца строки

```
postgres@10.0.66.3:fuzzy> explain analyze select patid from patinfo where fio(lname, fname, sname) ~~ 'смирно%дени%';
+-----------------------------------------------------------------------------------------------------------------------------------------------------------+
| QUERY PLAN                                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bitmap Heap Scan on patinfo  (cost=76.03..3219.10 rows=31 width=4) (actual time=1.016..8.603 rows=13 loops=1)                                             |
|   Filter: (fio(lname, fname, sname) ~~ 'смирно%дени%'::text)                                                                                               |
|   Rows Removed by Filter: 777                                                                                                                             |
|   Heap Blocks: exact=684                                                                                                                                  |
|     Bitmap Index Scan on patinfo_btree_idx  (cost=0.00..76.02 rows=1560 width=0) (actual time=0.206..0.206 rows=790 loops=1)                              |
|       Index Cond: ((fio(lname, fname, sname) ~>=~ 'смирно'::text) AND (fio(lname, fname, sname) ~<~ 'смирнп'::text))                                      |
| Planning time: 0.095 ms                                                                                                                                   |
| Execution time: 8.664 ms                                                                                                                                  |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------+
EXPLA
Time:    s
```

Filter after “%” :(

# Как работают триграммы?

- pg_trgm
- перед словом два пробела
- на конце один пробел
- режем по три символа

```
postgres@10.0.66.3:fuzzy> select show_trgm('cow');
+----------------------------------+
| show_trgm                        |
|----------------------------------|
| [u'  c', u' co', u'cow', u'ow '] |
+----------------------------------+
SELECT 1
Time: 0.010s
```

“__c”,”_co”,”cow”,”ow_“

# А биграммы?

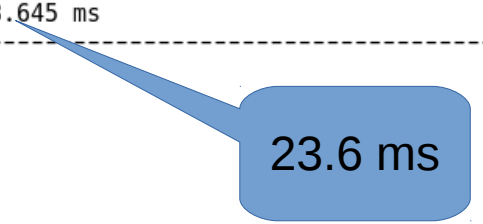
- pg_bigm
- созданы для иероглифической письменности
- быстрый поиск 1-2 символа
- не умеют нечеткий поиск
- gin only

# Устройство gist_trgm_ops

- GIST – R-tree framework
- RD-tree = russian doll (abstract sets, not boxes)
- Узел = предикат + ссылка (tid/дочерний узел)
- Поиск по функции согласованности

# gist_trgm_ops

```
postgres@10.0.66.3:fuzzy> create index patinfo_gist_trgm_idx on patinfo using gist (fio(lname, fname, sname) gist_trgm_ops);
CREATE INDEX
Time: 12.552s (12 seconds)
postgres@10.0.66.3:fuzzy> explain analyze select patid from patinfo where fio(lname, fname, sname) ~~ '%смирно%дени%';
+-----------------------------------------------------------------------------------------------------------------------------------------------------+
| QUERY PLAN                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Index Scan using patinfo_gist_trgm_idx on patinfo  (cost=0.28..128.82 rows=31 width=4) (actual time=1.181..23.589 rows=13 loops=1)                 |
|   Index Cond: (fio(lname, fname, sname) ~~ '%смирно%дени%'::text)                                                                                   |
| Planning time: 0.258 ms                                                                                                                             |
| Execution time: 23.645 ms                                                                                                                           |
+-----------------------------------------------------------------------------------------------------------------------------------------------------+
EXPLAIN
Time: 0.027s
```



# Нечеткий поиск gist_trgm_ops

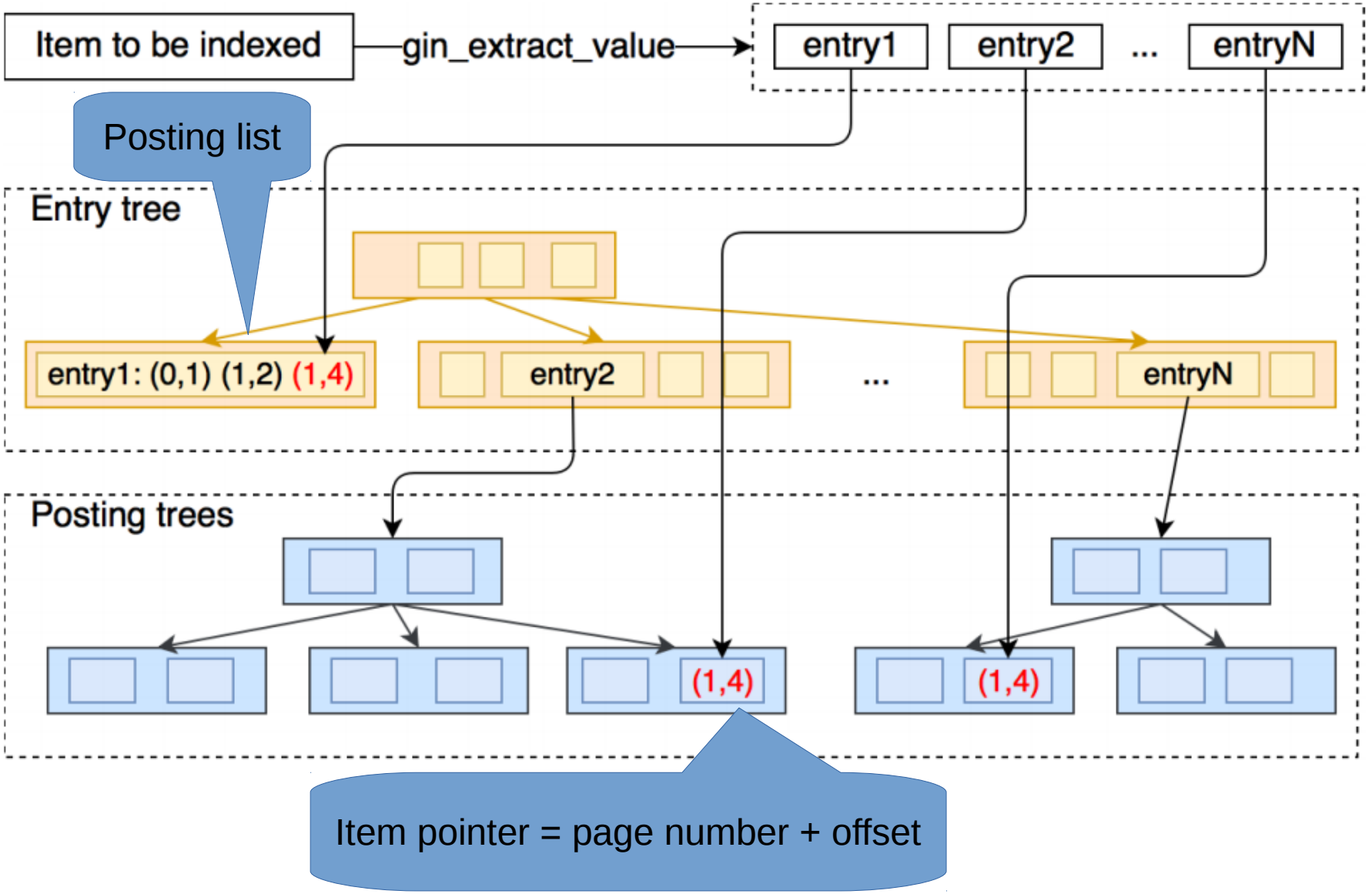
```
postgres@10.0.66.3:fuzzy> explain analyze
                          select patid, fio(lname, fname, sname),  fio(lname, fname, sname) <-> 'смирноф%динис%онатольевич%' as dist
                          from patinfo
                          order by dist limit 10;

+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| QUERY PLAN                                                                                                                                                                                    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Limit  (cost=0.28..6.65 rows=10 width=40) (actual time=190.626..191.925 rows=10 loops=1)                                                                                                      |
|   ->  Index Scan using patinfo_gist_trgm_idx on patinfo  (cost=0.28..198587.96 rows=312084 width=40) (actual time=190.622..191.893 rows=10 loops=1)                                           |
|         Order By: (fio(lname, fname, sname) <-> 'смирноф%динис%онатольевич%'::text)                                                                                                           |
| Planning time: 0.096 ms                                                                                                                                                                       |
| Execution time: 191.981 ms                                                                                                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
EXPLAIN
Time: 0.198s
postgres@10.0.66.3:fuzzy> select patid, fio(lname, fname, sname),  fio(lname, fname, sname) <-> 'смирноф%динис%онатольевич%' as dist
                          from patinfo
                          order by dist limit 10;

+---------+-------------------------------+----------+
| patid   | fio                           | dist     |
|---------+-------------------------------+----------|
| 7430    | смирнов денис анатольевич     | 0.470588 |
| 166805  | смирнов денис анатольевич     | 0.470588 |
| 166793  | смирнов денис анатольевич     | 0.470588 |
| 152011  | смирнов артём анатольевич     | 0.583333 |
| 97009   | смирнов павел анатольевич     | 0.594595 |
| 458558  | смирнов сергей анатольевич    | 0.594595 |
| 81913   | смирнов сергей анатольевич    | 0.594595 |
| 30296   | смирнов сергей анатольевич    | 0.594595 |
| 70081   | смирнов владимир анатольевич  | 0.615385 |
| 109956  | смирнов николай анатольевич   | 0.615385 |
+---------+-------------------------------+----------+
SELECT 10
Time: 0.199s
```

192 ms

# Устройство gin_trgm_ops



# gin_trgm_ops

```
postgres@10.0.66.3:fuzzy> create index patinfo_gin_trgm_idx on patinfo using gin (fio(lname, fname, sname) gin_trgm_ops);
CREATE INDEX
Time: 8.418s (8 seconds)
postgres@10.0.66.3:fuzzy> explain analyze select patid from patinfo where fio(lname, fname, sname) ~~ '%смирно%дени%';
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| QUERY PLAN                                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bitmap Heap Scan on patinfo  (cost=76.24..199.45 rows=31 width=4) (actual time=1.275..1.401 rows=13 loops=1)                                                                        |
|   Recheck Cond: (fio(lname, fname, sname) ~~ '%смирно%дени%'::text)                                                                                                                 |
|   Heap Blocks: exact=12                                                                                                                                                             |
|   ->  Bitmap Index Scan on patinfo_gin_trgm_idx  (cost=0.00..76.23 rows=31 width=0) (actual time=1.233..1.233 rows=13 loops=1)                                                      |
|         Index Cond: (fio(lname, fname, sname) ~~ '%смирно%дени%'::text)                                                                                                             |
| Planning time: 0.209 ms                                                                                                                                                             |
| Execution time: 1.458 ms                                                                                                                                                            |
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
EXPLAIN
Time: 0.   s
```

HEAP :(

1.4 ms

# Нечеткий поиск gin_trgm_ops

```
postgres@10.0.66.3:fuzzy> explain analyze
                          select patid, fio(lname, fname, sname), similarity(fio(lname, fname, sname), 'смирноф%динис%онатольевич%') as sml
                          from patinfo
                          where fio(lname, fname, sname) % 'смирноф%динис%онатольевич%'
                          order by sml desc limit 10;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| QUERY PLAN                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Limit  (cost=1122.50..1127.65 rows=10 width=40) (actual time=174.151..174.259 rows=10 loops=1)                                                                                                                  |
|   ->  Result  (cost=1122.50..1283.18 rows=312 width=40) (actual time=174.147..174.230 rows=10 loops=1)                                                                                                          |
|         ->  Sort  (cost=1122.50..1123.28 rows=312 width=58) (actual time=174.133..174.146 rows=10 loops=1)                                                                                                      |
|               Sort Key: (similarity(fio(lname, fname, sname), 'смирноф%динис%онатольевич%'::text)) DESC                                                                                                        |
|               Sort Method: top-N heapsort  Memory: 27kB                                                                                                                                                         |
|               ->  Bitmap Heap Scan on patinfo  (cost=26.70..1115.76 rows=312 width=58) (actual time=172.086..173.966 rows=79 loops=1)                                                                              |
|                     Recheck Cond: (fio(lname, fname, sname) % 'смирноф%динис%онатольевич%'::text)                                                                                                               |
|                     Heap Blocks: exact=78                                                                                                                                                                       |
|                     ->  Bitmap Index Scan on patinfo_gist_trgm_idx  (cost=0.00..26.62 rows=312 width=0) (actual time=171.975..171.975 rows=79 loops=1)                                                          |
|                           Index Cond: (fio(lname, fname, sname) % 'смирноф%динис%онатольевич%'::text)                                                                                                           |
| Planning time: 0.144 ms                                                                                                                                                                                         |
| Execution time: 174.325 ms                                                                                                                                                                                      |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
EXPLAIN
Time: 0.181s
postgres@10.0.66.3:fuzzy> select patid, fio(lname, fname, sname), similarity(fio(lname, fname, sname), 'смирноф%динис%онатольевич%') as sml
                          from patinfo
                          where fio(lname, fname, sname) % 'смирноф%динис%онатольевич%'
                          order by sml desc limit 10;
+---------+-----------------------------+----------+
| patid   | fio                         | sml      |
|---------+-----------------------------+----------|
| 7430    | смирнов денис анатольевич   | 0.529412 |
| 166805  | смирнов денис анатольевич   | 0.529412 |
| 166793  | смирнов денис анатольевич   | 0.529412 |
| 152011  | смирнов артём анатольевич   | 0.416667 |
| 81913   | смирнов сергей анатольевич  | 0.405405 |
| 458558  | смирнов сергей анатольевич  | 0.405405 |
| 30296   | смирнов сергей анатольевич  | 0.405405 |
| 97009   | смирнов павел анатольевич   | 0.405405 |
| 70081   | смирнов владимир анатольевич| 0.384615 |
| 109956  | смирнов николай анатольевич | 0.384615 |
+---------+-----------------------------+----------+
SELECT 10
Time: 0.172s
```

174.3 ms

# Исправление опечаток

- таблица со всеми уникальными словами (ts_stat)
- ищем через триграммы нечетким поиском слова, проверяем длину

# Как работает FTS?

- документ (parser)→ фрагменты (конфигурация словарей)→ лексемы → нормализация:
  - tsvector = лексемы + позиции
  - tsquery = лексемы + логические операции
- tsvector @@ tsquery

# Примеры FTS

```
postgres@10.0.66.3:fuzzy> select to_tsvector('russian','Кошки и тигры любят коробки');
+--------------------------------------+
| to_tsvector                          |
|--------------------------------------|
| 'коробк':5 'кошк':1 'люб':4 'тигр':3 |
+--------------------------------------+
SELECT 1
Time: 0.002s
postgres@10.0.66.3:fuzzy> select plainto_tsquery('russian','тигр и кошка');
+-------------------+
| plainto_tsquery   |
|-------------------|
| 'тигр' & 'кошк'   |
+-------------------+
SELECT 1
Time: 0.002s
postgres@10.0.66.3:fuzzy> select to_tsvector('russian','Кошки и тигры любят коробки') @@ plainto_tsquery('russian','тигр и кошка');
+------------+
| ?column?   |
|------------|
| True       |
+------------+
SELECT 1
Time: 0.002s
postgres@10.0.66.3:fuzzy> select phraseto_tsquery('russian','тигр и кошка');
+---------------------+
| phraseto_tsquery    |
|---------------------|
| 'тигр' <2> 'кошк'   |
+---------------------+
SELECT 1
Time: 0.002s
postgres@10.0.66.3:fuzzy> select to_tsvector('russian','Кошки и тигры любят коробки') @@ phraseto_tsquery('russian','тигр и кошка');
+------------+
| ?column?   |
|------------|
| False      |
+------------+
SELECT 1
Time: 0.002s
```

& - порядок лексем не важен

<n> - фразовый поиск

# Конфигурации FTS

- Стоп-слова
- Ispell (замена 1 слова)
- Тезаурус
- Snowball (склонения)


```
fuzzy=# \dF+ russian
Text search configuration "pg_catalog.russian"
Parser: "pg_catalog.default"
      Token      | Dictionaries
-----------------+--------------
 asciihword      | english_stem
 asciiword       | english_stem
 email           | simple
 file            | simple
 float           | simple
 host            | simple
 hword           | russian_stem
 hword_asciipart | english_stem
 hword_numpart   | simple
 hword_part      | russian_stem
 int             | simple
 numhword        | simple
 numword         | simple
 sfloat          | simple
 uint            | simple
 url             | simple
 url_path        | simple
 version         | simple
 word            | russian_stem
```

# FTS индексы

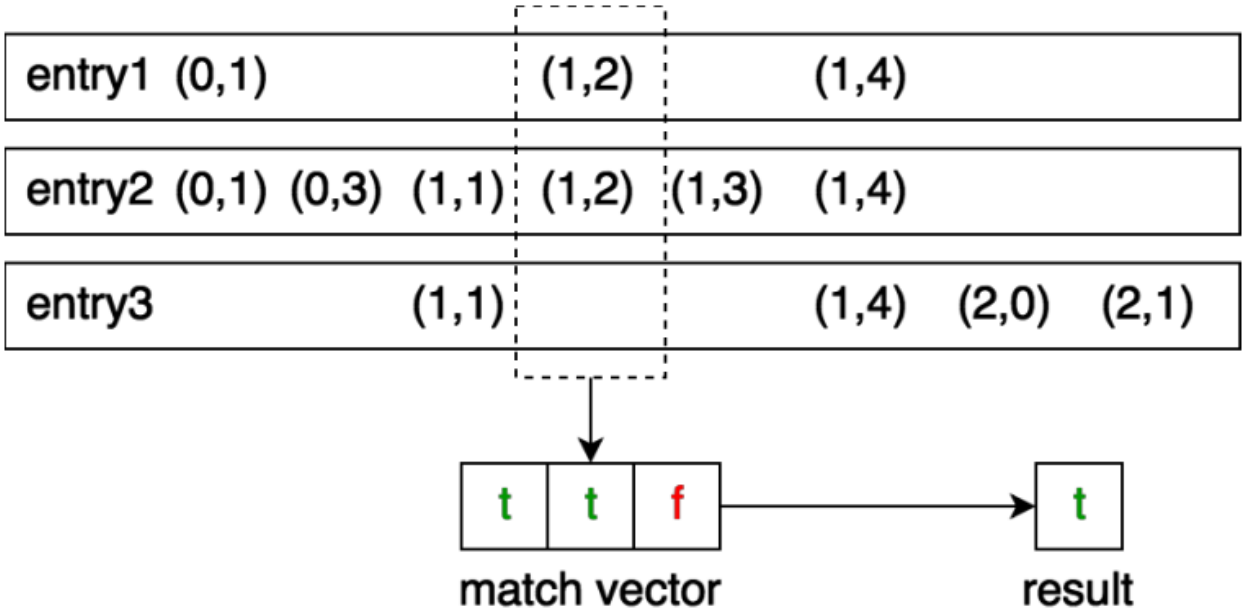
- GIST (медленно, heap)
- GIN (быстрее, heap)
- RUM (очень быстро, index only, но >=9.6)

# FTS + GIST

- отдельный столбец tsvector
- документы не положить в gist (т.к. toast)
- сигнатуры лексем → сигнатуры документов
- много ложноположительных результатов + недостаточно позиционной информации => index recheck в таблице

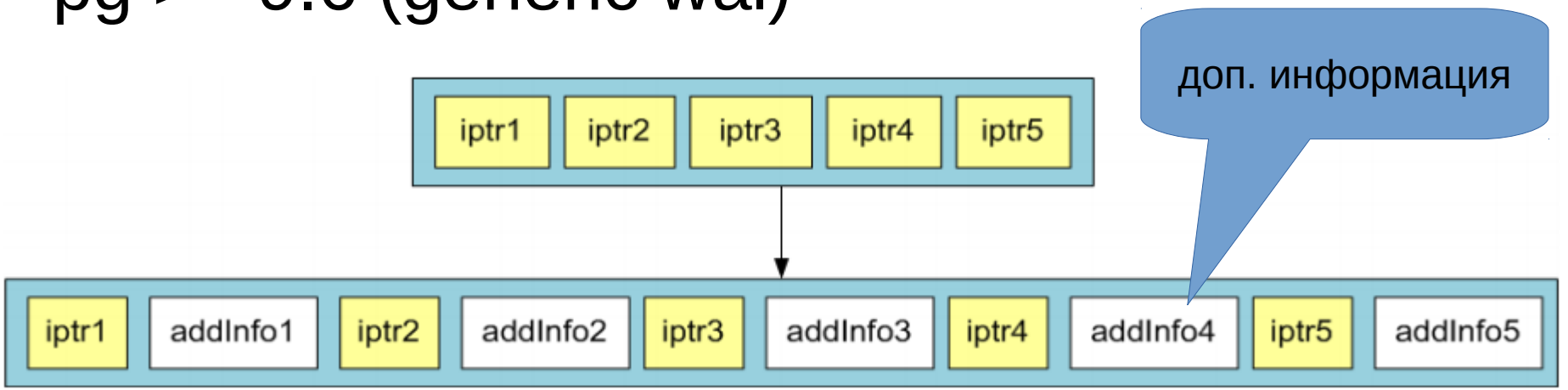
# FTS + GIN

- Отдельный столбец tsvector
- Нет позиционной информации, только tid
- => recheck в таблице



# FTS + RUM

- не нужен отдельный столбец tsvector
- в листья можно положить информацию, в т.ч. позиционную
- не трогаем таблицу!
- pg >= 9.6 (generic wal)



# А индекс у sphinx?

- Инвертированный индекс
- Ключи – лексемы
- Значения – список первичных ключей (разности между ними для уменьшения объема)
- => нет mvcc, все данные в индексе (recheck в heap не нужен)

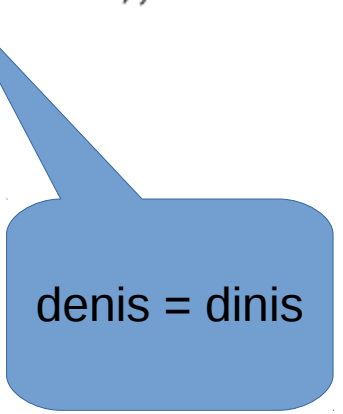
# Как ускорить FTS в PG?

- Больше классов операторов для rum!
- https://github.com/postgrespro/rum/issues/15

# Левенштейн

- считает близость слов по количеству замен/перестановок (1 – ok...4 – not ok)
- fuzzystrmatch
- нельзя проиндексировать => seq scan

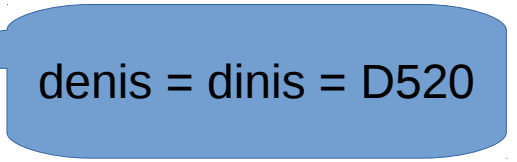
```
postgres@10.0.66.3:fuzzy> select levenshtein('denis','dinis');
+---------------+
| levenshtein   |
|---------------|
| 1             |
+---------------+
SELECT 1
Time: 0.002s
```



# Фонетические алгоритмы

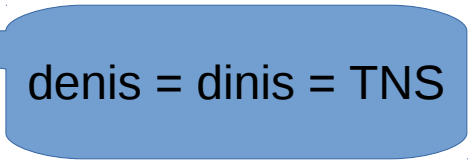
- Soundex (транслитерация)

```
postgres@10.0.66.3:fuzzy> select soundex('denis'), soundex('dinis'), difference('denis','dinis');
+-----------+-----------+--------------+
| soundex   | soundex   | difference   |
|-----------+-----------+--------------|
| D520      | D520      | 4            |
+-----------+-----------+--------------+
SELECT 1
Time: 0.003s
```




- (Double) Metaphone (есть русский алгоритм)

```
postgres@10.0.66.3:fuzzy> select dmetaphone('denis'), dmetaphone('dinis');
+--------------+--------------+
| dmetaphone   | dmetaphone   |
|--------------+--------------|
| TNS          | TNS          |
+--------------+--------------+
SELECT 1
Time: 0.016s
```




- plpgsql + plpython3u = любой алгоритм

# Благодарю за внимание!

https://habrahabr.ru/users/darthunix/
telegram: @darthunix