a gentle introduction to

PostgreSQL Full Text Search

Rémi Mercier ➜ remimercier.com

What we'll talk about:

1. How do we search without full text?

2. How do we search with full text?

3. Turning our data into lexemes (lexe-what?).

4. Fifty shades of parsing queries (well, more like four).

5. How to add full text search to your Rails app?

Rémi Mercier ➜ remimercier.com

Who am I?



👋 I'm Rémi Mercier.

Software developer, writer and former stained glass maker (yup).

Now an independent software engineer. Need help with you project? Let's talk!

I write about building software at remimercier.com.

You can also find me on ruby.social/@remi.

How to do search

without full text search?

Rémi Mercier ➜ remimercier.com

```sql
CREATE TABLE books (
  title VARCHAR(255),
  blurb TEXT
);

INSERT INTO books (title, blurb) VALUES
  ('The Fair Sorceress', 'A thrilling tale of magic and mystery where a witch meets elves.'),
  ('The Silent Grove', 'A journey through an ancient, forgotten forest, with sorcerers and dwarves.'),
  ('The Last Kingdom', 'A historical novel set in medieval times where sorceresses save the day.'),
  ('Deep Depth', 'A suspenseful story of adventure of sailors at sea.'),
  ('Sorcery School', 'A young-adult thriller where a sorceress and a sorcerer fight to keep the school from being eat up by zombies.');
  ('The Moonlit Pact', 'A secret covenant between a sorcerer and a werewolf threatens to unravel an ancient curse.'),
  ('Whispers of the Hollow', 'A haunted valley holds forgotten spells and a witch determined to reclaim them.'),
  ('Crown of Embers', 'In a fractured kingdom, a young mage must unite dragons and humans against a dark uprising.'),
  ('The Crimson Sigil', 'A magical symbol appears across the land, luring warlocks and thieves to their doom.'),
  ('Twilight Keep', 'A mysterious castle where time bends, ruled by a silent enchantress.'),
  ('Ashes of Arandel', 'A fire sorceress returns from exile to battle the shadows overtaking her homeland.'),
  ('The Elven Path', 'A thief stumbles into an elven realm, unlocking secrets that could ignite a war.'),
  ('Witchbane Woods', 'A cursed forest hides a portal guarded by undead druids and a vengeful spirit.'),
  ('Runes and Ruin', 'A scholar of ancient languages discovers a tome that whispers spells to awaken the dead.'),
  ('The Forgotten Spire', 'Adventurers climb a tower lost to history, facing magical trials left by its last guardian.'),
  ('Mistwalker’s Oath', 'A cloaked figure who walks between worlds is sworn to protect a child born under twin moons.'),
  ('The Iron Familiar', 'A rogue sorcerer crafts a mechanical beast to battle the magical elite.'),
  ('The Hollow Throne', 'A power-hungry witch queen seeks an ancient relic to cement her rule over shadow realms.'),
  ('Lanterns of Liora', 'Wizards gather each decade to light the lanterns that keep nightmares at bay.'),
  ('Stormbound', 'A skyship crew must harness forbidden storm magic to survive a cursed voyage.')
```

```
SELECT * FROM books
WHERE title ILIKE '%sorceresses%';

   Title | Blurb
---------+-------

(0 rows)
```

```
1 SELECT * FROM books
2 WHERE blurb ILIKE '%sorceresses%';
3
4          Title      |    Blurb
5 ------------------+-------------------------------------------------------------------------------
6  The Last Kingdom | A historical novel set in medieval times where sorceresses save the day.
```

```
SELECT * FROM books
WHERE blurb ILIKE '%sorceress%' OR blurb ILIKE '%sorceresses%';

        Title           |    Blurb
 ----------------------+-------------------------------------------------------------------------------------------
  The Last Kingdom     | A historical novel set in medieval times where sorceresses save the day.
  Sorcery School       | A young-adult thriller where a sorceress and a sorcerer fight to keep the school from being eat
up by zombies.
  Ashes of Arandel     | A fire sorceress returns from exile to battle the shadows overtaking her homeland.
```

```
1  SELECT * FROM books
2  WHERE blurb ILIKE '%sorceress%'
3     OR blurb ILIKE '%sorceresses%'
4     OR title ILIKE '%sorceress%';
5     OR blurb ILIKE '%sorceress%';
6
7          Title           |    Blurb
8   ---------------------+-----------------------------------------------------------------------------------
9    The Fair Sorceress   | A thrilling tale of magic and mystery where a witch meets elves.
10   The Last Kingdom     | A historical novel set in medieval times where sorceresses save the day.
11   Sorcery School       | A young-adult thriller where a sorceress and a sorcerer fight to keep the school from being eat
   up by zombies.
12   Ashes of Arandel     | A fire sorceress returns from exile to battle the shadows overtaking her homeland.
```

How to do search

with full text search?

Rémi Mercier ➜ remimercier.com

```
SELECT * FROM books
WHERE to_tsvector('english', coalesce(title, '') || ' ' || coalesce(blurb, ''))
      @@ plainto_tsquery('english', 'sorceress');

        Title           |    Blurb
------------------------+-------------------------------------------------------------------------------------------
 The Fair Sorceress     | A thrilling tale of magic and mystery where a witch meets elves.
 The Last Kingdom       | A historical novel set in medieval times where sorceresses save the day.
 Sorcery School         | A young-adult thriller where a sorceress and a sorcerer fight to keep the school from being eat
up by zombies.
 Ashes of Arandel       | A fire sorceress returns from exile to battle the shadows overtaking her homeland.
```

How does

full text search work?

Rémi Mercier ➜ remimercier.com

How does

full text search work?

1. Turning documents to tokens

Rémi Mercier ➜ remimercier.com

```
SELECT *
FROM ts_debug(
  'english',
  'A young-adult thriller where a sorceresses and' ||
  'a sorcerer fight to keep the school from being eat up by zombies.'
);


      alias      |           description            |    token
-----------------+----------------------------------+-------------
 asciiword       | Word, all ASCII                  | A
 blank           | Space symbols                    |
 asciihword      | Hyphenated word, all ASCII       | young-adult
 hword_asciipart | Hyphenated word part, all ASCII  | young
 blank           | Space symbols                    | -
 hword_asciipart | Hyphenated word part, all ASCII  | adult
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | thriller
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | where
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | a
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | sorceresses
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | and
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | a
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | sorcerer
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | fight
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | to
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | keep
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | the
 blank           | Space symbols                    |
 asciiword       | Word, all ASCII                  | school
```

How does

full text search work?

2. Matching tokens with dictionaries

Rémi Mercier ➜ remimercier.com

alias	description	token	dictionaries
asciiword	Word, all ASCII	A	{english_stem}
blank	Space symbols		{}
asciihword	Hyphenated word, all ASCII	young-adult	{english_stem}
hword_asciipart	Hyphenated word part, all ASCII	young	{english_stem}
blank	Space symbols	-	{}
hword_asciipart	Hyphenated word part, all ASCII	adult	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	thriller	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	where	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	a	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	sorceresses	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	and	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	a	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	sorcerer	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	fight	{english_stem}
blank	Space symbols		{}
asciiword	Word, all ASCII	to	{english_stem}

alias	description	token	dictionaries
asciiword	Word, all ASCII	A	{english_stem}
asciihword	Hyphenated word, all ASCII	young-adult	{english_stem}
hword_asciipart	Hyphenated word part, all ASCII	young	{english_stem}
hword_asciipart	Hyphenated word part, all ASCII	adult	{english_stem}
asciiword	Word, all ASCII	thriller	{english_stem}
asciiword	Word, all ASCII	where	{english_stem}
asciiword	Word, all ASCII	a	{english_stem}
asciiword	Word, all ASCII	sorceresses	{english_stem}
asciiword	Word, all ASCII	and	{english_stem}
asciiword	Word, all ASCII	a	{english_stem}
asciiword	Word, all ASCII	sorcerer	{english_stem}
asciiword	Word, all ASCII	fight	{english_stem}
asciiword	Word, all ASCII	to	{english_stem}
asciiword	Word, all ASCII	keep	{english_stem}
asciiword	Word, all ASCII	the	{english_stem}
asciiword	Word, all ASCII	school	{english_stem}

How does

full text search work?

3. Turning tokens into lexemes

Rémi Mercier ➜ remimercier.com

alias	description	token	dictionaries	lexemes
asciiword	Word, all ASCII	A	{english_stem}	{}
asciihword	Hyphenated word, all ASCII	young-adult	{english_stem}	{young-adult}
hword_asciipart	Hyphenated word part, all ASCII	young	{english_stem}	{young}
hword_asciipart	Hyphenated word part, all ASCII	adult	{english_stem}	{adult}
asciiword	Word, all ASCII	thriller	{english_stem}	{thriller}
asciiword	Word, all ASCII	where	{english_stem}	{}
asciiword	Word, all ASCII	a	{english_stem}	{}
asciiword	Word, all ASCII	sorceresses	{english_stem}	{sorceress}
asciiword	Word, all ASCII	and	{english_stem}	{}
asciiword	Word, all ASCII	a	{english_stem}	{}
asciiword	Word, all ASCII	sorcerer	{english_stem}	{sorcer}
asciiword	Word, all ASCII	fight	{english_stem}	{fight}
asciiword	Word, all ASCII	to	{english_stem}	{}
asciiword	Word, all ASCII	keep	{english_stem}	{keep}
asciiword	Word, all ASCII	the	{english_stem}	{}
asciiword	Word, all ASCII	school	{english_stem}	{school}

alias	description	token	dictionaries	lexemes
asciihword	Hyphenated word, all ASCII	young-adult	{english_stem}	{young-adult}
hword_asciipart	Hyphenated word part, all ASCII	young	{english_stem}	{young}
hword_asciipart	Hyphenated word part, all ASCII	adult	{english_stem}	{adult}
asciiword	Word, all ASCII	thriller	{english_stem}	{thriller}
asciiword	Word, all ASCII	sorceresses	{english_stem}	{sorceress}
asciiword	Word, all ASCII	sorcerer	{english_stem}	{sorcer}
asciiword	Word, all ASCII	fight	{english_stem}	{fight}
asciiword	Word, all ASCII	keep	{english_stem}	{keep}
asciiword	Word, all ASCII	school	{english_stem}	{school}

```
SELECT to_tsvector('magic MAGICS magical magically');
      to_tsvector
------------------------
 'magic':1,2,3,4
```

How does

full text search work?

4. Fifty shades of parsing queries

Rémi Mercier ➜ remimercier.com

How does

full text search work?

4. Fifty shades of parsing queries

to_tsquery()

PRO: granular control over the query
CON: needs preliminary normalisations

Rémi Mercier ➜ remimercier.com

```
1 SELECT to_tsquery('The fun elves and the magical sorceresses');
2 ERROR:  syntax error in tsquery: "The fun elves and the magical sorceresses"
```

```
SELECT to_tsquery('The & fun & elves & and & the & magical & sorceresses');
               to_tsquery
-----------------------------------------
 'fun' & 'elv' & 'magic' & 'sorceress'
```

```
SELECT to_tsquery('The & fun & elves & and & the & !magical & !sorceresses');
                to_tsquery
-------------------------------------------
  'fun' & 'elv' & !'magic' & !'sorceress'
```

How does

full text search work?

4. Fifty shades of parsing queries

plainto_tsquery()

PRO: delegates normalisation to PostgreSQL
CON: no logical operators beyond ‘&’

Rémi Mercier ➜ remimercier.com

```
SELECT plainto_tsquery('The fun elves and the magical sorceresses');
            plainto_tsquery
-----------------------------------
  'fun' & 'elv' & 'magic' & 'sorceress'
(1 row)
```

```
SELECT plainto_tsquery('The fun elves and the !magical !sorceresses');
            plainto_tsquery
-----------------------------------------
  'fun' & 'elv' & 'magic' & 'sorceress'
```

How does

full text search work?

4. Fifty shades of parsing queries

phraseto_tsquery()

PRO: handle meaningful group of words
CON: no logical operators

Rémi Mercier ➜ remimercier.com

```
1 SELECT phraseto_tsquery('the wheel of time');
2   phraseto_tsquery
3 ---------------------
4   'wheel' <2> 'time'
```

How does

full text search work?

4. Fifty shades of parsing queries

websearch_to_tsquery()

PRO: natural language and some logical operators
CON: ¯_(ツ)_/¯

Rémi Mercier ➜ remimercier.com

```
SELECT websearch_to_tsquery('The fun elves and -"the magical sorceresses"');
             websearch_to_tsquery
-------------------------------------------------
 'fun' & 'elv' & !( 'magic' <-> 'sorceress' )
```

Frequency or density?

Ranking results

Rémi Mercier ➜ remimercier.com

Frequency or density?

Ranking results

4. Fifty shades of parsing queries

ts_rank()

➜ Ranks vectors based on the frequency of their matching lexemes.

Rémi Mercier ➜ remimercier.com

```
SELECT title,
         ts_rank(
             to_tsvector(blurb),
             websearch_to_tsquery('fun elves and magical sorceresses')
         ) AS rank
  FROM recipes
  ORDER BY rank DESC
  LIMIT 10;

                    title                     |   rank
----------------------------------------------+----------
 The Fair Sorceress                           |      3.1
 The Last Kingdom                             |      2.4
 Sorcery School                               |  2.01317
```

Frequency or density?

Ranking results

4. Fifty shades of parsing queries

ts_rank_cd()

➜ Computes the *cover density* ranking for the given document vector and query.
➜ Adds a normalization option to adjust density based on the length of the document.

Rémi Mercier ➜ remimercier.com

```
SELECT title,
       ts_rank_cd(
           to_tsvector(blurb),
           websearch_to_tsquery('fun elves and magical sorceresses'),
           1
       ) AS rank
FROM recipes
ORDER BY rank DESC
LIMIT 10;
```

Frequency or density?

Ranking results

4. Fifty shades of parsing queries

setweight()

➜ Adds weighting options for each column.

Rémi Mercier ➜ remimercier.com

```sql
WITH search_data AS (
  SELECT
    title,
    blurb,
    setweight(to_tsvector('english', coalesce(title, '')), 'A') ||
    setweight(to_tsvector('english', coalesce(blurb, '')), 'B') AS document_vector,
    plainto_tsquery('english', 'sorceress') AS query
  FROM books
)
SELECT title, blurb
FROM search_data
WHERE document_vector @@ query
ORDER BY ts_rank(document_vector, query) DESC;
```

How to add

full text search to your Rails app?

Rémi Mercier ➜ remimercier.com

README | Code of conduct | MIT license

pg_search



DESCRIPTION

PgSearch builds named scopes that take advantage of PostgreSQL's full text search.

Read the blog post introducing PgSearch at https://tanzu.vmware.com/content/blog/pg-search-how-i-learned-to-stop-worrying-and-love-postgresql-full-text-search

REQUIREMENTS

- Ruby 3.1+
- Active Record 7.1+
- PostgreSQL 9.2+
- PostgreSQL extensions for certain features

Thank you!

Questions about this presentation?

In need of a product engineer for your project?

Let's talk!

Rémi Mercier ➜ remimercier.com