

# Introduction to full-text search

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

SQL

**Brian Piccolo**

Sr. Director, Digital Strategy

# Topics

- Full Text search
- Extending PostgreSQL
- Improving full text search with extensions

# The LIKE operator

**\_ wildcard:** Used to match exactly one character.

**% wildcard:** Used to match zero or more characters.

```
SELECT title
FROM film
WHERE title LIKE 'ELF%';
```

```
+-----+
| title          |
+-----+
| ELF PARTY     |
+-----+
```

# The LIKE operator

```
SELECT title
FROM film
WHERE title LIKE '%ELF';
```

```
+-----+
| title          |
+-----+
| ENCINO ELF     |
| GHOSTBUSTERS  ELF |
+-----+
```

# The LIKE operator

```
SELECT title  
FROM film  
WHERE title LIKE '%elf%';
```

```
+-----+  
| title |  
+-----+
```

# LIKE versus full-text search

```
SELECT title, description
FROM film
WHERE to_tsvector(title) @@ to_tsquery('eLf');
```

```
+-----+
| title          |
+-----+
| ELF PARTY      |
| ENCINO ELF     |
| GHOSTBUSTERS  ELF |
+-----+
```

# What is full-text search?

**Full text search provides a means for performing natural language queries of text data in your database.**

- Stemming
- Spelling mistakes
- Ranking

# Full-text search syntax explained

```
SELECT title, description  
FROM film  
WHERE to_tsvector(title) @@ to_tsquery('elf');
```



# Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

# Extending PostgreSQL

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL



**Brian Piccolo**

Sr. Director, Digital Strategy

# User-defined data types

## Enumerated data types

```
CREATE TYPE dayofweek AS ENUM (  
    'Monday',  
    'Tuesday',  
    'Wednesday',  
    'Thursday',  
    'Friday',  
    'Saturday',  
    'Sunday'  
);
```

# Getting information about user-defined data types

```
SELECT typename, typcategory
FROM pg_type
WHERE typename='dayofweek';
```

```
+-----+-----+
| typename | typcategory |
+-----+-----+
| dayofweek | E           |
+-----+-----+
```

# Getting information about user-defined data types

```
SELECT column_name, data_type, udt_name
FROM INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'film';
```

```
+-----+
| column_name | data_type          | udt_name      |
|-----|-----|-----|
| title       | character varying | varchar       |
| rating      | USER-DEFINED      | mpaa_rating   |
+-----+
```

# User-defined functions

```
CREATE FUNCTION squared(i integer) RETURNS integer AS $$  
    BEGIN  
        RETURN i * i;  
    END;  
$$ LANGUAGE plpgsql;
```

```
SELECT squared(10);
```

```
+-----+  
| squared |  
+-----+  
| 100     |  
+-----+
```

# User-defined functions in the Sakila database

- **get\_customer\_balance(customer\_id, effective\_data):** calculates the current outstanding balance for a given customer.
- **inventory\_held\_by\_customer(inventory\_id):** returns the customer\_id that is currently renting an inventory item or null if it's currently available.
- **inventory\_in\_stock(inventory\_id):** returns a boolean value of whether an inventory item is currently in stock.

# Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL



# Intro to PostgreSQL extensions

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL



**Brian Piccolo**

Sr. Director, Digital Strategy

# Intro to PostgreSQL extensions

## Commonly used extensions

- PostGIS
- PostPic
- fuzzystrmatch
- pg\_trgm

# Querying extension meta data

## Available Extensions

```
SELECT name
FROM pg_available_extensions;
```

```
+-----+
| name          |
+-----+
| dblink        |
| pg_stat_statements |
+-----+
```

## Installed Extensions

```
SELECT extname
FROM pg_extension;
```

```
+-----+
| name      |
+-----+
| plpgsql   |
+-----+
```

```
--Enable the fuzzystmatch extension
CREATE EXTENSION IF NOT EXISTS fuzzystmatch;
--Confirm that fuzzstrmatch has been enabled
SELECT extname FROM pg_extension;
```

```
+-----+
| name          |
|-----|
| plpgsql       |
| fuzzystmatch  |
+-----+
```

# Using fuzzystrmatch or fuzzy searching

```
SELECT levenshtein('GUMBO', 'GAMBOL');
```

```
+-----+  
| levenshtein |  
+-----+  
|           2 |  
+-----+
```

# Compare two strings with pg\_trgm

```
SELECT similarity('GUMBO', 'GAMBOL');
```

```
+-----+  
| similarity |  
+-----+  
| 0.18181818 |  
+-----+
```

# Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

# Putting it All Together

FUNCTIONS FOR MANIPULATING DATA IN POSTGRES SQL



**Brian Piccolo**

Sr. Director, Digital Strategy



# Functions for manipulating data recap and review

- Common data types in PostgreSQL
- Date/time functions and operators
- Parsing and manipulating text
- PostgreSQL Extensions and full-text search

# Thank you!

**FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL**