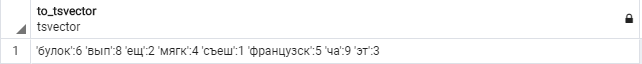
**Использование FTS в PostgreSQL**

**1. Тип tsvector**

> Задание 1

1. Изучите документацию к функции *to\_tsvector*

2. Вызовите эту функцию для следующей строки: *Съешь ещё этих мягких французских булок, да выпей чаю*

**

3. Почему в векторе нет слова *да*?

Ответ: Тип *tsvector* представляет собой что-то вроде нормализованной строки, по которой будет производиться поиск. Под нормализацией понимается выкидывание стоп-слов, таких, как предлоги, обрезание окончаний слов, и так далее. В данном примере слово да, распознается как стоп-слово, поэтому выкидывается из списка. В предыдущем примере, стоп-словом являлось the, из-за чего оно тоже отсутствовало в выводе.

**2. Тип tsquery**

Выполните по очереди запросы

*--№1*

*SELECT to\_tsvector('The quick brown fox jumped over the lazy dog')*

*@@ to\_tsquery('fox');*

*--№2*

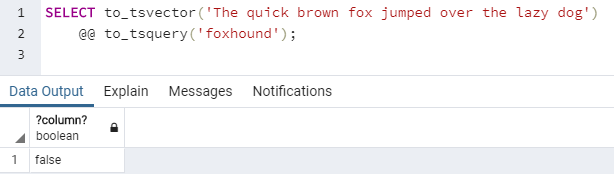
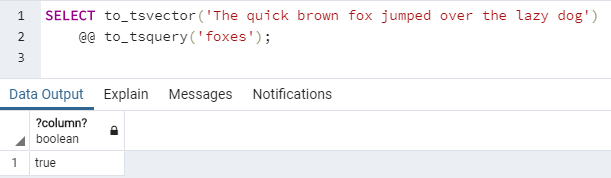
*SELECT to\_tsvector('The quick brown fox jumped over the lazy dog')*

*@@ to\_tsquery('foxes');*

*--№3*

*SELECT to\_tsvector('The quick brown fox jumped over the lazy dog')*

*@@ to\_tsquery('foxhound');*

**

> Задание 2

1. Что означают символы `*@@*`

Ответ: *@@* - оператор сопоставления, возвращает *true*, если *tsvector* соответствует tsquery

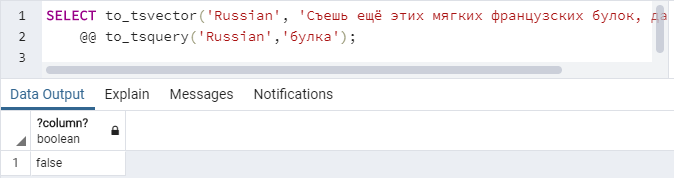
2. Почему второй запрос возвращает `*true*`, а третий не возвращает

Ответ: Т.к. лексема слова *foxhound*, является *foxhound*. А данной лексемы нет в предложении.

3. Выполните запрос

*SELECT to\_tsvector('Russian', 'Съешь ещё этих мягких французских булок, да выпей чаю.')*

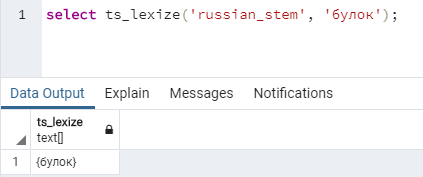
*@@ to\_tsquery('Russian','булка');*

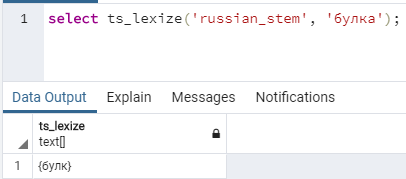


Почему слово булка не находится?

4. Используйте функцию `*select ts\_lexize('russian\_stem', 'булок');*` для того чтобы понять почему.

Ответ: У данных слов разные лексемы, поэтому лексему булка не нашло.





5. Замените в предложении слово `*булок*`, на слово `*пирожков*`

Выполните запросы

*--№1*

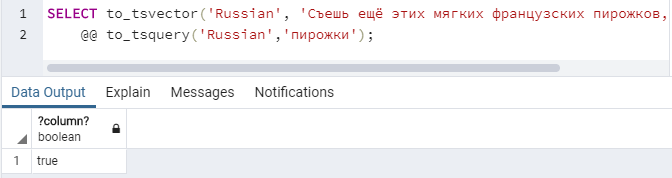
*SELECT to\_tsvector('Russian', 'Съешь ещё этих мягких французских пирожков, да выпей чаю.')*

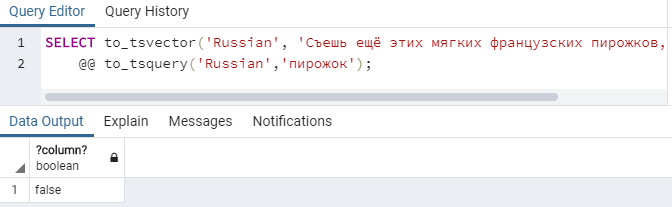
*@@ to\_tsquery('Russian','пирожки');*

*--№2*

*SELECT to\_tsvector('Russian', 'Съешь ещё этих мягких французских пирожков, да выпей чаю.')*

*@@ to\_tsquery('Russian','пирожок');*

**

**

Почему первый запрос возвращает `*true*`, а второй не возвращает?

Ответ: По той же самой причине, у слов разные лексемы.

**3. Операторы**

Выполните запрос

*--И*

*SELECT to\_tsvector('The quick brown fox jumped over the lazy dog')*

*@@ to\_tsquery('fox & dog');*

*--ИЛИ*

*SELECT to\_tsvector('The quick brown fox jumped over the lazy dog')*

*@@ to\_tsquery('fox | rat');*

*--отрицание*

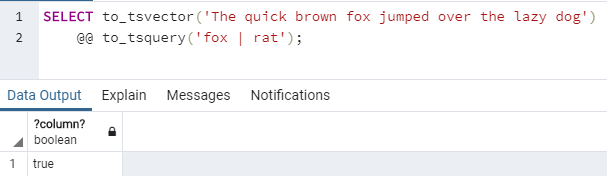
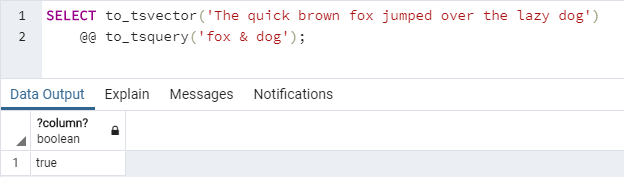
*SELECT to\_tsvector('The quick brown fox jumped over the lazy dog')*

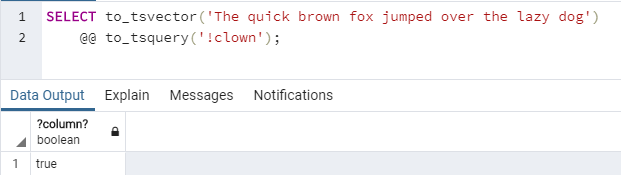
*@@ to\_tsquery('!clown');*

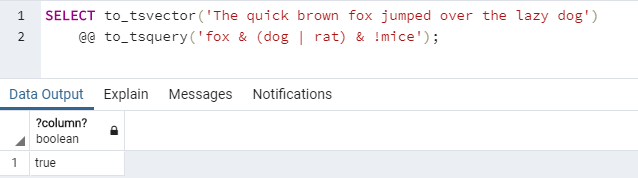
*--группировка*

*SELECT to\_tsvector('The quick brown fox jumped over the lazy dog')*

*@@ to\_tsquery('fox & (dog | rat) & !mice');*

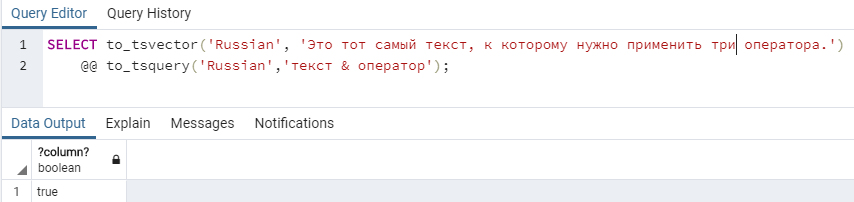
**

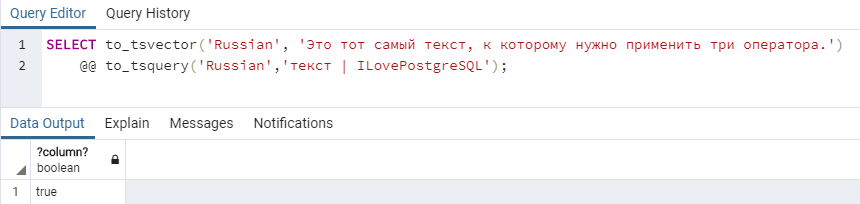
**

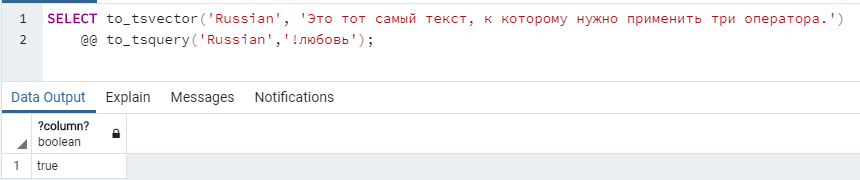
**

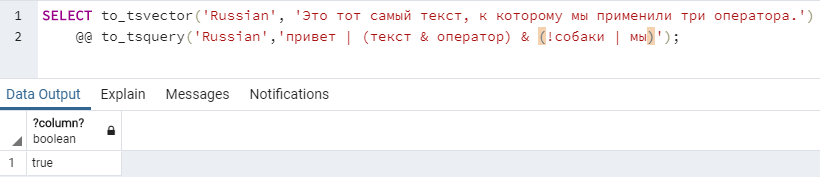
> Задание 3

1. Приведите аналогичные запросы для любого предложения на русском







**

2. Почему для английского языка не нужно указывать язык в первом аргументе и какой анализатор используется если никакой не указан?

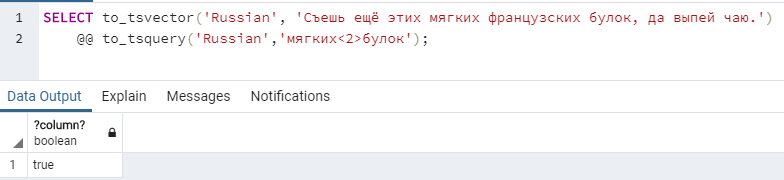
Ответ: т.к. Первым аргументом по умолчанию установлено *english* и оно будет выставлено, если явно не указать язык.

**4. Поиск фраз**

Выполните запрос

*SELECT to\_tsvector('Russian', 'Съешь ещё этих мягких французских булок, да выпей чаю.')*

*@@ to\_tsquery('Russian','мягких<2>булок');*

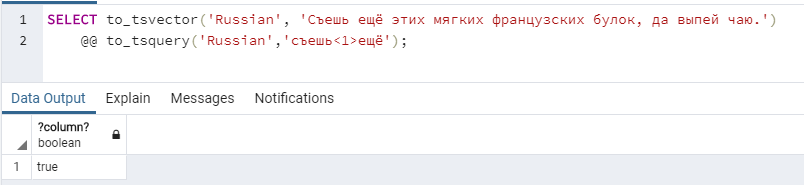


> Задание 4

1. Что означает число 2 в операторе `*<->*`

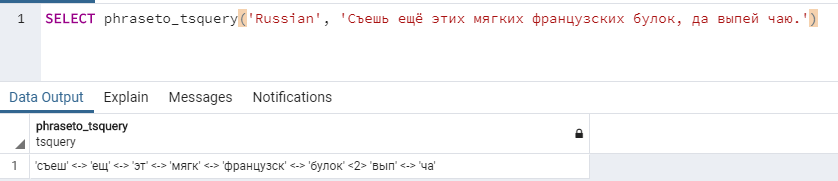
Ответ: Данный оператор помогает находить слова, которые идут друг за другом, через n значений, в данном случае n = 2

2. Модифицируйте запрос так, чтобы можно было найти фразу `*съешь ещё*`



3. Для чего нужно использовать функцию `*phraseto\_tsquery*`

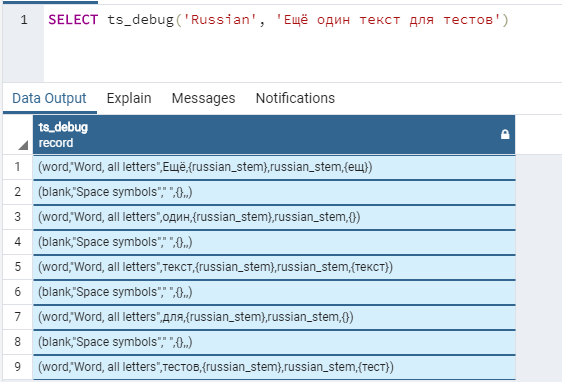
Ответ: Данная функция приводит запрос к типу данных *tsquery*. Ну и в документации написано, что он игнорирует пунктуацию



**5. Утилиты**

1. Приведите примеры использования функций `*ts\_debug*` и `*ts\_headline*`

Ответ: *ts\_debug* - отображает информацию о каждом элементе документа.



Ответ: *ts*\_*headline* - принимает документ вместе с запросом и возвращает выдержку из документа, в которой выделяются слова из запроса.

