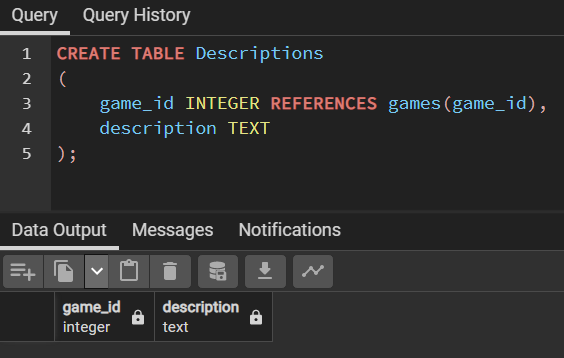
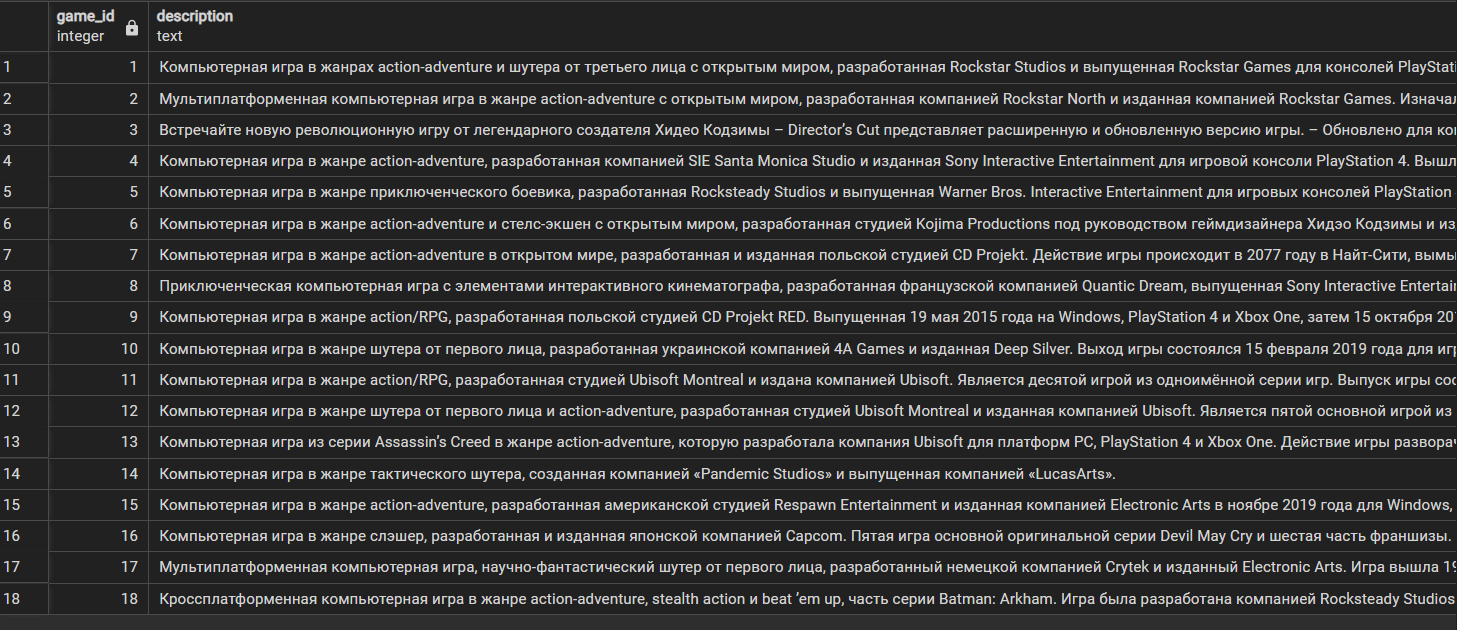
**Цель:** Узнать что такое полнотекстовый поиск и научиться им пользоваться

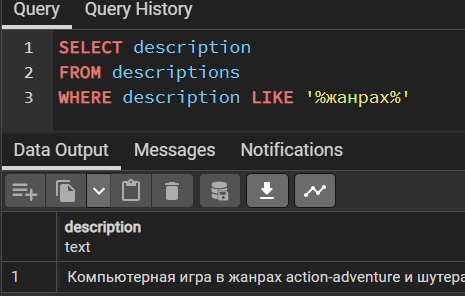
Несмотря на мощь языка запросов SQL, его возможностей не всегда достаточно для эффективной работы с данными. Особенно это стало заметно в последнее время, когда лавины данных, обычно плохо структурированных, заполнили хранилища информации. Изрядная доля Больших Данных приходится на тексты, плохо поддающиеся разбиению на поля баз данных. Поиск документов на естественных языках, обычно с сортировкой результатов по релевантности поисковому запросу, называют полнотекстовым поиском. В самом простом и типичном случае запросом считается набор слов, а соответствие определяется частотой слов в документе. Примерно таким поиском мы занимаемся, набирая фразу в поисковике Google или Яндекс.

Для изучения возможностей полнотекстового поиска создадим еще одну таблицу с описанием игр

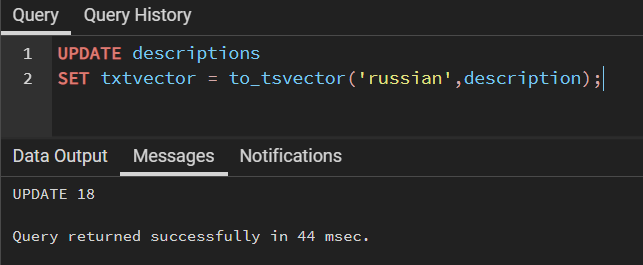
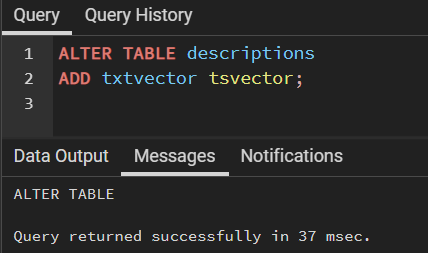
Заполним ее и проверим результат



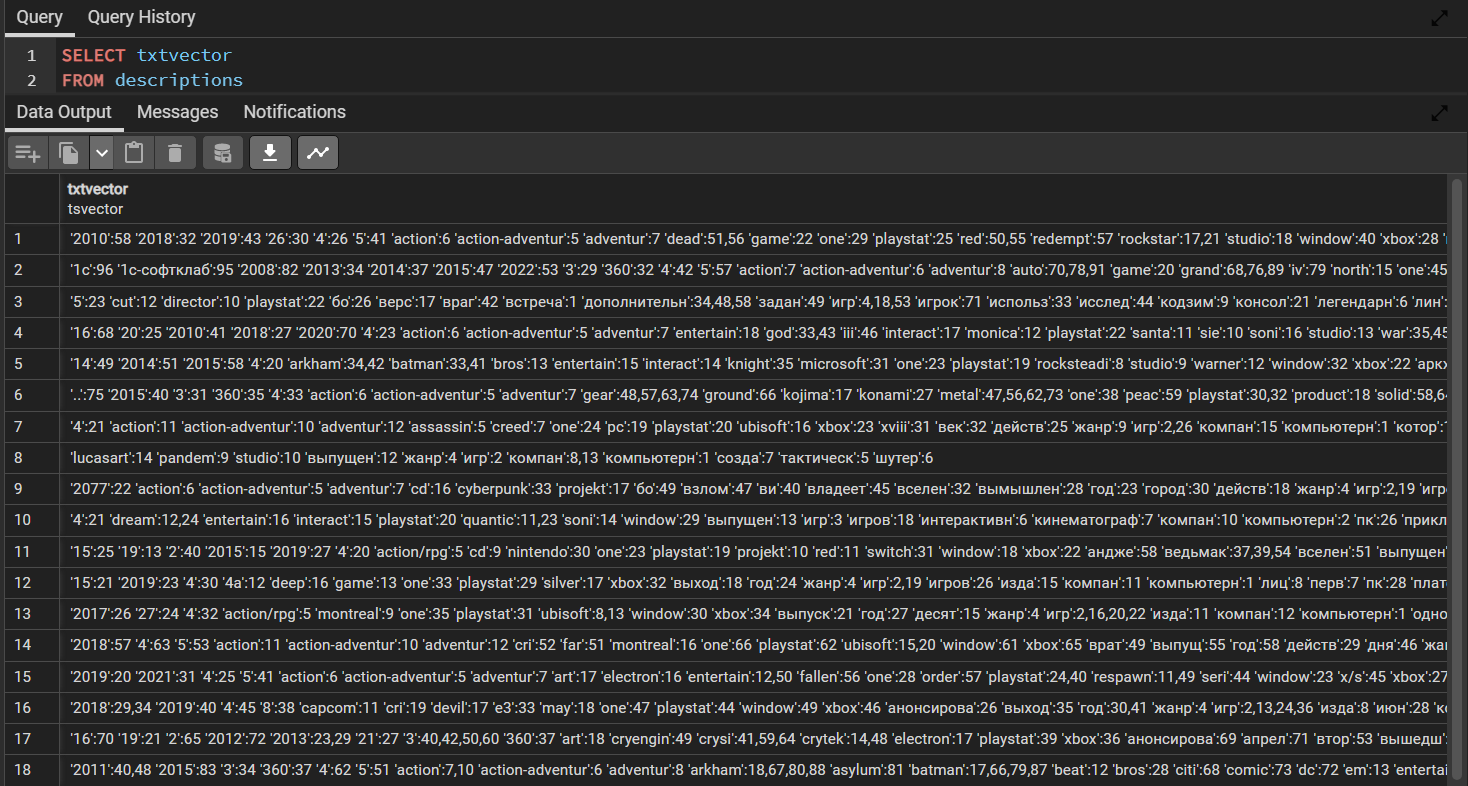
Найдем в таблице информацию по базам данных традиционными средствами SQL, используя оператор LIKE. Если ввести команду для поиска со словом “жанрах” то выйдет только одна строчка, хотя слово жанр есть в каждом описании. Ведь LIKE не знает, что в родительном падеже следует искать «жанров» или «жанр» в творительном.



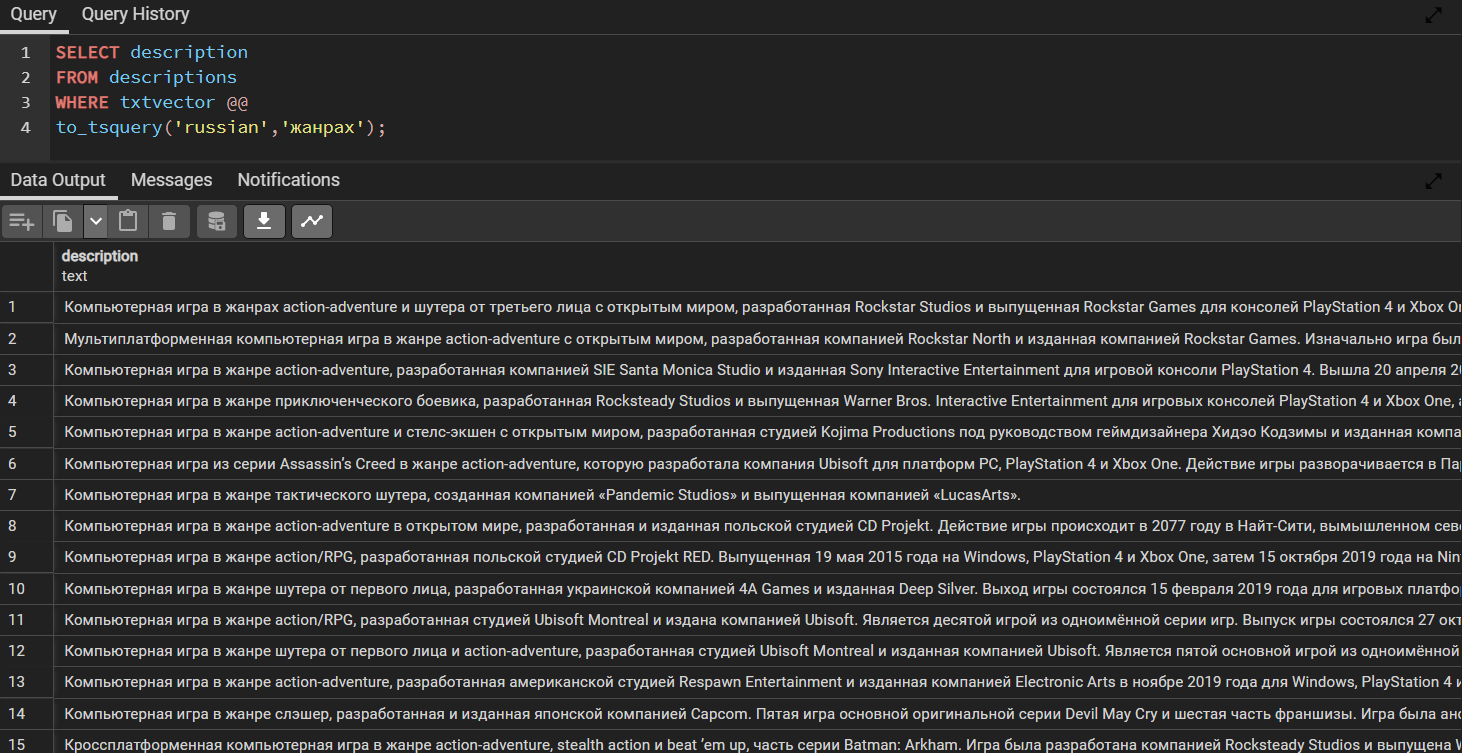
В PostgreSQL есть оператор ILIKE, который позволяет не заботиться о регистрах, а то бы пришлось еще думать и о прописных и строчных буквах. Конечно, в распоряжении знатока SQL есть и регулярные выражения (шаблоны поиска), составление которых занятие увлекательное, сродни искусству. Но когда не до искусства, хочется иметь инструмент, который думал бы за тебя. Поэтому мы добавим к таблице глав еще один столбец со специальным типом данных — tsvector:



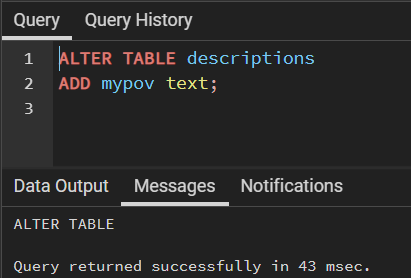
Теперь мы видим, что в строках: 1) слова сократились до своих неизменяемых частей (лексем), 2) появились цифры, означающие позицию вхождения слова в текст (видно, что некоторые слова вошли два раза), 3) в строку не вошли предлоги (а также не вошли бы союзы и прочие не значимые для поиска единицы предложения — так называемые стоп-слова).



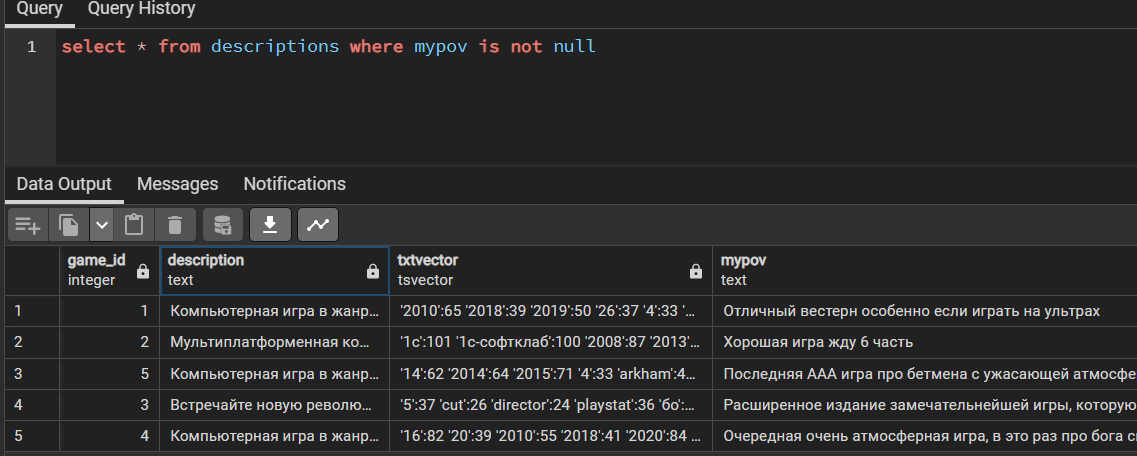
Сейчас при поиске по слову “жанрах” выходят все описания

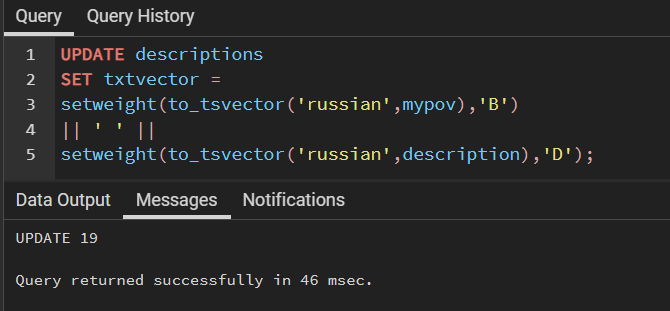


Для более продвинутого поиска нам хотелось бы включить в поисковую область и свое мнение об играх. Причем, дабы подчеркнуть их важность, мы наделим их весом при помощи функции setweight. Поправим таблицу: Добавим столбец с моим мнением о каждой игре



Заполнил частично столбец



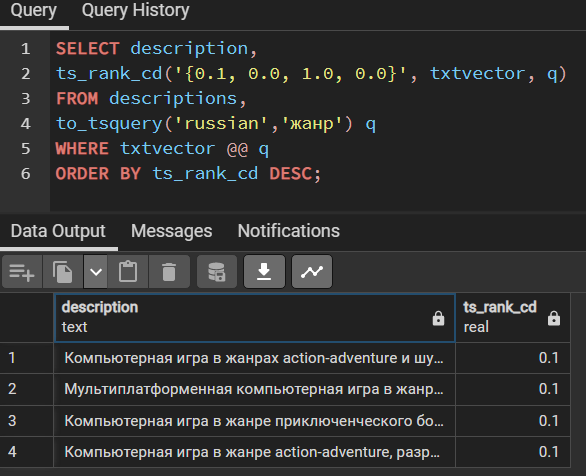
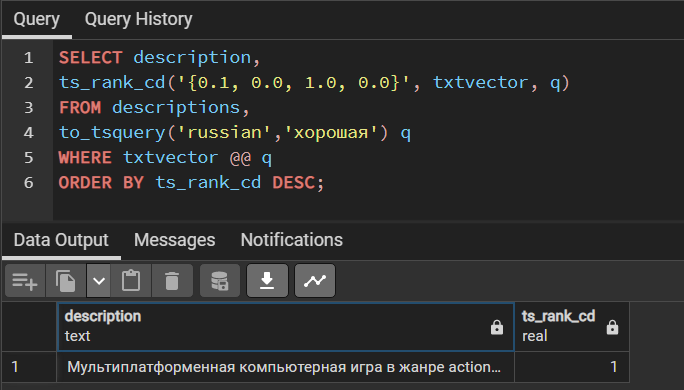
Задал вес столбцу mypov больше чем description

У лексем появился относительный вес — B (из четырех возможных — A, B, C, D).



Задал вес строке с помощью массиваю

Массив {0.1, 0.0, 1.0, 0.0} задает веса. Это не обязательный аргумент функции ts\_rank\_cd, по умолчанию массив {0.1, 0.2, 0.4, 1.0} соответствует D, C, B, A. Вес слова влияет на значимость найденной строки



Модифицировал выдачу. Будем считать, что найденные слова мы хотим выделить жирным шрифтом в странице html. Функция ts\_headline задает наборы символов, обрамляющих слово.