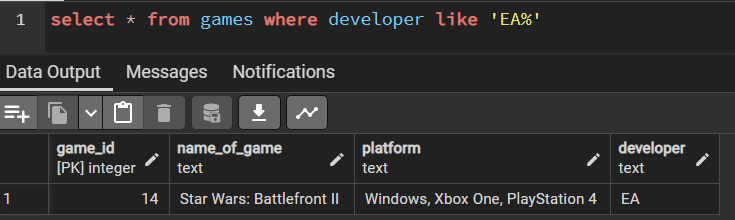
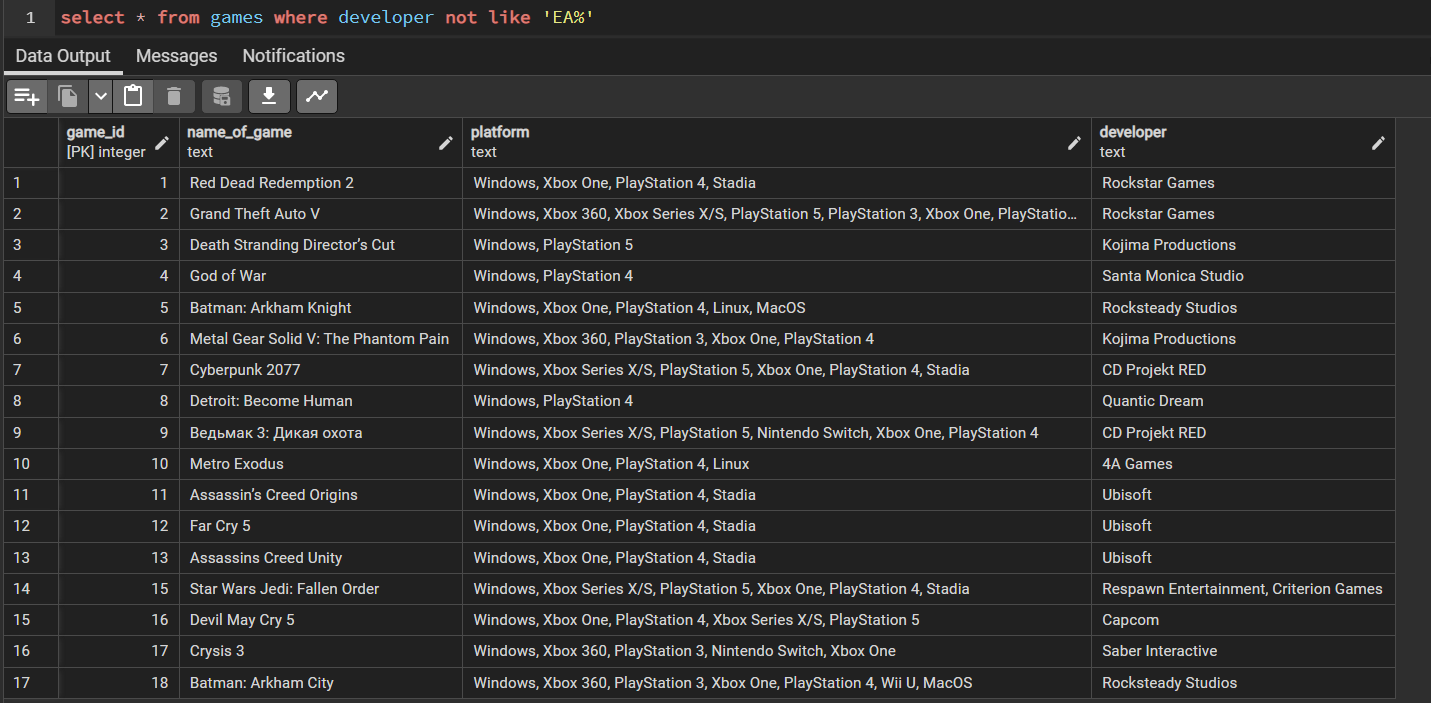
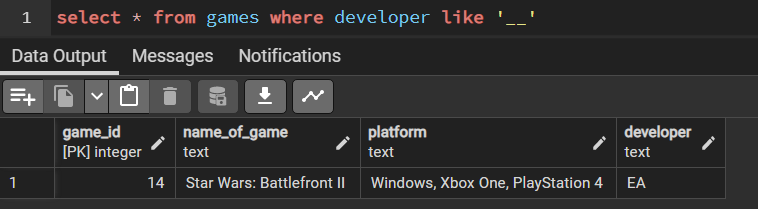
**Цель**: ознакомиться с доп. Возможностями команды select и научиться их применять

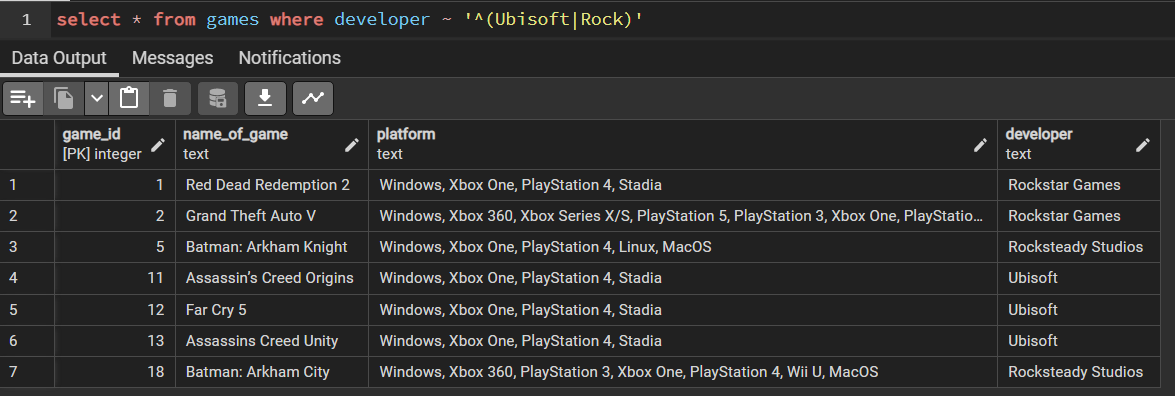
Для начала поставим перед собой такую задачу: выбрать все ишры компании EA. В этом нам поможет оператор поиска шаблонов LIKE:

Шаблон в операторе LIKE всегда покрывает всю анализируемую строку. Поэтому если требуется отыскать некоторую последовательность символов где-то внутри строки, то шаблон должен начинаться и завершаться символом «%». Однако в этом случае нужно учитывать следующие соображения. Если по тому столбцу, к которому применяется оператор LIKE, создан индекс для ускорения доступа к данным, то при наличии символа «%» в начале шаблона этот индекс использоваться не будет. Из-за этого может ухудшиться производительность, т. е. запрос будет выполняться медленнее.

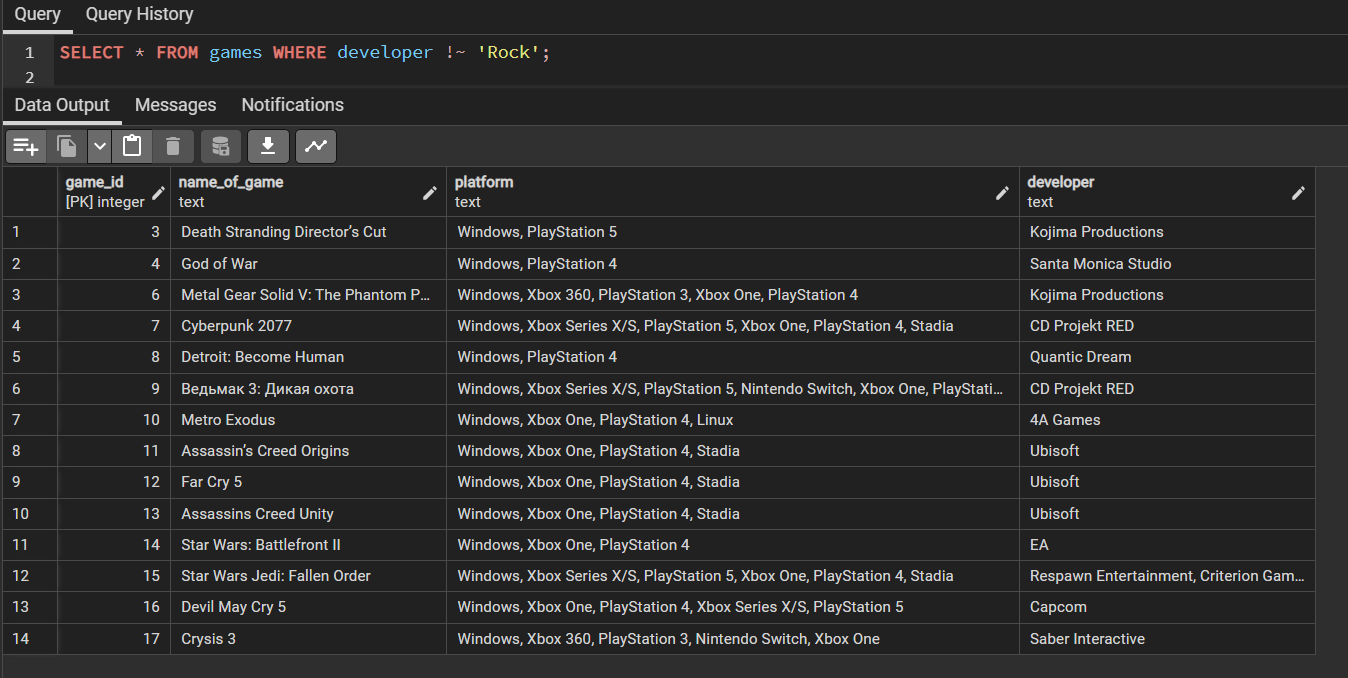
Конечно, существует и оператор NOT LIKE. Например, если мы захотим узнать, какие игры, кроме игр компании ЕА, есть в списке, то придется усложнить условие:

Кроме символа «%» в шаблоне может использоваться и символ подчеркивания — «\_», который соответствует в точности одному любому символу. В качестве примера найдем в таблице названия разработчиков игр, которые имеют названия длиной два символа (буквы). С этой целью зададим в качестве шаблона строку, состоящую издвух символов «\_». 

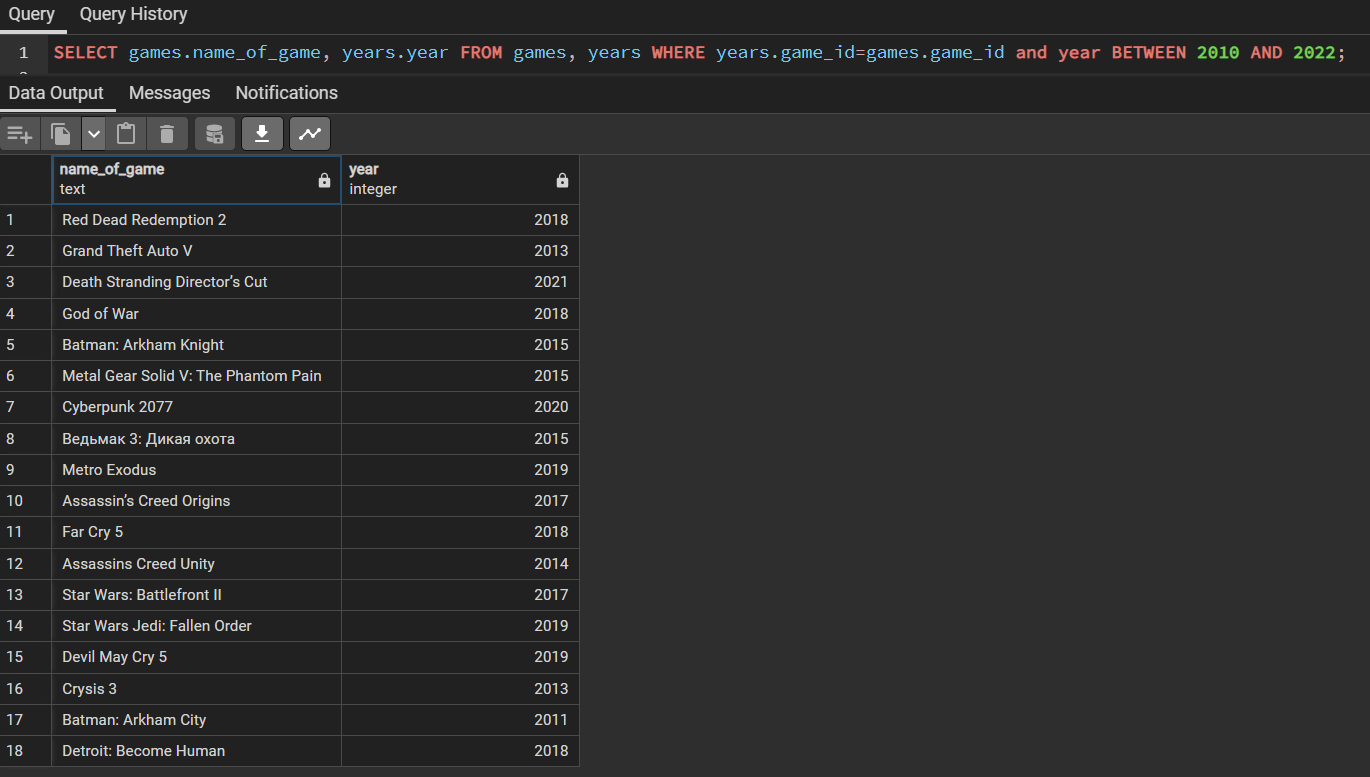
Существует ряд операторов для работы с регулярными выражениями POSIX. Эти операторы имеют больше возможностей, чем оператор LIKE. Для того чтобы выбрать, например, игры у которых разработчик Ubisoft или компания начинающаяся с Rock надо сделать так:



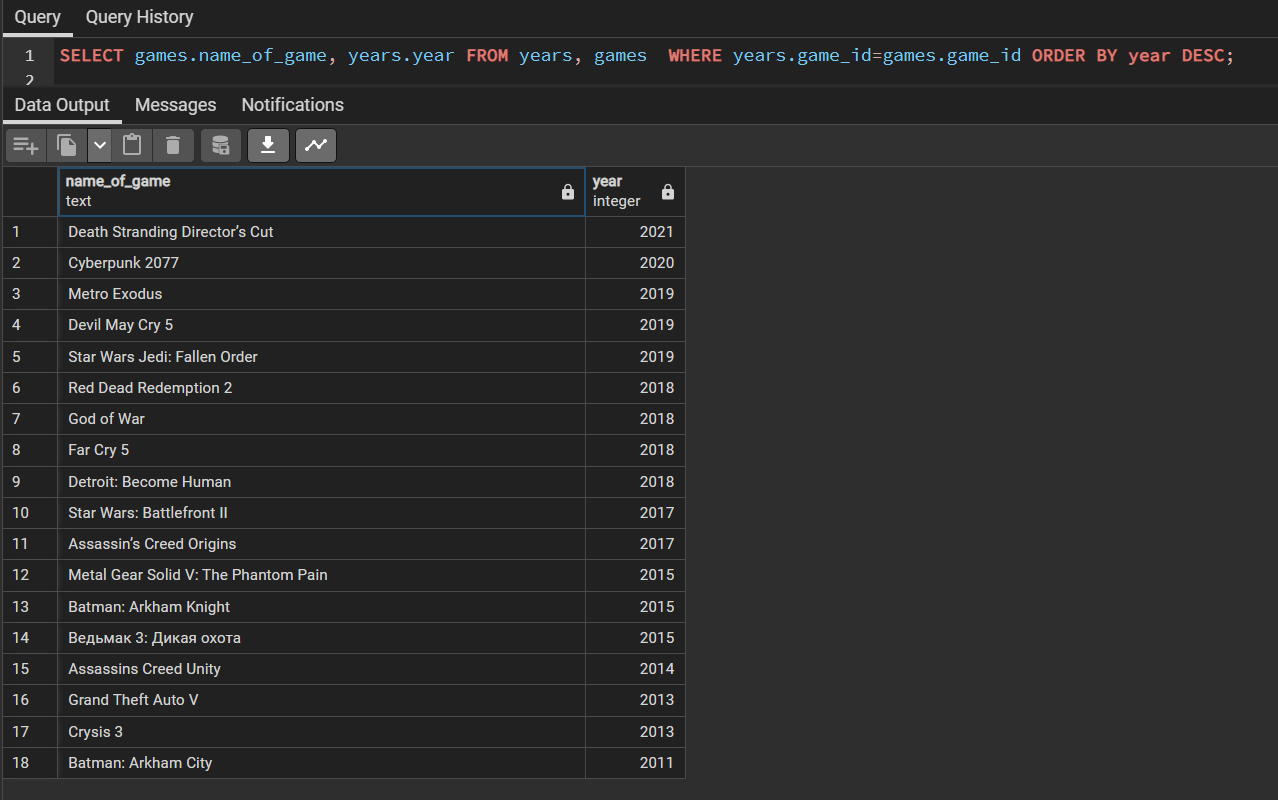
Для инвертирования смысла оператора ~ нужно перед ним добавить знак «!». В качестве примера возьмем игры где в имени разработчика нет Roc

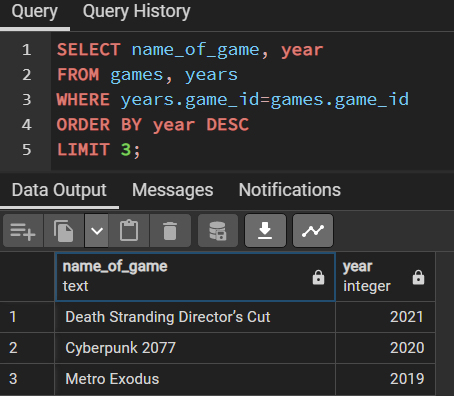


В качестве замены традиционных операторов сравнения могут использоваться предикаты сравнения, которые ведут себя так же, как и операторы, но имеют другой синтаксис. Давайте ответим на вопрос: какие игры вышлт в промежутке между 2010 и 2022? Ответ получим с помощью предиката BETWEEN.



Теперь обратимся к такому вопросу, как упорядочение строк при выводе. Если не принять специальных мер, то СУБД не гарантирует никакого конкретного порядка строк в результирующей выборке. Для упорядочения строк служит предложение ORDER BY, которое мы уже использовали ранее. Однако мы не говорили, что можно задать не только возрастающий, но также и убывающий порядок сортировки. Например, если мы захотим разместить игры по годам их выхода, то нужно сделать так:



В таблице «games» 18 записей. Если мы поставим задачу найти три самые новые игры, то для ее решения подошел бы такой алгоритм: отсортировать строки в таблице по убыванию значений столбца «год» (year) и включить в выборку только первые три строки. Для того чтобы ограничить число строк, включаемых в результирующую выборку, служит предложение LIMIT. 

А как найти еще три игры, которые вышли чуть раньше первой тройки, т. е. занимают места с четвертого по шестое? Алгоритм будет почти таким же, как в первой задаче, но он будет дополнен еще одним шагом: нужно пропустить три первые строки, прежде чем начать вывод. Для пропуска строк служит предложение OFFSET.

