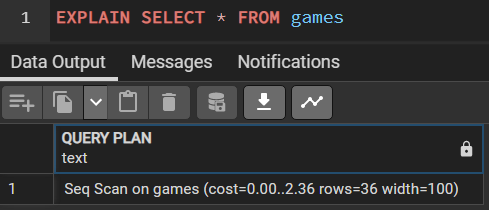
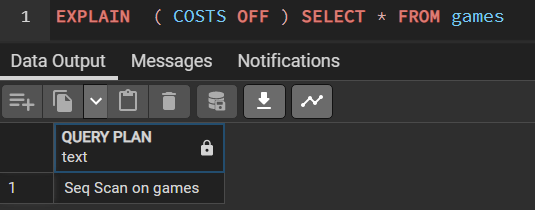
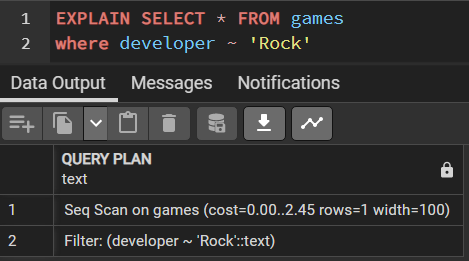
Прежде чем приступить к непосредственному выполнению запроса, PostgreSQL формирует план его выполнения. Чтобы достичь хорошей производительности, этот план должен учитывать свойства данных. Просмотреть план выполнения любого запроса можно с помощью команды EXPLAIN. Структура плана запроса представляет собой дерево, состоящее из так называемых узлов плана (plan nodes). Узлы на нижних уровнях дерева отвечают за просмотр и выдачу строк таблиц, которые осуществляются с помощью методов доступа. Если конкретный запрос требует выполнения операций агрегирования, соединения таблиц, сортировки, то над узлами выборки строк будут располагаться дополнительные узлы дерева плана. Например, для соединения наборов строк будут использоваться способы, которые мы только что рассмотрели. Для каждого узла дерева плана команда EXPLAIN выводит по одной строке, при этом выводятся также оценки стоимости выполнения операций на каждом узле, которые делает планировщик. В случае необходимости для конкретных узлов могут выводиться дополнительные строки. Самая первая строка плана содержит общую оценку стоимости выполнения данного запроса.



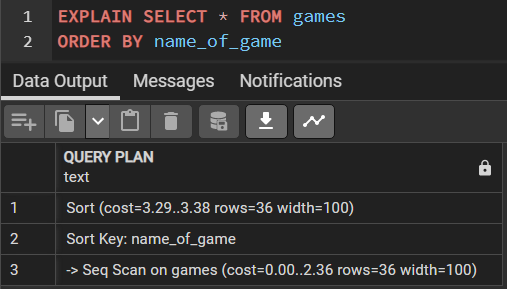
В том случае, когда нас не интересуют численные оценки, можно воспользоваться параметром COSTS OFF:



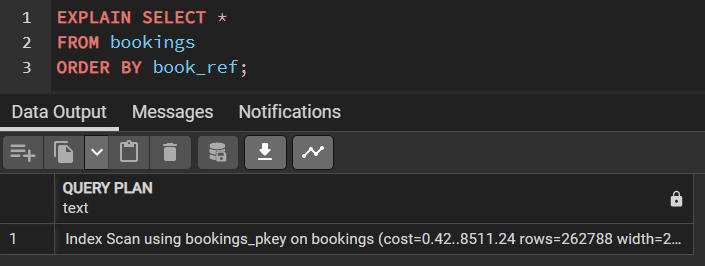
Сформируем запрос с предложением WHERE:



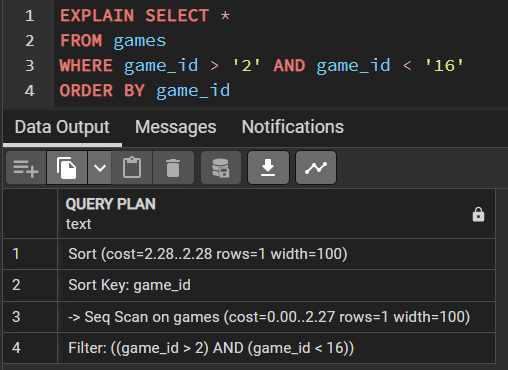
Теперь усложним запрос, добавив в него сортировку данных:



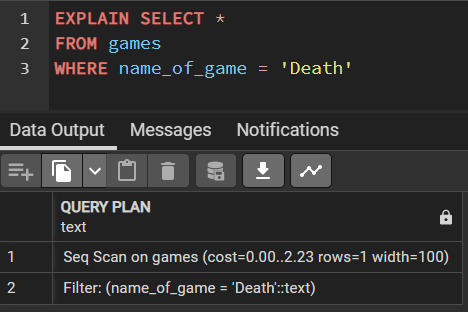
Обратимся к таблице bookings для иллюстрации сканирования по индексу.



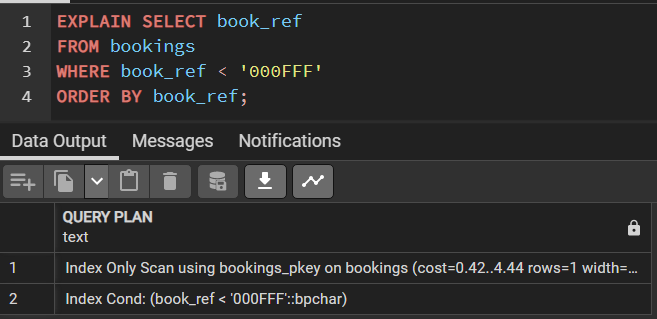
Если к сортировке добавить еще и условие отбора строк, то это отразится в дополнительной строке верхнего (и единственного) узла плана.



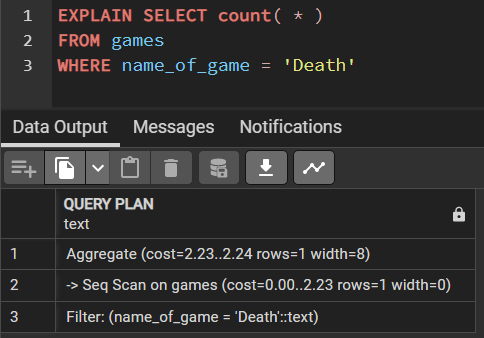
Теперь проиллюстрируем метод сканирования на основе битовой карты на примере таблицы games



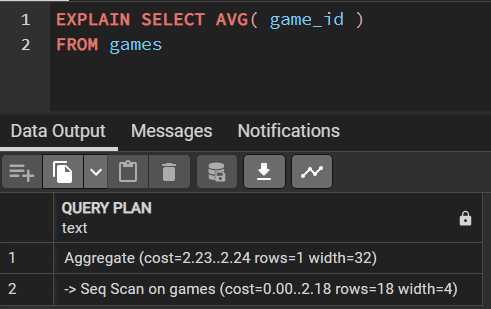
Если нам будет нужно выбрать только номера бронирований в каком-то диапазоне, то обращения к таблице не потребуется: достаточно сканирования исключительно по индексу.



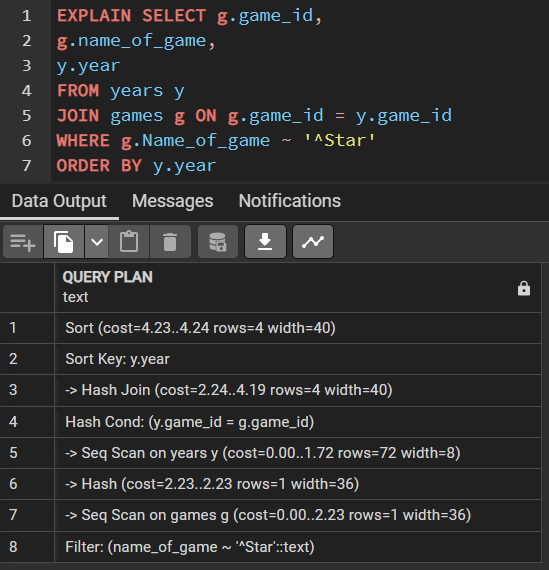
Посмотрим, как отражаются в планах выполнения запросов различные агрегатные функции. Начнем с простого подсчета строк. Должно быть 1 ведь игра в названии которой содержится Death только одна



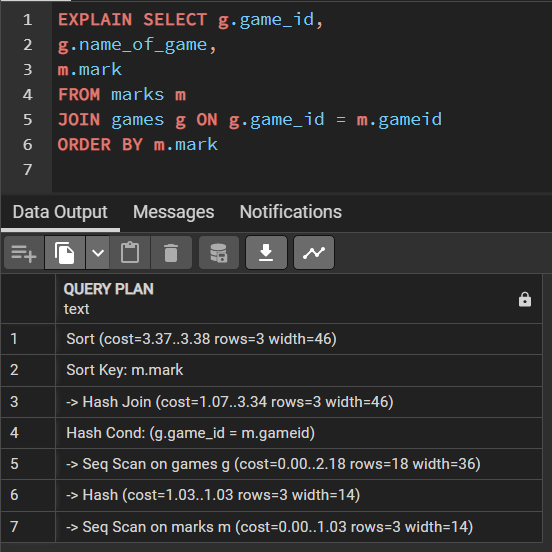
А в этом примере агрегирование связано уже с вычислениями на основе значений конкретного столбца, а не просто с подсчетом строк.



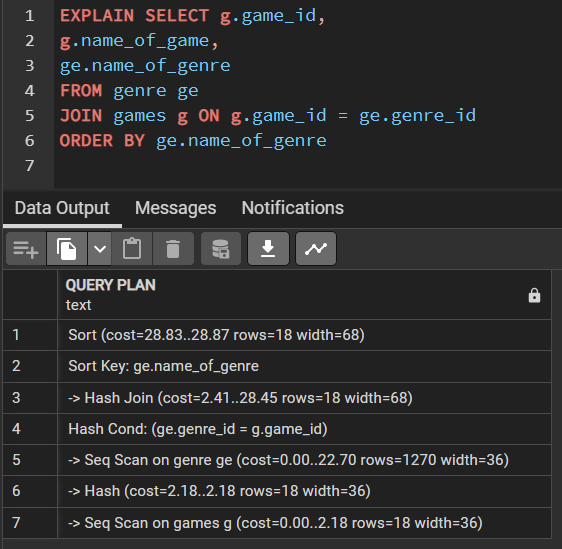
Теперь обратимся к методам, которые используются для формирования соединений наборов строк. Начнем с метода вложенного цикла (nested loop). Для получения списка игр с указанием их года выпуска сформируем запрос, в котором соединяются две таблицы: games и years



Следующий метод соединения наборов строк — соединение хешированием (hash join). Получим список игр с указанием их имен с оценками. Воспользуемся таблицами games и marks .



Последний из методов соединения наборов строк — соединение слиянием (merge join).



**Вопрос 3**

Самостоятельно выполните команду EXPLAIN для запроса, содержащего общее табличное выражение (CTE). Посмотрите, на каком уровне находится узел плана, отвечающий за это выражение, как он оформляется. Учтите, что общие табличные выражения всегда материализуются, т. е. вычисляются однократно и результат их вычисления сохраняется в памяти, а затем все последующие обращения в рамках запроса направляются уже к этому материализованному результату

