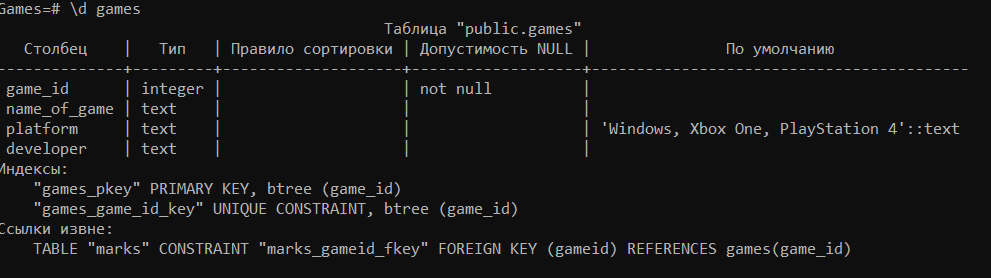
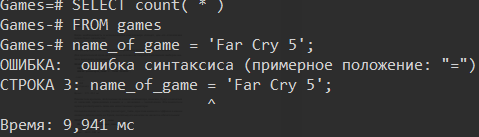
Проверяем какие индексы есть на данный момент



Включил секундомер



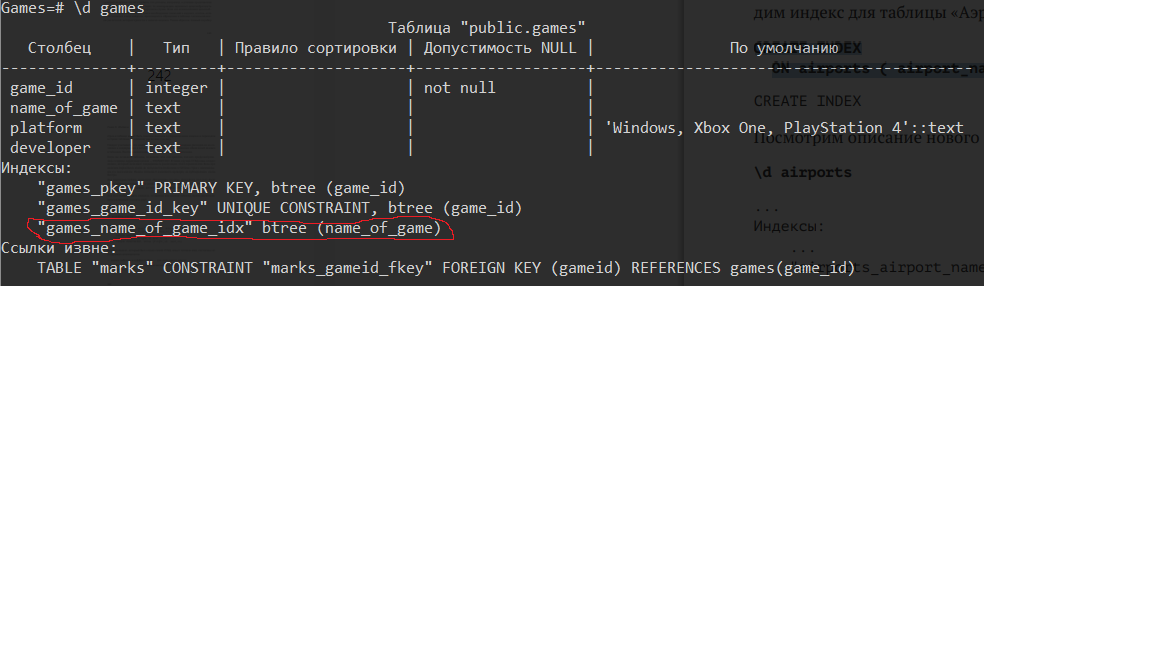
Для практической проверки влияния индекса на скорость выполнения выборок сначала выполнил следующий запрос:



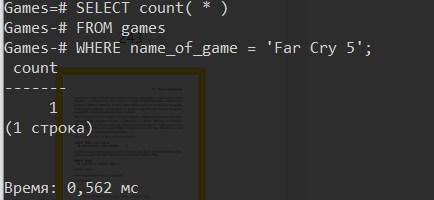
Создал индекс



Он появился



C индексом



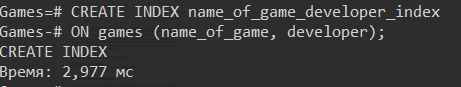
Как видно скорость обработки увеличилась примерно в 19 раз

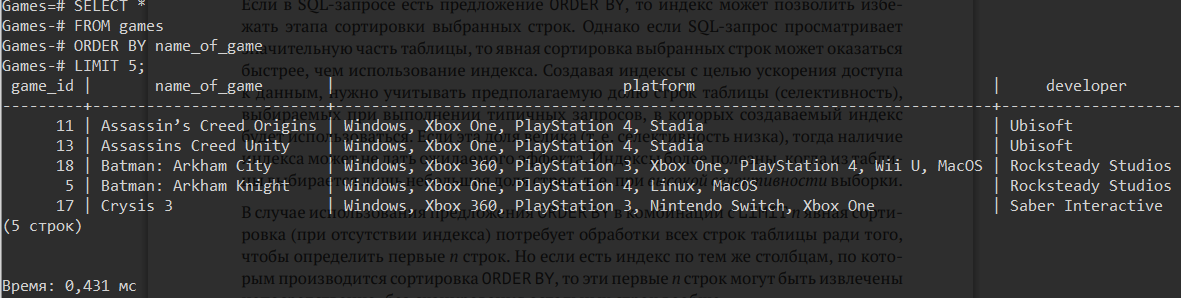
Удалил индекс



**8.2 Индексы по нескольким столбцам**

Создадал снова индекс, но теперь по двум столбцам

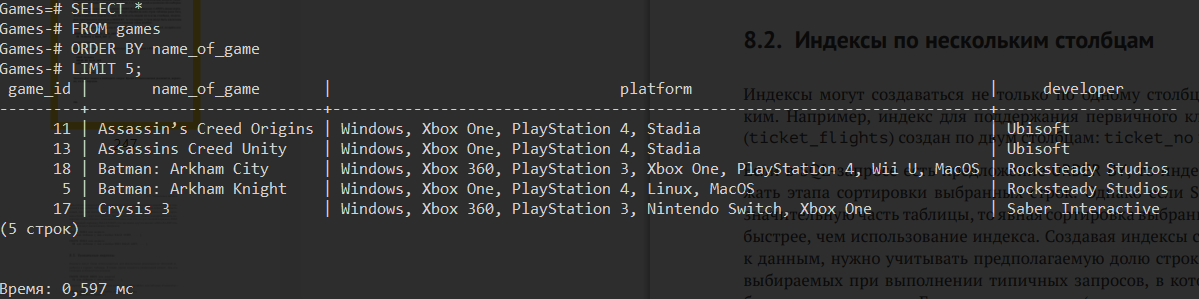


И выполнил запрос, в котором используется предложение LIMIT: 

Теперь удалил индекс

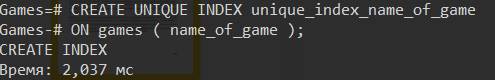


И выполнил команду снова

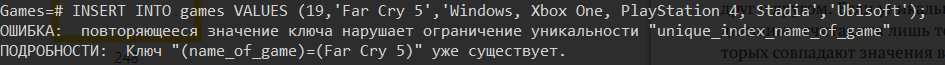
Тут разница где-то в 150 миллисекунд, но при увеличении объёмов разница станет более заметна

**8.3. Уникальные индексы**

Создадим уникальный индекс для столбца с именами игр чтобы они не повторялись

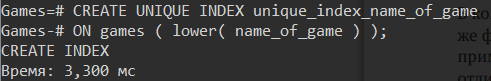


Теперь при попытки добавить значение которое уже имеется, появится ошибка:

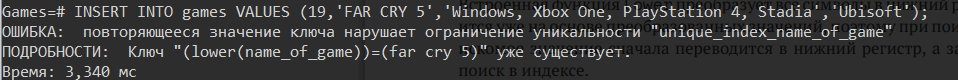


**8.4. Индексы на основе выражений**

Создадим индекс

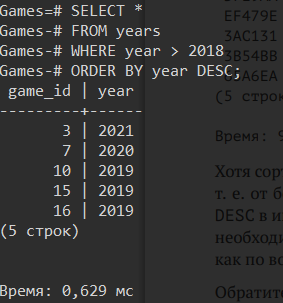


Теперь при попытке добавить игру с названием FAR CRY 5, будет выведена ошибка:

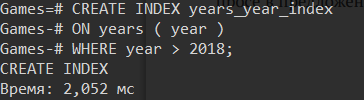


**8.5. Частичные индексы**

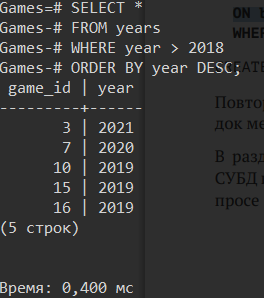
Выполним запрос в таблице years чтобы получить id игр вышедших после 2018



Создал индекс



И снова выполнил команду

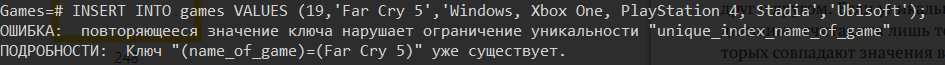


Как видно выполнилось в полтора раза быстрее

**Вопросы:**

1. Предположим, что для какой-то таблицы создан уникальный индекс по двум столбцам: column1 и column2. В таблице есть строка, у которой значение атрибута column1 равно ABC, а значение атрибута column2 — NULL. Мы решили добавить в таблицу еще одну строку с такими же значениями ключевых атрибутов, т. е. column1 — ABC, а column2 — NULL. Как вы думаете, будет ли операция вставки новой строки успешной или завершится с ошибкой? Объясните ваше решение.

Ответ: Операция вставки не будет успешной так как создан уникальный индекс. В пример могу показать данный скриншот



В таблице games уже имеется игра с названием Far Cry 5, а значение должно быть уникальным и поэтому будет появляться ошибка ограничения уникальности при попытке добавить строку с таким же значением

1. В тексте главы шла речь о выполнении одной и той же выборки из таблицы «Билеты» (tickets) при наличии индекса по столбцу passenger\_name и при его отсутствии. Вы видели, что наличие индекса ускоряет выполнение запроса почти на порядок. Проведите следующий эксперимент: выполните этот запрос несколько раз подряд при отсутствии индекса, а затем создайте индекс и опять выполните этот запрос несколько раз подряд.

SELECT count( \* ) FROM tickets WHERE passenger\_name = 'IVAN IVANOV'; Вы увидите, что время выполнения повторных запросов к таблице сокращается, причем, когда создан индекс, оно сокращается на порядок. Как вы думаете, почему?

Ответ: Подобные действия я выполнял на протяжении всей лабораторной работы и из них видно что с индексами запросы выполняются быстрее, поэтому не буду повторять. Время выполнения запросов сокращается из-за того, что индекс является упорядоченной структурой. Элементы (записи) в нем хранятся в отсортированном виде, что значительно ускоряет поиск данных в индексе. После отыскания в нем требуемой записи СУБД переходит к соответствующей строке таблицы по прямой ссылке, а не перебирает все строки, которых в таблице может быть много.

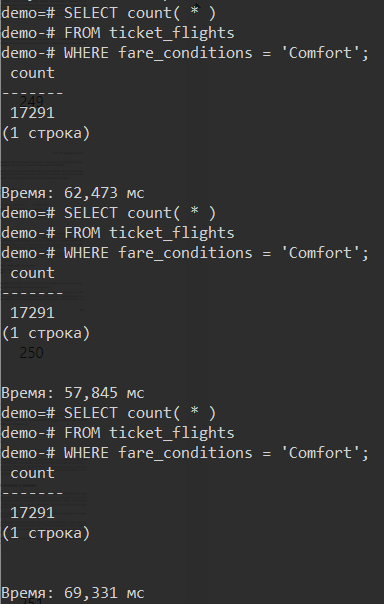
1. Известно, что индекс значительно ускоряет работу, если при выполнении запроса из таблицы отбирается лишь небольшая часть строк. Если же эта доля велика, скажем, половина строк или более, то большого положительного эффекта от наличия индекса уже не будет, а возможно даже, что не будет практически никакого эффекта. Наша задача — проверить это утверждение на практике.

Обратимся к таблице «Перелеты» (ticket\_flights). В ней имеется столбец «Класс обслуживания» (fare\_conditions), который отличается от остальных тем, что в нем могут присутствовать лишь три различных значения: Comfort, Business и Economy. Выполните запросы, подсчитывающие количество строк, в которых атрибут fare\_conditions принимает одно из трех возможных значений. Каждый из запросов выполните три-четыре раза, поскольку время может немного изменяться, и подсчитайте среднее время. Обратите внимание на число строк, которые возвращает функция count для каждого значения атрибута. При этом среднее время выполнения запросов для трех различных значений атрибута fare\_conditions будет различаться незначительно, поскольку в каждом случае СУБД просматривает все строки таблицы.

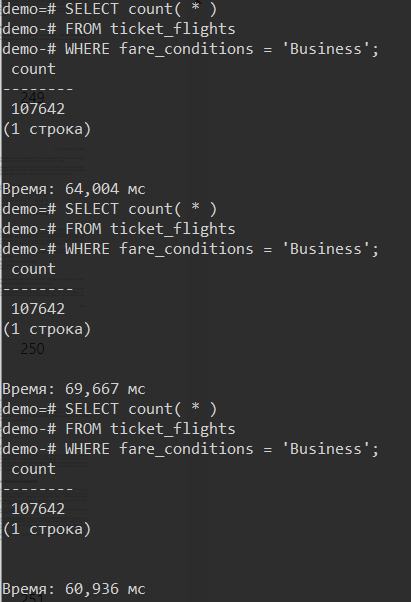
Создайте индекс по столбцу fare\_conditions. Конечно, в реальной ситуации такой индекс вряд ли целесообразен, но нам он нужен для экспериментов. Проделайте те же эксперименты с таблицей ticket\_flights. Будет ли различаться среднее время выполнения запросов для различных значений атрибута fare\_conditions? Почему это имеет место?

Ответ:

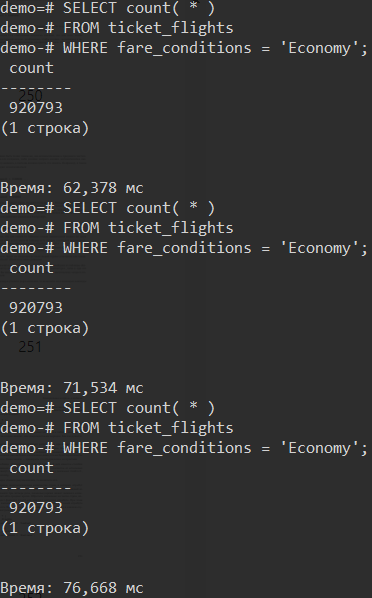
Я взял демобазу с самолётами, в которой много строк для наглядности



Среднее время = 63,2163 мс

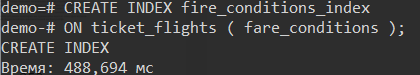


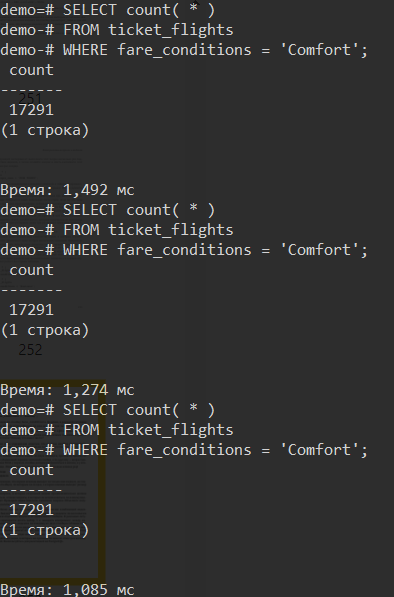
Среднее время = 64,869 мс



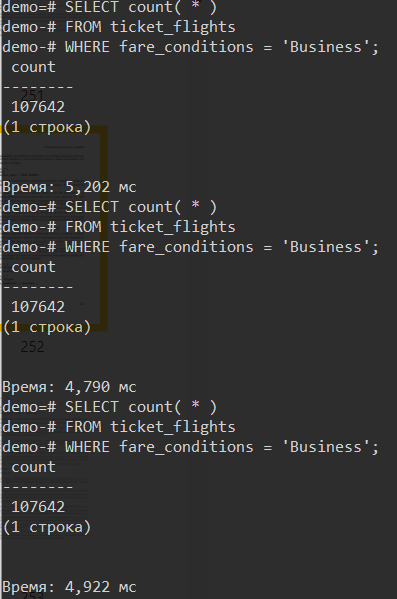
Среднее время = 70,193

Создал индекс по столбцу fare\_conditions

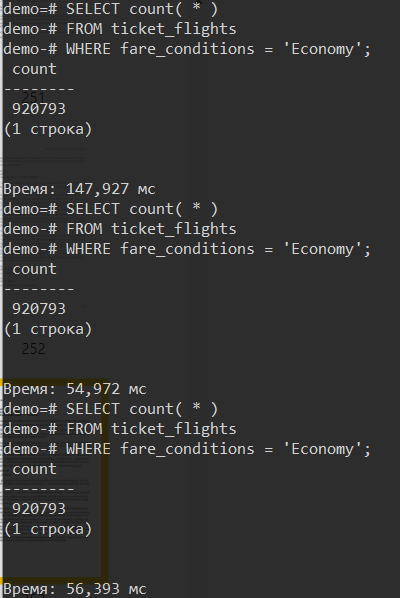




Среднее время = 1,2836



Среднее время = 4,9713

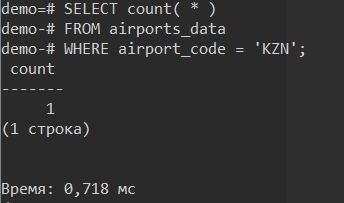


Среднее время = 86,4306

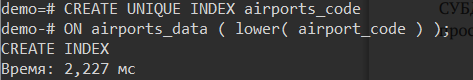
В среднем после добавления индекса запросы с относительно маленьким количеством строк начали выполняться быстрее, однако в первый раз при поиске соответствий для Economy планировщик выбрал неверный способ и запрос выполнялся гораздо дольше.

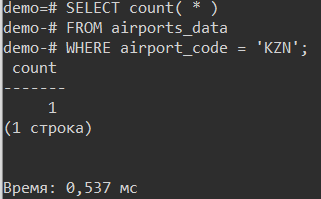
6. Предложите какую-нибудь таблицу в базе данных «Авиаперевозки» и смоделируйте ситуацию, в которой было бы целесообразно использование индекса на основе функции или скалярного выражения от двух или более столбцов.

Ответ: Например в таблице airports\_data целесообразнее использовать индексацию по выражению (на основе функции), ведь названия и коды аэропортов уникальны. Пример



Создал индекс





В итоге запрос выполнился быстрее.