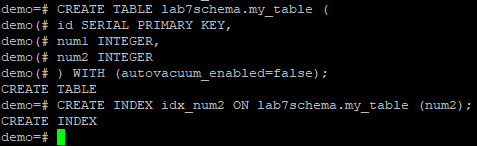
|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
|  | | |
| Лабораторная работа № 7 | | |
| по дисциплине «Администрирование информационных систем» | | |
|  | | |
| **ВЫПОЛНЕНИЕ ЗАПРОСОВ** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-02 |
| Бригада: | 8 |
| Студенты: | Сидоров Даниил, |
|  | Дюков Богдан |
| Преподаватель: | Аврунев О.Е. |
|  |  |
|
|  |  |
| Новосибирск | | |
| 2024 | | |

**Ход работы**

1. Создали таблицу со следующей структурой (вместе с autovacuum\_enabled=false):

|  |  |
| --- | --- |
| **Вариант** | **Столбцы, индексы** |
| 8 | Идентификатор и два числовых столбца индексы по второму. |

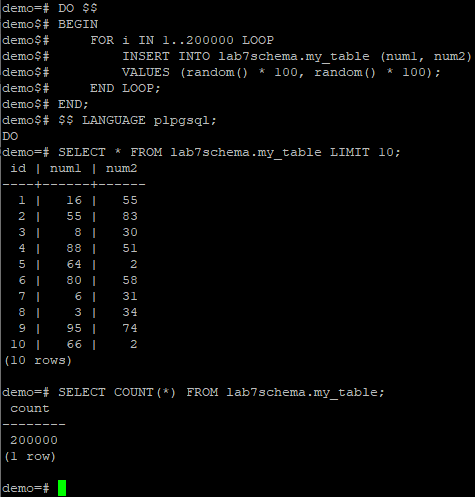




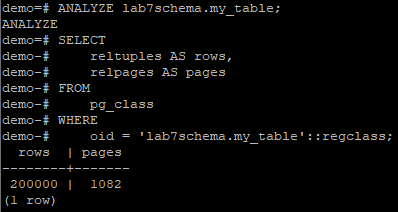
1. Заполнили таблицу данными:

|  |  |
| --- | --- |
| **Вариант** | **Количество строк** |
| 8 | 200000 |

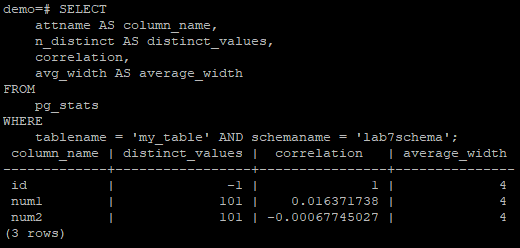
Использовали для генерации сценарий pgsql, сразу после убедились в корректности вставки:



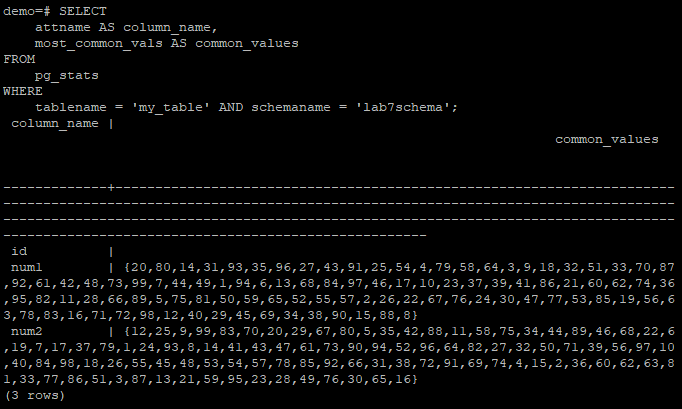
1. Привели значения статистики данных для таблицы в целом (количество строк и страниц):



Привели значения статистики данных для каждого из столбцов. Для удобства, сначала выводим: название столбца, кол-во различных значений, корреляция, ширина столбца:

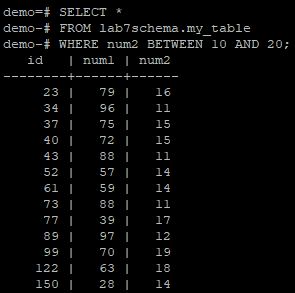


Теперь выводим: название столбца, наиболее часто встречающиеся значения в столбце:

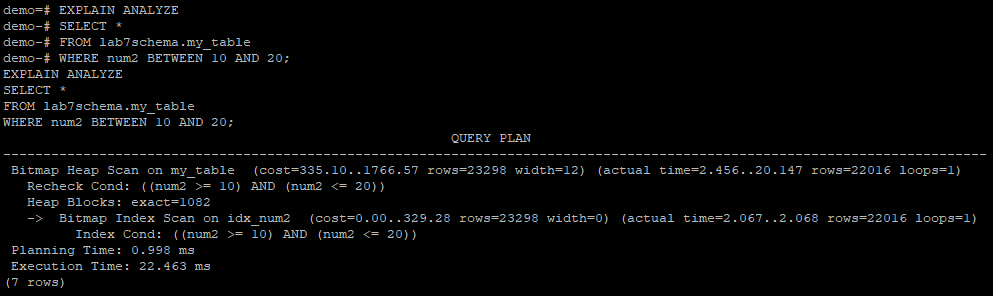


1. Выполнили запрос:

|  |  |
| --- | --- |
| **Вариант** | **Столбцы, индексы** |
| 8 | Получение строк из таблицы, условие вида between на проиндексированный столбец. |



1. Привели план выполнения и время выполнения запроса. Используем explain analyze:

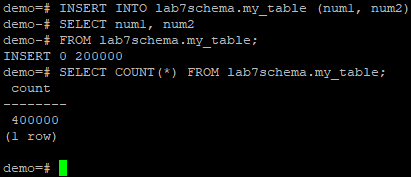


На основе полученных результатов можно сказать следующее:

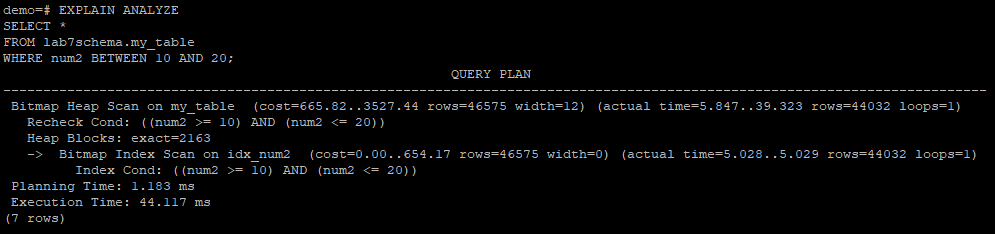
* сначала движок базы данных выполняет сканирование индекса, чтобы быстро найти строки, которые удовлетворяют условию (создается битовая карта “Bitmap”, указывающая, какие строки в таблице соответствуют условию);
* затем движок базы данных выполняет сканирование кучи (фактические данные таблицы), используя битовую карту, чтобы эффективно извлечь только те строки, которые были идентифицированы во время сканирования индекса.

Время планирования составило 0.998мс, а общее время выполнения - 22.463мс.

1. Увеличили количество строк в таблице в два раза. Для этого вставили в неё текущие строки ещё раз (сразу после вставки убедились, что строк стало действительно в 2 раза больше):



1. Повторно привели план и время выполнения запроса из п.4.

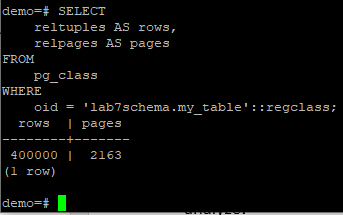


1. Собрали статистику данных для таблицы:

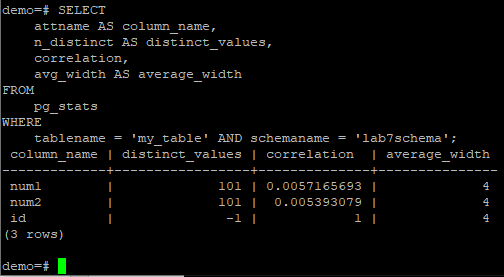


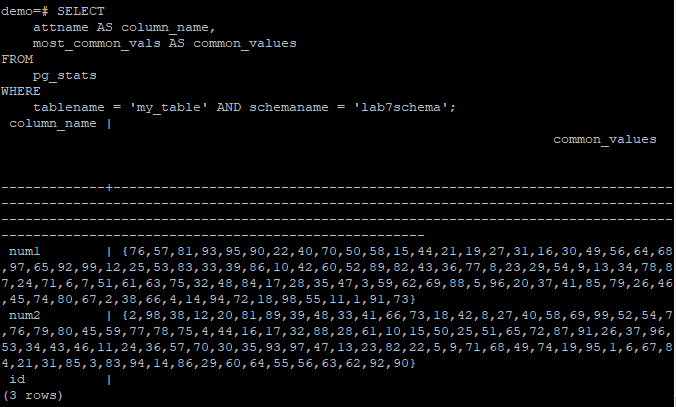
1. Привели значения статистики данных из п.3.

Статистика для таблицы в целом:

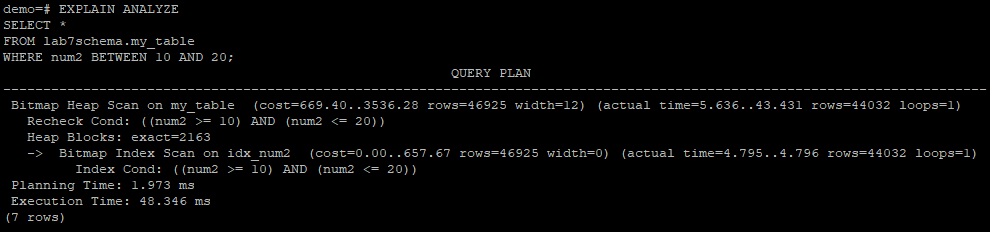


Статистика для каждого из столбцов (также разделили на два запроса):





1. Привели план и время выполнения запроса:



Сравнили планируемое и реальное время выполнения для пунктов:

**Пункт 5**:

* + Планируемое время выполнения: 1766.57 мс
  + Реальное время выполнения: 20.147 мс

**Пункт 7**:

* + Планируемое время выполнения: 3527.44 мс
  + Реальное время выполнения: 39.323 мс

**Пункт 10**:

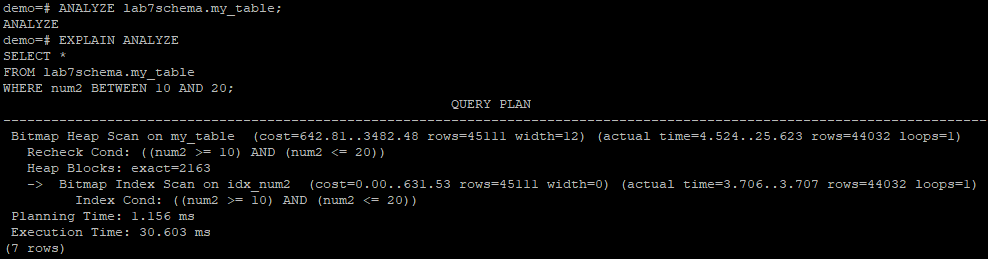
* + Планируемое время выполнения: 3536.28 мс
  + Реальное время выполнения: 43.431 мс

Как видно, реальное время выполнения каждого запроса значительно меньше планируемого времени. Планировщик запросов PostgreSQL делает консервативные оценки времени выполнения.

Также эти значения показывают, что реальное время выполнения увеличивается после вставки дополнительных строк, что ожидаемо, поскольку большее количество строк в таблице обычно приводит к большему времени выполнения запроса.

Хотя ANALYZE помогает планировщику запросов делать более обоснованные решения, он не гарантирует, что время выполнения запроса всегда будет уменьшаться. В нашем случае реальное время даже немного увеличилось (с 39 до 43 мс).

Мы можем попробовать ещё раз собрать статистику и привести план и время выполнения запроса:



В данном случае реальное время уменьшилось в разы (стало 25мс).