

Производительный SQL

Практическая работа 11.1

Practice 11.1. Планирование запросов

Запросить план для чтения и интерпретации информации, возвращаемой планировщиком.

Предположим, что мы имеем дело с базой данных **sqlida** с записями о клиентах и что компания хотела бы, чтобы мы внедрили систему для регулярного создания отчетов о деятельности клиентов в определенном географическом регионе.

Чтобы убедиться, что отчет может быть запущен своевременно, нужна оценка того, сколько времени займет **SQL-запрос**. Использовать команду **EXPLAIN**, чтобы узнать, сколько времени займут запросы отчета.

Practice 11.1. Планирование запросов

Шаги для выполнения запроса PostgreSQL:

1. Откройте PostgreSQL и подключитесь к базе данных `sqlida`.
2. Используйте команду `EXPLAIN`, чтобы вернуть план запроса для выбора всех доступных записей в таблице клиентов.
3. Прочтите выходные данные плана и определите общую стоимость запроса, стоимость установки, количество возвращаемых строк и ширину каждой строки. Проанализировать полученные величины, возвращенные из плана после выполнения этого шага?
4. Повторите запрос из шага 2, на этот раз ограничив количество возвращаемых записей до 15.

Practice 11.1. Планирование запросов

Шаги для выполнения запроса PostgreSQL:

Просмотреть обновленный план запроса, сколько шагов задействовано в плане запроса? Какова стоимость ограничивающего шага?

5. Сгенерируйте план запроса, выбрав все строки, где клиенты живут в пределах широты 30 и 40 градусов. Какова общая стоимость плана, а также количество строк, возвращаемых запросом?

Practice 11.1 Планирование запросов

Шаги для выполнения запроса PostgreSQL:

```
-----  
QUERY PLAN  
-----  
Seq Scan on customers (cost=0.00..1786.00 rows=26439 width=140)  
  Filter: ((latitude > '30'::double precision) AND (latitude < '40'::double precision))  
(2 rows)
```

Решение Practice 11.1 Планирование запросов



Шаги для выполнения запроса PostgreSQL:

-- Используйте команду EXPLAIN, чтобы вернуть план запроса для выбора всех доступных записей в таблице клиентов.

```
EXPLAIN SELECT * FROM customers;
```

-- Повторите запрос из шага 2 этого упражнения, на этот раз ограничив количество возвращаемых записей до 15.

```
EXPLAIN SELECT * FROM customers LIMIT 15;
```

-- Создайте план запроса, выбрав все строки, где клиенты живут в пределах широты 30 и 40 градусов.

```
EXPLAIN SELECT * FROM customers WHERE latitude > 30 and latitude < 40;
```

Практическая работа 11.2

Practice 11.2 Реализация сканирования индекса



Определить, можно ли использовать сканирование индекса для сокращения времени запроса. После создания нашей системы отчетности о клиентах для отдела маркетинга, мы получили еще один запрос, позволяющий идентифицировать записи по их IP-адресам или связанным именам клиентов. Мы знаем, что существует много разных IP-адресов, и нам нужен эффективный поиск. Спланируйте запросы, необходимые для поиска записей по IP-адресу, а также для определенных клиентов с суффиксом Jr в имени.

Practice 11.2 Реализация сканирования индекса

Шаги для выполнения запроса PostgreSQL:

1. Используйте команды **EXPLAIN** и **ANALYZE**, чтобы профилировать план запроса для поиска всех записей с IP-адресом **18.131.58.65**. Сколько времени занимает планирование и выполнение запроса?
2. Создайте общий индекс на основе столбца IP-адреса.
3. Повторите запрос шага 1. Сколько времени занимает планирование и выполнение запроса?
4. Создайте более подробный индекс на основе столбца IP-адреса с условием, что IP-адрес равен **18.131.58.65**.

Practice 11.2 Реализация сканирования индекса

Шаги для выполнения запроса PostgreSQL:



5. Повторите запрос шага 1. Сколько времени занимает планирование и выполнение запроса? Каковы различия между каждым из этих запросов?
6. Используйте команды **EXPLAIN** и **ANALYZE** для профилирования плана запроса для поиска всех записей с суффиксом **Jr**. Сколько времени занимает планирование и выполнение запроса?
7. Создайте общий индекс на основе столбца адреса суффикса.
8. Повторно запустите запрос шага 6. Сколько времени занимает планирование и выполнение запроса?

Practice 11.2 Реализация сканирования индекса

QUERY PLAN

```
-----  
Bitmap Heap Scan on customers (cost=5.12..318.44 rows=107 width=140) (actual time=0.146..0.448 rows=102 loops=1)  
  Recheck Cond: (suffix = 'Jr'::text)  
  Heap Blocks: exact=100  
    -> Bitmap Index Scan on ix_jr (cost=0.00..5.09 rows=107 width=0) (actual time=0.092..0.092 rows=102 loops=1)  
        Index Cond: (suffix = 'Jr'::text)  
Planning Time: 0.411 ms  
Execution Time: 0.511 ms  
(7 rows)
```

Практическая работа 11.3

Practice 11.3 Реализация сканирования индекса



Исследовать использование хэш-индексов для повышения производительности с помощью таблицы **emails** из базы данных **sqlida**.

Мы получили очередной запрос от отдела маркетинга. На этот раз они хотели бы, чтобы мы проанализировали эффективность маркетинговой кампании по электронной почте. Учитывая низкий уровень успеха кампаний по электронной почте, многим клиентам одновременно отправляется много разных писем.

Используйте команды **EXPLAIN** и **ANALYZE**, чтобы определить время и стоимость планирования, а также время и стоимость выполнения для выбора всех строк, где тема электронного письма «**Шокирующая экономия на электросамокатах в отпуске**».

Practice 11.3 Реализация сканирования индекса

Шаги для выполнения запроса PostgreSQL:



1. Используйте команды **EXPLAIN** и **ANALYZE**, чтобы определить время и стоимость планирования, а также время и стоимость выполнения для выбора всех строк, где тема электронной почты — «**Шокирующая экономия на праздничных покупках электрических самокатов**» в первом запросе и «**Черная пятница**». **Зеленые автомобили**. во втором запросе.
2. Создайте хеш-сканирование в столбце **email_subject**.
3. Повторите шаг 1. Сравните выходные данные планировщика запросов без **хэш-индекса** с выходными данными с хэш-индексом. Как повлияло сканирование хэша на производительность двух запросов?
4. Создайте хэш-сканирование столбца **customer_id**.
5. Используйте **EXPLAIN** и **ANALYZE**, чтобы оценить, сколько времени потребуется, чтобы выбрать все строки со значением **customer_id** больше 100. Какой тип сканирования использовался и почему?

Practice 11.3 Реализация сканирования индекса

Шаги для выполнения запроса PostgreSQL:



QUERY PLAN

```
Seq Scan on emails (cost=0.00..10651.98 rows=417309 width=79) (actual time=0.024..121.483 rows=417315 loops=1)
  Filter: (customer_id > 100)
  Rows Removed by Filter: 843
Planning Time: 0.199 ms
Execution Time: 152.656 ms
(5 rows)
```