

Программа Перезапуск

Модуль SQL.

Занятие 6

Преподаватель: Марат Гарафутдинов

Что смотрели на предыдущих занятиях

- Почему SQL нужен
- Реляционные таблицы
- Базовые операции
 - SELECT, INSERT, UPDATE, DELETE
- Операторы
 - WHERE, GROUP BY, AND, OR, NOT
- Операции JOIN
- Подзапросы
- Конструкции CASE WHEN
- Операторы ANY, ALL, IS NOT NULL
- Конструкции LIKE
- Конструкции CREATE TABLE & VIEW
- Оконные функции (ROW_NUMBER, RANK, DENSE_RANK)

Работаем с DBeaver Community Edition

Реквизиты:

52.157.159.24

login: student0

password: student0_password

database: sql_ex_for_student0

Но сейчас работаем только со своей базой!



Индексы

Индекс (англ. index) — объект базы данных, создаваемый с целью повышения производительности поиска данных.

Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному критерию путём последовательного просмотра таблицы строка за строкой может занимать много времени.

Индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и, таким образом, позволяет искать строки, удовлетворяющие критерию поиска.



Индексы

Что следует индексировать:

- Столбцы, используемые для объединения таблиц
- Столбцы, используемые для ограничения диапазона данных, которые анализируются при выполнении запросов
- Столбцы, используемые в директивах ORDER BY и GROUP BY запросов
- Столбцы, используемые в функциях суммирования и подведения итогов



Индексы

Что не следует индексировать:

- Таблицы содержат незначительное количество строк
- Столбцы имеют слабо выраженную селективность (т.е. очень широкий диапазон значений)
- Значения в столбцах имеют очень большую длину (не рекомендуется индексировать столбцы с значениями длиннее 25 байт)
- Столбцы при построении запросов не используются



Индексы

Индекс требует ресурсов на поддержание.

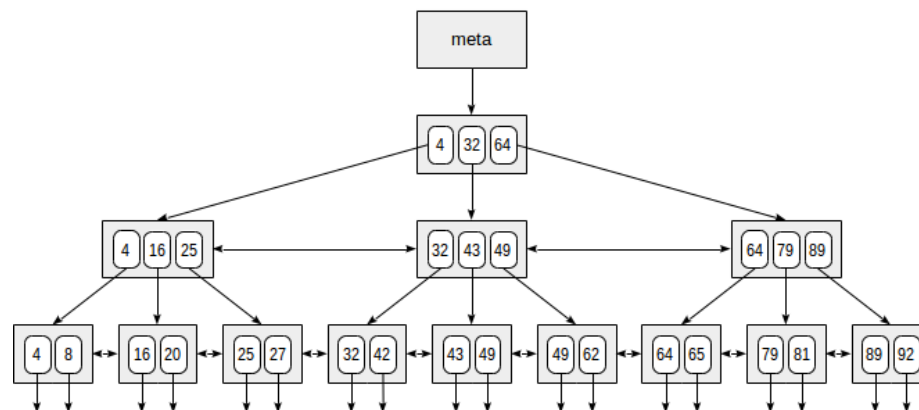
Ведь если мы изменили что-то в индексируемом столбце, нужно перестроить структуру данных которой представлен индекс.

Так, если, например, имеет место индекс на поле, которое редко используется в запросах, но при этом это поле часто обновляется, то мы скорее получим не прирост к производительности, а ее деградацию.

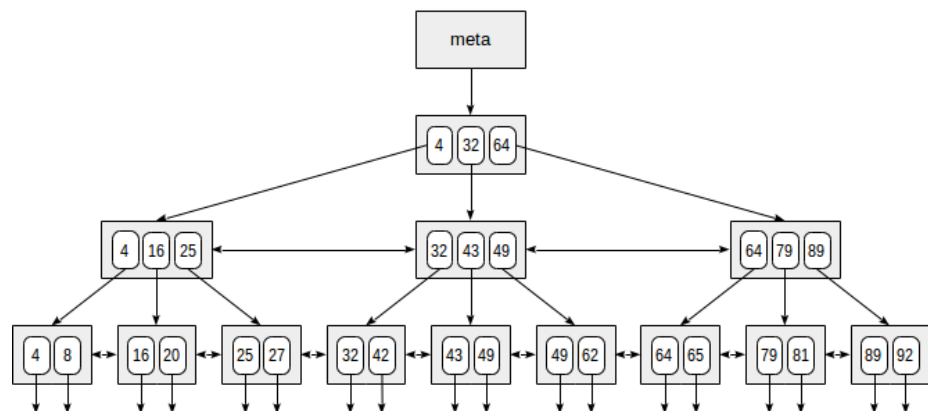
Если мы затрагиваем не индексируемые столбцы, то перестроение не требуется.



Индексы: B-tree



Индексы: B-tree



Семейство B-Tree индексов — это наиболее часто используемый тип индексов, организованных как сбалансированное дерево, упорядоченных ключей.

Они поддерживаются практически всеми СУБД как реляционными, так нереляционными, и практически для всех типов данных.

<https://postgrespro.ru/docs/postgresql/9.6/sql-createindex>

<https://postgrespro.ru/docs/postgresql/9.6/sql-dropindex>

Создание индекса-B-дерева по столбцу title в таблице films

```
CREATE UNIQUE INDEX title_idx ON films (title);
```

Пример удаления

```
DROP INDEX title_idx;
```

Оптимизация запросов

Ключ к оптимизации лежит через команду [EXPLAIN](#)

Перед выполнением запроса sql сервер разрабатывает *план запроса*

План составляется исходя из структуры запроса и данных.

<https://postgrespro.ru/docs/postgrespro/10/using-explain>



Оптимизация запросов

Есть специальная программа внутри sql сервера, которая перебирает разные варианты выполнения запроса, и принимает какой-то из планов исходя из весовой функции.

Например, планировщик может решить такие вопросы как “нужно ли использовать имеющийся индекс?” “какой индекс использовать?” “применить сортировку с помощью индекса или обычную, просканировав всю таблицу?” “на каком этапе выполнения лучше применить фильтрацию?”



Оптимизация запросов

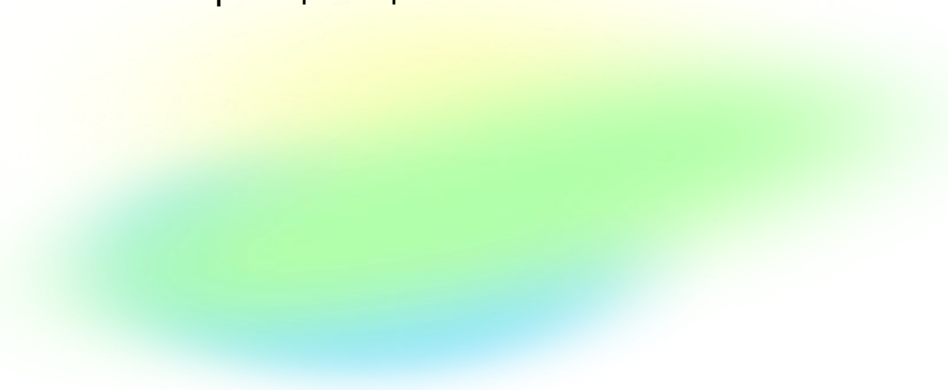
EXPLAIN ANALYZE + запрос

фактически выполнит запрос, и теперь в плане будет отражена информация о времени выполнения каждого узла плана, а также количество строк.

Оценка стоимости узла выражается в произвольных единицах, а вот время в миллисекундах, поэтому эти числа вряд ли будут совпадать.

Нам важно проверить оценку количества строк в каждом узле и фактическое значение.

Так мы сможем примерно осознать насколько адекватно планировщик оценивает выполнение данного запроса.



Оптимизация запросов

Выполнение EXPLAIN ANALYZE может существенно отличаться от реального выполнения запроса, например, ввиду задержек сети.

Кроме того, стоит тестировать на реальных данных, потому что выполнение существенно зависит от фактических данных.

Если все данные помещается в одну страницу памяти, то практически всегда будет иметь место полное сканирование таблицы, даже если в таблице есть индекс.



Linux: Работаем с Google Cloud Shell

<https://console.cloud.google.com/>

pwd текущая директория

ls список файлов и директорий в текущей (ls -al выведет красивее)

cp <a> копирует <a> в

cp -R <a> рекурсивно копирует <a> в

mv <a> перемещает <a> в

mkdir <имя папки> создать папку

cd <path to dir> перейти в папку по адресу <path to dir>

touch filename создаст пустой файл с именем filename

man <имя команды> руководство по команде

также рекомендуется гуглить первое время

Linux: Работаем с Google Cloud Shell и PostgreSQL

<https://console.cloud.google.com/>

Команда: `psql -h 52.157.159.24 -Ustudent0 sql_ex_for_student0`

Далее вводим пароль

Демонстрация и решение упражнений