



IT-ОБРАЗОВАНИЕ
В ПЕТЕРБУРГЕ
И УДАЛЕННО

Занятие 7. Производительность запросов

Преподаватель: Владимир
Кривец

Вспоминаем пройденное

Занятие 6

- Объединение результатов нескольких запросов
- Вынесение подзапроса в оператор WITH
- Аналитические функции
- Оконные функции
- Иерархические запросы
- Практика



Занятие 7.

Производительность запросов



План

- Что такое производительность запросов и что на нее влияет?
- Оптимизатор запроса
- Анализ плана запроса
- Индексы и статистики
- Типы данных - влияние на объем БД и производительность.
- Конкурентный доступ к данным, транзакции
- Практика. Оптимизация производительности запросов в своей БД.

Ресурсы необходимые для выполнения запроса



- Память – данные нужно где-то сохранить
 - Оперативная – быстрая
 - Долговременная(Диски) – медленная
- Процессор – на обработку каждой строки(чтение, фильтрация, вывод) тратится какое-то кол-во тактов процессора
- Время – на выполнение запроса требуется время
 - Время за которое отработает запрос зависит от **объема данных** в этом запросе, **сложности** запроса и **мощности сервера**



Что такое производительность запросов

- Кол-во запросов в единицу времени?
- Объем выводимых данных\ кол-во строк в ед. времени?
- Время выполнения запросов?
- Потребление памяти вашими запросами?

- Все эти определения верны.
 - Нам не важно, что такое производительность, пока мы не испытываем с ней проблем.

Главный индикатор того, что есть проблемы с производительностью – это время за которое отрабатывают ваши запросы.

Что влияет на производительность запросов



- Железо
- Объем данных
- Сложность запроса
- SQL – декларативный язык, СУБД сама решает как именно она будет выполнять запросы.
 - На производительность запросов влияет то, как именно СУБД будет выполнять запрос (План запроса).
 - План запроса зависит от:
 - Текста запроса (Select top 1 from table vs Select * from table1 inner join table 2 on t1)
 - Кол-ва \ объема данных
 - Индексов и статистики
 - Типов данных
- Конкурентный доступ к данным, блокировки



Выполнение SQL запроса

- Оптимизатор запросов парсит текст SQL запроса
 - Поиск и обработка команд, проверка существования объектов, считывание тех. Информации и тд
- Оптимизатор запросов составляет план запроса
- СУБД выполняет запрос согласно плану
- Существуют
 - Предполагаемый план запроса
 - Актуальный план запроса
- План запроса состоит из **операторов** обработки данных



Порядок обработки операторов при парсинге запроса

1. FROM – найти таблицу(ы)
 1. ON – узнать как соединять
 2. JOIN – узнать тип соединения
2. WHERE – узнать какие фильтры применить
3. GROUP BY – узнать как группировать
 1. HAVING – узнать как фильтровать агрегаты
4. SELECT – узнать какие поля нужно выводить
5. ORDER BY – узнать как сортировать



Порядок выполнения запроса

1. Чтение данных из таблиц
2. Все остальное
 - Соединение данных
 - Расчет констант
 - Группировка
 - Вывод

Чтение данных из таблиц, которые находятся на диске – самый медленный процесс в БД, тк скорость чтения с дисков на порядки меньше скорости обработки данных в оперативной памяти.



План выполнения запроса

План выполнения определяет, как именно будет происходить:

1. Чтение данных из таблиц
 1. Сканирование \ поиск по индексу
2. Все остальное
 - Соединение данных
 - Способ соединения данных
 - Расчет констант
 - Группировка
 - Вывод

План выполнения определяет какие операторы обработки данных будут использоваться



Анализ плана выполнения запроса

- План выполнения запроса состоит из операторов обработки данных, соединенных стрелками - «потоками данных».
- Главная цель анализа плана запроса – найти самый медленный оператор (или группу операторов) и понять, почему был выбран такой оператор и что можно изменить.
 - Для каждого оператора в плане указаны его относительные «стоимость» и доп. тех параметры
 - Для каждого оператора и потока данных в плане запроса указаны ожидаемые: кол-во строк, объем данных и тд.

Выполнение SQL запроса



- Оптимизатор запросов парсит текст SQL запроса
 - Поиск и обработка команд, проверка существования объектов, считывание тех. Информации и тд
- Оптимизатор запросов составляет план запроса
- СУБД выполняет запрос согласно плану



План выполнения запроса

План выполнения определяет, как именно будет происходить:

1. Чтение данных из таблиц
 1. Сканирование \ поиск по индексу
2. Все остальное
 - Соединение данных
 - Способ соединения данных
 - Расчет констант
 - Группировка
 - Вывод

План выполнения определяет какие операторы обработки данных будут использоваться

Чтение данных из таблиц



Есть 2 способа чтения данных из таблицы

1. Сканирование – чтение всех строк, строчка за строчкой
2. Поиск по индексу – чтение нужных строк

Игра:

Ведущий загадал число, от 1 до 100.

Игрок может говорить ответы(13?, 15?)

Ведущий может отвечать только так - загаданное число

больше или меньше предложенного.

За какое минимальное кол-во попыток можно гарантированно

угадать число?

Что такое индекс



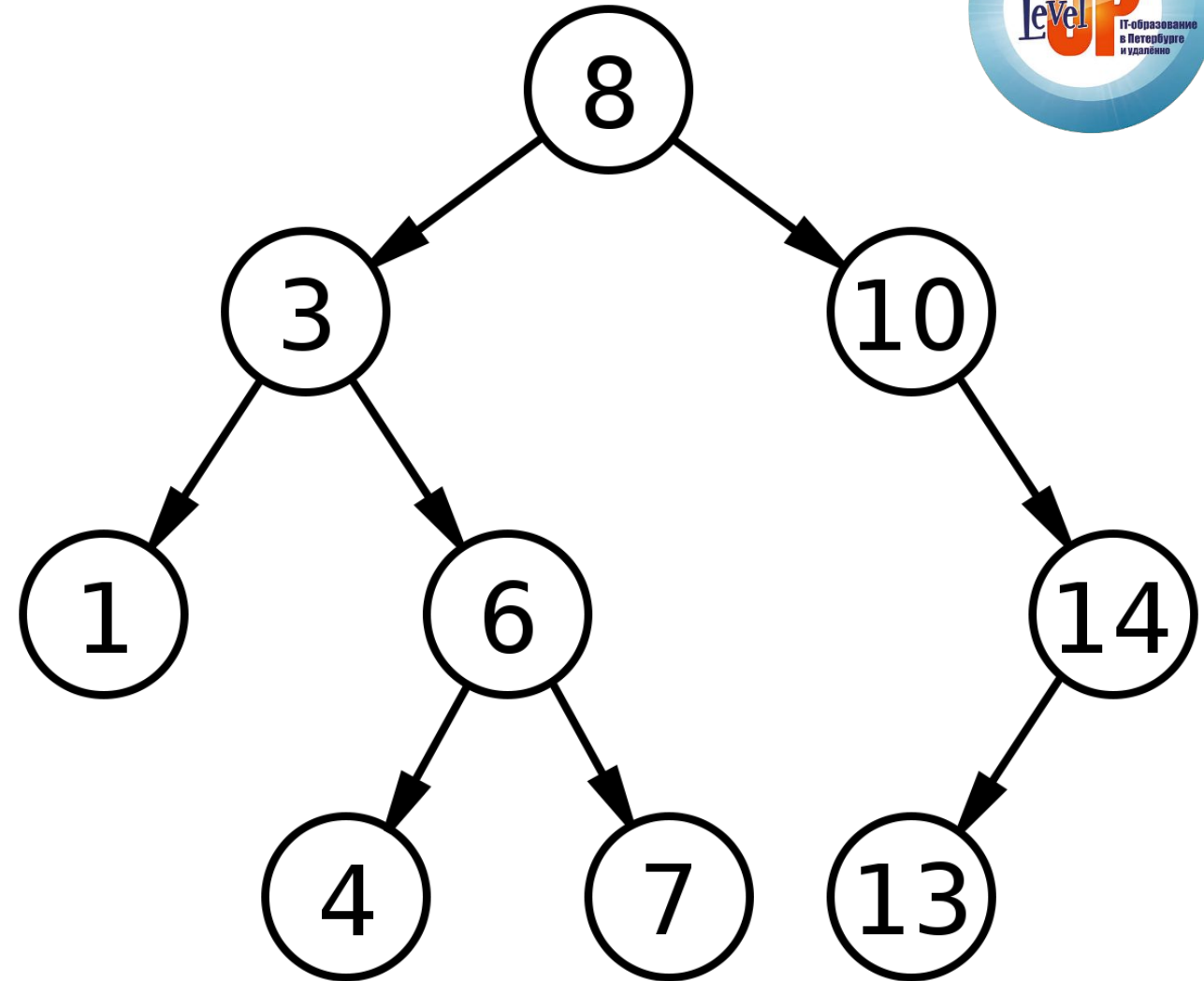
Индекс – бинарное дерево поиска.

У индекса есть узлы.

Поиск по индексу быстрее полного сканирования.

Например, для поиска нужного числа среди 1 000 000 элементов, нужно:

1. 1 000 000 итераций при полном сканировании
2. 20 итераций при поиске по индексу



Индексы в MS SQL Server



1. Кластерный индекс – структура, в которой хранится вся таблица, отсортированная по ключу. Кластерный индекс может быть только один.
2. Некластерный индекс – структура рядом с таблицей, в которой хранятся только значения ключа и ссылки на основную таблицу.
3. Куча – таблица без индексов



Чтение данных из таблиц

1. Оптимизатор запросов выберет сканирование таблицы в случаях:
 1. Когда у таблицы нет индексов
 2. Когда в запросе нет фильтров
 3. Когда объем данных в таблице достаточно мал
 4. Когда фильтры не используют ключи индекса
 5. Когда предикат соединения не содержит ключи индекса
 6. Когда нужно вывести слишком много данных
2. Оптимизатор запросов выберет поиск по индексу в случаях:
 1. Когда у таблицы есть индекс, есть фильтр по ключевым полям этого индекса и ожидаемый объем данных не большой.
 2. Когда предикат соединения содержит ключи индекса и ожидаемый объем данных не большой.



Статистики

Как оптимизатор понимает, сколько данных будет участвовать в запросе

Статистики – это *примерная* информация о данных в таблице, которую СУБД собирает самостоятельно в фоновом режиме.

Статистику можно построить для:

- Индекса
- Столбца
- Набора столбцов

Статистика разбивает все данные на 200 границ и сохраняет информацию о том, сколько примерно строк в каждой границе.

Оптимизатор запросов обращается к статистике во время построения плана.



Типы данных и их влияние на производительность

Типы данных столбцов таблицы влияют на производительность:

1. Из-за выделяемой памяти
 1. Вывод поля `nvarchar(1000)` потребует $2 * 1000 * N$ Байт, даже если в поле хранятся строки не более 5 символов
2. Из-за преобразования типов
 1. Если при фильтрации или соединении поле приводится к другому типу данных (явно или не явно), то СУБД не может доверять статистике по этому полю. Это приводит к сканированиям.

Влияние конкурентного доступа



Блокировка данных транзакциями

1. Транзакции могут накладывать блокировки на строки и таблицы.
2. Заблокированные одной транзакцией данные становятся недоступными для других транзакций.
3. Все прочие транзакции ожидают завершения первой транзакции и снятия блокировок.

Выход – переходить на другую модель изоляций транзакций.

Оптимистичная модель не содержит блокировок.