



ОНЛАЙН-ОБРАЗОВАНИЕ

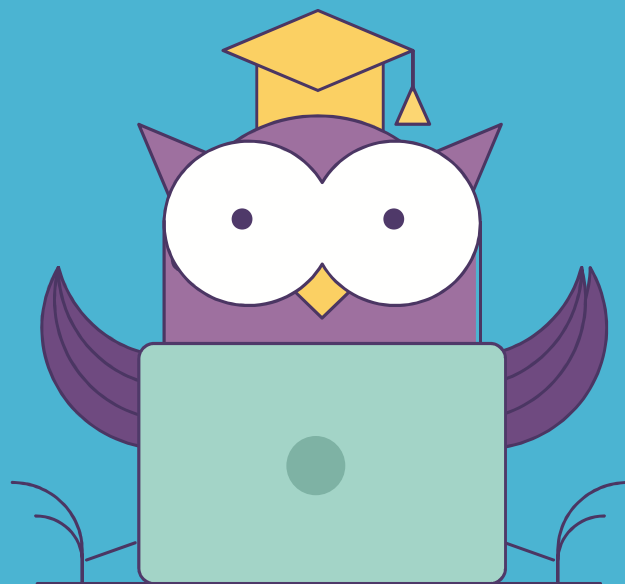
Введение в планы запросов

Курс «Разработчик MS SQL Server»

Занятие № 4



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте ☐ + если все хорошо
Ставьте ☐ - если есть проблемы

Цель вебинара

По окончании вебинара вы сможете:

- читать и объяснять план запроса
- оценивать разные запросы и сравнивать их стоимость



Порядок выполнения



1. Чем CTE отличается от представления?
2. Чем коррелированный запрос отличается от независимого?
3. Можно ли в одном представлении делать выборку из другого представления?

Читайте ~~матчасть~~ план запроса. Узнаете

1. Какой индекс используется
2. В каком порядке делается join
3. Что выбирается из буффера
4. Сколько сервер тратит ресурса на операцию
5. Разницу между гипотетическим и реальным планом



План запроса – это то как сервер будет выбирать данные физически, план действий.



SQL Server:

- генерирует много планов
- выбирает 1 с наименьшей стоимостью (CPU, IO, память)

Сервер создает не самый лучший план, а

Самый лучший план за самое короткое время

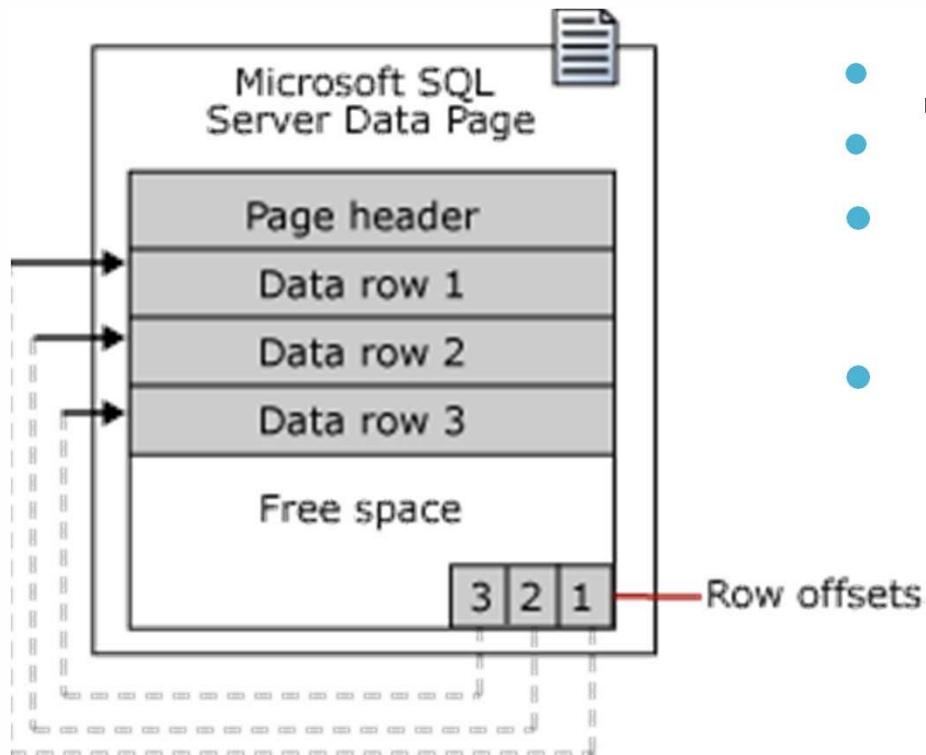


Understanding SQL Server Query Execution Plans

[Richard Douglas](#)

Оператор	Описание
Table Scan	Можно увидеть только на куче, перебор элементов всей таблицы
Index Scan	Перебор всех элементов индекса
Index Seek	Поиск по индексу
Key lookup	Довыборка данных, которых не хватает в индексе, из таблицы
RID lookup	Довыборка, которых не хватает в индексе, данных из кучи
Nested loop	Соединение циклом, значения из одной таблицы циклически ищутся в другой
Merge join	Соединение по индексу
Hash join	Соединение с построением хэш таблицы
Compute scalar	Вычисление значения
Stream aggregate	Аггрегатная функция
Sort	Сортировка

Размер страницы в SQL Server 8 Кб, типы страниц



- Данные
- Индекс
- Глобальная карта распределения
- Карта распределения индекса

Строка не может находиться на нескольких страницах.

Если строка не помещается на 8К страницы, то SQL Server перемещает один или более столбцов переменной длины на страницы в ROW_OVERFLOW_DATA allocation unit, начиная со столбца с наибольшей шириной.

Heap vs Clustered table

Heap – Куча – таблица без кластерного индекса, физический порядок хранения данных в таблице (то есть порядок расположения данных на диске не задан).

IAM – index allocation map pages

Отдельные строки идентифицируются по ссылке на идентификатор строки (RID)

RID состоит из номера файла, номера страницы данных и слота на странице

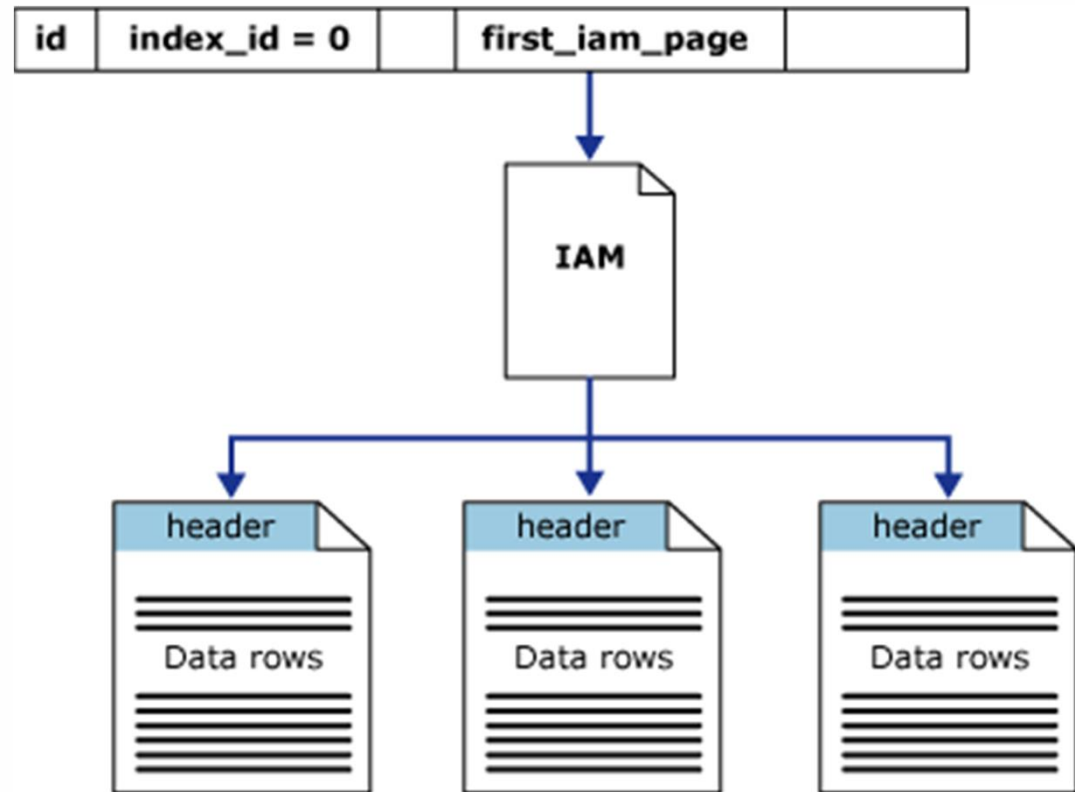


Table Scan



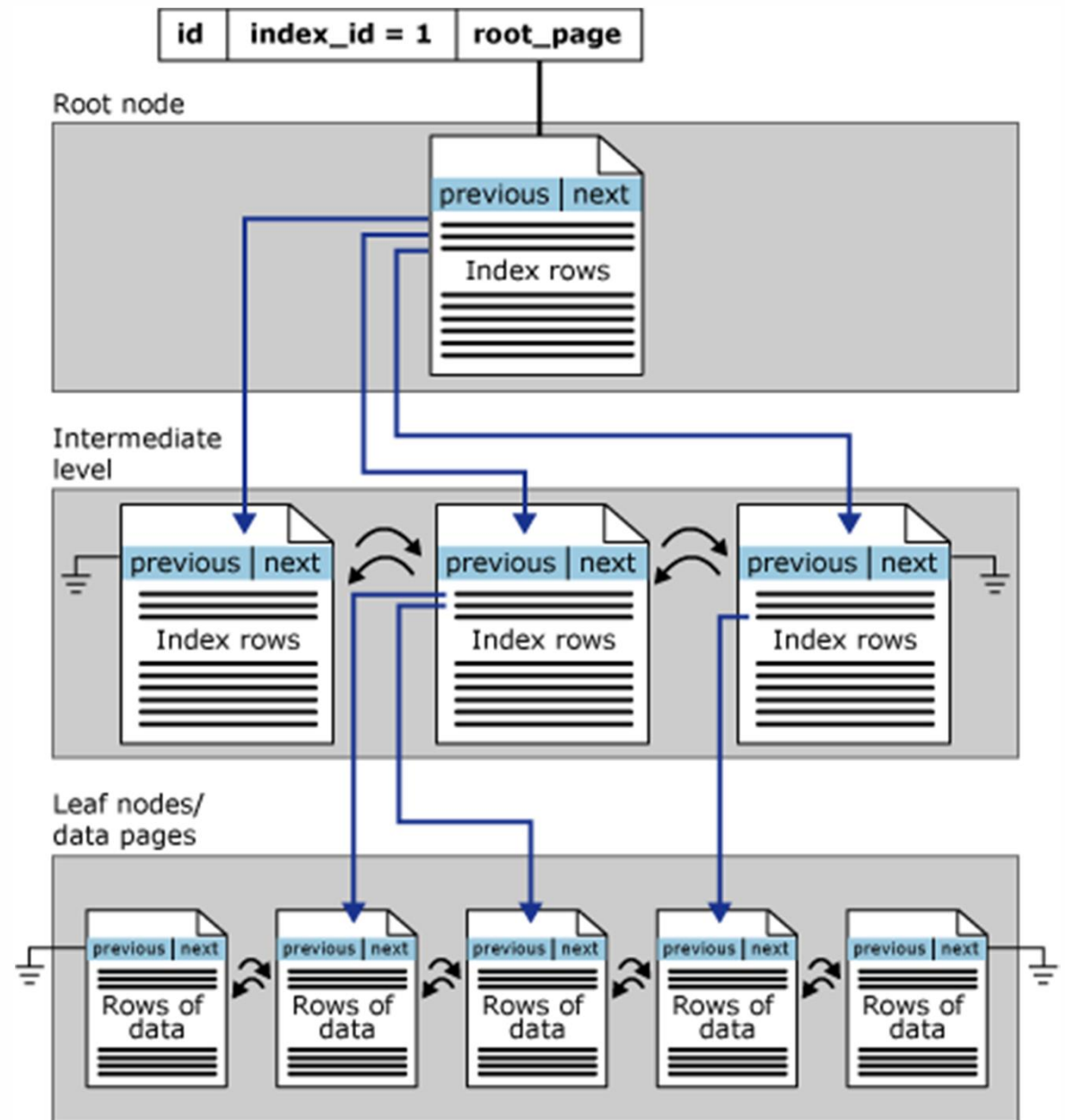
RID Lookup (Heap)

Heap vs Clustered table

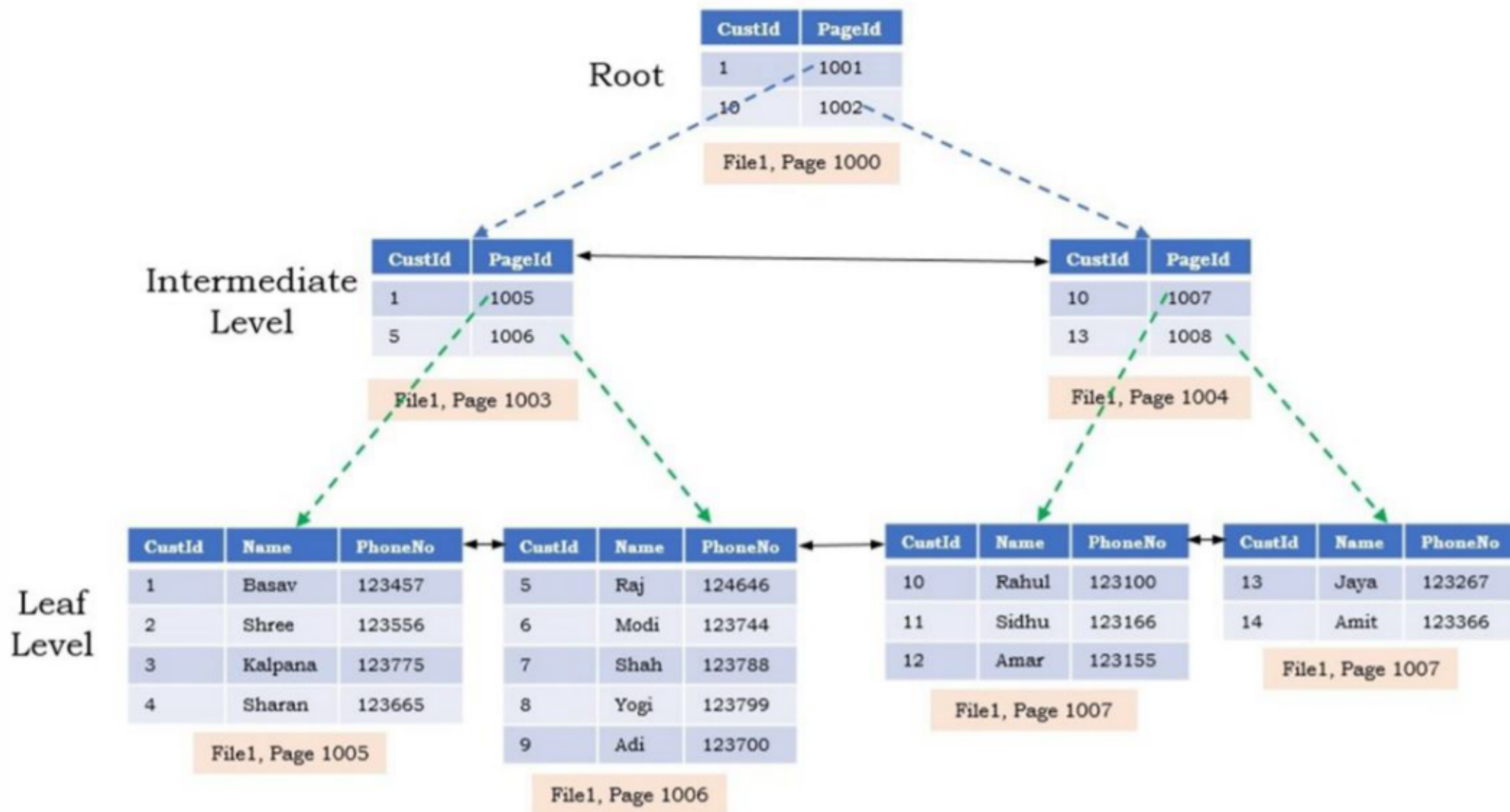
Clustered table – таблица с кластерным индексом.

Кластерный индекс – задает порядок расположения физических данных на диске.

Некластерные индексы содержат ссылку на кластерный индекс (само поле (поля) кластерного индекса).



B+ Tree Structure of a Clustered Index



Дерево - это граф, который характеризуется следующими свойствами:

1. Существует единственный элемент (узел или вершина), на который не ссылается никакой другой элемент - и который называется **КОРНЕМ**.
2. Начиная с корня и следуя по определенной цепочке элементов, можно осуществить доступ к любому элементу структуры.
3. На каждый элемент, кроме корня, имеется единственная ссылка, т.е. каждый элемент адресуется единственным указателем.

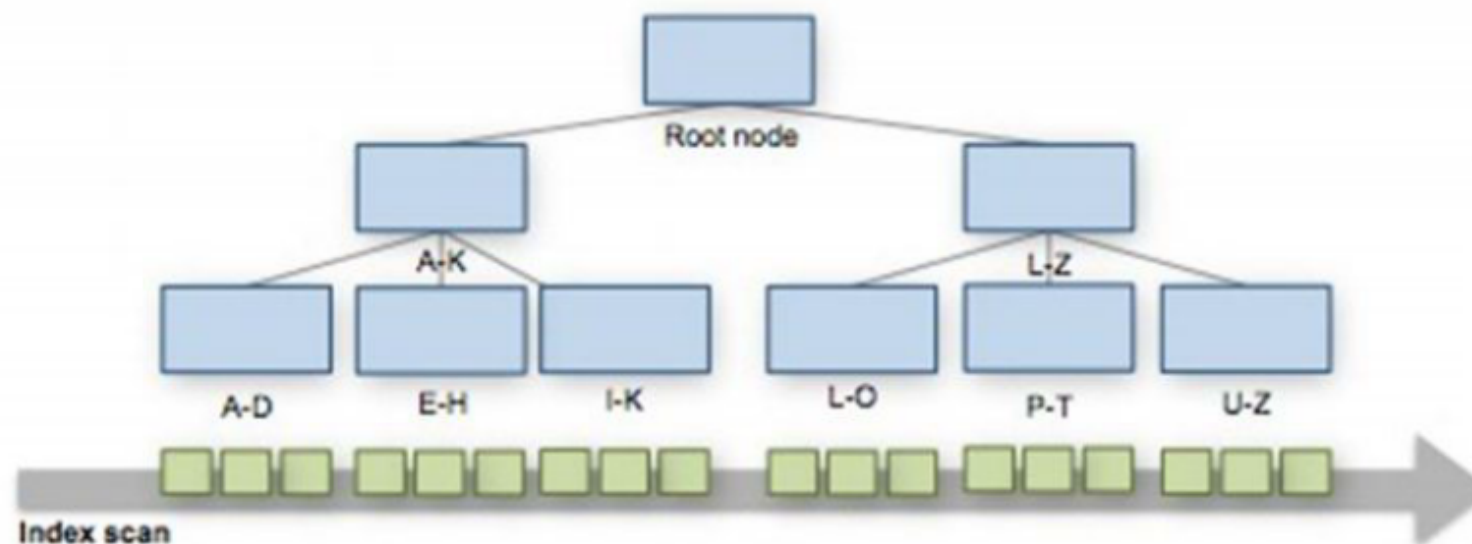
Дерево является СБАЛАНСИРОВАННЫМ тогда и только тогда, когда для каждого узла высота его двух поддеревьев различается не более чем на 1.

При такой структуре дерева время поиска элементов не превышает в среднем $\log N$


Источник: <http://khpi-iip.mipk.kharkiv.edu/library/datastr/book/prt06.html>

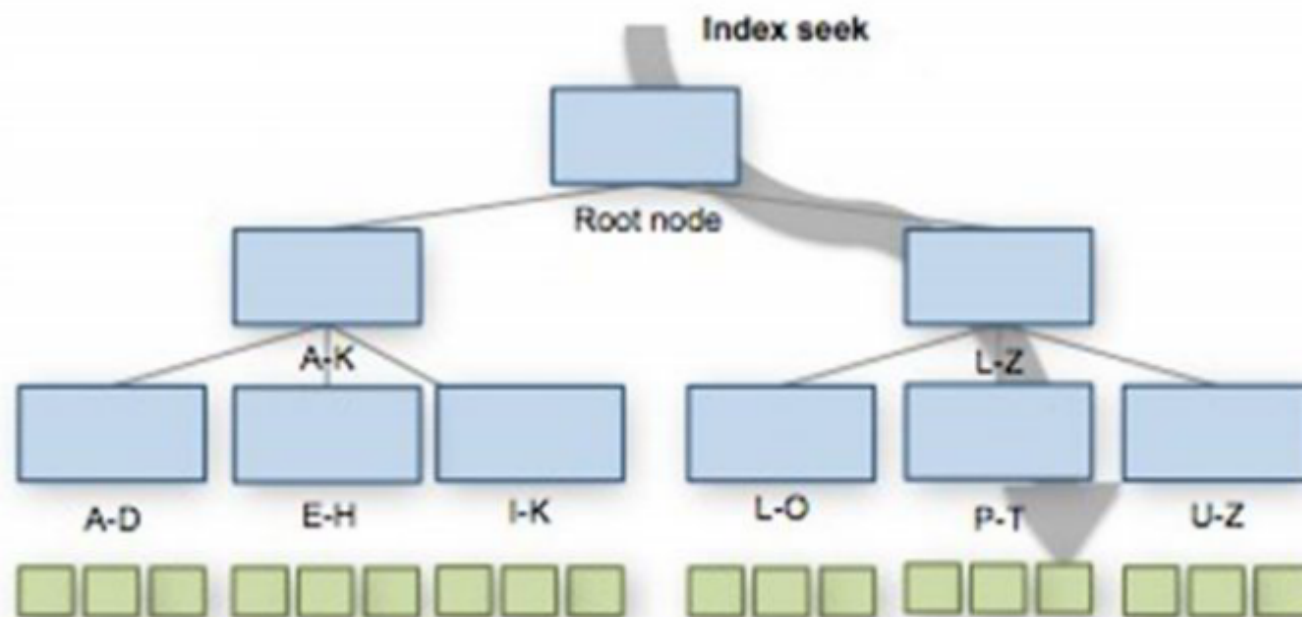
Index Scan vs Index Seek


O T U S




 Clustered Index Scan

 NonClustered Index Scan



 Clustered Index Seek

 NonClustered Index Seek

Вложенные циклы

Проходит по набору данных из таблицы A, по каждому значению в таблице A ищет соответствие в таблице B (если есть индекс, то использует его для поиска значения).

Так далее по следующему значению таблицы A.

Хорош для небольшого набора данных (одна из таблиц должна быть небольшой, она и будет выбрана для цикла)



Nested Loops

```
Для каждой строки [r] из [Ведущая таблица]
  Для каждой строки [s] из [Ведомая таблица]
    Если УдовлетворяетУсловию ([r],[s],[Условие соединения])
      Вывести ([r],[s]);
```

Используется, когда оба набора имеют индекс

Может использовать tempdb если в первом наборе много дубликатов

Лучший вариант для больших наборов данных



Merge Join

2 фазы:

Build – строится хэш таблица по наименьшей таблице



- Для каждого значения в таблице₁ считается хэш
- Сохраняется значение в хэш-таблицу, вычисленный хэш используется как ключ

Hash Match

Probe

- Для каждой строки из таблицы₂ считается значение хэш по полям, которые указаны в join (оператор =)
- Ищется хэш в хэш-таблице, проверяются значения полей

Если build таблица не помещается в память, она будет помещена на диск в tempdb.

CustomerId	CustomerName	Hash
1	Пупкин	
2	Иванов	
3	Петров	

InvoiceId	CustomerId	Total
1	2	500
2	3	1000
3	1	15000
4	2	100
5	3	1000
6	2	1020

<https://www.youtube.com/watch?v=uX6PmZhS2zU>

Build Table

CustomerId	CustomerName	Hash
1	Пупкин	050c5d21
2	Иванов	
3	Петров	

InvoiceId	CustomerId	Total
1	2	500
2	3	1000
3	1	15000
4	2	100
5	3	1000
6	2	1020

<https://www.youtube.com/watch?v=uX6PmZhS2zU>

Build Table

CustomerId	CustomerName	Hash
1	Пупкин	050c5d22
2	Иванов	051a5f21
3	Петров	052afd25

Probe Table

Hash	InvoiceId	CustomerId	Total
051a5f21	1	2	500
052afd25	2	3	1000
050c5d22	3	1	15000
	4	2	100
	5	3	1000
	6	2	1020

Result Table

CustomerId	CustomerName	InvoiceId	CustomerId	Total
2	Иванов	1	2	500
3	Петров	2	3	1000



Nested Loops

Nested loops

- Одна из таблиц небольшого размера



Merge Join

Merge Join

- Оба набора данных проиндексированы
- Хорош для больших наборов данных



Hash Match

Hash Match

- Неиндексированные наборы данных
- Обе таблицы большие
- Оператор соединения =
- Может использовать tempdb



- Сортируйте по полям с индексом, иначе это будет дорого
- Сортировка может пойти в tempdb, если будете выводить большой набор и не хватит памяти
- Иногда параллелизм ускоряет сортировку, но бывают случаи, когда без него быстрее



Sort



Sort

Set statistics io on

Set statistics time on

SQL Sentry Plan Explorer (+ анонимизация плана)

Средство для отображения статистики IO и CPU time

<https://statisticsparser.com/>

EXEC sp_helpindex 'Название таблицы'

Рефлексия

О чем мы говорили сегодня?

- Что такое план запроса?
- Что такое куча?
- Что такое кластерный индекс? Сколько их может быть на таблице?
- Чем estimated план отличается от actual?
- Что такое Hash Join? По каким таблицам строится хэш функция?
- Какие статистики вы помните?



Рефлексия

Напишите, пожалуйста, свое впечатление о вебинаре.

- Отметьте 3 пункта, которые вам запомнились с вебинара.
- Что вы будете применять в работе из сегодняшнего вебинара?



Заполните, пожалуйста,
опрос в ЛК о занятии



Спасибо
за внимание!

До встречи в **Slack** и на вебинаре

