ОНЛАЙН-ОБРАЗОВАНИЕ

OTUS

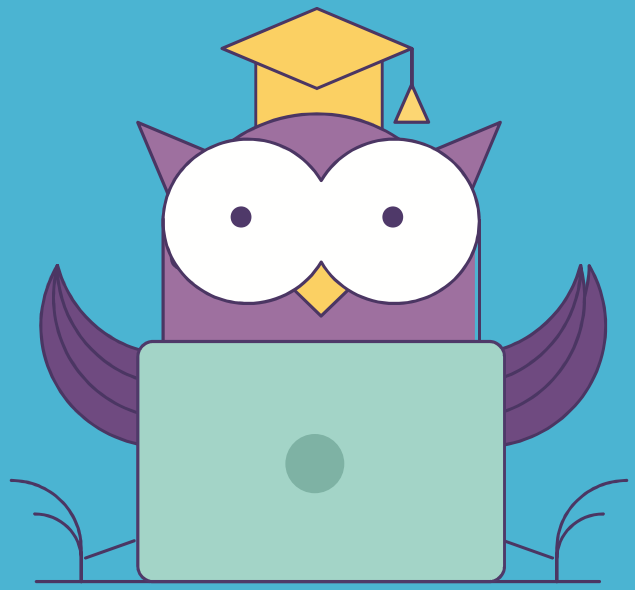
Итоговое занятие по SELECT

Курс «Разработчик MS SQL Server»

Занятие № 9

Кристина Кучерова

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 большие Select’ы
- Анализировать планы выполнения

Маршрут вебинара

Порядок выполнения

Операторы в плане

Анализ

Сложные SELECT

Выводы

Что было на прошлом занятии?

OTUS

Можно посчитать:

- Чем PIVOT отличается от UNPIVOT?
- Выполнится ли запрос?

```
SELECT * FROM
(
SELECT YEAR(ord.OrderDate) as SalesYear,
        ord.OrderId
 FROM Sales.Orders AS ord
) AS Sales
PIVOT (COUNT(OrderId)
FOR SalesYear IN (SELECT DISTINCT YEAR(ord.OrderDate)
 FROM Sales.Orders))
as PVT
```

- Выполнится ли запрос?

```
SELECT City.CityId, City.CityName, City.LatestRecordedPopulation, Acc.*
FROM Application.Cities AS City
INNER JOIN Application.DetermineCustomerAccess(City.CityId) AS Acc
ORDER BY City.CityId, City.CityName
```

Зачем читать план запроса?

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.

Осторожно!

Требуется выполнить запрос!

Особенности плана запроса

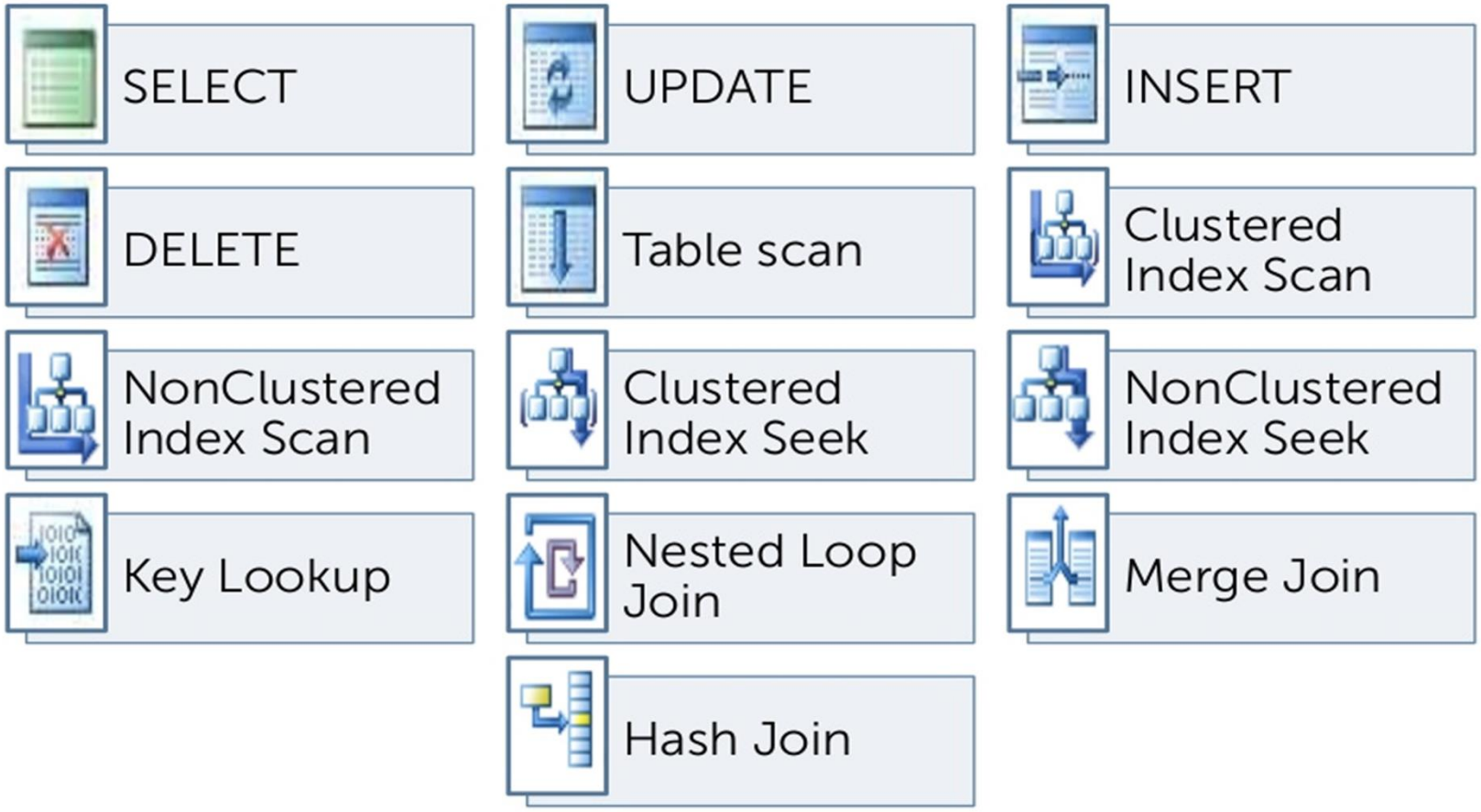
SQL Server:

- 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**

Операторы плана



Understanding SQL Server Query Execution Plans

Richard Douglas

,

Что все это значит?

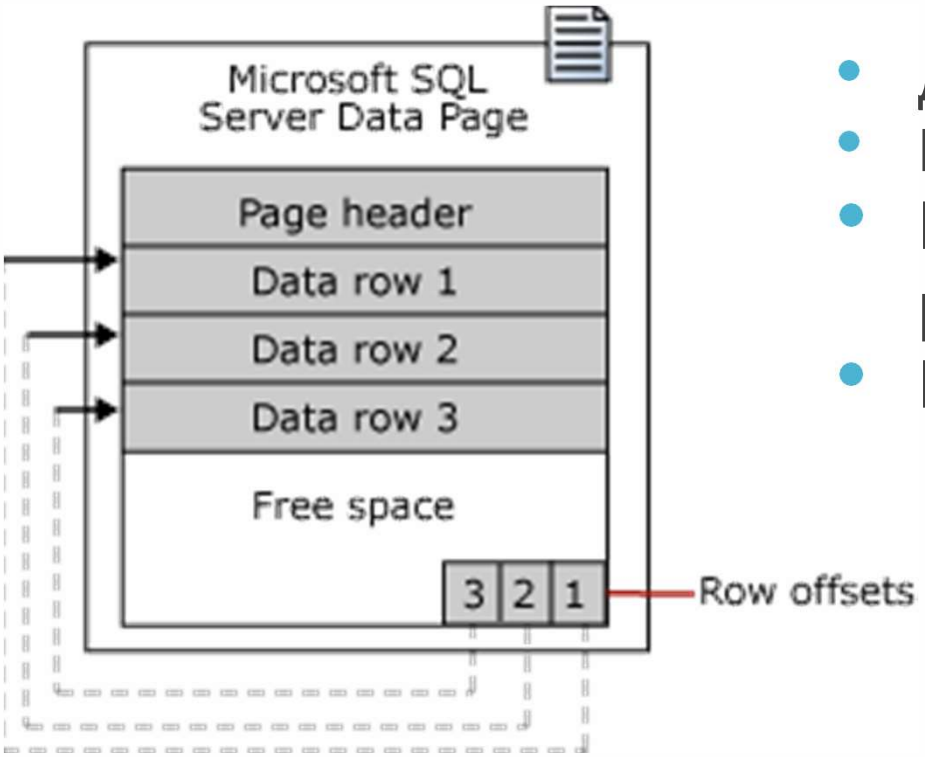
OTUS

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

Как данные хранятся в **SQL Server?**

OTUS

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

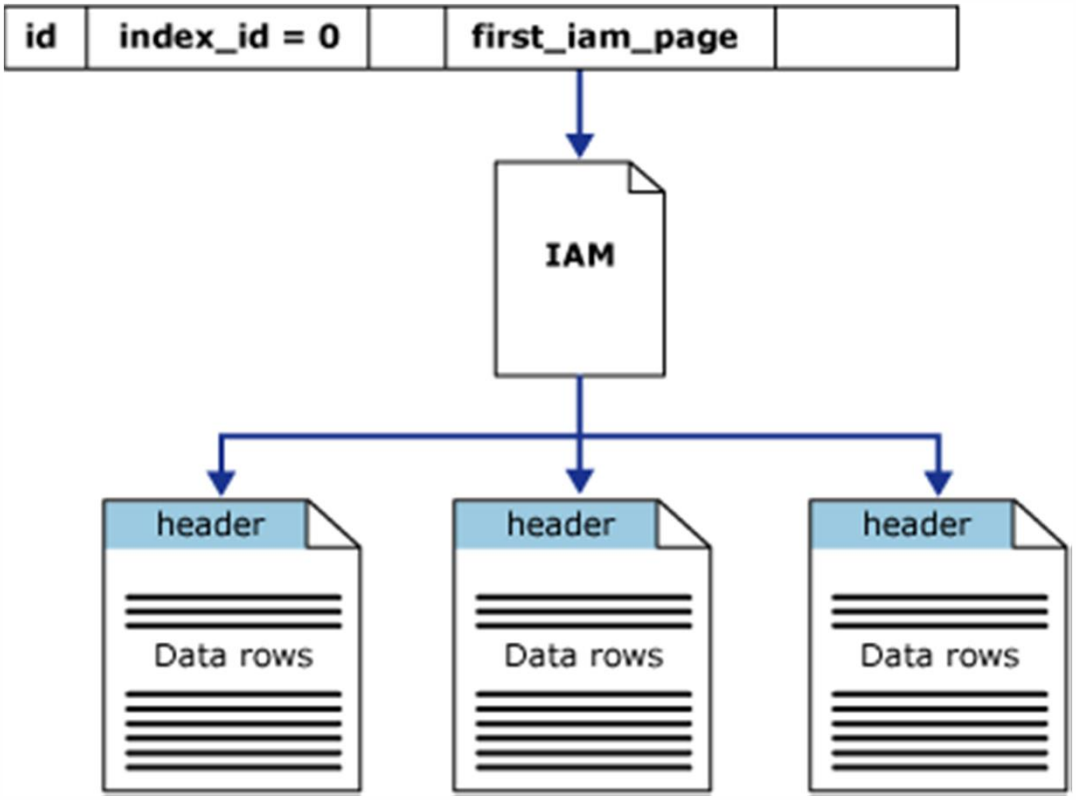


Table Scan

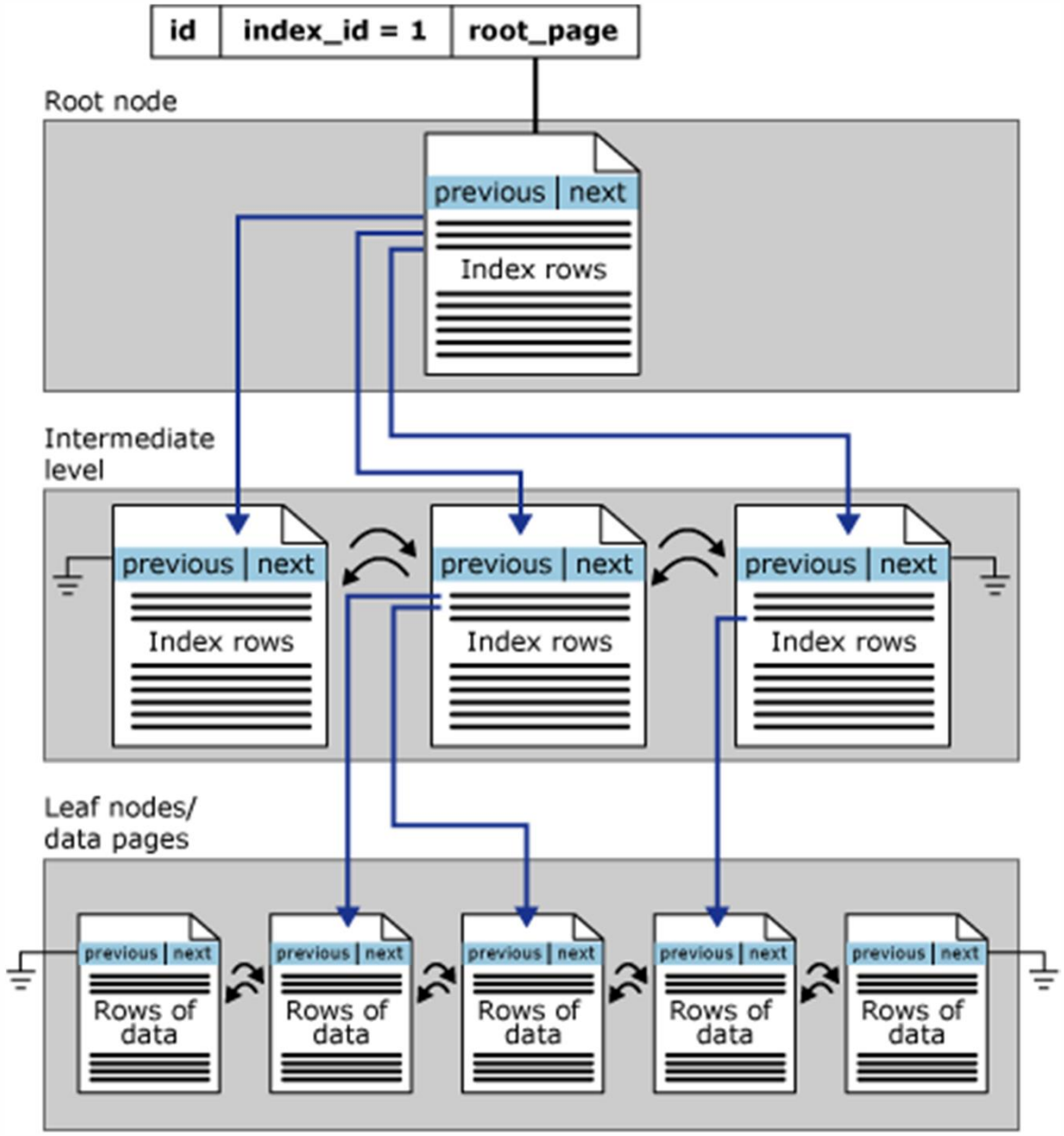
RID Lookup (Heap)

Heap vs Clustered table

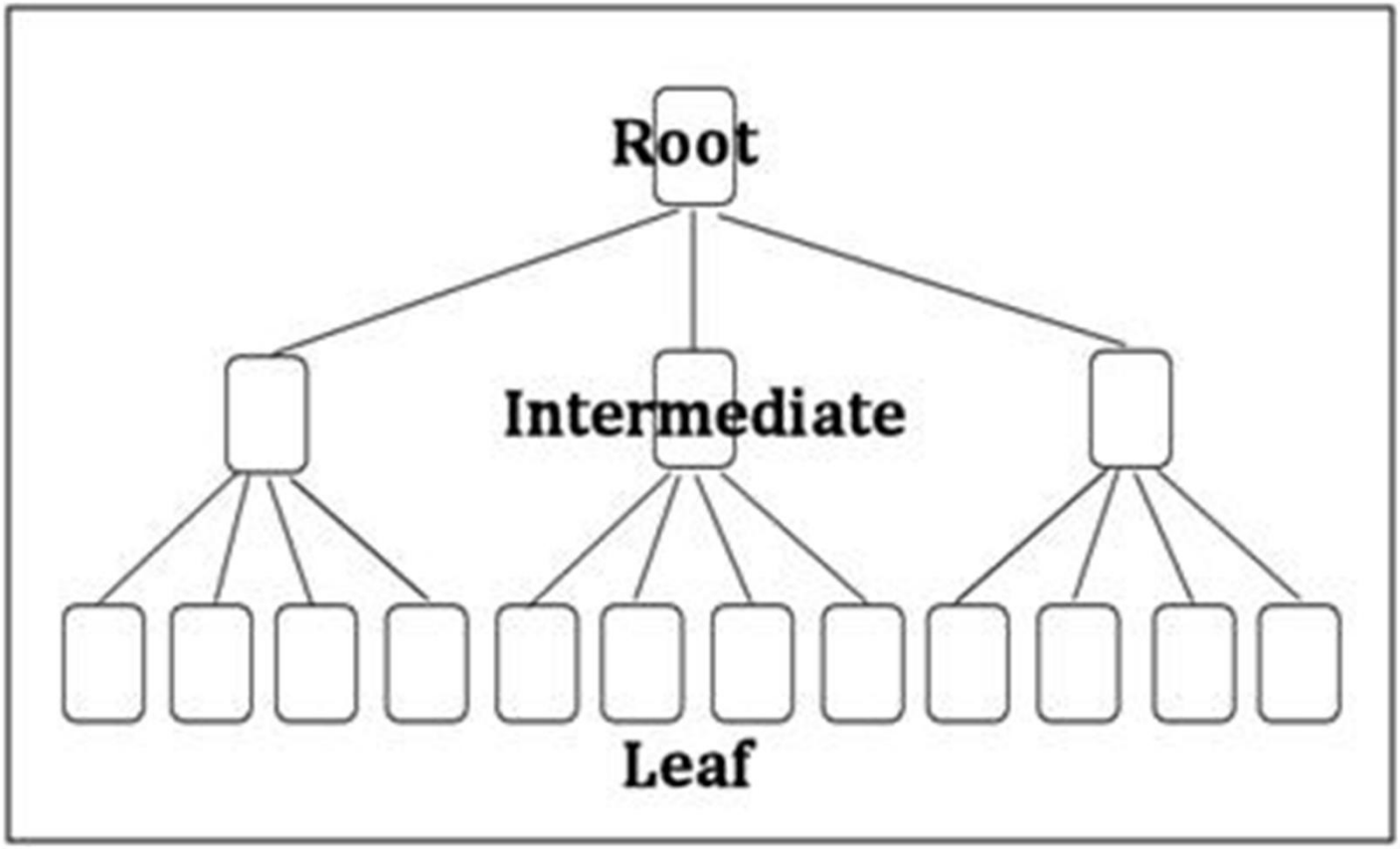
Clustered table – таблица с клатерным индексом.

Кластерный индекс – задает порядок расположения физических данных на диске.

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



B-tree – balanced-tree сбалансированное дерево



Теория графов или откуда взялись деревья

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.

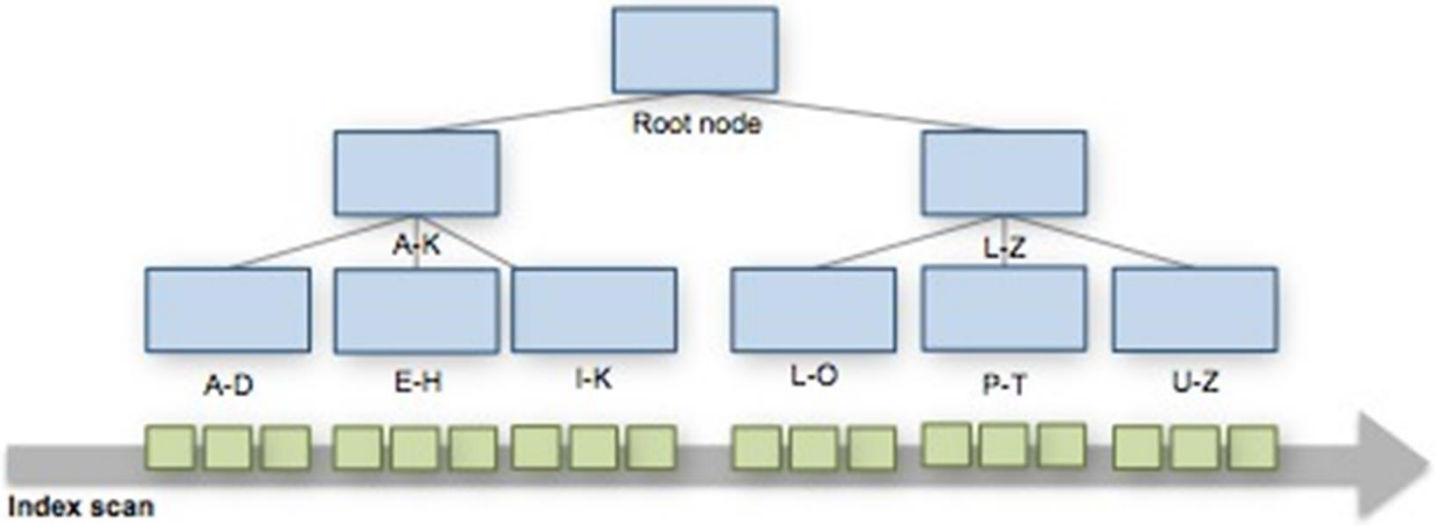
3. На каждый элемент, кроме корня, имеется единственная ссылка, т.е. каждый элемент адресуется единственным указателем.

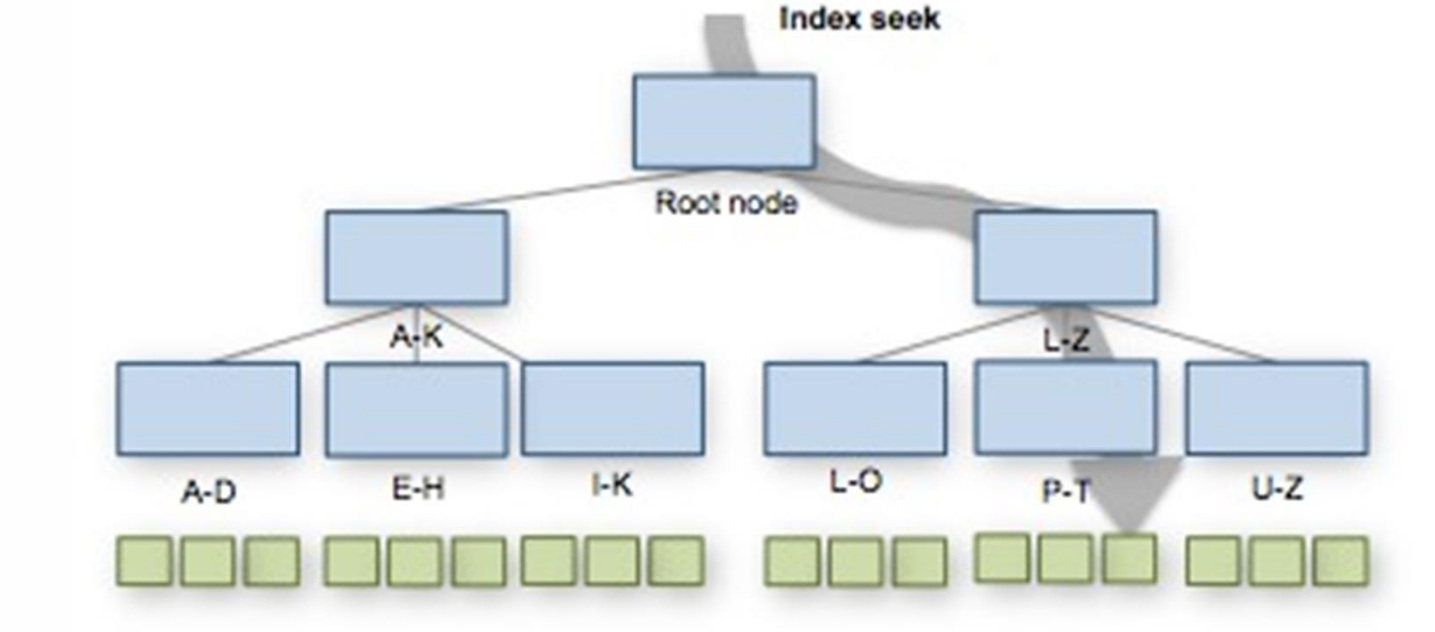
Дерево является СБАЛАНСИРОВАННЫМ тогда и только тогда, когда для каждого узла высота его двух поддеревьев различается не более чем на 1.

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





https://sqlsunday.com/2013/02/19/indexing-basics/

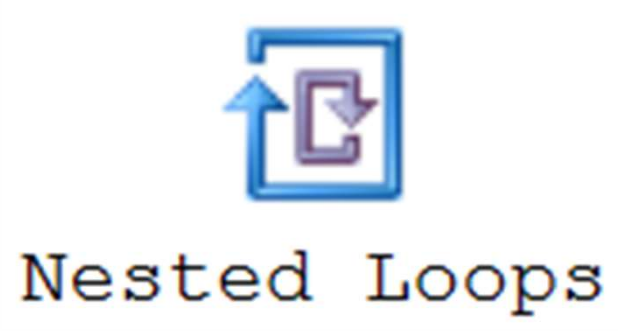
Nested loop

OTUS

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 есть 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)

Merge join

OTUS

Используется, когда оба набора имеют индекс

Может использовать tempdb если в первом наборе много дубликатов

Лучший вариант для больших наборов данных

Hash Join

2 фазы:
Build – строится хэш таблица по наименьшей таблице

Hash Match

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

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.

Hash Join

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

Hash Join

OTUS

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

Hash Join

OTUS

Build Table

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

Probe Table

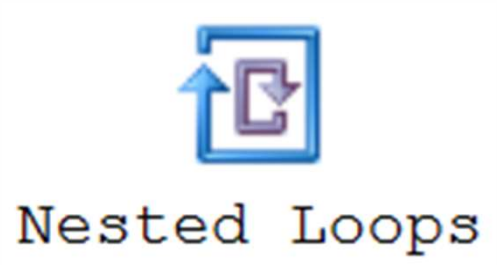
Hash	InvoiceId	CustomerId	Total
051a5f21	1	2	500
052afd25	2	3	1000
050c5d22	3	1	15000
	4	2	100
	5	3	1000
	6	2	1020

Result Table

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

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

Операторы физического соединения

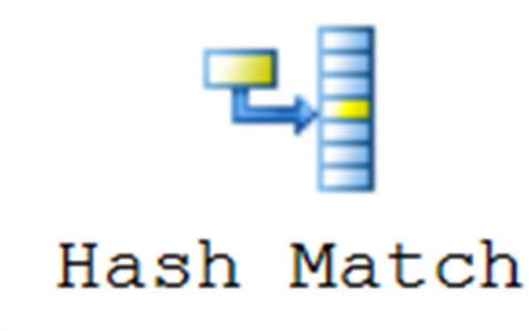


Nested loops

- Одна из таблиц небольшого размера

Merge Join

- Оба набора данных проиндексированы
- Хорош для больших наборов данных



Hash Match

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

Sort

- 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

Полезные инструменты

OTUS

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