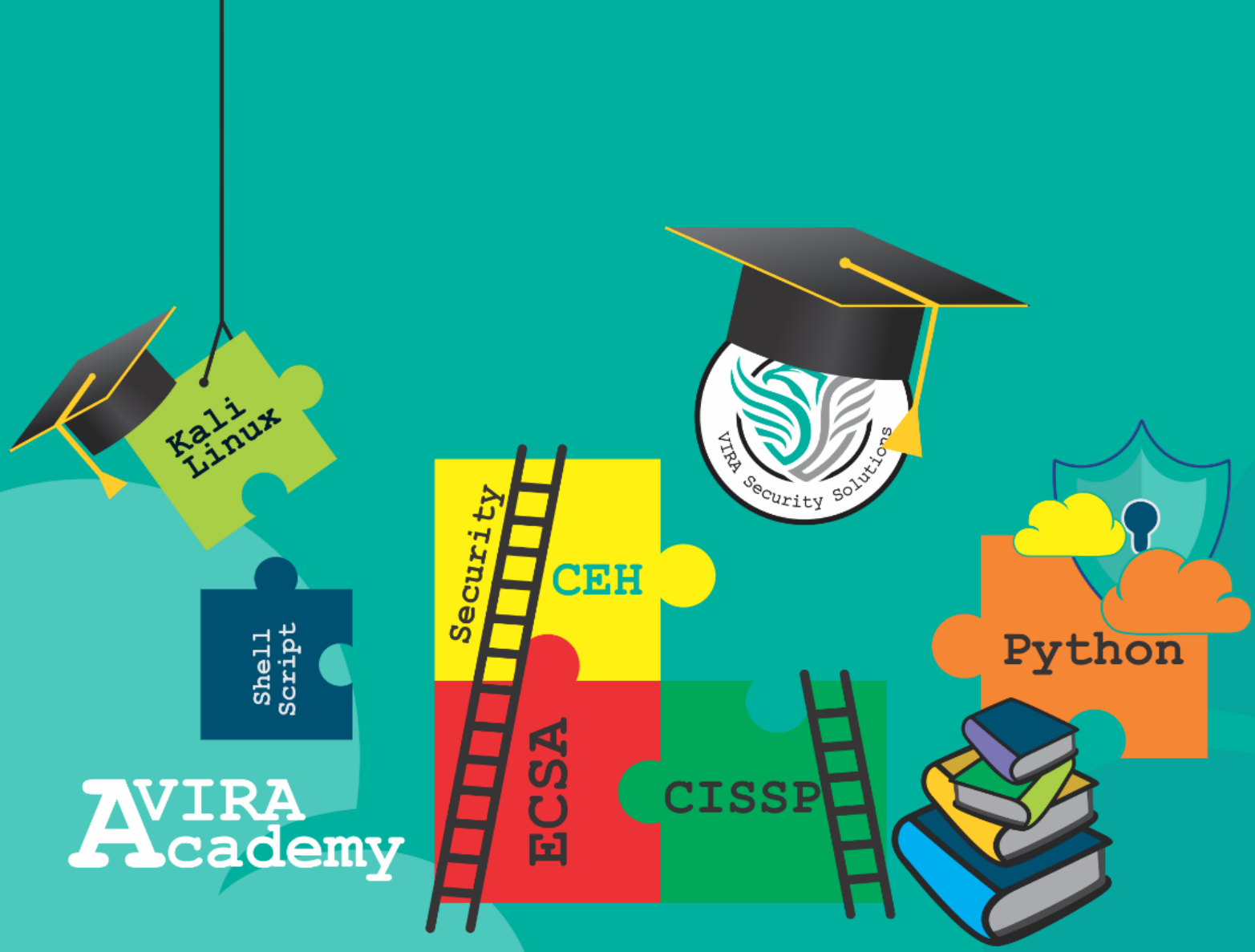

VIRA
Security
Solutions


راهکارهای امن ویرا
شماره ثبت: ۵۴۰۱۳۹
VIRA Security Solutions


Kali Linux
Shell Script
VIRA Academy
Security
CEH
ECSA
CISSP
VIRA Security Solutions
Python

DBA

SQL Server Administration

Part8

Mohammad Reza Gerami
mrgerami@aut.ac.ir
gerami@virasec.ir

Block and Locks

Block and Locks

در SQL Server برای حفظ یکپارچگی داده ها (data integrity)، برای خواندن ها و نوشتن ها از قفل هایی استفاده می کند، طوریکه **تنها یک فرایند در هر زمان** کنترل داده را در اختیار دارد.

وقتی قفل ها برای یک مدت طولانی حفظ می شود باعث مسدود شدن (blocking) می شوند که به این معناست که یک فرایند باید منتظر اتمام کار فرایند دیگر با داده شود و قبل از اینکه فرایند دیگر بتواند ادامه پیدا کند، قفل را آزاد کند. این مسئله مشابه deadlocking می باشد که در آن دو فرایند برای یک منبع در حال انتظار هستند، اما بر خلاف deadlocking، به محض اینکه فرایند اول منبع را رها می کند، blockingنیز آزاد می شود.

Block and Locks

blockingدر نتیجه انتظار دو فرآیند برای دسترسی به یک داده یا آبجکت می باشد، و لازم است فرایند دوم منتظر شود تا فرایند اول منبع را آزاد کند. چگونگی این کار در SQL Serverدر همه ی زمان ها رخ میدهد، اما معمولا شما blockingرا مشاهده نمی کنید، زیرا زمان ایجاد قفل خیلی کوتاه است.

قفل ها در هنگام آپدیت کردن داده ایجاد میشوند، اما این قفل ها در هنگام خواندن داده نیز استفاده می شوند. وقتی داده آپدیت می شود، یک قفل Update استفاده می شود و وقتی که داده خوانده می شود یک قفل Shared مورد استفاده قرار می گیرد.

یک قفل Update یک قفل انحصاری برای این فرایند روی داده ایجاد می کند و قفل Shared به دیگر فرایندها اجازه می دهد تا برای دسترسی به داده از یک قفل Shared استفاده کنند و زمانی که فرایندها سعی در دسترسی به یک داده ی مشابه دارند locking و blocking اتفاق می افتد.

دستورات مفید

sp_who2

	SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramNam
41	59	sleeping	ED...	EDGEW...	.	test	AWAITING COMMAND	0	19	12/28 11:06:49	Microsoft S.
42	60	SUSPENDED	ED...	EDGEW...	59	test	UPDATE	0	4	12/28 11:06:56	Microsoft S.

نشان می دهد SPID 60 توسط SPID 50 مسدود شده است

در استادیو در بخش Activity Monitor گزینه Processes را انتخاب و جزییات بیشتر را مشاهده فرمائید

Report - All Blocking Transactions

روش دیگر استفاده از گزارش های داخلی SSMS می باشد.
روی نام SQL Serverراست کلیک کرده و
Reports > Standard Reports > Activity - All Block Transactions
را انتخاب کنید.

All Blocking Transactions

The description of transactions which are blocking other transactions.

Transaction ID	# Directly Blocked Transactions	# Indirectly Blocke Transactions	State	Transaction Type	Start Time	Resource Type	Session ID	Blocking SQL Statement
⊟ 789800	1	0	Active	Full Transaction	12/28/2011 11:06:49 AM	RID	59	⊞ --

Direct/Indirect	Blocked Transaction ID	Blocked Transacti	State	Transaction Type	Start Time	Resource Type	Session ID	Blocked SQL Statement
Direct	789800	user_transaction	Active	Full Transaction	12/28/2011 11:07:00 AM	RID	60	⊞ UPDATE [dbo].[Categories] set [Description] = @1 WHERE [CategoryName]=@2

Block and Locks

Querying Dynamic Management Views

دستورات مفید

از دستورات DMVs نیز برای گرفتن اطلاعات در مورد blocking استفاده کنید.

```
SELECT session_id, command, blocking_session_id, wait_type, wait_time, wait_resource, t.TEXT
FROM sys.dm_exec_requests
CROSS apply sys.dm_exec_sql_text(sql_handle) AS t
WHERE session_id > 50
AND blocking_session_id > 0
UNION
SELECT session_id, '', '', '', '', t.TEXT
FROM sys.dm_exec_connections
CROSS apply sys.dm_exec_sql_text(most_recent_sql_handle) AS t
WHERE session_id IN (SELECT blocking_session_id
                     FROM sys.dm_exec_requests
                     WHERE blocking_session_id > 0)
```

	session_id	command	blocking_session_id	wait_type	wait_time	wait_resource	TEXT
1	59		0		0		BEGIN TRAN INSERT_FOO INSERT INTO dbo.Categ...
2	60	UPDATE	59	LCK_M_U	1157132	RID: 5:1:1217:0	(@1 varchar(8000),@2 varchar(8000))UPDATE [dbo].[Cat...

Block and Locks

منبع قابل قفل شدن برای SQL Server وجود دارد و آن‌ها سلسله مراتبی را تشکیل می‌دهند.

Database کل بانک اطلاعاتی قفل شده است، معمولاً طی تغییرات Schemaبانک اطلاعاتی روی می‌دهد.

Table کل جدول قفل شده است، شامل همه اشیای مرتبط با جدول.

Extent کل Extent متشکل از هشت Pageقفل شده است.

Page همه داده‌ها یا کلیدهای Index در آن Page قفل شده اند.

Key قفلی در کلید مشخصی یا مجموعه کلید هایی Index وجود دارد. ممکن است سایر کلیدها در همان Index Page تحت تاثیر قرار نگیرند.

Row or Row Identifier -RID هر چند قفل از لحاظ فنی در Row Identifier قرار می‌گیرد ولی اساساً کل ردیف را قفل می‌کند.

Shared Locks

زمانی استفاده می‌شود، که فقط باید داده‌ها را بخوانید، یعنی هیچ تغییری ایجاد نخواهید کرد. Shared Lock با سایر Shared Lockهای دیگر سازگار است، البته قفل‌های دیگری هستند که با Shared Lock سازگار نیستند.
یکی از کارهایی که Shared Lock انجام می‌دهد، ممانعت از انجام Dirty Read از طرف کاربران است.
به این معنا که اگر در طی یک تراکنش مشغول به تغییر اطلاعاتی باشیم، سایر کاربران از خواندن نتیجه آن (اصطلاحا به آن Dirty readگفته می‌شود) منع خواهند شد؛ تا زمانیکه این تراکنش با موفقیت به پایان برسد. هرچند در این حالت سایر تراکنش‌ها امکان ویرایش یا حذف اطلاعات را خواهند داشت.

Exclusive Locks

این قفل‌ها با هیچ قفل دیگری سازگار نیستند. اگر قفل دیگری وجود داشته باشد، نمی‌توان به Exclusive Lock دست یافت و همچنین در حالی که Exclusive Lock فعال باشد، به هر قفل جدیدی از هر شکل اجازه ایجاد شدن در منبع را نمی‌دهند.
این قفل از اینکه دو نفر همزمان به حذف کردن، بروز رسانی و یا هر کار دیگری مبادرت ورزند، پیشگیری می‌کند.

Block and Locks

Update Locks

این قفل ها نوعی پیوند میان Shared Locks و Exclusive Locks هستند.

برای انجام Update باید بخش Whereرا (در صورت وجود) تایید اعتبار کنید، تا دریابید فقط چه ردیف هایی را می‌خواهید بروز رسانی کنید. این بدان معنی است که فقط به Shared Lock نیاز دارید، تا زمانی که واقعاً بروز رسانی را انجام دهید. در زمان بروز سازی نیاز به Exclusive Lock دارید.

Update Lock نشان دهنده این واقعیت است که دو مرحله مجزا در بروز رسانی وجود دارد، Shared Lockای دارید که در حال تبدیل شدن به Exclusive Lockاست. Update Lockتمامی Update Lockهای دیگر را از تولید شدن باز می‌دارند، و همچنین فقط با Shared Lockو Intent Shared Lockها سازگار هستند.

Intent Locks

با سلسله مراتب شی سر و کار دارد. بدون Intent Lock، اشیای سطح بالاتر نمی‌دانند چه قفلی را در سطح پایین‌تر داشته اید. این قفل‌ها کارایی را افزایش می‌دهند و ۳ نوع هستند:

Intent Shared Lock

Shared Lock در نقطه پایین‌تری در سلسله مراتب، تولید شده یا در شرف تولید است. این نوع قفل تنها به Table و Page اعمال می‌شود.

Intent Exclusive Lock

همانند Intent Shared Lock است اما در شرف قرار گرفتن در آیتم سطح پایین‌تر است.

Shared With Intent Exclusive

Shared Lock در پایین سلسله مراتب شی تولید شده یا در شرف تولید است اما Intent Lock قصد اصلاح داده‌ها را دارد بنابراین در نقطه مشخصی تبدیل به Intent Exclusive Lock می‌شود

Schema Locks

به دو شکل هستند:

Schema Modification Lock

تغییر Schemaبه شی اعمال شده است. هیچ پرس و جویی یا سایر عبارت‌های Create، Alterو Dropنمی‌توانند در مورد این شی در مدت قفل Sch-M اجرا شوند. با همه حالات قفل ناسازگار است.

Schema Stability Lock

بسیار شبیه به Shared Lock است، هدف اصلی این قفل پیشگیری از Sch-M است وقتی که قبلاً قفل هایی برای سایر پرس و جو–ها (یا عبارت‌های Create، Alterو Drop) در شی فعال شده اند. این قفل با تمامی انواع دیگر قفل سازگار است به جز با Sch-M.

Bulk Update Locks

این قفل‌ها بارگذاری موازی داده‌ها را امکان پذیر می‌کنند، یعنی جدول در مورد هر فعالیت نرمال (عبارات T-SQL) قفل می‌شود، اما چندین عمل bcp یا Bulk Insert را می‌توان در همان زمان انجام داد. این قفل فقط با Sch-Sو سایر قفل های BUسازگار است.

Clustered Index

Clustered Index

Clustered Indexبه چه معناست؟

یکی از ساختارهای ذخیره سازی داده در جداول می باشد که بر اساس آن داده ها در ازای یک فیلد خاص که توسط ما Clustered Index مشخص می شود دارای نظم و ترتیب می باشد.
چینش فیزیکی رکوردها در این نوع از جداول بر اساس کلید ایندکسی است که مشخص شده است

نحوه ایجاد :Clustered Index

زمانیکه SQL SERVER قصد تبدیل یک Heap به یک Clustered Indexرا دارد مراحل زیر را انجام می دهد:

1. ایجاد یک کپی از جدول
2. مرتب سازی رکوردها بر اساس کلید ایندکس
3. ایجاد ساختار ایندکس و به عبارت دیگر ایجاد Root Levelو Intermediate Level
4. جایگزینی جدول Clustered با جدول قدیم

Clustered Index

ویژگی های Clustered Index:

هر Clustered Index دارای خصوصیات زیر می باشد:

1.تغییر ترتیب داده ها به اینصورت که با تغییر یک جدول به Clustered Index چینش فیزیکی داده ها بر اساس کلید ایندکس خواهد بود.

2.به ازای هر جدول صرفا یک Clustered Index وجود خواهد داشت.

3.ایجاد ساختار ایندکس و به عبارت دیگر ایجاد Root Level وIntermediate Level

4.جایگزینی جدول Clustered با جدول قدیم

توجه : کلید Clustered Index نباید یک فیلد بزرگ انتخاب شود زیرا از این کلید در NonClustered Index ها استفاده شده و این باعث اشغال حجم زیادی از حافظه و اتلاف آن می شود.

توجه: در صورتیکه طول کلیدهای Clustered Index از ۹۰۰ بایت بیشتر شود خود SQL SERVER مانع ساخت ایندکس روی جدول می گردد

Clustered Index

بررسی ساختار Clustered Index

در یک Clustered Index گره ریشه در sys.system_internals_allocation_units قرار می گیرد. برای یافتن داده از گره ریشه به سمت پایین حرکت کرده تا در Leaf Level به داده مورد نظر دسترسی پیدا نماید.

```
SELECT
   INDEXPROPERTY(OBJECT_ID('ClusteredTable'),'Clustered_IX','IsClustered')AS [Is Clustered],
   INDEXPROPERTY(OBJECT_ID('ClusteredTable'),'Clustered_IX','IndexDepth') AS [Index Depth]
GO
```

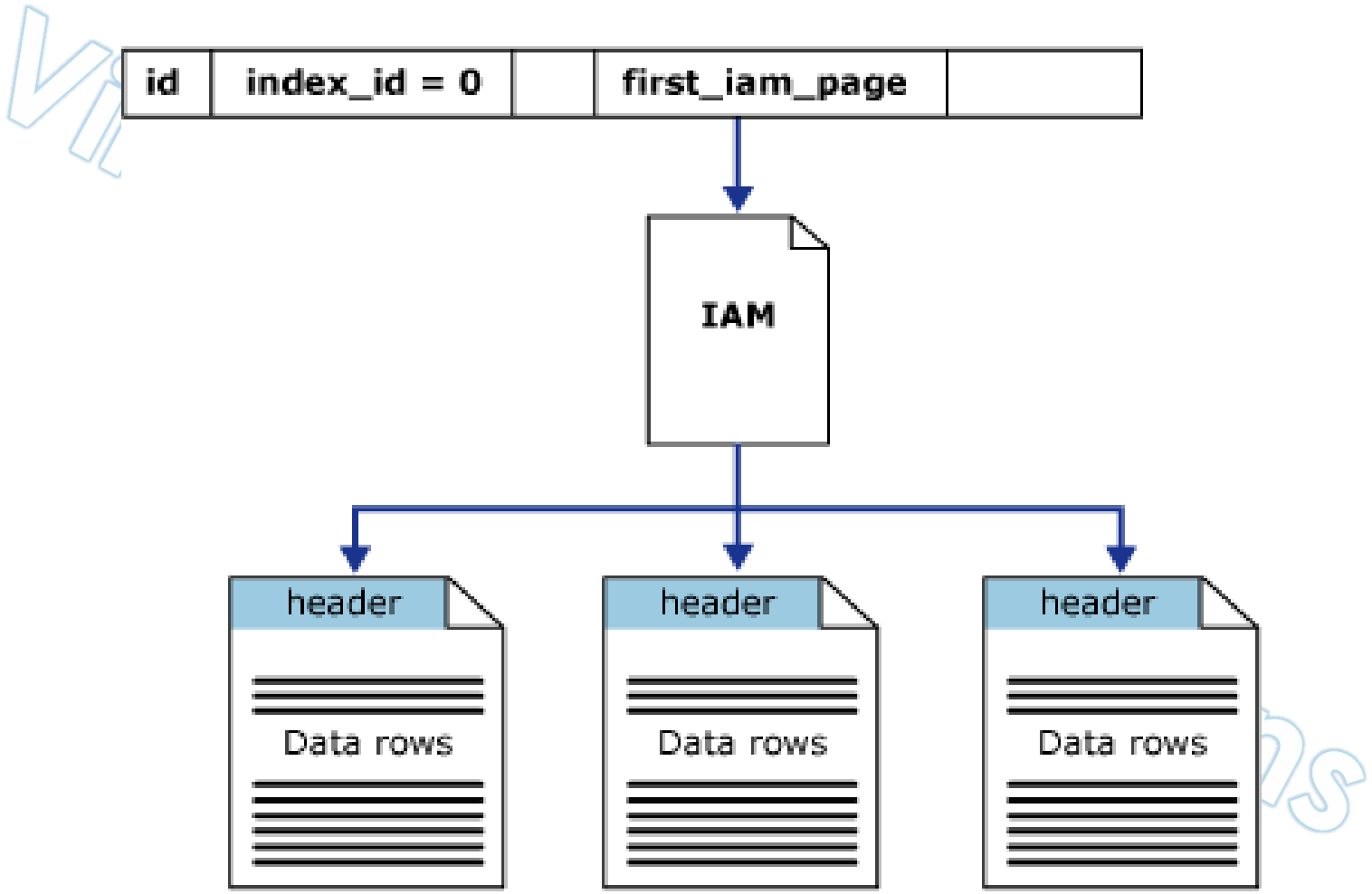
Heap

Heap tables are tables without a Clustered Index. A table in SQL Server can have a Clustered Index, then it's called a **Clustered Table**, and without a Clustered Index, it's called a **Heap Table**.

SQL Server HEAP

- Data is not stored in any particular order
- Specific data can not be retrieved quickly, unless there are also non-clustered indexes
- Data pages are not linked, so sequential access needs to refer back to the index allocation map (IAM) pages
- Since there is no clustered index, additional time is not needed to maintain the index
- Since there is no clustered index, there is not the need for additional space to store the clustered index tree
- These tables have a **index_id value of 0 in the sys.indexes catalog view**

Heap Table

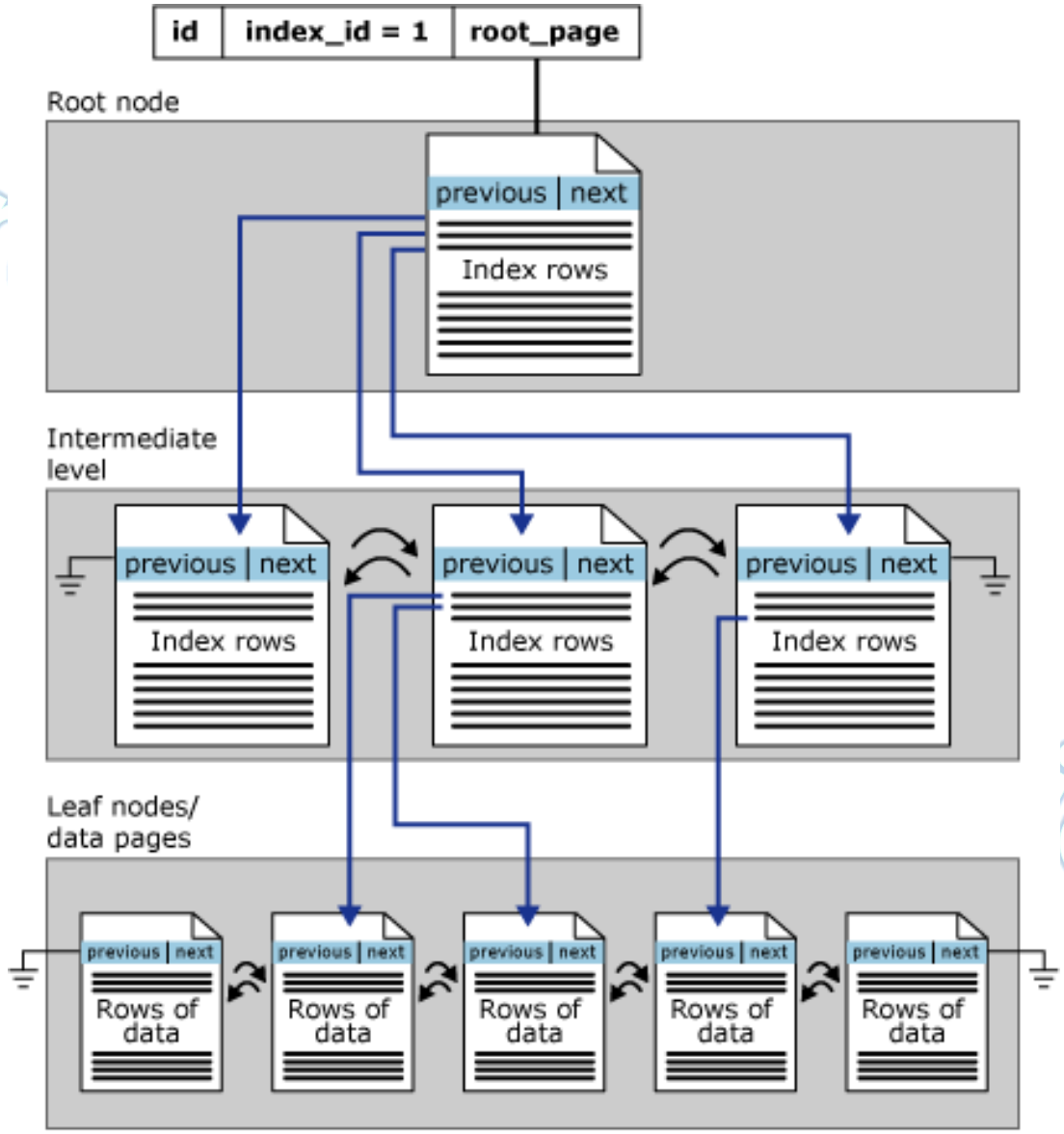

id
index_id = 0
first_iam_page
IAM
header
Data rows
header
Data rows
header
Data rows

Clustered Index

SQL Server Clustered Indexed Table

- Data is stored based on the clustered index key
- Data can be retrieved quickly based on the clustered index key, if the query uses the indexed columns
- Data pages are linked for faster sequential access
- Additional time is needed to maintain the clustered index based on INSERT, UPDATE and DELETE activity
- Additional space is needed to store the clustered index tree
- These tables have a **index_id value of 1 in the sys.indexes catalog view**

Clustered Index



NonClustered Indexچیست؟

NonClustered Indexیکی از ساختارهای ذخیره سازی داده در جداول می باشد که بر اساس آن داده ها در ازای یک فیلد خاص که توسط ما مشخص می شود دارای نظم و ترتیب بوده و فضای ذخیره سازی داده های شرکت کننده در ایندکس در مکانی مجزا است.

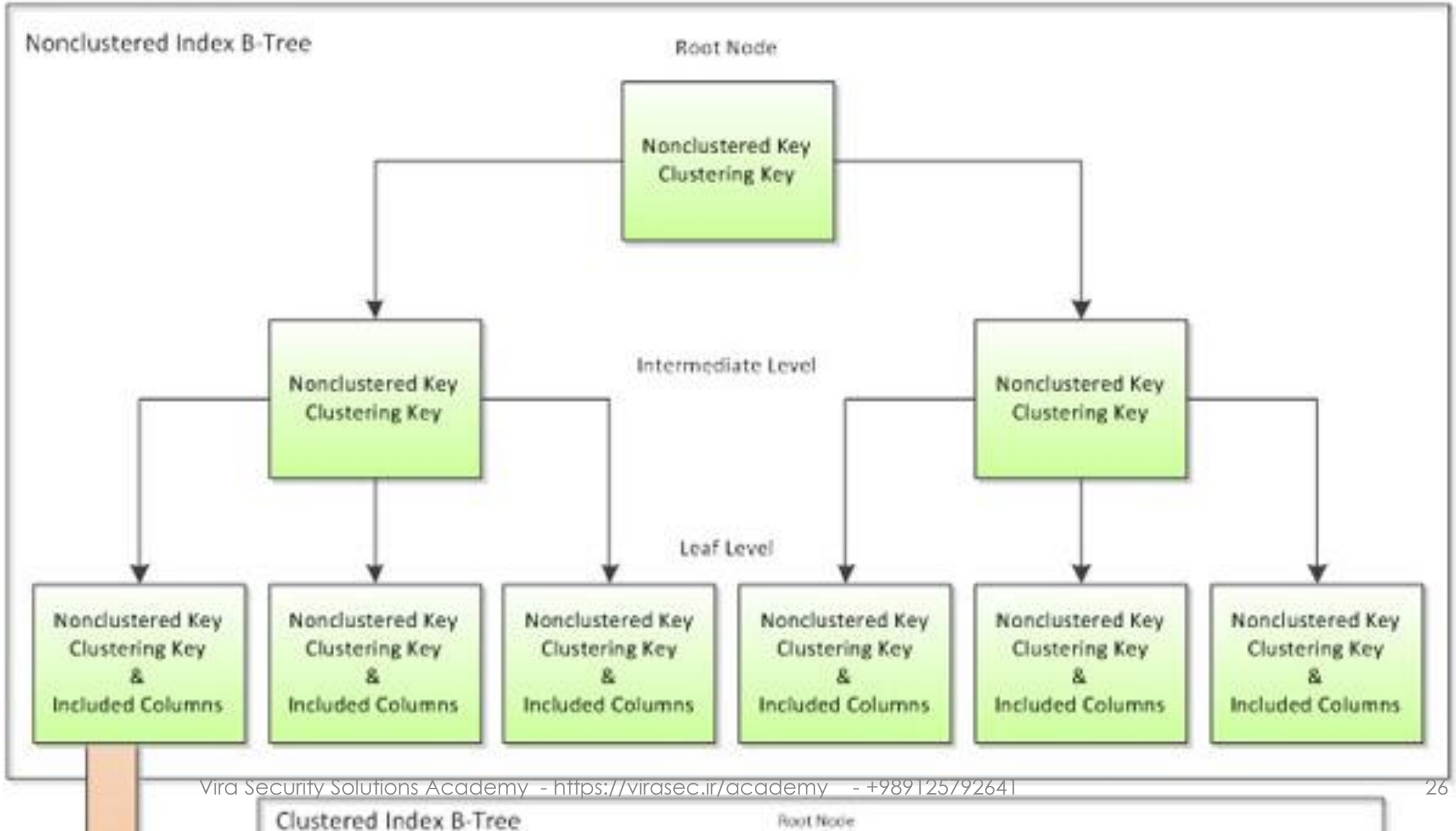
موارد زیر را میتوان از ویژگیهای آن دانست:

Pageهای مربوط به این نوع ایندکس از نوع Index Pageمی باشد.

ساختار ذخیره سازی NonClustered Index از نوع B-Treeمی باشد.

تعداد NonClustered Indexکه می توان برای یک جدول تعریف نمود ۹۹۹ عدد می باشد

NonClustered Index



NonClustered Index

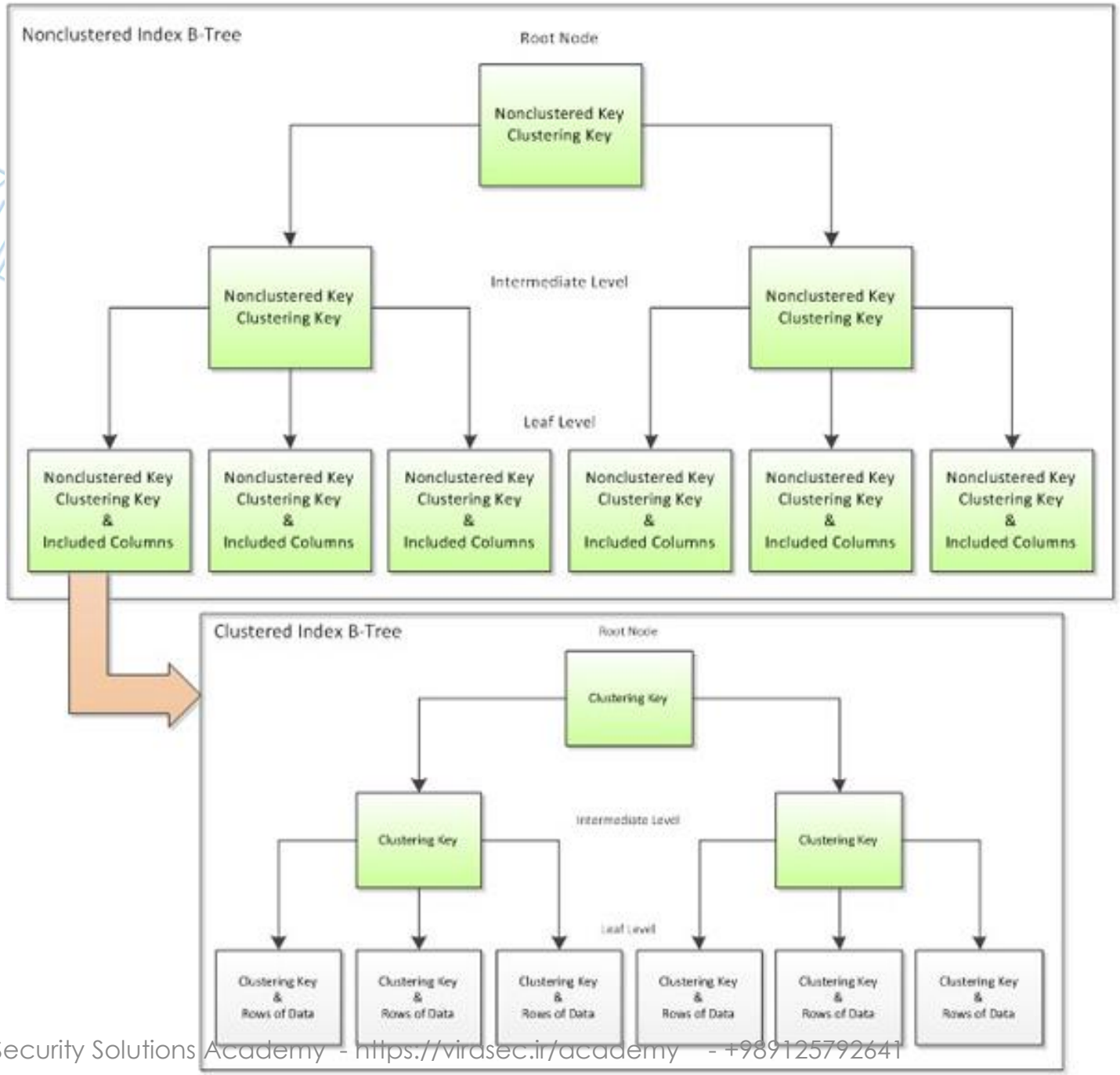
همانطور که در شکل مشاهده می شود یک NonClutered Index شامل لایه های Root,Intermediate و Leaf می باشد که جدا از داده ها در فایل ایندکس نگهداری می شوند. در لایه آخر یعنی Leaf Level هر خانه به یک رکورد در محیط فیزیکی داده ها اشاره دارد، به این صورت که قسمت اول آن به شماره Page و قسمت دوم به شماره رکورد موجود در آن Page اشاره می نماید

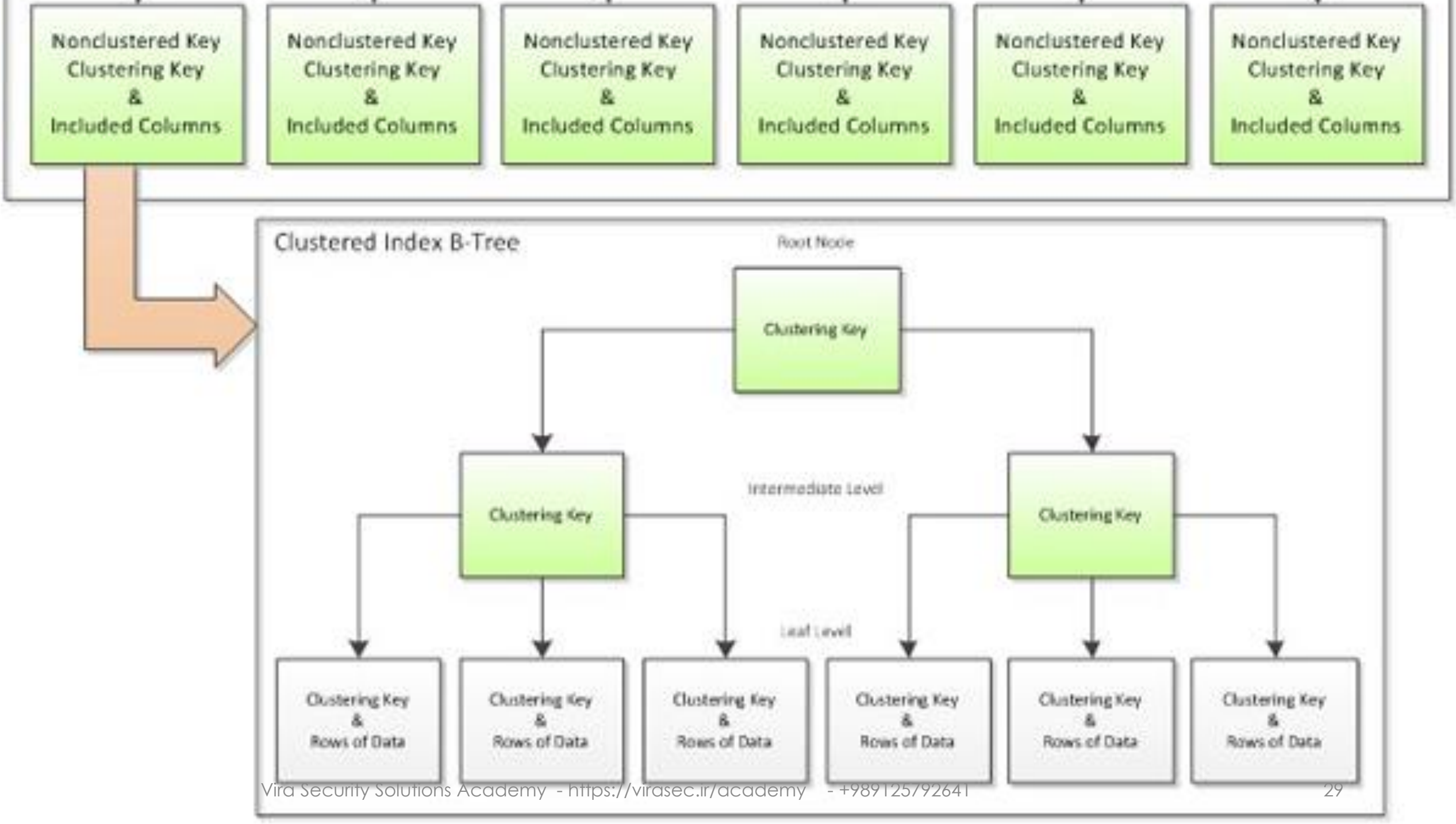
بررسی ساختار :NonClutered Index

NonClutered Indexها به دو دسته تقسیم می شوند:

یک دسته آنهایی هستند که روی یک Clustered Index تعریف می شوند که در این نوع از ایندکس بعد از اینکه از ریشه تا برگ در درخت NonClutered Index پیمایش شد در صورتیکه داده های مورد جستجو به همراه کلید ایندکس موجود نبود،برای یافتن داده به سراغ درخت Clutered Index رفته و آن را نیز تا رسیدن به داده اصلی پیمایش می نماید.شکل زیر ساختار این نوع ایندکس را نمایش می دهد.

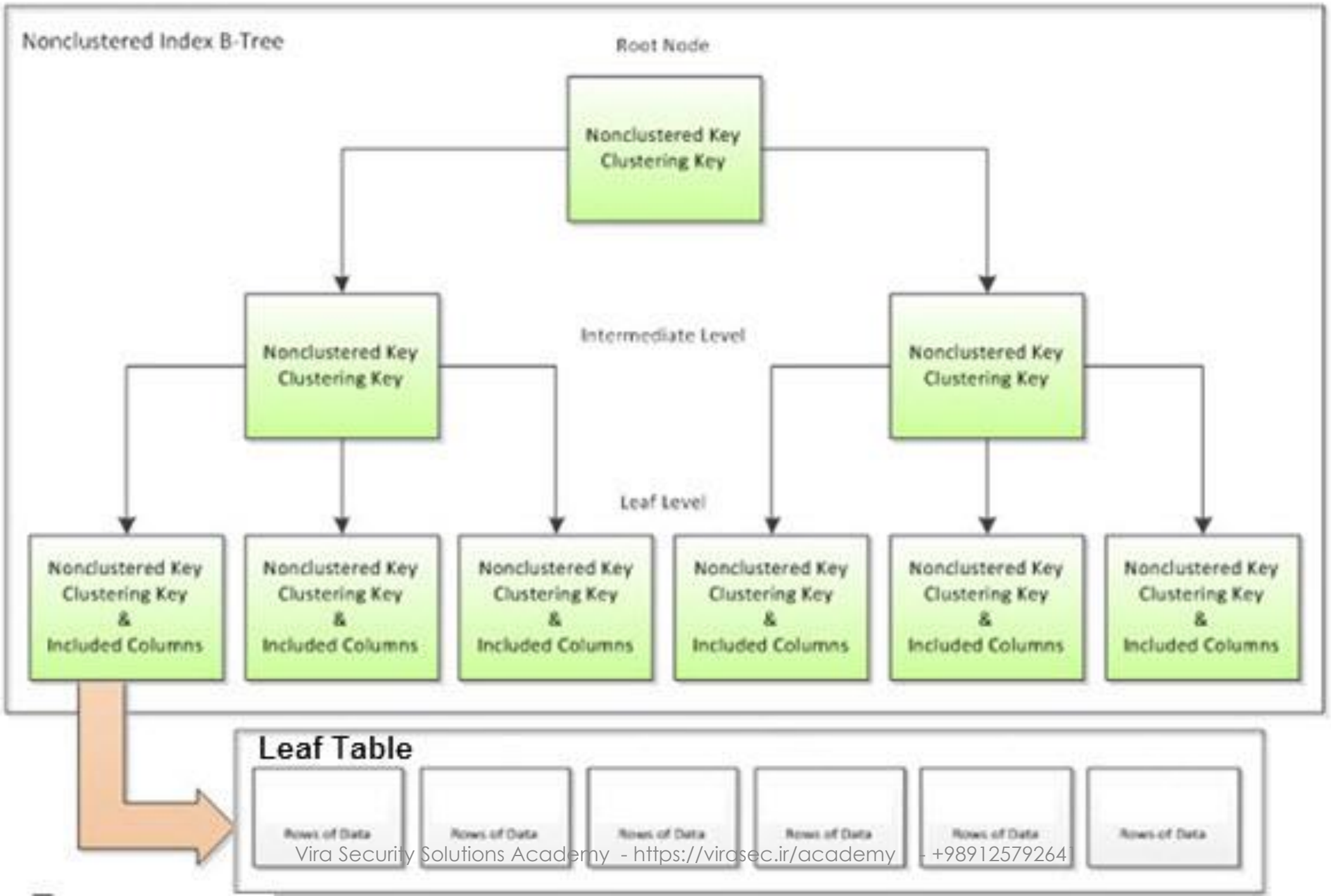
NonClustered Index



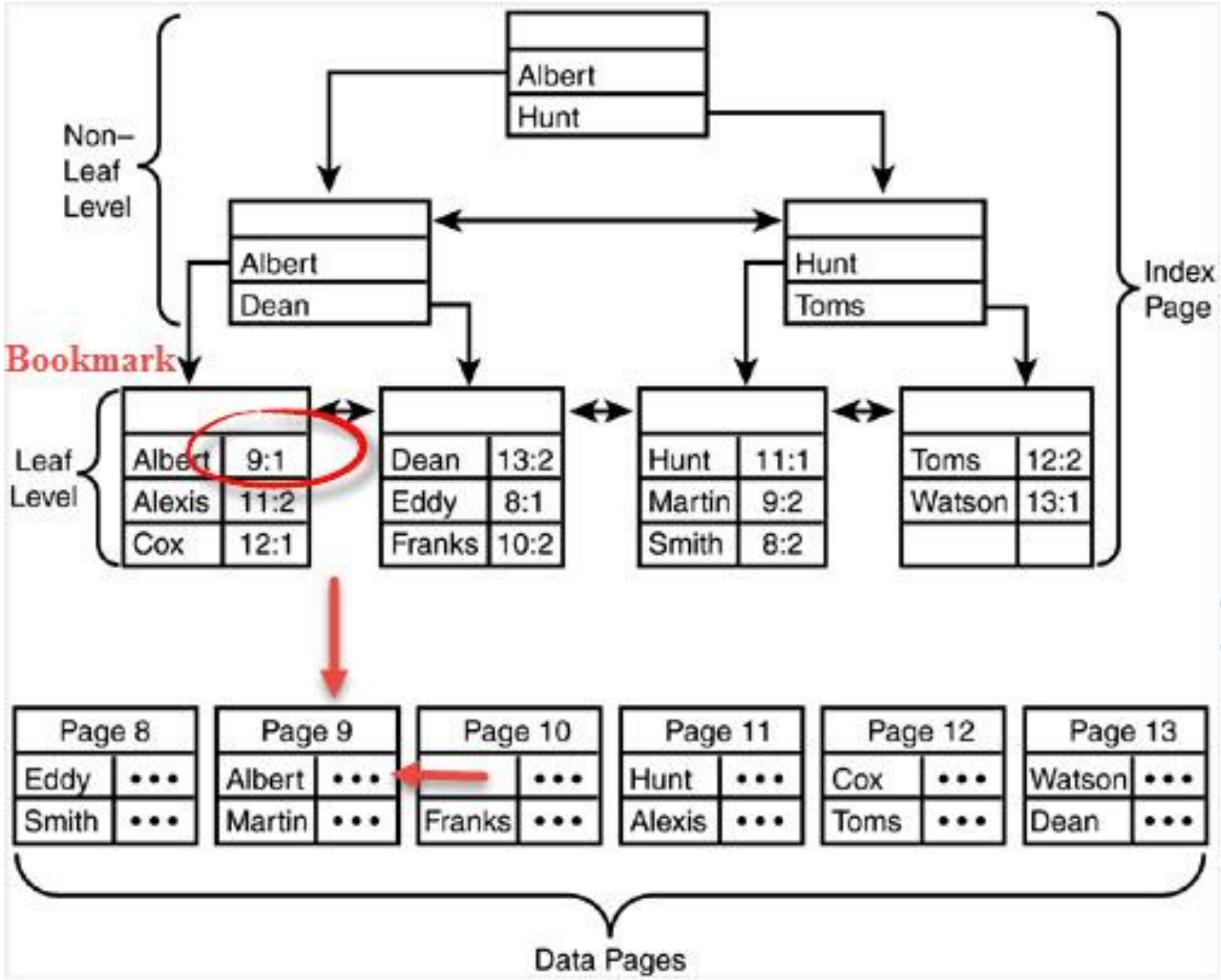

Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Clustered Index B-Tree
Root Node
Clustering Key
Clustering Key
Intermediate Level
Clustering Key
Leaf Level
Clustering Key
&
Rows of Data
Clustering Key
&
Rows of Data
Clustering Key
&
Rows of Data
Clustering Key
&
Rows of Data
Clustering Key
&
Rows of Data
Clustering Key
&
Rows of Data

NonClustered Index

نوع دوم آن دسته ای هستند که روی یک جدول Heap پیاده سازی می شوند که در این نوع نیز در صورتیکه بعد از پیمایش درخت ایندکس و رسیدن به لایه برگ درخت داده های مورد جستجو به همراه کلید ایندکس نبود باید پیمایشی روی جدول Heap برای یافتن رکوردهای مورد نظر صورت گیرد. شکل زیر ساختار این نوع ایندکس را نشان می دهد:

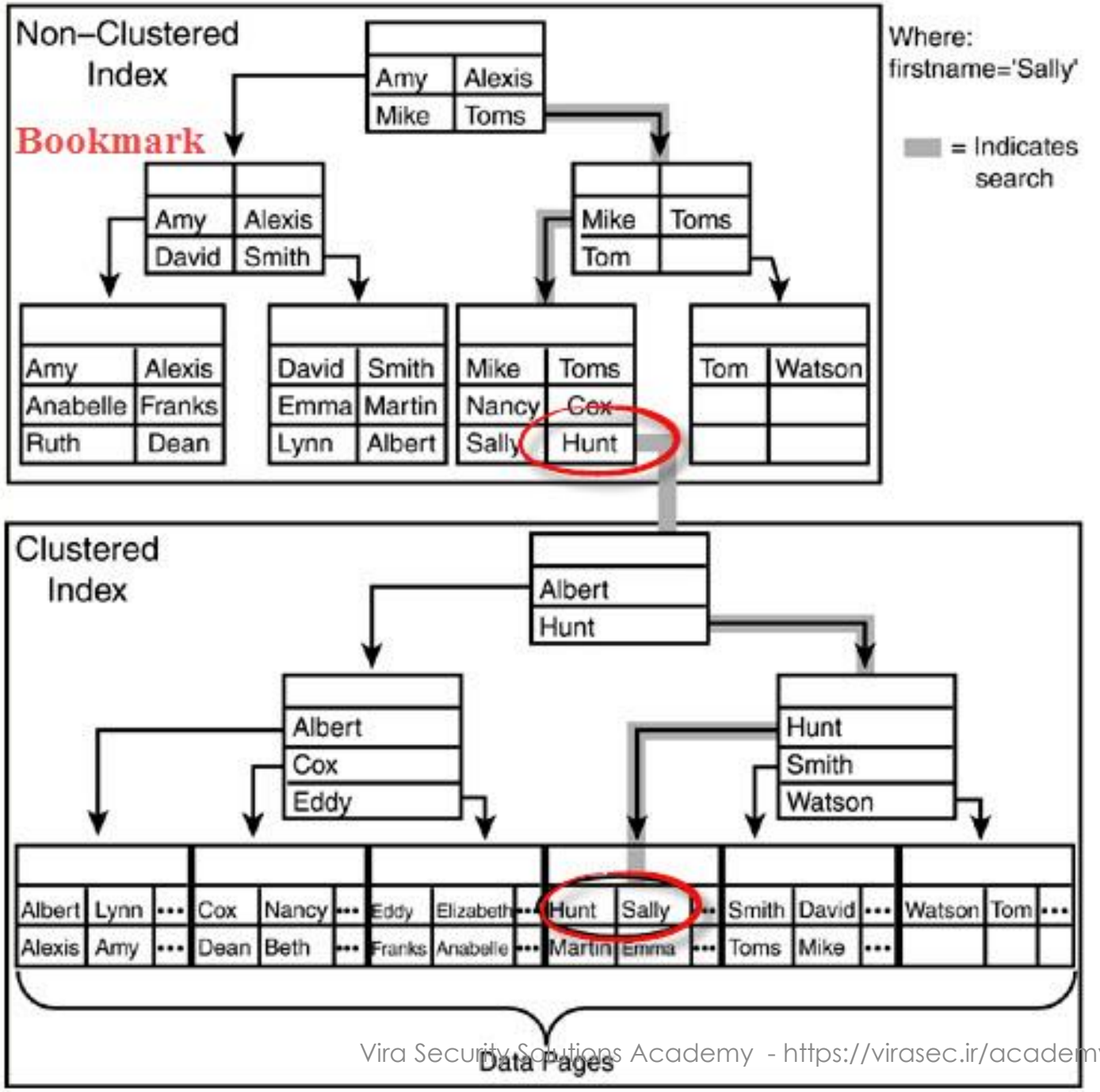

Nonclustered Index B-Tree
Root Node
Nonclustered Key
Clustering Key
Intermediate Level
Nonclustered Key
Clustering Key
Nonclustered Key
Clustering Key
Leaf Level
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Nonclustered Key
Clustering Key
&
Included Columns
Leaf Table
Rows of Data
Rows of Data
Rows of Data
Rows of Data
Rows of Data
Rows of Data

NonClustered Index



Bookmark در یک NonClustered Index

Bookmark در حقیقت نحوه ارتباط یک عنصر Leaf Level به یک رکورد در یک جدول Heap و یا Clusteredگفته می شود که مطابق با شکل این نوع ارتباط در جدول Heapاز دو قسمت تشکیل شده است که قسمت اول آن شماره صفحه و قسمت دوم آن شماره رکورد در صفحه را بیان می نماید:



اما در یک جدول Clustered این مقدار در حقیقت برابر با مقدار کلید ایندکس می باشد که این نوع از Bookmarkدر شکل نمایش داده می شود:

بررسی مفهوم Lookup در یک NonClustered Index:

به مراجعه از Leaf Level به Data Level گفته می شود.

در هنگام استفاده از NonClustered Index رخ می دهد.

دو نوع است:

Key Lookup-1 که در هنگام استفاده از یک NonClustered Index روی یک جدول Clustered رخ می دهد.

Lookup-2 که هنگام استفاده از یک NonClustered Index روی یک جدول Heap اتفاق می افتد.

توجه: هر چه تعداد Lookup در یک query بیشتر باشد IO بیشتر و Cost بالاتر را منجر می شود، بنابراین ما باید نسبت به کم نمودن تعداد LookUp با استفاده از تکنیکهایی مانند Cover Index تلاش نماییم

Cover Indexبه چه معناست؟

به طور کلی برای گنجاندن فیلدهای داخل کلید در داخل ایندکس انجام می شود و دارای خصوصیات زیر است:

1. باعث حذف Lookupو در نتیجه بالا رفتن کارایی می شود.
2. حجم ایندکس را افزایش می دهد
3. کارایی ایندکس ها را بالا می برد

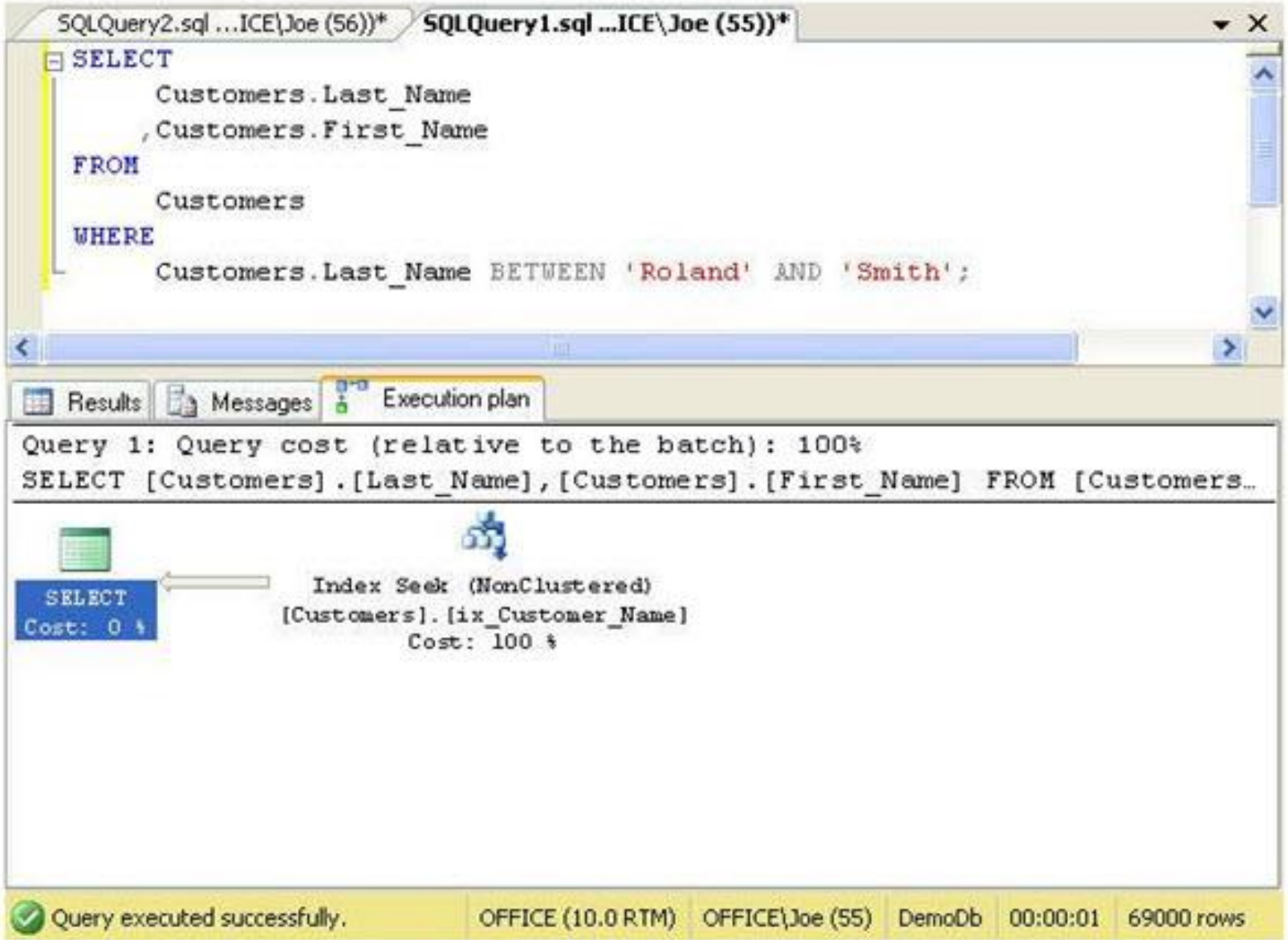
Cover Index

تاثیر استفاده از Cover Index:
اضافه کردن فیلدهای غیر کلید به ایندکس

```
CREATE NONCLUSTERED
INDEX [ix_Customer_Email] ON [dbo].[Customers]
(
            [Last_Name] ASC,
            [First_Name] ASC
)
INCLUDE ( [Email_Address])
```

توجه: Cover Index اگر به صورت درست استفاده نشود باعث سربار بالایی روی اجرای Queryهای جدول و در نتیجه کاهش کارایی می شود

Cover Index


SQLQuery2.sql ...ICE\Joe (56))*
SQLQuery1.sql ...ICE\Joe (55))*
SELECT
Customers.Last_Name
,Customers.First_Name
FROM
Customers
WHERE
Customers.Last_Name BETWEEN 'Roland' AND 'Smith';
Results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT [Customers].[Last_Name],[Customers].[First_Name] FROM [Customers…
SELECT
Cost: 0 %
Index Seek (NonClustered)
[Customers].[ix_Customer_Name]
Cost: 100 %
Query executed successfully.
OFFICE (10.0 RTM)
OFFICE\Joe (55)
DemoDb
00:00:01
69000 rows

مقدار هزینه IO پایین بوده و از NonClustered Index استفاده شده است

Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Number of Rows	69000
Estimated I/O Cost	0.150532
Estimated CPU Cost	0.070557
Estimated Number of Executions	1
Number of Executions	1
Estimated Operator Cost	0.221089 (100%)
Estimated Subtree Cost	0.221089
Estimated Number of Rows	64000
Estimated Row Size	24 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	0

Object

[DemoDb].[dbo].[Customers].[ix_Customer_Name]

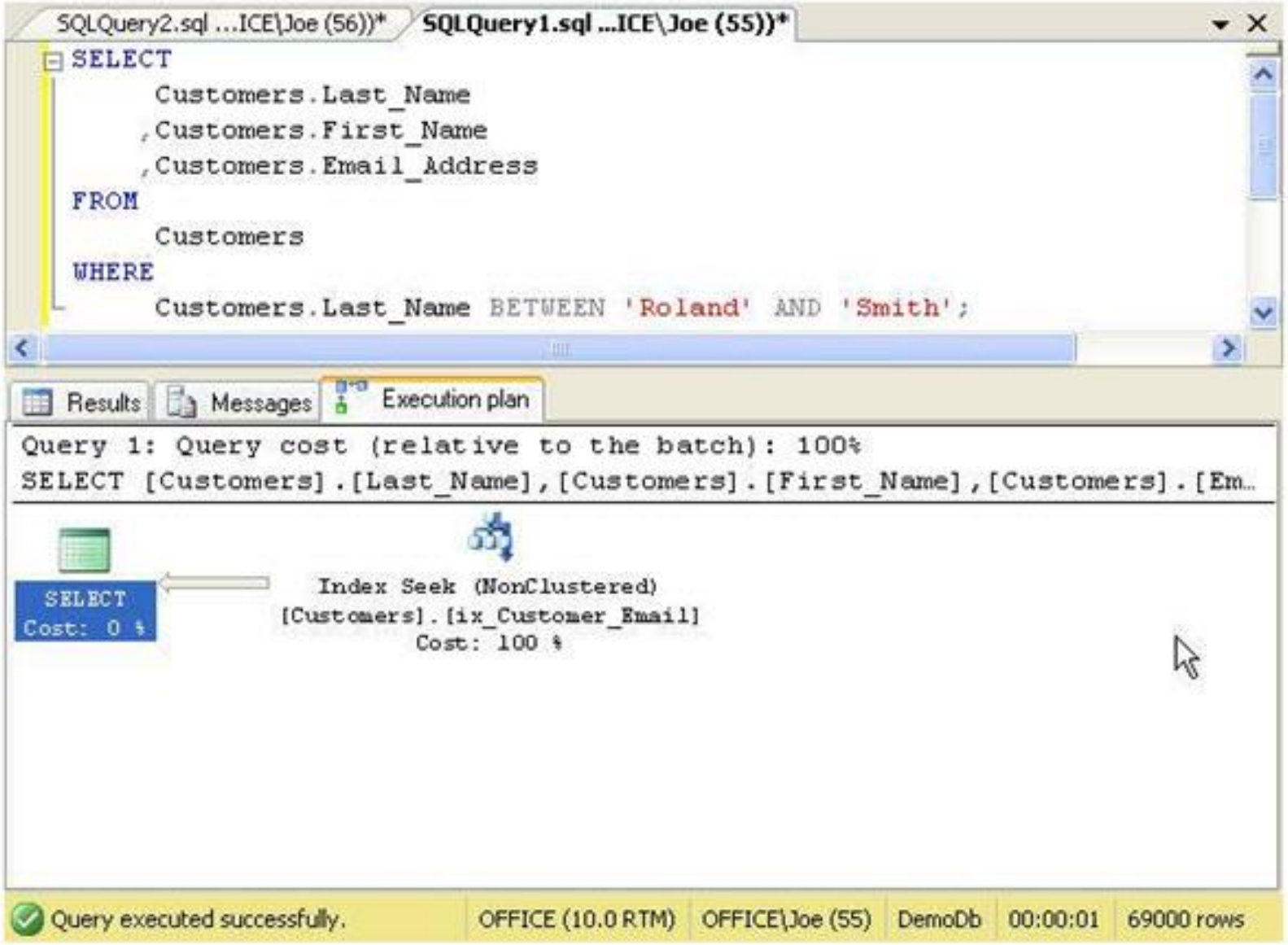
Output List

[DemoDb].[dbo].[Customers].Last_Name, [DemoDb].[dbo].[Customers].First_Name

Seek Predicates

Seek Keys[1]: Start: [DemoDb].[dbo].[Customers].Last_Name >= Scalar Operator([@1]), End: [DemoDb].[dbo].[Customers].Last_Name <= Scalar Operator([@2])

Cover Index


SQLQuery2.sql ...ICE\Joe (56))*
SQLQuery1.sql ...ICE\Joe (55))*
SELECT
Customers.Last_Name
,Customers.First_Name
,Customers.Email_Address
FROM
Customers
WHERE
Customers.Last_Name BETWEEN 'Roland' AND 'Smith';
Results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT [Customers].[Last_Name],[Customers].[First_Name],[Customers].[Em...
SELECT
Cost: 0 %
Index Seek (NonClustered)
[Customers].[ix_Customer_Email]
Cost: 100 %
Query executed successfully.
OFFICE (10.0 RTM)
OFFICE\Joe (55)
DemoDb
00:00:01
69000 rows

با مشاهده Execution Plan مربوط به پرس و جوی بالا طبق شکل زیر می بینیم که باز هم هزینه IOپایین بوده و باز هم همچنان از NonClustered Index استفاده نموده است و این به خاطر اضافه کردن فیلد Email_Address به ایدکس می باشد

Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Number of Rows	69000
Estimated I/O Cost	0.339421
Estimated CPU Cost	0.070557
Estimated Number of Executions	1
Number of Executions	1
Estimated Operator Cost	0.409978 (100%)
Estimated Subtree Cost	0.409978
Estimated Number of Rows	64000
Estimated Row Size	51 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	0

Object

[DemoDb].[dbo].[Customers].[ix_Customer_Email]

Output List

[DemoDb].[dbo].[Customers].Last_Name, [DemoDb].[dbo].[Customers].First_Name, [DemoDb].[dbo].[Customers].Email_Address

Seek Predicates

Seek Keys[1]: Start: [DemoDb].[dbo].[Customers].Last_Name >= Scalar Operator('Roland'), End: [DemoDb].[dbo].[Customers].Last_Name <= Scalar Operator('Smith')

Filtered Index به چه معناست؟

به طور کلی به معنای فیلتر کردن ایندکس و شرکت دادن رکوردهای حائز شرایط در ایندکس می باشد

خصوصیات Filtered Index:

1. افزایش سرعت ایجاد ایندکس
2. افزایش سرعت query ها
3. کاهش حجم ایندکس
4. هزینه کمتر برای بروزرسانی ایندکس

Filtered Index

تاثیر استفاده از Filtered Index

نحوه تعریف Filtered Index بصورت زیر می باشد

```
CREATE NONCLUSTERED INDEX <index name>
ON <table> (<columns>)

WHERE <criteria>;GO
```

مثال زیر تعریف یک Filtered Indexرا نشان می دهد

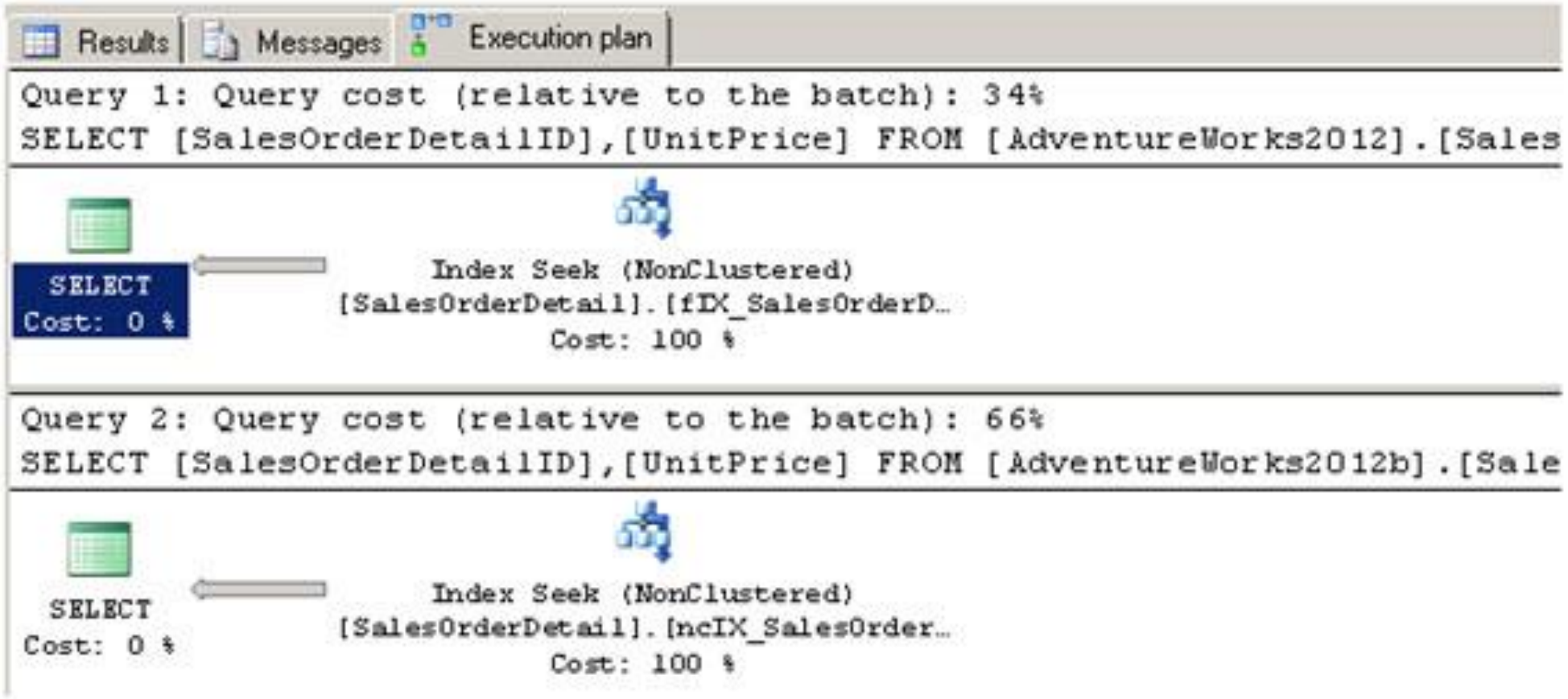
```
--add nonclustered filtered index to UnitPrice column

CREATE NONCLUSTERED INDEX fIX_SalesOrderDetail_UnitPriceON
AdventureWorks2019.Sales.SalesOrderDetail(UnitPrice)
WHERE UnitPrice > 1000 GO
```

Filtered Index


```
--find SalesOrderDetailIDs with UnitPrice > $2000 - now using nonclustered filtered index

SELECT SalesOrderDetailID, UnitPrice
FROM AdventureWorks2019.Sales.SalesOrderDetail
WHERE UnitPrice > 2000
GO

 --find SalesOrderDetailIDs with UnitPrice > $2000 - using nonclustered  index

SELECT SalesOrderDetailID, UnitPrice
FROM AdventureWorks2019b.Sales.SalesOrderDetail
WHERE UnitPrice > 2000 GO
```

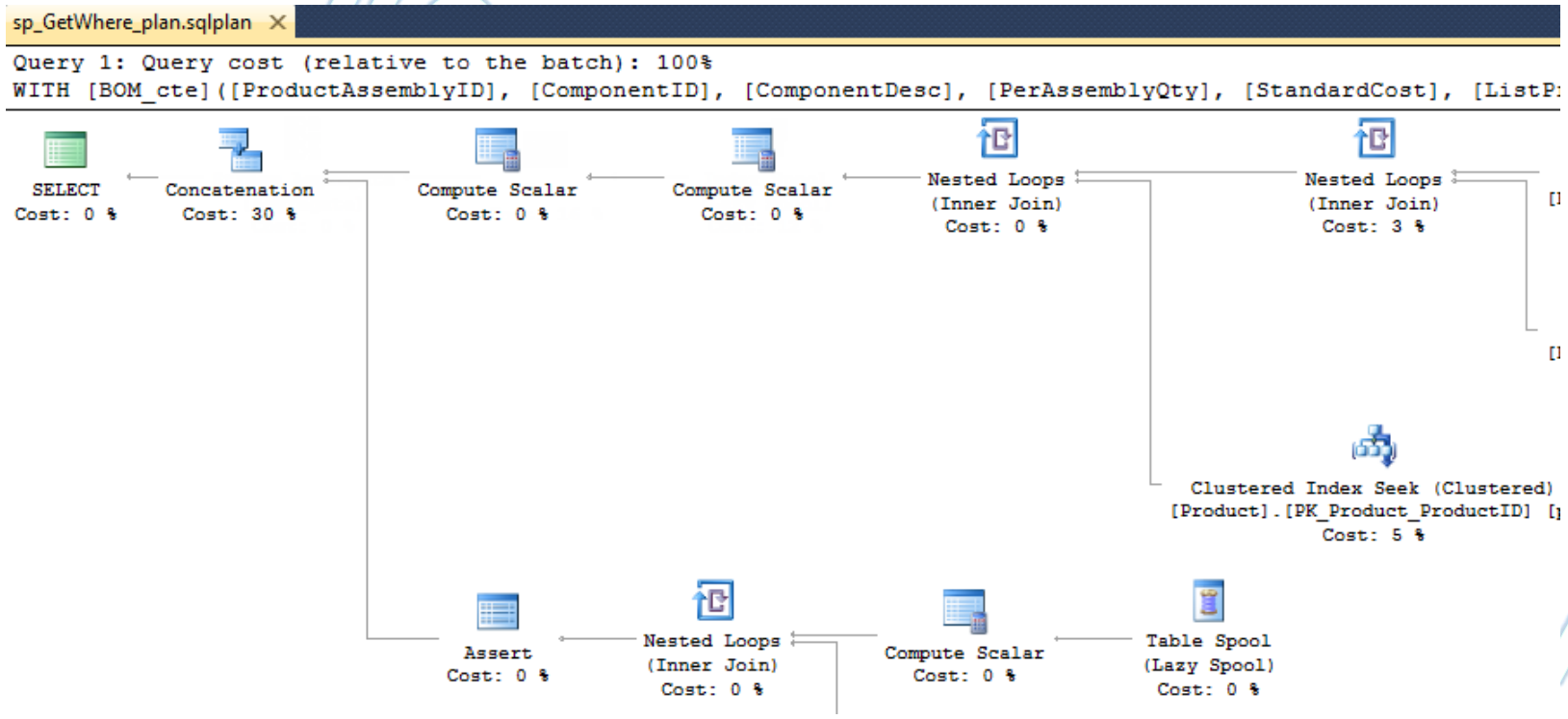
Filtered Index

شکل زیر نمایی از مقایسه Execution Plan مربوط به دو پرس و جوی بالا می باشد و همانطور که دیده می شود query اول کارایی بالاتری داشته است



Execution Plan

نقشه اجرایی و یا Execution Plan لغتی است که مابین کسانی که با مبحث Performance Tuning درگیر هستند زیاد مورد استفاده قرار می‌گیرد.



خواندن Execution Plan از سمت راست به چپ و بالا به پایین انجام می‌شود.

Execution Plan

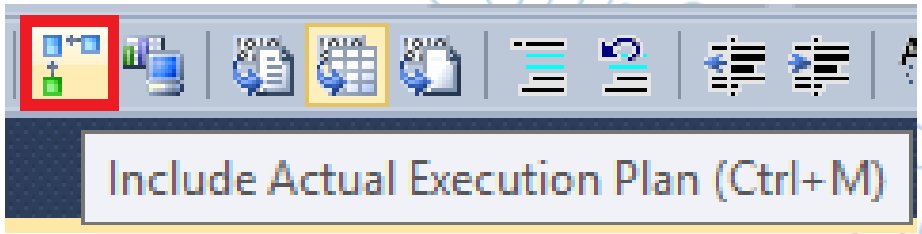
Execution Plan در SQL Server بر دو نوع می‌باشد.

Estimated Execution Plan

نقشه اجرایی تخمینی، این نوع نقشه بدون اجرای کوئری ایجاد شده و تخمینی از عملکرد کوئری می‌باشد. برای بدست آوردن این نوع نقشه کافی است کوئری مورد نظر را Highlight کرده و کلید Ctrl+L را فشار داده و یا از Tool Bar همانند تصویر زیر بر روی دکمه Display Estimated Execution Plan کلیک کنید تا تصویر گرافیکی Execution Plan به شما نمایش داده شود.

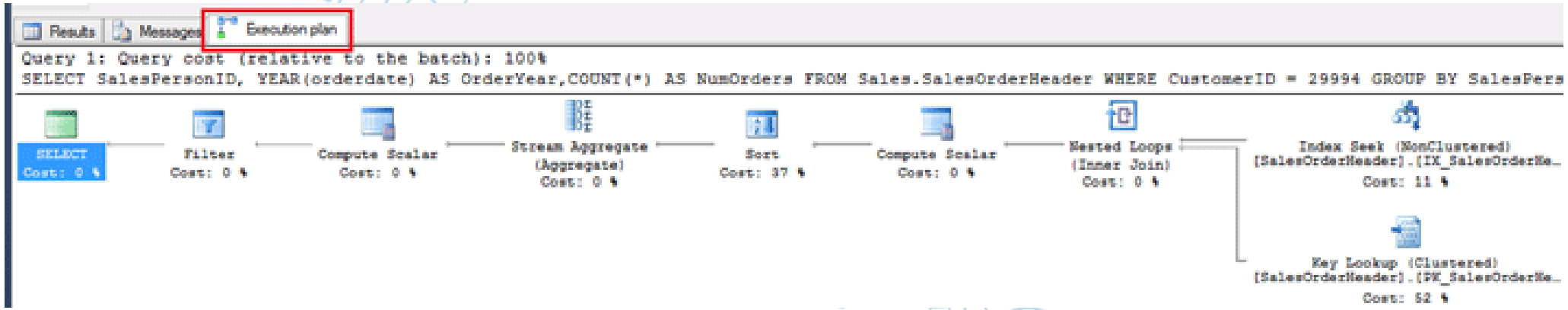
Actual Execution Plan

نقشه اجرایی واقعی، این نوع نقشه پس از اجرای کوئری ایجاد شده و عملکرد واقعی کوئری می‌باشد. برای بدست آوردن این نوع نقشه کافی است کوئری مورد نظر را Highlightکرده و کلید Ctrl+Mرا فشار داده و یا از Tool Barهمانند تصویر زیر بر روی دکمه Include Actual Execution Planکلیک کنید.



Execution Plan

پس از انجام اینکار کوئری مورد نظر خود را اجرا نمایید. پس از اجرای کوئری و نمایش نتایج آن در قسمت پایین یک Tabجدید با نام Execution Planایجاد می‌شودکه حاوی نقشه اجرایی واقعی کوئری شما می‌باشد



این اعداد بر حسب درصد بوده و نمایان‌گر (Costهزینه) کوئری هر کدام از آیتم‌ها می‌باشد. این هزینه بر اساس برآیندی از IOو CPU می‌باشد. هر چه قدر هزینه هر کدام از آیتم‌ها بالا باشد نشان دهنده مصرف IOو CPUبیشتر در آن قسمت می‌باشد. برای مثال در Execution Planمربوط به کوئری زیر همانگونه که مشاهده می‌کنید هزینه Key Lookupبه مراتب بالاتر از سایر آیتم‌های دیگر می‌باشد.

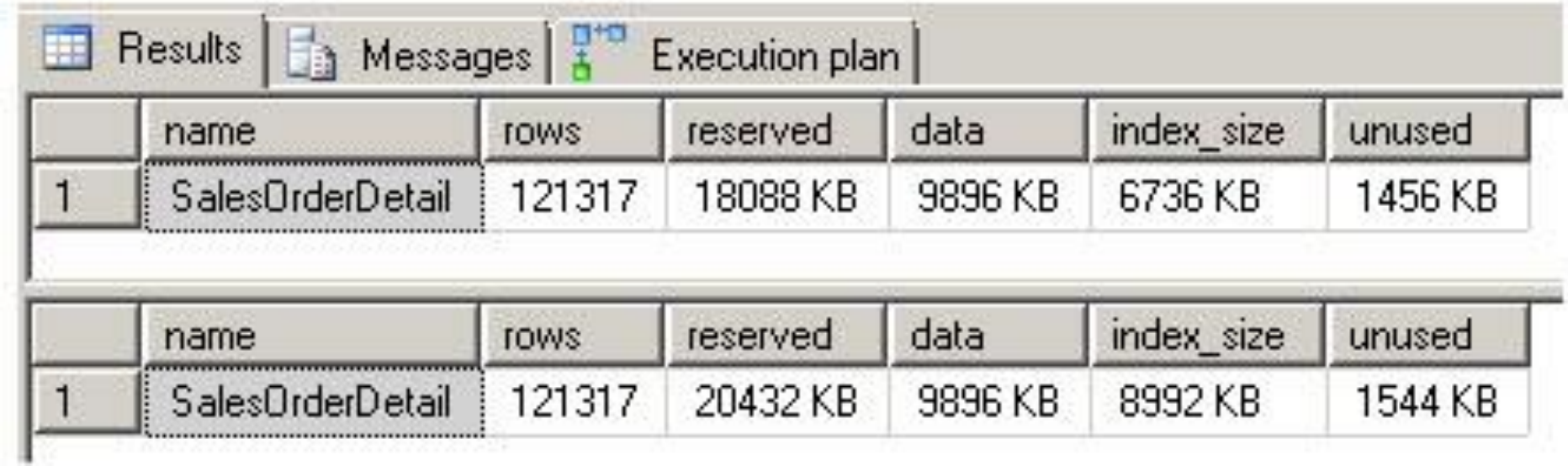
تاثیر استفاده از Filtered Index در حجم ایندکس :

با اجرای query زیر همانطور که در شکل نیز دیده می شود حجم ایندکس کاهش می یابد

```
--get total index size for AdventureWorks2012 database
USE AdventureWorks2012
GO
EXECUTE sp_spaceused 'Sales.SalesOrderDetail'

--get total index size for AdventureWorks2012b database
USE AdventureWorks2012b
GO
EXECUTE sp_spaceused 'Sales.SalesOrderDetail'
```

Filtered Index

در شکل زیر میزان فضا و صفحات بکار رفته در دو جدول با ایندکسی از نوع NonClustered و به صورت Filter و بدون Filter نشان داده شده است



Results | Messages | Execution plan

	name	rows	reserved	data	index_size	unused
1	SalesOrderDetail	121317	18088 KB	9896 KB	6736 KB	1456 KB

	name	rows	reserved	data	index_size	unused
1	SalesOrderDetail	121317	20432 KB	9896 KB	8992 KB	1544 KB

بررسی تاثیر تعریف بیش از حد ایندکس

به طور کلی در صورتی که تعداد **ایندکس** ها بر روی یک جدول بیش از حد متعارف باشد نه تنها کارایی را بالا نبرده بلکه باعث افت شدید کارایی به دلیل وقوع موارد زیر می گردد:

بالا رفتن بیش از حد هزینه IO

افزایش مدت زمان Lockingجدول هنگام بروزرسانی.

افزایش Fragmentationو زیاد شدن تعداد .Page Split

پیاده سازی Primary Keyبا استفاده از ایندکس

1. PKجهت پیاده سازی جامعیت داده – Data Integrityبکار می رود.
2. PKفاقد مقدار تکراری است.
3. PKفاقد مقدار Nullاست.

پیاده سازی Unique Key با استفاده از ایندکس

UKجهت پیاده سازی جامعیت داده – Data Integrityبکار می رود.
UKفاقد مقدار تکراری است.
UKمقدار Nullرا فقط یکبار و به عنوان مقداری معتبر قبول می کند.

Full-Text Searchچیست؟

Full-Text Searchیکی از سرویس های SQL Serverاست.
هدف استفاده از **Full-Text Search**جستجو در داده های حجیم است.
انواع داده هایی که در Full-Text Search قابل پشتیبانی هستند:

Char / NChar
VarChar / NVarChar
Text / NText
Binary / VarBinary
Image
XML
FILESTREAM

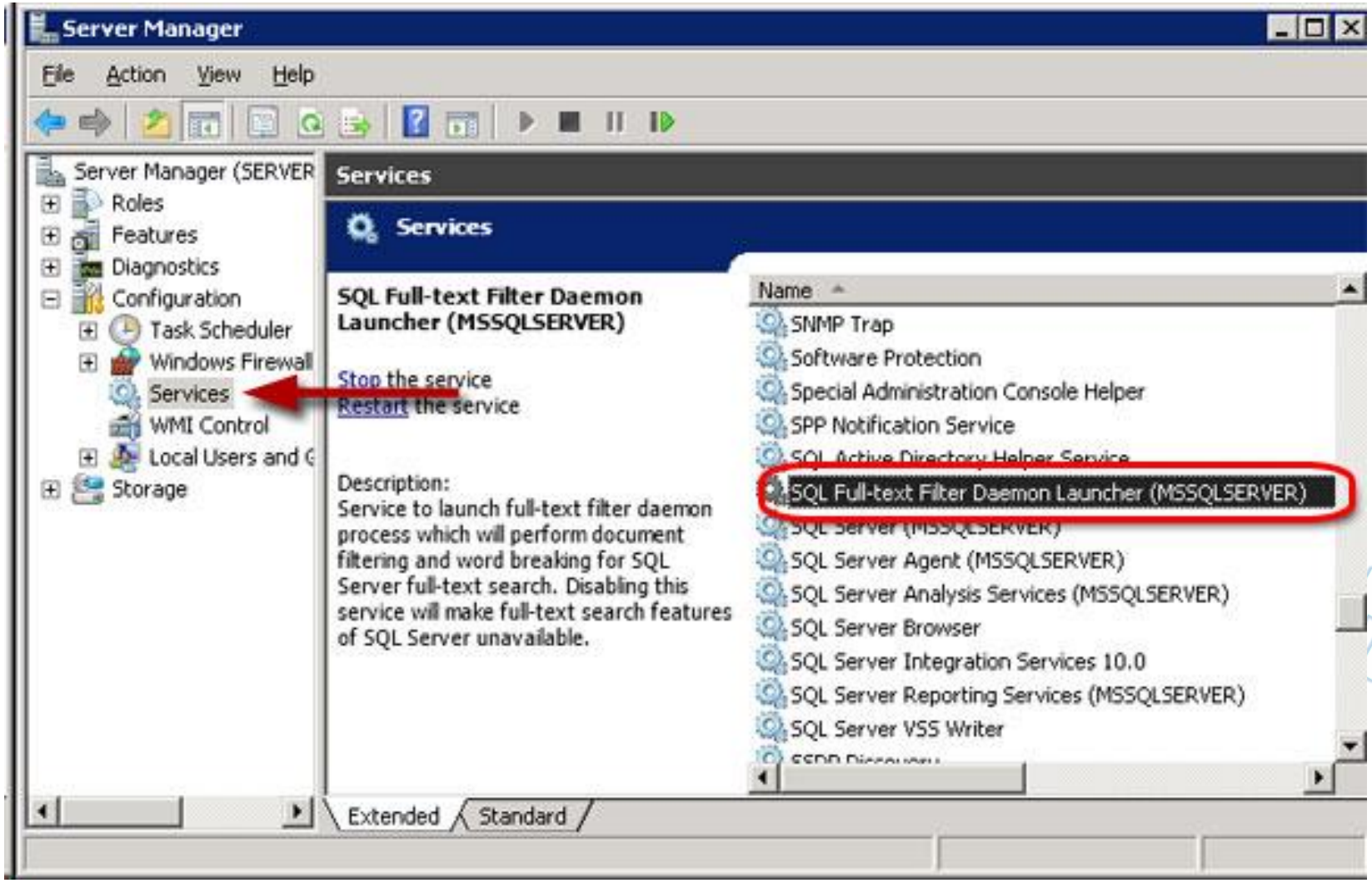
مقایسه Likeو Full-Text Search

در مقابل Full-Text Search از Like نمی توان برای جستجو در بین داده های باینری استفاده کرد و فقط برای جستجو در بین کاراکترها، طراحی شده است.

برای جستجو در بین حجم زیادی از داده ها، دستور Like در مقابل Full-Text Search بسیار کندتر عمل خواهد کرد.

برای جستجو در بین چند میلیون رکورد، دستور Like ممکن است چندین دقیقه طول بکشید در حالی که Full-Text Search در چند ثانیه نتیجه را نشان خواهد داد.

Full-Text Search



راه اندازی سرویس Full-Text Search برای Startیا راه اندازی سرویس Full Text Search، از منوی Startروی Computerراست کلیک کرده و گزینه Manageرا انتخاب نمایید، بلافاصله پنجره Server Managerنمایان می شود. مطابق شکل زیر از سمت چپ گزینه Servicesرا انتخاب کنید و برای Startیا راه اندازی سرویس SQL Full Text Searchروی گزینه Full-text Filter Daemon Launcherکلیک راست کرده و گزینه Startرا انتخاب نمایید:

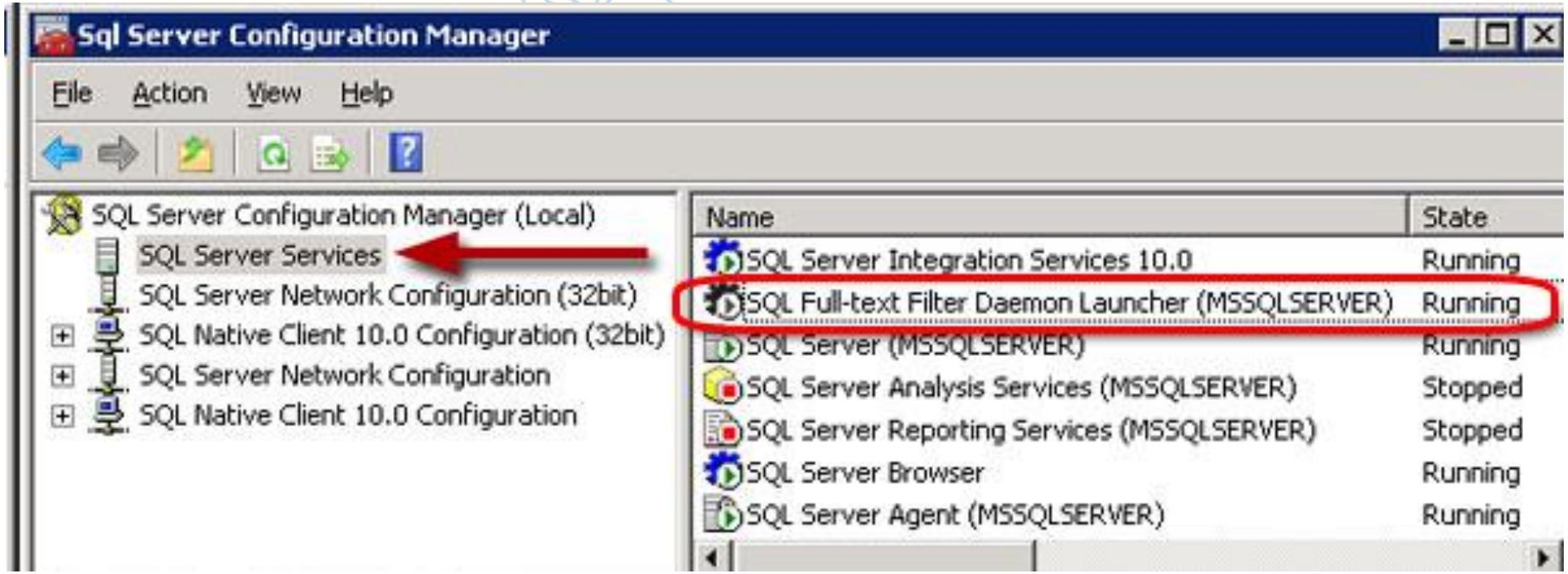
Full-Text Search

نکته: سرویس Full-Text Search به ازای هر Instance می باشد، یعنی اگر روی کامپیوترتان دو Instance از SQL Server نصب کرده باشید، در قسمت Services دو **سرویس Full-Text Search** خواهید دید، البته در انتهای نام سرویس، نام Instance داخل پرانتز ذکر شده است.

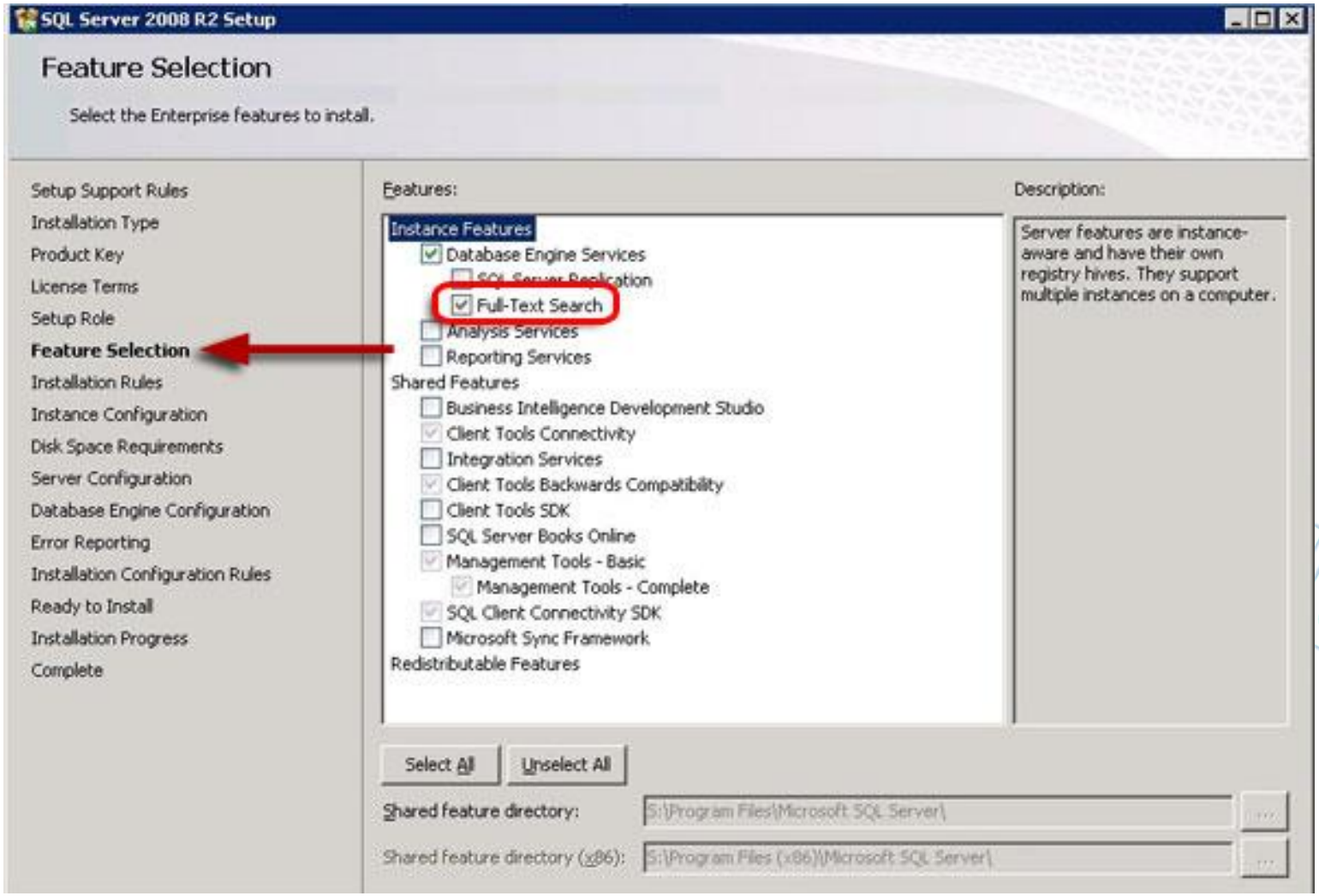
یک روش دیگر برای مشاهده سرویس های SQL Server، از طریق برنامه SQL Server Configuration Manager است، بنابراین روی منوی Start کلیک کرده و گزینه All Programs را انتخاب نمایید، سپس گزینه ... Microsoft SQL Server و سپس Configuration Tools را گسترش دهید و نهایتاً روی SQL Server Configuration Manager کلیک نمایید.

Full-Text Search

پنجره SQL Server Configuration Managerنمایان می شود، حالا با کلیک روی SQL Server Servicesمی توانید کلیه سرویس های SQL Serverرا مشاهده نمایید. حالا مانند قبل روی گزینه SQL Full-text Filter Daemon Launcherکلیک راست کرده و گزینه Startرا انتخاب نمایید



Full-Text Search



توجه: برای استفاده از ویژگی Full-Text Search، باید تیک Full-Text Searchرا هنگام نصب SQL Serverفعال کرده باشید، یعنی مطابق شکل زیر در مرحله Feature Selection، گزینه Full-Text Searchفعال شده باشد

روش کار Full-Text Search

همان طور که قبلاً گفته شد، از **Full-Text Search** برای جستجو در داده های حجیم استفاده می شود، به عنوان مثال جدول زیر را که شامل دو رکورد "ID" و "Description" است را تصور فرمایید:

ID	Description
1	برای جستجو در داده های حجیم Full-Text Searchهمان طور که قبلاً گفته شد، از قابلیت "Descriptionو " "IDاستفاده می شود، به عنوان مثال جدول زیر را که شامل دو رکورد " را تصور فرمایید:
2	است. SQL Serverیکی از سرویس های Full-Text Search
3	جستجو در داده های حجیم است. Full-Text Searchهدف استفاده از
4	است، پیشنهاد می شود قبل SQL Server، قسمتی از آموزش پیشرفته Full-Text Search ، قسمت های قبلی را مرور فرمایید.Full-Text Searchاز مطالعه ی
...	...

Full-Text Search

اگر قابلیت Full-Text Search را روی جدول بالا فعال کنید، به ازای تک تک کلمات بکار رفته در فیلد "Description" بهمراه شماره رکورد آنها، ایندکس ایجاد خواهد شد:

شماره رکورد	کلمه
1	همان
1	طور
1	Full-Text
2	Full-Text
3	Full-Text
4	Full-Text
1	جستجو
3	جستجو
...	...

از این لحظه به بعد اگر به دنبال کلمه ای در فیلد " "Descriptionباشید، بجای اینکه تمام ستون Description جستجو شود، جدول ایندکس های مربوط به Full-Text Search جستجو خواهد شد. و این کار بشدت سرعت اجرای کوئری ها را افزایش خواهد داد.

در Full-Text Search براساس یک یا چند ستون جدول ایندکس گذاری انجام می شود و برای هر ستون می توان از قواعد یک **زبان خاص** استفاده نمود.

توجه: کلمات اضافه مثل "که"، "است"، "را" و ... ارزش ایندکس کردن را ندارند، شما می توانید به راحتی این کلمات را به Full-Text search معرفی نمایید. (StopList)

Full-Text Search

بررسی اصطلاحات رایج در Full-Text Search

پیشنهاد می شود قبل از فعال سازی Full-Text Searchبا اصطلاحات زیر آشنا شوید، با فهم دقیق هر کدام از اصطلاحات زیر بهتر می توانید از امکانات Full-Text Searchاستفاده نمایید:

۱- Term

Term، جلمه یا کلمه ای است که برای تهیه ایندکس مورد استفاده قرار می گیرد.

۲- Full Text Index

Full Text Index ظرفی است که فیلدهای مورد نظر ما جهت ایندکس را نگهداری می کند.

توجه: در Full Text Index تعداد n فیلد از جداول مختلف می تواند قرار بگیرد.

بررسی اصطلاحات رایج در Full-Text Search

۳- Full Text Catalog

Full Text Catalogظرفی است که یک یا چند Full Text Indexرا نگهداری می کند.
قابلیت ذخیره سازی Full Text Catalogدر File-Groupاز نسخه SQL Server 2008به بعد اضافه شده است.
توجه: در مواردی که حجم اطلاعات زیاد می شود (چند صد گیگابایت در هر جدول) انتخاب نحوه ذخیره کاتالوگ و محل آن مهم است و می بایست با دقت انجام شود. مثلاً کاتالوک در همان هارد دیسکی باشد که جدول در آن ذخیره شده یا کاتالوگ و جدول در چند سرور پخش شود.

۴- Word Breaker

Word Breaker، مجموعه قواعد زبان شناسی است که براساس آن لغات یک جمله تکه تکه می شود. و در واقع با استفاده از Word Breaker، می توان فهرستی از کلمات مورد نیاز اینکدس ها را استخراج نمود.

بررسی اصطلاحات رایج در Full-Text Search

مثال: لطفاً به جمله **"آموزش افزایش سرعت بانک های اطلاعاتی"** توجه فرمایید. یک روش استخراج کلیمات جمله مذکور بصورت زیر خواهد بود:

آموزش
افزایش
سرعت
بانک های
اطلاعاتی

اما آیا کلمه **"بانک های"** را نمی توان به دو کلمه **"بانک"** و **"های"** شکست ...!؟ در واقع Word Breakerقاعده و الگوی شکستن و تکه تکه کردن کلمات را مشخص می کند.

بررسی اصطلاحات رایج در Full-Text Search

۵- Token

Tokenکلمه یا رشته ای که توسط Word Breakerاستخراج می شود. در مثال بالا، هر کدام از کلمات "**آموزش**"، "**افزایش**"، "**سرعت**"، "**بانک های**" و "**اطلاعاتی**" هر کدام یک Tokenهستند.

۶- Stemmer

با استفاده از Stemmerمی توان با توجه به قواعد زبان، هم خانواده ها یا صرف فعلهای یک لغت را تولید نمود.
مثال: writeو wroteیا footو feet

بررسی اصطلاحات رایج در Full-Text Search

۷- Filter

با استفاده از Filterمی توان متن ذخیره شده در نوع داده varbinaryرا استخراج نمود . انواع فایلهای MS Officeجزء نوع هایی است که فیلترها امکان استخراج متن از آنرا دارند.
به ازای هر نوع فایل یک DLLوجود دارد، مثلاً برای فایل های Wordیک DLLخاص وجود دارد و زمانی که نوع فایل تشخیص داده شد، DLLمربوطه روی سرور لود شده و با استفاده از آن عملیات Full-Text Searchانجام خواهد شد.
شرکتهای دیگر مثل Adobeبعنوان مثال برای نوع فایل PDFفیلترهای سفارشی خودشان را ساخته اند که با یک جستجو در گوگل آنها را خواهید یافت.

۸- Populationیا crawl

با استفاده از Populationعملیات خزش یا crawlدر متن انجام می شود. و در واقع **Population**وظیفه ایجاد و نگهداری و به روز رسانی Full Text Indexرا دارد.

بررسی اصطلاحات رایج در Full-Text Search

۹- Noise Words یا Stop Word

Stop Word، لغاتی هستند که کمکی در جستجوها نکرده و بی دلیل حجم ایندکس ها را افزایش می دهند.
Stop Wordها می بایست نادیده گرفته شوند.
مثال: در جمله "شما در آموزش افزایش سرعت بانک اطلاعاتی، جستجو در بین فایل و عکس را یاد می گیرد " کلمات زیر نیازی به ایندکس شدن ندارند:
شما
در
را

۱۰- Stop List

Stop Listظرفی است که Noise Wordها را در خود نگهداری می کند.

۱۱- Thesaurus

Thesaurusیک فایل XMLاست و حاوی اطلاعات مترادف ها می باشد.
مثال: Internet Explorerو IEمترادف است.

Full-Text Search

آیا Instance جاری از قابلیت Full-Text Search پشتیبانی می کند

```
SELECT SERVERPROPERTY('IsFullTextInstalled')
```

Output:

0 Not Installed
1 Installed

Fragmentation

Fragmentation چیست؟

Fragmentationبمعنی از هم گسیخته شدن یا متلاشی شدن است.

در مثال زیر، نحوه ی قرار گیری داده های A و B و C قبل از اینکه Fragmentation اتفاق بیافتد را مشاهده می کنید:

فرض کنید بخواهیم به A مقداری اضافه کنیم، اتفاقی که خواهد افتاد از قسمت "فضای خالی" مطابق شکل زیر به آن اختصاص خواهد داد و این باعث بروز مشکل Fragmentation یا از هم گسیختگی داده ها خواهد شد:

برای رفع مشکل Fragmentation باید مقدار A دوم در کنار اولی قرار بگیرد:

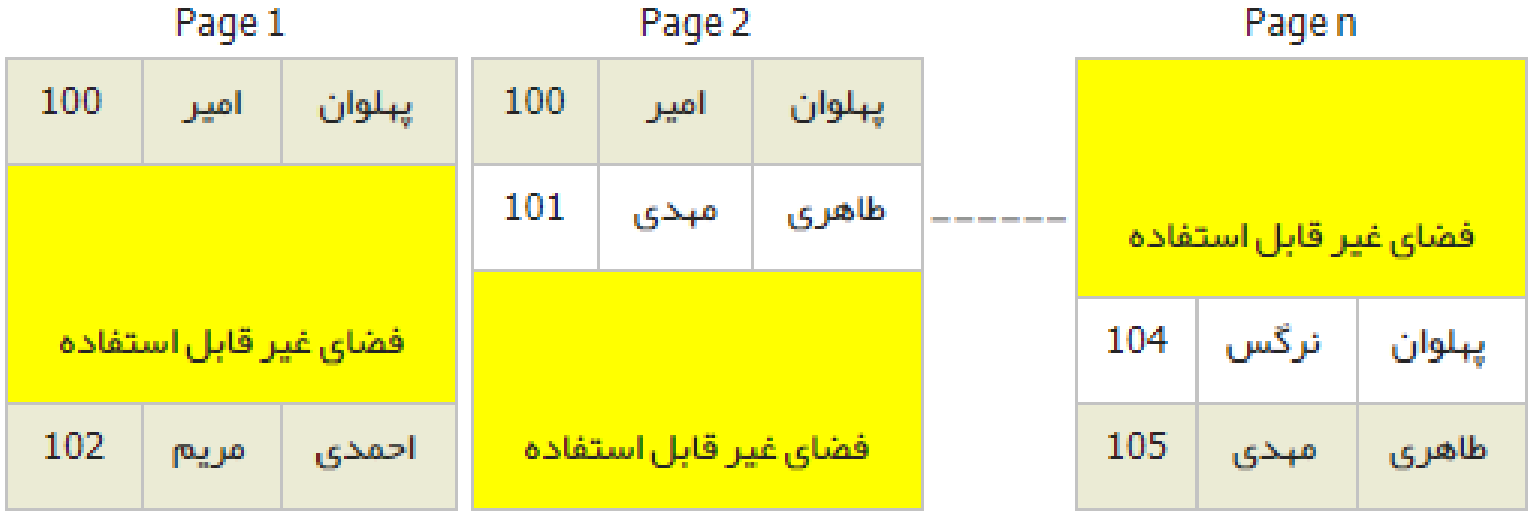
توجه: حالت بهینه ذخیره شدن داده ها بنحوی است که آدرس فیزیکی و منطقی آنها یکسان باشد. همان طور که می دانید داده ها در SQL Serverدر مفهومی بنام Page ذخیره می شوند، هر Page به صفحه ی "قبلی" و "بعدی" خودش اشاره می کند. بنابراین اگر صفحه ی قبلی و بعدی Page جاری دقیقاًPage های کناری خودش باشد، بهترین حالت ممکن اتفاق افتاده است و مشکل Fragmentation وجود نخواهد داشت.

انواع Fragmentation

1. **Fragmentation** یا بهم ریختگی Internal یا Logical
2. **Fragmentation** یا بهم ریختگی Extent یا External
3. **Fragmentation** یا بهم ریختگی در سطح فایل (File Level)

1- Fragmentation یا بهم ریختگی Internal یا Logical

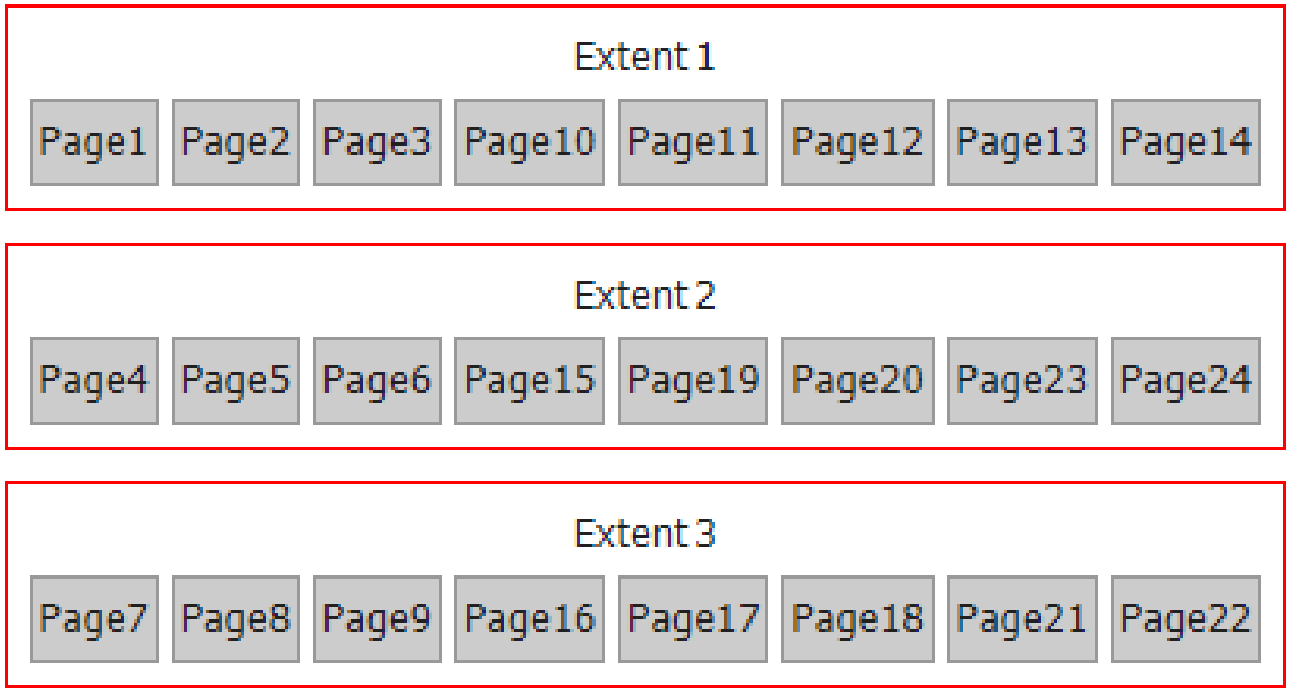
Fragmentation یا بهم ریختگی Internal در اثر وجود فضاهای غیرقابل استفاده در یک Page ایجاد می شود. لطفاً به مثال زیر توجه نمایید:



Fragmentation یا بهم ریختگی Extent یا External

در این نوع **Fragmentation**، ترتیب منطقی صفحات با ترتیب فیزکیشان یکسان نیست.

همان طور که قبلا توضیح داده شد، هر ۸ پیج داخل یک Extent قرار می گیرد، این نوع **Fragmentation** زمانی اوضاع را وخیم تر می کند که برای مرتب سازی مجبور شویم، بین Extentها حرکت کنیم. لطفاً به مثال زیر در این زمینه توجه فرمایید



Fragmentationیا بهم ریختگی در سطح فایل ((File Level

این نوع Fragmentationزمانی رخ می دهد که بخش های مختلف یک فایل در قسمت های مختلف دیسک ذخیره شوند. به مثال زیر که مربوط به نحوه ی قرار گیری فایل های MDFو NDFیک پایگاه داده فرضی است توجه فرمایید

توجه: یکی از راه هایی که می توان از بروز Fragmentationدر سطح فایل جلوگیری کرد، تخمین صحیح فضای مورد نیاز پایگاه داده، هنگام ایجاد آن است. بعنوان مثال اگر برای ایجاد پایگاه داده X، تخمینتان ۲۰ گیگ است، از همان ابتدای ساخت پایگاه داده، این فضا را در اختیار آن قرار دهید تا گسستگی بین فایل های MDFیا NDFو ... جلوگیری نمایید.

عوامل بوجود آمدن Fragmentationدر سطح فایل:

1. Shrinkکردن DataFile
2. عدم تنظیم درست گزینه Auto Growingهنگام ساخت پایگاه داده
3. عدم تنظیم درست گزینه های Allocation Unit Sizeو Strip Size

Execution Plan چیست؟

Execution Plan به شما کمک می کند تا دریابید:

1.چرا یک Query کند است و زمان اجرای زیادی را مصرف می کند؟

2.آیا **SQL SERVER** از ایندکس- Index من استفاده می کند یا خیر؟

3.چرا **SQL SERVER** از ایندکس- Index من استفاده نمی کند؟

4.آیا این پرس و جو - Query از دیگری سریعتر اجرا می شود؟

5.آیا نیاز است که من ایندکس دیگری تعریف کنم و اگر نیاز است روی چه فیلدهایی و چه نوع ایندکسی؟

6.و ...

نحوه نمایش Execution Plan

برای نمایش Execution Plan باید شما دسترسی لازم برای اجرای **Query**را روی پایگاه داده داشته باشید، در صورتیکه شما یکی از نقشهای sysadmin, dbcreator یا db_owner را داشته باشید به دسترسی دیگری نیاز ندارید.

برای نمایش **Execution Plan** می توانید از کلید میانبر ctrl+Lبا انتخاب Queryمورد نظر استفاده نمایید

انواع Plan

estimated plan : ارزیابی اجرای پرس و جو قبل از اجرا و برآورد وضعیت اجرا را که توسط Optimizerبدست می آید نشان می دهد و به یکی از روشهای زیر قابل دسترسی می باشد:

- ✓ با کلیک روی آیکن Display Estimated Execution Planروی نوار ابزار.
- ✓ با راست کلیلک روی پنجره queryو انتخاب same option.
- ✓ با کلیک روی Query optionروی نوار menu barو انتخاب same choice.
- ✓ کلیدهای CTRL+L.

actual plan : پلان واقعی اجرای queryرا نشان می دهد و به یکی از روشهای زیر قابل دسترسی می باشد:

- ✓ کلیک روی آیکن Include Actual Execution Planدر نوار ابزار.
- ✓ کلیک راست روی پنجره queryو انتخاب Include Actual Execution Planاز منو.
- ✓ انتخاب same optionدر منوی Query.
- ✓ کلیدهای CTRL+M.

Execution Plan

```
SELECT *
FROM dbo.DatabaseLog;
```

برای نمایش **Execution Plan** باید با دستور زیر امکان نمایش آنها فراهم شود:

```
-- For Getting the estimated text plan
SET SHOWPLAN_ALL ON;
-----------------------------------------
-- For Getting the actual text plan
SET STATISTICS PROFILE ON;
```

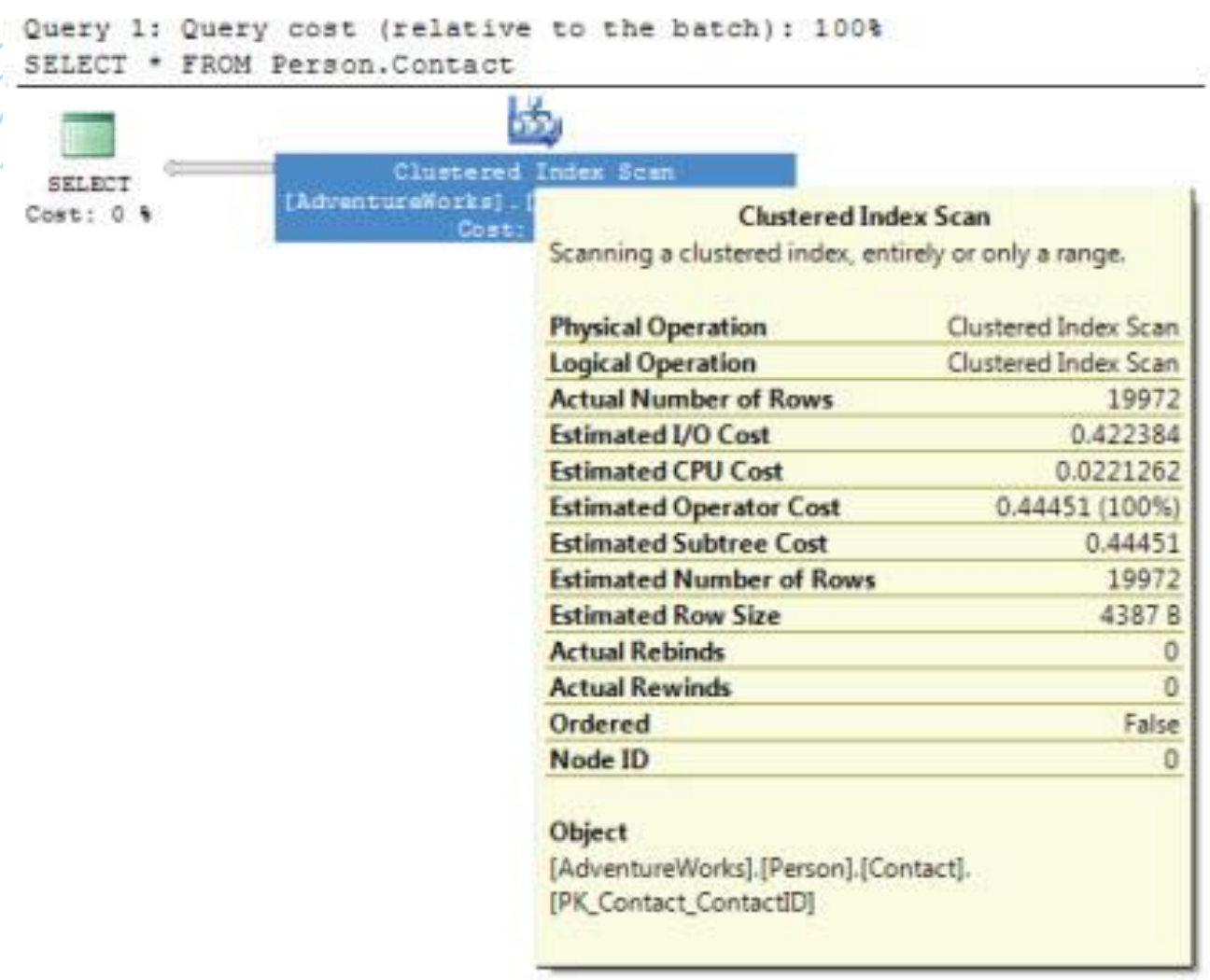
اطلاعات کسب شده از Execution Plan

با کمک ToolTipهای موجود در Execution Plan می توانیم به مطالب زیادی دسترسی پیدا کنیم. هر شکل موجود در این پلان گویای یک حقیقت است و ما برای تحلیل بهتر یک queryو استفاده مناسب از Execution Plan در راستای بهینه کردن query باید به تمامی این اطلاعات و اشکال اشراف کامل داشته باشیم.

در ادامه به تعدادی از این موارد اشاره می کنیم و برای آشنایی کامل با Execution Plan پیشنهاد می کنیم جلسه سوم مجموعه افزایش کارآیی و سرعت بانک اطلاعاتی را خریداری نمایید:

- ✓ Cached plan size : مقداری از حافظه که توسط این پلان در حین کش پلان مورد استفاده قرار می گیرد.
- ✓ Degree of Parallelism : تعداد پردازنده ای که توسط این پلان مورد استفاده قرار می گیرد.
- ✓ Estimated Operator Cost : هزینه اجرای پرس و جو را ارزیابی می نماید.
- ✓ Estimated Subtree Cost : هزینه اجرای این قسمت از queryرا نسبت به کل آن بیان می کند و از راست به چپ این مراحل دنبال می شوند.
- ✓ Estimated Number of Rows : ارزیابی تعداد ردیف انتخاب شده توسط queryرا نشان می دهد و توسط Optimizer محاسبه می شود.
- ✓ Actual Number of Rows : تعداد واقعی ردیف انتخاب شده توسط queryرا نشان می دهد.

Execution Plan



ترتیب اجرای مراحل در Execution Plan

ترتیب اجرای مراحل از بالا به پایین و از راست به چپ می باشد؛ شکل زیر این ترتیب را بهتر نشان می دهد:

```
Query 1: Query cost (relative to the batch): 100.00%
Query text: DELETE [Orders] WHERE [orderid]=@1
```