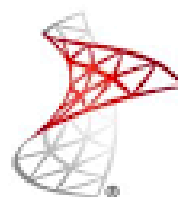




کوئری نویسی پیشرفته



Microsoft®
SQL Server®
2017

مدرس: مسعود طاهری

nikamooz;
آموزش برنامه نویسی و اجرای پروژه

www.NikAmooz.com

جلسه دهم



معرفی مسعود طاهری

1. مدرس و مشاور SQL Server (نیک آموز)
2. رئیس واحد توسعه پایگاه داده پرداخت الکترونیک سداد (بانک ملی)
3. کارشناس ارشد تحلیل و طراحی تجارت الکترونیک پارسین
4. مشاور شرکت‌های بزرگ نرم‌افزاری (رایورز، سازمان کشتیرانی و...)
5. متخصص انجام پروژه‌های SQL Server (امنیت بانک اطلاعاتی، HA و...)
6. و...

فرمت ذخیره سازی Columnstore Index

مروری بر ساختار Columnstore Index

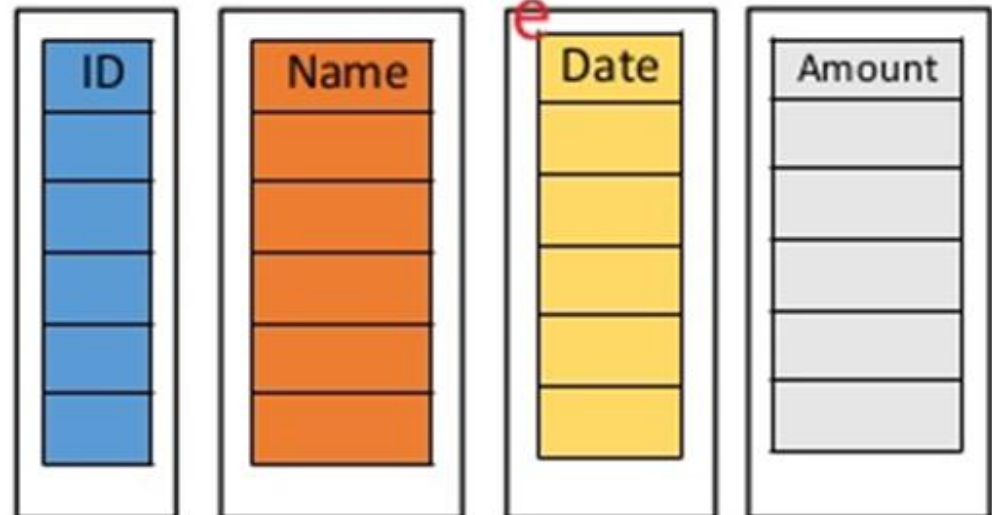
دخیره داده‌ها در قالب ستونی

ID	Name	Date	Amount

RowStore



ColumnStore

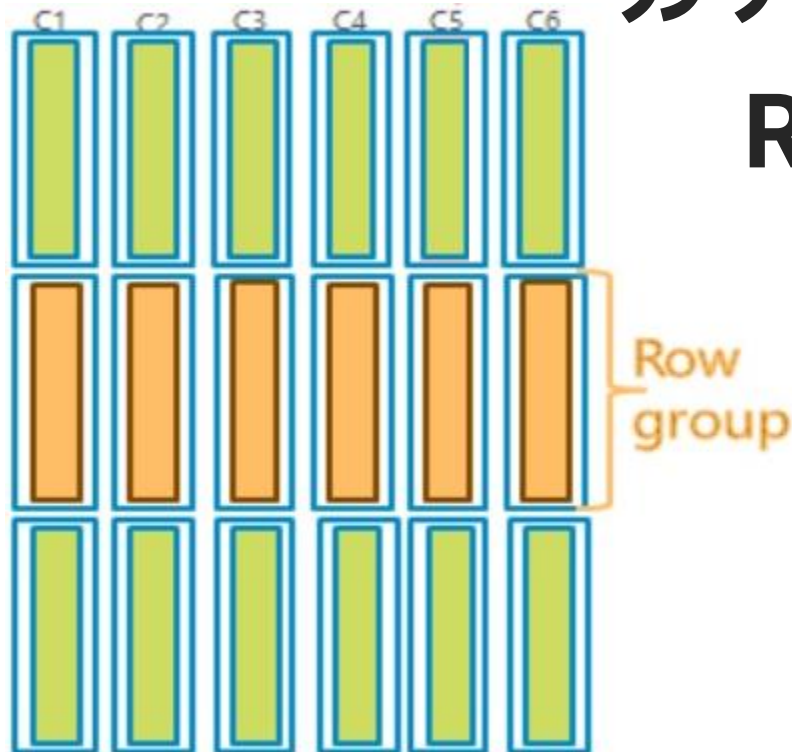


بررسی مفهوم Row Group

1. مجموعه‌ای مجزا از رکوردها

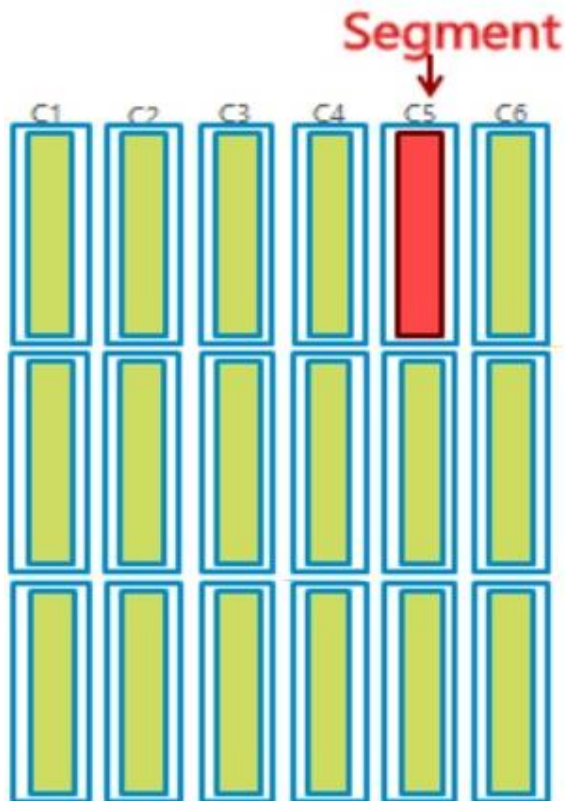
2. شامل $2^{20} = 1048576$ رکورد

3. دسترسی Thread ها به RG



بررسی مفهوم Segment

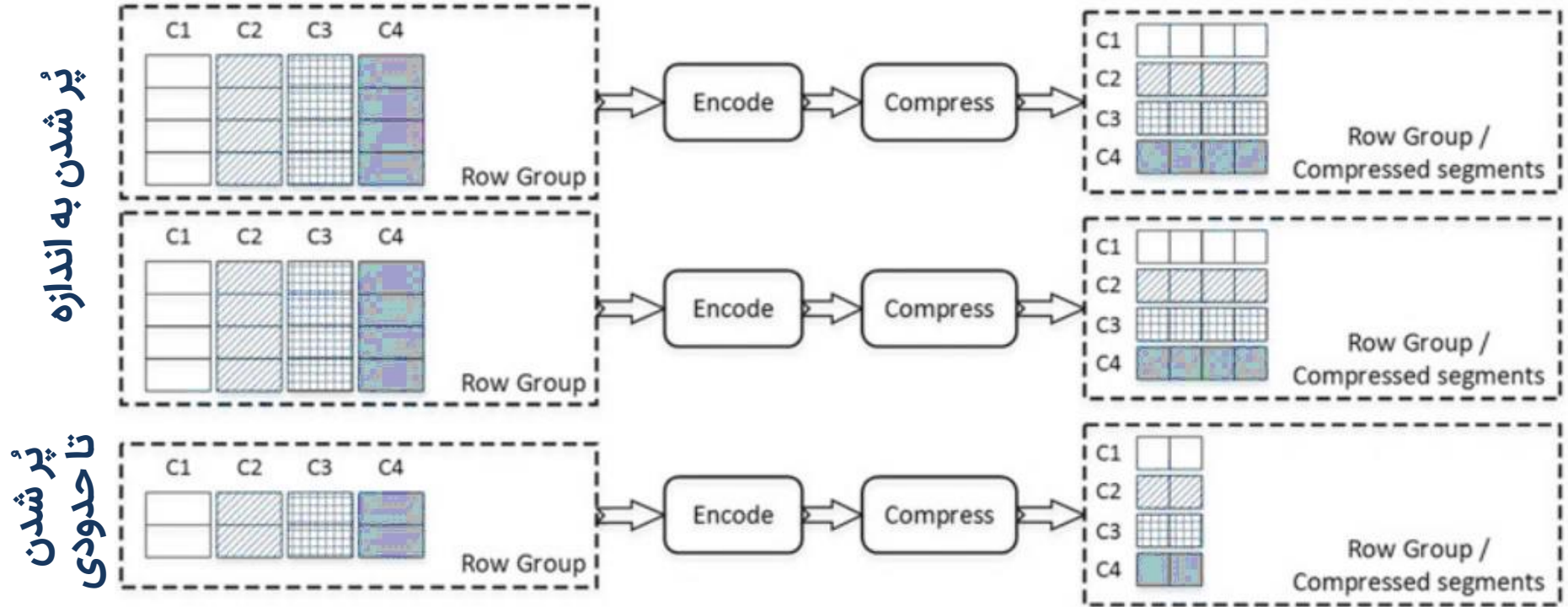
1. ستون‌های موجود در یک Row Group
2. لود کردن Segment به حافظه
3. ذخیره Meta Data در SG
4. انجام عملیات فشرده‌سازی



فرآیند ساخت Columnstore Index

1. انجام پروسه Encode (رمزگذاری) و Compress

2. Dictionary , Value Based = Encode



Encode مبتنی بر Dictionary

1. مورد استفاده داده‌های رشته‌ای

2. ایجاد Global Dictionary

3. Encode کردن مقادیر با داده‌های Dictionary

Original Data		Dmitri	Dmitri	Niko	Victor	Victor	Niko	Niko	Dmitri	Victor
Dictionary	ID	1	2	3						
	Value	Dmitri	Niko	Victor						
Encoded Data		1	1	2	3	3	2	2	1	3

Encode مبتنی بر Value

1. مورد استفاده اعداد Numeric , Integer

2. تبدیل مقادیر عددی به طیف کوچکی از اعداد 64b

3. انجام کار در دو مرحله

Original Data	Numeric	0.8	1.24	1.1	0.25	9.99	4.99	
	Integer	1340	20	2340	3210	220	3300	
Step 1	Numeric	80	124	110	25	999	499	Exponent: E+2 (value * 100) Exponent: E-1 (value / 10)
	Integer	134	2	234	321	22	330	
Step 2	Numeric	55	99	85	0	974	474	Base: 25 (value - 25) Base: 2 (value - 2)
	Integer	132	0	232	319	20	328	

فشرده سازی داده ها ...

1. انجام عملیات فشرده سازی پس از Encode

2. ذخیره داده ها به صورت LOB

	Index	index_id	Partition	Compression	type_desc	total_pages
1	PK_FactSales	1	1	PAGE	IN_ROW_DATA	46265
2	IDX_FactSales_ColumnStore	2	1	COLUMNSTORE	IN_ROW_DATA	0
3	IDX_FactSales_ColumnStore	2	1	COLUMNSTORE	LOB_DATA	16905

فشرده سازی داده ها ...

3. ارائه روش `DATA_COMPRESSION=COLUMNSTORE_ARCHIVE`

4. استفاده از کتابخانه **Xpress 8** (عدم وابستگی به داده)

5. استفاده از الگوریتم **LZ77** برای فشرده سازی

6. مناسب برای پارتیشن های کم مراجعه

HEAP Table (no compression)	HEAP Table (page compression)	Columnstore Compression	Archival Compression
10,504 MB	2,440 MB	831 MB	362 MB

بررسی اجرای Batch Mode

اجرای Batch Mode چیست؟

1. پردازش گروهی از سطرها در یک زمان مشخص

2. اندازه Batch متناسب با CPU Cache

3. کاهش CPU Time

4. افزایش سرعت اجرای کوئری‌ها

Columnstore Index Scan (Clustered)	
Scan a columnstore index, entirely or only a range.	
Physical Operation	Columnstore Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Batch
Estimated Execution Mode	Batch
Storage	ColumnStore
Actual Number of Rows	6599999



اجرای کوئری در حالت Row Mode

1. پیمایش رکوردها به صورت سطر به سطر (اعمال فیلتر)

2. استخراج ستون‌های مورد نیاز (ArticleID, Quantity)

SELECT

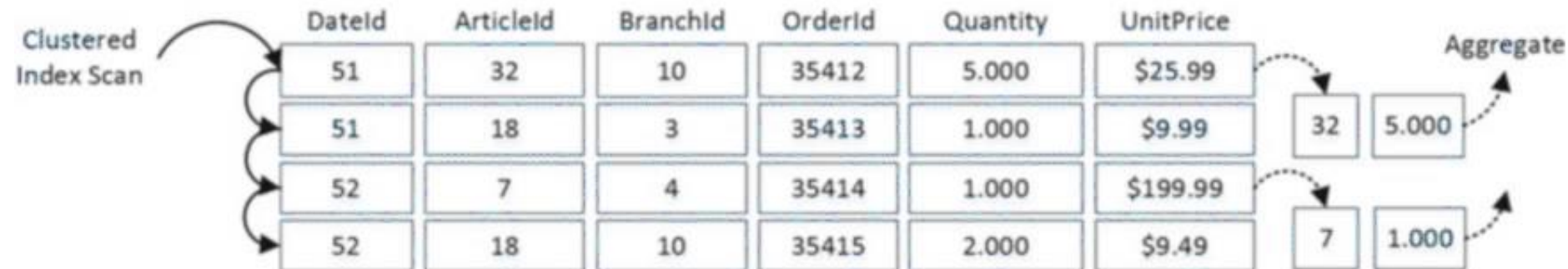
ArticleID, SUM(Quantity)

FROM dbo.FactInternetSales

WHERE UnitPrice >= 10.00

GROUP BY ArticleID

3. اعمال Aggregate



اجرای کوئری در حالت Batch Mode

1. بارگذاری ستون‌های مورد نیاز (فقط Page های 3 فیلد)

2. تنظیم Bitmap داخلی توسط عملگر Filter

3. در نظر گرفتن ردیف Valid

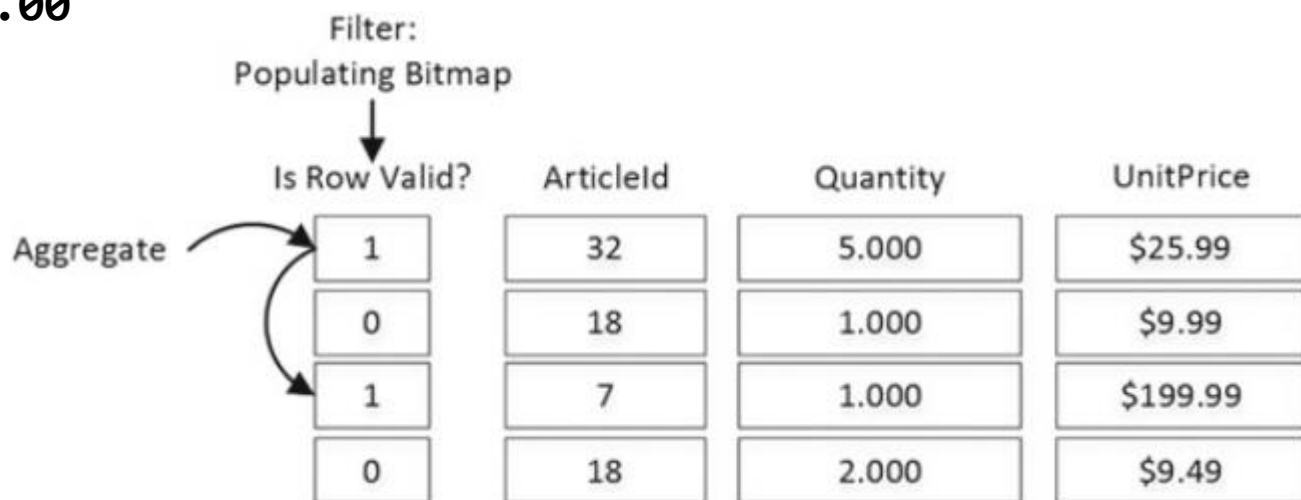
SELECT

ArticleID, SUM(Quantity)

FROM dbo.FactInternetSales

WHERE UnitPrice >= 10.00

GROUP BY ArticleID

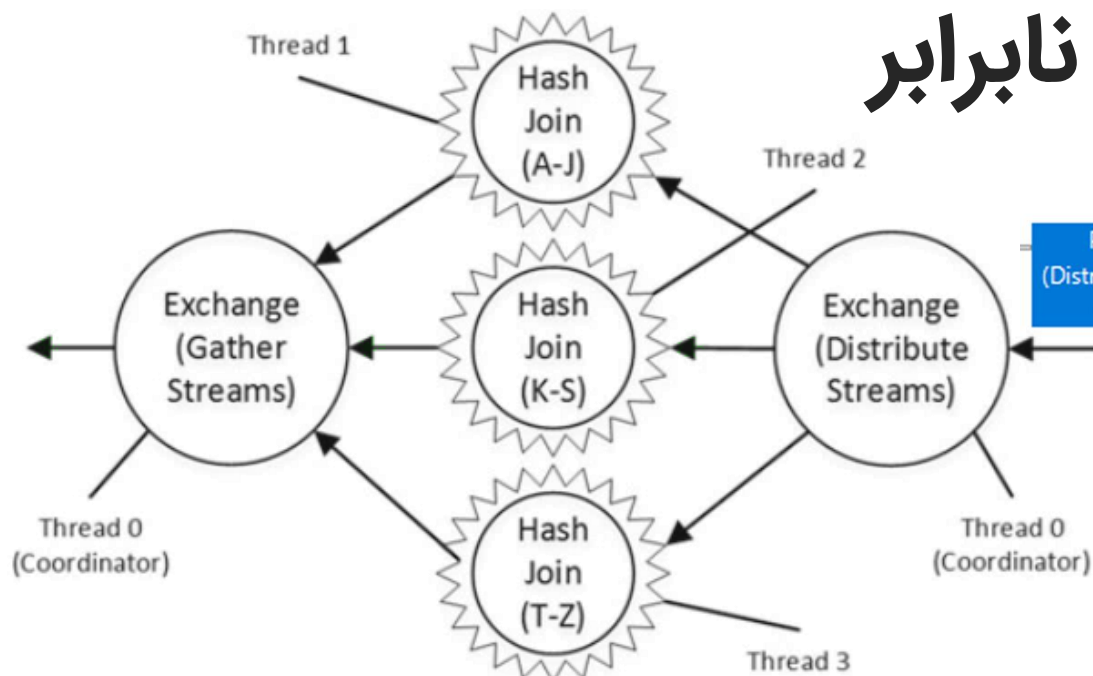


Row Mode Parallelism در حالت Row Mode

1. توزیع داده‌ها میان Thread ها

2. استفاده از انواع الگوریتم‌های توزیع

3. وجود مشکل توزیع نابرابر



Estimated Number of Executions	
Estimated Number of Rows	10.2458
Estimated Row Size	92 B
Actual Rebinds	0
Actual Rewinds	0
Partitioning Type	Broadcast
Node ID	3

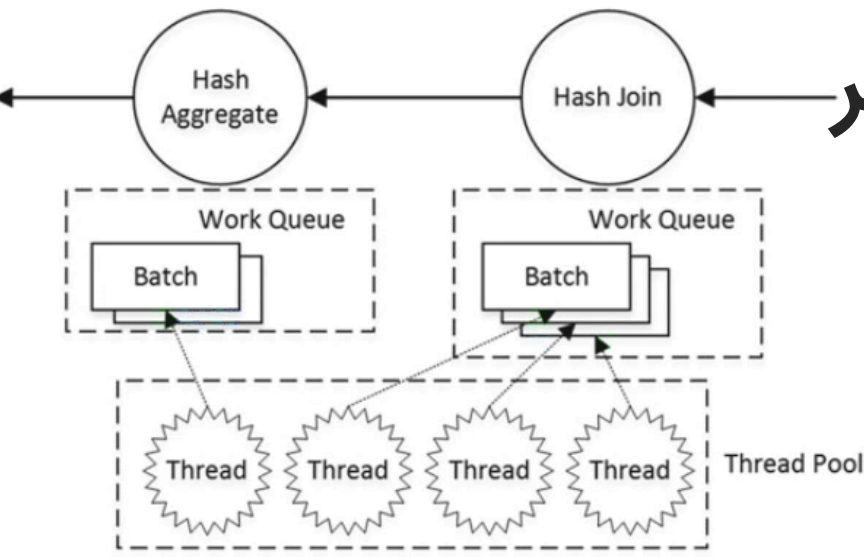
Parallelism در حالت Batch Mode

1. ایجاد صفی از Work Item ها برای هر عملگر

2. Batch = Work Item همان

3. پردازش آیتمها توسط Thread ها

4. عدم وجود مشکل توزیع نابرابر

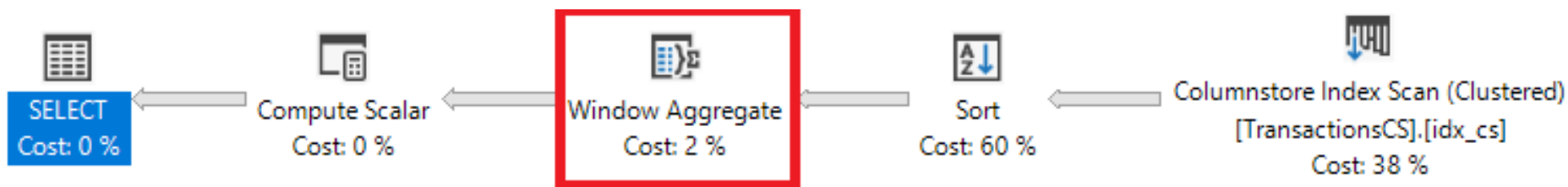


تاثیر Columnstore ایندکس بر روی Window Function

استفاده از Columnstore در WF

1. هدایت کوئری‌ها به سمت Batch Process

2. پردازش سریعتر کوئری‌ها



بررسی ساختار داخلی Columnstore Index

بررسی چند محدودیت ...

1. عدم پشتیبانی از Sparse Column

2. عدم ایجاد به صورت Unique Index

3. عدم پشتیبانی از حالت Index Seek (مرتب نبودن داده ایندکس)

4. عدم پشتیبانی از FileStream

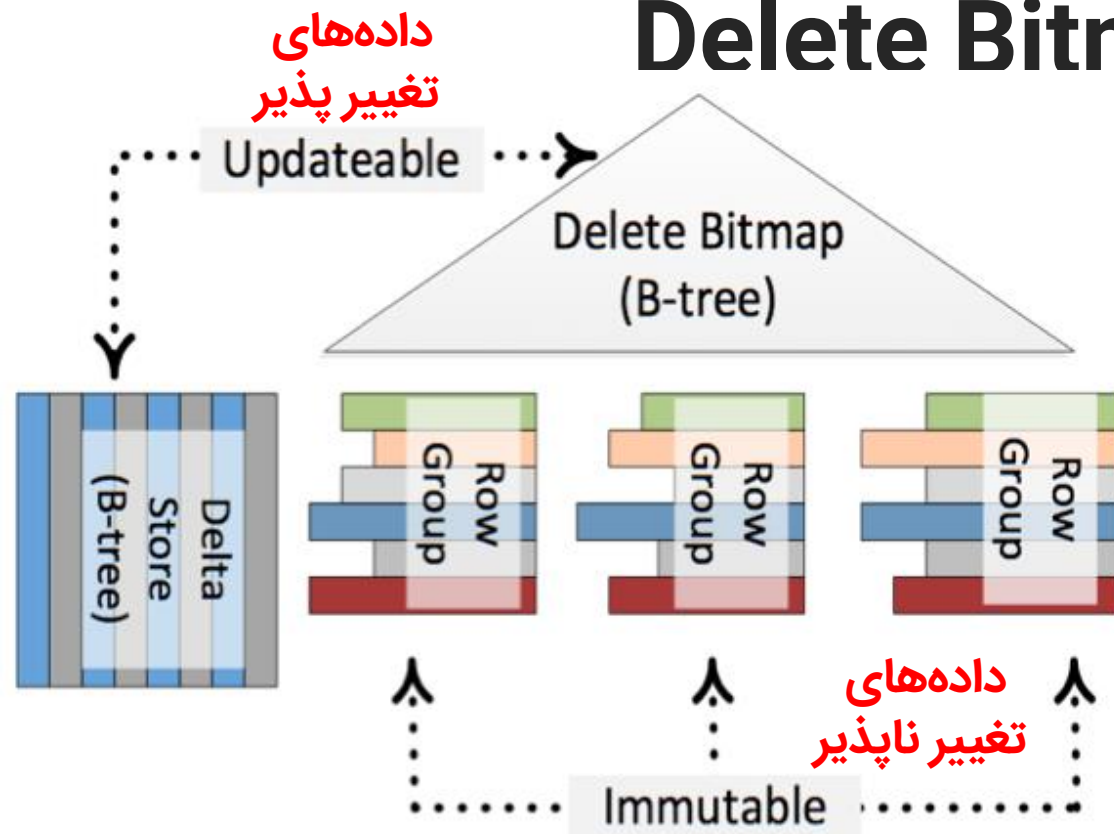
5. حداکثر تعداد ستون ها : 1024 Column



ساختار درونی Columnstore Index

1. دارای قسمت Delta Store

2. دارای قسمت Delete Bitmap



بررسی قسمت Delta Store

1. شامل سطرهای جدید درج شده در ایندکس

2. عملیات Ins, Upd داده جدید در Delta Store

3. تبدیل Delta Store باز به حالت بسته (1048576 Row)

4. استفاده از ساختار B-Tree

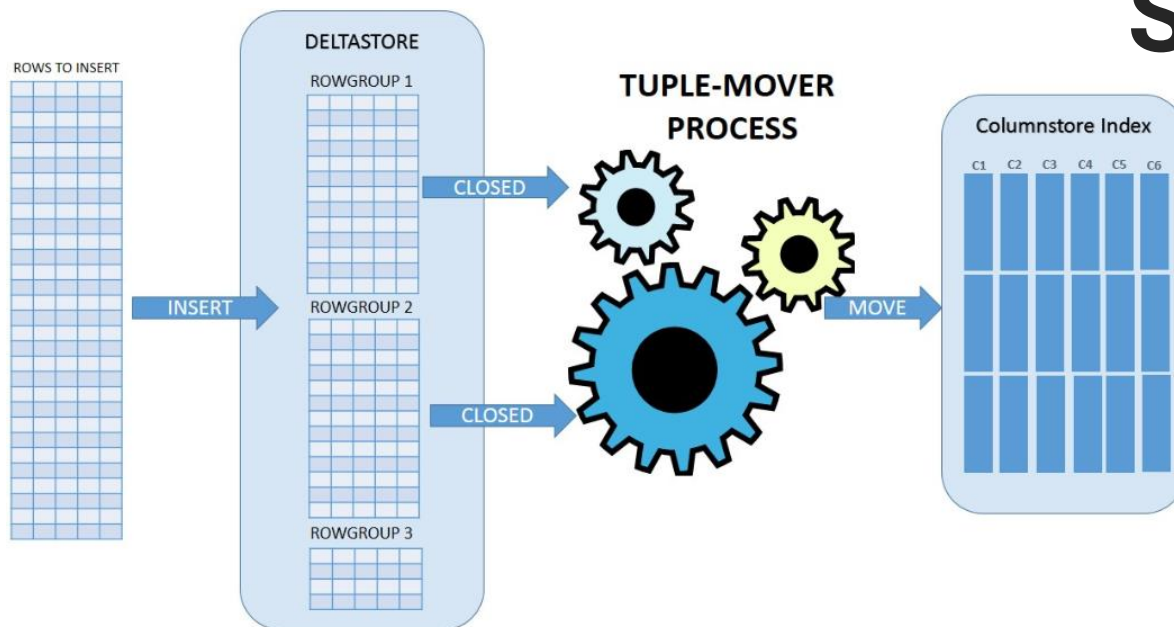
5. ایجاد = مبتنی بر تقاضا (انجام عملیات درج)

بررسی فرآیند Tuple Mover

1. اجرا هر 5 دقیقه یکبار

2. تبدیل Delta Store بسته به ساختار Columnar

3. Single Thread



غیر فعال کردن Tuple Mover
Trace Flag = T634

بررسی قسمت Delete Bitmap

1. شامل سطرهای حذف شده از Row Group

2. عدم حذف ردیف از Row Group (Difference)

3. وجود یک Delete Bitmap به ازای هر پارتیشن

4. استفاده از ساختار B-Tree

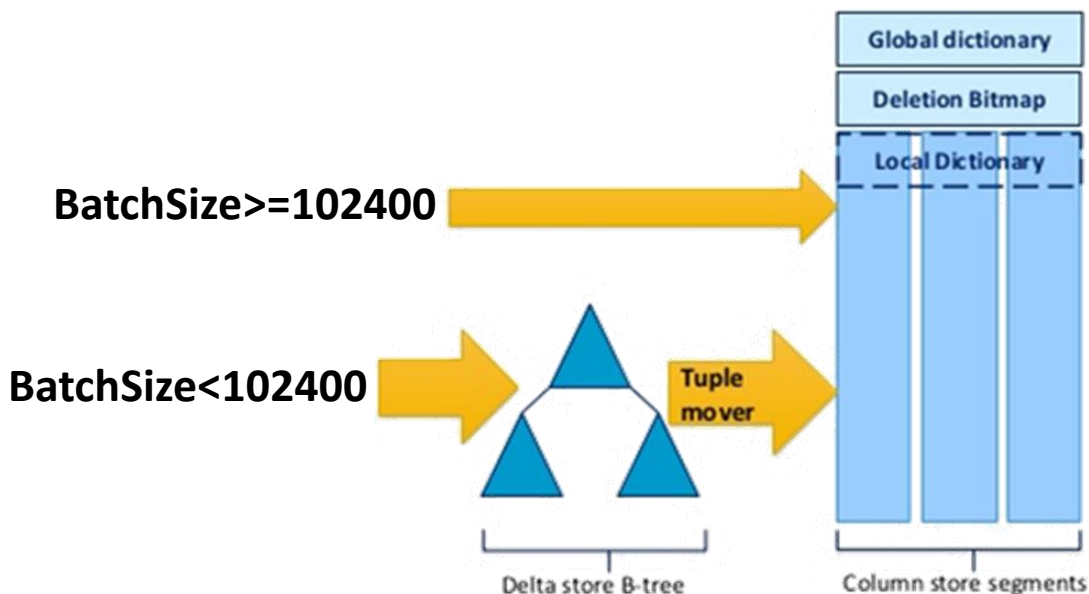
5. ایجاد = مبتنی بر تقاضا (انجام عملیات Delete)

تنظیمات مناسب برای Load Data

1. استفاده از Trickle Insert

1. درج قطره‌ای، درج معمولی، درج سنتی

2. انجام عملیات درج در Delta Store

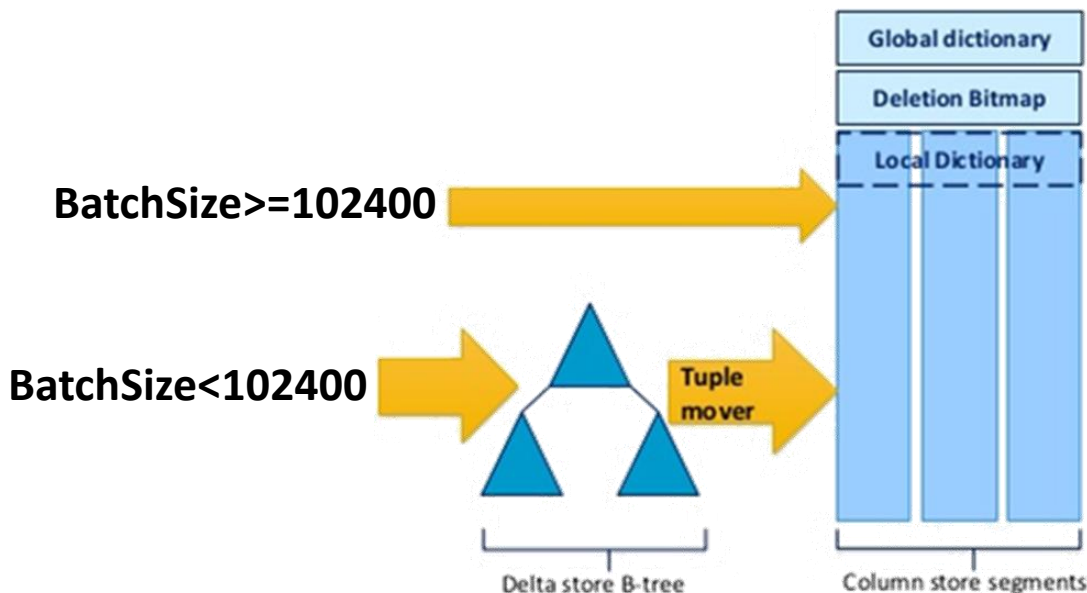


تنظیمات مناسب برای Load Data

1. استفاده از Bulk Insert

1. انجام درج داده‌ها به صورت Bulk (فله‌ای)

2. تنظیم مناسب $BatchSize=1048576$



- 1. ایجاد اتوماتیک Row Group
- 2. عدم درج Data در Delta Store
- 3. حذف پروسه Tuple Mover

تنظیمات مناسب برای Load Data

Batch size	Rows added to row groups (column-based storage)	Rows added to delta store (row-based storage)
99,000	0	99,000
150,000	150,000	0
1,048,577	1,048,576	1
2,100,000	1,048,576; 1,048,576	2,848
2,250,000	1,048,576; 1,048,576; 152,848	0

1. اگر $\text{BatchSize} \geq 102400$ باشد درج دیتا در Row Group

2. اگر $\text{BatchSize} < 102400$ باشد درج دیتا در Delta Store

3. اگر $\text{BatchSize} \geq 1048576$ باشد هر 1048576 ردیف در یک RowGroup درج می شود.

بهتر است Row Group با اندازه های کوچک نداشته باشیم. (مشکل : سگمنت های نیمه پر)

نگهداری Columnstore Index

بررسی فرآیند Rebuild

1. انجام عملیات حذف فیزیکی از Row Group

2. ادغام Row Group و Delta Store

3. بازسازی کلیه سگمنت‌های Row Group ها

4. بازسازی به صورت Offline (SQL Server 2016 و قبل‌تر از آن)



بررسی فرآیند Reorganize

1. سازماندهی ایندکس به صورت Online
2. انجام پروسه Tuple Mover
3. انجام برخی از بهینه سازی در نسخه 2016

