Module 4

Columnstore Indexes

Zyead Ahmed.
Aspiring To Learning Data Engineer.

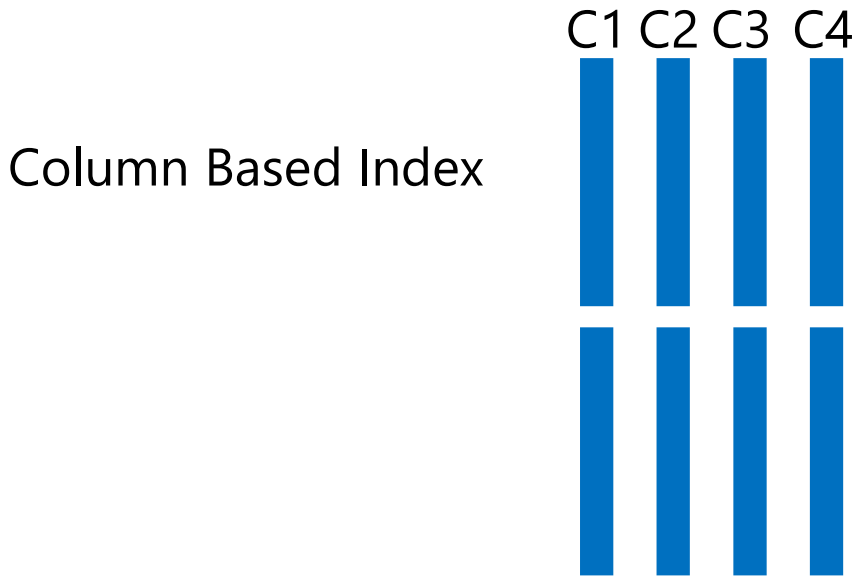
Module Overview

- Introduction to Columnstore Indexes
- Creating Columnstore Indexes
- Working with Columnstore Indexes

Lesson 1: Introduction to Columnstore Indexes

- What Are Columnstore Indexes?
- Nonclustered Columnstore Indexes
- Clustered Columnstore Indexes
- Demonstration: The Benefits of Using Columnstore Indexes

What Are Columnstore Indexes?



Pages only contain one column from many rows

Nonclustered Columnstore Indexes

Characteristics

Contains some or all columns

Used in combination with rowstore tables

Updatable

Can be filtered

Uses more space than just a rowstore

Clustered Columnstore Indexes

Characteristics
Must contain all columns
Optimizes data for storage and performance
Row based indexes can be added on top
Cannot be filtered

Demonstration: The Benefits of Using Columnstore Indexes

In this demonstration, you will see how to create a columnstore index

Lesson 2: Creating Columnstore Indexes

- Creating a Nonclustered Columnstore Index
- Creating a Clustered Columnstore Index
- Creating a Clustered Columnstore Table with Primary and Foreign Keys
- Demonstration: Creating Columnstore Indexes Using SQL Server Management Studio

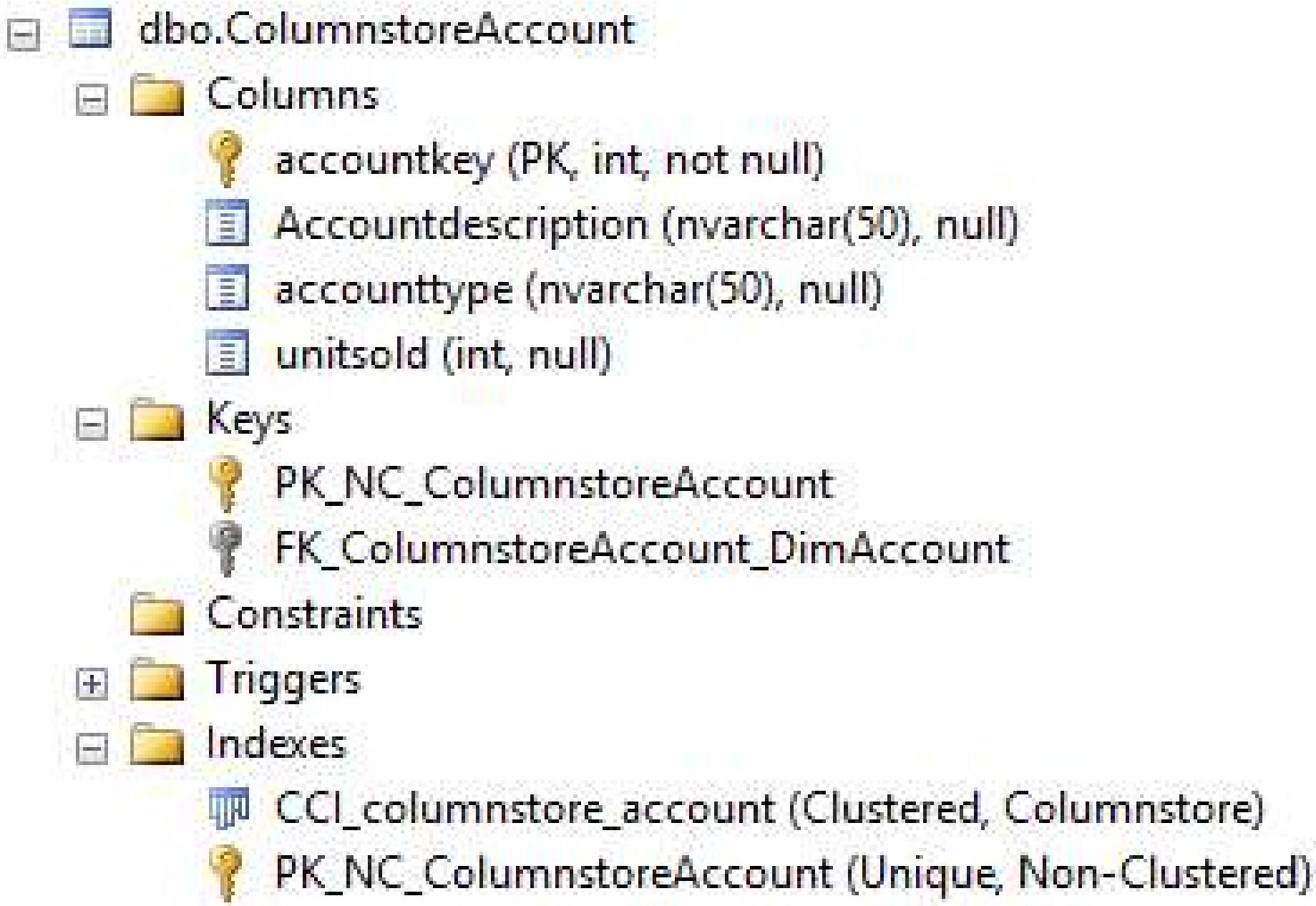
Creating a Nonclustered Columnstore Index

```
CREATE NONCLUSTERED COLUMNSTORE INDEX NCCSIX_FactInternetSales
ON FactInternetSales (
        CustomerKey
        ,SalesPersonKey
        ,ProductKey
        ,OrderDateKey,
        ,OrderNo
        ,ItemNo
        ,Quantity
        ,Cost
        ,Discount
);
```

Creating a Clustered Columnstore Index

```
CREATE CLUSTERED COLUMNSTORE INDEX CCSIX_FactSalesOrderDetails
ON FactSalesOrderDetails;
```

Creating a Clustered Columnstore Table with Primary and Foreign Keys



Demonstration: Creating Columnstore Indexes Using SQL Server Management Studio

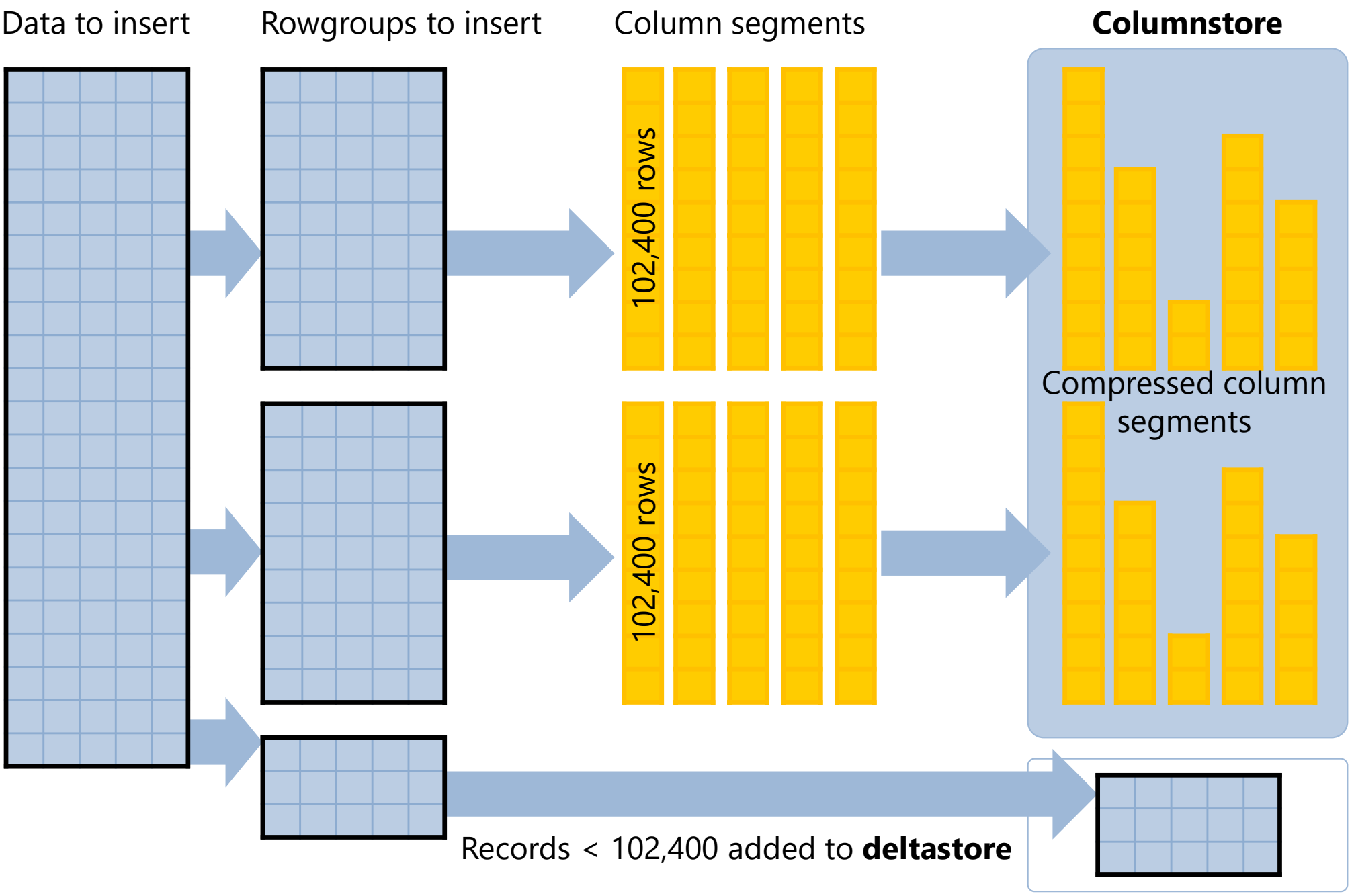
In this demonstration, you will see how to:

- Create a nonclustered columnstore index using SSMS
- Create a clustered columnstore index using SSMS

Lesson 3: Working with Columnstore Indexes

- Managing Columnstore Indexes
- Index Fragmentation
- Columnstore Indexes and Memory Optimized Tables

Managing Columnstore Indexes



Index Fragmentation

Use **sys.dm_db_index_physical_stats** to determine index fragmentation

Reorganize	vs.	Rebuild
Fragmentation Between **5%** and **30%**		Fragmentation Greater than **30%**

Columnstore Indexes and Memory Optimized Tables

In-memory columnstore tables

- The index has to be declared at runtime
- Tables can be up to 2 TB in size
- Can be combined with rowstore index
- Enable real-time operational analytics

Memory Optimization Advisor can be used to support moving a table from being disk-based to being memory-optimized.