解锁SQL查询性能

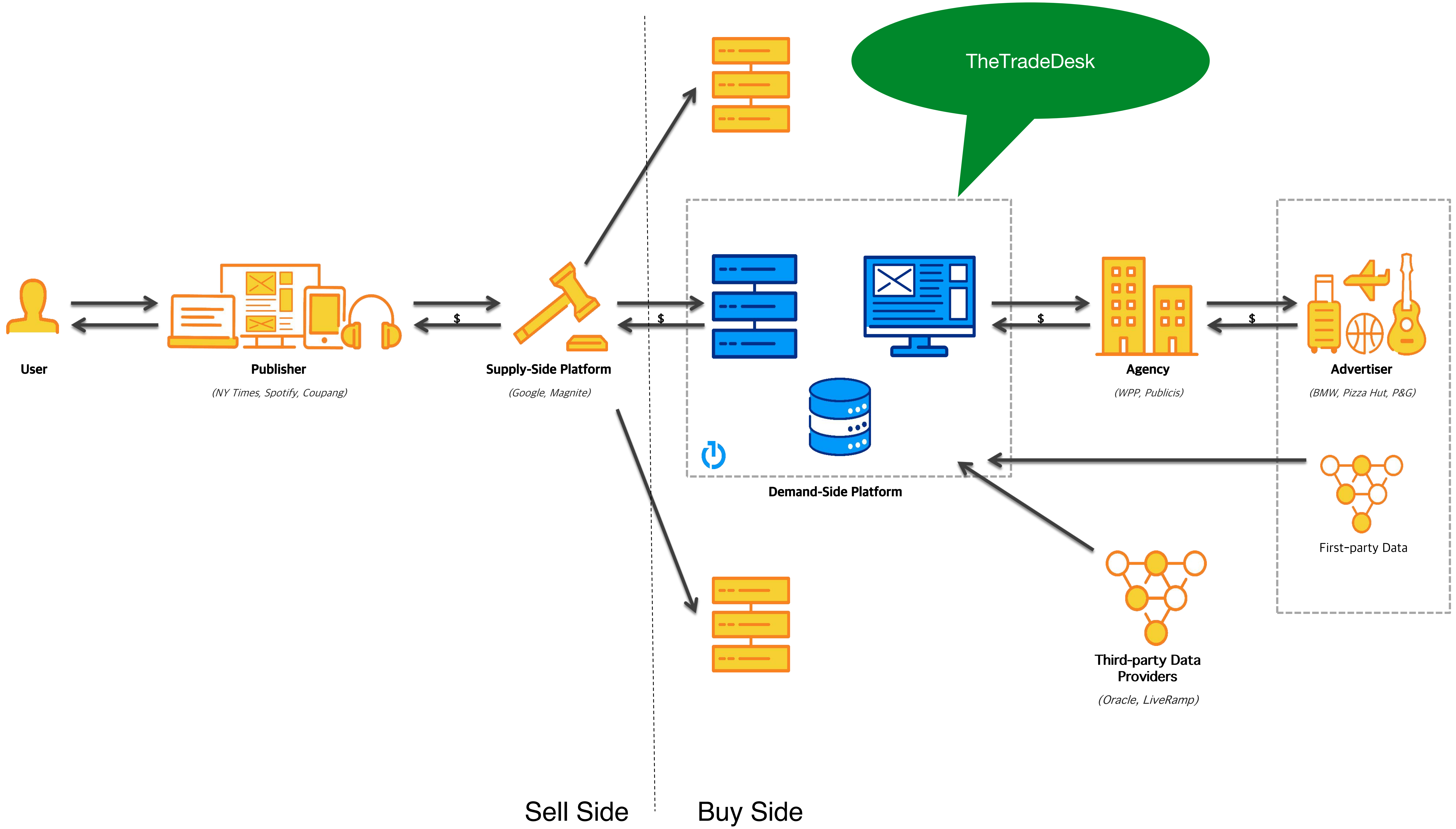
TheTradeDesk / Lisa Li

Self introduction

Lisa Li

- Ever worked in IBM, Intel: storage, distributed system
- Joined in TheTradeDesk in 2021, focus on Retail related business.

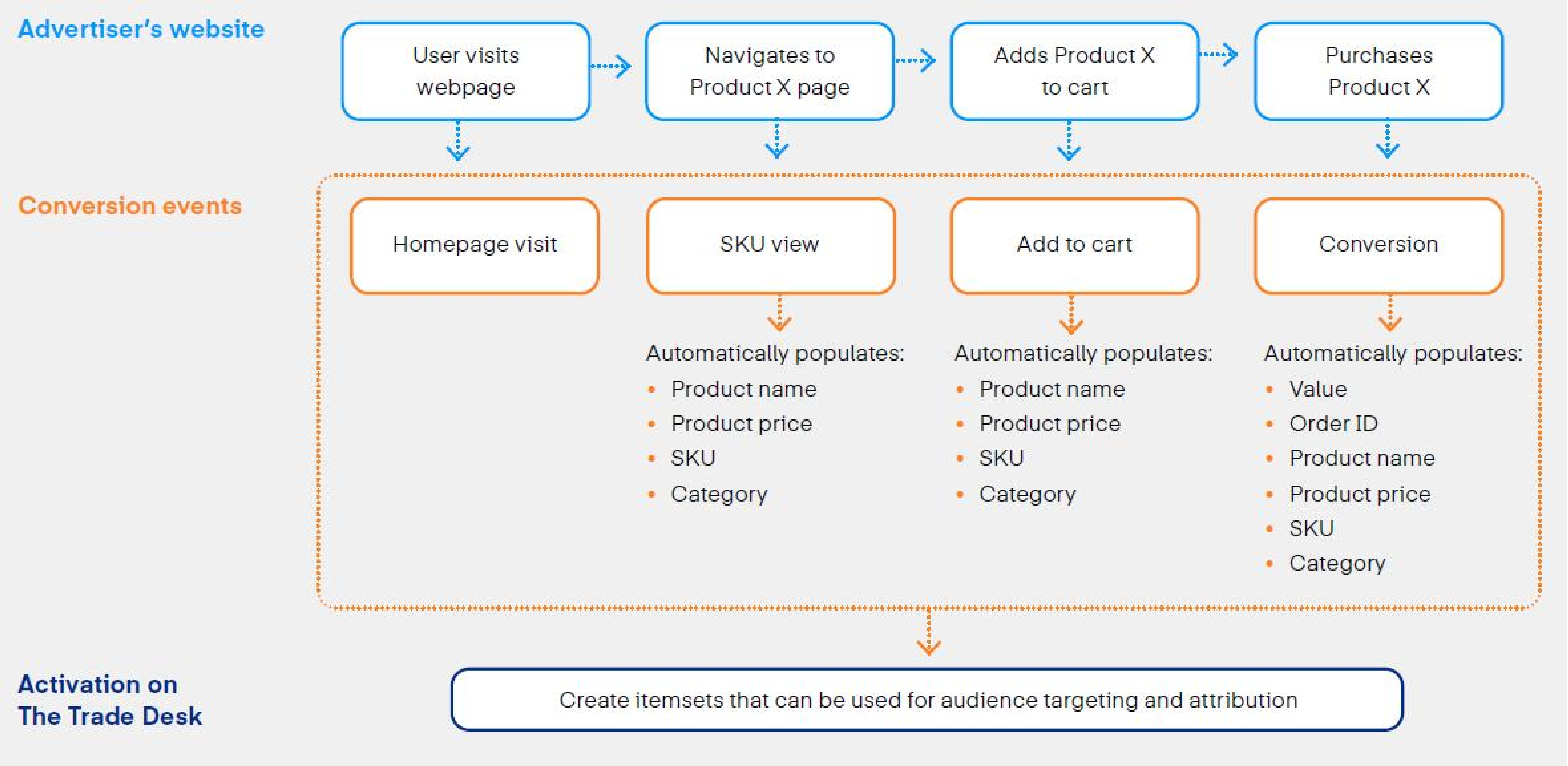
Programmatic advertising



Agenda

- Why investigate the topic
- Understanding Indexes and execution plans
- Optimizing with examples

Item-level events for targeting and measurement



Advertiser's website

User visits webpage → Navigates to Product X page → Adds Product X to cart → Purchases Product X

Conversion events

Homepage visit

SKU view

Automatically populates:

- Product name
- Product price
- SKU
- Category

Add to cart

Automatically populates:

- Product name
- Product price
- SKU
- Category

Conversion

Automatically populates:

- Value
- Order ID
- Product name
- Product price
- SKU
- Category

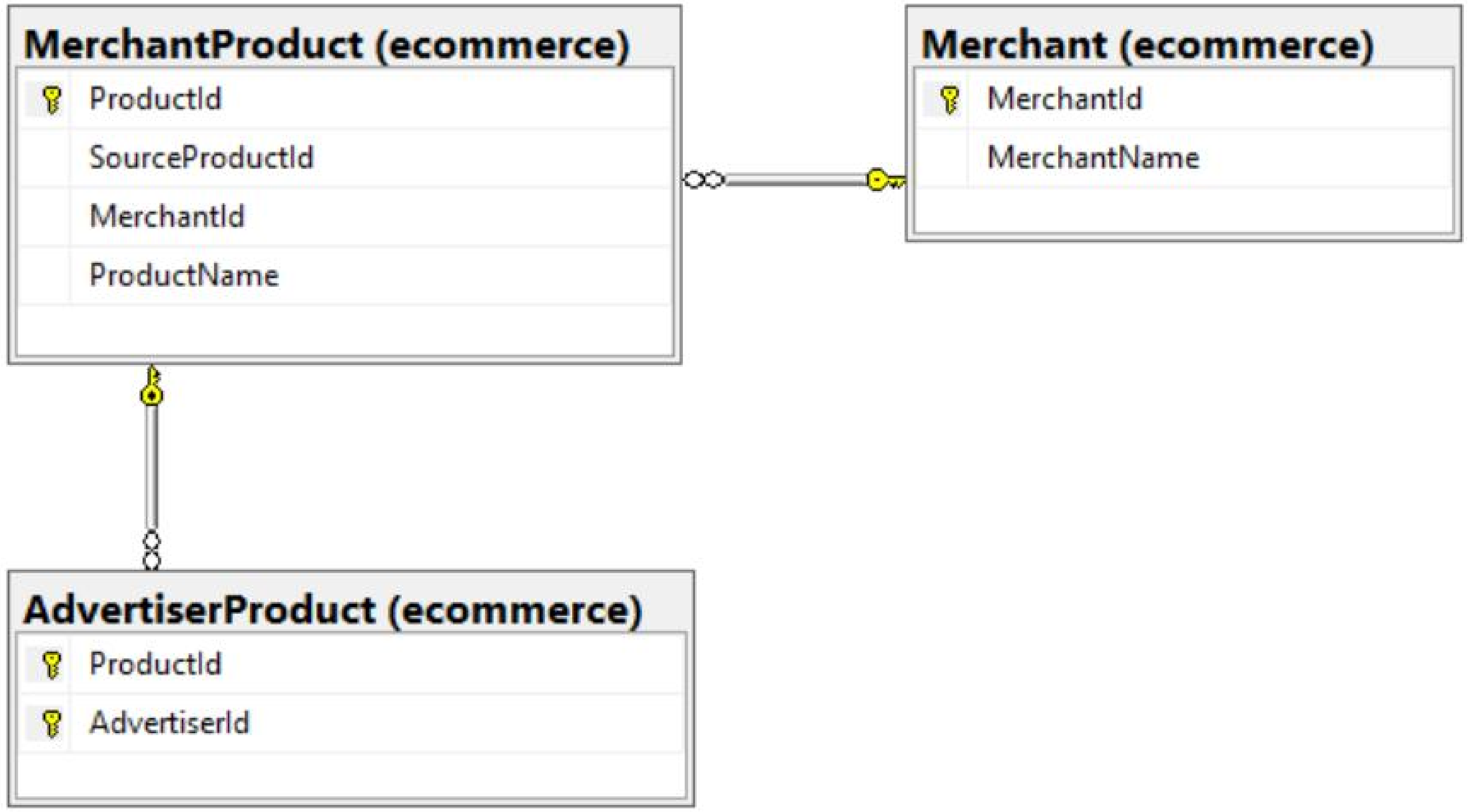
Activation on The Trade Desk

Create itemsets that can be used for audience targeting and attribution

Item-level events for targeting and measurement

- Tables:
  - Merchant
  - MerchantProduct
  - AdvertiserProduct

ProductId	SourceProductId	MerchantId	ProductName
11911	85693	1	Shampoo
11912	MX98675	2	Conditioner



MerchantProduct (ecommerce)

- ProductId
- SourceProductId
- MerchantId
- ProductName

Merchant (ecommerce)

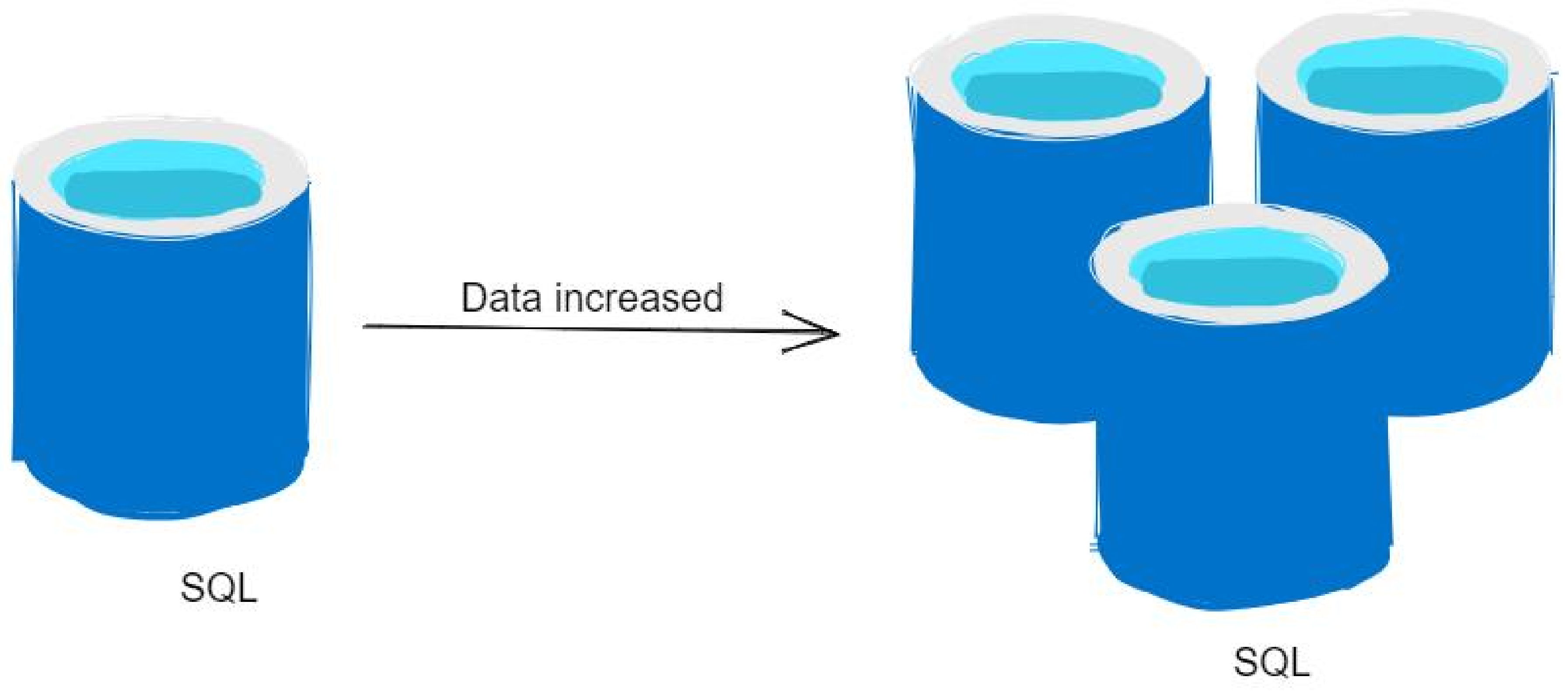
- MerchantId
- MerchantName

AdvertiserProduct (ecommerce)

- ProductId
- AdvertiserId

Problem

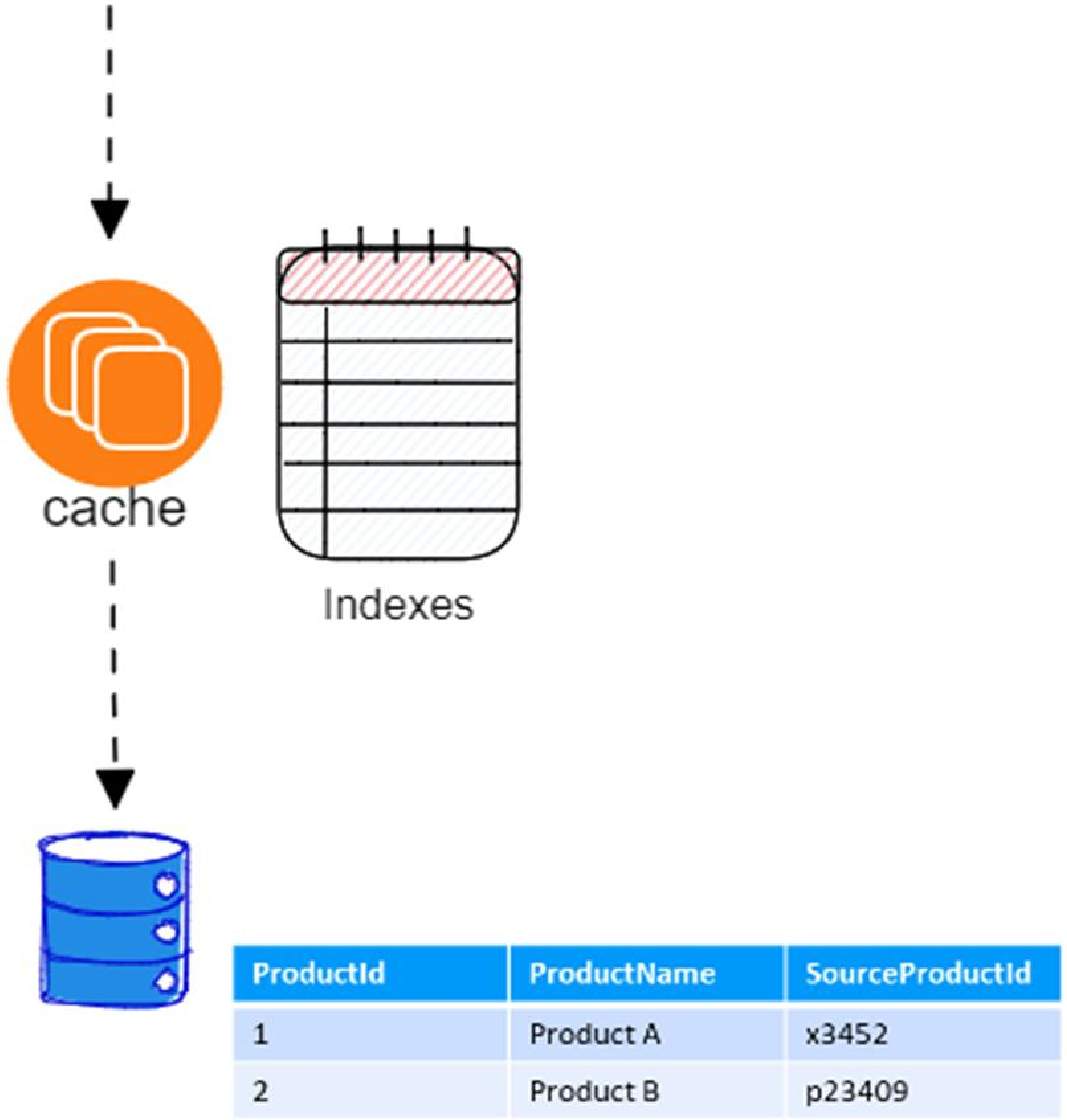
- Use SQL server to manage metadata
- Data increased 10x times
- Timeout may happen
- Understand how query works and Optimize



	ProductCount
Before	1,901,935
After	24,686,272

Indexes

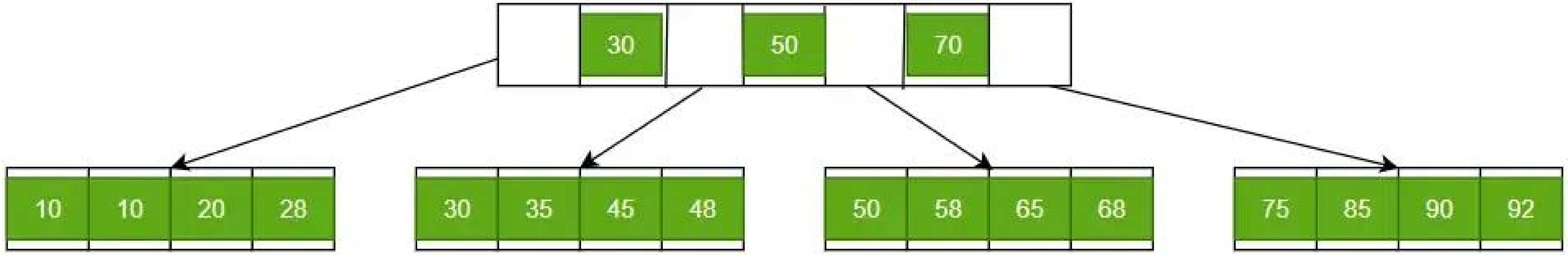
- What are indexes?
  - ➢ Contain sorted data of the columns in the index definition
  - ➢ On-disk structure associated with a table or view
  - ➢ Contains keys built from one or more columns in a table or view
- Why use indexes?
  - ➢ Speed up the retrieval of data
  - ➢ Reduce disk IOs
- Impact
  - ➢ Data updates and deletes



ProductId	ProductName	SourceProductId
1	Product A	x3452
2	Product B	p23409

B-Tree Data Structure

- Self-balanced tree data structure that is a generalized form of the Binary Search Tree (BST).
  - ➢Each node contains at most a fixed number of keys
  - ➢All leaf nodes are at the same level.
- Time complexity: O(logn), where *n* is the number of keys stored in the tree



From https://www.geeksforgeeks.org/introduction-of-b-tree/

Indexes

Clustered Index

- Sort data rows in the table itself
- Only one clustered index is allowed in a table
- A table without a clustered index is called HEAP

Non-Clustered Index

- A structure separate from the data rows
- Could be multiple index keys
- Each key points back to the data row

Single Index

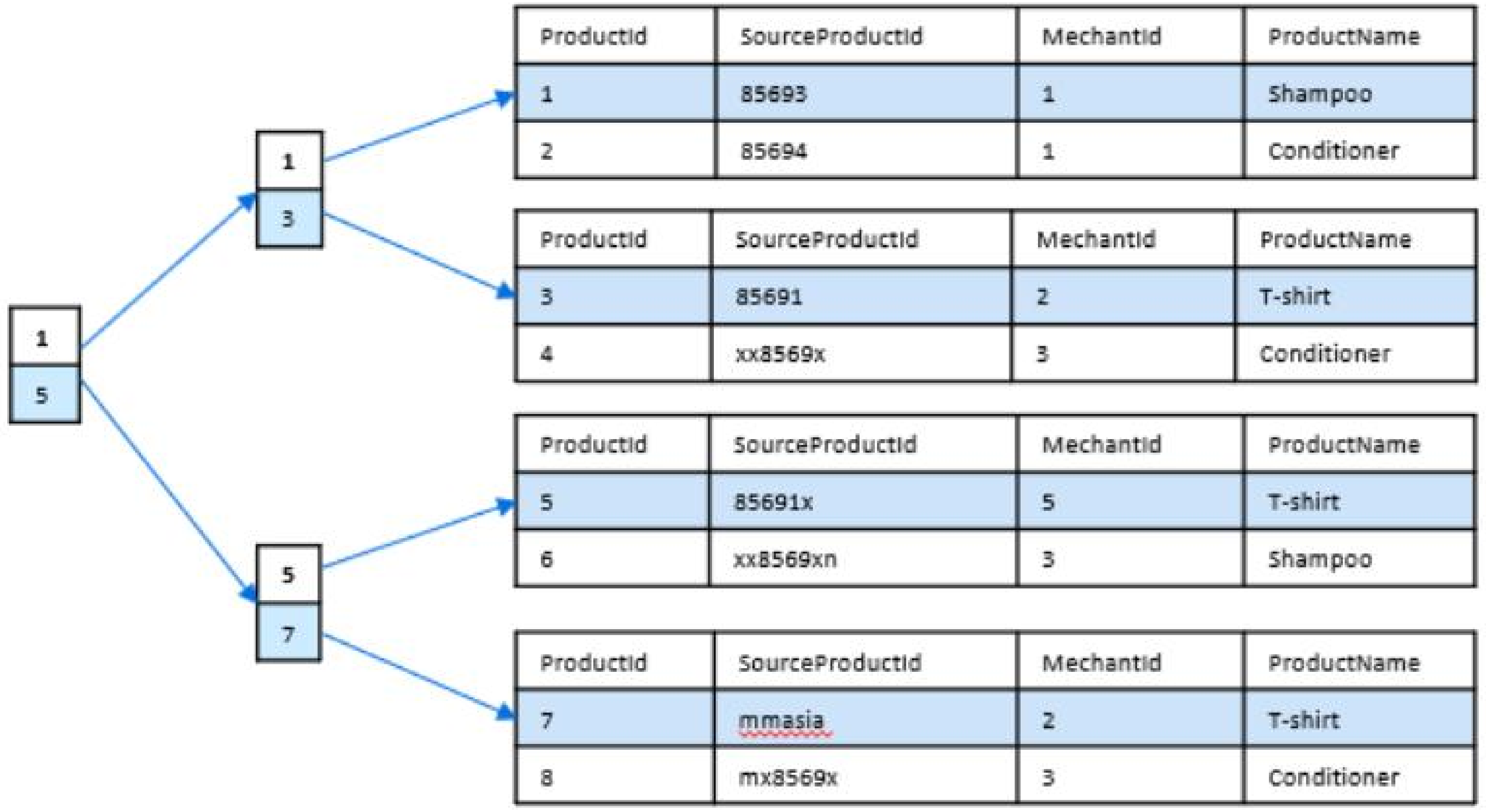
- Single-Column Indexes

Composite Index

- On two or more columns of a table

Clustered & non–clustered Index

Clustered



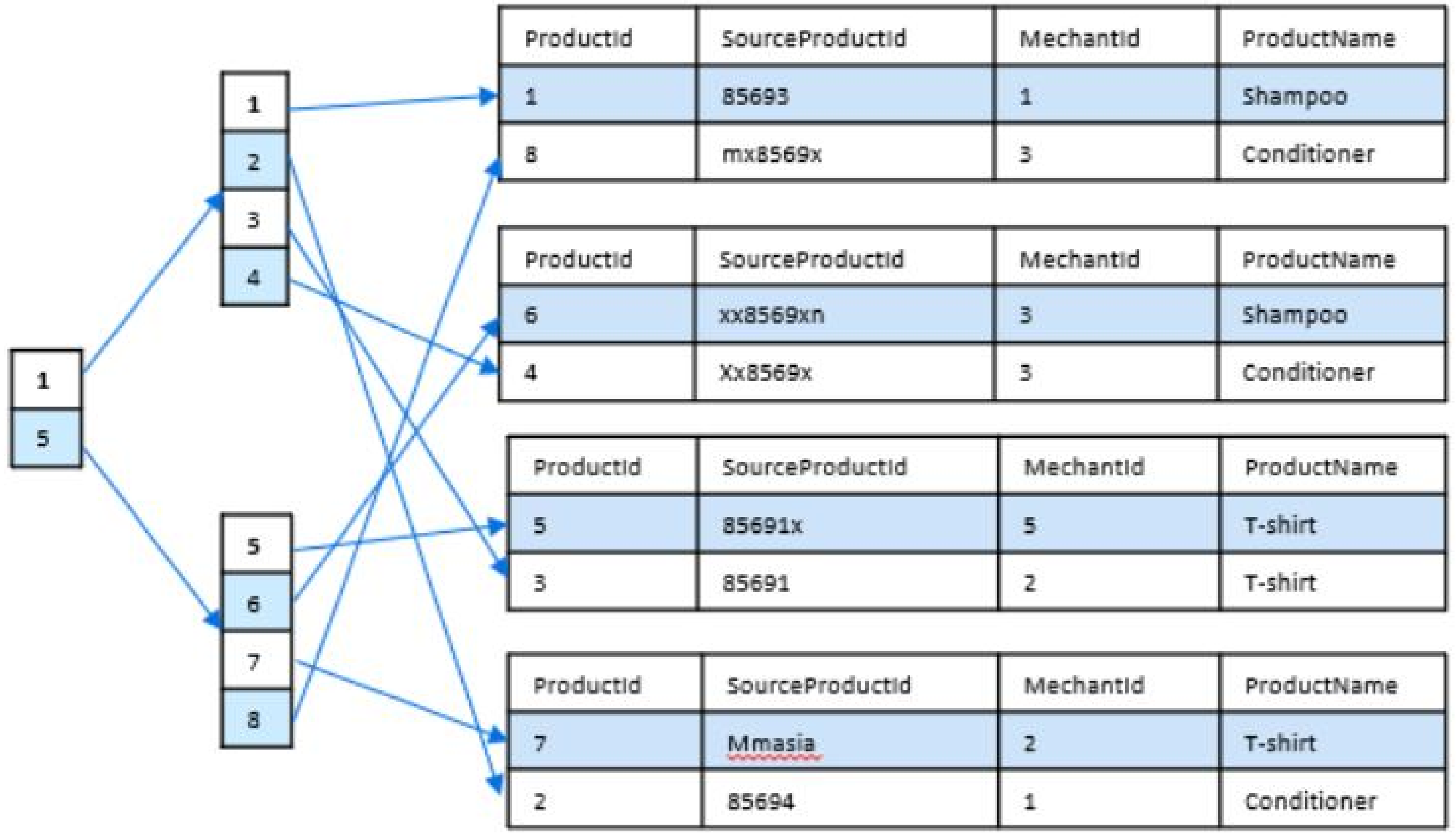
ProductId	SourceProductId	MechantId	ProductName
1	85693	1	Shampoo
2	85694	1	Conditioner

ProductId	SourceProductId	MechantId	ProductName
3	85691	2	T-shirt
4	xx8569x	3	Conditioner

ProductId	SourceProductId	MechantId	ProductName
5	85691x	5	T-shirt
6	xx8569xn	3	Shampoo

ProductId	SourceProductId	MechantId	ProductName
7	mmasia	2	T-shirt
8	mx8569x	3	Conditioner

Non–clustered



ProductId	SourceProductId	MechantId	ProductName
1	85693	1	Shampoo
8	mx8569x	3	Conditioner

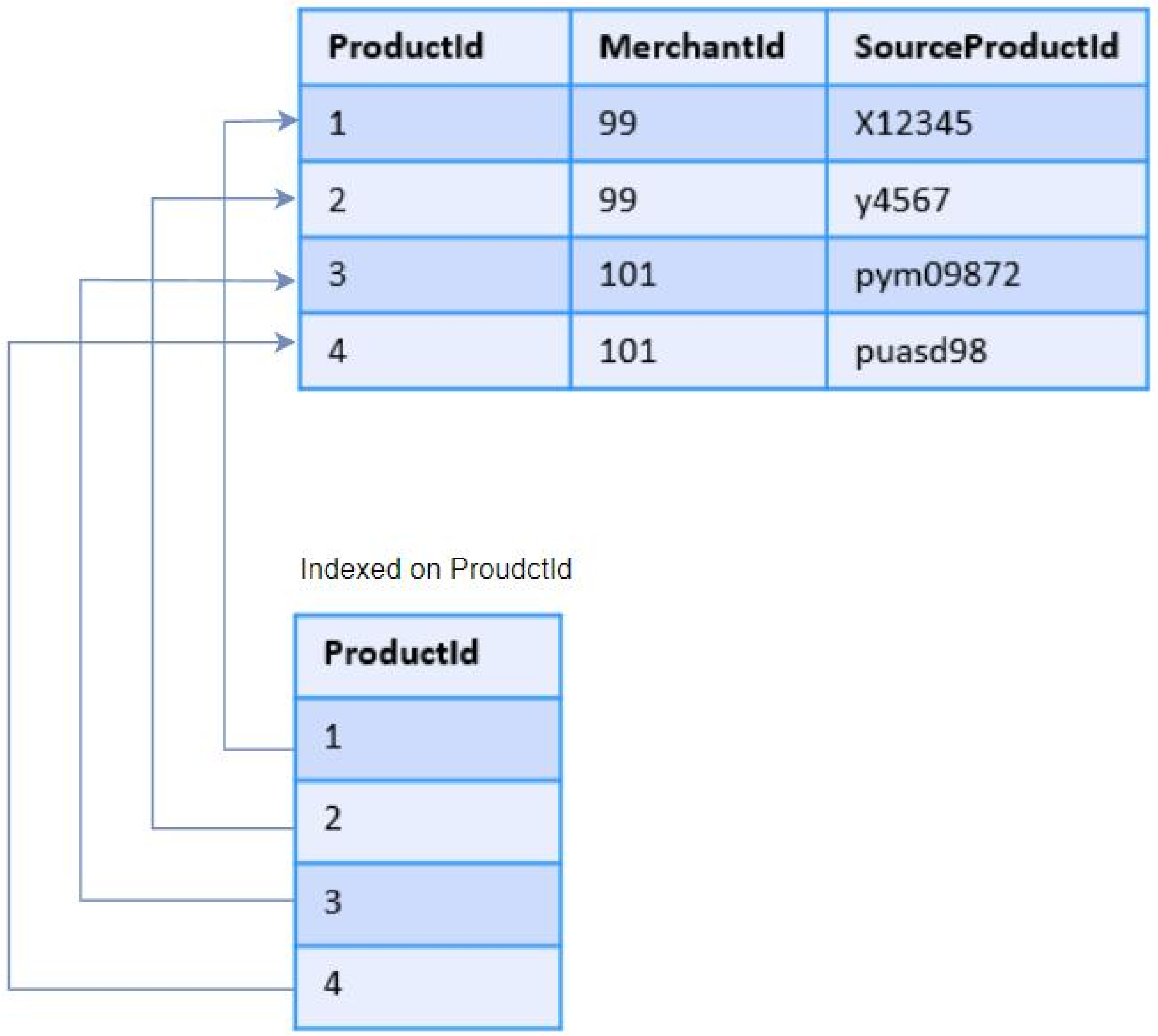
ProductId	SourceProductId	MechantId	ProductName
6	xx8569xn	3	Shampoo
4	Xx8569x	3	Conditioner

ProductId	SourceProductId	MechantId	ProductName
5	85691x	5	T-shirt
3	85691	2	T-shirt

ProductId	SourceProductId	MechantId	ProductName
7	Mmasia	2	T-shirt
2	85694	1	Conditioner

Single/Composite Indexes

Single

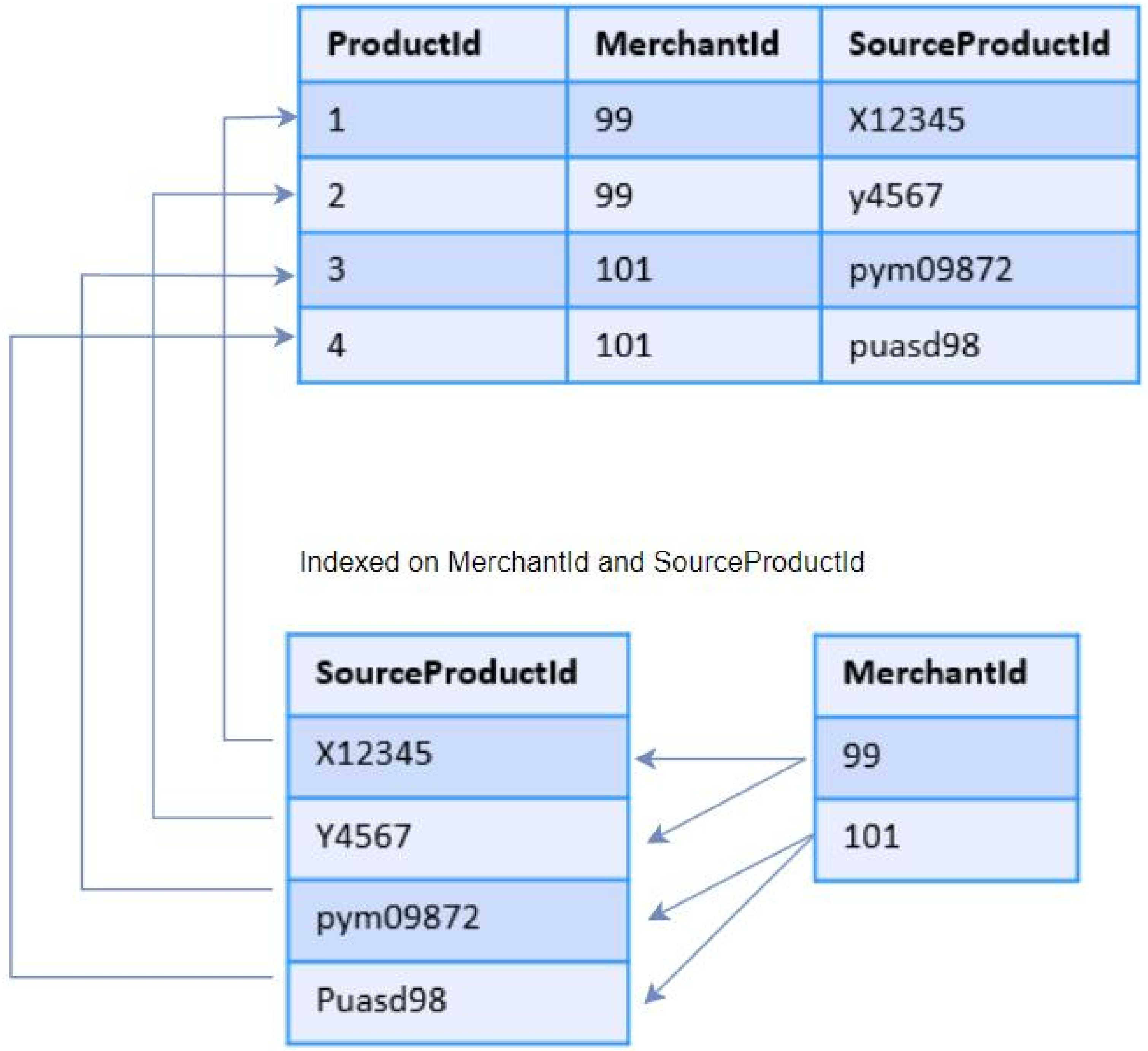


ProductId	MerchantId	SourceProductId
1	99	X12345
2	99	y4567
3	101	pym09872
4	101	puasd98

Indexed on ProudctId

ProductId
1
2
3
4

Composite



ProductId	MerchantId	SourceProductId
1	99	X12345
2	99	y4567
3	101	pym09872
4	101	puasd98

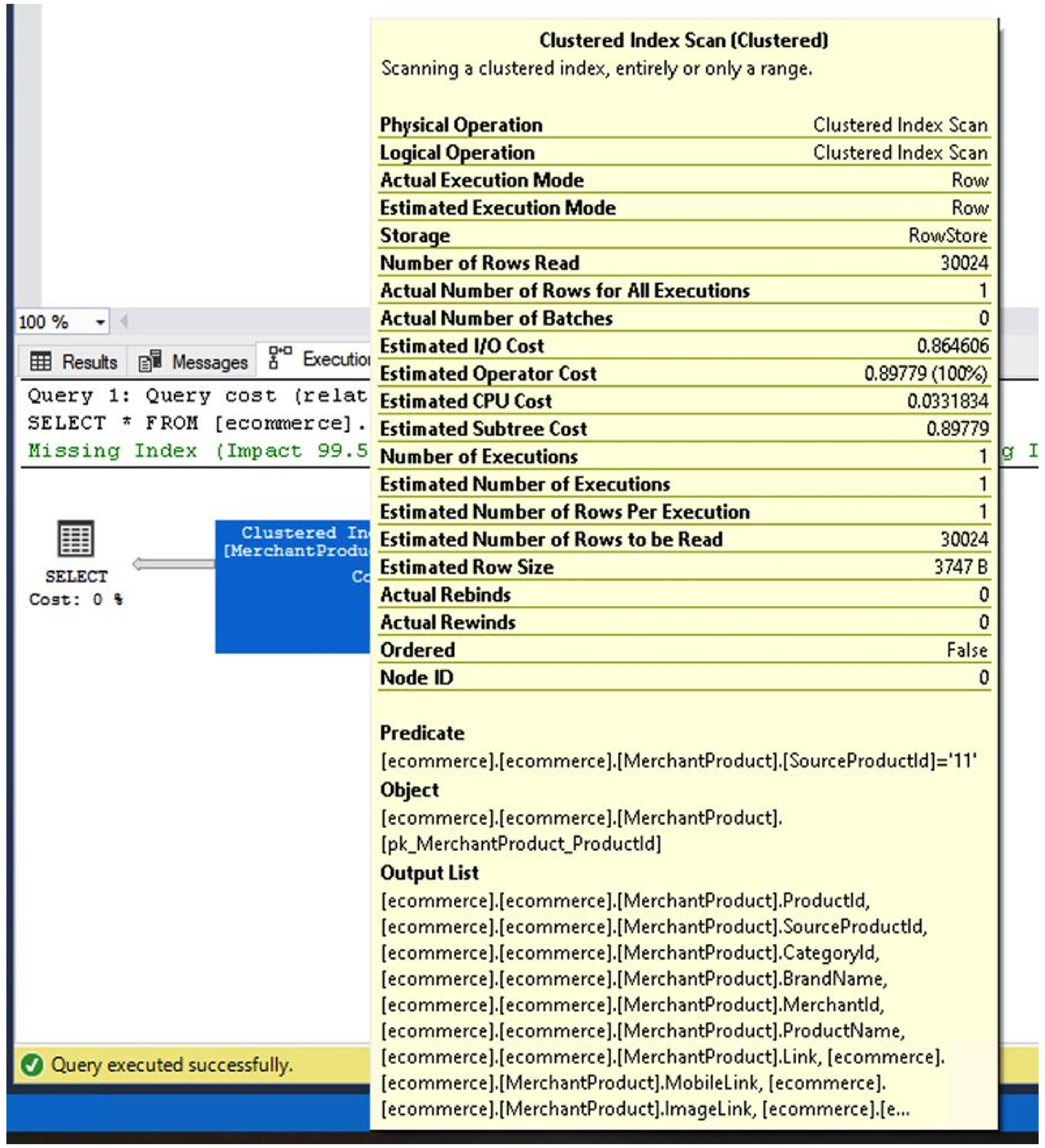
Indexed on MerchantId and SourceProductId

SourceProductId
X12345
Y4567
pym09872
Puasd98

MerchantId
99
101

Execution plans

- The sequence of steps that the database executes a query
- Example
  - ➢ Table: Product (ProductId, MerchantId, SourceProductId)
- Basic Metrics
  - ➢ Operation
  - ➢ Number of Rows Read
  - ➢ Predicate
  - ➢ Object
  - ➢ Output List



```
Query 1: Query cost (relat
SELECT * FROM [ecommerce].
Missing Index (Impact 99.5
```

Query executed successfully.

Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	30024
Actual Number of Rows for All Executions	1
Actual Number of Batches	0
Estimated I/O Cost	0.864606
Estimated Operator Cost	0.89779 (100%)
Estimated CPU Cost	0.0331834
Estimated Subtree Cost	0.89779
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	30024
Estimated Row Size	3747 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	0

Predicate

[ecommerce].[ecommerce].[MerchantProduct].[SourceProductId]='11'

Object

[ecommerce].[ecommerce].[MerchantProduct].[pk_MerchantProduct_ProductId]

Output List

[ecommerce].[ecommerce].[MerchantProduct].ProductId, [ecommerce].[ecommerce].[MerchantProduct].SourceProductId, [ecommerce].[ecommerce].[MerchantProduct].CategoryId, [ecommerce].[ecommerce].[MerchantProduct].BrandName, [ecommerce].[ecommerce].[MerchantProduct].MerchantId, [ecommerce].[ecommerce].[MerchantProduct].ProductName, [ecommerce].[ecommerce].[MerchantProduct].Link, [ecommerce].[ecommerce].[MerchantProduct].MobileLink, [ecommerce].[ecommerce].[MerchantProduct].ImageLink, [ecommerce].[e...

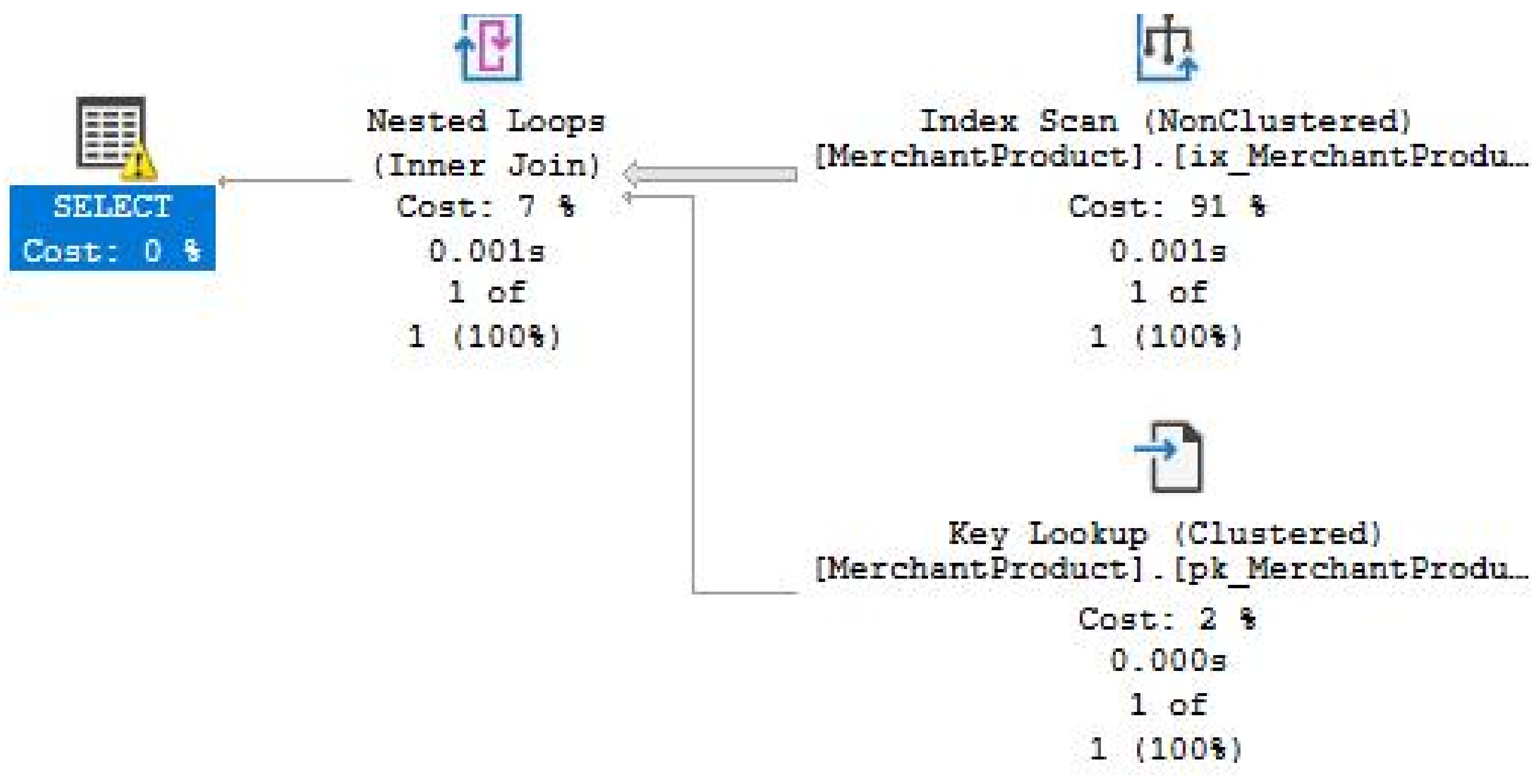
Execution Plan – cont.

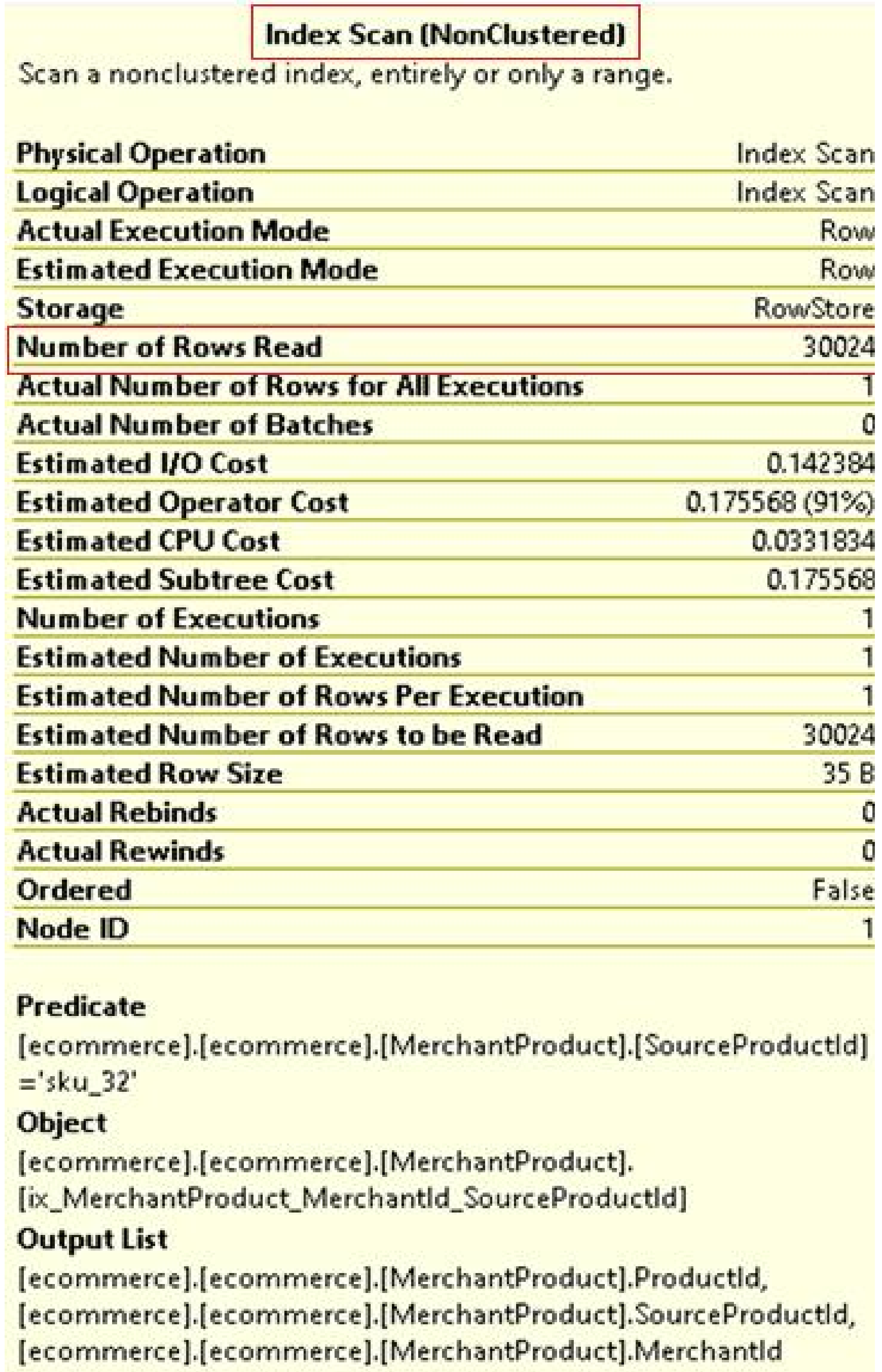
- Index seek
  - ➢ The Query Optimizer uses the b–tree (non–left level) solely for navigation to the leaf level with the actual data.
- Index scan
  - ➢ SQL Server reads all rows in a table, and then returns only those rows that satisfy the search criteria.
- Key lookup
  - ➢ The key lookup operator occurs when the query optimizer performs an index seek on a non–clustered index, and that index does not have all the columns needed to fulfill the result set so it has to go back to the table to retrieve those additional columns.
- Performance comparison
  - ➢ Index seek > Index scan > key lookup

Use Indexes to accelerate

- With/without indexes

- select * from ecommerce.MerchantProduct where MerchantId = 99 and SourceProductId = 'sku_32' and IsDeleted = 0;



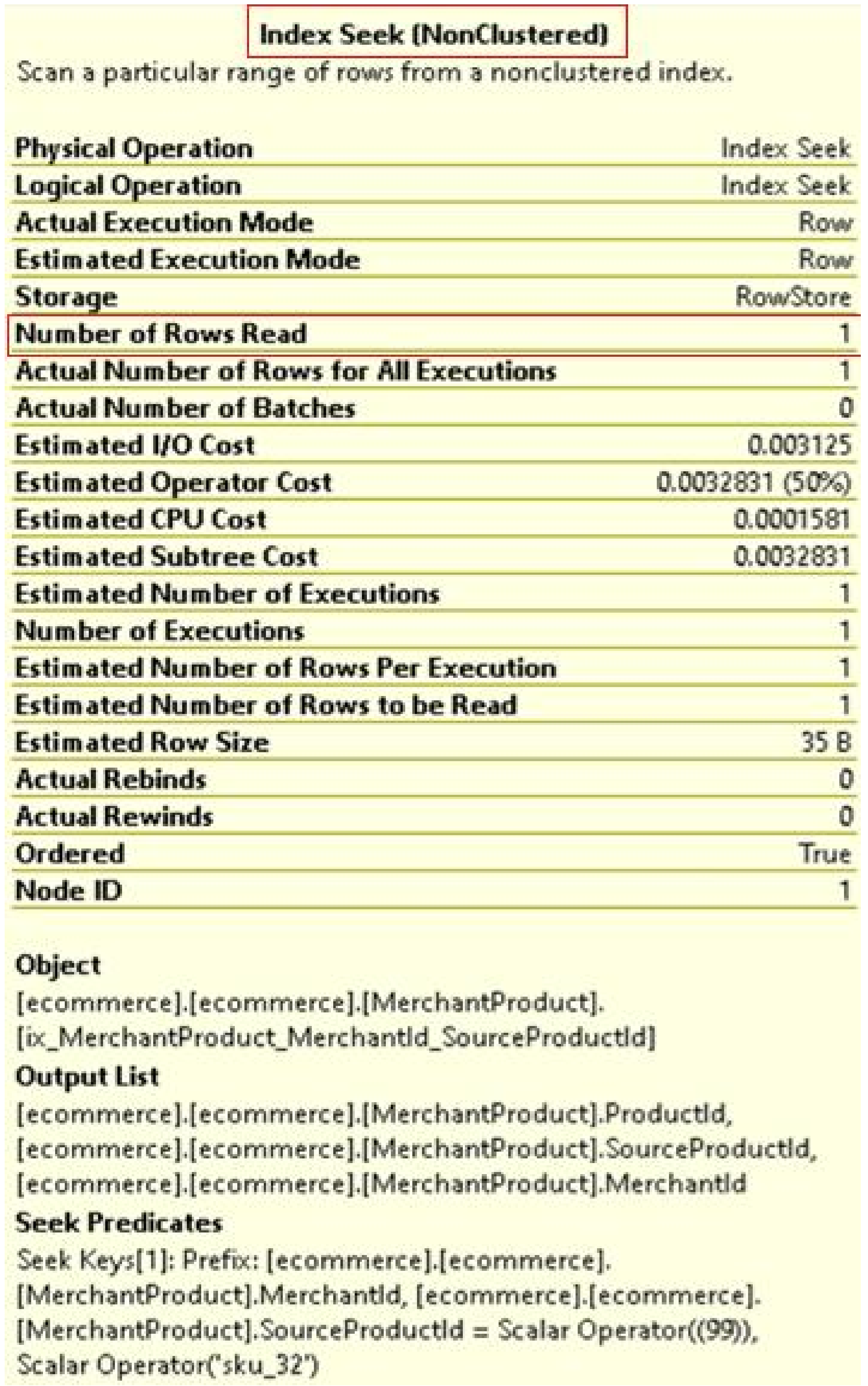


Index Scan (NonClustered)

Scan a nonclustered index, entirely or only a range.

Physical Operation	Index Scan
Logical Operation	Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	30024
Actual Number of Rows for All Executions	1
Actual Number of Batches	0
Estimated I/O Cost	0.142384
Estimated Operator Cost	0.175568 (91%)
Estimated CPU Cost	0.0331834
Estimated Subtree Cost	0.175568
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	30024
Estimated Row Size	35 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	1

Predicate
[ecommerce].[ecommerce].[MerchantProduct].[SourceProductId] ='sku_32'
Object
[ecommerce].[ecommerce].[MerchantProduct].[ix_MerchantProduct_MerchantId_SourceProductId]
Output List
[ecommerce].[ecommerce].[MerchantProduct].ProductId, [ecommerce].[ecommerce].[MerchantProduct].SourceProductId, [ecommerce].[ecommerce].[MerchantProduct].MerchantId



Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

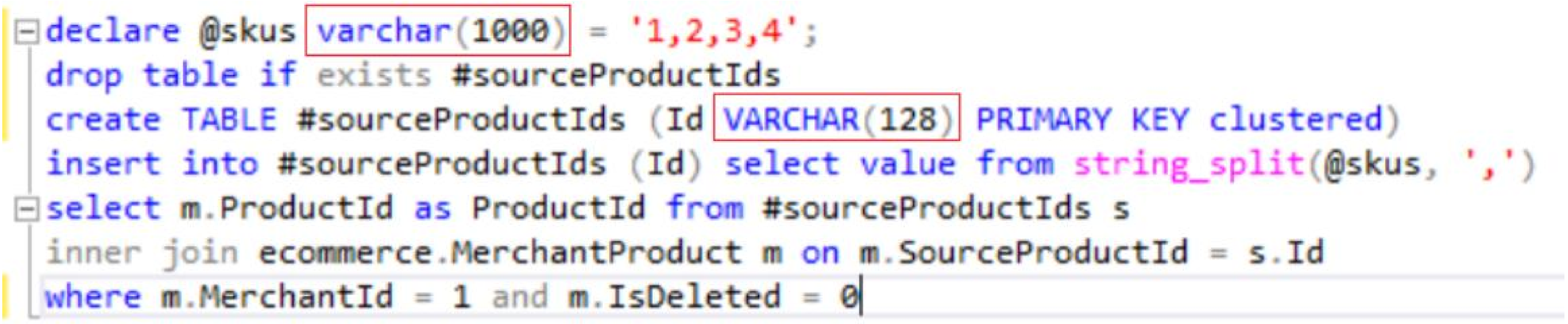
Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	1
Actual Number of Rows for All Executions	1
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0032831 (50%)
Estimated CPU Cost	0.0001581
Estimated Subtree Cost	0.0032831
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	1
Estimated Row Size	35 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	1

Object
[ecommerce].[ecommerce].[MerchantProduct].[ix_MerchantProduct_MerchantId_SourceProductId]
Output List
[ecommerce].[ecommerce].[MerchantProduct].ProductId, [ecommerce].[ecommerce].[MerchantProduct].SourceProductId, [ecommerce].[ecommerce].[MerchantProduct].MerchantId
Seek Predicates
Seek Keys[1]: Prefix: [ecommerce].[ecommerce].[MerchantProduct].MerchantId, [ecommerce].[ecommerce].[MerchantProduct].SourceProductId = Scalar Operator((99)), Scalar Operator('sku_32')

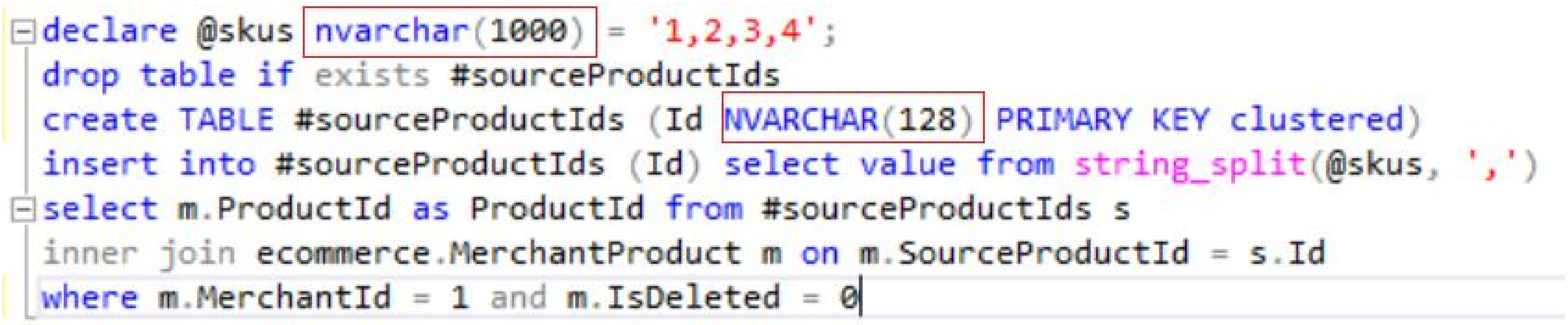
Whether indexes are used correctly

- How to use indexes

SourceProductId varchar(128) not null,

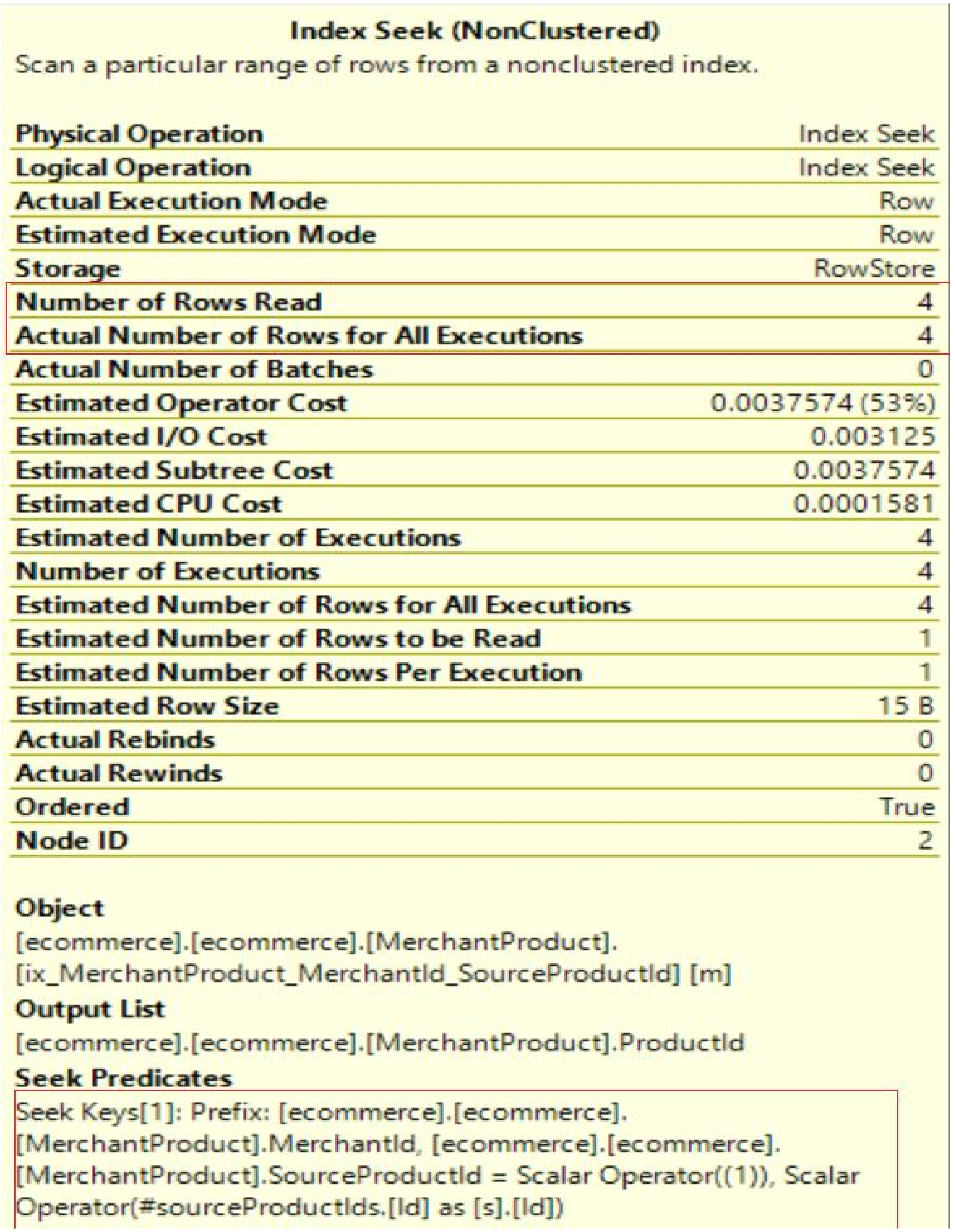


```
declare @skus varchar(1000) = '1,2,3,4';
drop table if exists #sourceProductIds
create TABLE #sourceProductIds (Id VARCHAR(128) PRIMARY KEY clustered)
insert into #sourceProductIds (Id) select value from string_split(@skus, ',')
select m.ProductId as ProductId from #sourceProductIds s
inner join ecommerce.MerchantProduct m on m.SourceProductId = s.Id
where m.MerchantId = 1 and m.IsDeleted = 0
```



```
declare @skus nvarchar(1000) = '1,2,3,4';
drop table if exists #sourceProductIds
create TABLE #sourceProductIds (Id NVARCHAR(128) PRIMARY KEY clustered)
insert into #sourceProductIds (Id) select value from string_split(@skus, ',')
select m.ProductId as ProductId from #sourceProductIds s
inner join ecommerce.MerchantProduct m on m.SourceProductId = s.Id
where m.MerchantId = 1 and m.IsDeleted = 0
```

Same type



Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	4
Actual Number of Rows for All Executions	4
Actual Number of Batches	0
Estimated Operator Cost	0.0037574 (53%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0037574
Estimated CPU Cost	0.0001581
Estimated Number of Executions	4
Number of Executions	4
Estimated Number of Rows for All Executions	4
Estimated Number of Rows to be Read	1
Estimated Number of Rows Per Execution	1
Estimated Row Size	15 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	2

Object

[ecommerce].[ecommerce].[MerchantProduct].[ix_MerchantProduct_MerchantId_SourceProductId] [m]

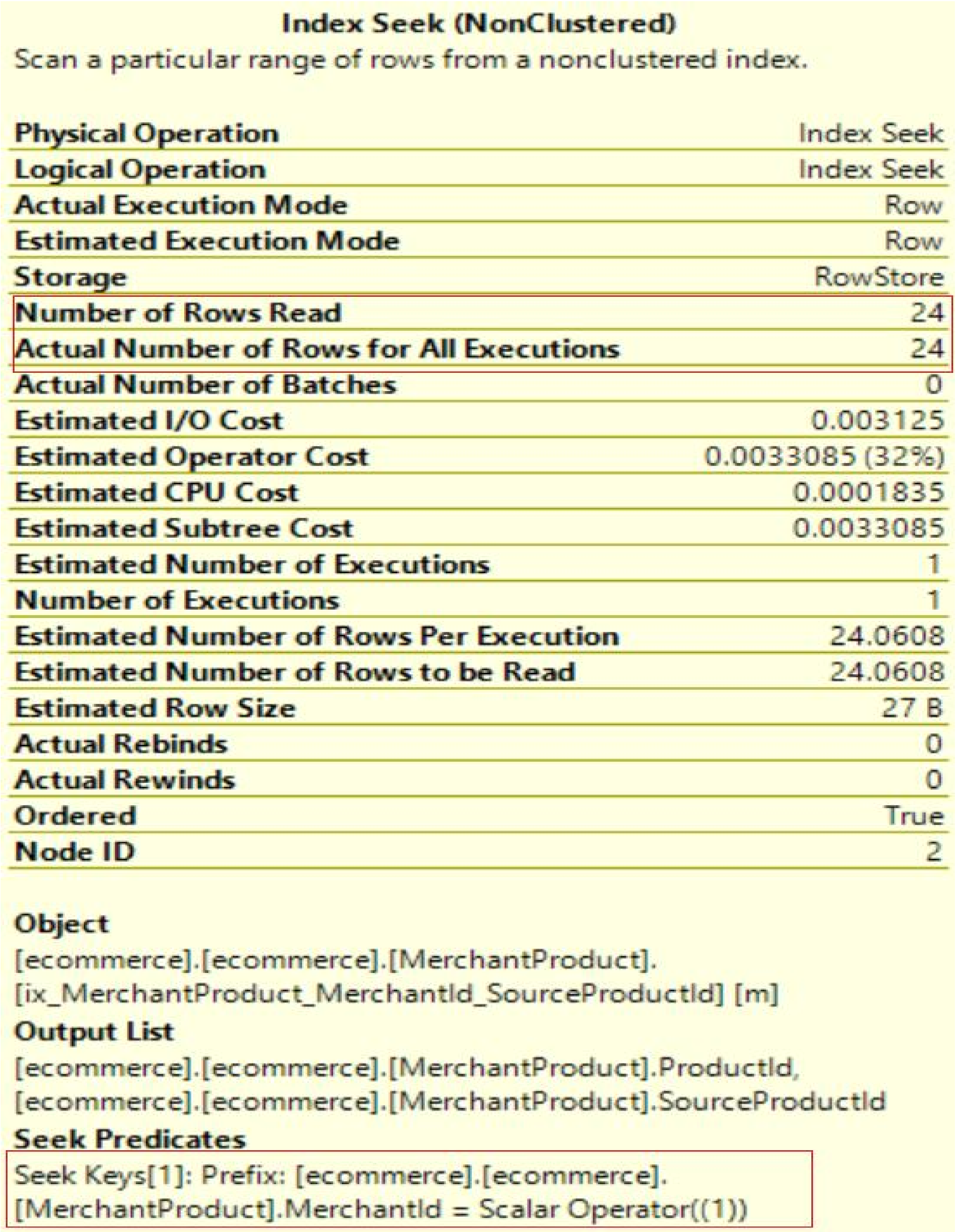
Output List

[ecommerce].[ecommerce].[MerchantProduct].ProductId

Seek Predicates

Seek Keys[1]: Prefix: [ecommerce].[ecommerce].[MerchantProduct].MerchantId, [ecommerce].[ecommerce].[MerchantProduct].SourceProductId = Scalar Operator((1)), Scalar Operator(#sourceProductIds.[Id] as [s].[Id])

Different types



Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	24
Actual Number of Rows for All Executions	24
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0033085 (32%)
Estimated CPU Cost	0.0001835
Estimated Subtree Cost	0.0033085
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	24.0608
Estimated Number of Rows to be Read	24.0608
Estimated Row Size	27 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	2

Object

[ecommerce].[ecommerce].[MerchantProduct].[ix_MerchantProduct_MerchantId_SourceProductId] [m]

Output List

[ecommerce].[ecommerce].[MerchantProduct].ProductId, [ecommerce].[ecommerce].[MerchantProduct].SourceProductId

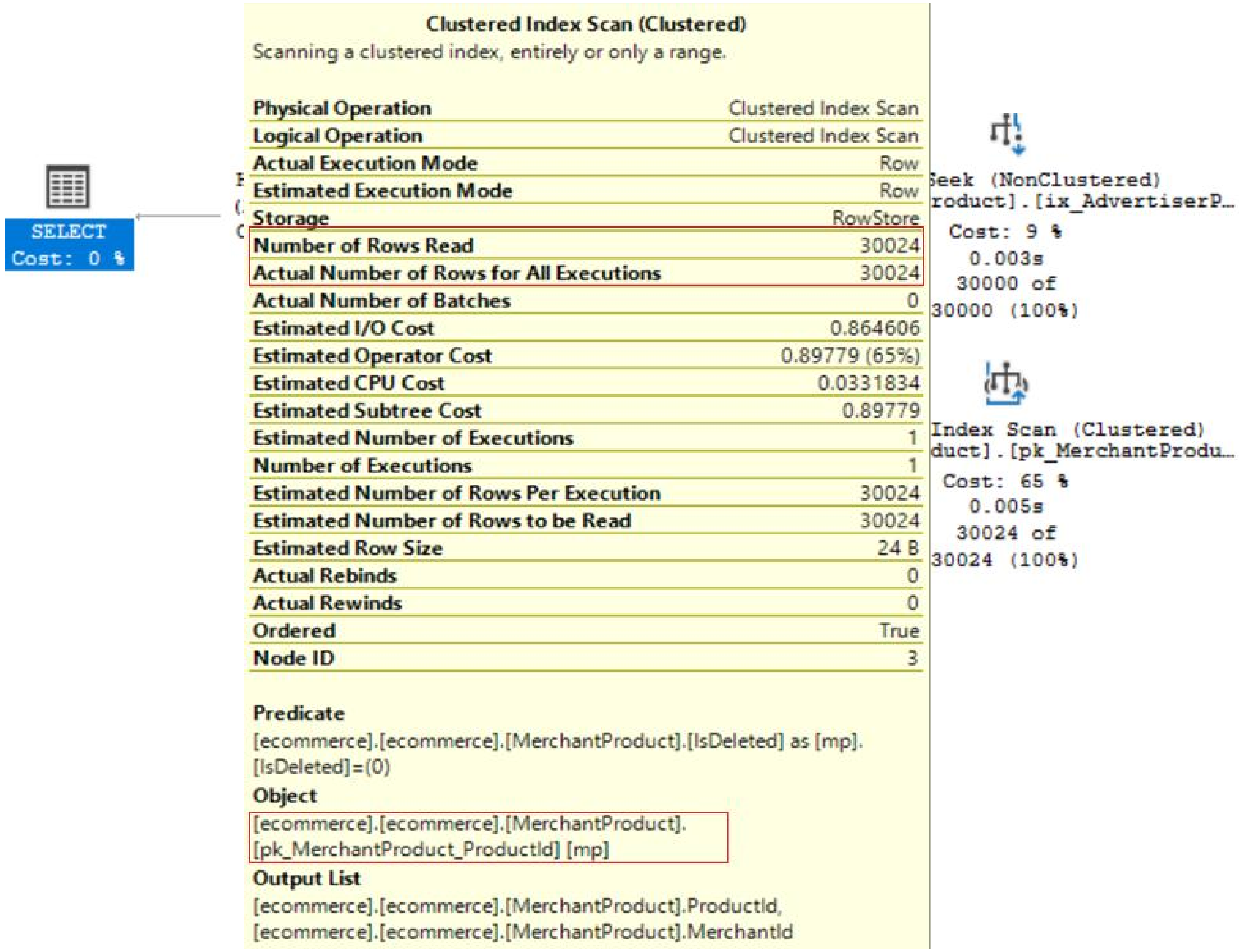
Seek Predicates

Seek Keys[1]: Prefix: [ecommerce].[ecommerce].[MerchantProduct].MerchantId = Scalar Operator((1))

Indexed view

- Need to scan all the records to aggregate

```
select
   mp.MerchantId, ap.AdvertiserId, count_big(*)
ProductCount
from ecommerce.AdvertiserProduct ap
   inner join ecommerce.MerchantProduct mp ON
ap.ProductId = mp.ProductId
WHERE mp.IsDeleted = 0
group by mp.MerchantId, ap.AdvertiserId
```



Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	30024
Actual Number of Rows for All Executions	30024
Actual Number of Batches	0
Estimated I/O Cost	0.864606
Estimated Operator Cost	0.89779 (65%)
Estimated CPU Cost	0.0331834
Estimated Subtree Cost	0.89779
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	30024
Estimated Number of Rows to be Read	30024
Estimated Row Size	24 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	3

Predicate

[ecommerce].[ecommerce].[MerchantProduct].[IsDeleted] as [mp].[IsDeleted]=(0)

Object

[ecommerce].[ecommerce].[MerchantProduct].[pk_MerchantProduct_ProductId] [mp]

Output List

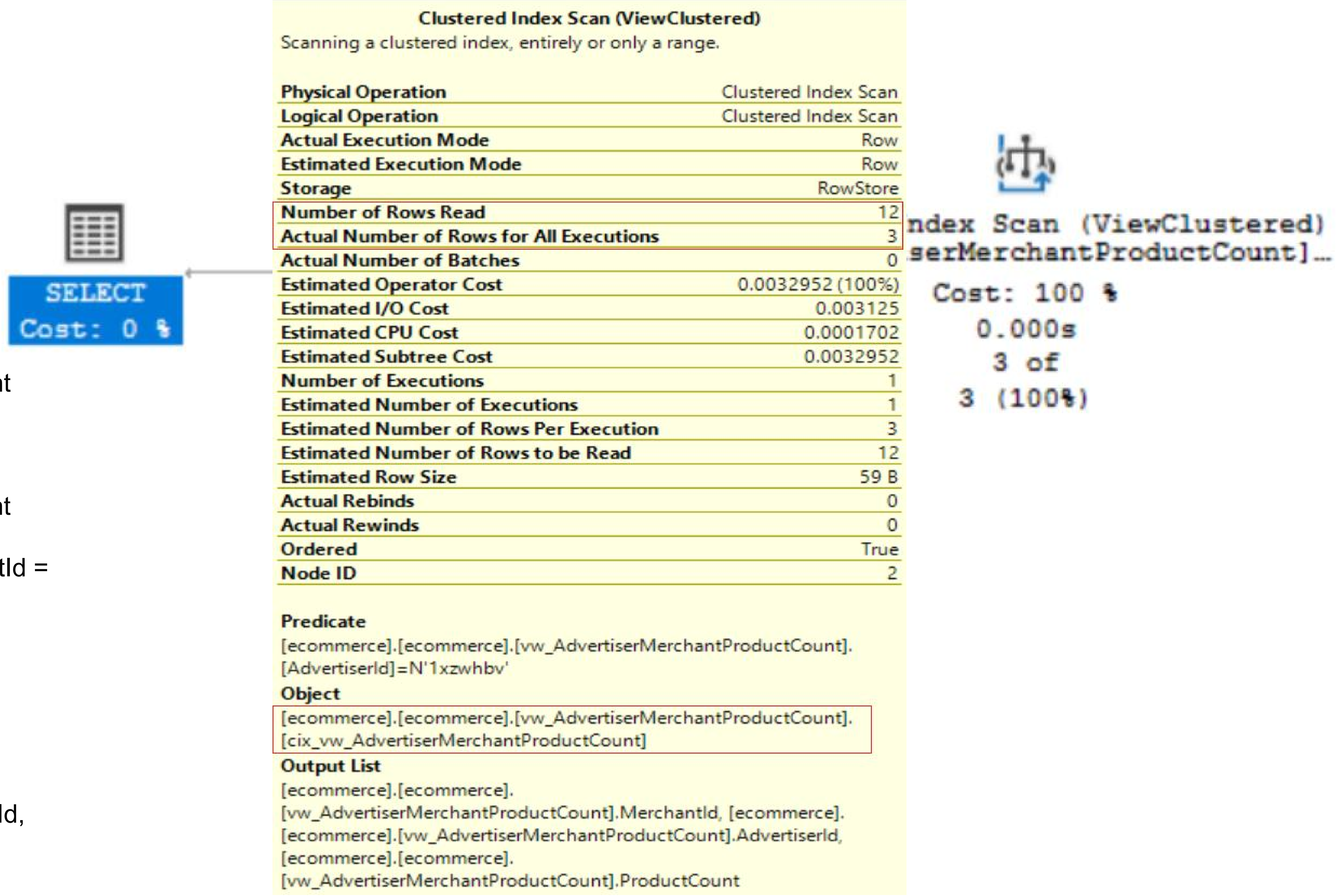
[ecommerce].[ecommerce].[MerchantProduct].ProductId, [ecommerce].[ecommerce].[MerchantProduct].MerchantId

Indexed View – cont.

- Indexed views are materialized views that store data like a table.
- Used to store some aggregated data.

```
create view ecommerce.vw_AdvertiserMerchantProductCount
with schemabinding
as
select
   mp.MerchantId, ap.AdvertiserId, count_big(*) ProductCount
from ecommerce.AdvertiserProduct ap
   inner join ecommerce.MerchantProduct mp ON ap.ProductId =
mp.ProductId
WHERE mp.IsDeleted = 0
group by mp.MerchantId, ap.AdvertiserId
go

create unique clustered index
cix_vw_AdvertiserMerchantProductCount on
ecommerce.vw_AdvertiserMerchantProductCount (MerchantId,
AdvertiserId)
go
```



SELECT
Cost: 0 %

ndex Scan (ViewClustered)
serMerchantProductCount]...
Cost: 100 %
0.000s
3 of
3 (100%)

Clustered Index Scan (ViewClustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	12
Actual Number of Rows for All Executions	3
Actual Number of Batches	0
Estimated Operator Cost	0.0032952 (100%)
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0001702
Estimated Subtree Cost	0.0032952
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	3
Estimated Number of Rows to be Read	12
Estimated Row Size	59 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	2

Predicate
[ecommerce].[ecommerce].[vw_AdvertiserMerchantProductCount].[AdvertiserId]=N'1xzwhbv'

Object
[ecommerce].[ecommerce].[vw_AdvertiserMerchantProductCount].[cix_vw_AdvertiserMerchantProductCount]

Output List
[ecommerce].[ecommerce].[vw_AdvertiserMerchantProductCount].MerchantId, [ecommerce].[ecommerce].[vw_AdvertiserMerchantProductCount].AdvertiserId, [ecommerce].[ecommerce].[vw_AdvertiserMerchantProductCount].ProductCount

Summay

- Create necessary indexes to improve query performance
- Check whether an index works as expected
- Use indexed view to accelerate some calculations

THANKS

软件正在重新定义世界

Software Is Redefining The World

TheTradeDesk