

Indexes – example

```

use Example_Lab1
go

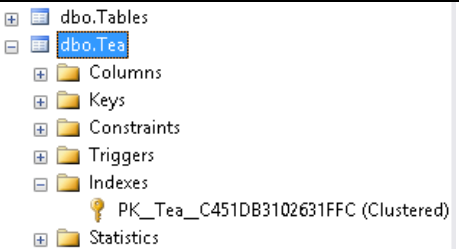
create table Tea(
Tid int primary key identity,
TName varchar(50),
Price int)

insert into Tea values ('Mint', 10), ('Ginger', 12), ('Fruits', 9), ('Rose', 8)
select * from Tea

```

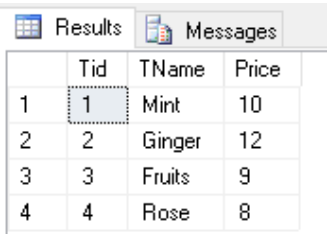
	Tid	TName	Price
1	1	Mint	10
2	2	Ginger	12
3	3	Fruits	9
4	4	Rose	8

Automatically a clustered index is created on the primary key (when this one is created). On a table one can have only one clustered index.

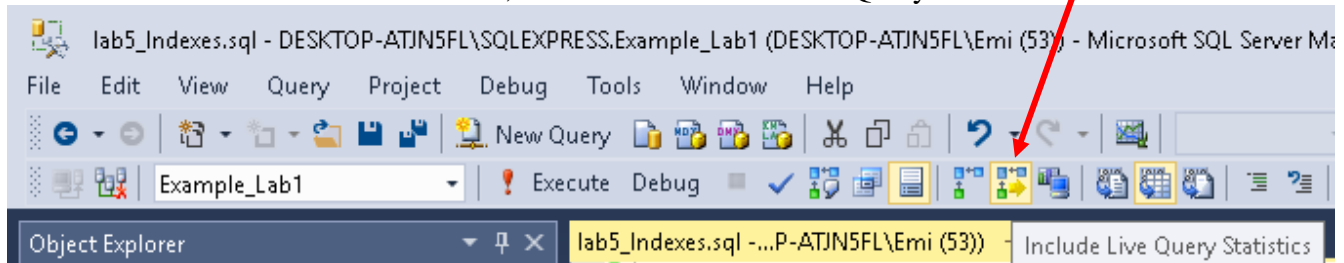


```
-- 1 clustered index was created on the primary key
-- PK_Tea_C451DB3102631FFC

select * from Tea
order by Tid
```



To check the indexes and how are used, one can use Include Live Query Statistics.



```
-- Include Live Query Statistics
```

```
select * from Tea
```

```
order by Tid
```

Results Messages Live Query Statistics

Estimated query progress: 100% Query 1: Query cost (relative to the batch): 100% select \* from Tea order by Tid

Clustered Index Scan (Clustered)  
[Tea].[PK\_Tea\_C451DB3102631FFC]  
4 of 4 (100%)

#### Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Estimated operator progress: 100%

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	4
Actual Number of Rows	4
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0032864 (100%)
Estimated Subtree Cost	0.0032864
Estimated CPU Cost	0.0001614
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows	4
Estimated Row Size	44 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	0

#### Object

[Example\_Lab1].[dbo].[Tea].[PK\_Tea\_C451DB3102631FFC]

#### Output List

[Example\_Lab1].[dbo].[Tea].Tid, [Example\_Lab1].[dbo].[Tea].TName, [Example\_Lab1].[dbo].[Tea].Price

### Create Non-Clustered Indexes by Design View

- in the table tabs – right click on Indexes – new Index – Non-Clustered Index – Name (if one wants to change it) – Add (choose the field(s) for the non-clustered index) – ok - ok

New Index

Add at least one index key column to the index

Select a page: General, Options, Storage, Filter, Extended Properties

Table name: Tea

Index name: NonClusteredIndex-20181211-132145

Index type: Nonclustered

☐ Unique

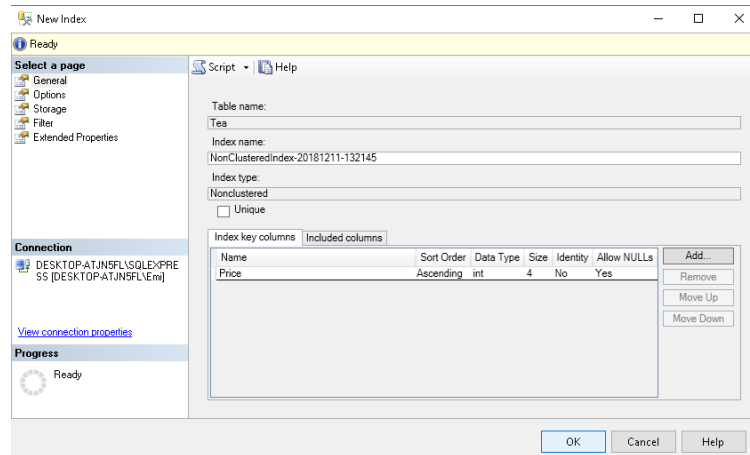
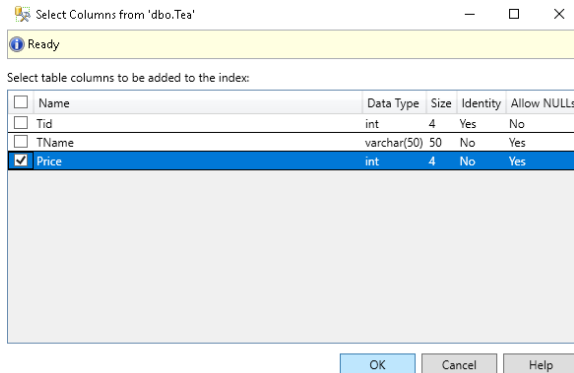
Index key columns: Included columns

Name: Sort Order: Data Type: Size: Identity: Allow NULLs: Add... Remove Move Up Move Down

Connection: DESKTOPATJNSFL\SQLEXPRESS [DESKTOPATJNSFL\Emilia]

Progress: Ready

OK Cancel Help



-- create non-clustered index - by Design View: NonClusteredIndex-20181211-132145

### Check the indexes for the current database

```
-- all the indexes from the
current database
select * from sys.indexes
```

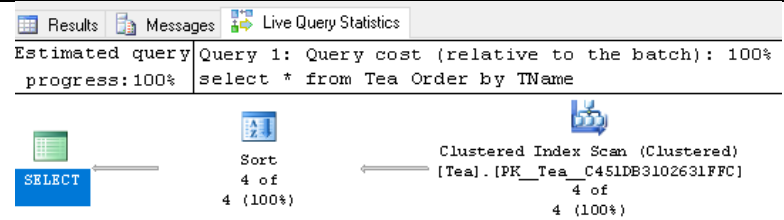
object_id	name	index_id	type	type_desc	is_unique	data_space_id	ignore_dup_key	is_primary_key	is_unique_constraint	fill_factor	is_padded
1	3	clst	1	1	CLUSTERED	1	1	0	0	0	0
2	5	clst	1	1	CLUSTERED	1	1	0	0	0	0
3	6	clst	1	1	CLUSTERED	1	1	0	0	0	0
4	7	clst	1	1	CLUSTERED	1	1	0	0	0	0
5	7	nc	2	2	NONCLUSTERED	1	1	0	0	0	0
6	8	NULL	0	0	HEAP	0	1	0	0	0	0
7	9	clst	1	1	CLUSTERED	1	1	0	0	0	0
8	17	cl	1	1	CLUSTERED	1	1	0	0	0	0
9	17	nc	2	2	NONCLUSTERED	1	1	0	0	0	0
10	17	nc2	3	2	NONCLUSTERED	1	1	0	0	0	0
11	18	cl	1	1	CLUSTERED	1	1	0	0	0	0

```
select name from sys.indexes
```

Results	
	name
1	clst
2	clst
3	clst
4	clst
5	nc
6	NULL
7	clst
8	cl
9	nc
10	nc2
11	cl

### Create Non-Clustered Indexes by Code

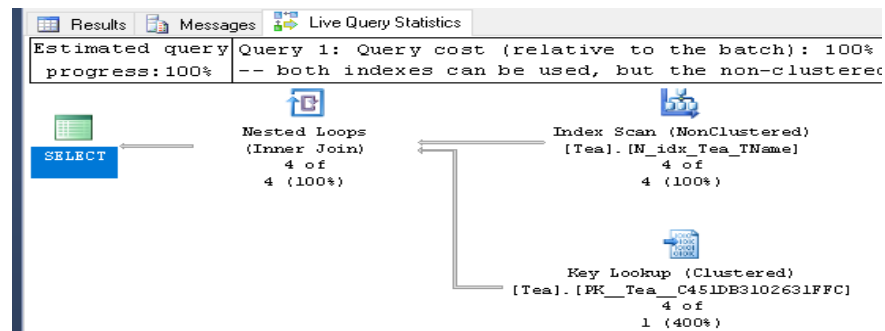
```
select * from Tea
Order by TName
-- only the clustered index is used
```



```
--create index non-clustered on the TName field
IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'N_idx_Tea_TName')
  DROP INDEX N_idx_Tea_TName ON Tea;
GO

CREATE NONCLUSTERED INDEX N_idx_Tea_TName ON Tea(TName);
GO
```

```
-- both indexes can be used, but the non-clustered one is more efficient
select * from Tea
Order by TName
```



#### Index Scan (NonClustered)

Scan a nonclustered index, entirely or only a range.

Estimated operator progress: 100%

Physical Operation	Index Scan
Logical Operation	Index Scan
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	4
Estimated Operator Cost	0.0032864 (47%)
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0001614
Estimated Subtree Cost	0.0032864
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows	4
Estimated Row Size	40 B
Ordered	True
Node ID	1

#### Object

[Example\_Lab1].[dbo].[Tea].[N\_idx\_Tea\_TName]

#### Output List

[Example\_Lab1].[dbo].[Tea].Tid, [Example\_Lab1].[dbo].[Tea].TName

Key Lookup (Clustered)	
Uses a supplied clustering key to lookup on a table that has a clustered index.	
Estimated operator progress: 100%	
Physical Operation	Key Lookup
Logical Operation	Key Lookup
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	4
Estimated Operator Cost	0.0037574 (53%)
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0001581
Estimated Subtree Cost	0.0037574
Number of Executions	4
Estimated Number of Executions	4
Estimated Number of Rows	1
Estimated Row Size	11 B
Ordered	True
Node ID	3
<b>Object</b> [Example_Lab1].[dbo].[Tea]. [PK_Tea_C451DB3102631FFC]	
<b>Output List</b> [Example_Lab1].[dbo].[Tea].Price	
<b>Seek Predicates</b> Seek Keys[1]: Prefix: [Example_Lab1].[dbo].[Tea].Tid = Scalar Operator([Example_Lab1].[dbo].[Tea].[Tid])	

dbo.Tea
 

- Columns
- Keys
- Constraints
- Triggers
- Indexes
  - N\_idx\_Tea\_TName (Non-Unique, Non-Clustered)
  - NonClusteredIndex-20181211-132145 (Non-Unique, Non-Clustered)
  - PK\_Tea\_C451DB3102631FFC (Clustered)
- Statistics

The Non-clustered index should be created on the fields involved in ORDER BY clauses, WHERE clause, JOIN clauses, to increase the efficiency and decrease the execution time.

### Create view

```
-- create view
create view vTea
as
    select * from Tea
    where TName LIKE 'a%'
go

-- execute
select * from vTea
order by TName
```

