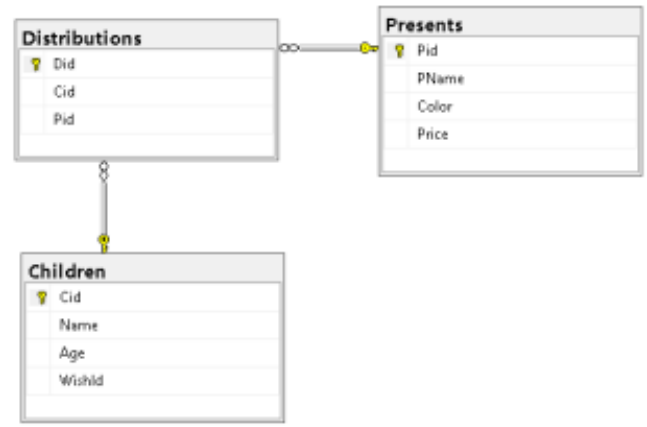
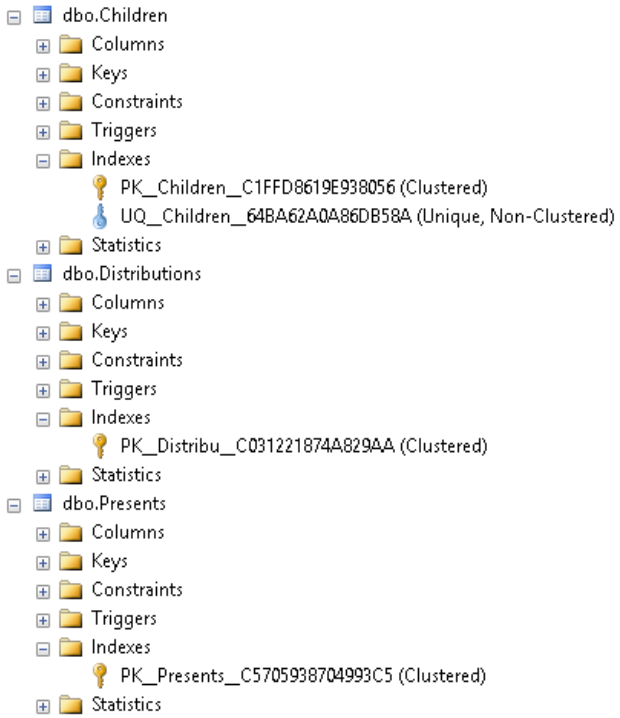
Seminar 5 – Examples with Indexes

We consider the following database:

```
-- drop database Seminar5_Example
create database Seminar5_Example
go
use Seminar5_Example
go

CREATE TABLE Children(
Cid INT PRIMARY KEY IDENTITY,
Name VARCHAR(50),
Age INT,
WishId INT UNIQUE
)

CREATE TABLE Presents(
Pid INT PRIMARY KEY IDENTITY,
PName VARCHAR(50),
Color VARCHAR(50),
Price INT
)

CREATE TABLE Distributions(
Did INT PRIMARY KEY IDENTITY,
Cid INT FOREIGN KEY REFERENCES Children(Cid),
Pid INT FOREIGN KEY REFERENCES Presents(Pid)
)

select * from Children
select * from Presents
select * from Distributions
```

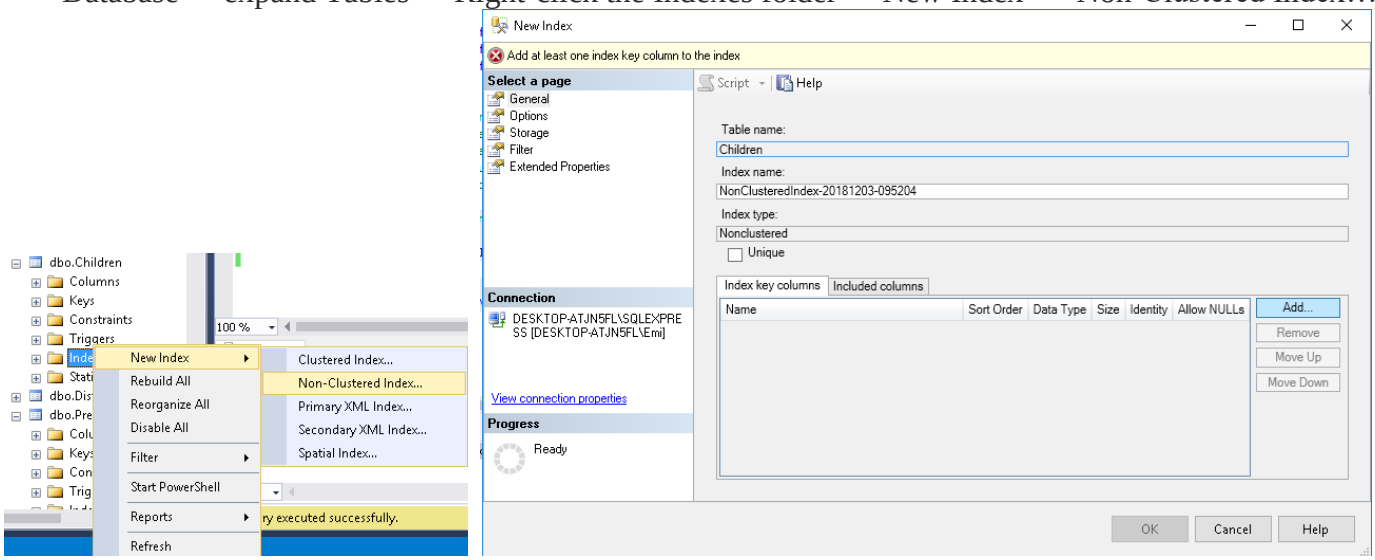



```
/* the clustered indexes have been automatically created on the tables:
- Children - clustered PK__Children__C1FFD8619E938056 and unique UQ__Children__64BA62A0A86DB58A
- Distributions -  PK__Distribu__C031221874A829AA
- Presents - PK__Presents__C5705938704993C5
*/
```

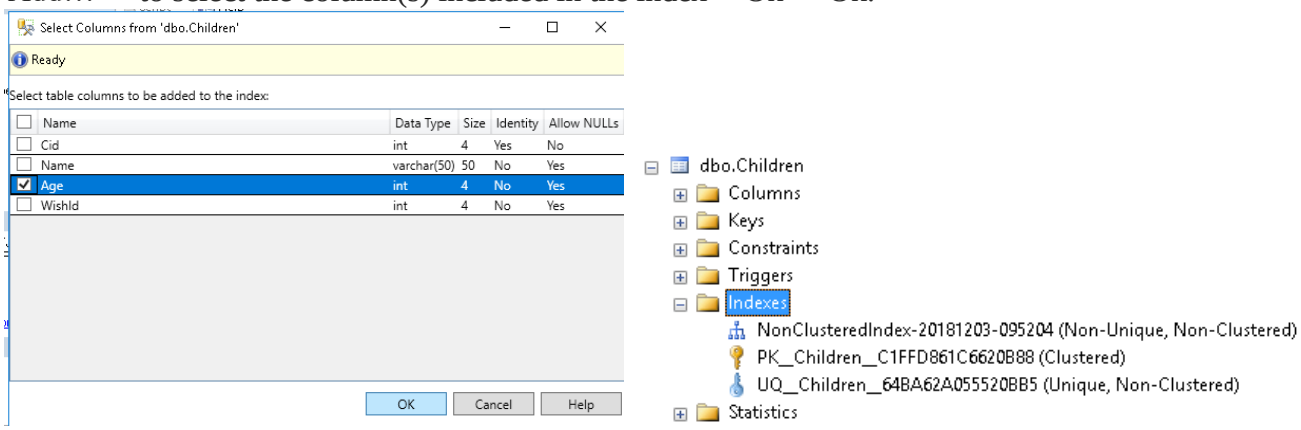


Create a *nonclustered index* on a table – with Object Explorer

- Database -> expand Tables -> Right-click the Indexes folder -> New Index -> Non-Clustered Index…



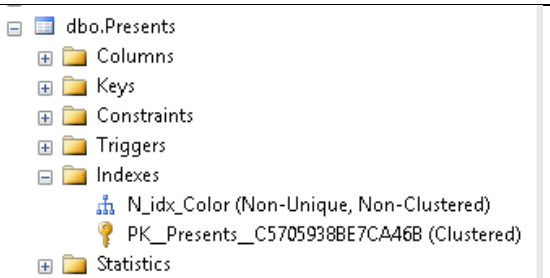
- Add… -> to select the column(s) included in the index ->Ok -> Ok.



Create a *nonclustered index* on a table – with Transact-SQL

- Choose the database -> New Query -> write the code -> Execute

```
-- Find an existing index named N_idx_Color and delete it if found.
IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'N_idx_Color')
    DROP INDEX N_idx_Color ON Presents;
GO
-- Create a nonclustered index called N_idx_Color on the Presents table using the Color column.
CREATE NONCLUSTERED INDEX N_idx_Color ON Presents(Color);
GO
```



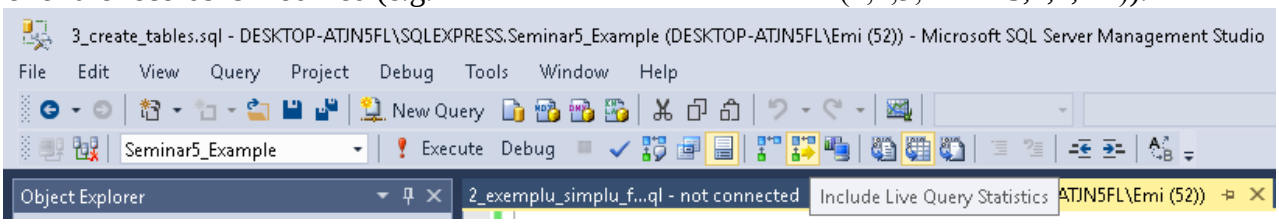

```
Command(s) completed successfully.
```

To see the use of the indexes it is better to have data(s) – records in the tables.
The indexes should be created on the fields (columns) that are used in the WHERE, JOIN, ORDER BY clause(s).

```
insert into Children (Name, Age, WishId) VALUES ('Mihai', 12, 4), ('Daniela', 6, 1), ('Paul', 5, 9),
('Andreia', 8, 7)
(4 row(s) affected)
```

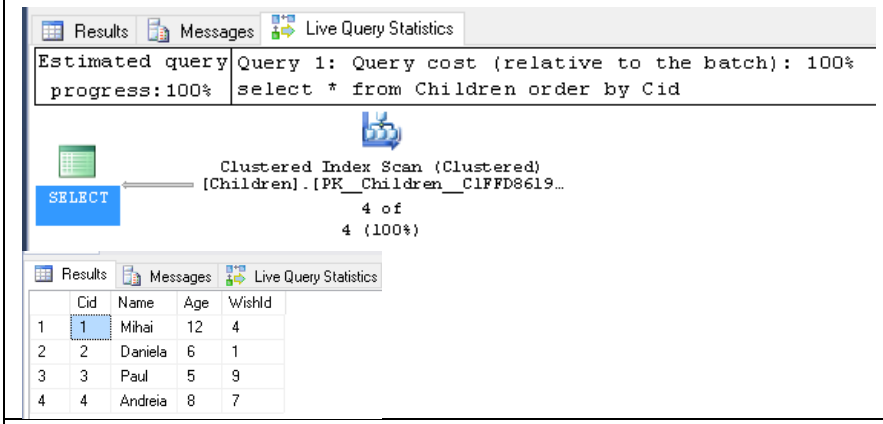
Check the Clustered/NonClustered indexes

– check **Include Live Query Statistics –** when a query is executed. After an operation (update, order by, …), the order of the records is modified (e.g. the indexes become unordered (1,2,3, … -> 3,1,2, …)).

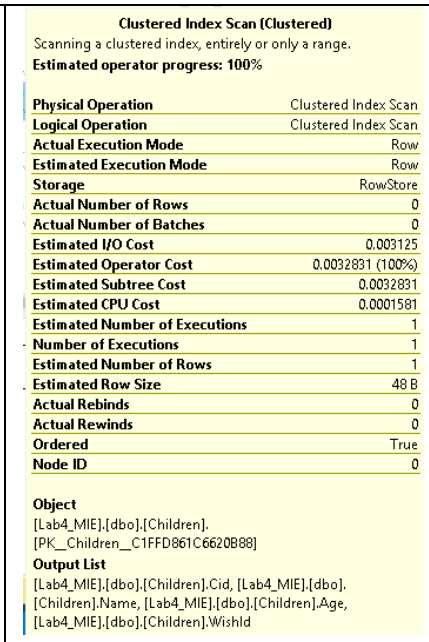


ORDER BY Examples

```
-- check Include Live Query Statistics
-- by moving the mouse through the indexes, you can check
properties of these

-- on Cid (the primary key) there is a clustered index

select * from Children order by Cid
```



	Cid	Name	Age	WishId
1	1	Mihai	12	4
2	2	Daniela	6	1
3	3	Paul	5	9
4	4	Andreia	8	7



Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

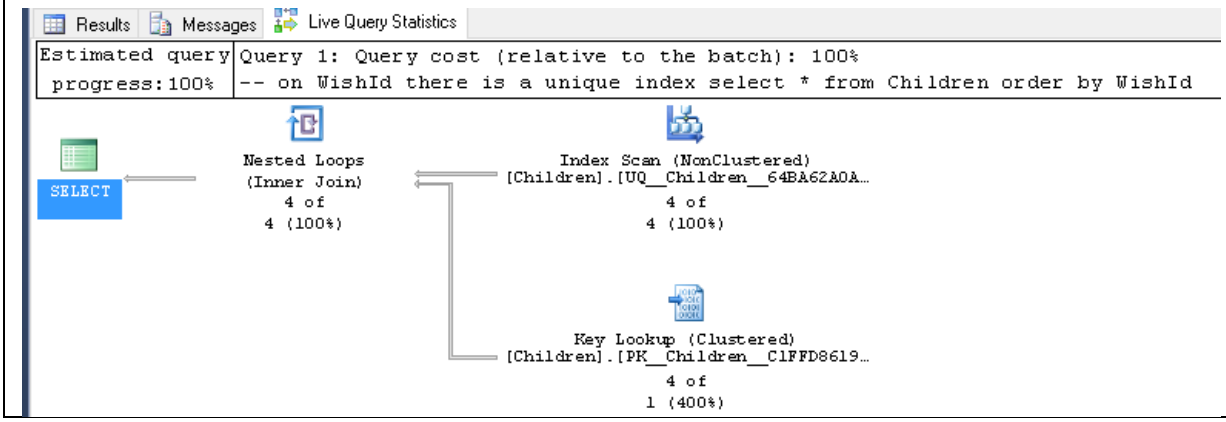
Estimated operator progress: 100%

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	0
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0032831 (100%)
Estimated Subtree Cost	0.0032831
Estimated CPU Cost	0.0001581
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows	1
Estimated Row Size	48 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	0

Object
[Lab4_MIE].[dbo].[Children].
[PK__Children__C1FFD861C6620B88]

Output List
[Lab4_MIE].[dbo].[Children].Cid, [Lab4_MIE].[dbo].
[Children].Name, [Lab4_MIE].[dbo].[Children].Age,
[Lab4_MIE].[dbo].[Children].WishId

```
-- on WishId there is a unique index
select * from Children order by WishId

-- Index Scan (NonClustered) - 47%
-- Key Lookup (Clustered) - 53%
```



Nested Loops

For each row in the top (outer) input, scan the bottom (inner) input, and output matching rows.

Estimated operator progress: 100%

Physical Operation	Nested Loops
Logical Operation	Inner Join
Estimated Execution Mode	Row
Actual Number of Rows	4
Estimated I/O Cost	0
Estimated Operator Cost	0.0000167 (0%)
Estimated Subtree Cost	0.0070605
Estimated CPU Cost	0.0000167
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows	4
Estimated Row Size	48 B
Node ID	0

Output List

[Seminar5_Example].[dbo].[Children].Cid,
[Seminar5_Example].[dbo].[Children].Name,
[Seminar5_Example].[dbo].[Children].Age,
[Seminar5_Example].[dbo].[Children].WishId

Outer References

[Seminar5_Example].[dbo].[Children].Cid

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	15 B
Ordered	True
Node ID	1

Object

[Seminar5_Example].[dbo].[Children].
[UQ__Children__64BA62A0A86DB58A]

Output List

[Seminar5_Example].[dbo].[Children].Cid,
[Seminar5_Example].[dbo].[Children].WishId

Key Lookup (Clustered)

Uses a supplied clustering key to look up 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	40 B
Ordered	True
Node ID	3

Object

[Seminar5_Example].[dbo].[Children].
[PK__Children__C1FFD8619E938056]

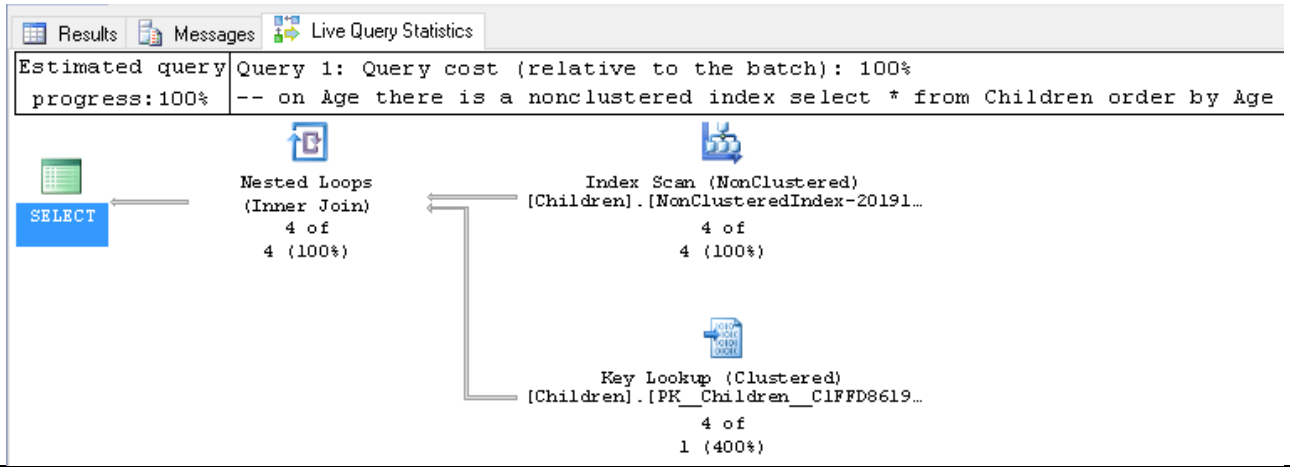
Output List

[Seminar5_Example].[dbo].[Children].Name,
[Seminar5_Example].[dbo].[Children].Age

Seek Predicates

Seek Keys[1]: Prefix: [Seminar5_Example].[dbo].[Children].Cid = Scalar Operator([Seminar5_Example].[dbo].[Children].[Cid])

```
-- on Age there is a nonclustered index
select * from Children order by Age

-- Index Scan (NonClustered) - 47%
-- Key Lookup (Clustered) - 53%
```



Index on primary key = it is created automatically when the primary key is created = index clustered
Index on unique key = it is created automatically when the unique constraint is created = index nonclustered

WHERE Examples

```
insert into Presents values ('Kitty Cat',
'blue', 9), ('Apples', 'red', 8), ('Kinder',
'White orange', 10), ('Car', 'Blue', 250)

select * from Presents
```

Results | Messages

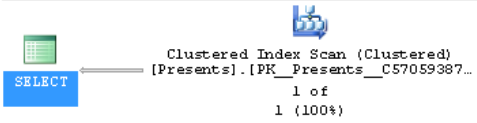
	Pid	PName	Color	Price
1	1	Kitty Cat	blue	9
2	2	Apples	red	8
3	3	Kinder	White orange	10
4	4	Car	Blue	250

```
-- without index on Price
select * from Presents where Price=8
```

Results | Messages | Live Query Statistics

Estimated query progress: 100%

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [Presents] WHERE [Price]=@1



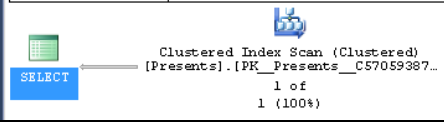

```
-- Find an existing index named N_idx_Price and delete it if found.
IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'N_idx_Price')
    DROP INDEX N_idx_Price ON Presents;
GO
-- Create a nonclustered index called N_idx_Price on the Presents table using the Price column.
CREATE NONCLUSTERED INDEX N_idx_Price ON Presents(Price);
GO
```

```
-- with index on Price
select * from Presents where Price=8
```

Results | Messages | Live Query Statistics

Estimated query progress: 100%

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [Presents] WHERE [Price]=@1



INNER JOIN Examples

```
-- INNER JOIN Examples
insert into Distributions values (1,2),
(2,1), (2, 3), (3, 4), (2,2)

select * from Children
select * from Presents
select * from Distributions
```

Results | Messages

	Cid	Name	Age	WishId
1	1	Mihai	12	4
2	2	Daniela	6	1
3	3	Paul	5	9
4	4	Andreia	8	7

	Pid	PName	Color	Price
1	1	Kitty Cat	blue	9
2	2	Apples	red	8
3	3	Kinder	White orange	10
4	4	Car	Blue	250

	Did	Cid	Pid
1	11	1	2
2	12	2	1
3	13	2	3
4	14	3	4
5	15	2	2

```
select c.Name, c.Age, p.PName, p.Color,
p.Price
from Children c INNER JOIN Distributions d
ON c.Cid=d.Cid
INNER JOIN Presents p ON p.Pid=d.Pid
Where Age BETWEEN 7 and 14 OR Price>5
```

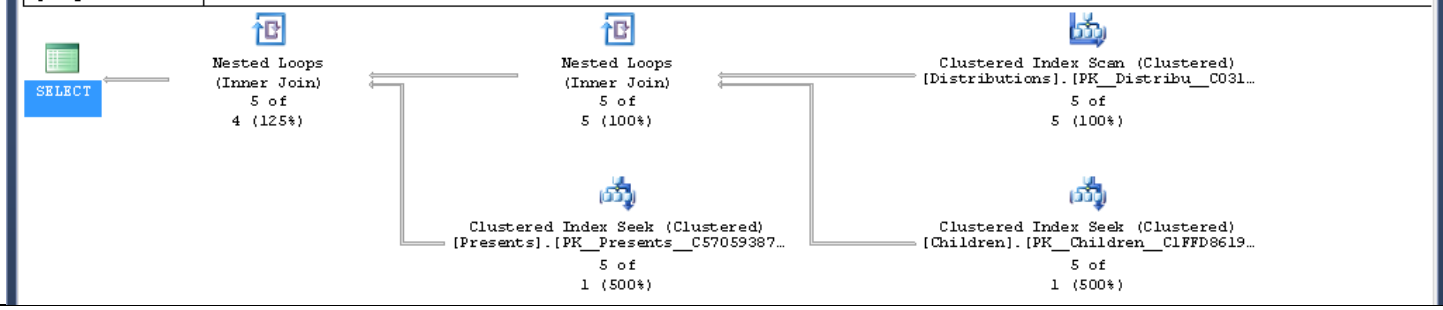
Results | Messages

	Name	Age	PName	Color	Price
1	Mihai	12	Apples	red	8
2	Daniela	6	Kitty Cat	blue	9
3	Daniela	6	Kinder	White orange	10
4	Paul	5	Car	Blue	250
5	Daniela	6	Apples	red	8

Results | Messages | Live Query Statistics

Estimated query progress: 100%

Query 1: Query cost (relative to the batch): 100%
select c.Name, c.Age, p.PName, p.Color, p.Price from Children c INNER JOIN Distributions d ON c.Cid=d.Cid



The INDEX PLAN

```
-- list the index plan
SET NOCOUNT ON;
GO
SET SHOWPLAN_ALL ON;
GO
--
SELECT *
FROM Presents
WHERE Price BETWEEN 6 AND 15;
GO
```

Results | Messages

	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument
1	SELECT * FROM Presents WHERE Price BETWEEN 6 AND 15;	1	1	0	NULL	NULL	1
2	|-Clustered Index Scan(OBJECT:([Lab4_MIE].[dbo].[Presents].[PK__Presents__C5705938…	1	2	1	Clustered Index Scan	Clustered Index Scan	OBJECT:([Lab4_MIE].[db

```
--
SET SHOWPLAN_ALL OFF
GO
SELECT *
FROM Presents
WHERE Price BETWEEN 6 AND 15;
```

Results | Messages

	Pid	PName	Color	Price
1	1	Kitty Cat	blue	9
2	2	Apples	red	8

	Rows	Executes	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument
1	2	1	SELECT * FROM [Presents] WHERE [Price]>=@1 AND […	1	1	0	NULL	NULL	NULL
2	2	1	|-Clustered Index Scan(OBJECT:([Lab4_MIE].[dbo].[Pre…	1	2	1	Clustered Index Scan	Clustered Index Scan	OBJECT:([Lab4_MIE].[dbo].[Presents

	Microsoft SQL Server 2005 XML Showplan
1	<ShowPlanXML xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan" Version="1.5" Build="13.0.1601.5"><BatchSequence><Batch><Statements><StmtSimple S…

```
SET SHOWPLAN_ALL ON;

select * from Children
```

```
select * from Presents
select * from Distributions
```

Results | Messages

	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument	Defined
1	select * from Children	1	1	0	NULL	NULL	1	NULL
2	|--Clustered Index Scan(OBJECT:([Lab4_MIE].[db...	1	2	1	Clustered Index Scan	Clustered Index Scan	OBJECT:([Lab4_MIE].[dbo].[Children].[PK__Children...	[Lab4_M
3	select * from Presents	2	3	0	NULL	NULL	2	NULL
4	|--Clustered Index Scan(OBJECT:([Lab4_MIE].[db...	2	4	3	Clustered Index Scan	Clustered Index Scan	OBJECT:([Lab4_MIE].[dbo].[Presents].[PK__Present...	[Lab4_M
5	select * from Distributions	3	5	0	NULL	NULL	3	NULL
6	|--Clustered Index Scan(OBJECT:([Lab4_MIE].[db...	3	6	5	Clustered Index Scan	Clustered Index Scan	OBJECT:([Lab4_MIE].[dbo].[Distributions].[PK__Distri...	[Lab4_M

```
SET SHOWPLAN_ALL OFF;

select * from Children
select * from Presents
select * from Distributions
```

Results | Messages

Cid	Name	Age	WishId

	Rows	Executes	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument
1	0	1	select * from Children	1	1	0	NULL	NULL	NULL
2	0	1	|--Clustered Index Scan(OBJECT:([Lab4_MIE].[db...	1	2	1	Clustered Index Scan	Clustered Index Scan	OBJECT:([Lab4_MIE].[dbo].[Children].[PK

	Microsoft SQL Server 2005 XML Showplan
1	<ShowPlanXML xmlns="http://schemas.microsoft.com...

	Pid	PName	Color	Price
1	1	Kitty Cat	blue	9
2	2	Apples	red	8

	Rows	Executes	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument	DefinedValues	EstimateRows	EstimateIO	EstimateCPU	AvgRowSize
1	2	1	select * from Presents	3	1	0	NULL	NULL	NULL	NULL	2	NULL	NULL	NULL
2	2	1	|--Clustered Index ...	3	2	1	Clustered...	Clustere...	OBJEC...	[Lab4_MIE].[...	2	0.003125	0.0001592	71

	Microsoft SQL Server 2005 XML Showplan
1	<ShowPlanXML xmlns="http://schemas.microsoft.com...

Did	Cid	Pid

	Rows	Executes	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument	DefinedValues	EstimateRows	EstimateIO	EstimateCPU	AvgRowSi
1	0	1	select * from Distributions	5	1	0	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL
2	0	1	|--Clustered Index Sca...	5	2	1	Clustered...	Clustere...	OBJEC...	[Lab4_MIE].[...	1	0.003125	0.0001581	19

	Microsoft SQL Server 2005 XML Showplan
1	<ShowPlanXML xmlns="http://schemas.microsoft.com...

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | Lab4_MIE | 00:00:00 | 11 rows

Ln 113 Col 16 Ch 16 INS

CHECK INDEXES

```
-- check the indexes (nonclustered) for the database used
SELECT TableName = st.name, IndexName = si.name, IndexId = si.index_id, ColumnId = sic.index_column_id,
ColumnName = sc.name, ObjectID=st.object_id, si.type, si.type_desc
FROM sys.indexes si INNER JOIN sys.index_columns sic ON  si.object_id = sic.object_id and si.index_id =
sic.index_id
INNER JOIN sys.columns sc ON sic.object_id = sc.object_id and sic.column_id = sc.column_id
INNER JOIN sys.tables st ON si.object_id = st.object_id
WHERE si.is_primary_key = 0 AND si.is_unique = 0 AND si.is_unique_constraint = 0 AND st.is_ms_shipped = 0
ORDER BY st.name, si.name
```

Results | Messages

	TableName	IndexName	IndexId	ColumnId	ColumnName	ObjectID	type	type_desc
1	Children	NonClusteredIndex-20191126-151509	3	1	Age	565577053	2	NONCLUSTERED
2	Presents	N_idx_Color	2	1	Color	613577224	2	NONCLUSTERED

```
-- all the indexes from table Children
select i2.name, i1.user_scans, i1.user_seeks, i1.user_updates,i1.last_user_scan,i1.last_user_seek,
i1.last_user_update
from sys.dm_db_index_usage_stats i1
inner join sys.indexes i2 on i1.index_id = i2.index_id
where i1.object_id = OBJECT_ID('Children') and i1.object_id = i2.object_id

-- all the indexes from table Presents
select i2.name, i1.user_scans, i1.user_seeks, i1.user_updates,i1.last_user_scan,i1.last_user_seek,
i1.last_user_update
from sys.dm_db_index_usage_stats i1
inner join sys.indexes i2 on i1.index_id = i2.index_id
where i1.object_id = OBJECT_ID('Presents') and i1.object_id = i2.object_id

-- all the indexes from table Distributions
select i2.name, i1.user_scans, i1.user_seeks, i1.user_updates,i1.last_user_scan,i1.last_user_seek,
i1.last_user_update
from sys.dm_db_index_usage_stats i1
inner join sys.indexes i2 on i1.index_id = i2.index_id
where i1.object_id = OBJECT_ID('Distributions') and i1.object_id = i2.object_id
```

Results | Messages

	name	user_scans	user_seeks	user_updates	last_user_scan	last_user_seek	last_user_update
1	PK__Children__C1FFD8619E938056	3	0	1	2019-11-26 15:24:52.287	NULL	2019-11-26 15:20:47.077
2	UQ__Children__64BA62A0A86DB58A	1	0	1	2019-11-26 15:27:51.483	NULL	2019-11-26 15:20:47.077
3	NonClusteredIndex-20191126-151509	1	0	1	2019-11-26 15:40:33.503	NULL	2019-11-26 15:20:47.077

	name	user_scans	user_seeks	user_updates	last_user_scan	last_user_seek	last_user_update
1	PK__Presents__C5705938704993C5	1	0	0	2019-11-26 14:25:53.267	NULL	NULL

	name	user_scans	user_seeks	user_updates	last_user_scan	last_user_seek	last_user_update
1	PK__Distribu__C031221874A829AA	1	0	0	2019-11-26 14:25:53.267	NULL	NULL

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (52) | Seminar5_Example | 00:00:00 | 5 rows

```
-- all the indexes from the current database
SELECT OBJECT_NAME(A.[OBJECT_ID]) AS [OBJECT NAME],
       I.[NAME] AS [INDEX NAME],
       A.LEAF_INSERT_COUNT,
       A.LEAF_UPDATE_COUNT,
       A.LEAF_DELETE_COUNT
FROM   SYS.DM_DB_INDEX_OPERATIONAL_STATS (NULL,NULL,NULL,NULL ) A
       INNER JOIN SYS.INDEXES AS I ON I.[OBJECT_ID] = A.[OBJECT_ID] AND I.INDEX_ID = A.INDEX_ID
WHERE  OBJECTPROPERTY(A.[OBJECT_ID],'IsUserTable') = 1
```

Results | Messages

	OBJECT NAME	INDEX NAME	LEAF_INSERT_COUNT	LEAF_UPDATE_COUNT	LEAF_DELETE_COUNT
1	Presents	PK__Presents__C5705938BE7CA46B	0	0	0
2	Children	UQ__Children__64BA62A055520BB5	432	0	0
3	Children	PK__Children__C1FFD861C6620B88	432	0	0
4	Presents	N_idx_Color	0	0	0
5	Children	PK__Children__C1FFD861C6620B88	0	0	0
6	Children	UQ__Children__64BA62A055520BB5	0	0	0
7	Presents	PK__Presents__C5705938BE7CA46B	0	0	0
8	Distributions	PK__Distribu__C031221883C8CCCE	0	0	0
9	sysdiagrams	PK__sysdiagr__C2B05B61B7EE2DE6	1	0	0
10	sysdiagrams	UK_principal_name	1	0	0
11	Presents	N_idx_Color	0	0	0
12	Children	NonClusteredIndex-20181203-095204	0	0	0

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (88) | Lab4_MIE | 00:00:13 | 12 rows

```
-- information related to a specified table and the indexes created on that table
EXEC sp_helpindex 'Children'
EXEC sp_helpindex 'Presents'
EXEC sp_helpindex 'Distributions'
```

Results | Messages

	index_name	index_description	index_keys
1	NonClusteredIndex-20191126-151509	nonclustered located on PRIMARY	Age
2	PK__Children__C1FFD8619E938056	clustered, unique, primary key located on PRIMARY	Cid
3	UQ__Children__64BA62A0A86DB58A	nonclustered, unique, unique key located on PRIMARY	WishId

	index_name	index_description	index_keys
1	N_idx_Color	nonclustered located on PRIMARY	Color
2	PK__Presents__C5705938704993C5	clustered, unique, primary key located on PRIMARY	Pid

	index_name	index_description	index_keys
1	PK__Distribu__C031221874A829AA	clustered, unique, primary key located on PRIMARY	Did

```
-- return primary key(s) for the given table
EXEC sp_pkeys @table_name = N'Children'
-- PK__Children__C1FFD8619E938056
```

Results | Messages

	TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	KEY_SEQ	PK_NAME
1	Seminar5_Example	dbo	Children	Cid	1	PK__Children__C1FFD8619E938056