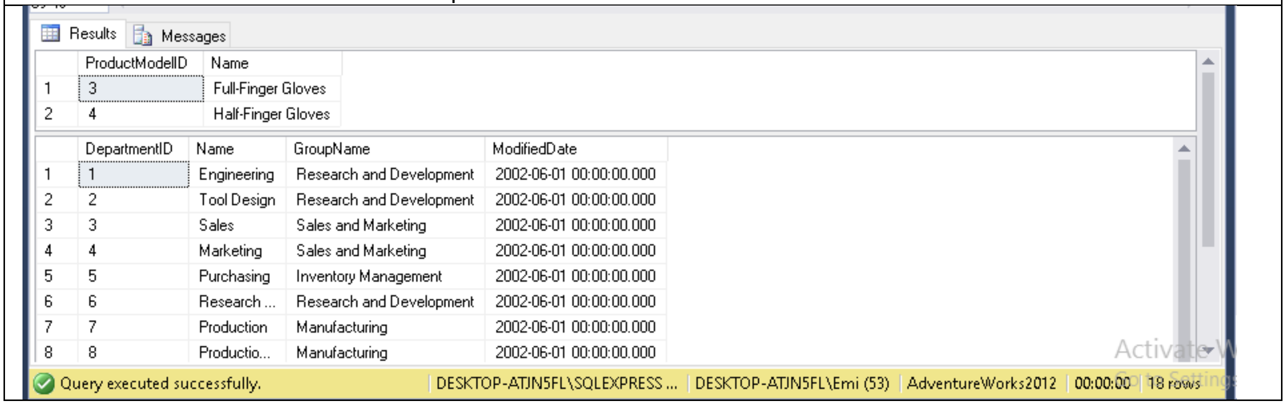
Seminar 7 – Mai multe despre optimizare

SLIDE 3

```
use AdventureWorks2012
go


SET NOCOUNT ON;
insert into Gloves values ('test')
-- Command(s) completed successfully.

SET NOCOUNT OFF;
insert into Gloves values ('test')
-- (1 row(s) affected)

Select * from dbo.Gloves
select * from HumanResources.Department
```



	ProductModelID	Name
1	3	Full-Finger Gloves
2	4	Half-Finger Gloves

	DepartmentID	Name	GroupName	ModifiedDate
1	1	Engineering	Research and Development	2002-06-01 00:00:00.000
2	2	Tool Design	Research and Development	2002-06-01 00:00:00.000
3	3	Sales	Sales and Marketing	2002-06-01 00:00:00.000
4	4	Marketing	Sales and Marketing	2002-06-01 00:00:00.000
5	5	Purchasing	Inventory Management	2002-06-01 00:00:00.000
6	6	Research ...	Research and Development	2002-06-01 00:00:00.000
7	7	Production	Manufacturing	2002-06-01 00:00:00.000
8	8	Productio...	Manufacturing	2002-06-01 00:00:00.000

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:00 | 18 rows

```
USE [AdventureWorks2012]
GO
/****** Object:  StoredProcedure [dbo].[execution_time_performance]    Script Date:
5/30/2018 9:21:58 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Author,,Name>
-- Create date: <Create Date,,>
-- Description:      <Description,,>
-- =============================================

ALTER PROCEDURE [dbo].[execution_time_performance]
       -- Add the parameters for the stored procedure here
       @table_name varchar(100)=NULL
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       -- interfering with SELECT statements.
       SET NOCOUNT ON;
```

```
        DECLARE @t1 DATETIME;
        DECLARE @t2 DATETIME;
        DECLARE @elapsed_ms INT;
        DECLARE @table_full_name varchar(100);

        IF @table_name is null
                PRINT N'You must provide a table name as an input parameter';
        ELSE
        IF not exists (select * from dbo.sysobjects where name = @table_name and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
                PRINT N'The provided table name (' + RTRIM(convert(varchar(100),
@table_name)) + ') does not exist';
        ELSE
        BEGIN
                PRINT N'The provided table name (' + RTRIM(convert(varchar(100),
@table_name)) + ') exists';
                SET @t1 = GETDATE();

                SELECT top 1 @table_full_name = SCHEMA_NAME(schema_id) + '.' + name FROM
sys.tables where name=@table_name
                EXEC('SELECT * FROM ' + @table_full_name)

                SET @t2 = GETDATE();
                SELECT DATEDIFF(millisecond,@t1,@t2) AS elapsed_ms;
                PRINT N'Execution time=' + RTRIM(convert(varchar(100), @elapsed_ms));
        END
END
-- select from a table and how much time does it takes
```

```
EXEC dbo.execution_time_performance
```

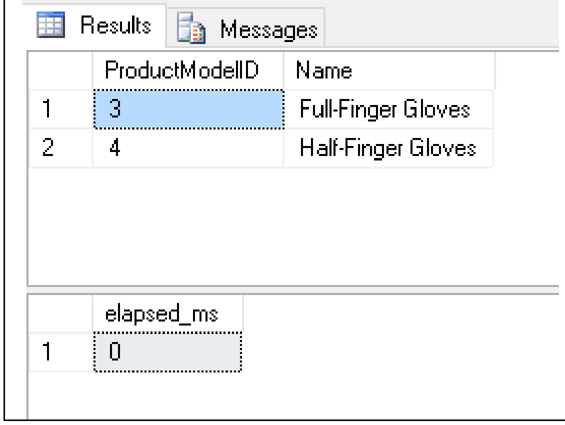
```
You must provide a table name as an input parameter
```

```
EXEC dbo.execution_time_performance [HumanResources.Department]
```

```
The provided table name (HumanResources.Department) does not exist
```

```
EXEC dbo.execution_time_performance Gloves
```

```
The provided table name (Gloves) exists
```

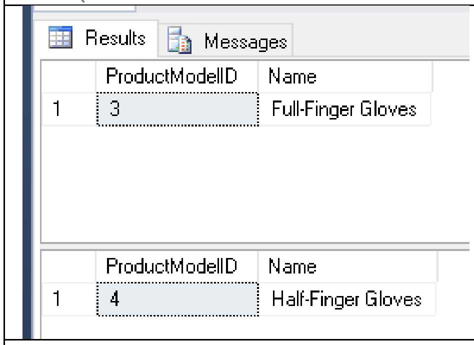


	ProductModelID	Name
1	3	Full-Finger Gloves
2	4	Half-Finger Gloves

	elapsed_ms
1	0

SLIDE 5

```
exec('Select * from Gloves where ProductModelID = 3')
exec('Select * from Gloves where ProductModelID = 4')
```



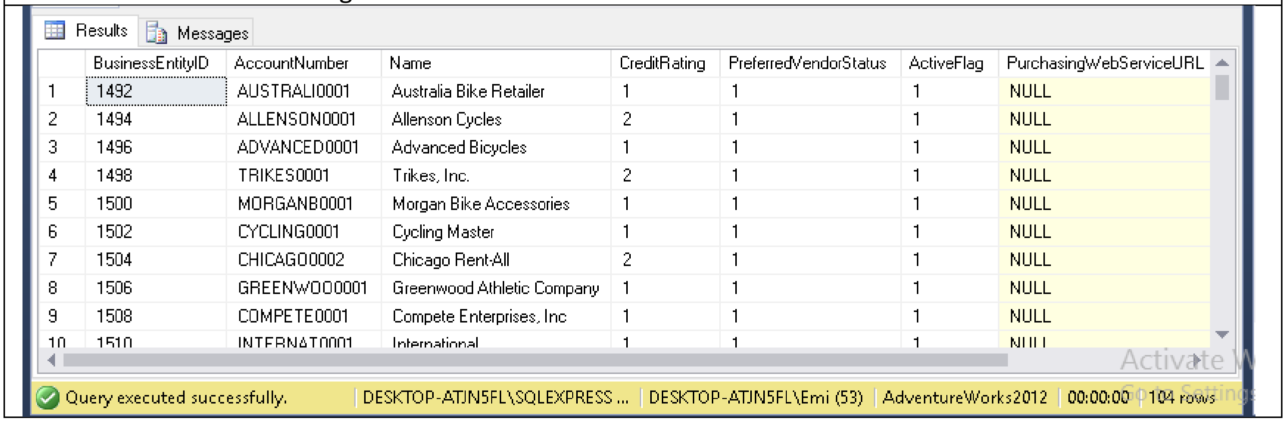
	ProductModelID	Name
1	3	Full-Finger Gloves

	ProductModelID	Name
1	4	Half-Finger Gloves

```
EXECUTE sp_executesql N'Select* from Gloves where ProductModelID = @ID', N'@ID int',@ID=3;
EXECUTE sp_executesql N'Select* from Gloves where ProductModelID = @ID', N'@ID int',@ID=4;
```

SLIDE 6 – Cursoare

```
select * from Purchasing.Vendor
```



	BusinessEntityID	AccountNumber	Name	CreditRating	PreferredVendorStatus	ActiveFlag	PurchasingWebServiceURL
1	1492	AUSTRALI0001	Australia Bike Retailer	1	1	1	NULL
2	1494	ALLENSON0001	Allenson Cycles	2	1	1	NULL
3	1496	ADVANCED0001	Advanced Bicycles	1	1	1	NULL
4	1498	TRIKES0001	Trikes, Inc.	2	1	1	NULL
5	1500	MORGANB0001	Morgan Bike Accessories	1	1	1	NULL
6	1502	CYCLING0001	Cycling Master	1	1	1	NULL
7	1504	CHICAGO0002	Chicago Rent-All	2	1	1	NULL
8	1506	GREENWOO0001	Greenwood Athletic Company	1	1	1	NULL
9	1508	COMPETE0001	Compete Enterprises, Inc	1	1	1	NULL
10	1510	INTERNAT0001	International	1	1	1	NULL

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:00 | 104 rows

```
-- The following example shows how cursors can be nested to produce complex reports.
-- The inner cursor is declared for each vendor.

SET NOCOUNT ON;
DECLARE @vendor_id int, @vendor_name nvarchar(50), @message varchar(80), @product
nvarchar(50);
PRINT '-------- Vendor Products Report --------';
DECLARE vendor_cursor CURSOR FOR
       SELECT BusinessEntityID, Name FROM Purchasing.Vendor WHERE PreferredVendorStatus =
1  ORDER BY BusinessEntityID;
OPEN vendor_cursor
FETCH NEXT FROM vendor_cursor
INTO @vendor_id, @vendor_name
WHILE @@FETCH_STATUS = 0  BEGIN
    PRINT ' '
    SELECT @message = '----- Products From Vendor: ' + @vendor_name
    PRINT @message
       -- Declare an inner cursor based on vendor_id from the outer cursor.
    DECLARE product_cursor CURSOR FOR
              SELECT v.Name FROM Purchasing.ProductVendor pv, Production.Product v
              WHERE pv.ProductID = v.ProductID AND pv.BusinessEntityID = @vendor_id  --
```

```
Variable value from the outer cursor
       OPEN product_cursor
    FETCH NEXT FROM product_cursor INTO @product
           IF @@FETCH_STATUS <> 0
        PRINT '         <<None>>'
              WHILE @@FETCH_STATUS = 0  BEGIN
                     SELECT @message = '         ' + @product
                     PRINT @message
                     FETCH NEXT FROM product_cursor INTO @product
        END
    CLOSE product_cursor
    DEALLOCATE product_cursor
        -- Get the next vendor.
    FETCH NEXT FROM vendor_cursor
    INTO @vendor_id, @vendor_name
END
CLOSE vendor_cursor;
DEALLOCATE vendor_cursor;
```

```
-------- Vendor Products Report --------

----- Products From Vendor: Australia Bike Retailer
         Thin-Jam Lock Nut 9
         Thin-Jam Lock Nut 10
         Thin-Jam Lock Nut 1
         Thin-Jam Lock Nut 2
         Thin-Jam Lock Nut 15
         Thin-Jam Lock Nut 16
         Thin-Jam Lock Nut 5
         Thin-Jam Lock Nut 6
         Thin-Jam Lock Nut 3
         Thin-Jam Lock Nut 4
         Thin-Jam Lock Nut 13
         Thin-Jam Lock Nut 14
         Thin-Jam Lock Nut 7
         Thin-Jam Lock Nut 8
         Thin-Jam Lock Nut 12
         Thin-Jam Lock Nut 11

----- Products From Vendor: Allenson Cycles
         Seat Post

----- Products From Vendor: Advanced Bicycles
         Thin-Jam Hex Nut 9
         Thin-Jam Hex Nut 10
         Thin-Jam Hex Nut 1
         Thin-Jam Hex Nut 2
         Thin-Jam Hex Nut 15
         Thin-Jam Hex Nut 16
         Thin-Jam Hex Nut 5
         Thin-Jam Hex Nut 6
         Thin-Jam Hex Nut 3
         Thin-Jam Hex Nut 4
         Thin-Jam Hex Nut 13
         Thin-Jam Hex Nut 14
         Thin-Jam Hex Nut 7
         Thin-Jam Hex Nut 8
         Thin-Jam Hex Nut 12
         Thin-Jam Hex Nut 11

----- Products From Vendor: Trikes, Inc.
         Mountain Tire Tube
         HL Mountain Tire

----- Products From Vendor: Morgan Bike Accessories
         HL Grip Tape

----- Products From Vendor: Cycling Master
         <<None>>

----- Products From Vendor: Chicago Rent-All
         Reflector

----- Products From Vendor: Greenwood Athletic Company
         LL Mountain Pedal
         ML Mountain Pedal

----- Products From Vendor: Compete Enterprises, Inc
         Guide Pulley
         Tension Pulley
         HL Road Pedal

----- Products From Vendor: International
         Front Brakes

----- Products From Vendor: Litware, Inc.
         Adjustable Race

----- Products From Vendor: Inner City Bikes
         Lock Washer 4
         Lock Washer 5
         Lock Washer 10
         Lock Washer 6
         Lock Washer 13
         Lock Washer 8
         Lock Washer 1
         Lock Washer 7
         Lock Washer 12
         Lock Washer 2
         Lock Washer 9
         Lock Washer 3
         Lock Washer 11

----- Products From Vendor: Trey Research
         Paint - Black
         Paint - Red
         Paint - Silver
         Paint - Blue
         Paint - Yellow

----- Products From Vendor: Mitchell Sports
         LL Road Pedal
         ML Road Pedal

----- Products From Vendor: Signature Cycles
         LL Road Tire
         ML Road Tire

----- Products From Vendor: SUPERSALES INC.
         Decal 1
         Decal 2

----- Products From Vendor: Lindell
         LL Nipple
         HL Nipple

----- Products From Vendor: Fitness Association
         Men's Sports Shorts, M
         Men's Sports Shorts, L
         Men's Sports Shorts, XL
         Women's Tights, S
         Women's Tights, M
         Women's Tights, L
         Men's Bib-Shorts, S
         Men's Bib-Shorts, M
         Men's Bib-Shorts, L

----- Products From Vendor: A. Datum Corporation
         <<None>>

----- Products From Vendor: Continental Pro Cycles
         Flat Washer 1
         Flat Washer 6
         Flat Washer 2
         Flat Washer 9
         Flat Washer 4
         Flat Washer 3
```

```
        Lower Head Race

----- Products From Vendor: Light Speed
        <<None>>

----- Products From Vendor: Training Systems
        Chainring Bolts
        Chainring Nut
        Chainring

----- Products From Vendor: International Trek Center
        Touring-Panniers, Large
        Cable Lock
        Minipump
        Mountain Pump
        Taillights - Battery-Powered
        Headlights - Dual-Beam
        Headlights - Weatherproof

----- Products From Vendor: G & K Bicycle Corp.
        Sport-100 Helmet, Red
        Sport-100 Helmet, Black
        Sport-100 Helmet, Blue

----- Products From Vendor: First National Sport Co.
        HL Shell

----- Products From Vendor: Recreation Place
        <<None>>

----- Products From Vendor: International Bicycles
        LL Road Rim
        ML Road Rim

----- Products From Vendor: Image Makers Bike Center
        <<None>>

----- Products From Vendor: Comfort Road Bicycles
        LL Mountain Rim
        ML Mountain Rim

----- Products From Vendor: Knopfler Cycles
        <<None>>

----- Products From Vendor: Ready Rentals
        Thin-Jam Lock Nut 9
        Thin-Jam Lock Nut 10
        Thin-Jam Lock Nut 1
        Thin-Jam Lock Nut 2
        Thin-Jam Lock Nut 15
        Thin-Jam Lock Nut 16
        Thin-Jam Lock Nut 5
        Thin-Jam Lock Nut 6
        Thin-Jam Lock Nut 3
        Thin-Jam Lock Nut 4
        Thin-Jam Lock Nut 13
        Thin-Jam Lock Nut 14
        Thin-Jam Lock Nut 7
        Thin-Jam Lock Nut 8
        Thin-Jam Lock Nut 12
        Thin-Jam Lock Nut 11

----- Products From Vendor: Cruger Bike Company
        Hex Nut 5
        Hex Nut 6
        Hex Nut 16
        Hex Nut 17
        Hex Nut 7
        Hex Nut 8
        Hex Nut 9
        Hex Nut 22
        Hex Nut 23
        Hex Nut 12
        Hex Nut 13
        Hex Nut 1
        Hex Nut 10
        Hex Nut 11
        Hex Nut 2
        Hex Nut 20
        Hex Nut 21
        Hex Nut 3
        Hex Nut 14
        Hex Nut 15
        Hex Nut 4
        Hex Nut 18
        Hex Nut 19

----- Products From Vendor: Vista Road Bikes
        LL Mountain Tire
        ML Mountain Tire

----- Products From Vendor: Bergeron Off-Roads
        Lock Nut 5
        Flat Washer 8
        Flat Washer 5
        Flat Washer 7

----- Products From Vendor: Federal Sport
        LL Shell

----- Products From Vendor: Northwind Traders
        <<None>>

----- Products From Vendor: Sport Playground
        Freewheel

----- Products From Vendor: Hybrid Bicycle Center
        HL Mountain Seat/Saddle

----- Products From Vendor: Midwest Sport, Inc.
        Cone-Shaped Race

----- Products From Vendor: Aurora Bike Center
        External Lock Washer 3
        External Lock Washer 4
        External Lock Washer 9
        External Lock Washer 5
        External Lock Washer 7
        External Lock Washer 6
        External Lock Washer 1
        External Lock Washer 8
        External Lock Washer 2
        Internal Lock Washer 3
        Internal Lock Washer 4
        Internal Lock Washer 9
        Internal Lock Washer 5
        Internal Lock Washer 7
        Internal Lock Washer 6
        Internal Lock Washer 10
        Internal Lock Washer 1
        Internal Lock Washer 8
        Internal Lock Washer 2

----- Products From Vendor: Metro Sport Equipment
        Seat Lug

----- Products From Vendor: Lakewood Bicycle
        LL Spindle/Axle

----- Products From Vendor: Speed Corporation
        Flat Washer 1
        Flat Washer 6
        Flat Washer 2
        Flat Washer 9
        Flat Washer 4
        Flat Washer 3
        Flat Washer 8
        Flat Washer 5
        Flat Washer 7

----- Products From Vendor: Competition Bike Training Systems
        LL Mountain Rim
        ML Mountain Rim

----- Products From Vendor: Hill Bicycle Center
        HL Spindle/Axle

----- Products From Vendor: Bicycle Specialists
        Touring Pedal

----- Products From Vendor: Indiana Bicycle Center
        <<None>>

----- Products From Vendor: Sport Fan Co.
        LL Mountain Tire
        ML Mountain Tire
        HL Mountain Tire

----- Products From Vendor: GMA Ski & Bike
        <<None>>

----- Products From Vendor: Integrated Sport Products
        AWC Logo Cap
        Long-Sleeve Logo Jersey, S
        Long-Sleeve Logo Jersey, M
        Long-Sleeve Logo Jersey, L
        Long-Sleeve Logo Jersey, XL
        Men's Sports Shorts, S
        Short-Sleeve Classic Jersey, S
        Short-Sleeve Classic Jersey, M
        Short-Sleeve Classic Jersey, L
        Short-Sleeve Classic Jersey, XL

----- Products From Vendor: Inline Accessories
        HL Mountain Pedal
        Lock Nut 6
        Lock Nut 16
        Lock Nut 17
        Lock Nut 7
        Lock Nut 8
        Lock Nut 9
        Lock Nut 22
        Lock Nut 23
        Lock Nut 12
        Lock Nut 13
        Lock Nut 1
        Lock Nut 10
        Lock Nut 11
        Lock Nut 2
        Lock Nut 20
        Lock Nut 21
        Lock Nut 3
        Lock Nut 14
        Lock Nut 15
        Lock Nut 4
        Lock Nut 19
        Lock Nut 18
        Lock Ring

----- Products From Vendor: Hill's Bicycle Service
        LL Road Seat/Saddle
        ML Road Seat/Saddle

----- Products From Vendor: Green Lake Bike Company
        Water Bottle - 30 oz.
        Mountain Bottle Cage
        Road Bottle Cage
        Patch Kit/8 Patches
        Hitch Rack - 4-Bike
        Bike Wash - Dissolver
        Fender Set - Mountain
        All-Purpose Bike Stand
        Hydration Pack - 70 oz.

----- Products From Vendor: Consumer Cycles
        HL Nipple

----- Products From Vendor: Merit Bikes
        <<None>>

----- Products From Vendor: Sports House
        <<None>>

----- Products From Vendor: West Junction Cycles
        HL Crankarm

----- Products From Vendor: Marsh
        <<None>>

----- Products From Vendor: Capital Road Cycles
        LL Spindle/Axle
        HL Spindle/Axle

----- Products From Vendor: Norstan Bike Hut
        Hex Nut 5
        Hex Nut 6
        Hex Nut 16
        Hex Nut 17
        Hex Nut 7
        Hex Nut 8
        Hex Nut 9
        Hex Nut 22
        Hex Nut 23
        Hex Nut 12
        Hex Nut 13
        Hex Nut 1
        Hex Nut 10
        Hex Nut 11
        Hex Nut 2
        Hex Nut 20
        Hex Nut 21
        Hex Nut 3
        Hex Nut 14
        Hex Nut 15
        Hex Nut 4
        Hex Nut 18
        Hex Nut 19

----- Products From Vendor: Illinois Trek & Clothing
        <<None>>

----- Products From Vendor: Burnett Road Warriors
        Pinch Bolt

----- Products From Vendor: Custom Frames, Inc.
        Metal Angle
        Metal Bar 1
        Metal Bar 2
```

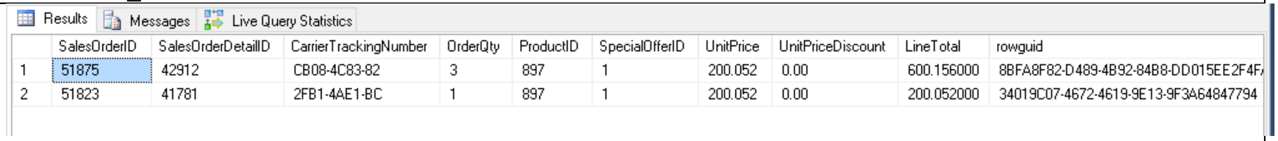
```
----- Products From Vendor: Legend Cycles
        <<None>>

----- Products From Vendor: Electronic Bike Co.
        <<None>>

----- Products From Vendor: International Sport Assoc.
        Spokes

----- Products From Vendor: Electronic Bike Repair & Supplies
        LL Road Rim
        ML Road Rim
        HL Road Rim

----- Products From Vendor: Wide World Importers
        Keyed Washer

----- Products From Vendor: American Bicycles and Wheels
        Headset Ball Bearings

----- Products From Vendor: Victory Bikes
        Road Tire Tube
        Touring Tire Tube
        HL Road Tire
        Touring Tire

----- Products From Vendor: American Bikes
        HL Road Rim

----- Products From Vendor: Crowley Sport
        LL Mountain Pedal
        ML Mountain Pedal

----- Products From Vendor: Magic Cycles
        <<None>>

----- Products From Vendor: Northern Bike Travel
        LL Nipple

----- Products From Vendor: Anderson's Custom Bikes
        Touring Rim

----- Products From Vendor: Touring Equipment Center
        <<None>>

----- Products From Vendor: Holiday Skate & Cycle
        <<None>>

----- Products From Vendor: Expert Bike Co
        External Lock Washer 3
        External Lock Washer 4
        External Lock Washer 9
        External Lock Washer 5
        External Lock Washer 7
        External Lock Washer 6
        External Lock Washer 1
        LL Touring Seat/Saddle
        ML Touring Seat/Saddle
        HL Touring Seat/Saddle

----- Products From Vendor: Varsity Sport Co.
        Chain

----- Products From Vendor: Team Athletic Co.
        Half-Finger Gloves, S
        Half-Finger Gloves, M
        Half-Finger Gloves, L
        Full-Finger Gloves, S
        Full-Finger Gloves, M
        Full-Finger Gloves, L
        Classic Vest, S
        Classic Vest, M
        Classic Vest, L
        Women's Mountain Shorts, S
        Women's Mountain Shorts, M
        Women's Mountain Shorts, L

----- Products From Vendor: Jackson Authority
        HL Mountain Rim
        LL Road Pedal
        ML Road Pedal

----- Products From Vendor: Premier Sport, Inc.
        Touring Rim

----- Products From Vendor: Professional Athletic Consultants
        LL Road Tire
        ML Road Tire
        HL Road Tire
        Touring Tire

----- Products From Vendor: Wood Fitness
        Bearing Ball
        Metal Plate 2
        Metal Plate 1
        Metal Plate 3
        Metal Sheet 2
        Metal Sheet 3
        Metal Sheet 7
        Metal Sheet 4
        Metal Sheet 5
        Metal Sheet 6
        Metal Sheet 1
        Metal Tread Plate

----- Products From Vendor: First Rate Bicycles
        LL Mountain Seat/Saddle
        ML Mountain Seat/Saddle
        HL Road Seat/Saddle

----- Products From Vendor: National Bike Association
        LL Grip Tape
        ML Grip Tape

----- Products From Vendor: Jeff's Sporting Goods
        Mountain Bike Socks, M
        Mountain Bike Socks, L
        Racing Socks, M
        Racing Socks, L

----- Products From Vendor: Superior Bicycles
        Rear Brakes

----- Products From Vendor: Bloomington Multisport
        Cup-Shaped Race

----- Products From Vendor: Carlson Specialties
        Paint - Black
        Paint - Red
        Paint - Silver
        Paint - Blue
        Paint - Yellow

----- Products From Vendor: Compete, Inc.
        Front Derailleur Cage
        Front Derailleur Linkage
        Rear Derailleur Cage

----- Products From Vendor: Chicago City Saddles
        LL Mountain Seat/Saddle
        ML Mountain Seat/Saddle
        HL Mountain Seat/Saddle
        LL Road Seat/Saddle
        ML Road Seat/Saddle
        HL Road Seat/Saddle
        LL Touring Seat/Saddle
        ML Touring Seat/Saddle
        HL Touring Seat/Saddle

----- Products From Vendor: Business Equipment Center
        Crown Race
```

SLIDE 8

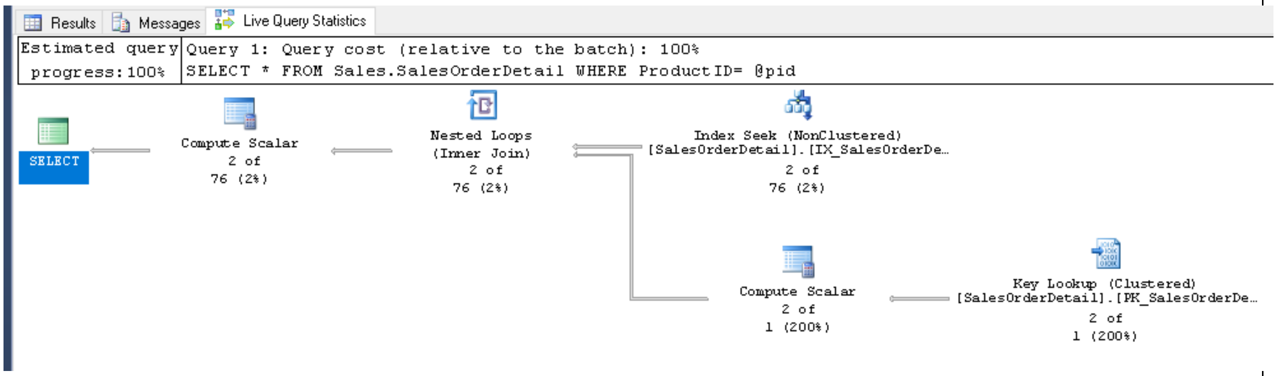
```
CREATE PROCEDURE test_1 (@pid int)
AS
SELECT * FROM Sales.SalesOrderDetail
WHERE ProductID= @pid
```

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

```
exec test_1 897
```

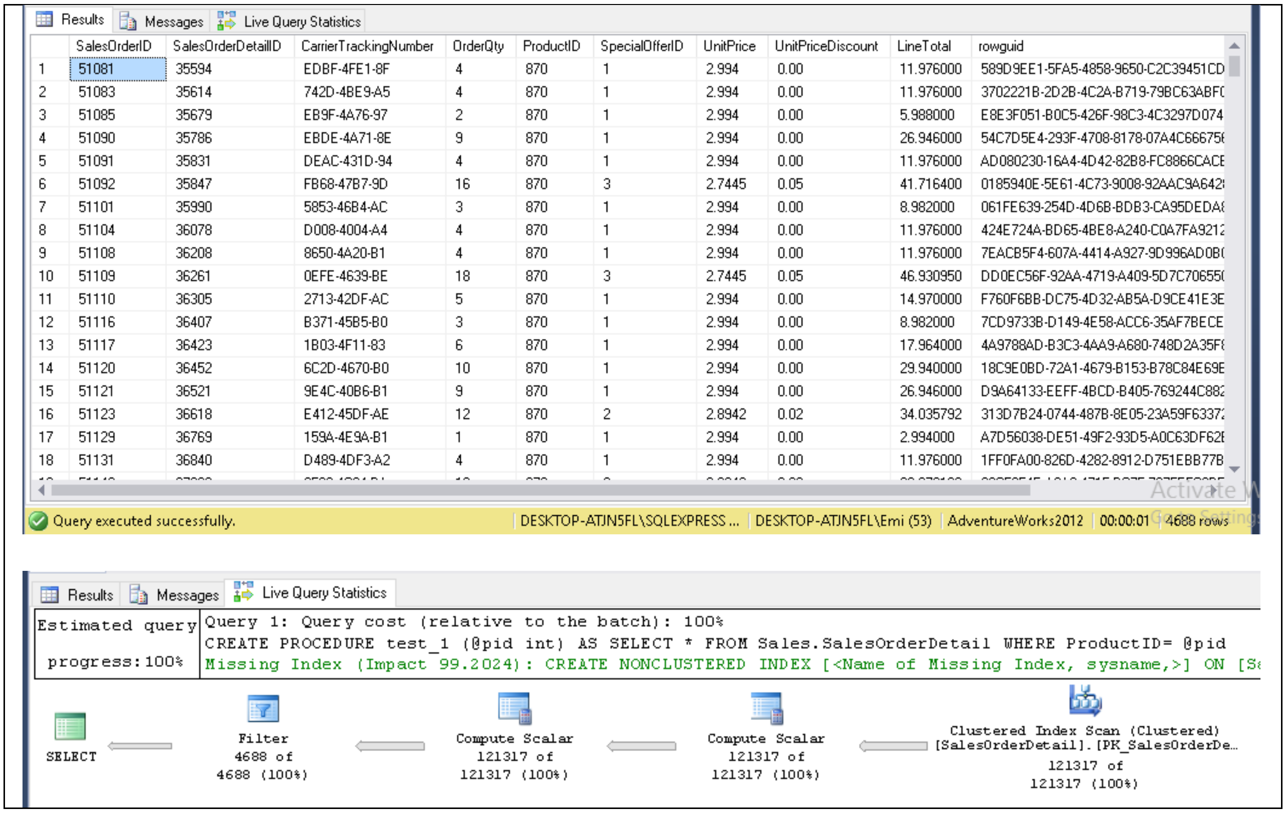


	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	51875	42912	CB08-4C83-82	3	897	1	200.052	0.00	600.156000	8BFA8F82-D489-4B92-84B8-DD015EE2F4F/
2	51823	41781	2FB1-4AE1-BC	1	897	1	200.052	0.00	200.052000	34019C07-4672-4619-9E13-9F3A64847794



```
Estimated query progress:100%
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM Sales.SalesOrderDetail WHERE ProductID= @pid
```

```
exec test_1 870
```



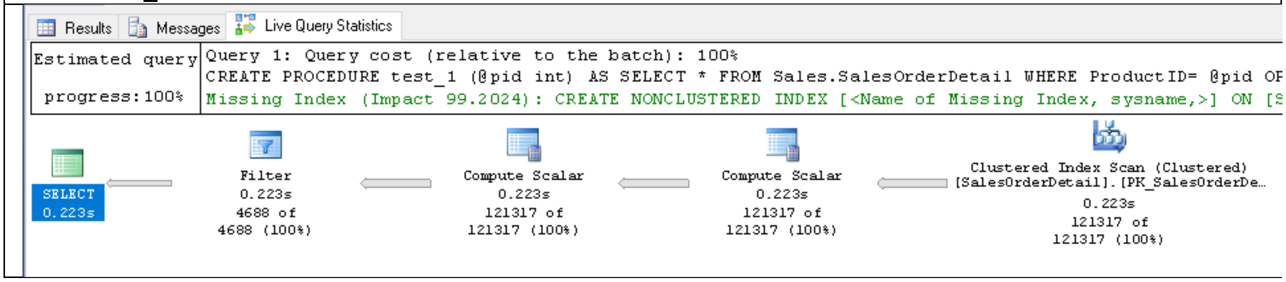
	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	51081	35594	EDBF-4FE1-8F	4	870	1	2.994	0.00	11.976000	589D9EE1-5FA5-4858-9650-C2C39451CD
2	51083	35614	742D-4BE9-A5	4	870	1	2.994	0.00	11.976000	3702221B-2D2B-4C2A-B719-79BC63ABF
3	51085	35679	EB9F-4A76-97	2	870	1	2.994	0.00	5.988000	E8E3F051-B0C5-426F-98C3-4C3297D074
4	51090	35786	EBDE-4A71-8E	9	870	1	2.994	0.00	26.946000	54C7D5E4-293F-4708-8178-07A4C66675
5	51091	35831	DEAC-431D-94	4	870	1	2.994	0.00	11.976000	AD080230-16A4-4D42-82B8-FC8866CAC
6	51092	35847	FB68-47B7-9D	16	870	3	2.7445	0.05	41.716400	0185940E-5E61-4C73-9008-92AAC9A642
7	51101	35990	5853-46B4-AC	3	870	1	2.994	0.00	8.982000	061FE639-254D-4D6B-BDB3-CA95DEDA
8	51104	36078	D008-4004-A4	4	870	1	2.994	0.00	11.976000	424E724A-BD65-4BE8-A240-C0A7FA921
9	51108	36208	8650-4A20-B1	4	870	1	2.994	0.00	11.976000	7EACB5F4-607A-4414-A927-9D996AD0B
10	51109	36261	0EFE-4639-BE	18	870	3	2.7445	0.05	46.930950	DD0EC56F-92AA-4719-A409-5D7C70655
11	51110	36305	2713-42DF-AC	5	870	1	2.994	0.00	14.970000	F760F6BB-DC75-4D32-AB5A-D9CE41E3E
12	51116	36407	B371-45B5-B0	3	870	1	2.994	0.00	8.982000	7CD9733B-D149-4E58-ACC6-35AF7BECE
13	51117	36423	1B03-4F11-83	6	870	1	2.994	0.00	17.964000	4A9788AD-B3C3-4AA9-A680-748D2A35F
14	51120	36452	6C2D-4670-B0	10	870	1	2.994	0.00	29.940000	18C9E0BD-72A1-4679-B153-B78C84E69E
15	51121	36521	9E4C-40B6-B1	9	870	1	2.994	0.00	26.946000	D9A64133-EEFF-4BCD-B405-769244C88
16	51123	36618	E412-45DF-AE	12	870	2	2.8942	0.02	34.035792	313D7B24-0744-487B-8E05-23A59F6337
17	51129	36769	159A-4E9A-B1	1	870	1	2.994	0.00	2.994000	A7D56038-DE51-49F2-93D5-A0C63DF62
18	51131	36840	D489-4DF3-A2	4	870	1	2.994	0.00	11.976000	1FF0FA00-826D-4282-8912-D751EBB77B

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:01 | 4688 rows

```
Estimated query   Query 1: Query cost (relative to the batch): 100%
                  CREATE PROCEDURE test_1 (@pid int) AS SELECT * FROM Sales.SalesOrderDetail WHERE ProductID= @pid
  progress:100%   Missing Index (Impact 99.2024): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [Sa
```

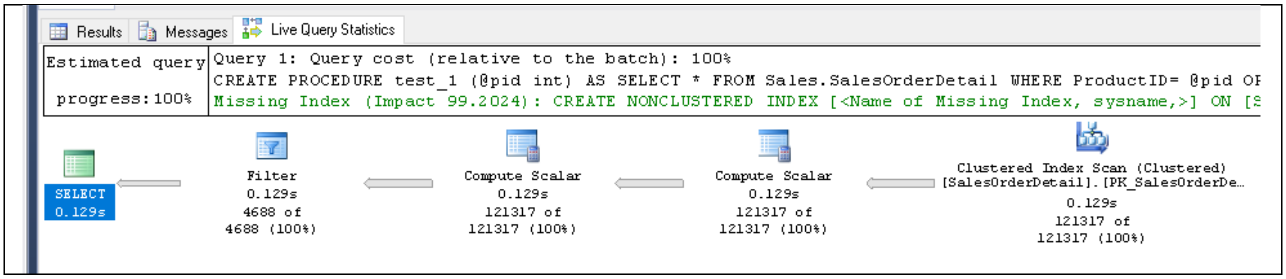
SLIDE 9

```
ALTER PROCEDURE test_1 (@pid int)
AS
SELECT * FROM Sales.SalesOrderDetail
WHERE ProductID= @pid
OPTION (OPTIMIZE FOR (@pid= 870))

exec test_1 870
```



```
Estimated query   Query 1: Query cost (relative to the batch): 100%
                  CREATE PROCEDURE test_1 (@pid int) AS SELECT * FROM Sales.SalesOrderDetail WHERE ProductID= @pid OF
  progress:100%   Missing Index (Impact 99.2024): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [S
```

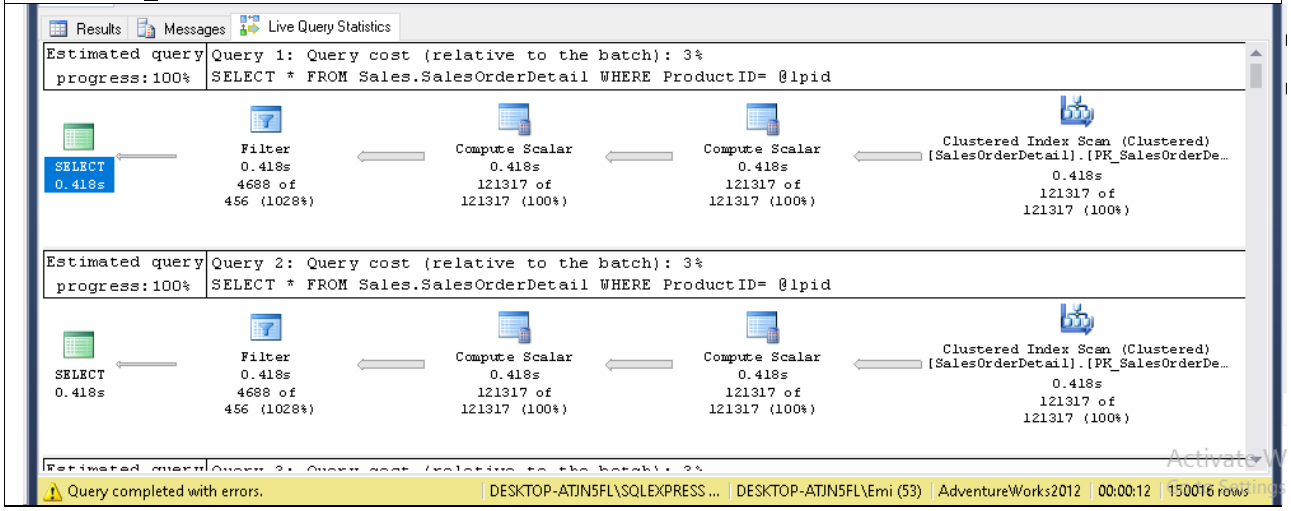
```
ALTER PROCEDURE test_1 (@pid int)
AS
SELECT * FROM Sales.SalesOrderDetail
WHERE ProductID= @pid
OPTION (RECOMPILE)

exec test_1 870
```



```
Estimated query progress:100%
Query 1: Query cost (relative to the batch): 100%
CREATE PROCEDURE test_1 (@pid int) AS SELECT * FROM Sales.SalesOrderDetail WHERE ProductID= @pid OF
Missing Index (Impact 99.2024): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [S
```

SLIDE 10

```
-- optimize for UKNOWN

ALTER PROCEDURE test_1 (@pid int)
AS
DECLARE @lpid int
SET @lpid= @pid
SELECT * FROM Sales.SalesOrderDetail
WHERE ProductID= @lpid

exec test_1 870
```



```
Estimated query progress:100%
Query 1: Query cost (relative to the batch): 3%
SELECT * FROM Sales.SalesOrderDetail WHERE ProductID= @lpid
```

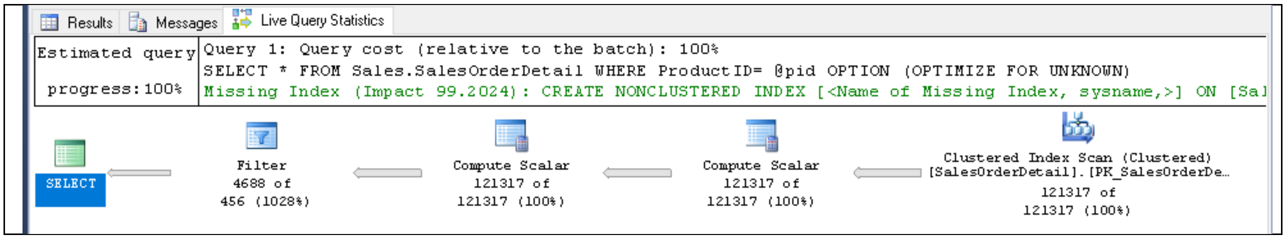
```
Estimated query progress:100%
Query 2: Query cost (relative to the batch): 3%
SELECT * FROM Sales.SalesOrderDetail WHERE ProductID= @lpid
```

```
Estimated query Query 3: Query cost (relative to the batch): 3%
```

```
Query completed with errors. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:12 | 150016 rows
```

SLIDE 11

```
ALTER PROCEDURE test_1 (@pid int)
AS
SELECT * FROM Sales.SalesOrderDetail
WHERE ProductID= @pid
OPTION (OPTIMIZE FOR UNKNOWN)

exec test_1 870
```

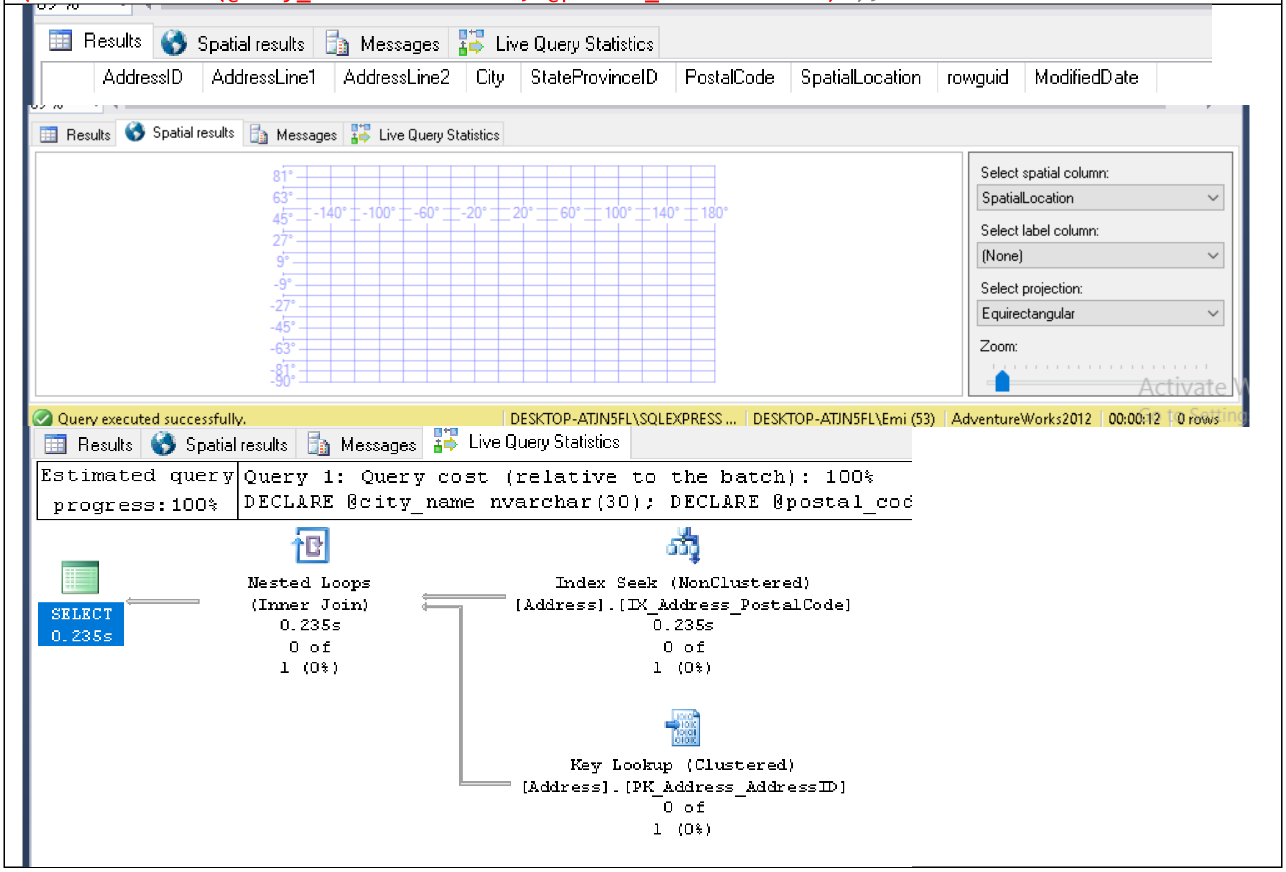


```
Estimated query   Query 1: Query cost (relative to the batch): 100%
                  SELECT * FROM Sales.SalesOrderDetail WHERE ProductID= @pid OPTION (OPTIMIZE FOR UNKNOWN)
 progress:100%    Missing Index (Impact 99.2024): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [Sa
```

```
SELECT    Filter          Compute Scalar     Compute Scalar     Clustered Index Scan (Clustered)
          4688 of         121317 of          121317 of          [SalesOrderDetail].[PK_SalesOrderDe…
          456 (1028%)     121317 (100%)      121317 (100%)      121317 of
                                                                121317 (100%)
```

SLIDE 12

```
-- optimize for
DECLARE @city_name nvarchar(30);
DECLARE @postal_code nvarchar(15);

SELECT * FROM Person.Address
WHERE City = @city_name AND PostalCode= @postal_code OPTION
(OPTIMIZE FOR (@city_name= 'Seattle', @postal_code UNKNOWN) );
```



	AddressID	AddressLine1	AddressLine2	City	StateProvinceID	PostalCode	SpatialLocation	rowguid	ModifiedDate

Select spatial column: SpatialLocation

Select label column: (None)

Select projection: Equirectangular

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:12 | 0 rows

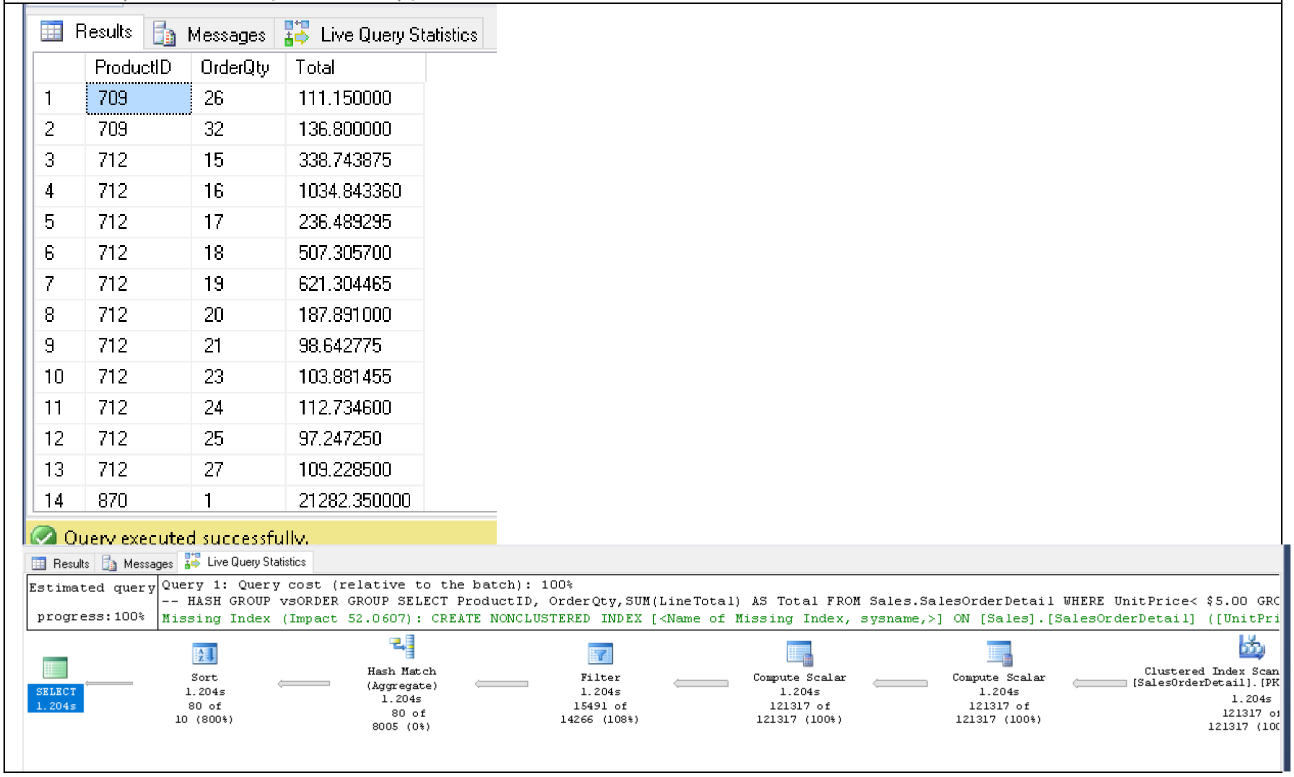
```
Estimated query   Query 1: Query cost (relative to the batch): 100%
 progress:100%    DECLARE @city_name nvarchar(30); DECLARE @postal_coc
```

```
SELECT     Nested Loops       Index Seek (NonClustered)
0.235s     (Inner Join)       [Address].[IX_Address_PostalCode]
           0.235s             0.235s
           0 of               0 of
           1 (0%)             1 (0%)

                              Key Lookup (Clustered)
                              [Address].[PK_Address_AddressID]
                              0 of
                              1 (0%)
```

SLIDE 13

```
-- HASH GROUP vsORDER GROUP
SELECT ProductID, OrderQty,SUM(LineTotal) AS Total
FROM Sales.SalesOrderDetail
WHERE UnitPrice< $5.00
```

```
GROUP BY ProductID, OrderQty
ORDER BY ProductID, OrderQty
OPTION (HASH GROUP, FAST 10);
```



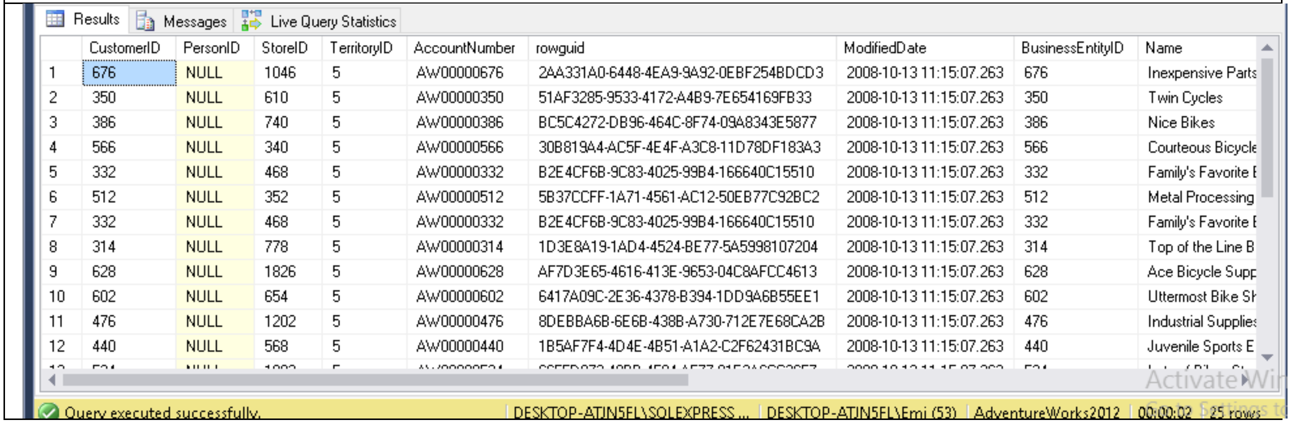
	ProductID	OrderQty	Total
1	709	26	111.150000
2	709	32	136.800000
3	712	15	338.743875
4	712	16	1034.843360
5	712	17	236.489295
6	712	18	507.305700
7	712	19	621.304465
8	712	20	187.891000
9	712	21	98.642775
10	712	23	103.881455
11	712	24	112.734600
12	712	25	97.247250
13	712	27	109.228500
14	870	1	21282.350000

Query executed successfully.

```
Estimated query progress:100%
Query 1: Query cost (relative to the batch): 100%
-- HASH GROUP vsORDER GROUP SELECT ProductID, OrderQty,SUM(LineTotal) AS Total FROM Sales.SalesOrderDetail WHERE UnitPrice< $5.00 GRC
Missing Index (Impact 52.0607): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [Sales].[SalesOrderDetail] ([UnitPri
```

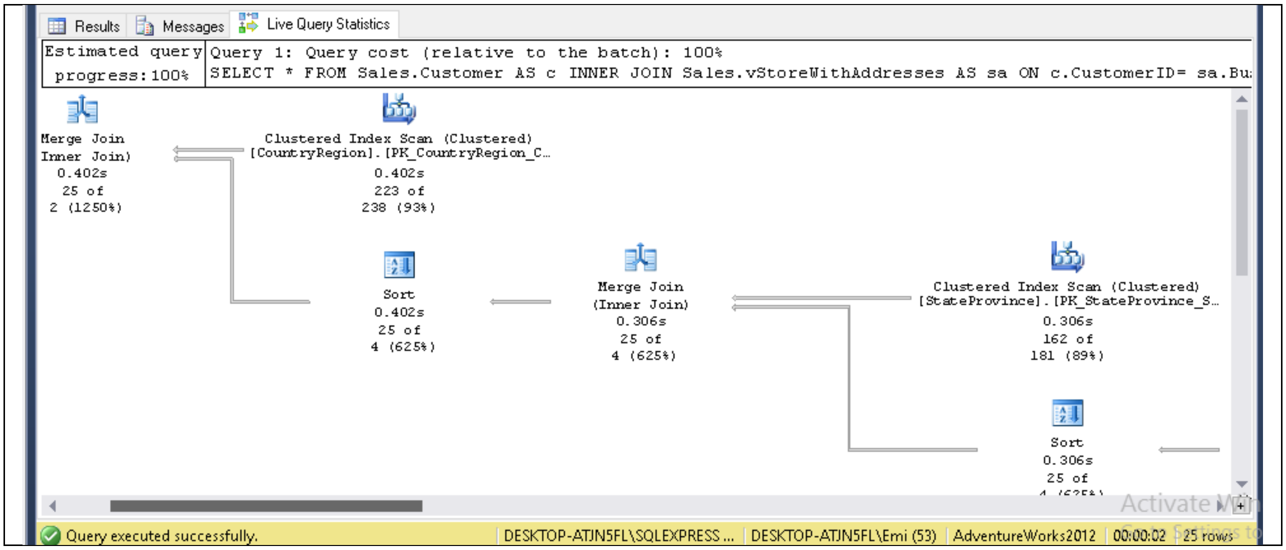
SLIDE 15

```
SELECT * FROM Sales.Customer AS c
INNER JOIN Sales.vStoreWithAddresses AS sa ON c.CustomerID= sa.BusinessEntityID
WHERE TerritoryID= 5
OPTION (MERGE JOIN);
GO
```



	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	rowguid	ModifiedDate	BusinessEntityID	Name
1	676	NULL	1046	5	AW00000676	2AA331A0-6448-4EA9-9A92-0EBF254BDCD3	2008-10-13 11:15:07.263	676	Inexpensive Parts
2	350	NULL	610	5	AW00000350	51AF3285-9533-4172-A4B9-7E654169FB33	2008-10-13 11:15:07.263	350	Twin Cycles
3	386	NULL	740	5	AW00000386	BC5C4272-DB96-464C-8F74-09A8343E5877	2008-10-13 11:15:07.263	386	Nice Bikes
4	566	NULL	340	5	AW00000566	30B819A4-AC5F-4E4F-A3C8-11D78DF183A3	2008-10-13 11:15:07.263	566	Courteous Bicycle
5	332	NULL	468	5	AW00000332	B2E4CF6B-9C83-4025-99B4-166640C15510	2008-10-13 11:15:07.263	332	Family's Favorite E
6	512	NULL	352	5	AW00000512	5B37CCFF-1A71-4561-AC12-50EB77C92BC2	2008-10-13 11:15:07.263	512	Metal Processing
7	332	NULL	468	5	AW00000332	B2E4CF6B-9C83-4025-99B4-166640C15510	2008-10-13 11:15:07.263	332	Family's Favorite E
8	314	NULL	778	5	AW00000314	1D3E8A19-1AD4-4524-BE77-5A5998107204	2008-10-13 11:15:07.263	314	Top of the Line B
9	628	NULL	1826	5	AW00000628	AF7D3E65-4616-413E-9653-04C8AFCC4613	2008-10-13 11:15:07.263	628	Ace Bicycle Supp
10	602	NULL	654	5	AW00000602	6417A09C-2E36-4378-B394-1DD9A6B55EE1	2008-10-13 11:15:07.263	602	Uttermost Bike Sh
11	476	NULL	1202	5	AW00000476	8DEBBA6B-6E6B-438B-A730-712E7E68CA2B	2008-10-13 11:15:07.263	476	Industrial Supplies
12	440	NULL	568	5	AW00000440	1B5AF7F4-4D4E-4B51-A1A2-C2F62431BC9A	2008-10-13 11:15:07.263	440	Juvenile Sports E

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:02 | 25 rows

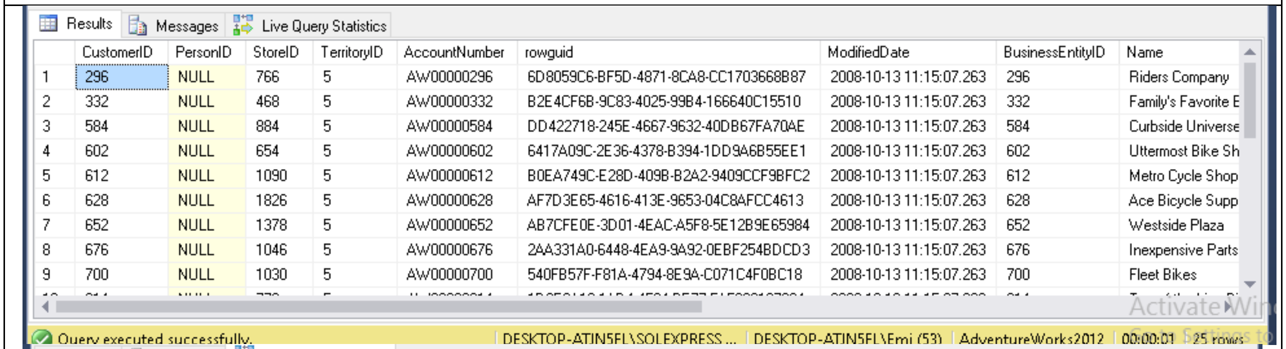


```
Estimated query progress:100%
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM Sales.Customer AS c INNER JOIN Sales.vStoreWithAddresses AS sa ON c.CustomerID= sa.Bu
```

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:02 | 25 rows

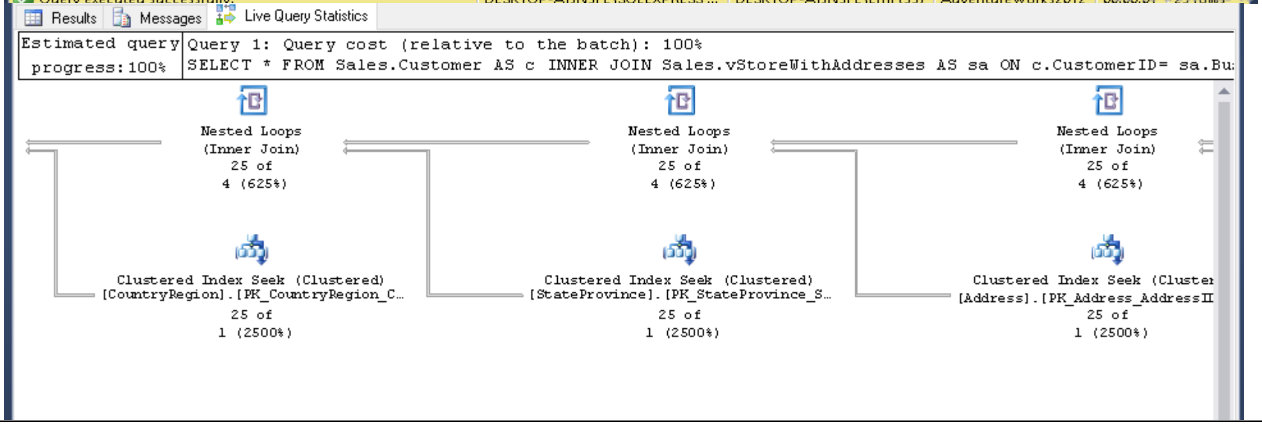
SLIDE 16

```
SELECT * FROM Sales.Customer AS c
INNER JOIN Sales.vStoreWithAddresses AS sa ON c.CustomerID= sa.BusinessEntityID
WHERE TerritoryID= 5
OPTION (FAST 10);
GO
```



	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	rowguid	ModifiedDate	BusinessEntityID	Name
1	296	NULL	766	5	AW00000296	6D8059C6-BF5D-4871-8CA8-CC1703668B87	2008-10-13 11:15:07.263	296	Riders Company
2	332	NULL	468	5	AW00000332	B2E4CF6B-9C83-4025-99B4-166640C15510	2008-10-13 11:15:07.263	332	Family's Favorite E
3	584	NULL	884	5	AW00000584	DD422718-245E-4667-9632-40DB67FA70AE	2008-10-13 11:15:07.263	584	Curbside Universe
4	602	NULL	654	5	AW00000602	6417A09C-2E36-4378-B394-1DD9A6B55EE1	2008-10-13 11:15:07.263	602	Uttermost Bike Sh
5	612	NULL	1090	5	AW00000612	B0EA749C-E28D-409B-B2A2-9409CCF9BFC2	2008-10-13 11:15:07.263	612	Metro Cycle Shop
6	628	NULL	1826	5	AW00000628	AF7D3E65-4616-413E-9653-04C8AFCC4613	2008-10-13 11:15:07.263	628	Ace Bicycle Supp
7	652	NULL	1378	5	AW00000652	AB7CFE0E-3D01-4EAC-A5F8-5E12B9E65984	2008-10-13 11:15:07.263	652	Westside Plaza
8	676	NULL	1046	5	AW00000676	2AA331A0-6448-4EA9-9A92-0EBF254BDCD3	2008-10-13 11:15:07.263	676	Inexpensive Parts
9	700	NULL	1030	5	AW00000700	540FB57F-F81A-4794-8E9A-C071C4F0BC18	2008-10-13 11:15:07.263	700	Fleet Bikes

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:01 | 25 rows



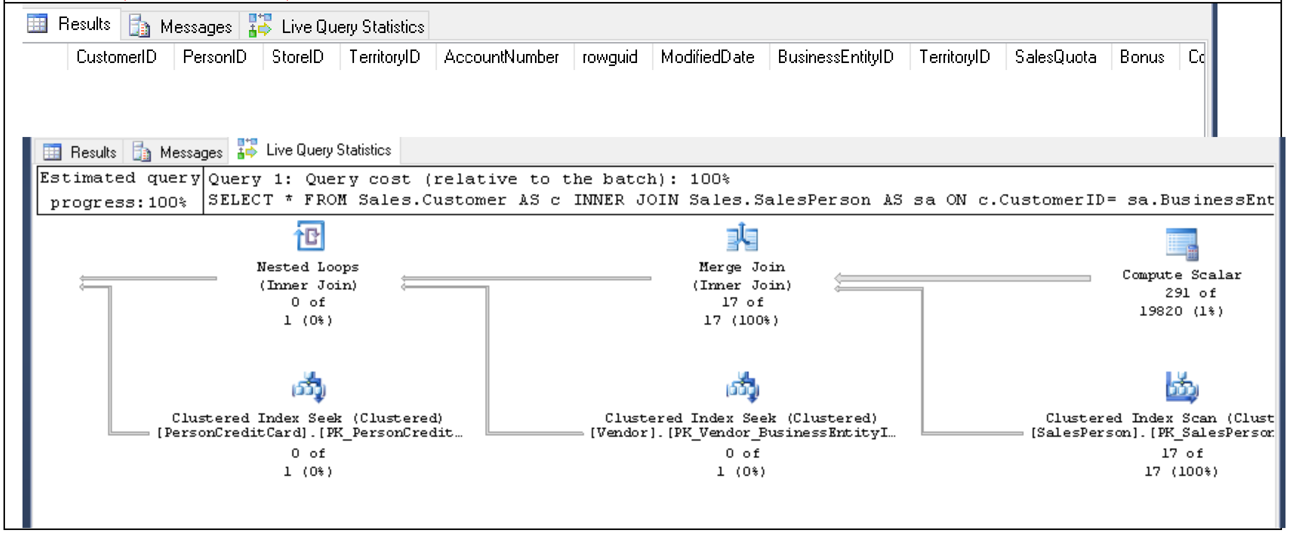
Results
Messages
Live Query Statistics

```
Estimated query progress:100%
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM Sales.Customer AS c INNER JOIN Sales.vStoreWithAddresses AS sa ON c.CustomerID= sa.Bu
```

Nested Loops
(Inner Join)
25 of
4 (625%)
Nested Loops
(Inner Join)
25 of
4 (625%)
Nested Loops
(Inner Join)
25 of
4 (625%)
Clustered Index Seek (Clustered)
[CountryRegion].[PK_CountryRegion_C...
25 of
1 (2500%)
Clustered Index Seek (Clustered)
[StateProvince].[PK_StateProvince_S...
25 of
1 (2500%)
Clustered Index Seek (Cluster
[Address].[PK_Address_AddressII
25 of
1 (2500%)

SLIDE 17

```
SELECT * FROM Sales.Customer AS c
INNER JOIN Sales.SalesPerson AS sa ON c.CustomerID= sa.BusinessEntityID
INNER JOIN Purchasing.Vendor AS v ON v.BusinessEntityID = c.CustomerID
INNER JOIN Sales.PersonCreditCard AS p ON p.BusinessEntityID = v.BusinessEntityID
OPTION (FORCE ORDER);
```



	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	rowguid	ModifiedDate	BusinessEntityID	TerritoryID	SalesQuota	Bonus	Co

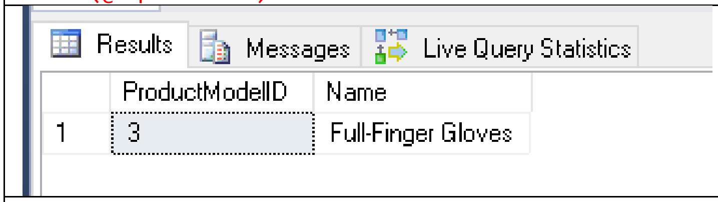
```
Estimated query progress:100%
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM Sales.Customer AS c INNER JOIN Sales.SalesPerson AS sa ON c.CustomerID= sa.BusinessEnt
```

Nested Loops
(Inner Join)
0 of
1 (0%)
Merge Join
(Inner Join)
17 of
17 (100%)
Compute Scalar
291 of
19820 (1%)
Clustered Index Seek (Clustered)
[PersonCreditCard].[PK_PersonCredit...
0 of
1 (0%)
Clustered Index Seek (Clustered)
[Vendor].[PK_Vendor_BusinessEntityI...
0 of
1 (0%)
Clustered Index Scan (Clust
[SalesPerson].[PK_SalesPerson
17 of
17 (100%)

SLIDE 19

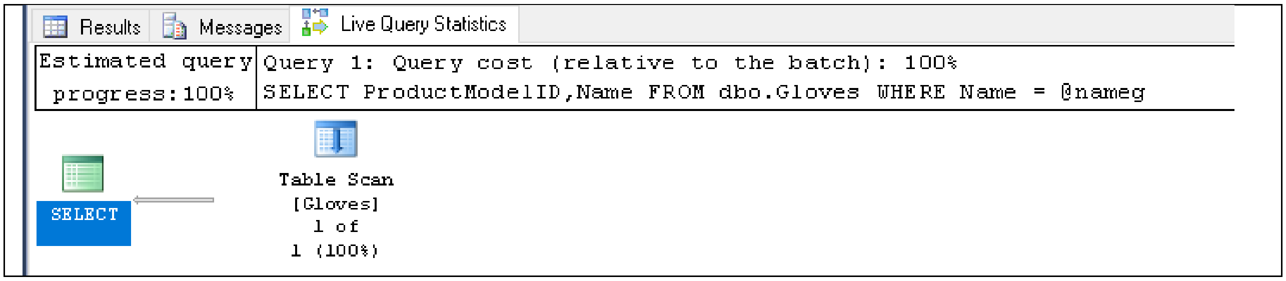
Dynamic execution

```
DECLARE @sqlCommand varchar(1000)
DECLARE @columnList varchar(75)
DECLARE @nameg nvarchar(50)
SET @columnList = 'ProductModelID,Name'
SET @nameg = '''Full-Finger Gloves'''
SET @sqlCommand = 'SELECT ' + @columnList + ' FROM dbo.Gloves WHERE Name=' + @nameg
EXEC (@sqlCommand)
```



	ProductModelID	Name
1	3	Full-Finger Gloves

```
DECLARE @sqlCommand nvarchar(1000)
DECLARE @columnList nvarchar(75)
DECLARE @nameg nvarchar(50)
SET @columnList = 'ProductModelID,Name'
SET @nameg = 'Full-Finger Gloves'
SET @sqlCommand = 'SELECT ' + @columnList + ' FROM dbo.Gloves WHERE Name = @nameg'
EXECUTE sp_executesql @sqlCommand, N'@nameg nvarchar(50)', @nameg = @nameg
```



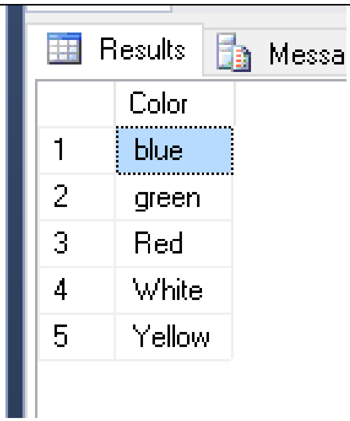
```
Estimated query
 progress:100%
```

```
Query 1: Query cost (relative to the batch): 100%
SELECT ProductModelID,Name FROM dbo.Gloves WHERE Name = @nameg
```

SLIDE 20

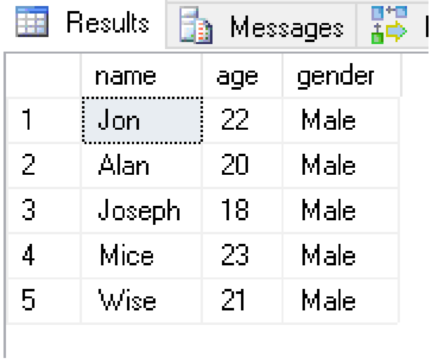
Temporary tables

```
-- temporary tables
CREATE table #Color(Color varchar(10) PRIMARY key)

INSERT INTO #color
SELECT 'Red'
 UNION SELECT 'White'
 UNION SELECT 'green'
 UNION SELECT'Yellow'
 UNION SELECT'blue'

SELECT * FROM #Color

DROP TABLE #Color
go
```



	Color
1	blue
2	green
3	Red
4	White
5	Yellow

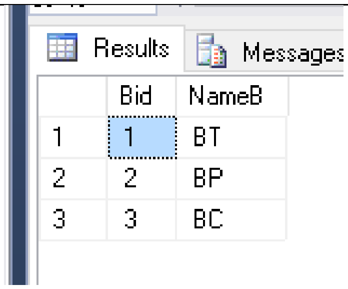
```
CREATE TABLE student
(
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    gender VARCHAR(50) NOT NULL,
    age INT NOT NULL,
    total_score INT NOT NULL,
)

INSERT INTO student
VALUES (1, 'Jolly', 'Female', 20, 500),
(2, 'Jon', 'Male', 22, 545),
(3, 'Sara', 'Female', 25, 600),
(4, 'Laura', 'Female', 18, 400),
(5, 'Alan', 'Male', 20, 500),
(6, 'Kate', 'Female', 22, 500),
(7, 'Joseph', 'Male', 18, 643),
(8, 'Mice', 'Male', 23, 543),
(9, 'Wise', 'Male', 21, 499),
(10, 'Elis', 'Female', 27, 400);

 CREATE TABLE #MaleStudents
(
       name VARCHAR(50),
       age int,
       gender VARCHAR (50)
)
```



	name	age	gender
1	Jon	22	Male
2	Alan	20	Male
3	Joseph	18	Male
4	Mice	23	Male
5	Wise	21	Male

```
 INSERT INTO #MaleStudents SELECT name,
age, gender FROM student WHERE gender =
'Male'
select * from #MaleStudents
```

```
-- example 3
CREATE table #Bank(
       Bid int primary key identity(1,1),
       NameB varchar(50)
)
INSERT INTO #Bank(NameB) VALUES ('BT'),
('BP'), ('BC')
SELECT * FROM #Bank
DROP TABLE #Bank
go
```



	Bid	NameB
1	1	BT
2	2	BP
3	3	BC

SLIDE 21

Triggers

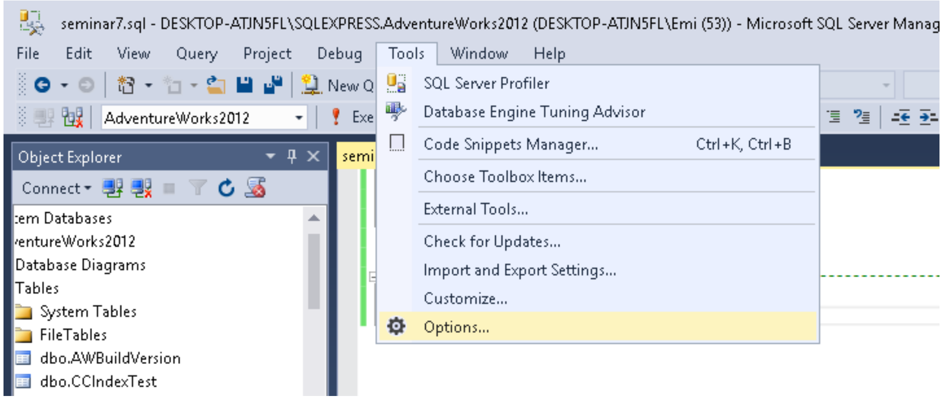
```
-- trigger with a reminder message
CREATE TRIGGER reminder1
ON Sales.Customer
AFTER INSERT, UPDATE
AS RAISERROR ('Notify Customer Relations', 16, 10);
GO
```

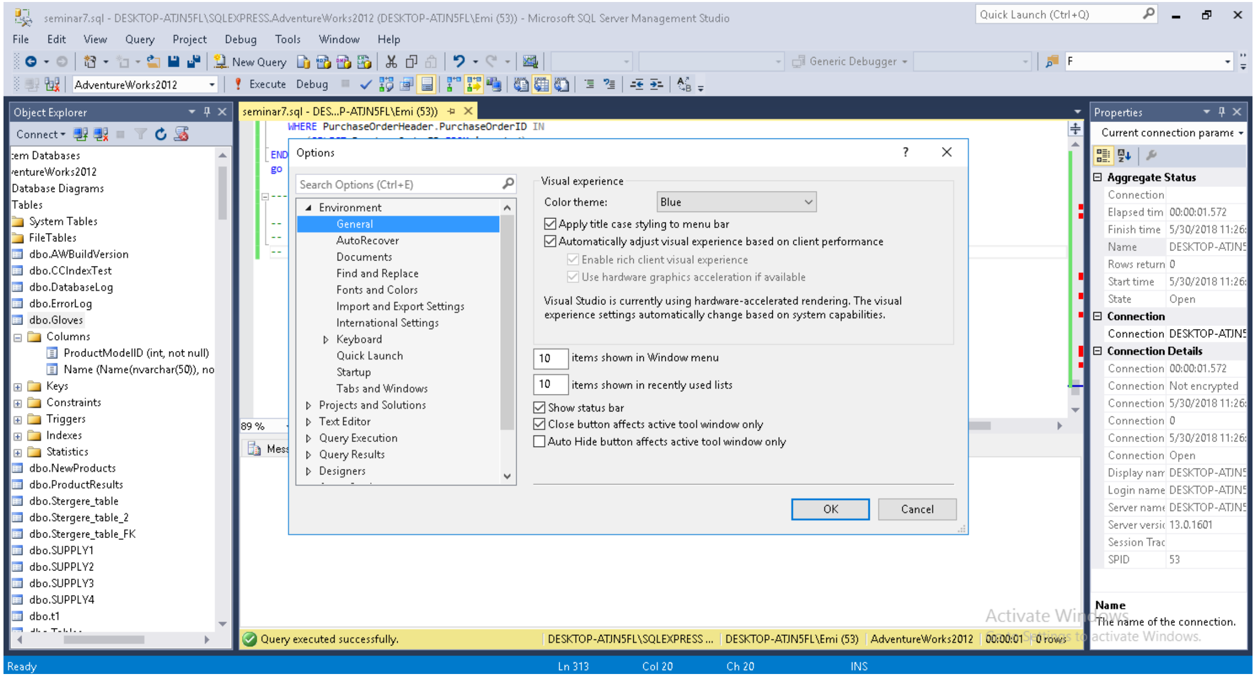
```
-- trigger with reminder e-mail message
CREATE TRIGGER reminder2
ON Sales.Customer
AFTER INSERT, UPDATE, DELETE
AS
   EXEC msdb.dbo.sp_send_dbmail
        @profile_name = 'AdventureWorks2012 Administrator',
        @recipients = 'danw@Adventure-Works.com',
        @body = 'Don''t forget to print a report for the sales force.',
        @subject = 'Reminder';
GO
```

```
-- Trigger valid for multirow and single row inserts
-- and optimal for single row inserts.
USE AdventureWorks2012;
GO
CREATE TRIGGER NewPODetail3
ON Purchasing.PurchaseOrderDetail
FOR INSERT AS
IF @@ROWCOUNT = 1
BEGIN
   UPDATE Purchasing.PurchaseOrderHeader
   SET SubTotal = SubTotal + LineTotal
   FROM inserted
   WHERE PurchaseOrderHeader.PurchaseOrderID = inserted.PurchaseOrderID

END
ELSE
BEGIN
      UPDATE Purchasing.PurchaseOrderHeader
   SET SubTotal = SubTotal +
      (SELECT SUM(LineTotal)
      FROM inserted
      WHERE PurchaseOrderHeader.PurchaseOrderID
       = inserted.PurchaseOrderID)
   WHERE PurchaseOrderHeader.PurchaseOrderID IN
      (SELECT PurchaseOrderID FROM inserted)
END;
Go
```

SLIDE 22

Server Options





Visual experience

Color theme: Blue

☑ Apply title case styling to menu bar

☑ Automatically adjust visual experience based on client performance

☑ Enable rich client visual experience

☑ Use hardware graphics acceleration if available

Visual Studio is currently using hardware-accelerated rendering. The visual experience settings automatically change based on system capabilities.

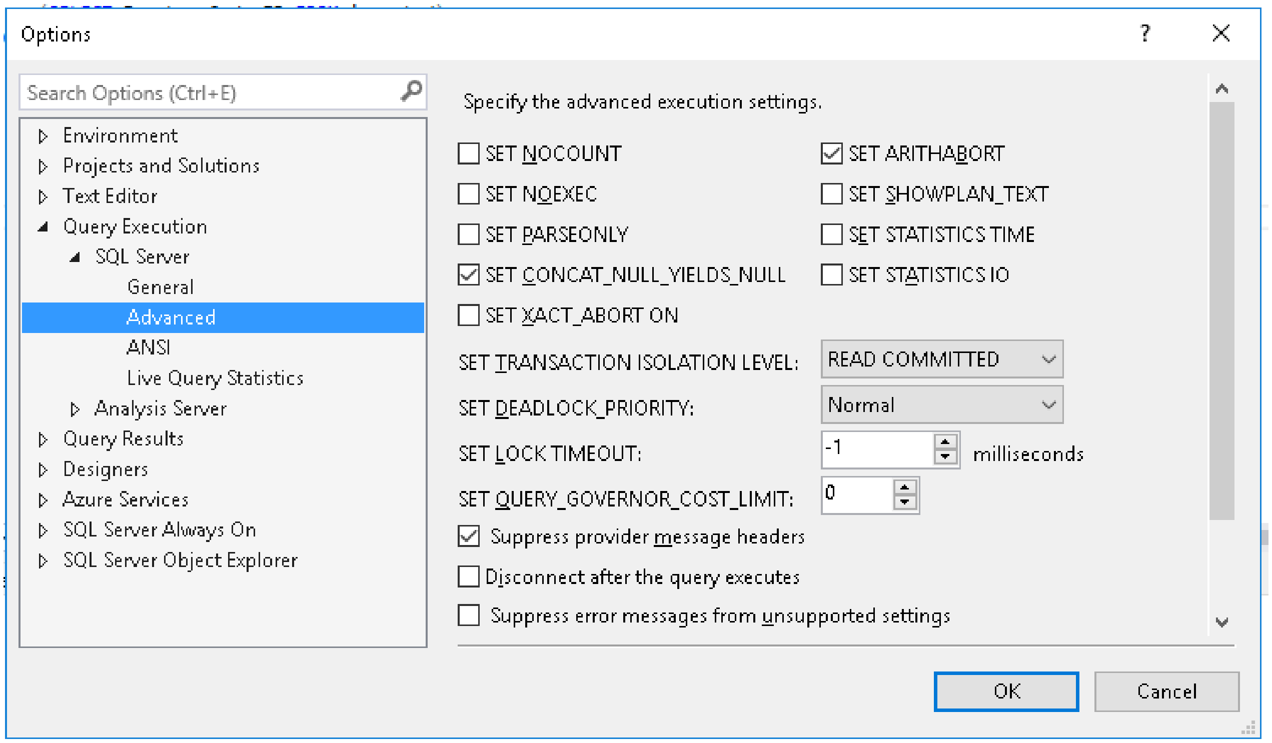
10 items shown in Window menu

10 items shown in recently used lists

☑ Show status bar

☑ Close button affects active tool window only

☐ Auto Hide button affects active tool window only



Specify the advanced execution settings.

☐ SET NOCOUNT
☑ SET ARITHABORT
☐ SET NOEXEC
☐ SET SHOWPLAN_TEXT
☐ SET PARSEONLY
☐ SET STATISTICS TIME
☑ SET CONCAT_NULL_YIELDS_NULL
☐ SET STATISTICS IO
☐ SET XACT_ABORT ON

SET TRANSACTION ISOLATION LEVEL: READ COMMITTED

SET DEADLOCK_PRIORITY: Normal

SET LOCK TIMEOUT: -1 milliseconds

SET QUERY_GOVERNOR_COST_LIMIT: 0

☑ Suppress provider message headers
☐ Disconnect after the query executes
☐ Suppress error messages from unsupported settings

SLIDE 23

Fragmentare

DBCC SHOWCONTIG

```
-- Displaying fragmentation information for a table - the Employee table.
USE AdventureWorks2012;
GO
DBCC SHOWCONTIG ('HumanResources.Employee');
GO
```

```
DBCC SHOWCONTIG scanning 'Employee' table...
Table: 'Employee' (1237579447); index ID: 1, database ID: 7
TABLE level scan performed.
- Pages Scanned................................: 7
- Extents Scanned..............................: 1
- Extent Switches..............................: 0
- Avg. Pages per Extent........................: 7.0
- Scan Density [Best Count:Actual Count].......: 100.00% [1:1]
- Logical Scan Fragmentation ..................: 0.00%
- Extent Scan Fragmentation ...................: 0.00%
- Avg. Bytes Free per Page.....................: 715.1
- Avg. Page Density (full).....................: 91.16%
DBCC execution completed. If DBCC printed error messages, contact your system
administrator.
```

```
-- uses OBJECT_ID and the sys.indexes catalog view to obtain the table ID and index ID
for the AK_Product_Name index of the Production.Product table in the AdventureWorks2012
database.
```

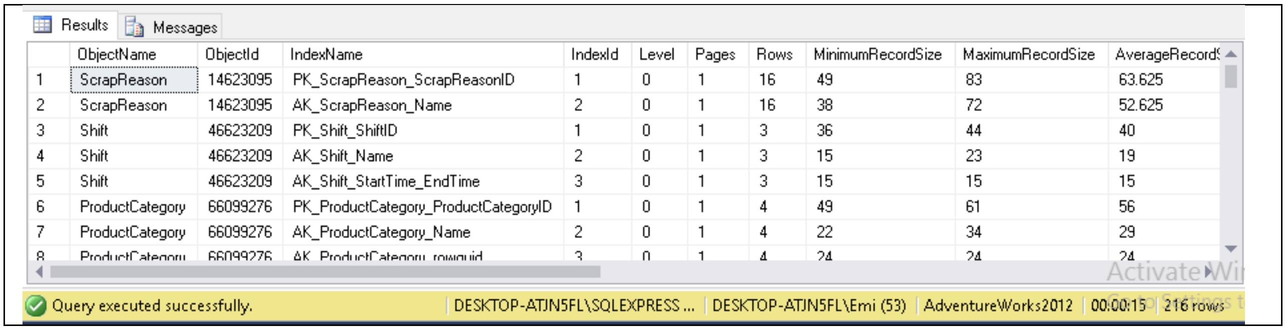
```
USE AdventureWorks2012;
GO
DECLARE @id int, @indid int
SET @id = OBJECT_ID('Production.Product')
SELECT @indid = index_id
FROM sys.indexes
WHERE object_id = @id
   AND name = 'AK_Product_Name'
DBCC SHOWCONTIG (@id, @indid);
GO
```

```
DBCC SHOWCONTIG scanning 'Product' table...
Table: 'Product' (1973582069); index ID: 3, database ID: 7
LEAF level scan performed.
- Pages Scanned................................: 4
- Extents Scanned..............................: 3
- Extent Switches..............................: 2
- Avg. Pages per Extent........................: 1.3
- Scan Density [Best Count:Actual Count].......: 33.33% [1:3]
- Logical Scan Fragmentation ..................: 75.00%
- Extent Scan Fragmentation ...................: 66.67%
- Avg. Bytes Free per Page.....................: 1684.5
- Avg. Page Density (full).....................: 79.19%
DBCC execution completed. If DBCC printed error messages, contact your system
administrator.
```

```
-- returns an abbreviated result set for the Product table in the AdventureWorks2012
database.
USE AdventureWorks2012;
GO
DBCC SHOWCONTIG ('Production.Product', 1) WITH FAST;
GO
```

```
DBCC SHOWCONTIG scanning 'Product' table...
Table: 'Product' (1973582069); index ID: 1, database ID: 7
TABLE level scan performed.
- Pages Scanned................................: 13
- Extent Switches..............................: 2
- Scan Density [Best Count:Actual Count].......: 66.67% [2:3]
- Logical Scan Fragmentation ..................: 15.38%
DBCC execution completed. If DBCC printed error messages, contact your system
administrator.
```

```
-- returns a full table result set for every index on every table in the
AdventureWorks2012 database.
USE AdventureWorks2012;
GO
DBCC SHOWCONTIG WITH TABLERESULTS, ALL_INDEXES;
GO
```

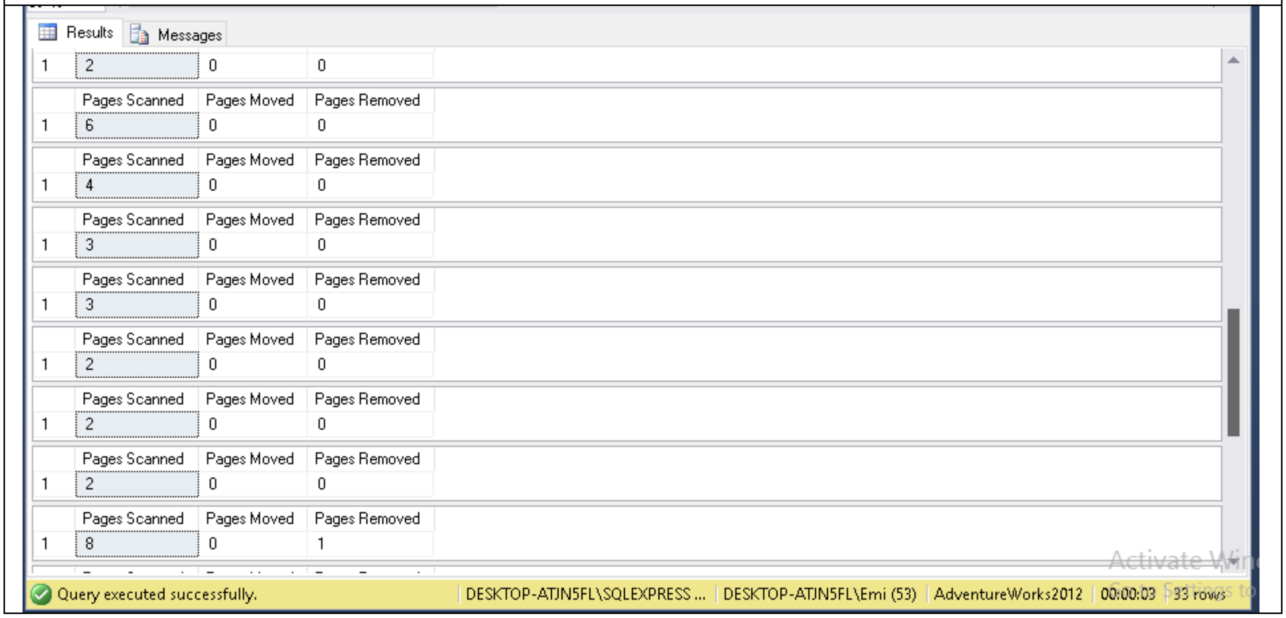


	ObjectName	ObjectId	IndexName	IndexId	Level	Pages	Rows	MinimumRecordSize	MaximumRecordSize	AverageRecord
1	ScrapReason	14623095	PK_ScrapReason_ScrapReasonID	1	0	1	16	49	83	63.625
2	ScrapReason	14623095	AK_ScrapReason_Name	2	0	1	16	38	72	52.625
3	Shift	46623209	PK_Shift_ShiftID	1	0	1	3	36	44	40
4	Shift	46623209	AK_Shift_Name	2	0	1	3	15	23	19
5	Shift	46623209	AK_Shift_StartTime_EndTime	3	0	1	3	15	15	15
6	ProductCategory	66099276	PK_ProductCategory_ProductCategoryID	1	0	1	4	49	61	56
7	ProductCategory	66099276	AK_ProductCategory_Name	2	0	1	4	22	34	29
8	ProductCategory	66099276	AK_ProductCategory_rowguid	3	0	1	4	24	24	24

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:15 | 216 rows

```
-- shows a simple way to defragment all indexes in a database that is fragmented above a
declared threshold.
/*Perform a 'USE <database name>' to select the database in which to run the script.*/
-- Declare variables
SET NOCOUNT ON;
DECLARE @tablename varchar(255);
DECLARE @execstr   varchar(400);
DECLARE @objectid  int;
DECLARE @indexid   int;
DECLARE @frag      decimal;
DECLARE @maxfrag   decimal;
-- Decide on the maximum fragmentation to allow for.
SELECT @maxfrag = 30.0;
-- Declare a cursor.
DECLARE tables CURSOR FOR
   SELECT TABLE_SCHEMA + '.' + TABLE_NAME
   FROM INFORMATION_SCHEMA.TABLES
   WHERE TABLE_TYPE = 'BASE TABLE';
-- Create the table.
CREATE TABLE #fraglist (
   ObjectName char(255),  ObjectId int,  IndexName char(255),  IndexId int,  Lvl int,
CountPages int,  CountRows int,  MinRecSize int,  MaxRecSize int,  AvgRecSize int,
ForRecCount int,  Extents int, ExtentSwitches int,  AvgFreeBytes int,  AvgPageDensity
int,     ScanDensity decimal,  BestCount int,  ActualCount int,  LogicalFrag decimal,
ExtentFrag decimal);
-- Open the cursor.
OPEN tables;
-- Loop through all the tables in the database.
FETCH NEXT
   FROM tables
   INTO @tablename;
WHILE @@FETCH_STATUS = 0
BEGIN
-- Do the showcontig of all indexes of the table
   INSERT INTO #fraglist
   EXEC ('DBCC SHOWCONTIG (''' + @tablename + ''')
      WITH FAST, TABLERESULTS, ALL_INDEXES, NO_INFOMSGS');
   FETCH NEXT
      FROM tables
      INTO @tablename;
END;
-- Close and deallocate the cursor.
CLOSE tables;
DEALLOCATE tables;
```

```sql
-- Declare the cursor for the list of indexes to be defragged.
DECLARE indexes CURSOR FOR
   SELECT ObjectName, ObjectId, IndexId, LogicalFrag
   FROM #fraglist
   WHERE LogicalFrag >= @maxfrag
      AND INDEXPROPERTY (ObjectId, IndexName, 'IndexDepth') > 0;
-- Open the cursor.
OPEN indexes;
-- Loop through the indexes.
FETCH NEXT
   FROM indexes
   INTO @tablename, @objectid, @indexid, @frag;
WHILE @@FETCH_STATUS = 0
BEGIN
   PRINT 'Executing DBCC INDEXDEFRAG (0, ' + RTRIM(@tablename) + ',
      ' + RTRIM(@indexid) + ') - fragmentation currently '
       + RTRIM(CONVERT(varchar(15),@frag)) + '%';
   SELECT @execstr = 'DBCC INDEXDEFRAG (0, ' + RTRIM(@objectid) + ',
       ' + RTRIM(@indexid) + ')';
   EXEC (@execstr);
FETCH NEXT
      FROM indexes
      INTO @tablename, @objectid, @indexid, @frag;
END;
-- Close and deallocate the cursor.
CLOSE indexes;
DEALLOCATE indexes;
-- Delete the temporary table.
DROP TABLE #fraglist;
GO
```

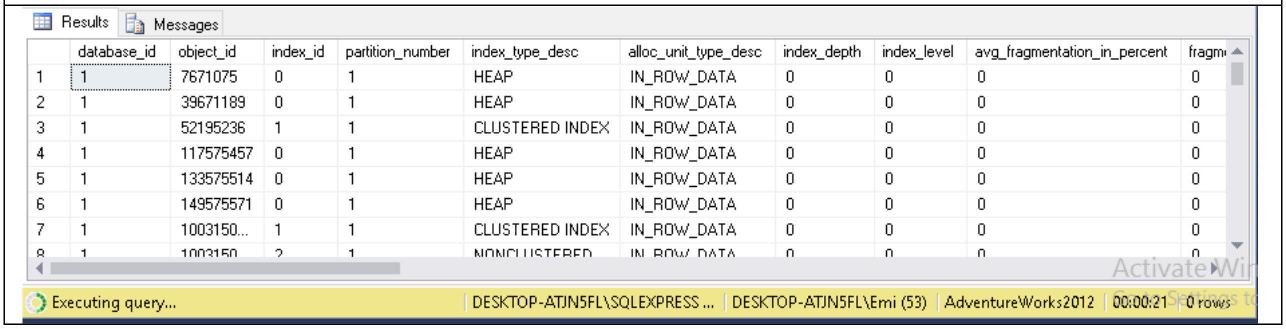


```
1   2                0             0

    Pages Scanned    Pages Moved   Pages Removed
1   6                0             0

    Pages Scanned    Pages Moved   Pages Removed
1   4                0             0

    Pages Scanned    Pages Moved   Pages Removed
1   3                0             0

    Pages Scanned    Pages Moved   Pages Removed
1   3                0             0

    Pages Scanned    Pages Moved   Pages Removed
1   2                0             0

    Pages Scanned    Pages Moved   Pages Removed
1   2                0             0

    Pages Scanned    Pages Moved   Pages Removed
1   2                0             0

    Pages Scanned    Pages Moved   Pages Removed
1   8                0             1

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:03 | 33 rows
```

sys.dm_db_index_physical_stats

```sql
use master
```

```
go
-- In this example, OBJECT_ID is evaluated in the context of the master database.
-- Because Person.Address does not exist in master, the function returns NULL.
-- When NULL is specified as an object_id, all objects in the database are returned.
-- The same results are returned when an object that is not valid is specified.
SELECT * FROM sys.dm_db_index_physical_stats
    (DB_ID(N'AdventureWorks'), OBJECT_ID(N'Person.Address'), NULL, NULL , 'DETAILED');
GO
-- This example demonstrates the results of specifying a valid object name that exists in
both the current database context and
-- in the database specified in the database_id parameter of the
sys.dm_db_index_physical_stats function.
-- An error is returned because the ID value returned by OBJECT_ID does not match the ID
value of the object in the specified database.
CREATE DATABASE Test;
GO
USE Test;
GO
CREATE SCHEMA Person;
GO
CREATE Table Person.Address(c1 int);
GO
USE AdventureWorks2012;
GO
SELECT * FROM sys.dm_db_index_physical_stats
    (DB_ID(N'Test'), OBJECT_ID(N'Person.Address'), NULL, NULL , 'DETAILED');
GO
-- Clean up temporary database.
DROP DATABASE Test;
GO
```

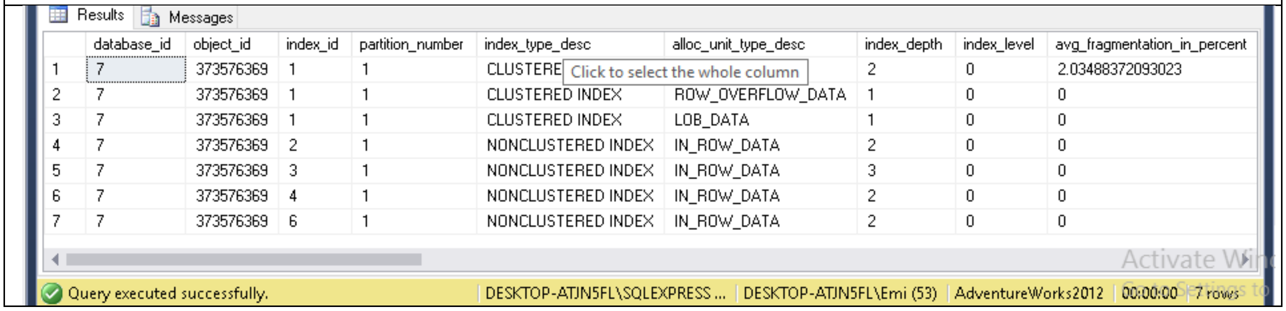


	database_id	object_id	index_id	partition_number	index_type_desc	alloc_unit_type_desc	index_depth	index_level	avg_fragmentation_in_percent	fragm
1	1	7671075	0	1	HEAP	IN_ROW_DATA	0	0	0	0
2	1	39671189	0	1	HEAP	IN_ROW_DATA	0	0	0	0
3	1	52195236	1	1	CLUSTERED INDEX	IN_ROW_DATA	0	0	0	0
4	1	117575457	0	1	HEAP	IN_ROW_DATA	0	0	0	0
5	1	133575514	0	1	HEAP	IN_ROW_DATA	0	0	0	0
6	1	149575571	0	1	HEAP	IN_ROW_DATA	0	0	0	0
7	1	1003150...	1	1	CLUSTERED INDEX	IN_ROW_DATA	0	0	0	0
8	1	1003150	2	1	NONCLUSTERED	IN_ROW_DATA	0	0	0	0

Executing query... | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:21 | 0 rows

```
-- Returning information about a specified table
-- returns size and fragmentation statistics for all indexes and partitions of the
Person.Address table.
-- The scan mode is set to 'LIMITED' for best performance and to limit the statistics
that are returned.
-- Executing this query requires, at a minimum, CONTROL permission on the Person.Address
table.
DECLARE @db_id SMALLINT;
DECLARE @object_id INT;

SET @db_id = DB_ID(N'AdventureWorks2012');
SET @object_id = OBJECT_ID(N'AdventureWorks2012.Person.Address');

IF @db_id IS NULL
BEGIN;
```

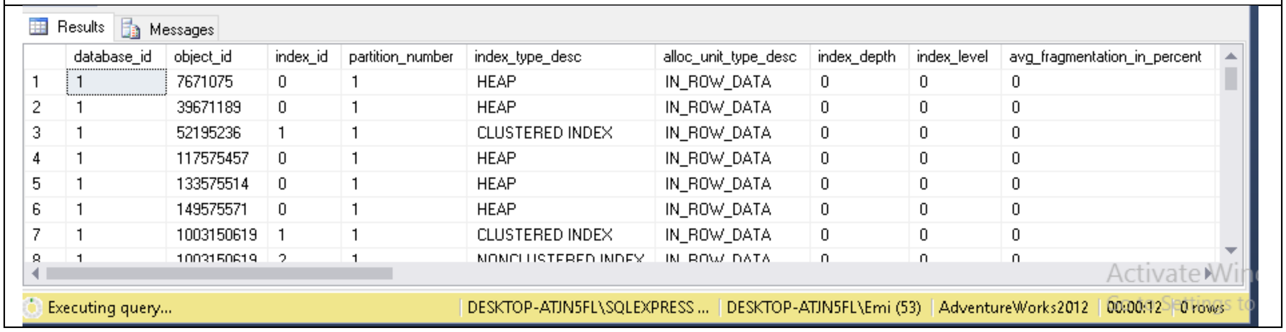
```
    PRINT N'Invalid database';
END;
ELSE IF @object_id IS NULL
BEGIN;
    PRINT N'Invalid object';
END;
ELSE
BEGIN;
    SELECT * FROM sys.dm_db_index_physical_stats(@db_id, @object_id, NULL, NULL ,
'LIMITED');
END;
GO
```



	database_id	object_id	index_id	partition_number	index_type_desc	alloc_unit_type_desc	index_depth	index_level	avg_fragmentation_in_percent
1	7	373576369	1	1	CLUSTERE		2	0	2.03488372093023
2	7	373576369	1	1	CLUSTERED INDEX	ROW_OVERFLOW_DATA	1	0	0
3	7	373576369	1	1	CLUSTERED INDEX	LOB_DATA	1	0	0
4	7	373576369	2	1	NONCLUSTERED INDEX	IN_ROW_DATA	2	0	0
5	7	373576369	3	1	NONCLUSTERED INDEX	IN_ROW_DATA	3	0	0
6	7	373576369	4	1	NONCLUSTERED INDEX	IN_ROW_DATA	2	0	0
7	7	373576369	6	1	NONCLUSTERED INDEX	IN_ROW_DATA	2	0	0

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:00 | 7 rows

```
-- Returning information for all databases
-- returns all statistics for all tables and indexes within the instance of SQL Server
-- by specifying the wildcard NULL for all parameters. Executing this query requires the
VIEW SERVER STATE permission.
SELECT * FROM sys.dm_db_index_physical_stats (NULL, NULL, NULL, NULL, NULL);
GO
```

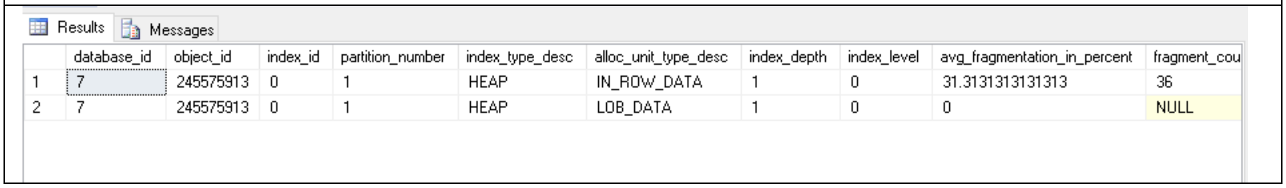


	database_id	object_id	index_id	partition_number	index_type_desc	alloc_unit_type_desc	index_depth	index_level	avg_fragmentation_in_percent
1	1	7671075	0	1	HEAP	IN_ROW_DATA	0	0	0
2	1	39671189	0	1	HEAP	IN_ROW_DATA	0	0	0
3	1	52195236	1	1	CLUSTERED INDEX	IN_ROW_DATA	0	0	0
4	1	117575457	0	1	HEAP	IN_ROW_DATA	0	0	0
5	1	133575514	0	1	HEAP	IN_ROW_DATA	0	0	0
6	1	149575571	0	1	HEAP	IN_ROW_DATA	0	0	0
7	1	1003150619	1	1	CLUSTERED INDEX	IN_ROW_DATA	0	0	0
8	1	1003150619	2	1	NONCLUSTERED INDEX	IN_ROW_DATA	0	0	0

Executing query... | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (53) | AdventureWorks2012 | 00:00:12 | 0 rows

```
-- Returning information about a heap
-- returns all statistics for the heap dbo.DatabaseLog in the AdventureWorks2012
database.
-- Because the table contains LOB data, a row is returned for the LOB_DATA allocation
unit
-- in addition to the row returned for the IN_ROW_ALLOCATION_UNIT that is storing the
data pages of the heap.
-- Executing this query requires, at a minimum, CONTROL permission on the dbo.DatabaseLog
table.
DECLARE @db_id SMALLINT;
DECLARE @object_id INT;
SET @db_id = DB_ID(N'AdventureWorks2012');
SET @object_id = OBJECT_ID(N'AdventureWorks2012.dbo.DatabaseLog');
IF @object_id IS NULL
```

```
BEGIN;
    PRINT N'Invalid object';
END;
ELSE
BEGIN;
    SELECT * FROM sys.dm_db_index_physical_stats(@db_id, @object_id, 0, NULL ,
'DETAILED');
END;
GO
```

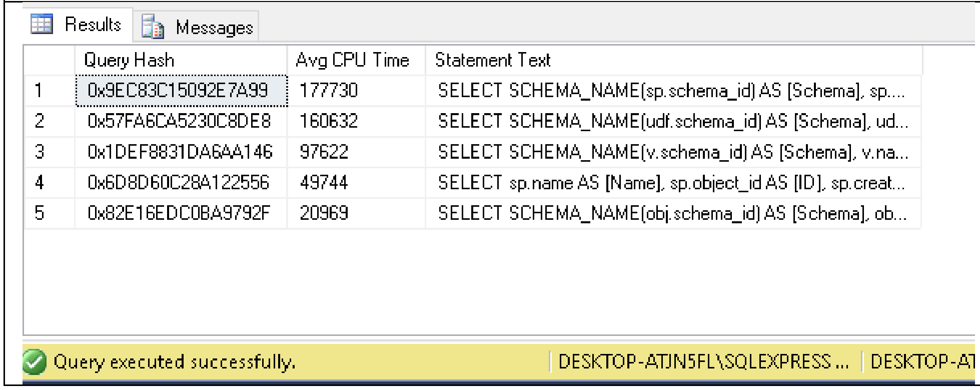


	database_id	object_id	index_id	partition_number	index_type_desc	alloc_unit_type_desc	index_depth	index_level	avg_fragmentation_in_percent	fragment_cou
1	7	245575913	0	1	HEAP	IN_ROW_DATA	1	0	31.3131313131313	36
2	7	245575913	0	1	HEAP	LOB_DATA	1	0	0	NULL

SLIDE 25

sys.dm_exec_query_stats

```
-- Finding the TOP N queries
-- returns information about the top five queries ranked by average CPU time.
-- This example aggregates the queries according to their query hash so that logically
equivalent queries
-- are grouped by their cumulative resource consumption.
SELECT TOP 5 query_stats.query_hash AS "Query Hash",
    SUM(query_stats.total_worker_time) / SUM(query_stats.execution_count) AS "Avg CPU
Time",
    MIN(query_stats.statement_text) AS "Statement Text"
FROM
    (SELECT QS.*,
    SUBSTRING(ST.text, (QS.statement_start_offset/2) + 1,
    ((CASE statement_end_offset
        WHEN -1 THEN DATALENGTH(ST.text)
        ELSE QS.statement_end_offset END
            - QS.statement_start_offset)/2) + 1) AS statement_text
     FROM sys.dm_exec_query_stats AS QS
     CROSS APPLY sys.dm_exec_sql_text(QS.sql_handle) as ST) as query_stats
GROUP BY query_stats.query_hash
ORDER BY 2 DESC;
```

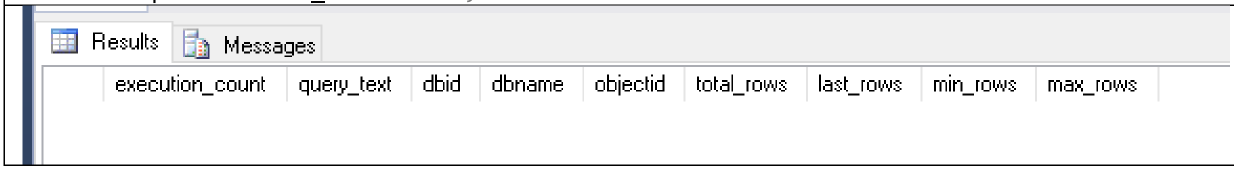


	Query Hash	Avg CPU Time	Statement Text
1	0x9EC83C15092E7A99	177730	SELECT SCHEMA_NAME(sp.schema_id) AS [Schema], sp....
2	0x57FA6CA5230C8DE8	160632	SELECT SCHEMA_NAME(udf.schema_id) AS [Schema], ud...
3	0x1DEF8831DA6AA146	97622	SELECT SCHEMA_NAME(v.schema_id) AS [Schema], v.na...
4	0x6D8D60C28A122556	49744	SELECT sp.name AS [Name], sp.object_id AS [ID], sp.creat...
5	0x82E16EDC0BA9792F	20969	SELECT SCHEMA_NAME(obj.schema_id) AS [Schema], ob...

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-AT

```
-- Returning row count aggregates for a query
```

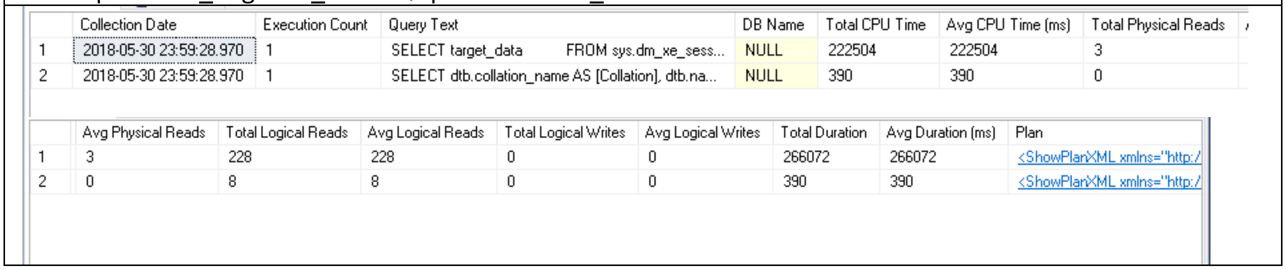
```sql
-- The following example returns row count aggregate information
-- (total rows, minimum rows, maximum rows and last rows) for queries.
SELECT qs.execution_count,
    SUBSTRING(qt.text,qs.statement_start_offset/2 +1,
                 (CASE WHEN qs.statement_end_offset = -1
                       THEN LEN(CONVERT(nvarchar(max), qt.text)) * 2
                       ELSE qs.statement_end_offset end -
                            qs.statement_start_offset
                 )/2
             ) AS query_text,
     qt.dbid, dbname= DB_NAME (qt.dbid), qt.objectid,
     qs.total_rows, qs.last_rows, qs.min_rows, qs.max_rows
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt
WHERE qt.text like '%SELECT%'
ORDER BY qs.execution_count DESC;
```



	execution_count	query_text	dbid	dbname	objectid	total_rows	last_rows	min_rows	max_rows

```sql
-- example
SELECT TOP 20
    GETDATE() AS "Collection Date",
    qs.execution_count AS "Execution Count",
    SUBSTRING(qt.text,qs.statement_start_offset/2 +1,
                 (CASE WHEN qs.statement_end_offset = -1
                       THEN LEN(CONVERT(NVARCHAR(MAX), qt.text)) * 2
                       ELSE qs.statement_end_offset END -
                            qs.statement_start_offset
                 )/2
             ) AS "Query Text",
     DB_NAME(qt.dbid) AS "DB Name",
     qs.total_worker_time AS "Total CPU Time",
     qs.total_worker_time/qs.execution_count AS "Avg CPU Time (ms)",
     qs.total_physical_reads AS "Total Physical Reads",
     qs.total_physical_reads/qs.execution_count AS "Avg Physical Reads",
     qs.total_logical_reads AS "Total Logical Reads",
     qs.total_logical_reads/qs.execution_count AS "Avg Logical Reads",
     qs.total_logical_writes AS "Total Logical Writes",
     qs.total_logical_writes/qs.execution_count AS "Avg Logical Writes",
     qs.total_elapsed_time AS "Total Duration",
     qs.total_elapsed_time/qs.execution_count AS "Avg Duration (ms)",
     qp.query_plan AS "Plan"
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt
CROSS APPLY sys.dm_exec_query_plan(qs.plan_handle) AS qp
WHERE
     qs.execution_count > 50 OR
     qs.total_worker_time/qs.execution_count > 100 OR
     qs.total_physical_reads/qs.execution_count > 1000 OR
     qs.total_logical_reads/qs.execution_count > 1000 OR
     qs.total_logical_writes/qs.execution_count > 1000 OR
     qs.total_elapsed_time/qs.execution_count > 1000
ORDER BY
```

```
      qs.execution_count DESC,
      qs.total_elapsed_time/qs.execution_count DESC,
      qs.total_worker_time/qs.execution_count DESC,
      qs.total_physical_reads/qs.execution_count DESC,
      qs.total_logical_reads/qs.execution_count DESC,
      qs.total_logical_writes/qs.execution_count DESC
```



	Collection Date	Execution Count	Query Text	DB Name	Total CPU Time	Avg CPU Time (ms)	Total Physical Reads
1	2018-05-30 23:59:28.970	1	SELECT target_data FROM sys.dm_xe_sess...	NULL	222504	222504	3
2	2018-05-30 23:59:28.970	1	SELECT dtb.collation_name AS [Collation], dtb.na...	NULL	390	390	0

	Avg Physical Reads	Total Logical Reads	Avg Logical Reads	Total Logical Writes	Avg Logical Writes	Total Duration	Avg Duration (ms)	Plan
1	3	228	228	0	0	266072	266072	<ShowPlanXML xmlns="http:/
2	0	8	8	0	0	390	390	<ShowPlanXML xmlns="http:/

References:

http://www.cnblogs.com/princessd8251/p/3679519.html

http://codingsight.com/grouping-data-using-the-over-and-partition-by-functions/

https://msdn.microsoft.com/ro-ro/library/ms190227(v=sql.120).aspx

https://msdn.microsoft.com/da-dk/library/ms190752(v=sql.120).aspx

https://docs.microsoft.com/pl-pl/previous-versions/sql/sql-server-2012/ms190227(v=sql.110)

https://docs.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-showcontig-transact-sql?view=sql-server-2017

http://157.56.25.106/en-us/library/ms188917(SQL.100).aspx

https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-exec-query-stats-transact-sql?view=sql-server-2017

https://www.mssqltips.com/sqlservertip/2602/collecting-and-storing-poor-performing-sql-server-queries-for-analysis/

https://gist.github.com/ezgigurkan/53394de28b278232d173