

Create a Join Query

Wednesday, May 24, 2017 1:49 PM

Create Joining-Query

Query Designer

The Query Designer window displays a join query between five tables: Customers, Orders, Order Details, Products, and Shippers. The diagram shows the relationships between the tables and the selected columns for the query.

| Column | Alias | Table | Outp... | Sort Type | Sort Order | Filter | Or... | Or... | Or... |
|--------------|---------|---------------|-------------------------------------|-----------|------------|--------|-------|-------|-------|
| CompanyNa... | | Customers | <input checked="" type="checkbox"/> | | | | | | |
| Country | | Customers | <input checked="" type="checkbox"/> | | | | | | |
| OrderDate | | Orders | <input checked="" type="checkbox"/> | | | | | | |
| RequiredDate | | Orders | <input checked="" type="checkbox"/> | | | | | | |
| UnitPrice | | [Order Det... | <input checked="" type="checkbox"/> | | | | | | |
| Quantity | | [Order Det... | <input checked="" type="checkbox"/> | | | | | | |
| ProductName | | Products | <input checked="" type="checkbox"/> | | | | | | |
| CompanyNa... | Shipper | Shippers | <input checked="" type="checkbox"/> | | | | | | |

SELECT Customers.CompanyName, Customers.Country, Orders.OrderDate, Orders.RequiredDate, [Order Details].UnitPrice, [Order Details].Quantity, Products.ProductName, Shippers.CompanyName AS Shipper
FROM Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID INNER JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID INNER JOIN Products ON [Order Details].ProductID = Products.ProductID INNER JOIN Shippers ON Orders.ShipVia = Shippers.ShipperID

OK Cancel

☐ Helps with Concurrency and Identity

When multiple users are executing the plan concurrently, there needs to be a structure that holds data that is related to their individual executions of the plan.

- Multiple users may execute a plan concurrently
- Execution plan details how the query would be executed
 - Does not hold data for a specific execution
- Execution context holds the data associated with a specific execution of the plan
 - One execution context is needed for each concurrent execution
 - Execution contexts are also cached
 - Not all execution contexts are identical

SQL Server can choose to Join in various Ways - Depending on what it sees

☐ Nested Joins, Merge Joins, Hash Match

Nested Loops **1**

- Complexity: $O(N \log M)$
- Used usually when one table is significantly small
- The larger table has an index which allows seeking it using the join key

Merge Join **2**

- Complexity: $O(N+M)$
- Both inputs are sorted on the join key
- An equality operator is used
- Excellent for very large tables

Hash Match **3**

- Complexity: $O(N \cdot h_c + M \cdot h_m + J)$
- Last-resort join type
- Uses a hash table and a dynamic hash match function to match rows

☐ Aggregation

• Stream Aggregate **1**

- Highly efficient
- Data is already in the correct order for processing the aggregate

• Hash Match Aggregate **2**

- Hash table is used to form the aggregate because the data is not in the necessary order

Join Operations

Saturday, February 9, 2019 6:29 PM

```
02-JoinOptionPlans...-OFFICE\delip (56)  X 03-ShowPlan.sql - (...-OFFICE\delip (55))*
1  Use Northwind;
2  Go
3
4  SELECT Customers.CompanyName, Orders.OrderDate, Products.ProductName, [Order Details].UnitPrice
5  FROM    Orders INNER JOIN
6          Customers ON Orders.CustomerID = Customers.CustomerID INNER JOIN
7          [Order Details] ON Orders.OrderID = [Order Details].OrderID INNER JOIN
8          Products ON [Order Details].ProductID = Products.ProductID INNER JOIN
9          Categories ON Products.CategoryID = Categories.CategoryID INNER JOIN
10         Suppliers ON Products.SupplierID = Suppliers.SupplierID
11  Where Customers.Country = 'USA'
12  OPTION (Merge Join)
13  --OPTION (Hash Join)
14  --OPTION (Loop Join)
```

Simple Execution Plans

Wednesday, May 24, 2017 7:03 PM

☐ Create Table - NO Primary Key

```
ExecutionPlan.sql - ...ls (LABS\delip (57))* X ShowPlan.sql - (loc...d (LABS\delip (57))
1  USE Schools;
2  GO
3
4  --Step One
5  -----
6  CREATE TABLE ExecutionPlan
7  (
8      PlanID int Identity(1,1),
9      PlanName VarChar(25),
10     PlanDescription VarChar(50)
11 )
12 GO
13
```

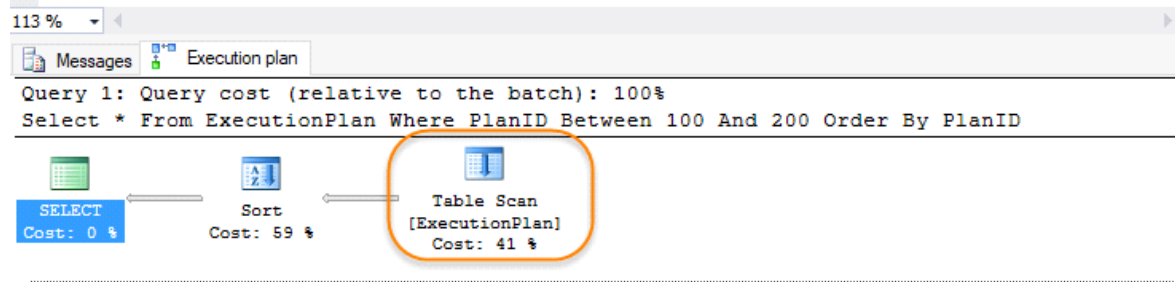
| | | |
|--------------------------------|---|----------------|
| | Cut | Ctrl+X |
| | Copy | Ctrl+C |
| | Paste | Ctrl+V |
| | Insert Snippet... | Ctrl+K, Ctrl+X |
| | Surround With... | Ctrl+K, Ctrl+S |
| Connection | | |
| Open Server in Object Explorer | | |
| | Execute | F5 |
| | Display Estimated Execution Plan | Ctrl+L |
| | IntelliSense Enabled | Ctrl+Q, Ctrl+I |
| | Trace Query in SQL Server Profiler | Ctrl+Alt+P |
| | Analyze Query in Database Engine Tuning Advisor | |
| | Design Query in Editor... | Ctrl+Shift+Q |
| | Include Actual Execution Plan | Ctrl+M |
| | Include Live Query Statistics | |
| | Include Client Statistics | Shift+Alt+S |
| Results To | | |
| | Properties Window | F4 |
| | Query Options... | |

☐ Insert 1000 rows

```
14 --Step Two
15 -----
16 Insert Into ExecutionPlan
17 (PlanName,PlanDescription)
18 Values ('Plan....','This is a Description')
19 Go 1000
20
```

☐ Create Select Statement and Show Estimated Execution Plan

```
21 --Step Three
22 -----
23 Select * From ExecutionPlan
24 Where PlanID Between 100 And 200
25 Order By PlanID;
26
27
```



☐ Create Primary Key

```

28 --Step Four
29 -----
30 ALTER TABLE ExecutionPlan
31 ADD PRIMARY KEY (PlanID);
32 Go

```

☐ Select the Select Statement and Show Estimated Execution Plan

```

35 --Step Five
36 -----
37 Select * From ExecutionPlan
38 Where PlanID Between 100 And 200
39 Order By PlanID;
40

```

113 %

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

----- Select * From ExecutionPlan Where PlanID Between 1...

SELECT
Cost: 0 %

Clustered Index Seek (Clustered)
[ExecutionPlan].[PK_Executio__755C...]
Cost: 100 %

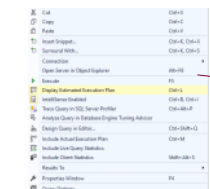
Estimated and Actual Plans

Wednesday, May 24, 2017 5:35 PM

☐ Step One - Run Query / 01a-ExecutionPlan.sql

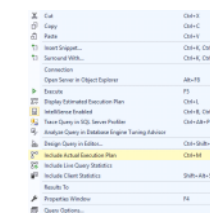
```
1 Use Northwind;
2 Go
3
4 SELECT Customers.CompanyName, Orders.OrderDate, Products.ProductName, [Order Details].UnitPrice,
5 [Order Details].Quantity, Categories.CategoryName, Suppliers.CompanyName AS Expr1
6 FROM
7 Orders INNER JOIN
8 Customers ON Orders.CustomerID = Customers.CustomerID INNER JOIN
9 [Order Details] ON Orders.OrderID = [Order Details].OrderID INNER JOIN
10 Products ON [Order Details].ProductID = Products.ProductID INNER JOIN
11 Categories ON Products.CategoryID = Categories.CategoryID INNER JOIN
12 Suppliers ON Products.SupplierID = Suppliers.SupplierID
13 Where Customers.Country = 'USA'
```

☐ Step Two - Show Estimated Execution Plan to show how SQL solved the Query



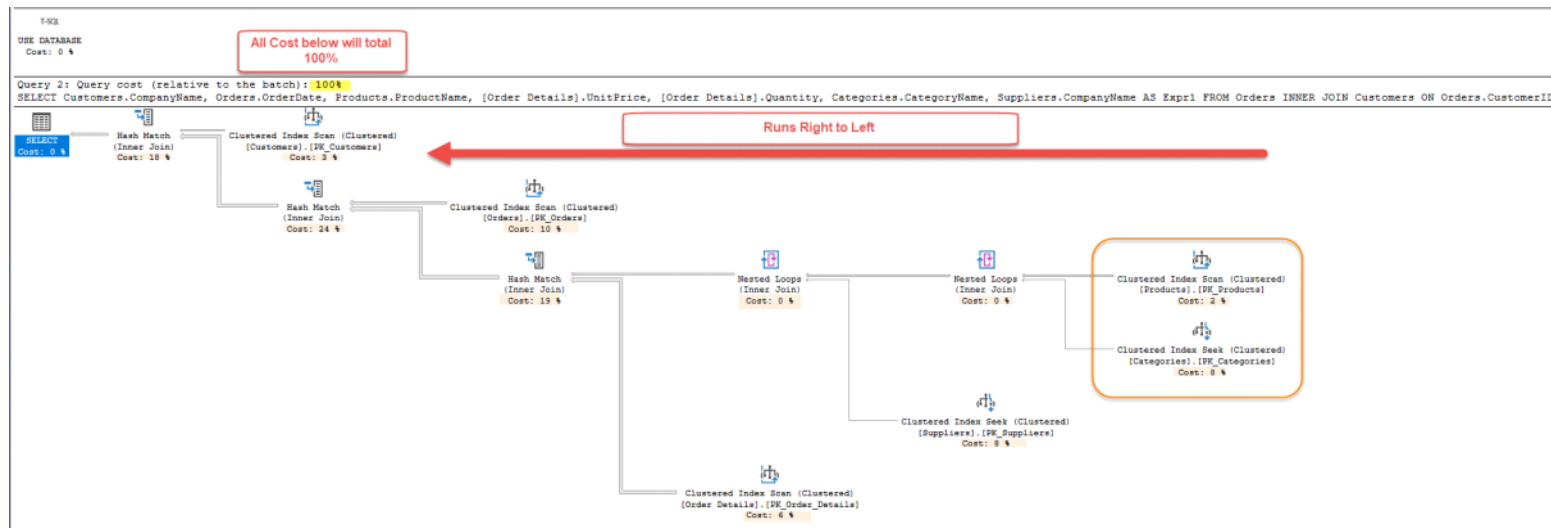
Does NOT run Query, If NO Plan then must create one. (Pay the Penalty)
NO RESULTS PANE

☐ Step Three- Show Actual Execution Plan to show how SQL solved the Query



Runs Query, Create Plan if NOT there. (Pay the Penalty)
Results & Execution Plan

☐ SQL Server Identifies the Plan with the Lowest cost.



☐ Multiple Queries

Query 2: Query cost (relative to the batch): 50%

```
SELECT Customers.CompanyName, Orders.OrderDate, Products.ProductName, [Order Details].UnitPrice, [Order Details].Quantity, Categories.CategoryName, Suppliers.CompanyName AS Expr1 FROM Orders INNER JOIN
```

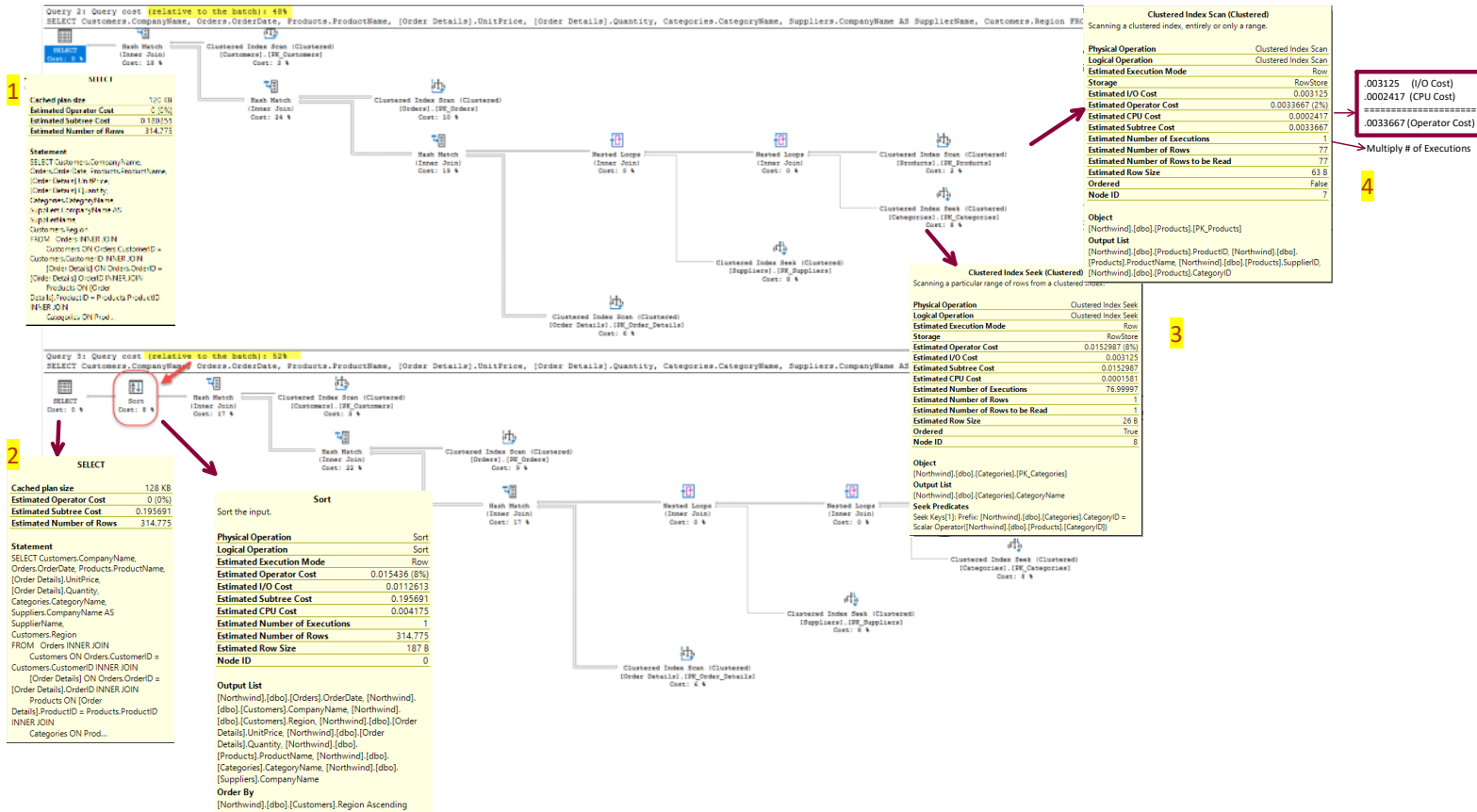


Saturday, February 9, 2019 5:12 PM

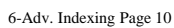
```

4  SELECT Customers.CompanyName, Orders.OrderDate, Products.ProductName, [Order Details] UnitPrice,
5  [Order Details] Quantity, Categories.CategoryName, Suppliers.CompanyName AS SupplierName,
6  Customers.Region
7  FROM Orders INNER JOIN
8  Customers ON Orders.CustomerID = Customers.CustomerID INNER JOIN
9  [Order Details] ON Orders.OrderID = [Order Details].OrderID INNER JOIN
10 Products ON [Order Details].ProductID = Products.ProductID INNER JOIN
11 Categories ON Products.CategoryID = Categories.CategoryID INNER JOIN
12 Suppliers ON Products.SupplierID = Suppliers.SupplierID
13 Where Customers.Country = 'USA'
14
15
16 SELECT Customers.CompanyName, Orders.OrderDate, Products.ProductName, [Order Details] UnitPrice,
17 [Order Details] Quantity, Categories.CategoryName, Suppliers.CompanyName AS SupplierName,
18 Customers.Region
19 FROM Orders INNER JOIN
20 Customers ON Orders.CustomerID = Customers.CustomerID INNER JOIN
21 [Order Details] ON Orders.OrderID = [Order Details].OrderID INNER JOIN
22 Products ON [Order Details].ProductID = Products.ProductID INNER JOIN
23 Categories ON Products.CategoryID = Categories.CategoryID INNER JOIN
24 Suppliers ON Products.SupplierID = Suppliers.SupplierID
25 Where Customers.Country = 'USA'
26 Order By Customers.Region

```



Saturday, February 9, 2019 5:49 PM



Save Execution Plan As...

Show Execution Plan XML...

Compare Showplan

Analyze Actual Exec

Find Node

Zoom In

Zoom Out

Custom Zoom...

Zoom to Fit

Properties

Clustered Index

[Suppliers]. [

Cost

```
1 <?xml version="1.0" encoding="utf-16"?>
2 <ShowPlanXML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" Version="1.5" Build="13.0.4001.0" xmlns="http://schemas.microsoft.com/sqlserver/reporting/execution/2005/"/>
3 <BatchSequence>
4 <Batch>
5 <Statements>
6 <Statement StatementCompId="1" StatementId="1" StatementText="Use Northwind; &#xD;&#xA;" StatementType="USE DATABASE" RetrievedFromCache="false" Database="Northwind" />
7 </Statement>
8 </Batch>
9 </BatchSequence>
10 <Statements>
11 <StmtSimple StatementCompId="1" StatementEstRows="314.775" StatementId="1" StatementOptmLevel="FULL" StatementOptmEarlyAbortReason="GoodEnoughPlanFound" Cardinality="314.775" StatementText="Use Northwind; &#xD;&#xA;" StatementType="USE DATABASE" RetrievedFromCache="false" Database="Northwind" />
12 <StatementSetOptions ANSI_NULLS="true" ANSI_PADDING="true" ANSI_WARNINGS="true" ARITHABORT="true" CONCAT_NULL_YIELDS_NULL="true" NUMERIC_ROUNDABORT="false" QUOTED_IDENTIFIER="true" />
13 <QueryPlan CachedPlanSize="120" CompileTime="9" CompileCPU="9" CompileMemory="1072">
14 <MemoryGrantInfo SerialRequiredMemory="3072" SerialDesiredMemory="3568" />
15 <OptimizerHardwareDependentProperties EstimatedAvailableMemoryGrant="209030" EstimatedPagesCached="209030" EstimatedAvailableDegreeOfParallelism="8" MaxCompileMemory="209030" />
16 <RelOp AvgRowSize="187" EstimateCPU="0.0329186" EstimateIO="0" EstimateRebinds="0" EstimateRebinds="0" EstimateRows="314.775" LogicalOpName="Hash" />
17 <OutputList>
18 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Orders]" Column="OrderDate" />
19 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Customers]" Column="CompanyName" />
20 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Customers]" Column="Region" />
21 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Order Details]" Column="UnitPrice" />
22 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Order Details]" Column="Quantity" />
23 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Products]" Column="ProductName" />
24 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Categories]" Column="CategoryName" />
25 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Suppliers]" Column="CompanyName" />
26 </OutputList>
27 <MemoryFractions Input="0.16129" Output="0.16129" />
28 <Hash>
29 <DefinedValues />
30 <HashKeysBuild>
31 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Customers]" Column="CustomerID" />
32 </HashKeysBuild>
33 <HashKeysProbe>
34 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Orders]" Column="CustomerID" />
35 </HashKeysProbe>
36 <ProbeResidual>
37 <ScalarOperator ScalarString="[Northwind].[dbo].[Customers].[CustomerID]=[Northwind].[dbo].[Orders].[CustomerID]" />
38 <Compare CompareOp="eq">
39 <ScalarOperator>
40 <Identifier>
41 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Customers]" Column="CustomerID" />
42 </Identifier>
43 </ScalarOperator>
44 <ScalarOperator>
45 <Identifier>
46 <ColumnReference Database="Northwind" Schema="[dbo]" Table="[Orders]" Column="CustomerID" />
47 </Identifier>
48 </ScalarOperator>
49 </Compare>
50 </ProbeResidual>
51 </Hash>
52 </QueryPlan>
53 </Statements>
54 </ShowPlanXML>
```

Show Plan - Statistics

Wednesday, May 24, 2017 7:13 PM

```
Execution plan.xml    Execution plan.xml    01b-ExecutionPlanS...OFFICE\delip (72)*    01a-ExecutionPlan.s...OFFICE\delip (69)*    CustomerOrders.sqlplan    DBCC_Statistics.sql...C

1  USE Northwind;
2  GO
3
4  /*
5   Updating statistics ensures that queries compile with up-to-date statistics,
6   providing a current plan; however, updating statistics causes queries to
7   recompile. We recommend not updating statistics too frequently.
8  */
9  --UPDATE STATISTICS Customers CompanyName WITH FULLSCAN;
10 GO
11
12 --SET SHOWPLAN_TEXT ON
13 --SET SHOWPLAN_TEXT OFF
14
15 --SET SHOWPLAN_ALL ON
16 --SET SHOWPLAN_ALL OFF
17
18 --SET SHOWPLAN_XML ON
19 --SET SHOWPLAN_XML OFF
20
21 --SET STATISTICS PROFILE ON
22 --SET STATISTICS PROFILE OFF
23
24 --SET STATISTICS XML ON -- The is the recommended option to use
25 --SET STATISTICS XML OFF
26 GO
27
28 SELECT Customers.CompanyName, Customers.ContactName, Orders.OrderDate,
29        Orders.RequiredDate, Products.ProductName, [Order Details].UnitPrice,
30        [Order Details].Quantity
31 FROM    Customers INNER JOIN
32        Orders ON Customers.CustomerID = Orders.CustomerID INNER JOIN
33        [Order Details] ON Orders.OrderID = [Order Details].OrderID INNER JOIN
34        Products ON [Order Details].ProductID = Products.ProductID
35 Go
36
```

SET SHOWPLAN_TEXT ON

| StmntText |
|--|
| 1 SELECT Customers.CompanyName, Customers Contact... |
| StmntText |
| 1 I-Hash Match (Inner Join, HASH ([Northwind].[d... |
| 2 I-Index Scan (OBJECT ([Northwind].[dbo].[Pr... |
| 3 I-Hash Match (Inner Join, HASH ([Northwind]... |
| 4 I-Clustered Index Scan (OBJECT ([Northwi... |
| 5 I-Hash Match (Inner Join, HASH ([Northwi... |
| 6 I-Clustered Index Scan (OBJECT ([Nort... |
| 7 I-Clustered Index Scan (OBJECT ([Nort... |

SET SHOWPLAN_ALL ON

| StmntText | StmtId | NodeId | Parent | PhysicalOp | LogicalOp | Argument | DefinedValues | EstimateRows | EstimateIO | EstimateCPU | AvgRowSize | TotalSize |
|--|--------|--------|--------|----------------------|----------------------|---|--|--------------|-------------|-------------|------------|-----------|
| 1 SELECT Customers.CompanyName, Customers Contact... | 1 | 1 | 0 | NULL | NULL | 1 | NULL | 2155 | NULL | NULL | NULL | 0.15668 |
| 2 I-Hash Match (Inner Join, HASH ([Northwind].[dbo].[Pr... | 1 | 2 | 1 | Hash Match | Inner Join | HASH ([Northwind].[dbo].[Products].[ProductID])=(N... | NULL | 2155 | 0 | 0.03145821 | 151 | 0.15668 |
| 3 I-Index Scan (OBJECT ([Northwind].[dbo].[Products]... | 1 | 3 | 2 | Index Scan | Index Scan | OBJECT ([Northwind].[dbo].[Products].[ProductID])=[North... | [Northwind].[dbo].[Products].[ProductID].[North... | 77 | 0.003125 | 0.0002417 | 55 | 0.00336 |
| 4 I-Hash Match (Inner Join, HASH ([Northwind].[dbo].[... | 1 | 4 | 2 | Hash Match | Inner Join | HASH ([Northwind].[dbo].[Customers].[CustomerID])=... | NULL | 2155 | 0 | 0.04034815 | 113 | 0.12176 |
| 5 I-Clustered Index Scan (OBJECT ([Northwind].[dbo]... | 1 | 5 | 4 | Clustered Index Scan | Clustered Index Scan | OBJECT ([Northwind].[dbo].[Customers].[PK_Custom... | [Northwind].[dbo].[Customers].[CustomerID].[Nor... | 91 | 0.004606482 | 0.0002571 | 93 | 0.00486 |
| 6 I-Hash Match (Inner Join, HASH ([Northwind].[dbo]... | 1 | 6 | 4 | Hash Match | Inner Join | HASH ([Northwind].[dbo].[Orders].[OrderID])=(North... | NULL | 2155 | 0 | 0.04671476 | 47 | 0.07656 |
| 7 I-Clustered Index Scan (OBJECT ([Northwind].[dbo]... | 1 | 7 | 6 | Clustered Index Scan | Clustered Index Scan | OBJECT ([Northwind].[dbo].[Orders].[PK_Orders])... | [Northwind].[dbo].[Orders].[OrderID].[Northwind]... | 830 | 0.01719907 | 0.00107 | 37 | 0.01826 |
| 8 I-Clustered Index Scan (OBJECT ([Northwind].[dbo]... | 1 | 8 | 6 | Clustered Index Scan | Clustered Index Scan | OBJECT ([Northwind].[dbo].[Order Details].[PK_Orde... | [Northwind].[dbo].[Order Details].[OrderID].[Nort... | 2155 | 0.009050926 | 0.0025275 | 25 | 0.01157 |

SET SHOWPLAN_XML ON

| StmntText |
|---|
| 1 Microsoft SQL Server 2005 XML Showplan |
| 2 <ShowPlanXML xmlns="http://schemas.microsoft.com... |

SET STATISTICS XML ON

| Results | | Messages | | | | | |
|--|--|---------------|-------------------------|-------------------------|----------------------------------|-----------|----------|
| | CompanyName | ContactName | OrderDate | RequiredDate | ProductName | UnitPrice | Quantity |
| 1 | Vins et alcools Chevalier | Paul Henriot | 1996-07-04 00:00:00.000 | 1996-08-01 00:00:00.000 | Queso Cabrales | 14.00 | 12 |
| 2 | Vins et alcools Chevalier | Paul Henriot | 1996-07-04 00:00:00.000 | 1996-08-01 00:00:00.000 | Singaporean Hokkien Fried Mee | 9.80 | 10 |
| 3 | Vins et alcools Chevalier | Paul Henriot | 1996-07-04 00:00:00.000 | 1996-08-01 00:00:00.000 | Mozzarella di Giovanni | 34.80 | 5 |
| 4 | Toms Spezialitäten | Karin Josephs | 1996-07-05 00:00:00.000 | 1996-08-16 00:00:00.000 | Tofu | 18.60 | 9 |
| 5 | Toms Spezialitäten | Karin Josephs | 1996-07-05 00:00:00.000 | 1996-08-16 00:00:00.000 | Manjimp Dried Apples | 42.40 | 40 |
| 6 | Hanari Cames | Mario Pontes | 1996-07-08 00:00:00.000 | 1996-08-05 00:00:00.000 | Jack's New England Clam Chowder | 7.70 | 10 |
| 7 | Hanari Cames | Mario Pontes | 1996-07-08 00:00:00.000 | 1996-08-05 00:00:00.000 | Manjimp Dried Apples | 42.40 | 35 |
| 8 | Hanari Cames | Mario Pontes | 1996-07-08 00:00:00.000 | 1996-08-05 00:00:00.000 | Louisiana Fiery Hot Pepper Sauce | 16.80 | 15 |
| 9 | Victualies en stock | Mary Saveley | 1996-07-08 00:00:00.000 | 1996-08-05 00:00:00.000 | Gustaf's Knäckebröd | 16.80 | 6 |
| 10 | Victualies en stock | Mary Saveley | 1996-07-08 00:00:00.000 | 1996-08-05 00:00:00.000 | Ravioli Angelo | 15.60 | 15 |
| 11 | Victualies en stock | Mary Saveley | 1996-07-08 00:00:00.000 | 1996-08-05 00:00:00.000 | Louisiana Fiery Hot Pepper Sauce | 16.80 | 20 |
| Microsoft SQL Server 2005 XML Showplan | | | | | | | |
| 1 | ShowPlanXML xmlns="http://schemas.microsoft.com/ | | | | | | |

Create Query

```
SQLQuery9.sql - (local)\\sqlserver71\\Northwind (LAPS\\delp (S2)) - Microsoft SQL Server 2017
--SELECT Customers.CompanyName, Customers.ContactName, Orders.OrderDate, Orders.RequiredDate,
[Order Details].UnitPrice, [Order Details].Quantity, Products.ProductName
FROM Customers INNER JOIN
Orders ON Customers.CustomerID = Orders.CustomerID INNER JOIN
[Order Details] ON Orders.OrderID = [Order Details].OrderID INNER JOIN
Products ON [Order Details].ProductID = Products.ProductID
```

| Company Name | Contact Name | Order Date | Required Date | Unit Price | Quantity | Product Name |
|---------------------------|--------------|-------------------------|-------------------------|------------|----------|-------------------------------|
| Vins et alcools Chevalier | Paul Henriot | 1996-07-04 00:00:00.000 | 1996-08-01 00:00:00.000 | 14.00 | 12 | Queijo Cabrales |
| Vins et alcools Chevalier | Paul Henriot | 1996-07-04 00:00:00.000 | 1996-08-01 00:00:00.000 | 9.80 | 10 | Singaporean Hokkien Fried Mee |
| Vins et alcools Chevalier | Paul Henriot | 1996-07-04 00:00:00.000 | 1996-08-01 00:00:00.000 | 34.80 | 5 | Mozzarella di Giovanni |
| Toms Spezialitäten | Karin Joseph | 1996-07-05 00:00:00.000 | 1996-08-16 00:00:00.000 | 18.60 | 9 | Tofu |

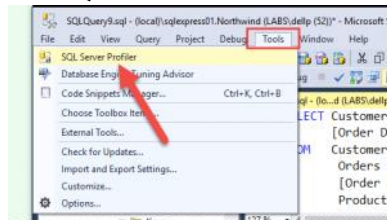
USE AdoDotNet C# Windows Forms

Form1

SP

| CustomerID | CompanyName | ContactName | ContactTitle | Address |
|------------|------------------------|-------------------|----------------------|---------------|
| COMMI | Comércio Mineiro | Pedro Afonso | Sales Associate | Av. dos Lus |
| FAMIA | Familia Arquibaldo | Aria Cruz | Marketing Assistant | Rua Orós, 9 |
| GOURL | Gourmet Lanchonetes | André Fonseca | Sales Associate | Av. Brasil, 4 |
| QUEEN | Queen Cozinha | Lúcia Carvalho | Marketing Assistant | Alameda d |
| TRADH | Tradição Hipermercados | Anabela Domingues | Sales Representative | Av. Inês de |
| WELLI | Wellington Importadora | Paula Parente | Sales Manager | Rua do Me |

Start SQL Profiler



Trace Properties

General | Events Selection

Trace name: Testing CustomerOrders Query

Trace provider name: sqlserver

Trace provider type: Microsoft SQL Server 2017 version: 13.0.4022

Use the template: TSQL

☒ Save to file ☒ Save to table

☒ Save to file ☒ Save to table

☐ Enable trace stop time

Run Cancel Help

Trace Properties

General | Events Selection

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

| Events | TextData | SPID | StartTime | BinaryData |
|--|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
| <input checked="" type="checkbox"/> Security Audit | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Audit Login | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Audit Logout | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Sessions | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> ExistingConnection | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Stored Procedures | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> RPC Starting | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> TSQL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> SQL Batch Starting | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

Security Audit: Includes event classes that are used to audit server activity.

No data column selected.

Run Cancel Help

Run Query

Testing CustomerOrders Query (sqlserver01)

| EventClass | TextData | SPID | StartTime | BinaryData |
|--------------------|--|------|------------------------|-------------|
| Trace Start | | | 2017-05-24 23:43:24... | |
| ExistingConnection | -- network protocol: LPC set quote... | 51 | 2017-05-24 23:43:04... | |
| ExistingConnection | -- network protocol: LPC set quote... | 52 | 2017-05-24 23:43:04... | |
| ExistingConnection | -- network protocol: LPC set quote... | 53 | 2017-05-24 23:43:09... | |
| ExistingConnection | -- network protocol: LPC set quote... | 54 | 2017-05-24 23:43:09... | |
| ExistingConnection | -- network protocol: LPC set quote... | 56 | 2017-05-24 23:43:16... | |
| ExistingConnection | -- network protocol: LPC set quote... | 57 | 2017-05-24 23:43:09... | |
| RPC:Starting | exec sp_executesql N'exec sp_trace... | 56 | 2017-05-24 23:43:34... | 0X000000... |
| SQL:BatchStarting | SELECT SERVERPROPERTY('EngineEditio... | 52 | 2017-05-24 23:43:36... | |
| SQL:BatchStarting | SELECT Customers.CompanyName, Custo... | 52 | 2017-05-24 23:43:36... | |

```
SELECT Customers.CompanyName, Customers.ContactName, Orders.OrderDate, Orders.RequiredDate,
[Order Details].UnitPrice, [Order Details].Quantity, Products.ProductName
FROM Customers INNER JOIN
Orders ON Customers.CustomerID = Orders.CustomerID INNER JOIN
[Order Details] ON Orders.OrderID = [Order Details].OrderID INNER JOIN
Products ON [Order Details].ProductID = Products.ProductID
```

Trace is running.

Ln 10, Col 1 Rows: 10

Other Statistics

Sunday, May 28, 2017 5:38 PM

Object Explorer

- Connect
- SQL Server 13.0.4202 - LABS\
- Databases
 - System Databases
 - Database Snapshots
 - AdventureWorks2012
 - LabData
 - NormalizedForms
 - Northwind
 - ProcedureLabDB
 - ReportServer
 - ReportServerTempDB
 - Schools

Index Physical Statistics LABS\SQLEXPRESS01

Index Usage Statistics LABS\SQLEXPRESS01

Index Usage Statistics [Schools]

on LABS\SQLEXPRESS01 at 5/28/2017 5:37:45 PM

This report provides details on usage of individual Indexes within the Database as well as data on the cost of maintaining them. It provides data for indexes on memory optimized tables.

Index Usage Statistics
Shows how the users and system use the indexes.

Index Operational Statistics
Shows details of the no. of operations performed on the indexes.

| Table Name | Index Name | Partition Number | # Range Scans | # Singleton Lookups | # Page Allocations Leaf | # Page Allocations Non-leaf | # Insert Leaf |
|--------------------------|--------------------|------------------|---------------|---------------------|-------------------------|-----------------------------|---------------|
| dbo.Course | CourseColumn_Index | 1 | 2 | 0 | 0 | 0 | 0 |
| dbo.Course | CourseColumn_Index | 1 | 0 | 0 | 0 | 0 | 0 |
| dbo.Course | CourseColumn_Index | 1 | 0 | 0 | 0 | 0 | 0 |
| dbo.CourseColumnStore | CourseColumn_Index | | | | | | 0 |
| dbo.EmployeeClustered | EmployeeID_Index | | | | | | 0 |
| dbo.EmployeeNonClustered | EmployeeID_Index | | | | | | 0 |
| dbo.Person | FullName_Index | | | | | | 0 |

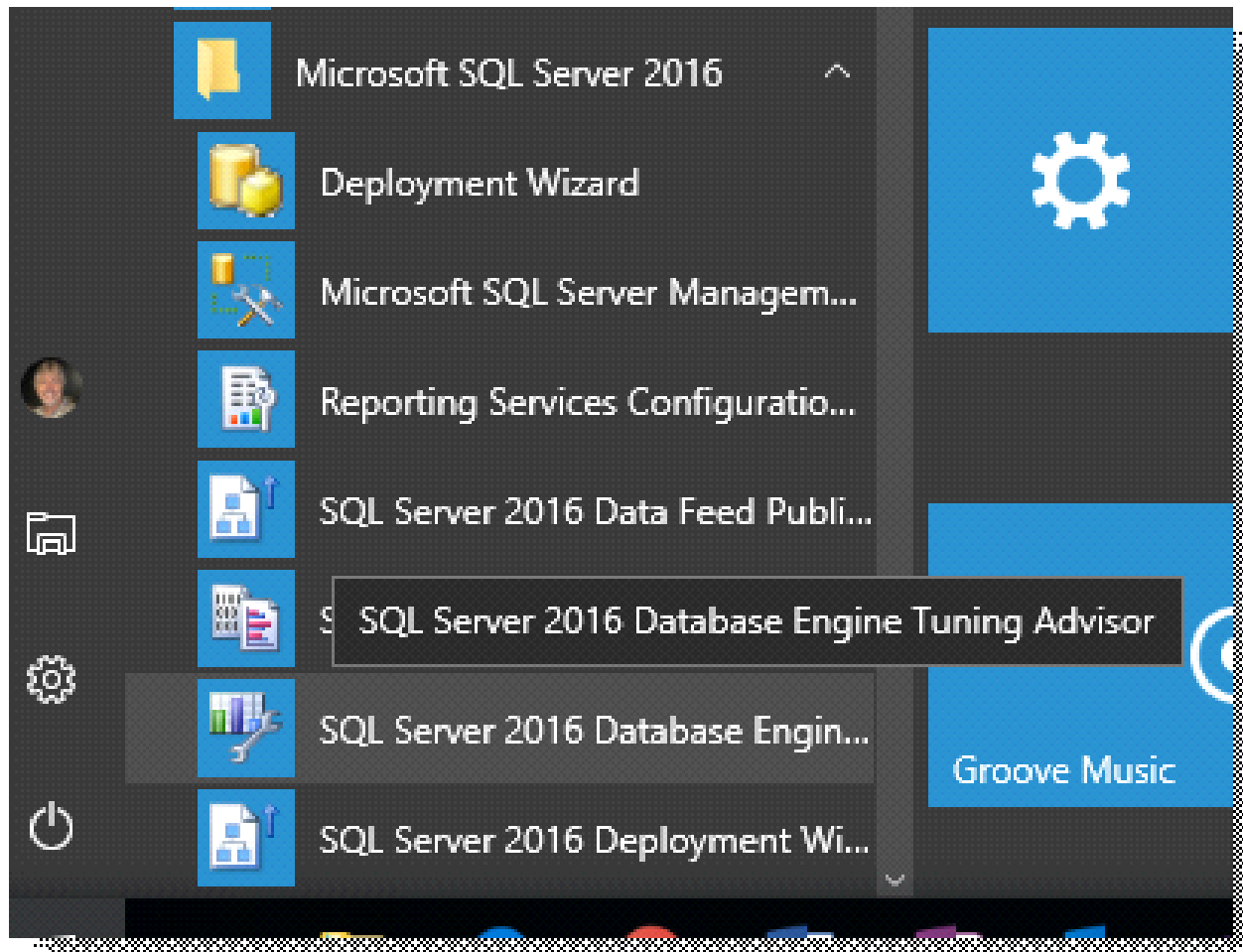
Reports

- New Database...
- New Query
- Script Database as
- Tasks
- Policies
- Facets
- Start PowerShell
- Reports
 - Standard Reports
 - Disk Usage
 - Disk Usage by Top Tables
 - Disk Usage by Table
 - Disk Usage by Partition
 - Backup and Restore Events
 - All Transactions
 - All Blocking Transactions
 - Top Transactions by Age
 - Top Transactions by Blocked Transactions Count
 - Top Transactions by Locks Count
 - Resource Locking Statistics by Objects
 - Object Execution Statistics
 - Database Consistency History
 - Memory Usage By Memory Optimized Objects
 - Transaction Performance Analysis Overview
 - Index Usage Statistics**
 - Index Physical Statistics
 - Schema Changes History
 - User Statistics
 - Custom Reports...
 - Index Physical Statistics
 - Index Usage Statistics
- Rename
- Delete
- Refresh
- Properties

Database Tuning Advisor

Thursday, June 8, 2017

8:37 PM

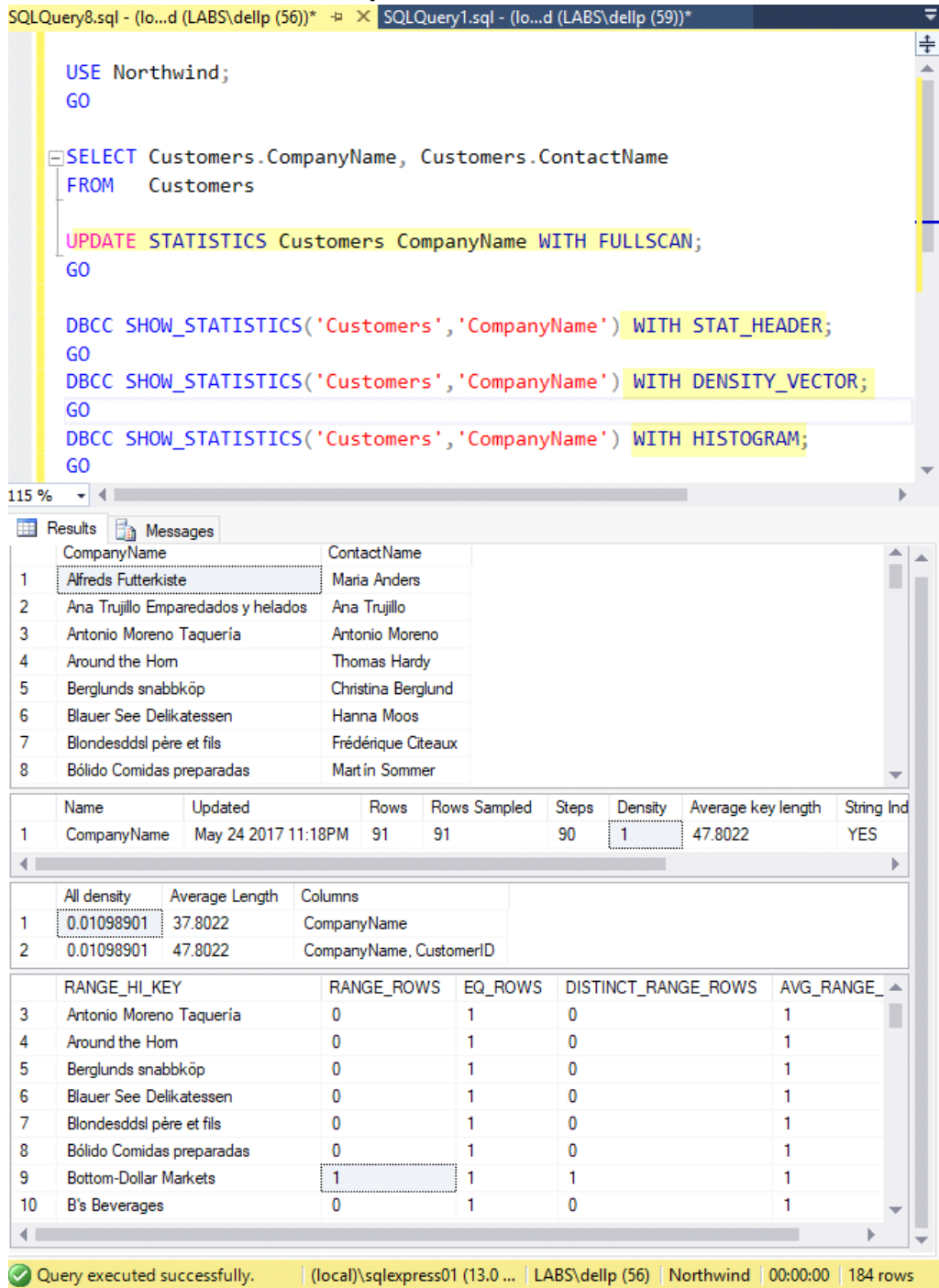


Tuning Session usually takes one hour by default

DBCC Statistics

Wednesday, May 24, 2017 11:13 PM

Database Constancy Checker



```
USE Northwind;
GO

SELECT Customers.CompanyName, Customers.ContactName
FROM Customers

UPDATE STATISTICS Customers CompanyName WITH FULLSCAN;
GO

DBCC SHOW_STATISTICS('Customers', 'CompanyName') WITH STAT_HEADER;
GO
DBCC SHOW_STATISTICS('Customers', 'CompanyName') WITH DENSITY_VECTOR;
GO
DBCC SHOW_STATISTICS('Customers', 'CompanyName') WITH HISTOGRAM;
GO
```

| CompanyName | ContactName |
|--------------------------------------|--------------------|
| 1 Alfreds Futterkiste | Maria Anders |
| 2 Ana Trujillo Emparedados y helados | Ana Trujillo |
| 3 Antonio Moreno Taquería | Antonio Moreno |
| 4 Around the Horn | Thomas Hardy |
| 5 Berglunds snabbköp | Christina Berglund |
| 6 Blauer See Delikatessen | Hanna Moos |
| 7 Blondesddsl père et fils | Frédérique Citeaux |
| 8 Bólido Comidas preparadas | Martín Sommer |

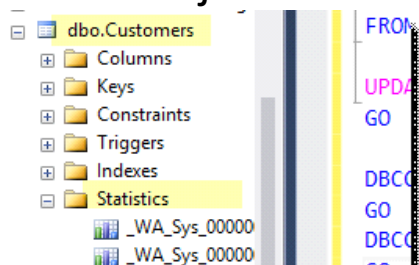
| Name | Updated | Rows | Rows Sampled | Steps | Density | Average key length | String Ind |
|---------------|---------------------|------|--------------|-------|---------|--------------------|------------|
| 1 CompanyName | May 24 2017 11:18PM | 91 | 91 | 90 | 1 | 47.8022 | YES |

| All density | Average Length | Columns |
|--------------|----------------|-------------------------|
| 1 0.01098901 | 37.8022 | CompanyName |
| 2 0.01098901 | 47.8022 | CompanyName, CustomerID |

| RANGE_HI_KEY | RANGE_ROWS | EQ_ROWS | DISTINCT_RANGE_ROWS | AVG_RANGE_ |
|-----------------------------|------------|---------|---------------------|------------|
| 3 Antonio Moreno Taquería | 0 | 1 | 0 | 1 |
| 4 Around the Horn | 0 | 1 | 0 | 1 |
| 5 Berglunds snabbköp | 0 | 1 | 0 | 1 |
| 6 Blauer See Delikatessen | 0 | 1 | 0 | 1 |
| 7 Blondesddsl père et fils | 0 | 1 | 0 | 1 |
| 8 Bólido Comidas preparadas | 0 | 1 | 0 | 1 |
| 9 Bottom-Dollar Markets | 1 | 1 | 1 | 1 |
| 10 B's Beverages | 0 | 1 | 0 | 1 |

Query executed successfully. (local)\sqlexpress01 (13.0 ... LABS\dellp (56) | Northwind | 00:00:00 | 184 rows

Statistic Objects From Explorer



```
FROM
UPDA
GO
DBCC
GO
DBCC
GO
```

