

Übung Komprimierung

Ziel: Kennenlernen des Komprimierungs-Assistenten; Auswirkungen der Komprimierung

Aufgabenstellung:

1. Verwenden Sie die KU1 Tabelle, die im Kurs erstellt wurde. Wenn Sie diese Übung später ausführen und die KU1 Tabelle nicht mehr vorhanden ist, erstellen Sie eine Tabelle mit Testdaten, z.B.:

```
SELECT  c.CustomerID
        , c.CompanyName
        , c.ContactName
        , c.ContactTitle
        , c.City
        , c.Country
        , o.EmployeeID
        , o.OrderDate
        , o.freight
        , o.shipcity
        , o.shipcountry
        , od.OrderID
        , od.ProductID
        , od.UnitPrice
        , od.Quantity
        , p.ProductName
        , e.LastName
        , e.FirstName
        , e.birthdate
into Test.dbo.KundenUmsatz
FROM    Northwind.dbo.Customers c
        INNER JOIN Northwind.dbo.Orders o ON c.CustomerID =
o.CustomerID
        INNER JOIN Northwind.dbo.Employees e ON o.EmployeeID =
e.EmployeeID
        INNER JOIN Northwind.dbo.[Order Details] od ON o.orderid =
od.orderid
        INNER JOIN Northwind.dbo.Products p ON od.productid =
p.productid
GO
```

2. Wenn es die KU1 Tabelle nicht mehr gibt: Generieren Sie eine große Datenmenge:

```
INSERT INTO KundenUmsatz  
SELECT * FROM KundenUmsatz  
GO 9
```

3. Wenn es die KU1 Tabelle nicht mehr gibt: Legen Sie eine ID-Spalte an:

```
-- Kopie mit ID Spalte  
SELECT * INTO KU1 FROM KundenUmsatz  
ALTER TABLE KU1 ADD ID INT IDENTITY
```

4. Wenn es die KU1 Tabelle noch gibt und Sie schon mit der KU1-Tabelle geübt haben und beispielsweise Indizes oder Partitionen dafür erstellt haben, erstellen Sie eine Kopie davon:

```
SELECT *  
INTO KUn  
FROM KU1
```

5. Schalten Sie die Statistik ein und aktivieren Sie den Actual Execution Plan.

```
SET STATISTICS IO, TIME ON
```



Sie können den Execution Plan über die Menüleiste aktivieren, oder indem Sie STRG+M drücken.

6. Leeren Sie den Pufferspeicher:

```
dbcc dropcleanbuffers
```

7. Finden Sie heraus, wie viel sich im Cache befindet:

```
SELECT count(*)*8/1024 AS 'Data Cache Size(MB)'
      , CASE database_id
          WHEN 32767 THEN 'RESOURCEDB'
          ELSE db_name(database_id)
          END AS 'DatabaseName'
FROM sys.dm_os_buffer_descriptors
GROUP BY db_name(database_id) , database_id
ORDER BY 'Data Cache Size(MB)' DESC
```

Das Ergebnis wird variieren!

Beispiel:

	Data Cache Size(MB)	DatabaseName
1	248	Test
2	0	RESOURCEDB
3	0	tempdb
4	0	master
5	0	Northwind

8. Führen Sie eine ressourcenintensive Abfrage durch, z.B.:

```
SELECT AVG(Freight), CompanyName
FROM KU1
WHERE ID >= 1000000
GROUP BY CompanyName
```

Die Ausführungszeiten werden variieren! Logical Reads können abweichen, abhängig von der verwendeten Tabelle/Datenmenge! Zum Beispiel:

```
-- logical reads 56452
-- CPU time = 203 ms, elapsed time = 714 ms
```

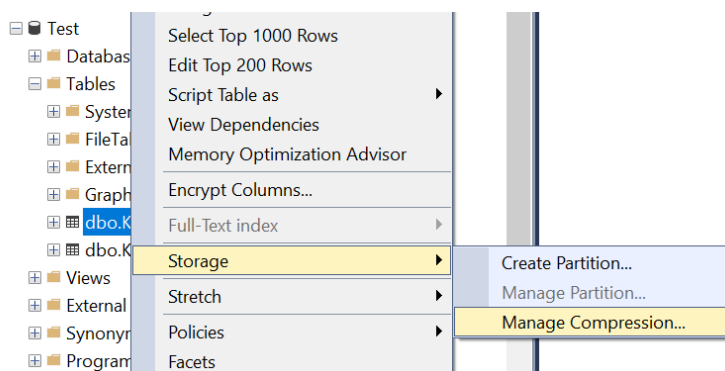
9. Checken Sie noch einmal den Cache:

```
SELECT count(*)*8/1024 AS 'Data Cache Size(MB)'
      , CASE database_id
          WHEN 32767 THEN 'RESOURCEDB'
          ELSE db_name(database_id)
          END AS 'DatabaseName'
FROM sys.dm_os_buffer_descriptors
GROUP BY db_name(database_id) , database_id
ORDER BY 'Data Cache Size(MB)' DESC
```

Das Ergebnis kann wieder variieren! Z.B.:

	Data Cache Size(MB)	DatabaseName
1	328	Test
2	3	RESOURCEDB
3	1	master
4	0	Northwind
5	0	msdb
6	0	AdventureWorks2019
7	0	tempdb
8	0	Demo
9	0	model

10. Stellen Sie fest, ob sich eine Komprimierung für die Tabelle KU1 lohnt: Gehen Sie im *Object Explorer* mit Rechtsklick auf die Tabelle KU1, wählen Sie *Storage* und *Manage Compression...*

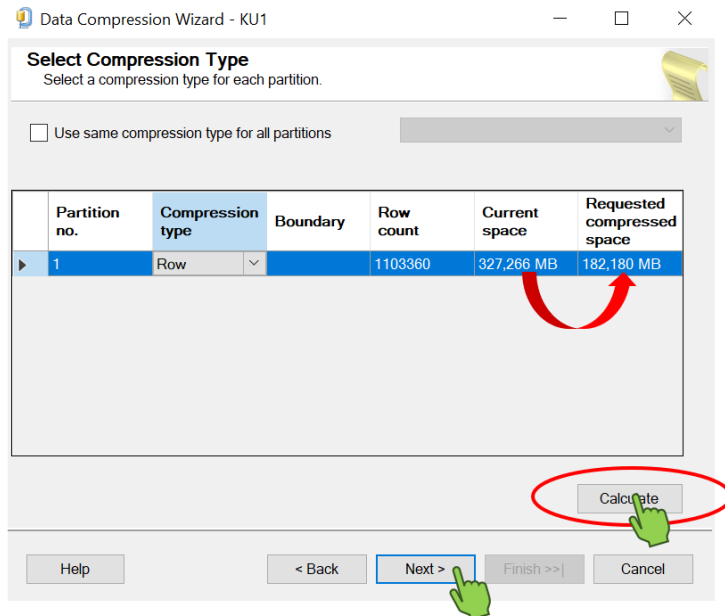


11. Es öffnet sich der *Data Compression Wizard*. Klicken Sie auf *Next >*.

12. Wählen Sie zunächst *Row Compression* im Dropdown-Menü aus.

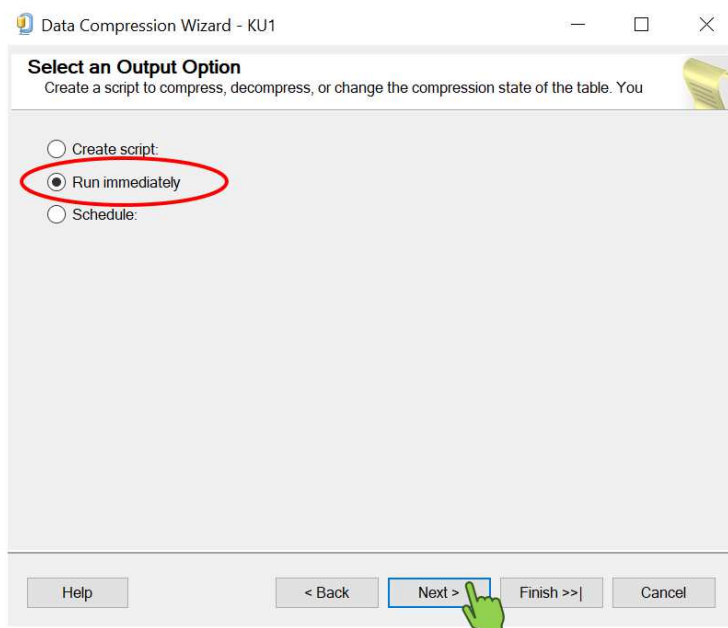
	Partition no.	Compression type	Boundary	Row count	Current space	Requested compressed space
	1	Row		1103360		

13. Der Wizard hilft Ihnen, zu entscheiden, ob sich eine Komprimierung lohnt. Klicken Sie auf *Calculate*.

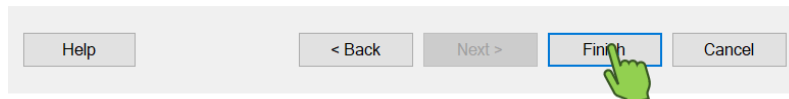


14. Derzeit werden > 327MB verbraucht; nach einer Komprimierung wären es nur noch 182MB. Die Komprimierung lohnt sich für uns, also klicken wir auf *Next >*.

Wir haben nun die Möglichkeit, uns ein *Create Script* erstellen zu lassen (die Anweisung zum Komprimieren als Skript ausgeben zu lassen), oder die Komprimierung sofort auszuführen. Wir wollen sie sofort ausführen; wählen Sie *Run immediately* und klicken Sie auf *Next >*.



15. Im nächsten Schritt können Sie noch einmal überprüfen, ob Sie mit Ihren Einstellungen zufrieden sind. Richtige Datenbank/Tabelle/... verwendet? Wenn alles stimmt, klicken Sie auf *Finish*. Diesen Schritt können Sie auch überspringen, indem Sie gleich auf *Finish >>|* klicken.



16. Überprüfen Sie wieder, wie viel Data sich im Cache befindet (das Ergebnis kann wieder variieren), z.B.:

	Data Cache Size(MB)	DatabaseName
1	511	Test
2	63	tempdb
3	6	RESOURCEDB
4	2	master
5	1	AdventureWorks2019
6	1	msdb
7	0	model
8	0	Northwind
9	0	Demo

17. Führen Sie die Abfrage noch einmal aus:

```
SELECT AVG(Freight), CompanyName
FROM KU1
WHERE ID >= 1000000
GROUP BY CompanyName
```

Das Ergebnis wird wieder variieren, z.B.:

```
-- logical reads 23273
-- CPU time = 375 ms, elapsed time = 168 ms
```

Die Lesevorgänge konnten im Vergleich zu vorhin drastisch verringert werden! Die Abfrage wird tendenziell schneller gelaufen sein; zugleich wird aber vermutlich die CPU time höher sein als vorhin.

18. Wenn Sie noch weiter üben möchten, führen Sie die Schritte ab Punkt 6. noch einmal aus, allerdings mit *page compression*.

Hinweis:

Da wir auf unserer KU1 Tabelle keinen Index haben und somit ein *Table Scan* durchgeführt wurde, entsprechen die *logical reads* auch der Anzahl an *pages*, auf denen unsere Tabelle abgespeichert wurde. Eine Möglichkeit, die Anzahl an *pages* vor und nach der Komprimierung zu überprüfen, ist der *showcontig*-Befehl.

```
dbcc showcontig('KU1')
```