

SQL Performance Tuning

Performance

- Latenzzeiten
- Transactions /Sec
- IOPS
- Restore in min
- Perfekte Lösung sinnlos

Messungen & Tools

- CPU
- RAM
- IO
- Network
- Locks
- Tempdb
- Perfmon
- Profiler
- XEvents
- DMVs
- QueryStore
- SSMS (Reports, Activity Monitor)
- Scripts & Tools
 - Nexus, PAL, Perf Dashboard
 - SP_blitz, Glenn Berry

Analyse

- Was ist das aktuelle Problem?
- Muster?
- AdHoc?
- VM?
- Storage (BlackBox: SAN)

Haufigsten Probleme

- Indizes
 - Falsche, fehlende, zuviele
- HDDs
- Blocking
- Kompilierungen
 - Falsches Planverhalten
- TempDB
- CPU
- Code
 - Cursor, Trigger, F(), Row by Row

Baseline

- Perfmon (PAL)
 - Zeit, Peaks
- DMV
 - Erste Analyse
- Historisierung mit XEvents

Problemlösung

- Einzelfall
 - Abfrage direkt analysieren
 - Xevents
 - Regelmäßig od sporadisch
- Server
 - Perfmon
 - Trace

Grenzwerte

- Avg/disk Reads writes /sek
- Disk Reads /Writes /sek
- Available Free Memory
- Memory pressure hints

Agenda

- Server Settings
- MAXDOP
- DB Settings
- HDD
 - Dateigruppen
 - Part. Sicht
 - Partitionierung
- DB Design for Admins
 - Normalisierung vs Redundanz
 - Datentypen
 - <> Seiten und Blöcke
- Indizes und Statistiken
- DMV
- Perfmon und Profiler
- XEvents

Indizes und Statistiken

Perfmon

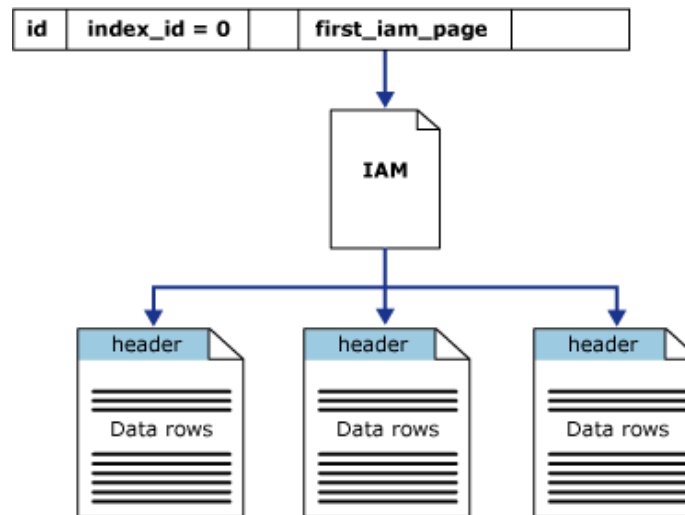
- Leistungsindikatoren
 - Messwerte zu OS
 - Arbeitsspeicher: Seiten/sek
 - Prozessor: Prozessorzeit in %
 - Physikalischer Datenträger: unter 2
 - Messwerte zu SQL
 - Puffercache: Puffercache Trefferquote < 90%
 - Puffercache: Lebenserwartung der Seiten > 300 .. Eigtl deutlich mehr
 - General Statistics: Benutzerverbindungen
 - SQL Locks: durchschn. Wartezeit
 - SQL Plancache: > 60%
 - SQL Statistics: Kompilierungen/sek; Batchanforderungen

Arbeitsweise der Indizes

- Indizes werden wie Datenbanken in Seiten verwaltet
- Seiten enthalten 8192 bytes
- Tabellen ohne Clustered Index = Heap
- B-Tree (balancierter Baum)
- Suche ab Wurzelknoten
 - Wie Telefonbuch

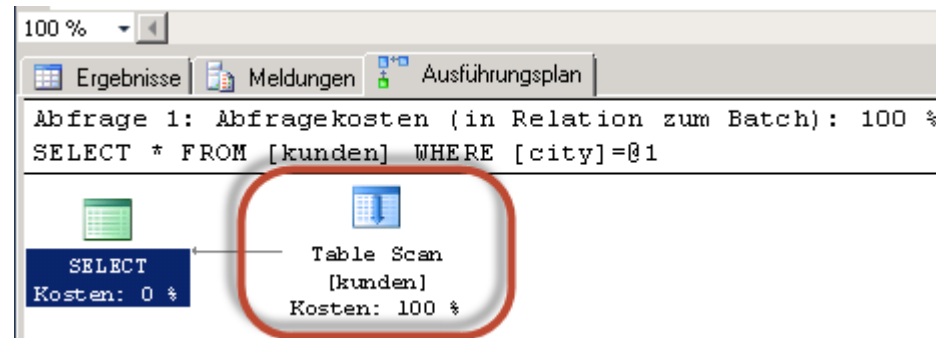
Heap

- Ein „Sau“-Haufen an Daten
- Eigtl keine Reihenfolge der Datensätze vorhersagbar
- Heap besteht aus vielen Seiten



Heap

- Suche nach bestimmten Datensätzen muss immer den kompletten Heap durchlaufen
- Suche = Durchsuchen aller Seiten
 - SET STATISTICS IO ON
- Suche = TABLE SCAN

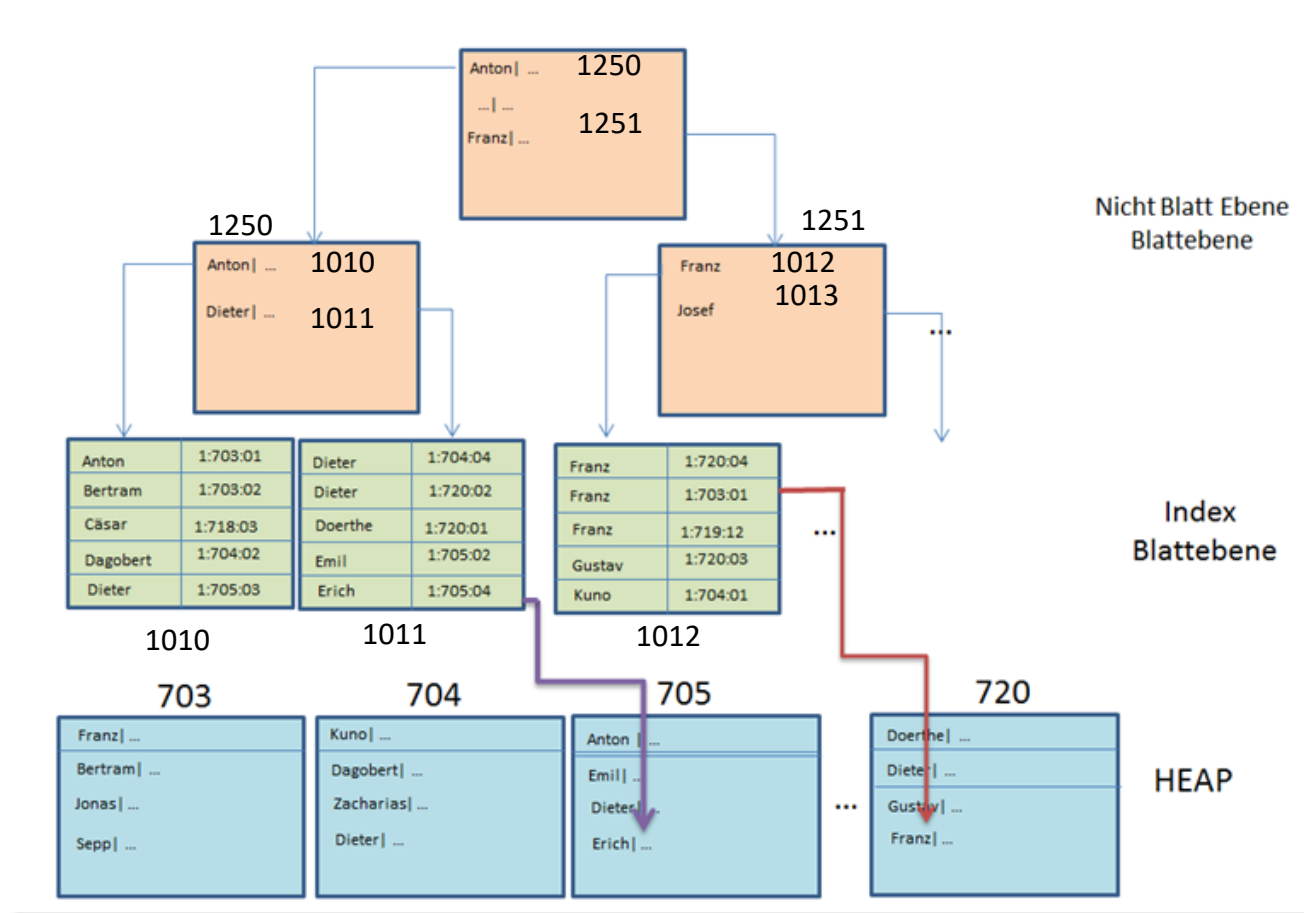


Wie funktioniert denn der Index?

- Wer das weiss, weiss auch welcher Index verwendet werden sollte
- Indizes werden ähnlich wie Telefonbücher verwaltet
 - Suche nach Tel von „Maier Hans“
Gezieltes Suchen im Telefonbuch..
...Treffer.. TelNr gefunden.
- Gezieltes Suchen im Index ist ein „Seek“



Wie funktioniert der Index?



Wie funktionieren Indizes

- Man kann auch nachschauen ;-)
 - sys.dm_db_index_physical_stats
 - DBCC IND (DB, Tabelle, 1)
 - DBCC PAGE (DB, Datei, Seite, [1,2,3])
 - DBCC TRACEON (3604)

Welche Indizes gibt es denn?

- Nicht gruppierter Index
- Gruppierter Index
- Zusammengesetzter Index
- Eindeutiger Index
- Index mit eingeschlossenen Spalten
- Gefilterter Index
- Partitionierter Index
- Columnstored Index
- Indizierte Sicht
- Abdeckender Index
- Realer hypothetischer Index

Welche Indizes gibt es denn?

- Spaß bei Seite!

- Nur 2!

- Bzw. 3

Nicht gruppierter Index

Gruppierter Index

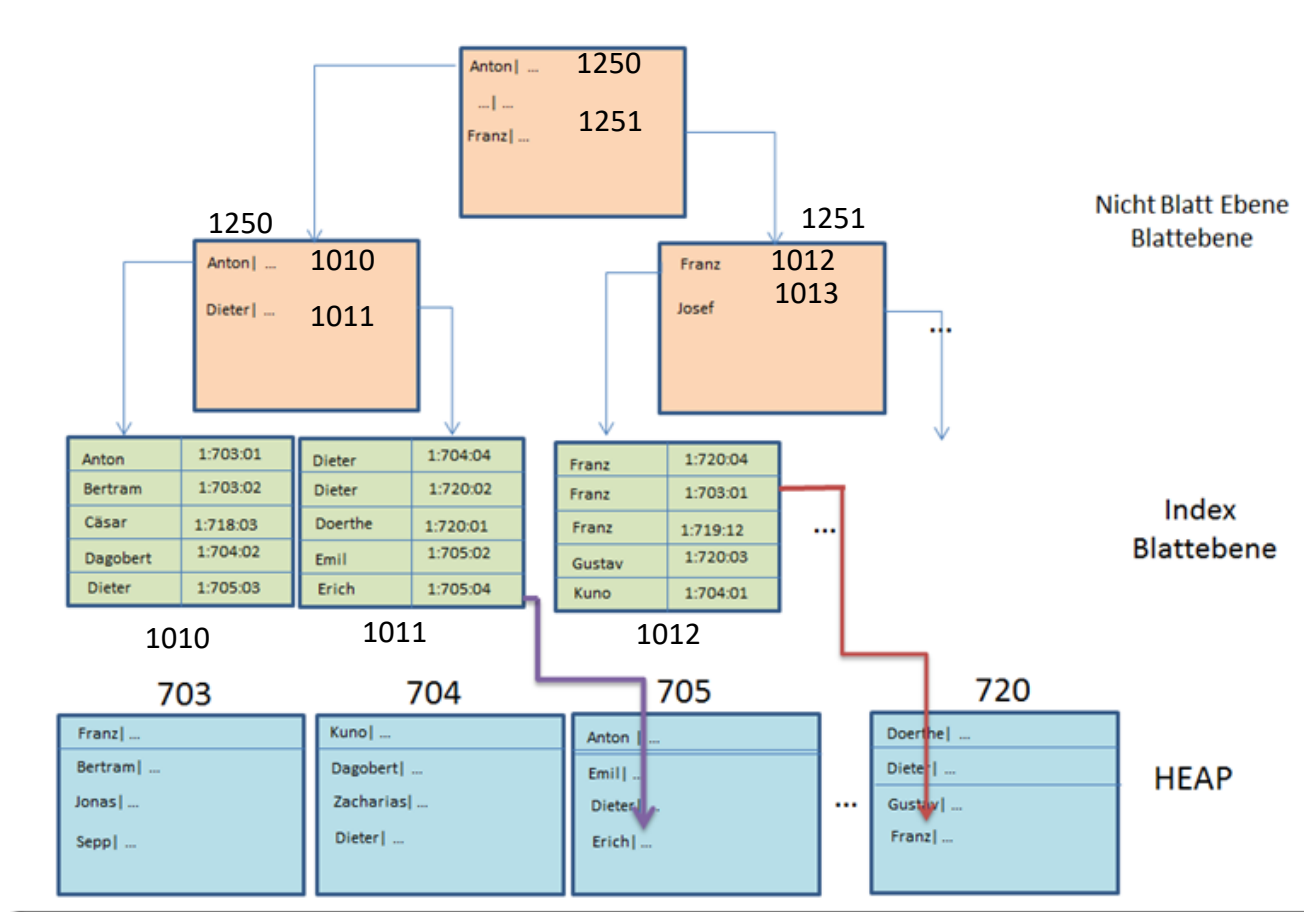
Columnstored Index

Spezialindizes: XML, Geo-Indizes

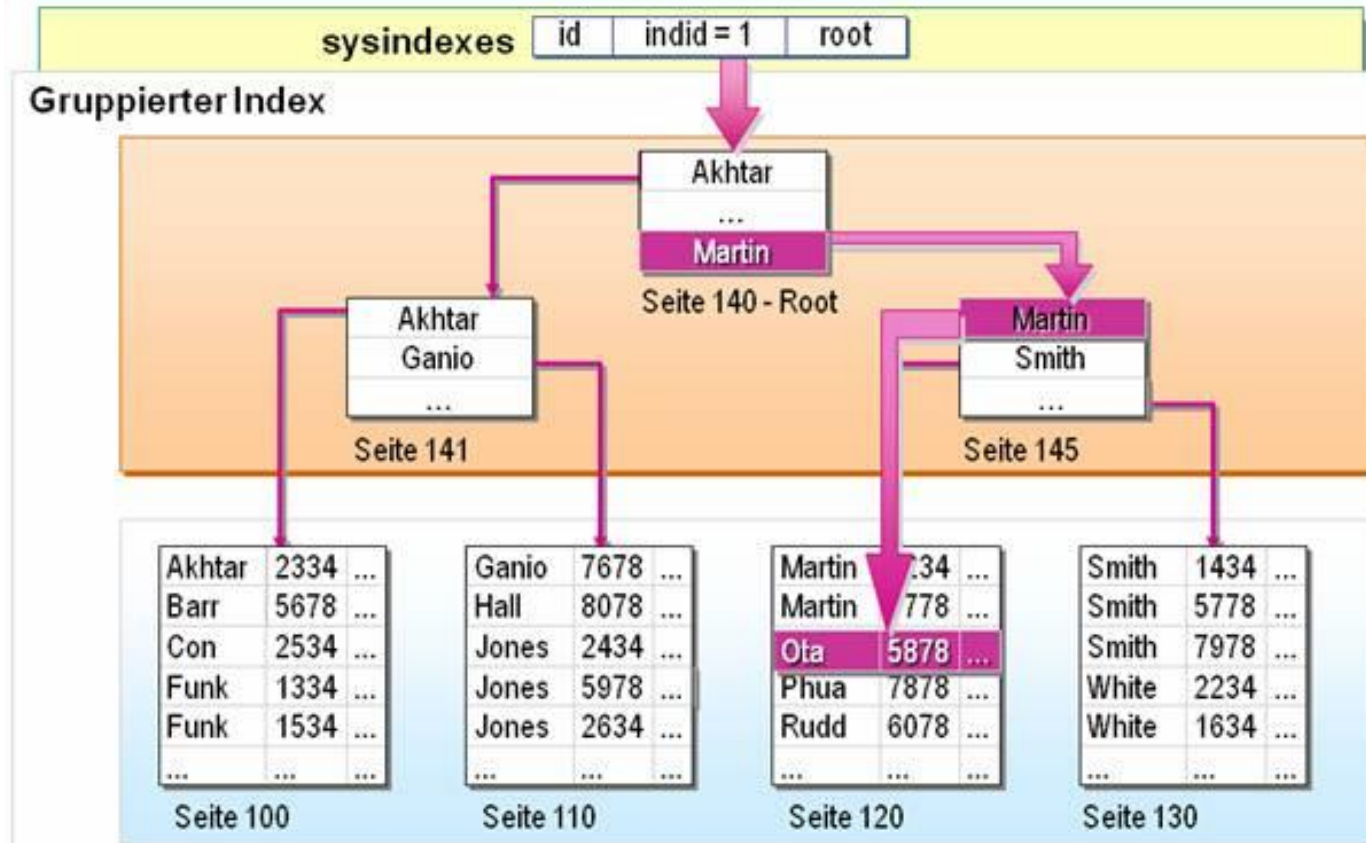
Wie funktioniert der Index?

- Nicht gruppierter Index lediglich sortierte Kopie der Indexspalten mit Zeiger auf den Originaldatensatz (1:204:02)
- Gruppierter Index ist Tabelle in physikalischer sortierter Form

Nicht gruppierter Index



Gruppiertes Index



Indizes

- Nicht gruppierte Indizes besitzen Kopien der Daten und verwenden Zeiger auf den Originaldatensatz
- Gruppierte Indizes sind die Tabellen!
...in physikalisch sortierter Form

Einsatzgebiete

- Gruppierter Index
 - Sehr gut bei Abfragen nach Bereichen und rel. Großen Ergebnismengen: < , > , between, like
Kandidaten: Bestelldatum, PLZ,..
Gibt's nur 1-mal, daher zuerst vergeben!
- Nicht gruppierter Index
 - Sehr gut bei Abfragen auf rel. eindeutige Werte bzw. geringen Ergebnismengen: =
Kandidaten: ID; Firmenname, ...
kann mehrfach verwendet werden (999-mal)
 - ➔ PK oft Gruppierter Index!! = Verschwendung

..und die anderen Indizes?

- Gefilterter Index:
 - Es müssen nicht mehr alle Datensätze in den Index mit aufgenommen werden.
- Mit Eingeschlossenen Spalten
 - Der Index kann zusätzliche Werte enthalten (→ SELECT), der Indexbaum wird dadurch nicht belastet.
- Partitionierter Index
 - Physikalische Verteilung der Indexdaten per Partitionierung

..und die anderen Indizes?

- Eindeutiger Index
 - Erzwingt eindeutige Werte.
Kandidat: Primary Key
- Zusammengesetzter Index
 - Index besteht aus mehreren Spalten. Auch im Indexbaum enthalten.
 - Kandidat: where umfaßt mehrere Spalten
 - Land , Stadt
- Abdeckender Index
 - ;-) leider nicht per „CREATE“, sondern ergibt aus der Abfrage. Bester Index! Alle Ergebnisse werden aus dem Index geliefert.
Keine Lookup Vorgänge!

..und die anderen Indizes?

- Indizierte Sicht
 - Perfekt für Aggregate!
 - = Clustered Index (materialized View)
 - Viele Bedingungen
 - Schemabinding, big_count()
 - In Enterprise Version können Statements „überschrieben“ werden
Statt Abfrage auf Tabelle, verwendet SQL Server die Sicht
 - Aber auch Probleme: Locks

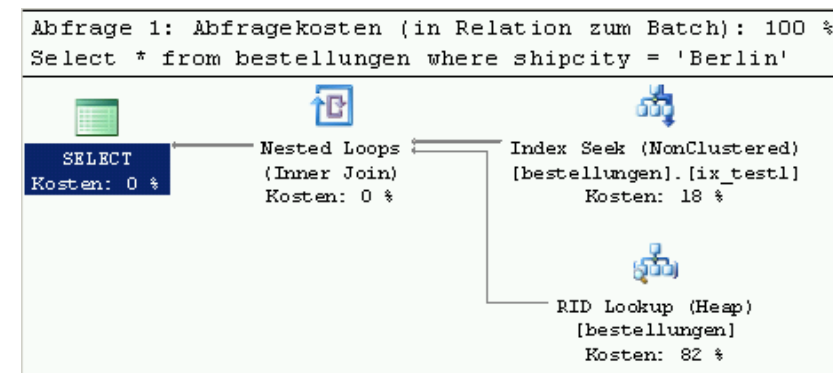
..und die anderen Indizes?

- Columnstored Index (ab SQL 2012)
 - Statt Datenätze werden Spalten in Seiten verwalten
 - Sehr gut bei Datawarehouse Szenarien
 - Mehrfach vorkommende Werte lassen sich gut komprimieren
 - Abfragen verwenden nur noch die Seiten, in denen die entsprechenden Daten vorhanden sind

DEMO

Welchen Indizes sollte man denn erstellen?

- Nur die, die man benötigt!
 - Jeder weitere Index stellt bei INS, UP ...eine Last dar
 - Keine überflüssigen Indizes (ABC, AB, A)
 - Wieviele Telefonbücher benötigen man pro Stadt?
- Die, die fehlen!
 - SQL Server merkt sich fehlende Indizes
- Nicht nur das WHERE ist entscheidend
 - Sondern auch der SELECT



Wie wirken sich Indizes auf die Leistung aus?

- Hervorragend,
 - Sofern keine Messdatenerfassung erfolgt
- Entscheidend ist die Anzahl der Indexebenen
 - Statt 100000 Seiten im Heap für 1 DS durchlaufen zu müssen, benötigt man über den Index so viele Seiten wie Ebenen vorhanden sind. (3 bis 4 Ebenen)
 - Ob 1 Mio oder 100 Mio DS, oft kaum mehr als 3 Ebenen

Worauf sollte man Indizes achten?

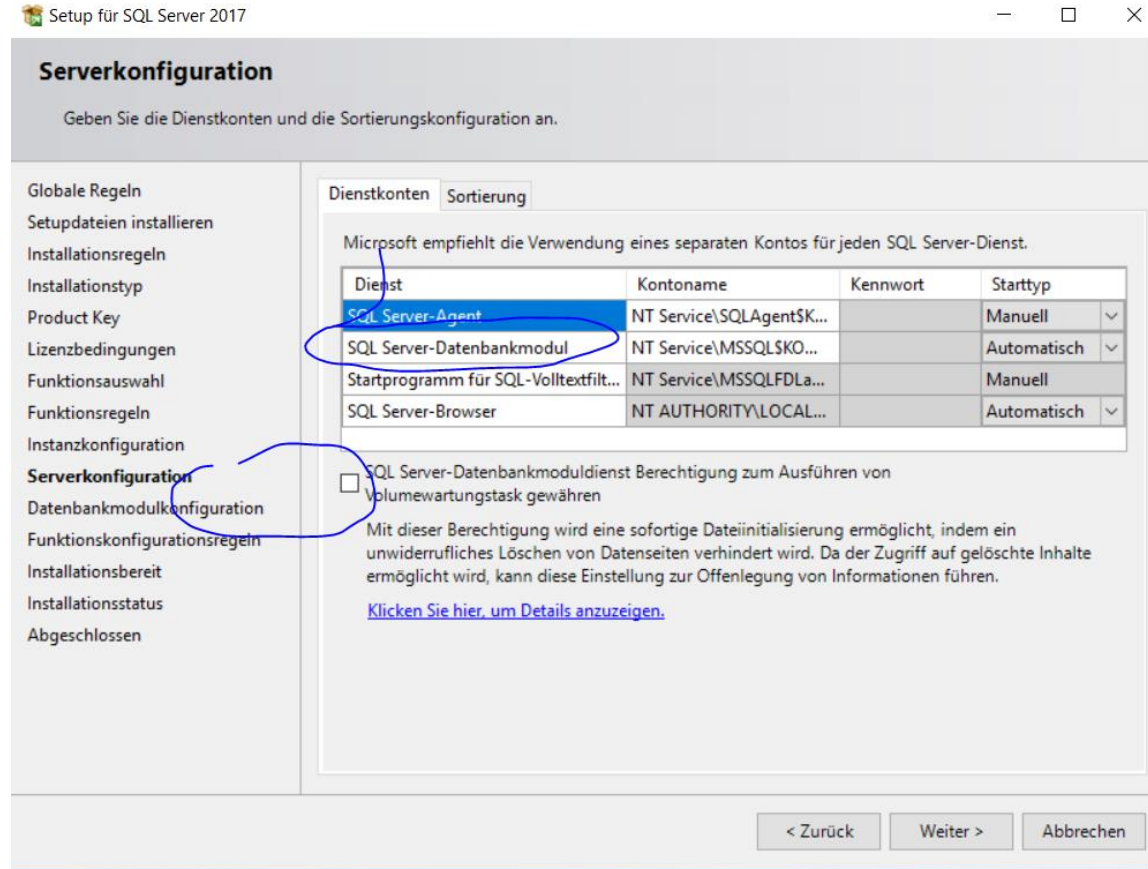
- Indizes müssen gewartet werden?
 - Reorg oder Neuerstellung
- Suche nach korrekten Indizes
- Suche nach doppelten, überflüssigen, fehlenden Indizes
- Gute Übersicht durch Systemsichten
 - Sys.dm_db_index_physical_stats

..was ist besser?

- Table Scan
- Index Scan
- Index Seek
- Clustered Index Seek

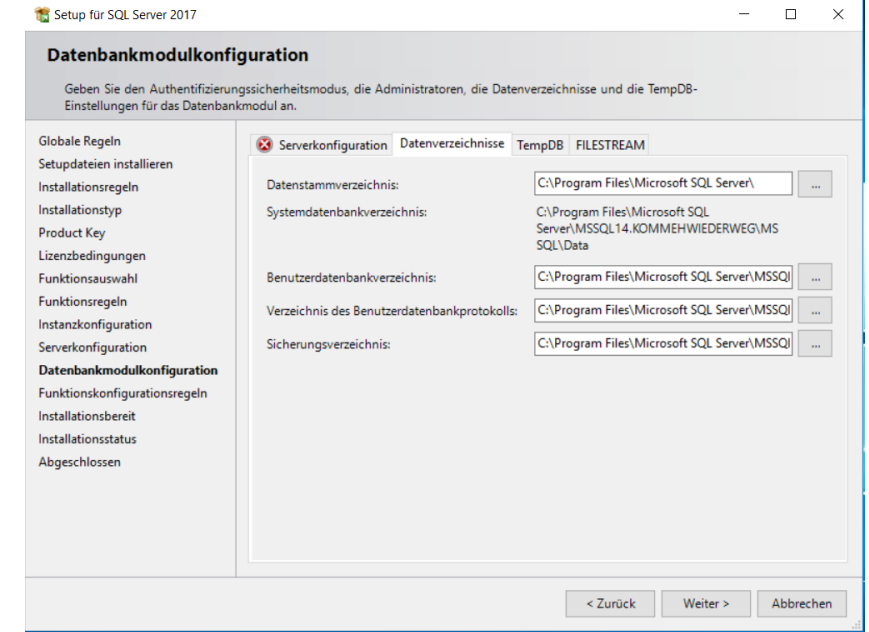
Lokale Sicherheitsrichtlinie

- Ausnullen



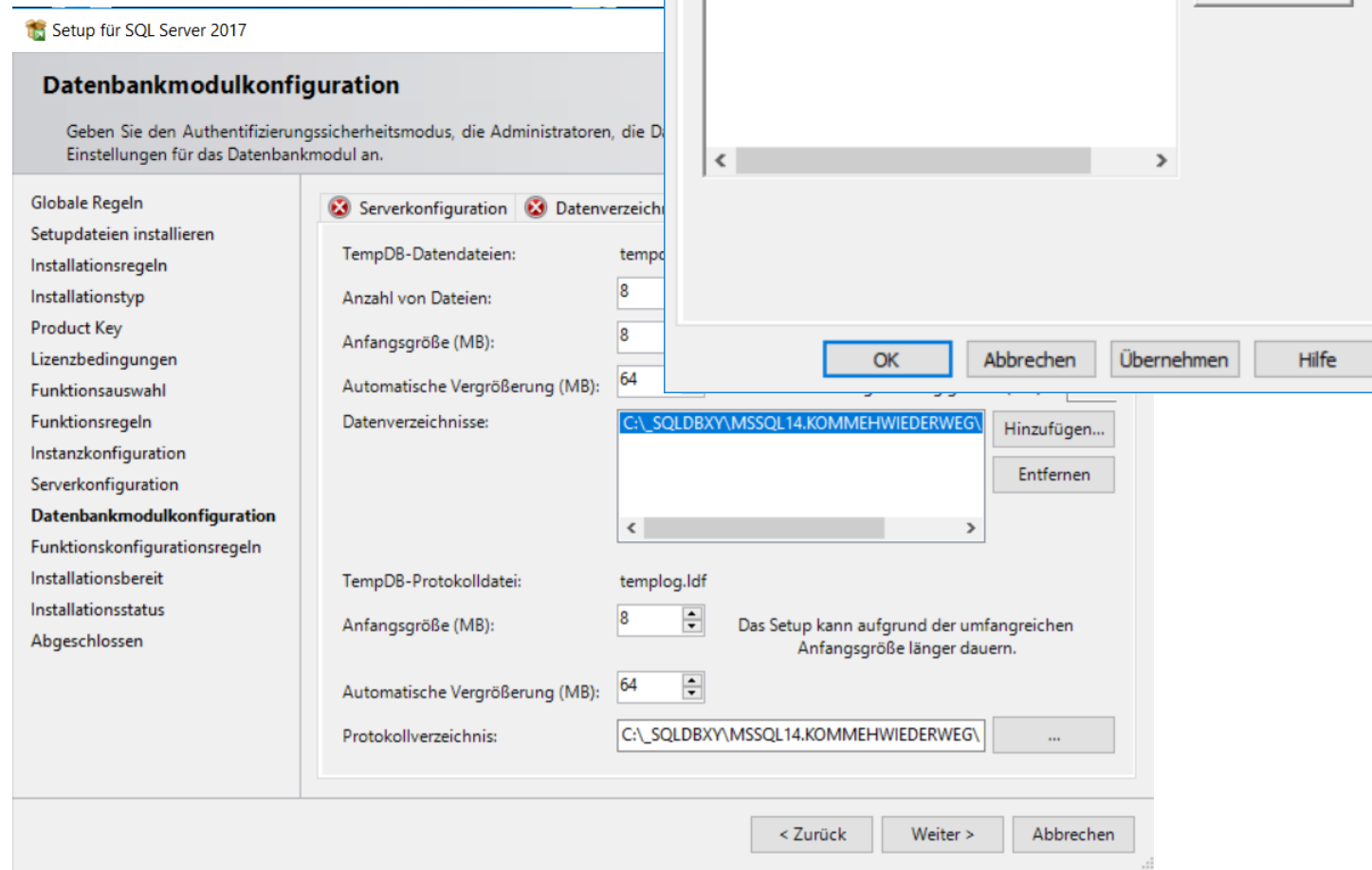
Datenbankverzeichnis

- Trenne Log von Daten
 - Lgfile sollte ungehindert Schreiben können
 - Logfile schreibt sequentiell
 - Datenfile random Zugriff
- Pauschalregel!
 - 50 LDF auf einer HDD sind auch kein Spaß



TempDB

- Gleiche Regel wie für normale DBs
 - Trenne Log von Daten
- Anzahl von Dateien:
 - Anzahl der Kerne; max 8
 - Soviele Dateien wie Kerne
- Traceflags
 - T1118; T1119
 - Gleich große Datendateien
 - Universal Extents
- Seit SQL 2016 default



Server Settings

- RAM
 - MIN Arbeitsspeicher (Garantie für SQL Server)
 - MAX Arbeitsspeicher (Garantie für andere)
 - Mit Arbeitsspeicher ist nur der BufferPool gemeint
 - Datenpuffer und Plan-cache, Tables, Indexes, Proc Cache, Lock Hash Tables
 - !! Der Min Speicher wird erst fix reserviert, wenn er auch von SQL verwendet wird
 - → GPO
- HDD
 - Trenne Log von Daten
 - Keine Regel für ewig → 100DBs → 100 Logfiles auf einem Datenträger
- CPU
 - Affinitäten sind gut eingestellt

MAXDOP

- Maximaler Grad der Parallelität
 - Wieviele CPUs verwendet SQL Server für eine einzelne Abfrage?
 - Default: 5 – 0
 - Ab 5 SQL Dollar alle CPUs
 - Laufende Abfragen sind von Änderungen nicht betroffen
 - kein Neustart
- Faustregel:
OLAP 50 und max 8 Core bzw 50%
- CXPACKET

Serversettings - vorher

Servereigenschaften - BOOM

Seite auswählen

- Allgemein
- Arbeitsspeicher
- Prozessoren
- Sicherheit
- Verbindungen
- Datenbankeinstellungen
- Erweitert
- Berechtigungen

Skript ? Hilfe

Arbeitsspeicheroptionen für den Server

Erweitert

Berechtigungen

Mindestmenge an Serverarbeitsspeicher (in MB):

0

Maximaler Serverarbeitsspeicher (in MB):

2147483647

Weitere Arbeitsspeicheroptionen

Arbeitsspeicher für Indexerstellung (in KB, 0 = dynamisch):

0

Mindestmenge an Arbeitsspeicher pro Abfrage (in KB):

1024

Verbindung

Server: BOOM

Verbindung: BOOM\Boss

FILESTREAM

FILESTREAM-Freigabename	MSSQLSERVER
FILESTREAM-Zugriffsebene	Vollzugriff aktiviert

Netzwerk

Netzwerkpaketgröße	4096
Timeout für Remoteanmeldung	10

Parallelität

Abfragewartezeit	-1
Kostenschwellenwert für Parallelität	5
Max. Grad an Parallelität	0
Sperren	0

Sonstiges

Querschwellenwert	1
-------------------	---

Standardspeicherorte für Datenbank

Daten: C:_SQL\

Protokoll: C:_SQL\

Sicherung: C:_BOOMBACKUP

Serversettings - nachher

Mindestmenge an Serverarbeitsspeicher (in MB):

8000

Maximaler Serverarbeitsspeicher (in MB):

10000

Standardspeicherorte für Datenbank

Daten: E:_SQL\

Protokoll: L:_SQL\

Sicherung: Z:_BOOMBACKUP

- Erweitert
- Berechtigungen

▼	FILESTREAM	
	FILESTREAM-Freigabename	MSSQLSERVER
	FILESTREAM-Zugriffsebene	Vollzugriff aktiviert
▼	Netzwerk	
	Netzwerkpaketgröße	4096
	Timeout für Remoteanmeldung	10
▼	Parallelität	
	Abfragewartezeit	-1
	Kostenschwellenwert für Parallelität	50
	Max. Grad an Parallelität	4
	Sperren	0
▼	Sonstiges	

Datenbank Settings

- Grundeinstellungen sind
 - Merkwürdig
 - SQL 2017: 8 MB Daten und 8 MB Logfile ; 64 MB Wachstumsrate
 - SQL 2014: 5MB Daten und 2 MB Logfile: 1 MB bzw 10% Wachstumsrate
- Wiederherstellungsmodel
 - Vollständig

Datenbank Design

- Theorie und Realität
 - Normalisierung vs Redundanz
 - Seiten und Blöcke
- Diagramm
 - Primärschlüssel ohne Fremdschlüssel
 - Datentypen
 - „Breite“ Tabellen

Optimierung per HDD - Salamiaktik

- Lastverteilung
- Dateigruppen
 - Tabellen auf andere Datenträger legen
- Partitionierte Sicht
 - Große Tabellen in viele kleine Tabellen splitten
- Partitionierung
 - Physikalische Partitionierung

Datenbank - Initialgrößen

- Wie groß ist die Dateien 3 Jahren
 - Hardwarewechsel
- Wachstumsraten
 - 10%??
- Wiederherstellungsmodell
 - Einfach – Voll – Massenprotokolliert
- Checkpoint
 - Indirekt ..alle 60 Sekunden

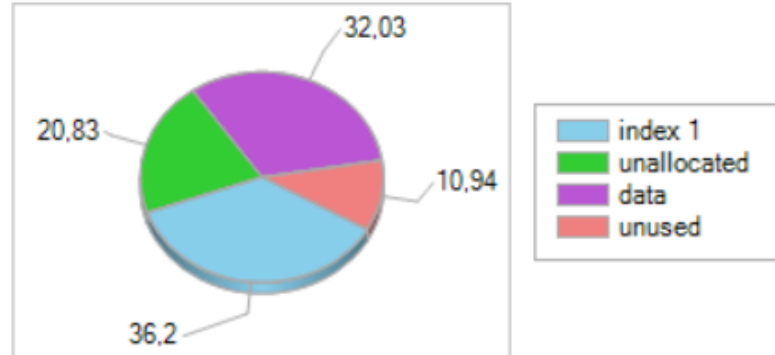
Verketteten von NULL-Werten ergibt NULL	False
Vertrauenswürdig	False
Verzögerte Dauerhaftigkeit	Allowed
▼ Wiederherstellung	
Seitenüberprüfung	CHECKSUM
Wiederherstellungszeit für das Ziel (Sekunde)	60
▼ Zustand	
Datenbank schreibgeschützt	False
Datenbankstatus	NORMAL
Verschlüsselung aktiviert	False
Zugriff beschränken	MULTI_USER

DB Vorher

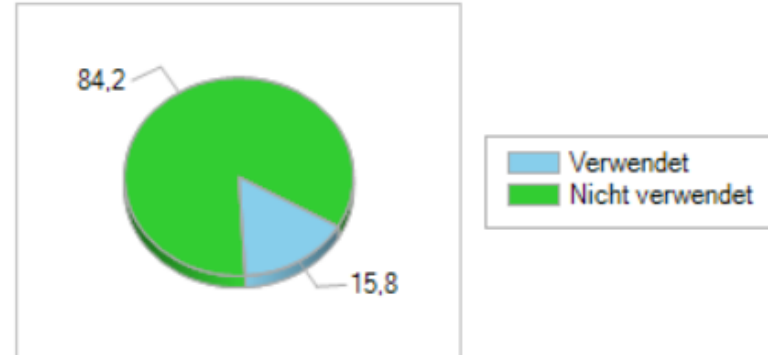
Dieser Bericht bietet einen Überblick über die Nutzung des Speicherplatzes in der Datenbank.

Gesamter reservierter Speicherplatz	6,88 MB
Reservierter Speicherplatz für Datendateien	3,00 MB
Reservierter Speicherplatz für Transaktionsprotokoll	3,88 MB

Speicherplatz für Datendateien (%)



Speicherplatz für Transaktionsprotokoll (%)



Es wurde kein Ereignisseintrag für automatische Vergrößerung/Verkleinerung für die Datenbank "mucdb" im Ablaufverfolgungsprotokoll gefunden.

+ Von Datendateien verwendeter Speicherplatz

DB Nachher

Datenträgerverwendung

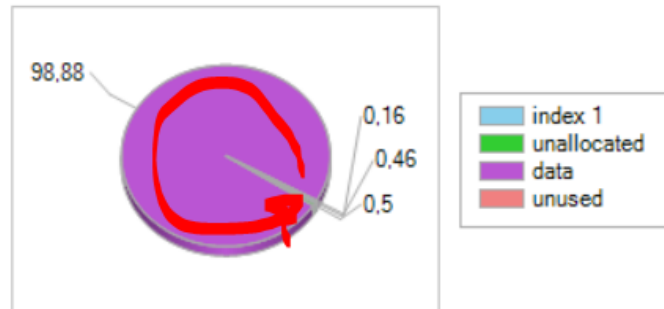
[mucdb]

am BOOM um 22.11.2018 14:27:57

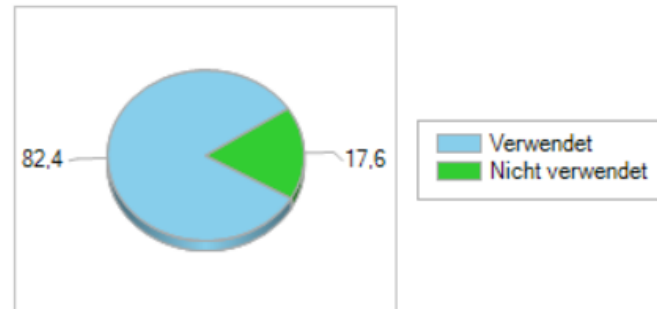
Dieser Bericht bietet einen Überblick über die Nutzung des Speicherplatzes in der Datenbank.

Gesamter reservierter Speicherplatz	265,88 MB
Reservierter Speicherplatz für Datendateien	238,00 MB
Reservierter Speicherplatz für Transaktionsprotokoll	27,88 MB

Speicherplatz für Datendateien (%)



Speicherplatz für Transaktionsprotokoll (%)

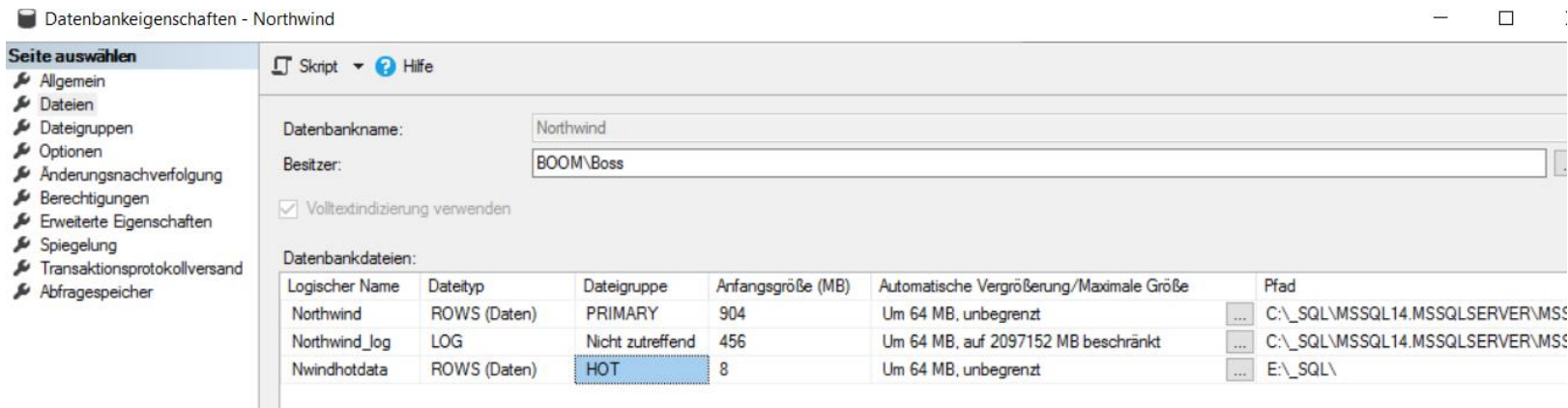


+ Ereignisse zur automatischen Vergrößerung/Verkleinerung für Daten-/Protokolldateien

+ Von Datendateien verwendeter Speicherplatz

Dateigruppen und CO

- Einfache Methode mehr HDDs ins Spiel zu bringen
- Tabellen können auf Dateigruppen gelegt werden
 - Create table () on Dateigruppe



Partitionierung

- Verteilung der Daten mit Hilfe einer
 - Partitionierungsfunktion
 - -----100-----200-----
 - Partitionierungsschema
 - DG1 ---- DG2 ---- DG3