

Schnellkurs Indizes

Andreas Rauch – ppedv AG

Agenda

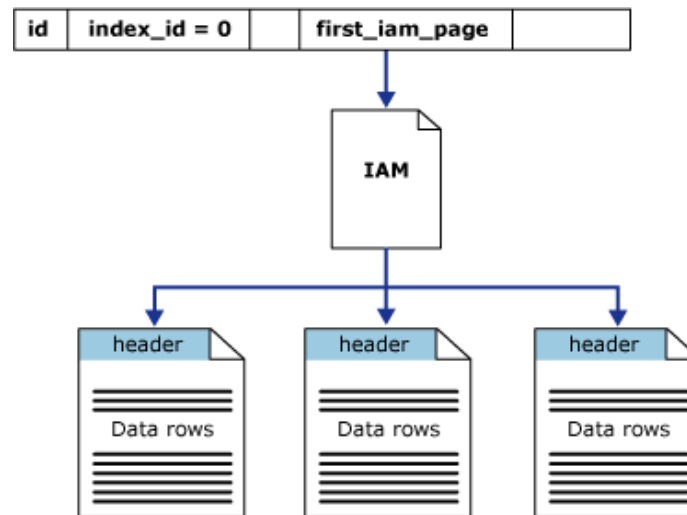
- Wie funktionieren denn Indizes?
- Welche Indizes gibt es denn?
- Welche Indizes sollte man denn erstellen?
- Wie wirken sich Indizes auf die Leistung aus?
- Worauf sollte man bei Indizes achten?

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

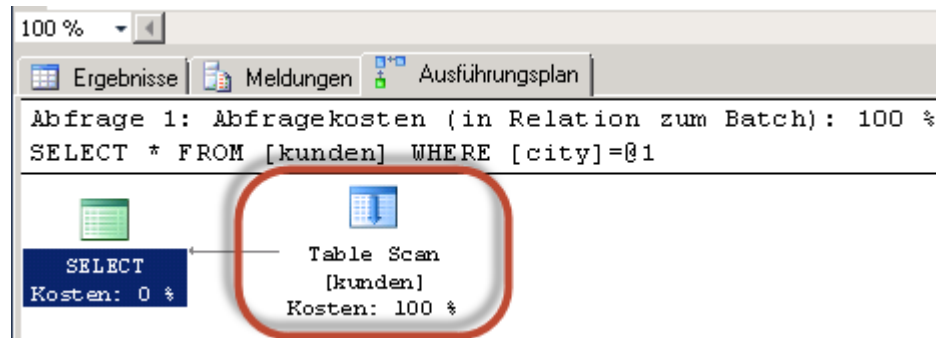
Heap

- 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

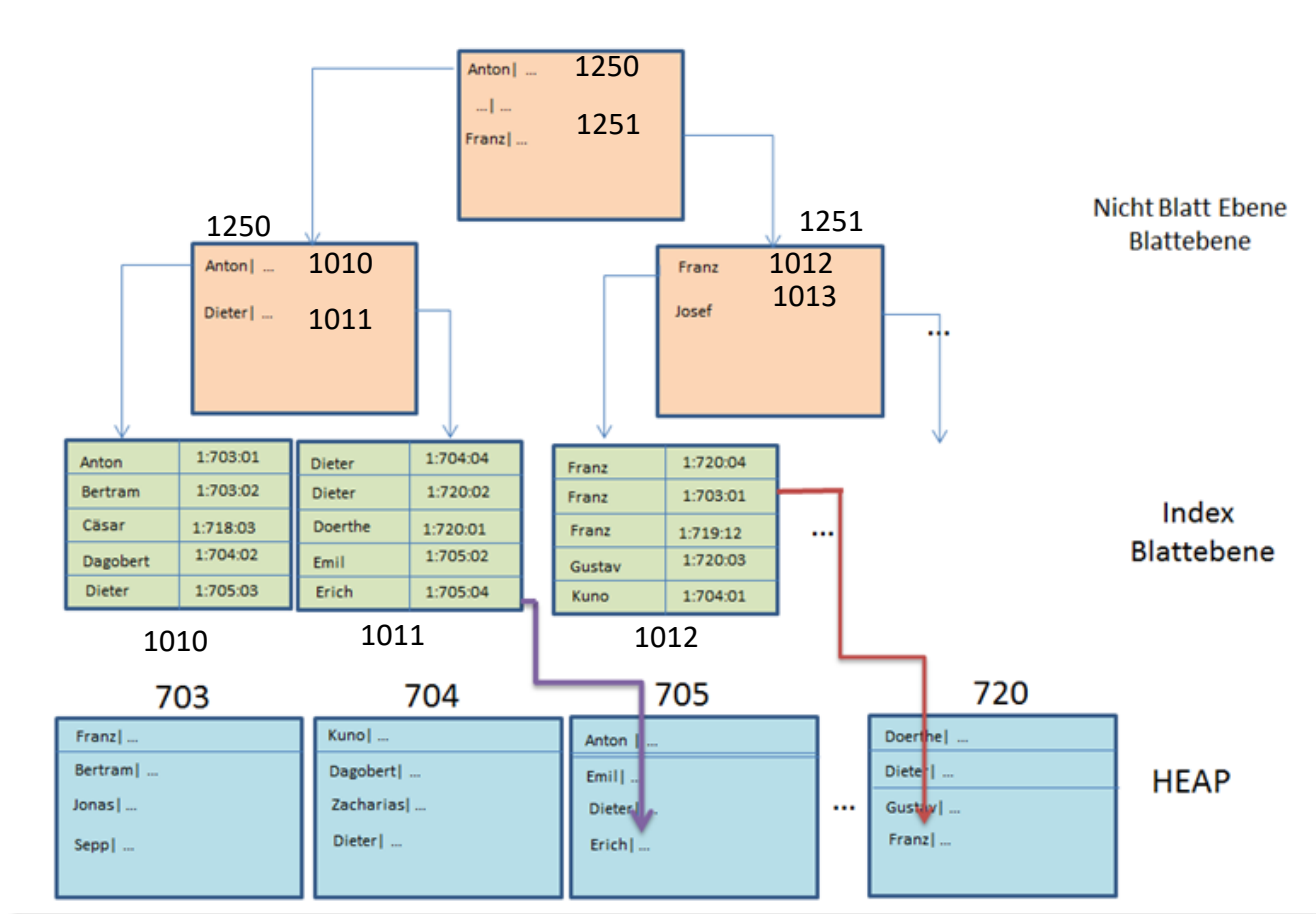


Wie funktioniert denn der Index?

- 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

Welche Indizes gibt es denn?

- Spaß bei Seite!

- Nur 2!

- Bzw. 3

- Nicht gruppierter Index

- Gruppierter Index

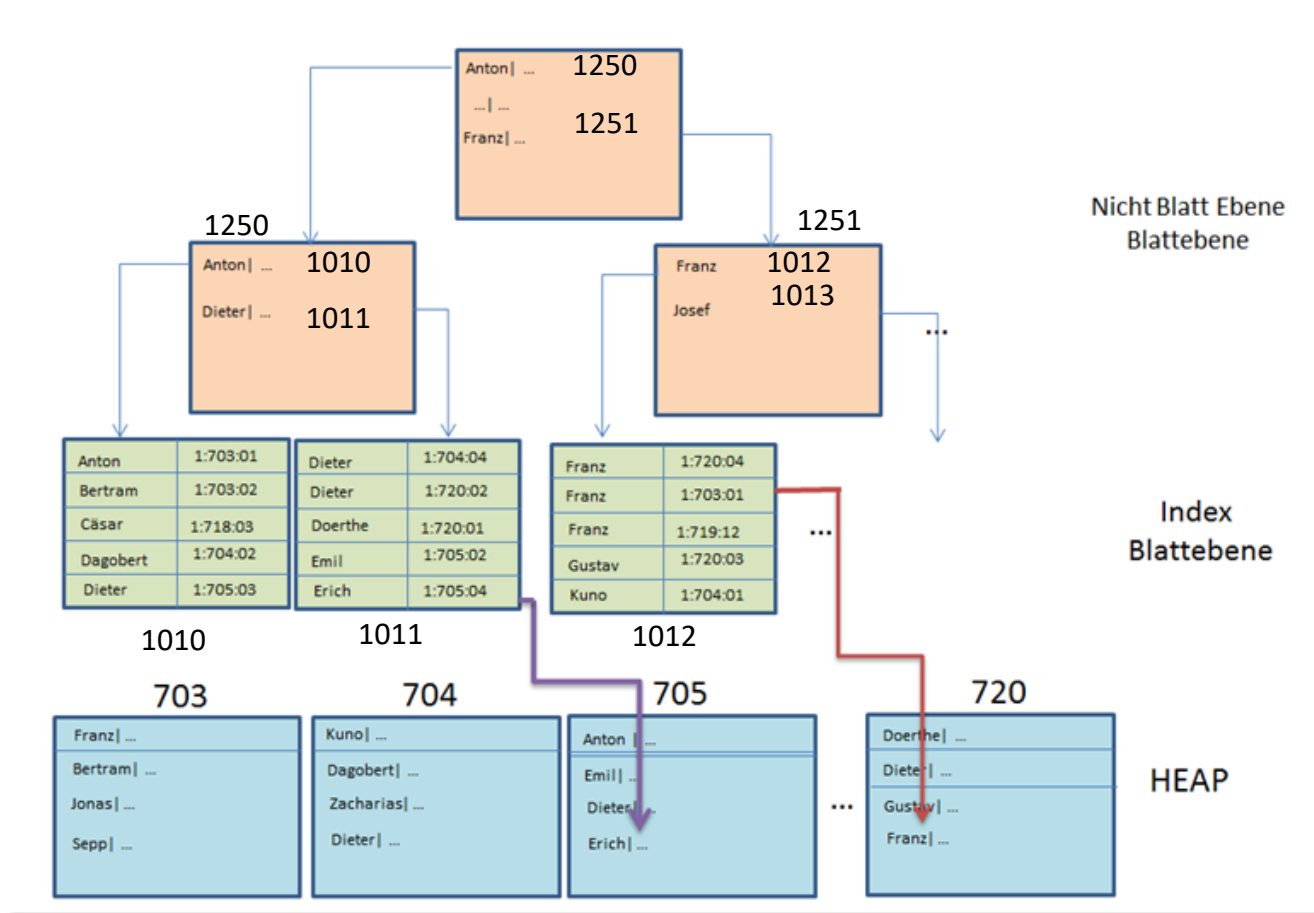
- Columnstored Index

- Spezialindizes: XML, Geo-Indizes

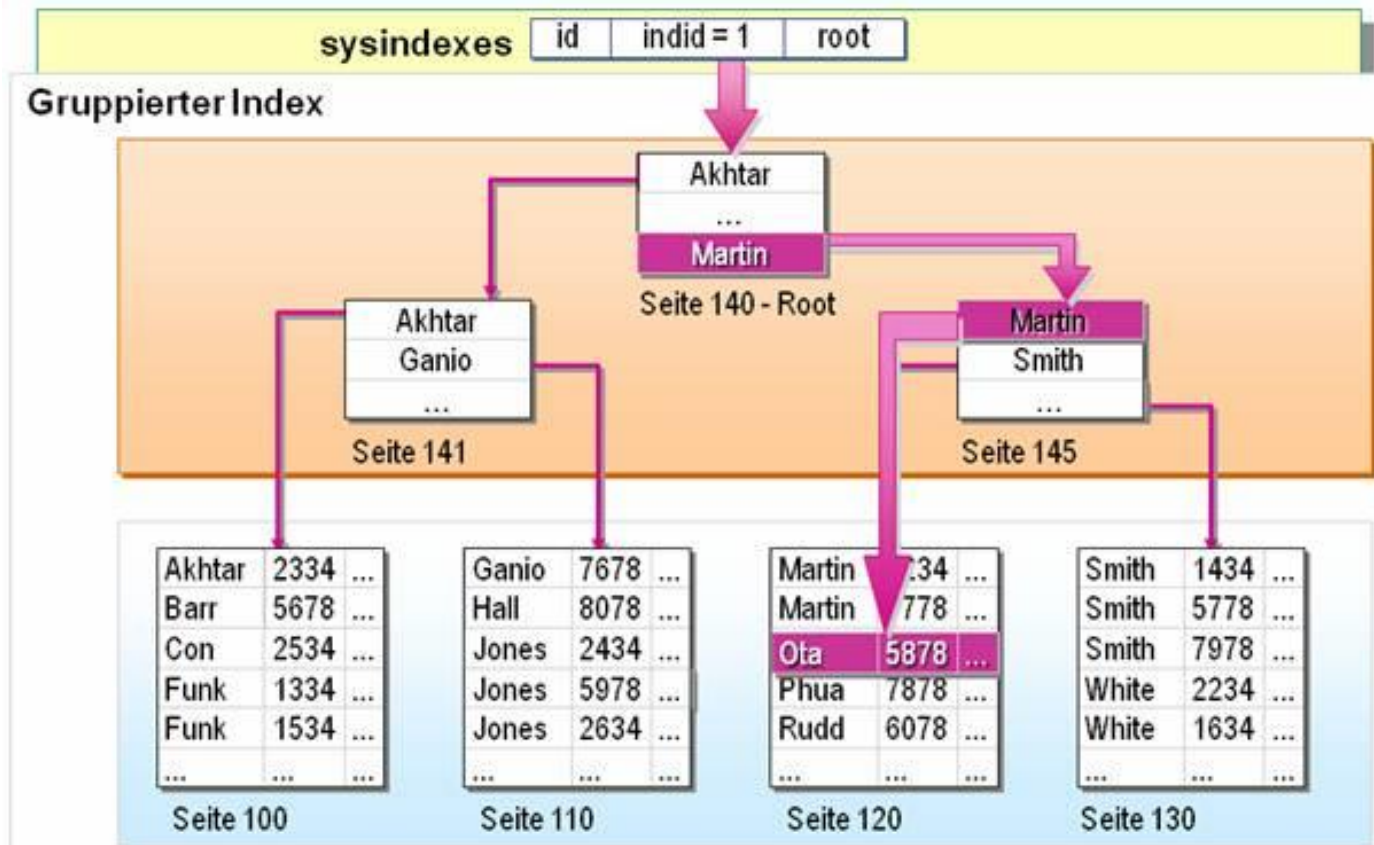
Wie funktioniert der Index?

- 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

- Gruppiertes 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 gruppiertes Index

- Sehr gut bei Abfragen auf rel. eindeutige Werte bzw. geringen Ergebnismengen: =
Kandidaten: ID; Firmenname, ...
kann mehrfach verwendet werden (999-mal)
- ➔ PK oft Gruppiertes 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

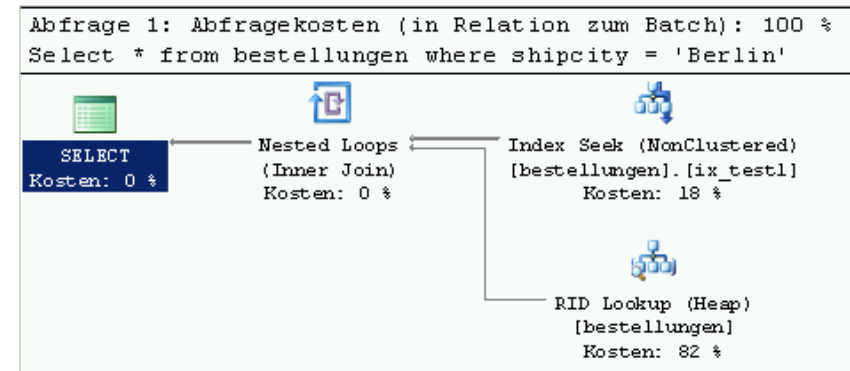
..und die anderen Indizes?

- Columnstored Index (ab SQL 2012)
 - Statt Datensätze werden Spalten in Seiten verwaltet
 - 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