



**Top Tips and New Features**  
SQL Server 2000: Best Practices *and*  
SQL Server 2005: Favorite Features

Kimberly L. Tripp  
President  
SQLskills.com

**SQL**  
skills.com  
immerse yourself in sql server

16 to 17 March 2005 | Interlaken, Switzerland

Microsoft  
**TechDays** 5

## Speaker – Kimberly L. Tripp

- Independent Consultant/Trainer/Speaker/Writer
- Founder, SYSolutions, Inc. [www.SQLskills.com](http://www.SQLskills.com)  
*email: Kimberly@SQLskills.com*  
*Become a subscriber on SQLskills.com and learn about new resources which can improve your productivity and server performance!*
- SQL Server MVP <http://mvp.support.microsoft.com/>
- Microsoft Regional Director  
<http://www.microsoftregionaldirectors.com/Public/>
- Writer/Editor for SQL Magazine [www.sqlmag.com](http://www.sqlmag.com)
- Coauthor MSPress: *SQL Server 2000 High Availability*
- Presenter/Technical Manager for SQL Server 2000 High Availability Overview DVD



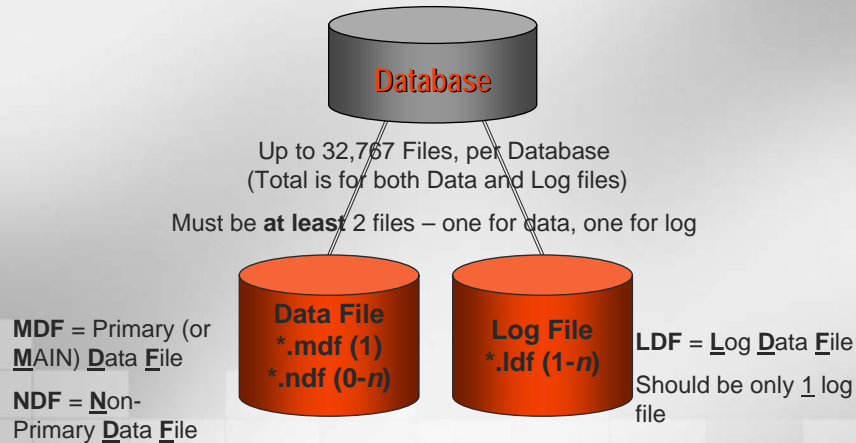
## Review

- **Optimizing Database Structures**  
Minimizing VLF Fragmentation
- **Index Internals**  
The Clustered Index Debate
- **Optimizing Recompilation**  
When Procedures *should* recompile
- **New Features in SQL Server 2005**
  - Partitioned Tables and Indexes
  - Snapshot Isolation
  - Online Index Rebuilds
  - SQLCMD



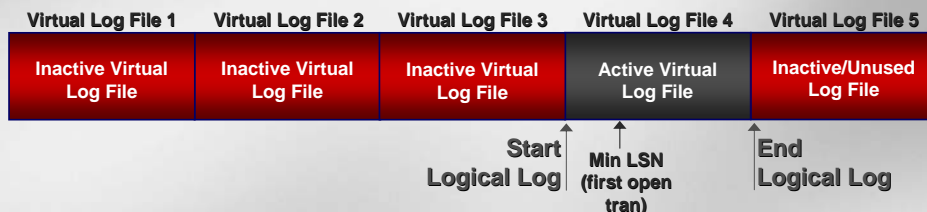
## Database Structure

Up to 32,767 Databases per Instance



TechDays

## How the Transaction Log Works



- On commit, activity is written to the log – active and likely sequential
- Activity moves through log sequentially, fills and then goes to a second file or autogrows
- Excessive autogrowth causes:
  - Slight pause on autogrow
  - Windows call to extend a file (may cause file fragmentation)
  - Adds “VLFs” to the file on each autogrowth

TechDays

## Minimize VLFs (1 of 3)

- Excessive VLFs add overhead to log related activities:
  - Transaction logging
  - Log Backups – may cause problems in clearing the log as well...
  - Logreader (replication)
  - After Triggers (*inserted/deleted*), etc.)
- Should have less than 50
- BPA (Best Practices Analyzer only generates warning if over 200)



## Minimize VLFs (2 of 3)

- Execute DBCC LOGINFO
  - Number of rows = Number of VLFs
- If excessive (> 50) then:
  - Free up log space by first clearing space from the transaction log (using BACKUP LOG)
    - ▶ If your Database Recovery Model is FULL or Bulk\_Logged then perform a regular transaction log backup – this should remove the inactive portion of the log
    - ▶ If your Database Recovery Model is Simple then clear the log



## Minimize VLFs (3 of 3)

- Shrink the transaction log file

```
DBCC SHRINKFILE(logfilename,  
TRUNCATEONLY)
```

- Alter the Database and modify log file size to a more appropriate size (*not requiring all of the autogrowth*) in ONE increment

```
ALTER DATABASE dbname  
MODIFY FILE  
( NAME = name,  
  SIZE = new_size )
```



## Index Concepts – Tree Analogy

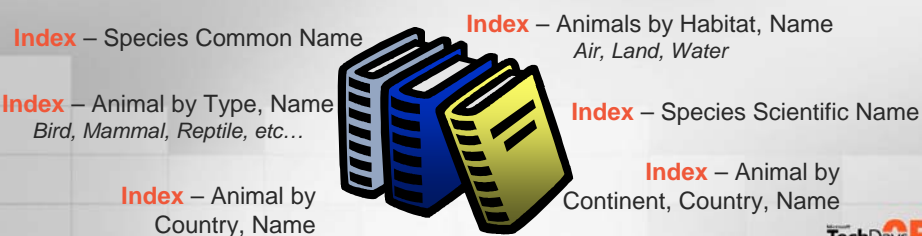
If a tree were data and you were looking for leaves with a certain property, you would have two options to find that data....

- 1) Touch every leaf – interrogating each one to determine if they held that property...SCAN
- 2) If those leaves (which had that property) were grouped such that you could start at the root, move to the branch and then directly to those leaves...**SEEK**



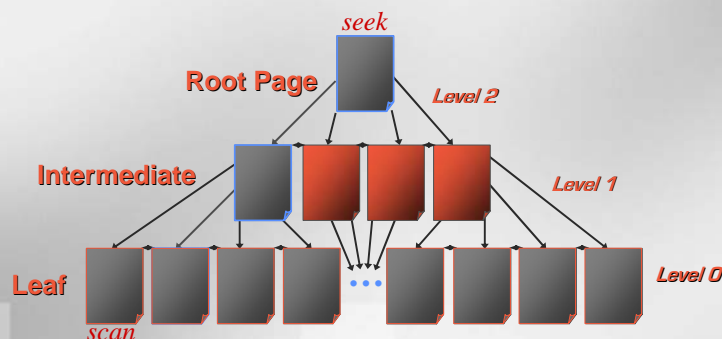
## Index Concepts – Book Analogy

- Think of a book – with indexes in the back
- The book has one form of logical ordering
- For references – you use the indexes in the back... to find the data in which you are interested you look up the key
- When you find the key – you must lookup the data based on its location... i.e. a “bookmark” lookup
- The bookmark always depends on the (book) content order



## Seek vs. Scan

- Seek – Starts at the Root and uses the Balanced Structure to move from top to bottom



- Scan – Moves through the Leaf Level from Left to Right (possibly Right to Left)



## Table Structure – Overview

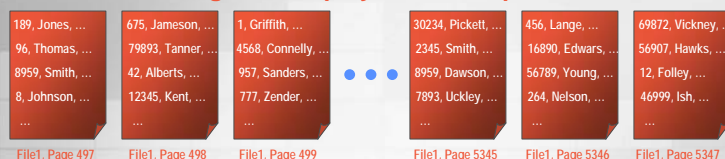
- HEAP – A table without a clustered index
- Clustered Table – A table with a clustered index
- Non-clustered Indexes DO NOT affect the base table's structure
- However, Non-clustered Indexes are affected by whether or not the table is Clustered...

TechDays 05

## Table Structure Heap

- Table without a Clustered Index
- Records are NOT ORDERED
- No Doubly-Linked List
- Access via Index Allocation Map (IAM)
  - IAMs = 8K Page (Chain) which Tracks Object Usage
  - 1 - IAM per Table/Index, per File, per 4 GB
- If NO Indexes exist – a full Table Scan required.  
At least 4000 I/Os on the Employee Table Heap.

### 4000 Pages of Employees in No Specific Order



TechDays 05

## Heap Issues – Why Cluster?

- **Heap has overhead**
  - Space
    - ▶ Fixed RID Assigned INSERT
    - ▶ Forwarding Pointers
  - Time
    - ▶ Negative for Table Scans (Negative for OLAP/DSS)
  - Optimized for SAVING space
  - Optimized for data loading – when empty and no updates (parallel data loading!!!)
- **Clustered Tables**
  - Usually a better choice for OLTP



## Table Usage

- Usage ALWAYS Defines Indexes Necessary
- **OLTP**
  - Predominantly Small Modifications
  - Lots of Users
- **OLAP**
  - Predominantly Large Analysis Queries
  - Typically a Smaller Number of Users
- **Combination OLAP/OLTP**
  - Hardest combination to optimize
  - Need to find a balance between too many and too few





## Clustered Index Overview

- Not Required – Although Highly Recommended
- Only One Per Table
- Physical Order Applied at Creation
- Logical Order Maintained through a Doubly-Linked List
- Requires ongoing and automated maintenance
- Need to choose WISELY!



## Clustered Index Criteria

- **Unique**
  - Yes – No overhead, data takes care of this criteria
  - NO – SQL Server must “uniquify” the rows on INSERT. This costs time and space. Each duplicate has a 4-byte “uniquifier.”
- **Narrow**
  - Yes – Keeps the NC indexes narrow
  - NO – Possibly wastes space
- **Static**
  - Yes – Improves Performance
  - NO – Costly to maintain during updates to the key

*In fact, an identity column that's ever increasing is often ideal...*



## Clustering on an Identity

### Pros

- **Naturally Unique**  
*(should be combined with constraint to enforce uniqueness)*
- **Naturally Static**  
*(should be enforced through permissions and/or trigger)*
- **Naturally Narrow**  
*(only numeric values possible, whole numbers with scale = 0)*
- **Naturally creates a hot spot...**
  - Needed pages for INSERT already in cache
  - Minimizes cache requirements
  - Helps reduce fragmentation due to INSERTs
  - Helps improve availability by naturally needing less defrag



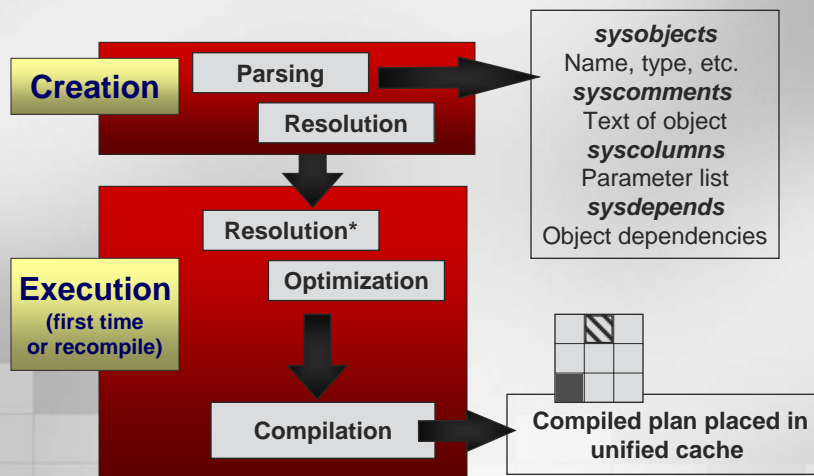
## Non-Clustered Index Overview

- Not Required – Although Critical to Achieving Optimal Performance
- Maximum of 249 Per Table
- Leaf Structure Separate from Base Table
- Based on the Heap's Fixed RID or Clustering Key
- Logical Order of Index Entries Maintained through a Doubly-Linked List
- By Far – the **FASTEST** Type of Range Index with very few exceptions!

*Don't ask for \*, limit your queries!!!*



## Processing Stored Procedures



## Recompilation Issues

RECOMPILATION = OPTIMIZATION

OPTIMIZATION = RECOMPILATION

- When do you want to recompile?
- What options do you have Recompilation?
- How do you know you need to recompile?
- Do you want to recompile the entire procedure or only part of it?
- Can you test it?

## When to recompile?

- When the plan for a given statement within a procedure is not consistent in execution plan (*due to parameter and/or data changes*)
- Cost of recompilation might be significantly less than the execution cost of a bad plan!

Why?

- Faster Execution with a better plan
- Saving plans for reuse is NOT always beneficial
- Some plans should NEVER be saved



## How do you know?

- You Test!
  - Test optimization plans consistency using EXECUTE WITH RECOMPILE
  - Choose what needs to be recompiled
    - ▶ Whole Procedure
    - ▶ Portions of the procedure
  - Test final performance using strategy
    - ▶ Procedure Recompilation (CREATE with RECOMPILE)
    - ▶ Statement Recompilation (Dynamic String Execution)
    - ▶ Modularized Code (Sub procedures created with or without WITH RECOMPILE)



## Table's Shared Data/Indexes

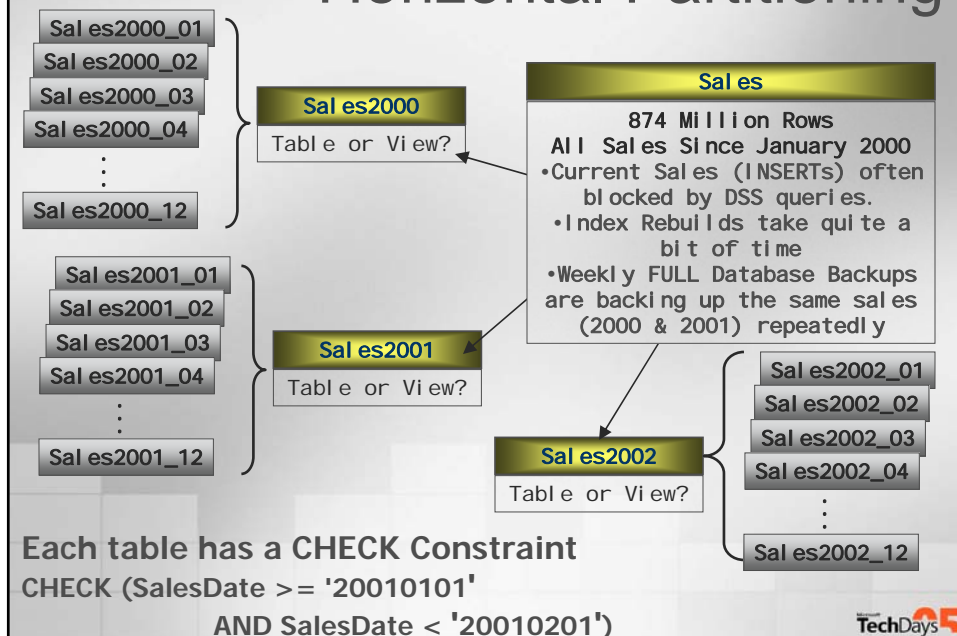
### ■ A Single Large Table...

- Presents different types of problems
  - ▶ Management
  - ▶ Index create/build or rebuilds
  - ▶ Backup/Restore and Recovery
  - ▶ Escalation
- May have different access patterns
  - ▶ Some data new – OLTP (inserts/updates for new and current sales)
  - ▶ Some data old for historical lookups – small singleton for OLTP lookups or larger for analysis

### ■ Does NOT have to be one table!



## Horizontal Partitioning



## Sliding Window Scenario

- Data Load
  - Single Table
    - ▶ Active Table impacted
    - ▶ Indexes need to be updated
  - Partitioned Table (2000/2005)
    - ▶ Table outside of active view manipulated
    - ▶ Indexes can be built separately of active tables
- Data Removal
  - Single Table – same problem
    - ▶ Active Table impacted
    - ▶ Indexes need to be updated
  - Partitioned Table (2000/2005)
    - ▶ Table can be removed from view definition *or*
    - ▶ Table can be switched out of partitioned table *and* then dropped

TechDays 05

## Data Partitioning

### Motivation

- Easy management of very large tables and indexes
- Maintain high availability through partition based operations
- Make effective use of large machines (16, 32,... CPUs) for any type of workload

TechDays 05



## Data Partitioning

### Basics

- Partitioning breaks a single object into multiple manageable pieces
- Partitioning is transparent to the application
- Partitioning applies to
  - Tables
  - Indexes
- The row is the unit of partitioning
- All partitions reside in a single database



## SQL 2005 Partitioned Tables

- Administrative Simplicity
  - Create a SINGLE table on a Partition Function
  - Partition Function defines each “range”
  - Partition Scheme names filegroup(s) to use and which function creates row to storage mapping
- Tables and Indexes can use the same functional breakdown ⇒ Aligned
- Tables and Indexes can use the same exact scheme ⇒ Storage Aligned
- In fact, if storage aligned desired then Table and Index can use exact same Function/Scheme



## SQL Server 2005 Partitioned Tables and Indexes

- Types of Partitioning = Range
  - Date ranges
  - Can be based on a function
  - Can create “list” partitions
- Implementing Partitioned Tables and Indexes
  - Partition Function
  - Partition Scheme
  - Partitioned Table
  - Partitioned Index



## Snapshot Isolation

- Increased data availability for read applications
  - Allows non-blocking consistent reads in an OLTP environment
  - Writers don't block readers
  - Readers don't block writers
- Permits writes, which can cause conflicts
  - BUT...includes mandatory conflict detection



## Snapshot Isolation

### Scenarios

#### ■ Business

- Reporting and ad hoc queries running concurrently with OLTP
- Read-mostly database with relatively few writes
- Migration from Oracle to SQL Server

#### ■ DBAs and Application Developers

- Consistent aggregates (e.g., AVG, SUM)
- Index intersections and index joins without escalating read scans to a higher isolation level
- Deadlock reduction

**Snapshot Isolation trades cost of concurrency (locking exclusion) for cost of CPU & I/O to construct transaction consistent view**



## Index Maintenance – Rebuild

#### ■ SQL Server 2000

- DROP and re-CREATE
- DBCC DBREINDEX
- CREATE with DROP\_EXISTING

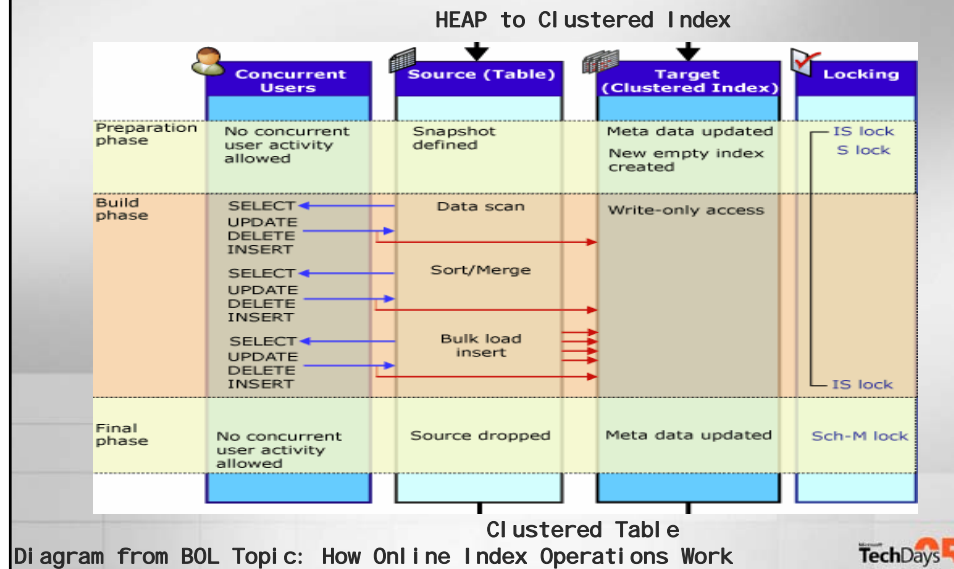
#### ■ SQL Server 2005

- ALTER INDEX...REBUILD replaces DBCC DBREINDEX
- Two modes of rebuilding with ALTER INDEX:
  - ▶ ONLINE – allows concurrent user access (queries as well as modifications) to the index during rebuild
  - ▶ OFFLINE – works using locks (same as 2000) DBCC DBREINDEX is OFFLINE only



## Index Maintenance

### How ONLINE Index Rebuilds Work



## Online Progress Report

SQL Profiler - [OnlineIndexOps (kltrippw2k3\sql2005idw11)]

EventClass	TextData	EventSubClass	ObjectID	ObjectName	BigIntData1	BigIntData2	Applicatio
ExistingConnection	-- network protocol: LPC set quoted_id...						SQL Serv
SQLBatchStarting	alter index test on member rebuild with (...)						SQL Serv
Progress Report: Online Inde...		1 - Start	213575799	test			SQL Serv
Progress Report: Online Inde...		2 - Stage 1 execution begin					SQL Serv
Progress Report: Online Inde...		6 - Inserted row count	213575799		7221	0	SQL Serv
Progress Report: Online Inde...		6 - Inserted row count	213575799		10000	0	SQL Serv
Progress Report: Online Inde...		3 - Stage 1 execution end					SQL Serv
Progress Report: Online Inde...		7 - Done	213575799	test			SQL Serv
SQLBatchCompleted	alter index test on member rebuild with (...)						SQL Serv
Trace Pause							
Trace Start							

alter index test on member rebuild with (online = on)  
go

Trace is running. Ln 11, Col 3 Rows: 29  
Connections: 1

## SQLCMD Favorites

```
C: \> sql cmd
1> exit

C: \> SET SQLCMDI NI=path\scri pt. sql
C: \> SET SQLCMDEDI TOR=notepad. exe

C: \> sql cmd
1> : LI STVAR
```

- Environment Variables
- SQLCMD Initialization Script
  - Runs on every execution
  - Great to give you connection status
  - Works for ALL executions of SQLCMD including the “dedicated admin” connection

©Kimberly L. Tripp  
Indexes from Every Angle

TechDays 05

## SQLCMD Favorites

```
: SETVAR DB AdventureWorks

BACKUP DATABASE $(DB)
TO DISK = ' \\KLTri ppW2K3\d$\SQLDemo\$(DB). bak'
WITH INIT
```

- Parameterization
- SETVAR variables defined within script and/or at the top of the script
- SQLCMD ... -v DB=“AdventureWorks” for parameterization
- Replaces all instances BEFORE any execution
  - Allows placement even in dynamic strings!

Watch your connections though (need GO)

©Kimberly L. Tripp  
Indexes from Every Angle

TechDays 05

## SQLCMD Favorites

```
:CONNECT Server1
BACKUP DATABASE $(DB)
    TO DISK = '\\Server\d$\SQLDemo\$(DB).bak'
    WITH INIT
go
:CONNECT Server2
SET NOCOUNT ON
```

- Can create multiple connections based on Windows Authentication or SQL Auth
- Can supply *-Username -Password*

©Kimberly L. Tripp  
Indexes from Every Angle



## SQLiMail (Database Mail)

- Not MAPI-based (yeah!)
- Multiple profiles
- Database specific
- Instantaneous send (releases transaction/process immediately) and queues emails
- Verified through system tables in enabled database





## Review

- **Optimizing Database Structures**  
Minimizing VLF Fragmentation
- **Index Internals**  
The Clustered Index Debate
- **Optimizing Recompilation**  
When Procedures *should* recompile
- **New Features in SQL Server 2005**
  - Partitioned Tables and Indexes
  - Snapshot Isolation
  - Online Index Rebuilds
  - SQLCMD



## Resources

- Check out [www.SQLskills.com](http://www.SQLskills.com) for information about upcoming **SQL Immersion** events, useful links and event scripts. All of the scripts used in this presentation will be available on the Past Events page within one week.
- Read my blog:  
<http://www.SQLskills.com/Blogs/Kimberly/>
- Subscribe to SQLskills:  
<http://www.sqlskills.com/login.aspx>
- MPress: *SQL Server 2000 High Availability*  
Authors: Allan Hirt with Cathan Cook, Kimberly L. Tripp and Frank McBath  
ISBN: 0-7356-1920-4  
On the SQLskills.com homepage you can download a sample chapter!



## Resources

- WP: SQL Server 2005 Snapshot Isolation  
Released on MSDN, link on home page:  
[www.SQLskills.com](http://www.SQLskills.com)
- WP: SQL Server 2005 Partitioned Tables  
Released on MSDN, link on home page:  
[www.SQLskills.com](http://www.SQLskills.com)
- MSDN Whitepapers, Access “Library”  
<http://msdn.microsoft.com/library>  
then “Servers and Enterprise Development”  
then “SQL Server”



## Resources

- *Using Partitions in a SQL 2000 Data Warehouse*  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq12k/html/partitionsindw.asp>
- *Improving Performance w/SQL Server 2000 Indexed Views*  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq12k/html/indexedviews1.asp>
- *Statistics Used by the Query Optimizer in SQL Server 2000*  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq12k/html/statquery.asp>
- *SQL Server 2000 Index Defragmentation Best Practices*  
<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/ss2kidbp.msp>



## Resources

- Support Webcast: Indexing for Performance – Finding the Right Balance (recorded 11 June 2004)  
<http://msevents.microsoft.com/CUI/EventDetail.aspx?EventID=1032254503&Culture=en-US>
- Support Webcast: Indexing for Performance – Proper Index Maintenance (recorded 19 July 2004)  
<http://msevents.microsoft.com/CUI/EventDetail.aspx?EventID=1032256511&Culture=en-US>



## Resources

- Support WebCast: SQL Server 2000 Profiler: What's New and How to Effectively Use It  
<http://support.microsoft.com/default.aspx?scid=%2Fservicedesks%2Fwebcasts%2Fwc111400%2Fwcblurb111400%2Easp>
- Whitepaper: Index Tuning Wizard for Microsoft SQL Server 2000  
<http://msdn.microsoft.com/library/en-us/dnsq12k/html/itwforsql.asp?frame=true>



## Resources

- See [www.microsoft.com/sql](http://www.microsoft.com/sql) for all sorts of stuff!
- Have the latest BOL? See [www.microsoft.com/sql](http://www.microsoft.com/sql)
- Go to the SQL Server Development Center  
<http://msdn.microsoft.com/sql/>
- For TechNet articles use:  
<http://www.microsoft.com/technet/prodtechnol/sql/default.asp?frame=true>
- MSDN Webcasts:  
<http://support.microsoft.com/default.aspx?PR=pwebcst&FR=0&SD=MSDN&LN=EN-US&CT=SD&SE=NONA>
- Support Resources:  
<http://www.microsoft.com/sql/support/default.asp>



# SQL skills

Please fill out your evaluation  
Thank you!

**Kimberly L. Tripp**

Consultant . Trainer . Writer . Speaker

email: [Kimberly@SQLskills.com](mailto:Kimberly@SQLskills.com)

Make sure to register for special offers  
and other helpful information and resources!

[www.SQLskills.com](http://www.SQLskills.com)



© 2005 Microsoft Corporation. All rights reserved.  
This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary.

