Indexing for Performance

Index Defragmentation Best Practices

Kimberly L. Tripp
Founder, President – SYSolutions, Inc.
www.SQLskills.com

Instructor – Kimberly L. Tripp

- Independent Consultant/Trainer/Writer/Speaker
- President, ***SY*Solutions, Inc.** (www.SQLskills.com)
  - EMAIL: Kimberly@SQLskills.com
- SQL Server MVP (http://mvp.support.microsoft.com/)
- Microsoft Regional Director (http://www.microsoftregionaldirectors.com/Public/)
- Writer/Editor for TSQL Solutions/SQL Magazine (www.tsqlsolutions.com and www.sqlmag.com)
- Presenter/Technical Manager for SQL Server 2000 High Availability Overview DVD (MS# 098-96661)
- Coauthor – MSPress title *SQL Server 2000 High Availability*
- Very approachable. Ask questions!

Overview

- How Fragmentation Occurs
- What Fragmentation Means
- How to See Fragmentation
- How to Minimize Fragmentation
- Rebuilding an Index
  - Why
  - When
  - How
- Automating Index Rebuilds
- Looking Forward…
  SQL Server 2005

How Fragmentation Occurs

- Heaps do not have 50/50 Page Splits, only forwarding pointers (no fragmentation, per se).
- Clustered Tables
  - INSERTS: Can be main problem if Primary Key is not monotonically increasing/decreasing.
  - UPDATES: Depend on Row Definition and existence of variable width columns
  - DELETES: Are a relatively minor problem compared to INSERTs/UPDATEs as they leave gaps (depending on type of deletes) – gaps that can be used for INSERTs/UPDATEs
- Nonclustered Indexes
  - INSERTS: Very likely to cause splits – unless increasing
  - UPDATES: Only when a variable width column in the index changes and/or the change causes the record to change location in the index

Index Fragmentation (Leaf Level)

Data Modifications [can] lead to Fragmentation

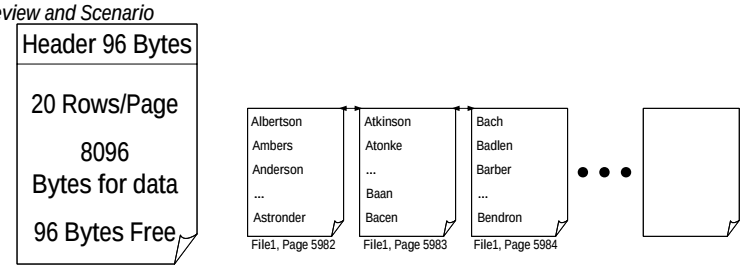
- INSERT
  - Yes – Key value is not ever increasing/decreasing
  - NO – Key is ever increasing/decreasing
- UPDATE
  - Yes – Updates… to variable width columns – where the values are getting wider
  - NO – Columns are fixed width, columns have "place holder" values (i.e. DEFAULT constraints) to minimize row expansion on update OR no updates
- DELETE
  - Yes – Deletes are singleton deletes (swiss cheese problem)
  - NO – Deletes are RANGE deletes for archival purposes

How Fragmentation Occurs

Modifications Can Lead to Page Splits

- INSERTs to a page which is already full
  (there is a special case when the INSERT is greater than the last row or less than the first row – on the page)
- UPDATEs to a variable width column on a page which is already full

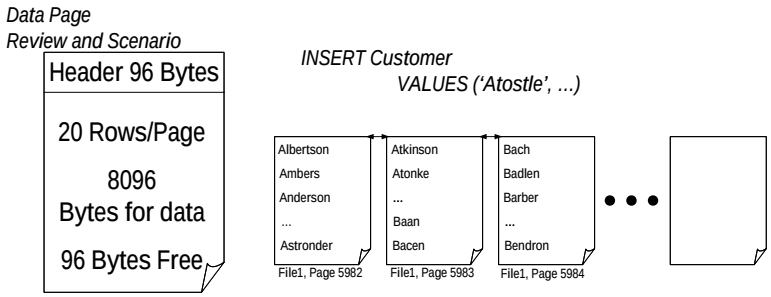
Data Page
Review and Scenario



How Fragmentation Occurs

Modifications Can Lead to Page Splits

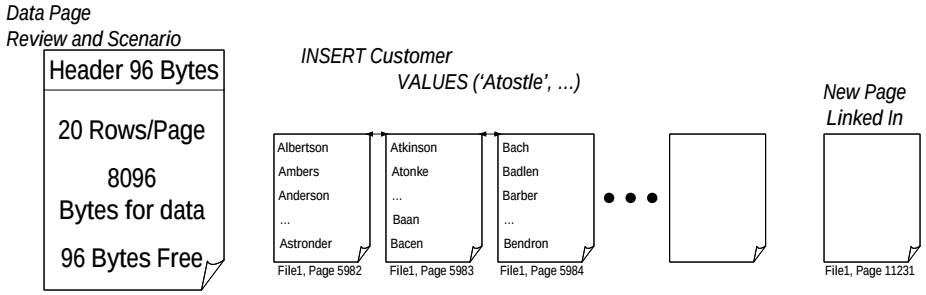
- INSERTs to a page which is already full
  (there is a special case when the INSERT is greater than the last row or less than the first row – on the page)
- UPDATEs to a variable width column on a page which is already full



How Fragmentation Splits Pages

If the page is full – on INSERT or UPDATE

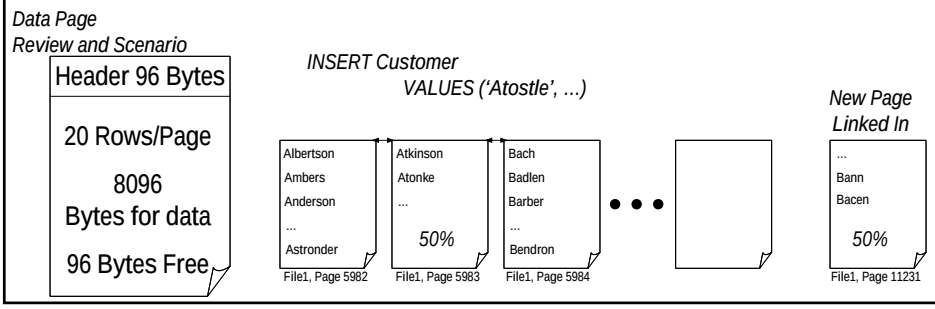
1) A New Page is "linked" in



How Fragmentation Splits Pages

If the page is full – on INSERT or UPDATE

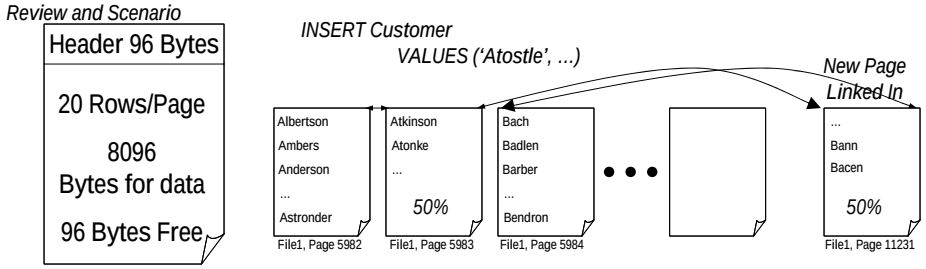
1) A New Page is "linked" in
2) 50% of the page being split is moved to the newly linked in page



How Fragmentation Splits Pages

If the page is full – on INSERT or UPDATE

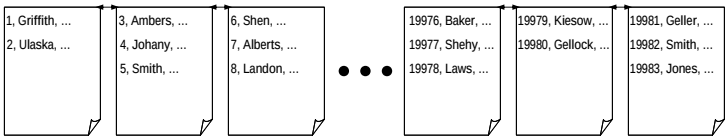
1) A New Page is "linked" in
2) 50% of the page being split is moved to the newly linked in page
3) Doubly-linked list/Pointers are updated to LOGICALLY follow the list



What Fragmentation Means

Internal Fragmentation

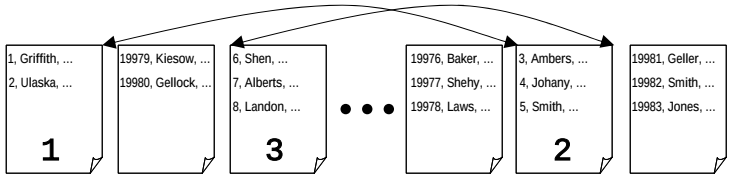
- Free Space on the Page
  - Desirable for OLTP
    - Free Space for New Records
    - Free Space for Updates
  - Undesirable OLAP
    - Table requires more pages
    - Large Analysis queries require more I/O



What Fragmentation Means

External Fragmentation

- Logical Left to Right Order of Leaf Level is NOT Physically Left to Right
  - Almost Always A Concern
  - Typically means splits have also resulted in internal fragmentation
  - Can cause disk thrashing for low cache and low selectivity queries



How to See Fragmentation

- Poor Query Performance over time
- More disk activity
- Poor Cache Utilization
- Verify Query I/O
  - SET STATISTICS IO ON
- Verify Scan Density
  - DBCC SHOWCONTIG
    - Manually
    - Programmatically
    - Automatically
- Periodically re-verify Query I/O

How To See Fragmentation

- Manually execute DBCC SHOWCONTIG
- May take some time on large tables – use FAST to see if the table is in need of a DEFRAG

```
DBCC SHOWCONTIG scanning 'member' table...
Table: 'member' (2025058250); index ID: 1, database ID: 7
TABLE level scan performed.
- Pages Scanned................................: 183
- Extents Scanned..............................: 24
- Extent Switches..............................: 34
- Avg. Pages per Extent........................: 7.6
- Scan Density [Best Count:Actual Count].......: 65.71% [23:35]
- Logical Scan Fragmentation ..................: 3.83%
- Extent Scan Fragmentation ...................: 12.50%
- Avg. Bytes Free per Page.....................: 1605.0
- Avg. Page Density (full).....................: 80.17%
DBCC execution completed. If DBCC printed error messages,
contact your system administrator.
```

DBCC SHOWCONTIG Output

- External Fragmentation
  - Scan Density
    - Uses a percentage to show extent switches
  - Logical Scan Fragmentation
    - Determines whether or not the pages owned – are contiguous (useless on a HEAP)
  - Extent Scan Fragmentation
    - Determines whether or not the extents owned have gaps (interleaved objects/indexes)
- Internal Fragmentation
  - Average Bytes Free Per Page
    - Amount of free space (if OLTP +, if OLAP -)
  - Average Page Density (fullness)
    - Shown as a percentage – how FULL are the pages

DBCC SHOWCONTIG Options

- DBCC SHOWCONTIG
  - All Tables/Heap Structures Only
  - No Nonclustered Indexes
- DBCC SHOWCONTIG (*tablename*)
  - Only the Table/Heap Structure for specified tablename
- DBCC SHOWCONTIG (*tablename*) WITH ALL_INDEXES
  - Includes all Nonclustered Indexes
- DBCC SHOWCONTIG (*tablename*) WITH ALL_INDEXES, TABLERESULTS
  - Returns the results in a tabular result set – excellent for programmatic analysis and automation!

DBCC SHOWCONTIG w/ TABLERESULTS

- Programmatically execute and analyze DBCC SHOWCONTIG results

```
INSERT #ShowContigOutput
  EXEC ('DBCC SHOWCONTIG WITH TABLERESULTS')
go
SELECT ScanDensity FROM #ShowContigOutput
  WHERE ObjectName = 'TableName'
  AND IndexName = 'IndexName'
go
```

- See Script for Temp Table Definition and Creation
  "DBCC ShowContig with Tableresults.sql"

How to Minimize Fragmentation

- Add Internal Space on Pages to Allow Fluctuation in Row Size (due to updates against variable width columns)
- How?
  - Rebuild Index Structures with appropriate FILLFACTOR Setting
- Other Data Dependant Options?
  - Use Only Fixed Width Columns
  - No Updates
  - No Deletes

FILLFACTOR

FILLFACTOR impacts ONLY the LEAF level of an index (with PAD_INDEX affects B-tree)

- 0 – Default value w/special meaning. Leaf level is filled to 100%
  - Excellent for OLAP
  - Not ideal for OLTP
- 100 – Leaf level is filled to 100% (same as default of 0)
  - Excellent for OLAP
  - Not ideal for OLTP
- 1-99
  - IDEAL for OLTP – you know your data!
  - *But how do you* OPTIMALLY *set this?*

FILLFACTOR

- Estimating an approximate FILLFACTOR
  - *What is going to cause splits in this structure?*
    - UPDATEs to variable width datatypes?
      - How wide? How volatile?
        - The wider and/or more volatile ⇨ LOWER FillFactor
    - INSERTs?
      - How often? Is the data ever-increasing?
        - The more volatile ⇨ LOWER FillFactor
  - *How often can you rebuild?*
    - The more frequent ⇨ HIGHER FillFactor
- Test your Guesstimate!
  - Use DBCC SHOWCONTIG between rebuilds to see how far and how fast the scan density drops
    - The faster the table becomes fragmented (i.e. the lower the scan density goes)
      - ⇨ LOWER FillFactor
        - *or* DECREASE the time between defrag/rebuilds

Restructuring an Index for more optimal Performance…

- Why?
- When?
- How?
  - Rebuilding an Index
  - Defraging an Index
- Automation!

Why?

- Minimize the Number of Pages Required to Store the Data
- Minimize Cache Requirements
- Improve OLTP Performance
  - Inserts faster – no Splits
  - Updates faster – no relocation/splits
- Improve OLAP Performance – Fewer Pages to Read (Condensed)
- Less Resources Required

When?

- Depends on the Table
  - OLTP – Depends on the level of modifications
  - DSS – Based on build strategy
    - Drop all indexes (NC first, then CL)
    - Load data into HEAP
    - Create all indexes (CL first, then NC, use a Fillfactor of 100)
- Depends on the Usage
  - Large Queries
  - Amount of Cache
- Performance Gains Outweigh the Maintenance Requirements and Potential Inaccessibility (due to Locking) of the Table

How?

- Rebuilding an Index
  - DROP and re-CREATE
  - DBCC DBREINDEX
  - CREATE with DROP_EXISTING
- Defraging an Index
  - DBCC INDEXDEFRAG
- Comparisons

Rebuilding an Index

- Generally better results w/rebuild v. defrag
- Completely removes all levels of fragmentation – from both the leaf level as well as the b-tree (i.e. completely rebalances the tree)
- Completely updates statistics – with the equivalent of a "full scan" (i.e. accurate statistics)
- Requires Locks ⇨ downtime
  - Rebuild CL requires Exclusive Table Lock
  - Rebuild NC requires Shared Table Lock
- How?
  - DBCC DBREINDEX (easier to automate)
  - CREATE w/DROP_EXISTING
    (can change the CL index definition)

Defraging an Index

- Does not completely rebuild ALL levels of index – focuses on LEAF level
- Does not update statistics
- Does not require locks for length of transaction – defrag executes as mini-trans
- May take longer if table is extremely fragmented
- May take less time if table is not fragmented
- Does not run as a single large transaction – transaction log backups can execute and the transaction log space can be freed WHILE running
- How ⇨ DBCC INDEXDEFRAG

How – Drop and Recreate

- Pros - None, really!
- Cons
  - Primary or Unique Key
    - Cannot drop without dropping Foreign Key references first
  - If you drop/re-create the Clustered you will rebuild ALL non-clustered indexes twice!
    - Drop causes the Table becomes a HEAP
      - ∴ All non-clustered are rebuilt
    - Recreate causes the table to become clustered again
      - ∴ All non-clustered are rebuilt again
  - Not Easy to Automate
- Recommendation ⇨ Do Not Use

How – DBCC DBREINDEX

- Pros
  - Easy Syntax, Easy to Automate
  - Doesn't require Foreign Keys to be dropped first
  - SP2+ Only rebuilds the index specified (RTM/SP1 had a bug)
  - Completely rebuilds the LEAF Level AND the B-Tree
  - Completely updates statistics
- Cons
  - Causes locks to be held for the duration of the rebuild
    - Clustered Index Rebuild = eXclusive (X) Table Level Lock
    - Non-clustered Index Rebuild = Shared (S) Table Level Lock
  - Creates a large transaction – cannot be cleared from the log until completion i.e. a lot of log activity during the rebuild
  - Non-Clustered indexes are rebuilt if the clustered key is not unique
- Recommendations
  - ⇨ Excellent choice for automation!

How – CREATE w/DROP_EXISTING

- Pros
  - Synonym to DBCC DBREINDEX (so all of the same PROs)
  - Can be used to CHANGE the definition of the Clustering Key (this would force the nonclustered indexes to be rebuilt)
- Cons
  - More complex syntax, harder to automate
  - Non-Clustered indexes are rebuilt if the clustered key is not unique
- Recommendation
  - ⇨ Use when you want to change the Clustering Key

How – DBCC INDEXDEFRAG

- Pros
  - Does not cause blocking
  - Easy Syntax, Easy to Automate
  - Doesn't require Foreign Keys to be dropped first
  - Runs as Small Transactions – Can create a LOT log activity however the transaction log can be backed up WHILE this is running. This does not execute as a single large transaction.
- Cons
  - Does not update statistics
  - Does not rebuild the non-leaf level(s)
  - Does not move the table for better disk scans
- Recommendations
  - ⇨ Use When Table Must be Highly Available
  - ⇨ Most Frequent Usage for all Indexes
  - ⇨ Use for Quick Fix during Business Hours

Comparisons

	Option	If Used with Clustered Index Rebuilds NonClustered	Updates Statistics	Lock on Base Table for CL	Lock on Base Table for NC	Transaction Log Impact
Rebuilding	DROP and re-CREATE	Yes, twice	Yes	Exclusive	Shared	Longest Time, Most Activity
	DBCC DBREINDEX	Pre-sp2 YES (and only once) 2000sp2 NO - unless the CL Key is NON-unique. When the CL is not unique, the NC indexes are rebuilt EACH time the CL is rebuilt (the uniqifiers are regenerated).	Yes	Exclusive	Shared	Cannot Clear Until Complete
	CREATE with DROP_EXISTING	Same as above EXCEPT when the Clustering Key CHANGES - then all NC indexes are rebuilt only once. Otherwise, NO*	Yes	Exclusive	Shared	Cannot Clear Until Complete
	DBCC INDEXDEFRAG	No	No	N/A	N/A	Minimal Impact in sp1+

** In SQL Server 2000 sp2 DBCC DBREINDEX and CREATE with DROP_EXISTING are almost identical.The only exception is that CREATE with DROP_EXISTING can be used to CHANGE the definition of the clustered index.*

Automating Index Rebuilds

- Use a Loop/Cursor to Walk the Tables
- Programmatically Determine the Level of Fragmentation (Scan Density) for Each Index
- Only Rebuild those with a S.D. less than desired
- Rebuild the Clustered with DROP_EXISTING
  - The script uses this solely for the challenge of it
    - Index column list – use INSERT with sp_helpindex to get Index information
    - Uniqueness – also from “description” portion of sp_helpindex output
- Rebuild the Nonclustered with DBCC DBREINDEX
- Create as a Procedure
- Setup a Job!

Index Column List?

```
CREATE TABLE #IndexInfo
(
   IndexName    sysname,
   IndexDesc    varchar(210),
   IndexKeys    nvarchar(2126)
)
GO
INSERT #IndexInfo
   EXEC sp_helpindex tname
GO
SELECT * FROM #IndexInfo
GO
SELECT IndexKeys FROM #IndexInfo
   WHERE IndexName = indexname
go
```

Automating Index Rebuilds

DEMO – sp_IndexRebuilds

- sp_RebuildIndexes
  - To Rebuild All Indexes with a Scan Density < 100%
- sp_RebuildIndexes @ScanDensity = 80
  - To Rebuild All Indexes with Scan Density of < 80%
- sp_RebuildIndexes 'Authors'
  - To Rebuild All Indexes on the authors table with a Scan Density of < 100%
- sp_RebuildIndexes 'Authors', 80
  - To Rebuild All Indexes on the authors table with a Scan Density of < 80%

Object Name and ScanDensity are both optional parameters.
ScanDensity must be a whole number between 1 and 100.

SQL Server 2005

- DBCC SHOWCONTIG – replaced by fn_indexinfo

```
SELECT *
FROM fn_indexinfo (table, index,
partition_id, mode)
```

- Returns more details:
  - Number of fragments (if object is "interleaved" within the database) and average size of fragment
  - Index Depth – Total Levels in the index
  - Forwarded Records – only useful in HEAP
  - Snapshot details – Ghosted records

SQL Server 2005

- DBCC DBREINDEX – replaced by ALTER INDEX … REBUILD
- DBCC INDEXDEFRAG – replaced by ALTER INDEX … REORGANIZE
- Indexes can be rebuilt ONLINE
  - Long terms locks are not held for life of transaction
  - Not possible if table includes LOB data
- Indexes can be "disabled"
  - Metadata remains, index structure is removed
  - Index must be "rebuilt" through ALTER INDEX REBUILD or CREATE with DROP_EXISTING

Review

- How Fragmentation Occurs
- What Fragmentation Means
- How to See Fragmentation
- How to Minimize Fragmentation
- Rebuilding an Index
  - Why
  - When
  - How
- Automating Index Rebuilds
- Looking Forward…
  SQL Server 2005

Resources

- Check out www.SQLskills.com for information about upcoming events, useful downloads and excellent scripts! There are quite a few resources and/or links to use.
- MSPress title: *SQL Server 2000 High Availability A*uthors: Allan Hirt with Cathan Cook, Kimberly L. Tripp, Frank McBath
  ISBN: 0-7356-1920-4
- Check www.SQLskills.com to download for a sample chapter!

SQL skills

Resources

- Credit Database and Demo Scripts are on www.SQLskills.com, Events, Past Events
  - All the conference scripts are listed as zips in descending order based on conference dates
  - Find useful demo scripts in many areas!
- Articles on programming best practices, performance, etc. check out www.SQLskills.com, Articles – for a well ordered list of SQL Server Magazine Links
  - Many articles are free after registering on SQLMag
  - Check out the 5-part series on Joins including "n-Table Joins" where I discuss adding redundant keys. Instant Doc ID#23733

KB Articles Worth Reading!

- Q224587: *INF: Troubleshooting Application Performance with SQL Server*
- Q224453: *INF: Understanding and Resolving SQL Server 7.0 or 2000 Blocking Problems*
- Q271509: *INF: How to Monitor SQL Server 2000 Blocking*
- Q243586: *INF: Troubleshooting Stored Procedure Recompilation*
- Q263889: *INF: SQL Blocking Due to [COMPILE] Locks*

KB Articles Worth Reading!

- Q243588: HOW TO: Troubleshoot Ad-Hoc Queries
- 243589: HOW TO: Troubleshoot Slow-Running Queries
- 319942: HOW TO: Proper Configuration Settings
- 319892: INF: New Concurrency and Scheduling Diagnostics Added to SQL Server
- 305977: INF: Frequently Asked Questions - SQL Server 2000 - Table Variables
- 323630: INF: Resolving Blocking Problems That Are Caused by Lock Escalation in SQL Server

Resources

- Whitepaper: Microsoft SQL Server 2000 Index Defragmentation Best Practices http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/sql/maintain/Optimize/SS2KIDBP.asp
- Whitepaper: *Using Partitions in a Microsoft SQL Server 2000 Data Warehouse,* http://msdn.microsoft.com/library/techart/partitionsindw.htm
- *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

Resources

- Whitepaper: *Statistics Used by the Query Optimizer in Microsoft SQL Server 2000*, http://msdn.microsoft.com/library/techart/statquery.htm
- Whitepaper: *Query Recompilation in SQL Server 2000*, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsql2k/html/sql_queryrecompilation.asp
- Whitepaper: Index Tuning Wizard for Microsoft SQL Server 2000, http://msdn.microsoft.com/library/en-us/dnsql2k/html/itwforsql.asp?frame=true
- Whitepaper: *Improving Performance with Microsoft SQL Server 2000 Indexed Views*, http://msdn.microsoft.com/library/en-us/dnsql2k/html/indexedviews1.asp?frame=true

Support Resources

- Lots of Great links www.microsoft.com/sql
- Support Overview http://www.microsoft.com/sql/support/default.asp
- Newgroups – Listings, info on how to use, etc… http://www.microsoft.com/sql/support/newsgroups/default.asp
- Technical Resources http://www.microsoft.com/sql/techinfo/default.asp
- SQL Server Whitepapers (some already listed) http://www.microsoft.com/isapi/redir.dll?Prd=msdn&Ar=sqlserver

Resources

- See www.sqlmag.com & www.tsqlsolutions.com for articles on Backup/Restore
- From Books Online "Home Page" select White Papers to get to msdn
- For Tech Net articles use: http://www.microsoft.com/technet/prodtechnol/sql/default.asp?frame=true
- See www.microsoft.com/sql for all sorts of stuff!
- Support Resources listed: http://www.microsoft.com/sql/support/default.asp
- Webcasts: http://support.microsoft.com/default.aspx?PR=pwebcst&FR=0&SD=MSDN&LN=EN-US&CT=SD&SE=NONA

Community Resources

- Community Resources
  http://www.microsoft.com/communities/default.mspx
- Microsoft Regional Directors (MS RD)
  http://www.microsoftregionaldirectors.com/Public/
- Most Valuable Professional (MVP)
  http://www.mvp.support.microsoft.com/
- Newsgroups
  Converse online with Microsoft Newsgroups – worldwide!
  http://www.microsoft.com/communities/newsgroups/default.mspx
- User Groups
  http://www.microsoft.com/communities/usergroups/default.mspx

Questions and Answers

- Submit questions using the text box
- All questions will be answered either HERE and NOW and/or they will be included in a post-webcast write-up. I will post the Q&A and all additional resources on my blog:
  - www.SQLskills.com/blogs/Kimberly

Thank you!

Kimberly L. Tripp
Consultant . Trainer . Writer . Speaker

email: **Kimberly@SQLskills.com**
Make sure to register for special offers and other helpful information and resources!
www.SQLskills.com