

SQLIntersection

PRECON04

Practical Disaster Recovery Techniques

Paul S. Randal
CEO/Co-owner
Paul@SQLskills.com



Paul S. Randal



- **Consultant/Trainer/Speaker/Author**
- **CEO, [SQLskills.com](http://www.SQLskills.com)**
 - Email: Paul@SQLskills.com
 - Blog: <http://www.SQLskills.com/blogs/Paul>
 - Twitter: @PaulRandal
 - 5 years at DEC responsible for the VMS file-system and check/repair
 - Almost 9 years as developer/manager in the SQL Storage Engine team through August 2007, ultimately responsible for Core Storage Engine
- **Instructor-led training: Immersion Events (US, UK, Australia)**
- **Online training: [pluralsight](http://pluralsight.com/) <http://pluralsight.com/>**
- **Consulting: health checks, hardware, performance, upgrades**
- **Remote DBA: system monitoring and troubleshooting**
- **Conferences: PASS Summit, SQLIntersection**
- **Become a SQLskills Insider: <http://www.sqlskills.com/insider>**



© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Materials

- **Send me email (paul@sqlskills.com) to get free 30-day access to all SQLskills online training classes**
 - No strings attached, no credit-card required
 - Subject something like “Intersection Pluralsight code”
- **Pre-con demo zip will be on SQLskills.com**
- **Zip file password will be “watchoutforsheep”**
- **Email me if you have problems finding it/opening it from Thursday onward**



3

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Why Is This Material Important?

- **Disasters happen, many times per day, all around the world**
 - Check out the SQL Server Central data corruption forums
- **Many people are not checking for problems**
 - Either they don't know how to check for corruption or they miss the warning signs that corruption has occurred
 - The faster you realize you have corruption, the more likely you will be able to recover with minimal downtime and data loss
- **Many people don't know what to do when a disaster occurs:**
 - More data loss and downtime than necessary
 - Monetary and even job losses
 - Overall lowered perception of SQL Server
 - Makes it harder to convince management that SQL Server is Enterprise-capable



4

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

What Can Happen to an Unprepared DBA Confronted by a Disaster?



- Image from http://commons.wikimedia.org/wiki/File:Panic_button.jpg

Things That Many People Try First

- **Restart SQL Server**
 - Just wastes time and delays getting back online
 - Especially if the physical server is power-cycled too
 - Although I've seen it work rarely when the I/O subsystem is doing stale reads
- **Immediately jump to a "last resort" and cause data loss without working through options**
 - E.g. running repair instead of using backups
 - E.g. removing the transaction log
- **Forcibly detach/drop a damaged or SUSPECT database**
 - It will fail to attach again so now the situation is even worse!

Root Cause Analysis

- **No matter what method you use to recover from corruption, you should always determine why it happened to avoid future problems**
- **It may be obvious what happened**
 - E.g. a known SAN problem or power failure
- **You may have no idea what happened, so what to do?**
 - Google/Bing for the corruption message you saw
 - Run I/O subsystem and server memory diagnostics
 - Examine the SQL Server error log and Windows event logs for clues
 - Check that firmware is up-to-date
 - Investigate NTFS filter drivers
 - Possibly contact Microsoft Customer Support for assistance



7

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Agenda

- **This material applies to all versions of SQL Server from SQL Server 2005 onward, except where noted**
- **I'm assuming basic knowledge of SQL Server terminology**
 - E.g. database, transaction log, backup
- **Disaster recovery strategy**
- **Backup strategy and optimizing restores**
- **Consistency checking**
- **Transaction log problems**
- **Restore techniques**
- **Repair techniques**
- **Summary**



8

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Disaster Recovery Planning

- **Disaster recovery means the steps taken after a disaster occurs**
- **High availability means putting technology in place to avoid disaster**
- **Designing a DR strategy is integral to designing a highly-available system**
- **Even with the most sophisticated redundancy, recovery from total loss of all data centers can only be done using backups**
 - Even if redundancy is available when disaster strikes, failover may not be automatic or the first resort
- **The requirements gathering process feeds into the backup/restore strategy too**
- **What restores you need to be able to do depends on:**
 - What needs to be brought online first
 - Data loss SLA
 - Downtime SLA



9

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Downtime SLA

- **Maximum allowable downtime or RTO**
 - Recovery Time Objective
- **Commonly discussed in terms of 'number of nines'**
 - 5-nines = 99.999% uptime, slightly over 5 minutes downtime per year
 - 4-nines = 99.99% uptime, almost 52.5 minutes downtime per year
 - 3-nines = 99.9% uptime, almost 8.75 hours downtime per year
- **Must consider how downtime is defined for you**
 - Is it 24x7 or, maybe just 9am-5pm weekdays
- **Achieving 5-nines of 24x7 operation is very hard**
- **Management may ask for zero downtime!**
 - Not technically possible
- **You have to know how technologies introduce downtime**
 - E.g. failover time with database mirroring



10

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Data-Loss SLA

- **Maximum allowable data loss or RPO**
 - Recovery Point Objective
- **Must consider how data loss is defined for you**
 - Number of transactions
 - Work done in a period of time
- **May be different for different tables or databases**
- **Zero data loss is much more easily achievable than 5-nines uptime**
- **Press the business owners for proper answer**
- **Management may ask for zero data loss too!**
 - This IS achievable, but is it absolutely necessary?



11

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Defining SLAs

- **Although management will want zero downtime and zero data loss, this is usually unreasonable**
- **All interested parties need to be involved in defining SLAs otherwise agreement may not be reached**
- **SLAs may need to be revisited and tweaked after:**
 - The design limitations have been identified
 - The design is complete
 - The implementation and testing phases are complete
- **SLAs should also be periodically reassessed to account for:**
 - Changing business requirements
 - Increases/decreases in workload
 - Changes in application behavior



12

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Assess the Risk of Unavailability

- **Assess your potential risks for downtime**
 - Site is unavailable
 - System is unavailable
 - Database is completely offline/unavailable
 - Database is partially offline/unavailable
 - Table is unavailable
 - Data/row(s) are unavailable
- **What is going to be the greatest impact on the business?**
- **The strategy should cope with the highest impact disasters first**
 - Site loss might not be the highest impact if critical data is available in another location, but failover isn't feasible



13

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

A Good Disaster Recovery Plan...

- **Should be written by the most senior staff members who have seen the most failures**
- **Should be tested by the most junior staff members who may be on duty late at night**
 - It won't be the most senior people on call on a holiday...
- **Should be comprehensive**
 - 'Restore the database from backups' isn't good enough
 - What if something goes wrong?
- **Should consider human factors in a widespread disaster**
- **Should be tested regularly and updated**



14

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

More DR Planning Considerations

- **Consider glitches at each step:**
 - What if the server is physically damaged?
 - Performing a bare-metal install of back-end and mid-tier servers...
 - What if the SAN is physically damaged?
 - How many USB drives can you find?
 - What if there's no power at the data centre?
 - Home Depot sells generators...
 - What if there's no data centre?
 - Where are the off-site backups stored?
 - What if the backups are corrupt?
 - What if no people are available?
- **What insurance do you have to buy new equipment?**
- **How do people work with no data center? Phones?**



15

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

DR Planning Survey

Does your disaster recovery strategy account for human nature during a disaster?

No. The plan assumes everyone will be pulling for the company.		39%	50
Kind of. The plan lists backup personnel contact info.		24%	30
Yes. We assume it might be chaos but the plan doesn't list contingencies.		17%	22
Yes. The plan describes different natural disasters and how they may affect the DR.		11%	14
Yes. The plan is very comprehensive, even considering terrorism and disease epidemics.		6%	8
Other? Enter here...		2%	3
Total: 127 responses			

- Source: my blog post at <http://bit.ly/RbcFP6>



16

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

DR Planning: People Issues

- Who gets notified first of failures?
- Who is responsible (i.e. in charge of decision process) at each phase of recovery?
- Who's the 'sponsor' that can resolve disputes around progress? (CIO?)
- Who needs to be kept informed? (e.g. CIO, customers, users)
- Who has to authorize a failover?
- How do things get escalated if progress is not being made?
- Who's in overall command of the DR effort?
- Contact info for everyone who may become involved
- Which other teams need to be involved for success?
 - Server team, SAN team, network team, 3rd-party companies
- How to test the application is working after DR is performed



17

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

More DR Run Book/Handbook...

- Apart from the issues on the previous slides, more things to include in your run book/handbook:
 - Where are the various scripts online? Offline?
 - Where is the install media online? Offline?
 - Who has the account info to retrieve media from Microsoft? Product Keys?
 - How long will each step take?
 - What error/completion messages are expected?
 - What are alternative steps?



18

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Increase Protection Using RAID

- **RAID**
 - Redundant Array of Independent/Inexpensive Disks
- **Various configurations give degrees of redundancy**
 - RAID-0 (striping)
 - RAID-5 (rotating parity)
 - RAID-1 (mirroring)
 - RAID-10 (striping over mirrors)
 - RAID-50 (striping over RAID-5)
- **RAID levels:** <http://en.wikipedia.org/wiki/RAID>
- **This applies to SSDs (Solid State Drives) as well**
 - A single SSD card/drive is RAID-0
 - Even an SSD which presents two “drives” is still a single point of failure
 - Use multiple SSDs in at least RAID-1 configuration



19

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Agenda

- Disaster recovery strategy
- **Backup strategy and optimizing restores**
- Consistency checking
- Transaction log problems
- Restore techniques
- Repair techniques
- Summary



20

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Recovering Using Backups

- **Although this is the best way to avoid data loss, it's not necessarily the best way to avoid downtime**
 - Depends what kind of backups are available
 - For example, with a backup strategy of full backup every Sunday and half-hourly transaction log backups:
 - If recovery is necessary towards the end of the week, lots of log backups need to be restored, which will take a long time
 - Better to have daily full or differential backups to allow faster restore
- **Remember:**
 - Backups have to exist to be useful
 - Backups have to be valid to avoid data loss
- **How often do you validate your backups?**
 - Perform a full test restore on another server at least once a week



21

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Backup Types (1)

- See my blog post at <http://bit.ly/15SCN5e> for more details on backup types and recovery models
- **Full database backup**
 - The starting point for any restore sequence is usually the most recent full database backup
 - This provides a single point in time to restore to, or the basis for further restore operations
- **Transaction log backup**
 - Commonly known as just a “log backup”
 - The set of transaction log records generated since the last log backup completed, using the full or bulk_logged recovery models
 - Must be used to roll forward a database to the desired point in time



22

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Backup Types (2)

- **Differential database backup**
 - The set of data file pages that have changed since the most recent full database backup preceding this backup
 - A differential backup is the net effect of all transaction log backups between the most recent full backup and this differential backup
 - Where possible, the restore sequence will start with restoring a full backup and then restore the most recent differential backup, before finishing with transaction log backups
 - This makes the restore sequence shorter and faster
- **There are also full file, full filegroup, differential file , and differential filegroup backups**
 - Their use is more advanced and we will not consider them here
 - If they exist in your environment, it is imperative that you practice restore sequences with them



23

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

How to Plan a Backup Strategy?

Don't Do It.



24

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Plan A Restore Strategy

- **Never plan a backup strategy**
- **Always plan the restore operations you need to be able to do, and that will dictate what backups you need and what recovery model to use**
- **This usually turns out to be FULL recovery model, with a full + log + differential backup strategy for most databases...**
- **VLDBs may look at more granular backups including:**
 - Filegroup backups, filegroup differentials and log backups
 - File backups, file differentials and log backups
- **Planning your backups without considering restore can lead to not meeting your SLAs (more downtime) and possibly data loss**

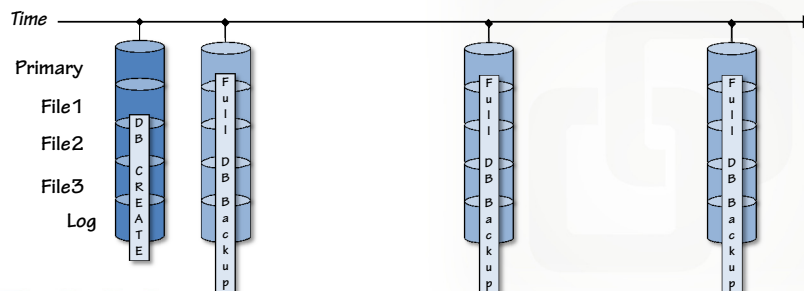


25

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Full Database Backup Only Strategy

- **Full backups only provide a single point in time to which the database can be restored**
- **All work since the last full backup is lost**

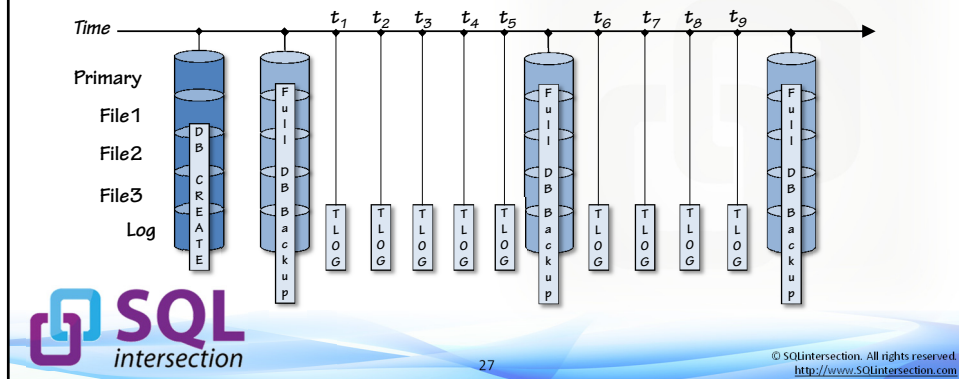


26

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

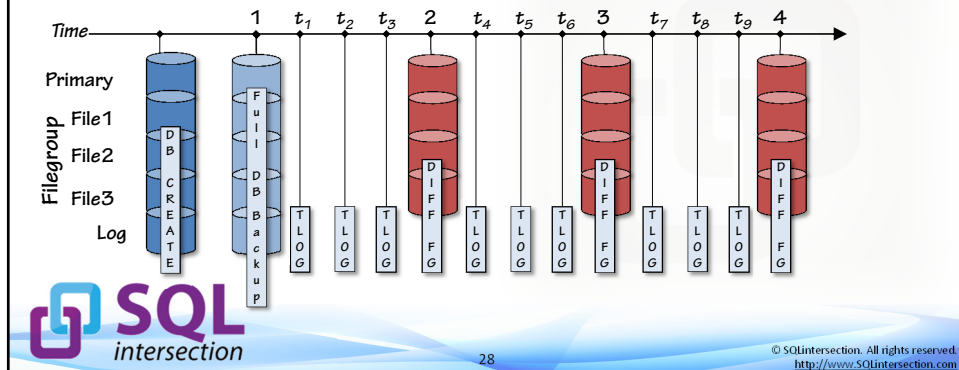
Full Database Backup Plus Log Backups Strategy

- Up-to-point-of-crash recovery is possible with no data loss
- Possible to work around a damaged full backup if prior log backups are still available
- Restoring a large number of log backups can take a significant amount of time so may need to include differential backups

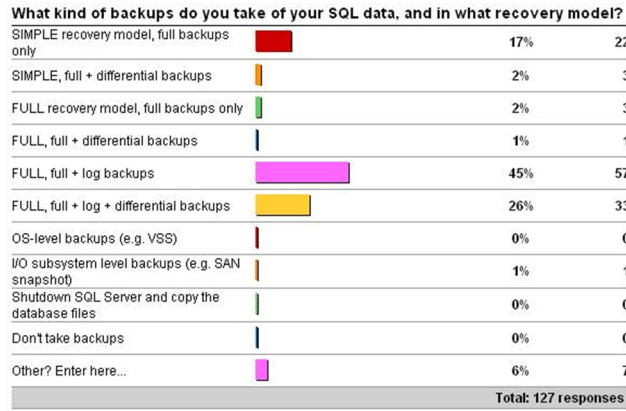


Full, Diff Database, and Log Backup Strategy

- Differential database backups are only the extents changed since the last full database backup
- Simplifies recovery time by allowing you to recover the database, the last differential, and only the logs after that, until the time of the disaster



Backup Strategy Survey



- Source: my blog post at <http://bit.ly/nd1Cv>



29

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Backup Checksums

- Use **WITH CHECKSUM** when performing a backup
 - Validates all page checksums in the database as the pages are read
 - Calculates a checksum over the entire backup stream and records it in the backup header
 - Negligible performance impact
 - Gives peace of mind that no I/O errors are present in the database
 - I recommend you always do this
- Allows the integrity of the backup to be checked using **RESTORE VERIFYONLY ... WITH CHECKSUM**
 - Gives peace of mind that no I/O errors are present in the backup
 - Backups are susceptible to I/O subsystem corruption just like data files
- Allows page checksums to be checked during restore operations
- NOT a substitute for consistency checking**



30

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Making Restores Faster

- **Use backup compression**
- **Use striped backups**
- **Enable instant file initialization**
- **Make use of filegroups**
 - Allows partial database availability in Enterprise Edition
 - Allows piecemeal restore of only the damaged file/filegroup
- **This all only works if you've designed your database to be able to take advantage of it**
 - A single file/filegroup database cannot be partially available
 - Use "manual" partitioning
 - Use the table/index partitioning feature of 2005
- **With a single file filegroup, you have no option but to restore it entirely if it is badly damaged/destroyed**

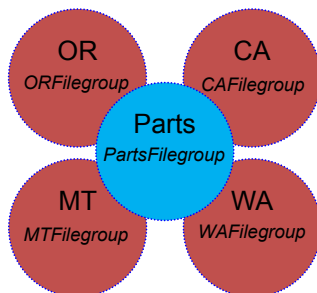


31

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Manual Partitioning (1)

- **Multiple ways to split up the data**
 - E.g. regionally based with some shared components. Each filegroup contains a separate table



Region-specific data:

- Customers, Sales, etc. are found within the region-specific filegroup
- If CAFilegroup is damaged, customers in Oregon, Washington and Montana are not affected...
- However, damage to "Parts" would mean downtime
- What about a full restore?



32

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

Manual Partitioning (2)

- Geographically split up



- Which time zone / region / state to bring online first?



33

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Agenda

- Disaster recovery strategy
- Backup strategy and optimizing restores
- Consistency checking
- Transaction log problems
- Restore techniques
- Repair techniques
- Summary



34

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Page Protection Options

- SQL Server allows pages to be “protected” on disk
- This allows fast detection of corruption when a page is read into memory
- Set using `ALTER DATABASE SET PAGE_VERIFY <option>`
- Three options:
 - NONE (don't do this...)
 - TORN_PAGE_DETECTION
 - CHECKSUM
- All page protection operations are performed by the buffer pool (also known as the buffer manager or buffer cache)



35

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

I/O Errors

- Three types
 - 823: a hard I/O error, severity 24
 - 824: a soft I/O error, severity 24
 - 825: a read-retry error, severity 10
- 823 and 824 are:
 - Logged in the `msdb.dbo.suspect_pages` table
 - Written to the SQL Server error log and Windows Application event log
- Much better to use an automated monitoring process
 - SQL Agent alerts, Microsoft SCOM, 3rd-party monitoring tool
- Create alerts for at least severities 19 through 24 and error 825
- Glenn Berry has a comprehensive blog post and detailed Transact-SQL script for creating SQL Agent alerts at <http://bit.ly/1d86rcv>



36

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

DBCC CHECKDB

- **This is the main consistency checking command**
- **Runs all consistency checks against a single database**
- **Very simple to execute:**
 - DBCC CHECKDB (N'database')
 - Or change context to the database and just do DBCC CHECKDB
- **The only way to read all allocated pages in the database**
 - Use to force page checksums to be checked
- **Very resource intensive with CPU, memory, and I/O**
- **Will use multiple threads on Enterprise Edition**
 - Parallelism can be disabled using documented trace flag 2528
- **My comprehensive blog post series is at <http://bit.ly/TFvxmG>**
- **Detailed information (~90 pages) in the SQL Server 2008 Internals and SQL Server 2012 Internals books**



37

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

How To Run Consistency Checks?

- **Three options for running consistency checks:**
 - Manually
 - Automated script, using SQL Agent
 - Automated using a SQL Server Maintenance Plan
- **Easy-to-use set of maintenance scripts available from Ola Hallengren at <http://ola.hallengren.com/>**
- **I like to recommend running consistency checks at least once a week**



38

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Which Databases Should Be Checked?

- **All databases should be consistency checked**
 - Include read-only databases
- **Definitely include system databases**
 - master, as it contains information about databases, users, logins
 - Corruptions in it can cause an instance to fail to start
 - mssqlsystemresource is checked automatically when master is checked
 - msdb, as it contains SQL Agent jobs, schedules, backup history, and more
 - model, as its content is copied to create new databases
 - tempdb, as corruptions in it can cause an instance to shut down
 - Note that if you use the Maintenance Plan Wizard to create a job to check integrity, tempdb is excluded
 - distribution, to avoid replication problems



39

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Using Another Server (1)

- **This method removes the consistency checking workload from the production server**
- **Methodology:**
 - Perform full database backup on the production server
 - Restore database on another server
 - Run consistency checks on the restored database
- **What to do if a corruption is found?**
 - Is it the I/O subsystem on this server?
 - Is it the backup file?
 - Is it the database on the production server?
- **If corruption is found, which should be rare, it forces you to run consistency checks on the production server**



40

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Using Another Server (2)

- Some people advocate consistency checking a mirror database or an availability group secondary replica as a way of consistency checking the principal database or primary replica
- This is not valid
- The various servers are stored on different I/O subsystems and corruptions do not propagate between servers
 - E.g. ensuring that a mirror database has no corruption does not imply anything about the state of the principal database
- Consistency checking should ideally be performed on all copies of a database
 - Especially true for availability groups if a secondary replica is being used to offload full backups from the primary replica



41

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Consistency Checking Survey (1)

Run DBCC CHECKDB with no options on the production database		62%	261
Run DBCC CHECKDB WITH PHYSICAL_ONLY on the production database		10%	42
Run DBCC CHECKDB on a restored backup on another server		11%	47
Run DBCC CHECKDB on a database snapshot on a mirror database		1%	4
Spread consistency checking over several days using DBCC CHECKTABLE		0%	1
Spread consistency checking over several days using DBCC CHECKFILEGROUP		1%	4
Use BACKUP WITH CHECKSUM to validate page checksums, no DBCCs		1%	4
Run DBCC CHECKDB on a log shipping secondary, or 2012 AG secondary		1%	3
Run DBCC CHECKDB on a SAN copy/mirror		2%	7
Create your own database snapshot of the production database and run DBCC CHECKDB on it		0%	2
Other? Enter here...		11%	47
Total: 422 responses			

- Source: my blog post at <http://bit.ly/Hq3m77>



42

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Consistency Checking Survey (2)

How often do you run consistency checks? (regardless of "how" you run them)			
What are consistency checks?		9%	25
Never		5%	14
Only when corruption is detected some other way		8%	22
Only during an event like an upgrade or migration		1%	2
Regularly, but less than monthly		3%	8
Monthly		5%	14
Weekly		37%	103
Daily		25%	69
More frequently than daily		1%	3
Only after performing a restore, or after a failover		1%	3
Other? Enter here...		5%	13
Total: 276 responses			

- Source: my blog post at <http://bit.ly/1mUuRw>



43

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Making DBCC CHECKDB Go Faster (1)

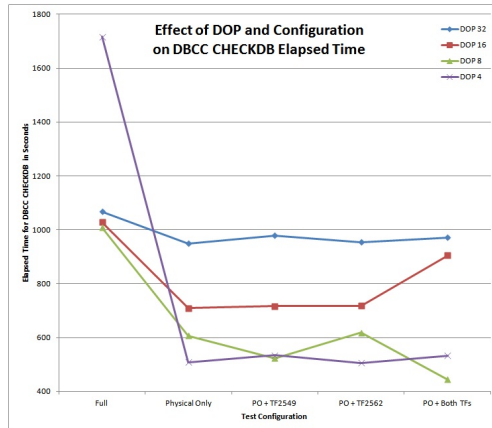
- Recent work has been done to improve the performance of DBCC CHECKDB by:
 - Allowing it to check all tables as a single batch
 - Slightly changing the I/O pattern it generates
 - Reducing contention on the DBCC_MULTIOBJECT_SCANNER latch
- Ported to SQL Server 2008, 2008 R2, 2012
 - Uses documented trace flags 2549 and 2562
 - Check the KB article to see which trace flags are required in which versions
- Be aware these changes may push the I/O subsystem much harder, and you may not see any changes at all
- Bob Ward's blog post is at <http://bit.ly/yAFY7W>
- KB article describing this is at <http://support.microsoft.com/kb/2634571>



44

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Making DBCC CHECKDB Go Faster (2)



- Consider reducing DOP
- to lower run time when
- using the option WITH
- PHYSICAL_ONLY
- On systems with high numbers of physical cores (32+), consider reducing DOP to 4 or 8 to lower run time
- Resource Governor can
- help with this

• Source: my blog post at <http://bit.ly/10rgH70> (those are zeroes, not Os)



45

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Making DBCC CHECKDB Go Faster (3)

- Consider reducing the memory available to DBCC CHECKDB
- The Query Processor by default will grant DBCC CHECKDB a very large query execution memory grant, potentially causing buffer pool shrinkage
 - Requested grant is many terabytes because of unknown cardinality
 - On a system with 100 GB of memory, grant may be as high as 10 GB!
- Performance testing by Jonathan Kehayias has shown that limiting the amount of memory for DBCC CHECKDB to 1 GB can lead to 5-10% performance increase, and very limited buffer pool pressure
- Use Resource Governor to do this
- Anecdotal evidence from people with very large buffer pools is that can lead to a huge performance gain
- Jonathan's blog post at <http://bit.ly/YBb7iH>



46

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Making DBCC CHECKDB Go Faster (4)

- **Indexes on computed columns drastically slow down DBCC CHECKDB**
 - Also applies to DBCC CHECKTABLE and DBCC CHECKFILEGROUP
- **I discovered this problem during benchmarking tests for the degree-of-parallelism data on the previous slide**
- **Any time a table has an index on a computed column, all threads bottleneck on the DBCC_OBJECT_METADATA latch**
 - This controls access to an Expression Evaluator construct in the Query Processor that evaluates computed column expressions
 - The latch can only be acquired in EX mode, hence the bottleneck
- **This can lead to 20x performance decrease**
- **Solution is to disable such indexes while DBCC CHECKDB is running**
- **My blog post on this is at <http://bit.ly/YAzY6h>**



47

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Running DBCC CHECKDB

- **Use the following command with only these options:**
 - DBCC CHECKDB (yourdb) WITH NO_INFOMSGS
 - Add ALL_ERRORMSGs when using SQL Server 2008 RTM or before
- **Good idea to know how long it usually takes to run for a database**
 - Allows you to report to management how long before results are known when a disaster occurs
 - Longer run time usually indicates some corruption has been found
- **Wait for the command to complete**
 - It's the only way to know what corruptions you have
 - Don't jump the gun and assume you need to restore
- **Look through the output for anything significant**



48

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Interpreting DBCC CHECKDB Output (1)

- **There are over 100 errors that DBCC CHECKDB can output, some with over 200 message states**
 - Effectively there are roughly a thousand different corruption conditions that can be reported
- **Figuring out what one corruption means isn't too bad**
 - MSDN has some of them published
- **Figuring out multiple corruptions can become very hard and usually isn't worth the time**
- **There are some tips and tricks you can use to determine the course of action to take**



49

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Interpreting DBCC CHECKDB Output (2)

- **Did DBCC CHECKDB fail?**
 - If it stops before completing successfully, something bad has happened that is preventing it from running correctly
 - This means there is no choice but to restore from a backup or try exporting the data, as DBCC CHECKDB cannot be forced to run (and hence repair)
- **Examples of fatal (to DBCC CHECKDB) errors:**
 - 7984 – 7988: corruption in critical system tables
 - 8967: invalid states within DBCC CHECKDB itself
 - 8930: corrupt metadata such that DBCC CHECKDB could not run
- **The SQL Server error log message will list an error state**
 - See the “Understanding DBCC Error Messages” portion of the Books Online for DBCC CHECKDB for details at <http://bit.ly/179p6At>



50

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Interpreting DBCC CHECKDB Output (3)

- **Are the corruptions only in nonclustered indexes?**
 - If recommended repair level is REPAIR_REBUILD, then yes
 - Otherwise, check all the index IDs in the errors and if they're all greater than 1, then yes
 - If yes, you don't need to restore or run repair, as we'll see later
- **Was there an un-repairable error found?**
 - Examples:
 - 2570 error: invalid data for the column type (data purity error)
 - 8992 error: CHECKCATALOG (metadata mismatch) error
 - 8909, 8938, 8939 (page header corruption) errors where type is 'PFS'
 - None of these can be automatically repaired so your options are to restore or to attempt manual repairs



51

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Interpreting DBCC CHECKDB Output (4)

- **Everything else I classify as "general corruptions"**
 - Your options are to restore, repair, or export to a new database
- **The more corruptions there are, the harder it is to figure out what's actually broken**
 - It also depends on what kind of page is corrupt
- **For example, for a data page in the leaf level of a clustered index:**
 - There may just be one error about one row on the page
 - And may also be a matching error about each nonclustered index
 - And may also be a matching error about each off-row LOB value in the row
 - Etc...
 - There may be an error that the entire page cannot be processed
 - And will also be errors about broken links to other pages
 - And may also be errors about nonclustered indexes
 - Etc...



52

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Agenda

- Disaster recovery strategy
- Backup strategy and optimizing restores
- Consistency checking
- **Transaction log problems**
- Restore techniques
- Repair techniques
- Summary



53

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Damaged/Missing Log With Clean Shutdown

- The transaction log is essential for crash recovery to work
- However, a damaged or missing transaction log does not matter if the database had a clean shutdown
 - The transaction log is not required as there is no crash recovery to run
- In that case, a new transaction log can be created
 - Requires re-attaching the database using CREATE DATABASE and using either the FOR ATTACH or FOR ATTACH_REBUILD_LOG options
 - FOR ATTACH creates a single 0.5 MB transaction log file, but only works if there was previously one transaction log file
 - FOR ATTACH_REBUILD_LOG also creates a single 0.5 MB transaction log file, but works no matter how many log files there were previously
- Make sure to provision the correct size and auto-growth afterwards



54

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

EMERGENCY Mode

- **If the log is missing or damaged, and the database was not cleanly shut down, the database state will be RECOVERY_PENDING or SUSPECT**
 - Recovery couldn't start, or recovery couldn't finish, respectively
- **EMERGENCY mode is also known as "bypass recovery" mode, as this is exactly what it instructs the Storage Engine to do**
 - ALTER DATABASE mydb SET EMERGENCY
- **Although this allows access to the database, the data is transactionally inconsistent**
 - Furthermore, the database itself may be structurally inconsistent
- **You can try to export data out into a new database**
- **Without backups, the only way to recover the existing database is to try EMERGENCY-mode repair**



55

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

EMERGENCY-Mode Repair

- **This was added to SQL Server 2005 as a documented and supported way to attempt recovery with a damaged transaction log**
 - It used to be possible using DBCC REBUILD_LOG in SQL Server 2000
- **The database must be set to EMERGENCY and SINGLE_USER**
- **Running DBCC CHECKDB (mydb, REPAIR_ALLOW_DATA_LOSS) does the following in EMERGENCY mode:**
 - Run as much crash recovery as possible, skipping damaged log records
 - Build a new transaction log file
 - Run a full DBCC CHECKDB with REPAIR_ALLOW_DATA_LOSS
 - Try to bring the database online
- **This is not guaranteed to work in all scenarios**
 - Severe data file damage may prevent the database coming online
 - File system damage may prevent the log file being recreated



56

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Detached SUSPECT Database

- When attaching a database, crash recovery must be completed before the database will attach
- If a detached database requires crash recovery and the transaction log is damaged or missing, the attach will fail
- Prior to SQL Server 2008 it was possible to detach a SUSPECT database
- Reattaching a database in this state requires the following:
 - Create a database with the same name and file IDs as the detached database
 - Set the dummy database offline and delete the log and data files
 - Copy in the files from the detached database
 - Try to bring the database online again
- Once the database is attached, you can proceed to EMERGENCY mode



57

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Agenda

- Disaster recovery strategy
- Backup strategy and optimizing restores
- Consistency checking
- Transaction log problems
- Restore techniques
- Repair techniques
- Summary



58

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

How to Restore?

- **In an emergency, always go to the operations guide/DR handbook/run book (do you have one?)**
- **Operations guide should define:**
 - Location of backups
 - How to put together the restore sequence, depending on what is damaged
 - Commands or scripts to use
 - Amount of time expected per step during restore process
- **There may be a 3rd-party backup/restore tool that you must use**
 - We're just talking about SQL Server native backups in this module
- **Make sure to use the correct RESTORE options**
 - We'll discuss these later



59

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Restore Options

- **WITH RECOVERY is the default**
 - Does not allow further restore options as database is fully recovered
 - Frustrating that it is the default, as it causes problems
- **Use WITH NORECOVERY for safety and make it your default for all restore operations in the restore sequence**
 - Postpones fully recovering the database
 - Then use RESTORE DATABASE dbname WITH RECOVERY to complete the restore sequence
 - If you make a mistake, you must restart the restore sequence
- **WITH STANDBY is the same as WITH NORECOVERY but creates a recovered, read-only database for you to look in**
 - Creates an UNDO file to save info from transactions that are pending at the end of the restore



60

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Restoring an Entire Database to a Point in Time

- **Common examples:**
 - Restore a database to the time just before a user or application error occurred (e.g. dropping a table)
 - Restore a database to the point before a known corruption-causing event occurred
- **Start with most recent full backup and proceed with smallest set of backups to the desired point**
- **Best practice is to use WITH STOPAT on all operations in the restore sequence**
 - Date and time specific, or
 - Transaction name (i.e. marked transaction)
- **Note: All work after the stop point is lost**
- **You MUST have an unbroken sequence of backups to get there!**



61

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Tail-Of-The-Log Backups

- **After a disaster, where it is necessary to restore from backups, the final portion of the transaction log must first be backed up to allow restoration with zero data loss**
- **This backup is called a tail-of-the-log (or tail-log) backup**
- **I recommend performing a tail-of-the-log backup manually rather than through SSMS**
- **If the data files are inaccessible you can only perform a tail-of-the-log backup using the WITH NO_TRUNCATE option**
- **If there was a minimally-logged operation since the last log backup, and the data files are inaccessible:**
 - Until SQL Server 2008 R2, the tail-of-the-log backup will fail
 - From SQL Server 2008 R2 onwards, the tail-of-the-log backup will succeed, but restoring it will result in a corrupt database



62

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

What if SQL Server is Unavailable?

- **You may need a tail-of-the-log backup during disaster recovery**
- **If the SQL Server instance is unavailable, you'll have to perform a "hack-attach" of the log file to another instance to perform the backup**
 - Similar to hack-attaching a detached SUSPECT database
- **Steps to perform:**
 - Create dummy database with the same name
 - Set the dummy database offline
 - Delete all data and log files
 - Drop in the salvaged log file you want to back up
 - Try to bring the database online
 - Perform the tail-of-the-log backup
- **Note: this only works on the same version of SQL Server on which the original database existed**



63

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Determining the Restore Sequence

- **In a complex restore sequence, there may be differential and log backups that cover the same time interval**
- **No point restoring unneeded log backups, but also must make sure that the log backup covers the right time period (i.e. set of LSNs in the log)**
- **Finding the minimum required LSN or min-LSN can be done using:**
 - Backup history tables in msdb
 - Make sure you're backing up msdb (and other system databases too)
 - Output from RESTORE HEADERONLY
 - Best to have scripts written and tested in advance of having to use them
- **Restore sequence can also be found using the SSMS Database Recovery Advisor**
- **More information in Books Online at <http://bit.ly/19RnAqn>**



64

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Page Restore

- **One or more pages can be restored**
 - Incredibly valuable if the database is not architected for partial database availability and you don't want to restore the entire database
- **Backups must exist to allow the page to be restored to the same point in time as the PRIMARY filegroup**
- **The following page types cannot be page restored:**
 - Boot page
 - Fileheader pages
 - Allocation bitmaps (not including IAM pages)
 - Certain pages in critical system tables



65

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Restoring Part of a Database to a Point in Time

- **Uses partial database availability**
- **Example: you want to restore the OLTP portion of a deleted database before the rest of it**
 - Restore a subset of filegroups to create a new, smaller database
 - First restore must use the PARTIAL option and be the PRIMARY filegroup
 - Then roll forward to desired point in time
 - Restore other filegroups later, and online in Enterprise Edition
- **Other use cases:**
 - Restoring a portion of a VLDB on a test or development system
 - Restoring a portion of a VLDB from an older backup for comparison



66

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Restoring Corrupt Backups

- You may encounter a backup that is corrupt, but it's the only backup
- It can be restored using **WITH CONTINUE_AFTER_ERROR**
- Be careful about doing this with corrupt log backups
 - They will create a corrupt database
- If a backup in the middle of a restore sequence is corrupt, it may be better to end the restore sequence with the previous backup
 - No corruption to deal with, but further back in time
- Beware of tail-of-the-log backups that contain minimally-logged operations where the data file was not present
 - Possible with SQL Server 2008 R2 onward
 - Guaranteed to corrupt the database during restore



67

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Looking Into Log Backups

- Two undocumented TVFs exist to allow transaction log analysis
- **fn_dblog** for looking into the log
 - `select * from fn_dblog (startLSN, endLSN)`
- **fn_dump_dblog** for looking into log backups
 - `select * from fn_dump_dblog (startLSN, endLSN, 'DISK' | 'TAPE', devicenum, backuppath, DEFAULT, DEFAULT,...)`
- When trying to find the exact time to restore to, **fn_dump_dblog** can vastly reduce the amount of time required
 - E.g. finding the point at which a table was dropped
- Can also be used for off-production analysis of activity
- **Beware: fn_dump_dblog leaks a thread and scheduler per execution**
 - Use on a test server, then use the Log Sequence Number (LSN) to do the restore in production



68

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Restoring to a Log Sequence Number

- **Once the desired LSN has been found, you usually want to restore everything up to, but not including, the transaction that begins there**
 - E.g. a transaction that does DROP TABLE or a large DELETE operation
- **Use RESTORE ... WITH STOPBEFOREMARK='lsn:<lsnstring>'**
- **The hexadecimal LSN you get from the fn_dblog or fn_dump_dblog output needs to be translated into the form required by RESTORE**
 - The formula and code to do the translation are documented on my blog at <http://bit.ly/1OJT5b0> (those are zeros, not Os)



69

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Restoring System Databases (1)

- **The backup must be of the same SP level as the SQL Server instance**
 - E.g. an RTM backup of msdb will not restore on SP1
- **model**
 - Restore in same way as user databases
- **msdb**
 - Restore in same way as user databases
 - Stop the SQL Server Agent before restoring
 - If using log backups, make sure to restore them as well
- **mssqlsystemresource**
 - Not a real database and cannot be backed up or restored
 - Use file-system copy but it must be the correct version
- **tempdb**
 - Cannot be backed up or restored



70

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Restoring System Databases (2)

- **Everyone should practice restoring master to see how easy it can be**
- **Restoring master**
 - Start SQL Server in single-user mode with the –m startup parameter
 - Restore the latest master backup, using WITH REPLACE
 - Server shuts down automatically, so remove –m and restart
 - For any changes after the master backup was made:
 - Reattach any new databases and recreate any new users/logins and server-level objects such as linked servers and server role memberships
- **Rebuilding master**
 - Necessary if no backup exists or the instance will not start
 - All data in the master database is lost so restore master if possible using steps above
 - If no restore is possible:
 - Reattach all databases and recreate all users/logins and server-level objects



71

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Agenda

- **Disaster recovery strategy**
- **Backup strategy and optimizing restores**
- **Consistency checking**
- **Transaction log problems**
- **Restore techniques**
- **Repair techniques**
- **Summary**



72

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Running Repair

- **Not always a last resort compared to restore**
 - Depends what backups are available
 - Depends if downtime is more important than data loss
- **All repair options require SINGLE_USER mode, i.e. they're offline**
- **Specifying a repair option makes DBCC CHECKDB run single-threaded**
- **Try to repair smallest thing**
 - DBCC CHECKDB is largest, then
 - DBCC CHECKTABLE, then
 - DBCC CHECKALLOC
- **You can put a transaction around your repair operation and roll it back if you don't like what repair did**
- **Note: most system tables can't be repaired successfully**



73

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Beware of REPAIR_ALLOW_DATA_LOSS

- **It was very deliberately named**
- **It usually fixes structural inconsistencies by de-allocating**
 - This is the fastest and most provably correct way
- **It doesn't take into account:**
 - Foreign-key constraints or inherent data relationships
 - Replication
- **Before running repair, protect yourself**
 - Take a backup and quiesce replication topologies involved
- **After running repair, check the data**
 - Run DBCC CHECKDB again to make sure all corruptions were repaired
 - Run DBCC CHECKCONSTRAINTS if necessary
 - Reinitialize any replication topologies involved
- **If you had to run repair, update your backup/restore strategy!**



74

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Manually Fixing Nonclustered Indexes

- It doesn't make sense to put the database offline and run DBCC CHECKDB or DBCC CHECKTABLE with REPAIR_REBUILD to fix corrupt nonclustered indexes
- All it will do is essentially disable and rebuild the index, so why not do it yourself?
- You cannot just rebuild the index
 - Online rebuild reads the old index to build the new index
 - Offline rebuild does that from SQL Server 2008 onward
- Steps to use, inside a transaction:
 - ALTER INDEX name ON tablename DISABLE
 - ALTER INDEX name ON tablename REBUILD
- Using a transaction is necessary to prevent any index-enforced constraints from being violated while the index is disabled



75

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Manually Fixing Data Purity Errors

- If a 2570 data purity error was found, SQL Server cannot repair it
 - The error is that a column value is out-of-bounds
 - Which value should SQL Server pick to repair it?
- You must manually repair data purity errors
 - Work out which row is invalid using SELECT or DBCC PAGE
 - Update the column to a valid value
 - <http://support.microsoft.com/kb/923247> explains the options
- Make sure you choose a value that makes sense for your business logic and the data the column value represents



76

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Metadata Corruption

- **Someone manually changed system tables in SQL Server 2000...**
- **Hidden system tables accessed with Dedicated Admin Connection (DAC)**
 - sqlcmd -A or prefix the SSMS connection string with "admin:"
 - 2012: Error message in SSMS can be ignored
 - 2012: TF7806 at startup plus running SQL Browser allows DAC connections under all circumstances, including named instances
 - SQL Performance blog post on this at <http://bit.ly/RKxGOV>
- **These tables cannot be changed unless the server is booted with -m**
- **From SQL Server 2008 onward, if a system table is directly modified, it is noted in the database metadata**
 - This modification makes your database technically unsupported
 - Check DBCC DBINFO (requires trace flag 3604)
 - If you update a system catalog using 3-part naming, the dbi_updSysCatalog of the database you're in is erroneously updated



77

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Rebuilding System Table Indexes

- **Structurally corrupt system table clustered indexes cannot be repaired successfully**
- **Structurally corrupt system table nonclustered indexes usually sometimes can be rebuilt by repair**
 - May have to play around with DBCC CHECKTABLE to find which table is corrupted as DBCC CHECKDB may fail
 - Not guaranteed to work...be careful
- **I would try to avoid running repair on system tables completely**
 - Or at least test it on a copy of the database



78

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Using Older Backups and Repair

- **If you don't have valid, recent backups to recover with, you may be able to use older backups along with repair**
- **Steps to use:**
 - Perform the repair operation
 - Determine what data is missing or damaged
 - Restore the old backup as a different database name
 - Determine if any of the missing data can be salvaged from the old database
- **Of course, no guarantee that the salvaged data is up-to-date, but it may be better than nothing**

Reconstructing Data From Nonclustered Indexes

- **If repair is the only option and heap or clustered index structures are damaged, you may be able to recover data from nonclustered indexes**
 - This may also be the case if repair is impossible because of corrupt metadata
- **Create SELECT statements to figure out which table rows will be lost from running repair**
- **Create SELECT statements to force selection from non-damaged nonclustered indexes to salvage some of the column data**
- **Success rates will vary depending on the amount of column coverage in the nonclustered indexes**
 - May need to use DBCC PAGE to work around corrupt pages
- **Do this BEFORE running repair, as the repair will rebuild nonclustered indexes based on the repaired heap or clustered index**

The Ultimate Advanced Recovery

- **Manually editing data files using a hex editor or DBCC WRITEPAGE is the most advanced method of corruption recovery**
 - Usually requires very detailed knowledge of database structures
- **Always take a backup before manually editing a database**
- **I usually do this once or twice a year for clients**
 - Some examples:
 - Fixing up broken off-row LOB links in SQL Server 2000 sysindexes
 - Recreating a file header page by copying from an older backup
 - Deallocating a broken page to allow data export when other metadata problems prevent DBCC CHECKDB repair from running
- **Always run DBCC CHECKDB to make sure you haven't made things worse and have fixed the original problem**
- **Always do it on a copy of the corrupt database**
 - If a backup/restore won't work, use a file system copy and hack attach



81

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Any Other Ways to Salvage Data?

- **Possible other places to get data:**
 - Data warehouse
 - Test/dev systems
 - 3rd-parties that host/consume some of your data
 - Replication subscribers
 - Hard-copy printouts
- **I've heard some incredible stories of tenacity in the face of corruption**
 - "Unbelievable tale of disaster recovery" at <http://bit.ly/185Z2bl>



82

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Agenda

- Disaster recovery strategy
- Backup strategy and optimizing restores
- Consistency checking
- Transaction log problems
- Restore techniques
- Repair techniques
- Summary



83

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Customer Examples to Learn From

- All of the following examples happened while I was at Microsoft
- I was involved with each customer
- No names will be divulged
- In increasing magnitude of loss
- These are just the worst ones...
- Looked at from perspective of disaster recovery



84

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Examples to Learn From #1

- **Story**
 - Database is corrupt and takes literally *days* to bring online
 - Customer is a small bank in the US
- **Cause**
 - Customer only had a full backup from January 2004 plus log backups every ½ hour until April 2004 when corruption hit
- **Resolution**
 - Restored ALL the backups (approx 5000!) – luckily none were corrupt
- **Cost**
 - Aggravation to bank customers while bank was down for several days
- **DR lessons learned**
 - They were able to recover but not in the time they wanted
 - They were very lucky that their backups all worked



85

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Examples to Learn From #2

- **Story**
 - Database lost with corrupt backup
 - Customer is a large investment firm and the db has all 401k data for its customers
- **Cause**
 - Database was backed up prior to moving to new hardware
 - Old h/w flattened *before* database restored on new h/w
 - Backup corrupt
- **Resolution**
 - Force corrupt backup to restore and patch together broken data
- **Cost**
 - Job losses (from DBA to VP level)
- **DR lessons learned**
 - Always validate backups
 - Always have multiple copies of backups



86

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Examples to Learn From #3

- **Story**
 - Database and application down for 3 days after corruption
 - Database stores stock trades for UK customers of a large US bank
- **Cause**
 - Started DBCC CHECKDB but took too long so killed it
 - Started to restore from backups but found they were corrupt
 - Re-ran DBCC CHECKDB again and let it complete
- **Resolution**
 - It was a nonclustered index corruption, fixable with an index rebuild
- **Cost**
 - Customer fined >\$10m by UK government
- **DR lessons learned**
 - Don't panic when disaster strikes
 - Wait for DBCC CHECKDB to tell you what is wrong
 - Validate backups



87

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Examples to Learn From #4

- **Story**
 - DBA comes to work to find the database and all backups gone
 - Database stores all tax records for a US state
- **Cause**
 - 3rd party technician accidentally formatted the drive where the database *AND* backups (!!!) were stored while installing hardware
- **Resolution**
 - 200 staff members spent 3 weeks entering data for all state residents from scratch
- **Cost**
 - Unsubstantiated report of state losing more than \$2 *billion* in tax revenue
 - No job losses...
- **DR lessons learned**
 - Always have multiple backups, including off-site
 - Keep backups on separate media from databases



88

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Take Aways

- **Having a disaster recovery plan is critical**
- **Never plan a backup strategy, plan a restore strategy!**
- **Design your database to allow the smallest possible restores**
 - Remember backup compression, instant file initialization, partitioning, and so on...
- **Make sure your backups allow you to restore within the bounds of your SLAs**
- **Disaster recovery probably won't be successful unless you've practiced, and practiced, and practiced... ☺**



89

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Questions?



Don't forget to complete an online evaluation on EventBoard!

Session: PRECON04

Practical Disaster Recovery Techniques

Your evaluation helps organizers build better conferences
and helps speakers improve their sessions.



Thank you!