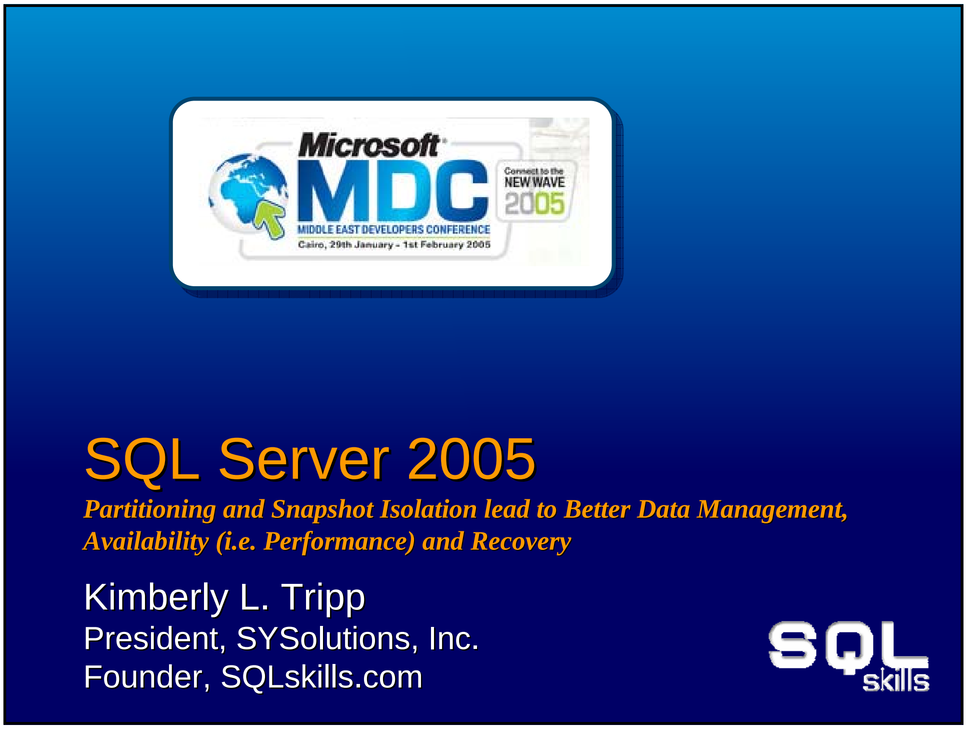
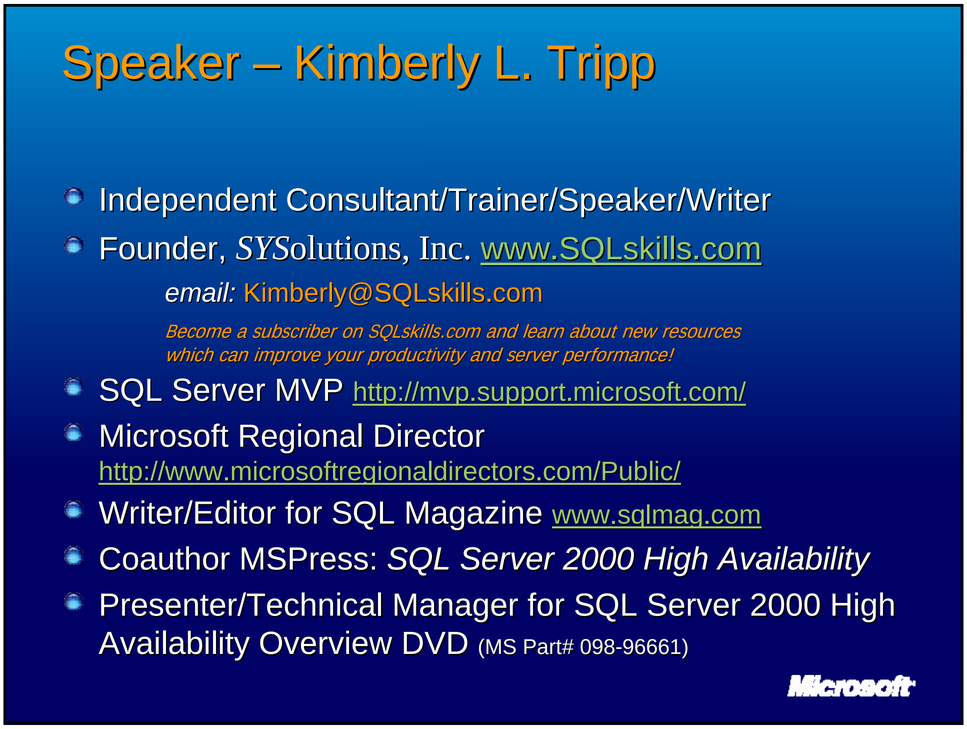

Microsoft MDC
MIDDLE EAST DEVELOPERS CONFERENCE
Cairo, 29th January - 1st February 2005
Connect to the NEW WAVE 2005
SQL skills

SQL Server 2005

Partitioning and Snapshot Isolation lead to Better Data Management, Availability (i.e. Performance) and Recovery

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

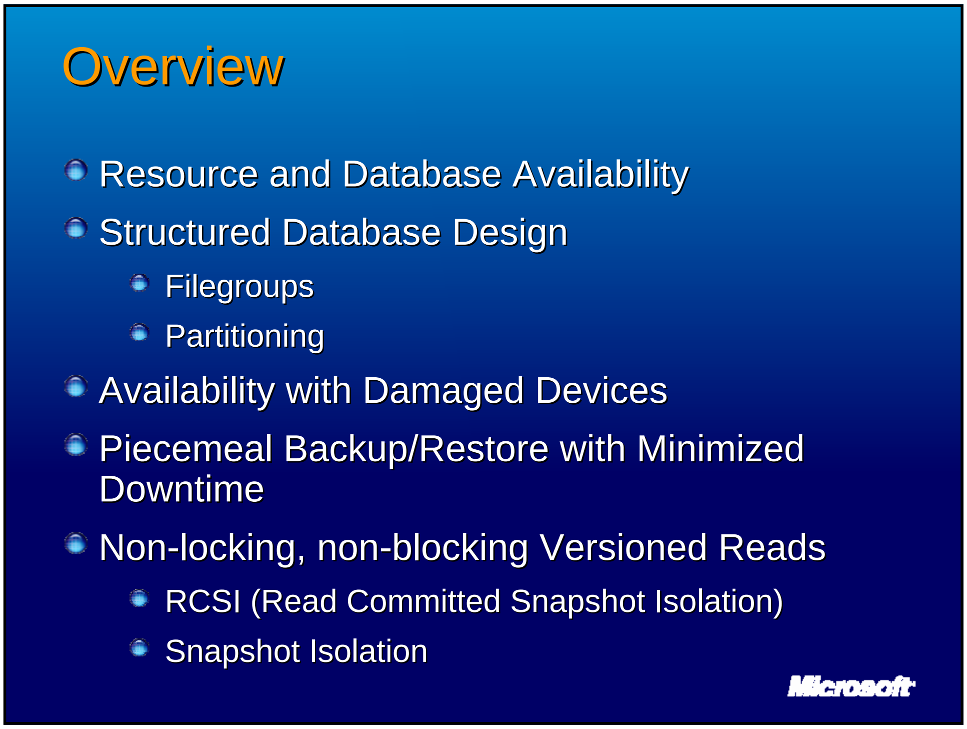


Speaker – Kimberly L. Tripp

- Independent Consultant/Trainer/Speaker/Writer
- Founder, *SYS*olutions, Inc. www.SQLskills.com

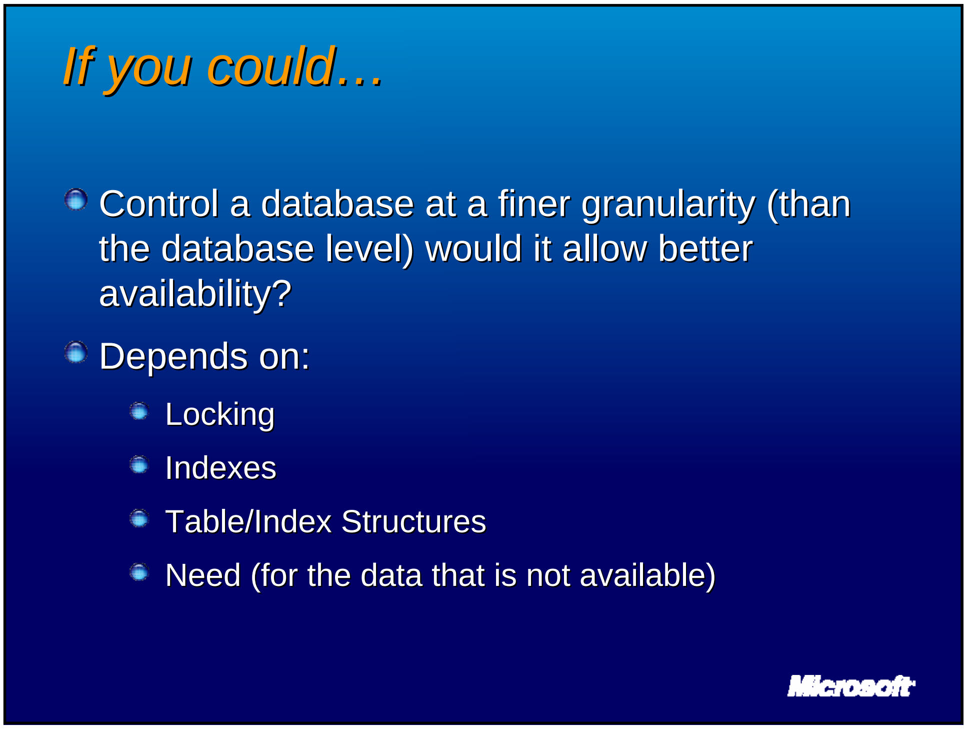
  email: Kimberly@SQLskills.com

  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
- Coauthor MSPress: *SQL Server 2000 High Availability*
- Presenter/Technical Manager for SQL Server 2000 High Availability Overview DVD (MS Part# 098-96661)



Overview

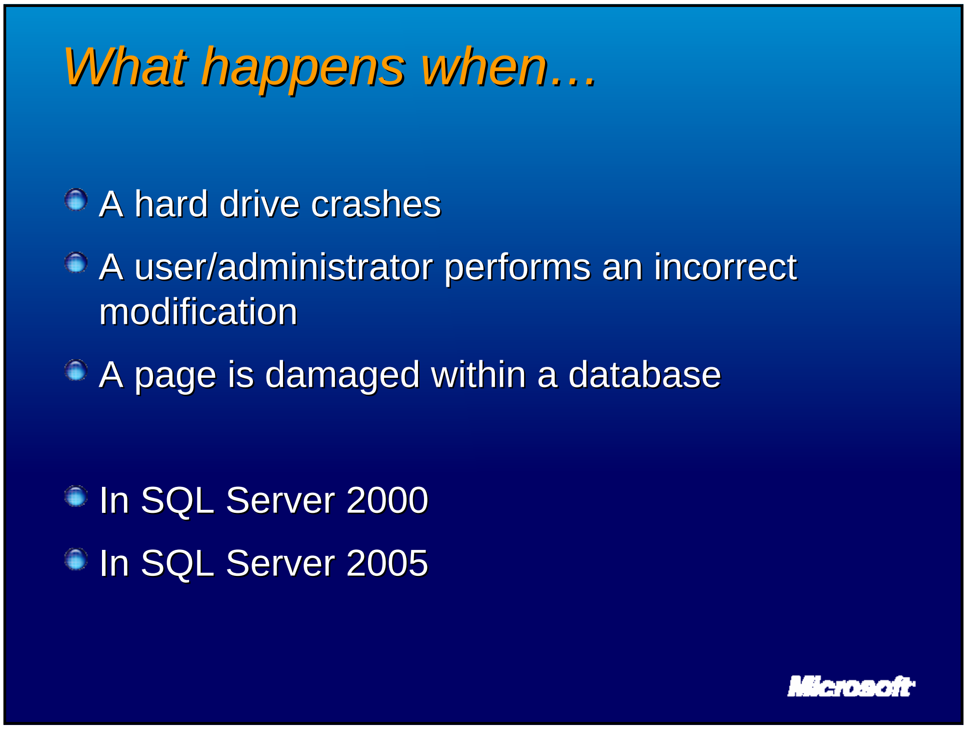
- Resource and Database Availability
- Structured Database Design
  - Filegroups
  - Partitioning
- Availability with Damaged Devices
- Piecemeal Backup/Restore with Minimized Downtime
- Non-locking, non-blocking Versioned Reads
  - RCSI (Read Committed Snapshot Isolation)
  - Snapshot Isolation



If you could…

- Control a database at a finer granularity (than the database level) would it allow better availability?
- Depends on:
  - Locking
  - Indexes
  - Table/Index Structures
  - Need (for the data that is not available)

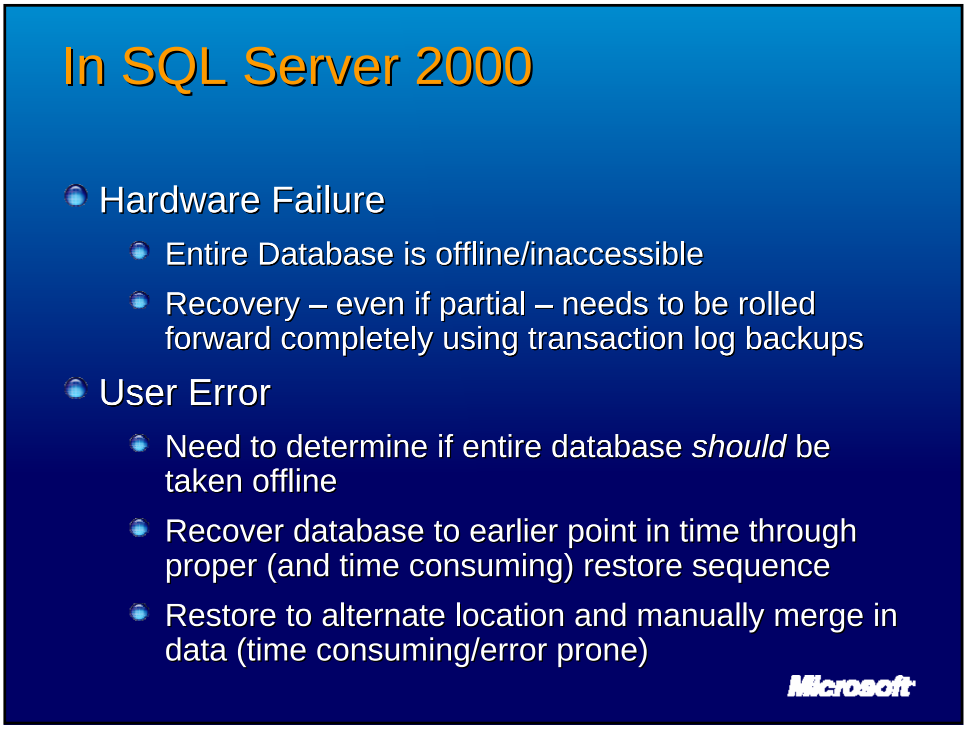
This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.



What happens when…

- A hard drive crashes
- A user/administrator performs an incorrect modification
- A page is damaged within a database

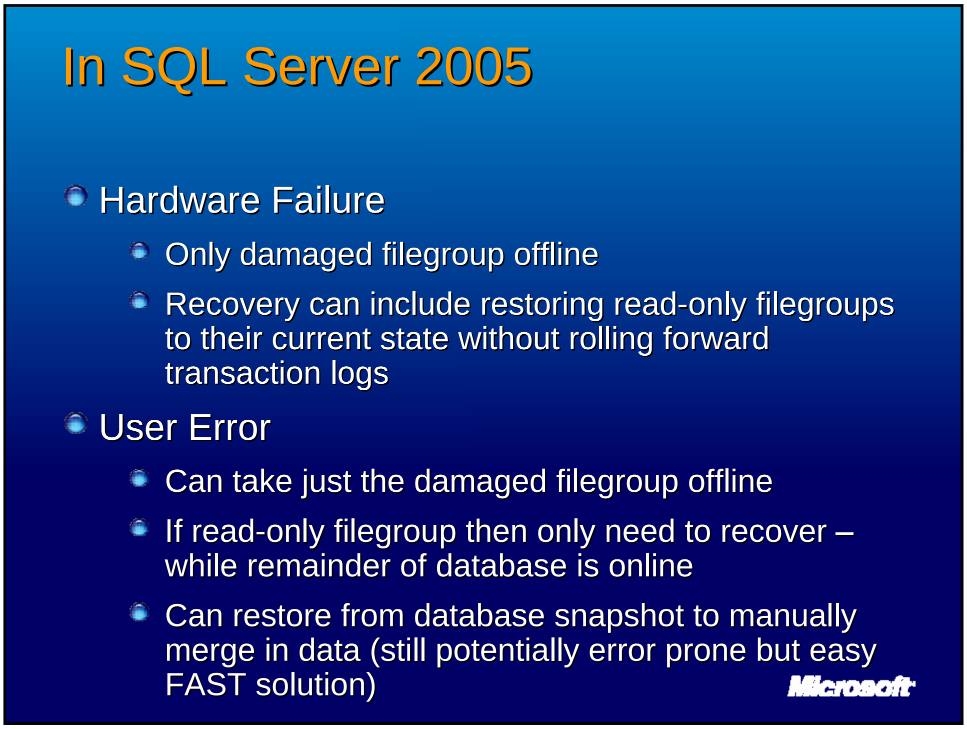
- In SQL Server 2000
- In SQL Server 2005



In SQL Server 2000

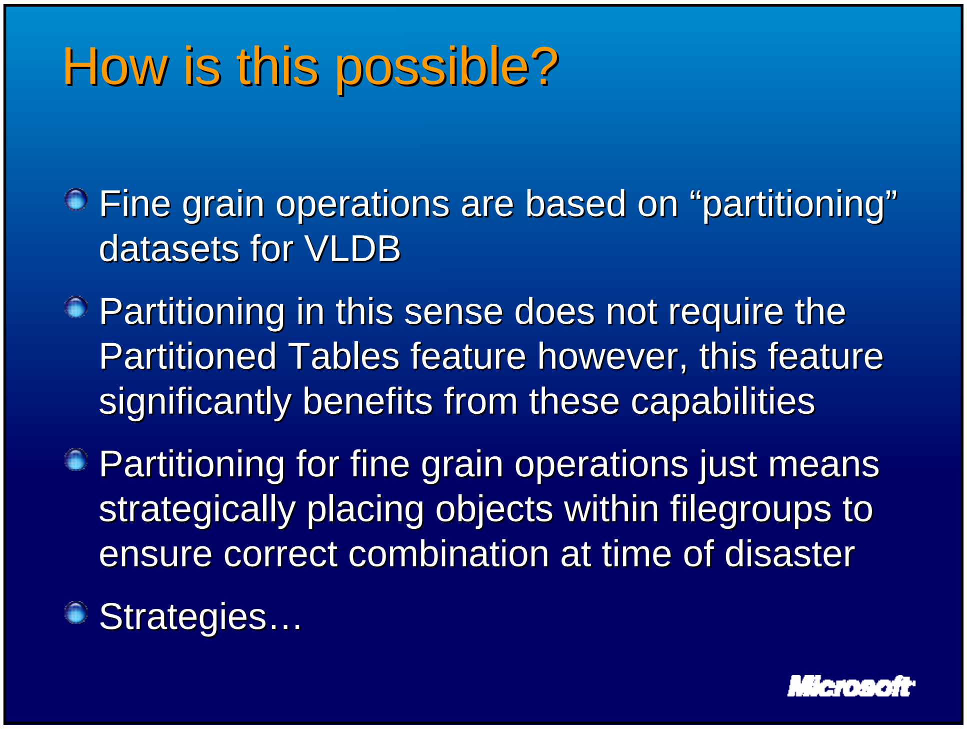
- Hardware Failure
  - Entire Database is offline/inaccessible
  - Recovery – even if partial – needs to be rolled forward completely using transaction log backups
- User Error
  - Need to determine if entire database *should* be taken offline
  - Recover database to earlier point in time through proper (and time consuming) restore sequence
  - Restore to alternate location and manually merge in data (time consuming/error prone)

This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.



In SQL Server 2005

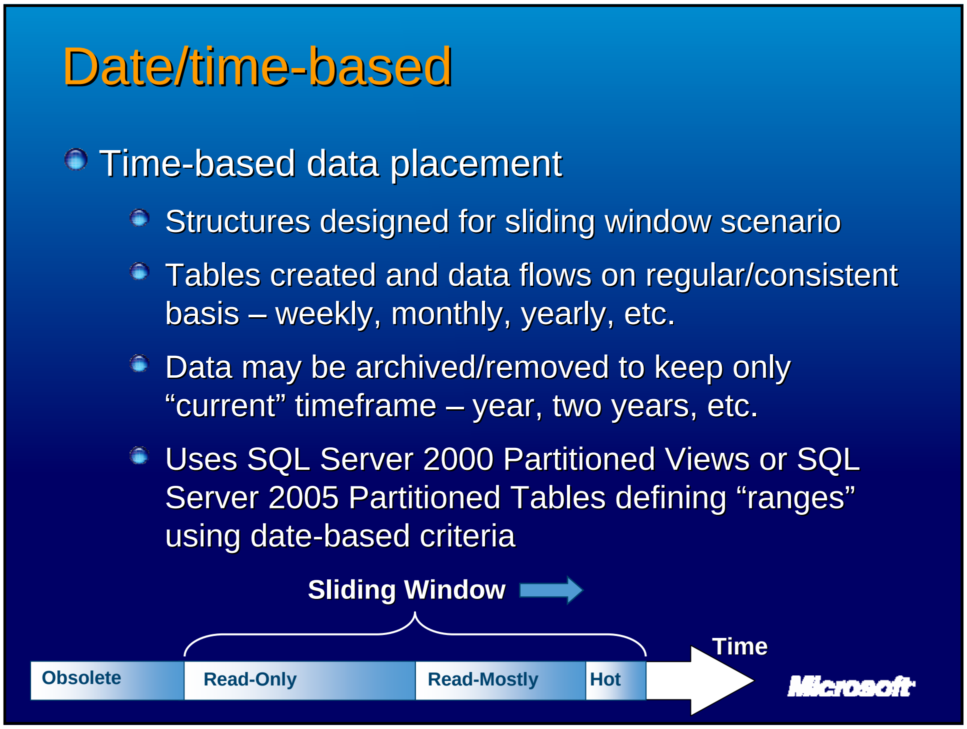
- Hardware Failure
  - Only damaged filegroup offline
  - Recovery can include restoring read-only filegroups to their current state without rolling forward transaction logs
- User Error
  - Can take just the damaged filegroup offline
  - If read-only filegroup then only need to recover – while remainder of database is online
  - Can restore from database snapshot to manually merge in data (still potentially error prone but easy FAST solution)



How is this possible?

- Fine grain operations are based on “partitioning” datasets for VLDB
- Partitioning in this sense does not require the Partitioned Tables feature however, this feature significantly benefits from these capabilities
- Partitioning for fine grain operations just means strategically placing objects within filegroups to ensure correct combination at time of disaster
- Strategies…

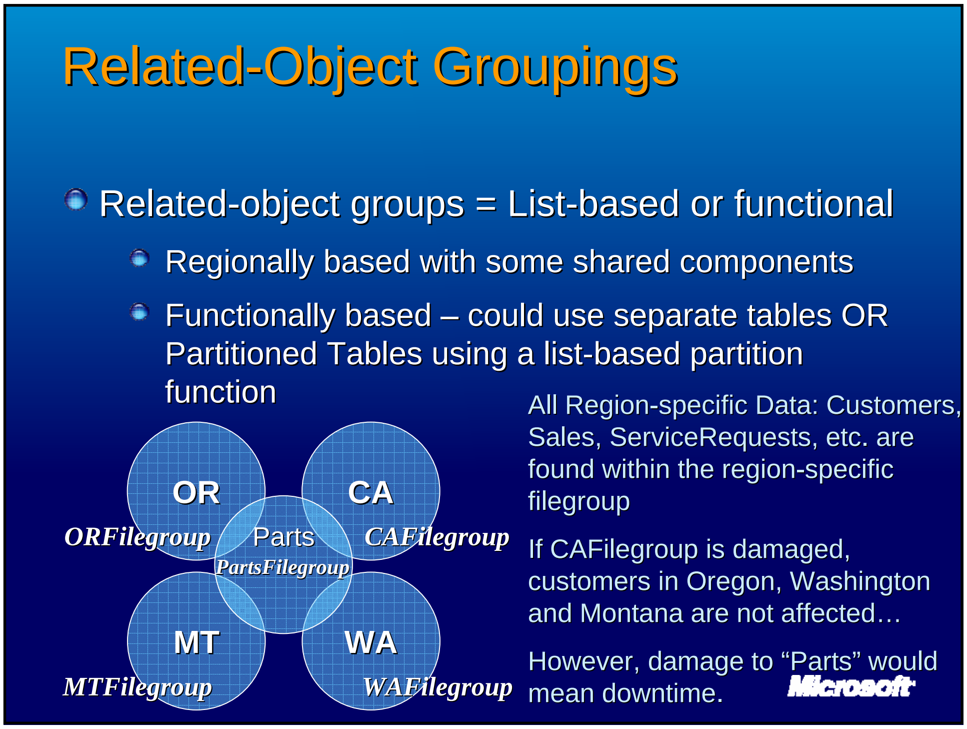
This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.



Date/time-based

- Time-based data placement
  - Structures designed for sliding window scenario
  - Tables created and data flows on regular/consistent basis – weekly, monthly, yearly, etc.
  - Data may be archived/removed to keep only “current” timeframe – year, two years, etc.
  - Uses SQL Server 2000 Partitioned Views or SQL Server 2005 Partitioned Tables defining “ranges” using date-based criteria

Sliding Window
Time
Obsolete
Read-Only
Read-Mostly
Hot



Related-Object Groupings

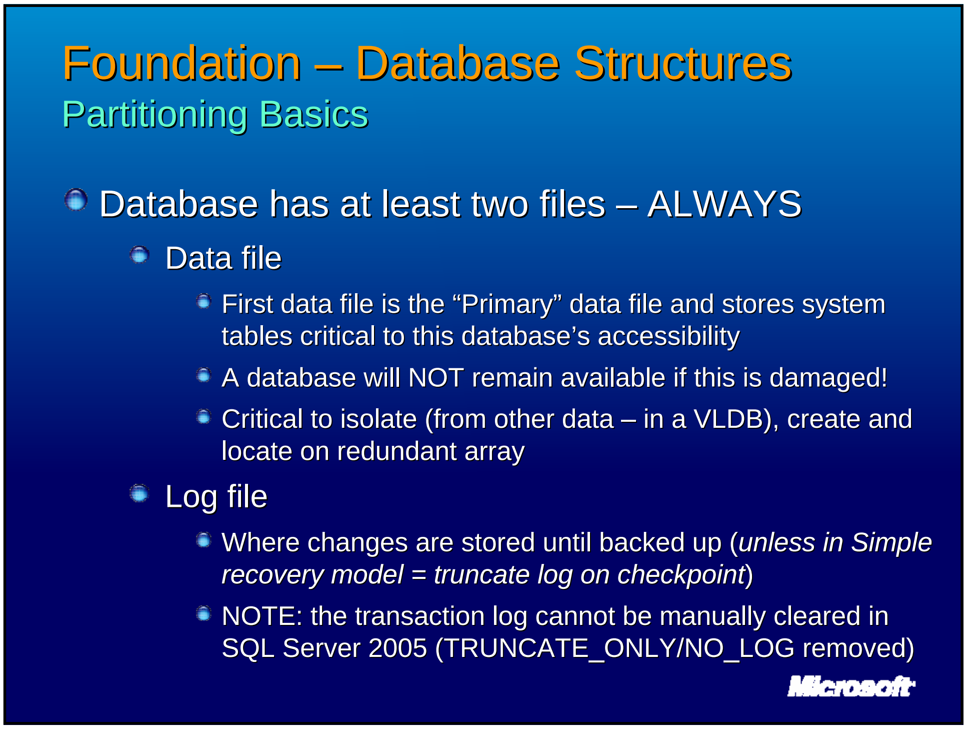
- Related-object groups = List-based or functional
  - Regionally based with some shared components
  - Functionally based – could use separate tables OR Partitioned Tables using a list-based partition function

All Region-specific Data: Customers, Sales, ServiceRequests, 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.

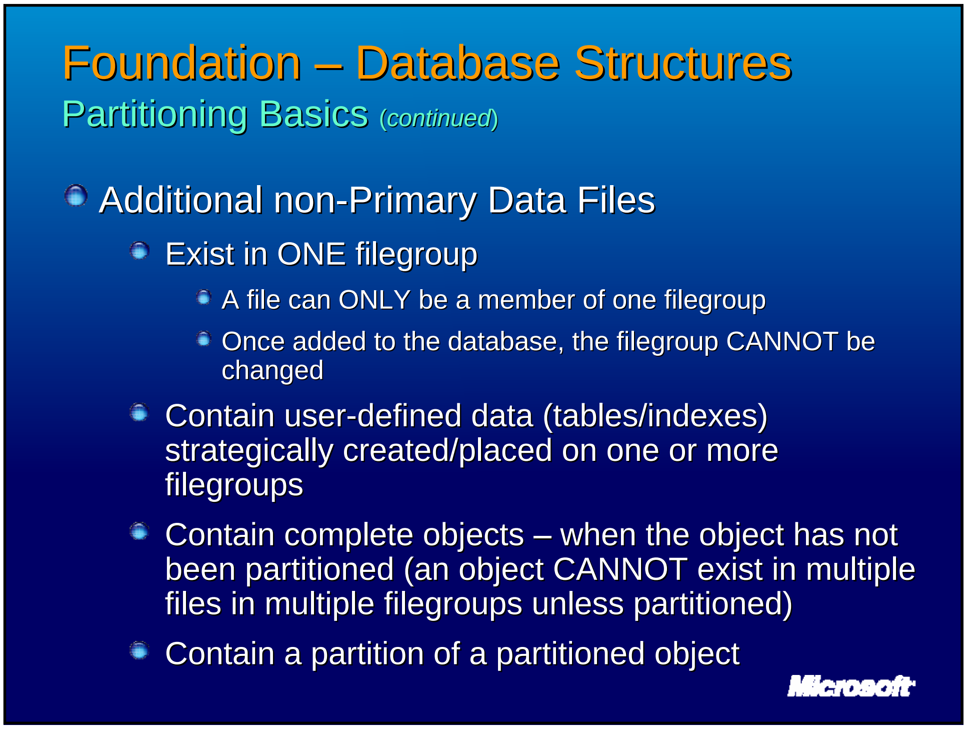
This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.



Foundation – Database Structures

Partitioning Basics

- Database has at least two files – ALWAYS
  - Data file
    - First data file is the "Primary" data file and stores system tables critical to this database's accessibility
    - A database will NOT remain available if this is damaged!
    - Critical to isolate (from other data – in a VLDB), create and locate on redundant array
  - Log file
    - Where changes are stored until backed up (*unless in Simple recovery model = truncate log on checkpoint*)
    - NOTE: the transaction log cannot be manually cleared in SQL Server 2005 (TRUNCATE_ONLY/NO_LOG removed)



Foundation – Database Structures

Partitioning Basics *(continued)*

- Additional non-Primary Data Files
  - Exist in ONE filegroup
    - A file can ONLY be a member of one filegroup
    - Once added to the database, the filegroup CANNOT be changed
  - Contain user-defined data (tables/indexes) strategically created/placed on one or more filegroups
  - Contain complete objects – when the object has not been partitioned (an object CANNOT exist in multiple files in multiple filegroups unless partitioned)
  - Contain a partition of a partitioned object

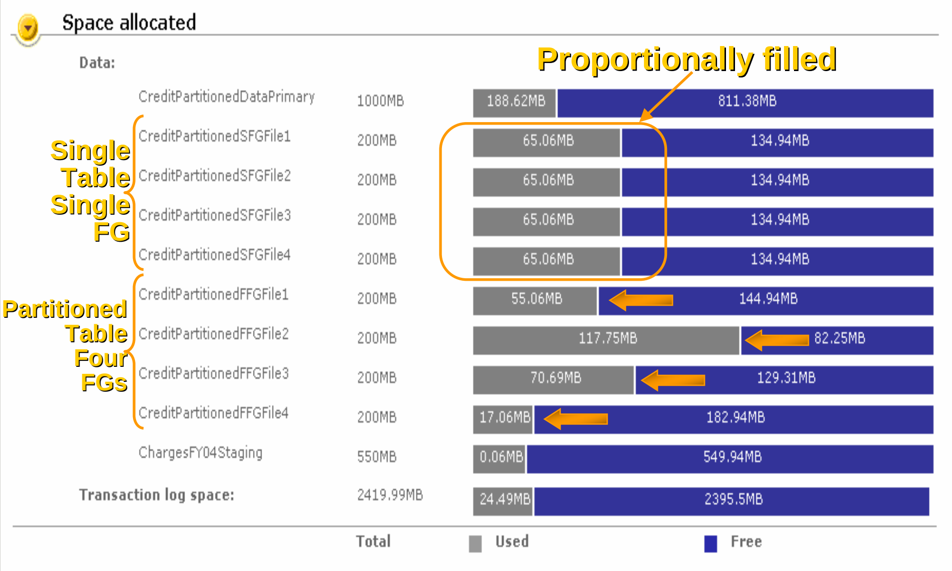
Creating Objects on Filegroups

- Objects = Tables/Indexes can be created
  - A Filegroup
    - Can contain one or more files
    - Data is proportionally filled among the files in the filegroup
  - A Partition Scheme
    - Can contain one or more filegroups
    - Data is placed into the appropriate filegroup based on a partition function

```
CREATE TABLE Customers
( column   datatype   nullability, ...)
ON FILEGROUP
```

```
CREATE TABLE Sales
( column   datatype   nullability, ...)
ON PartitionScheme
```

Proportional Fill v. Partitioned Table



Space allocated

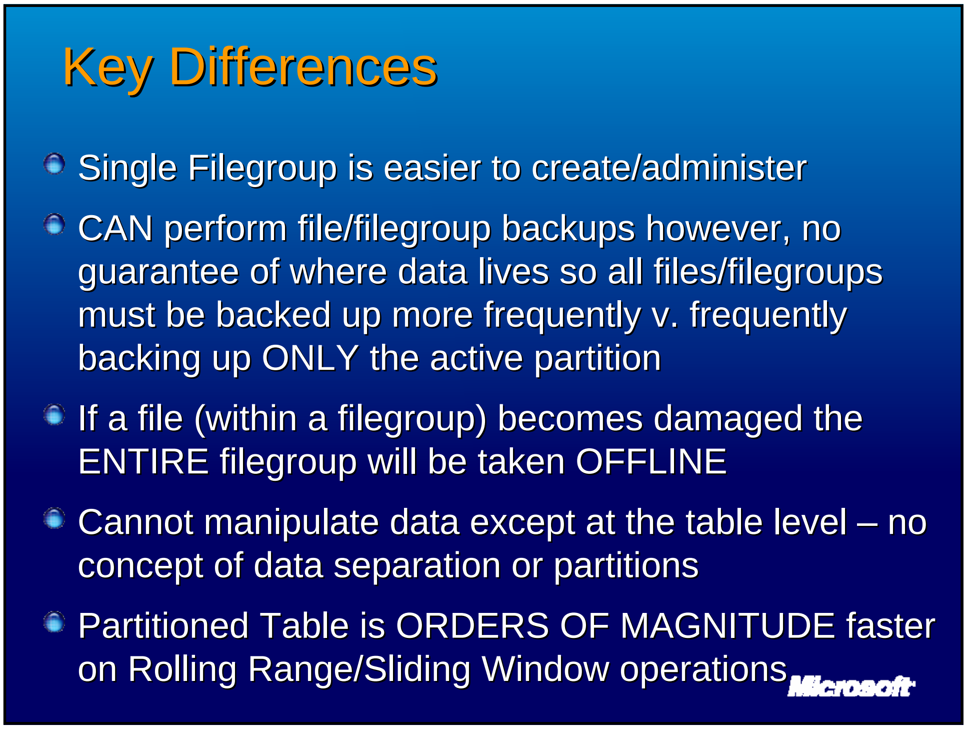
Proportionally filled

Single Table Single FG

Partitioned Table Four FGs

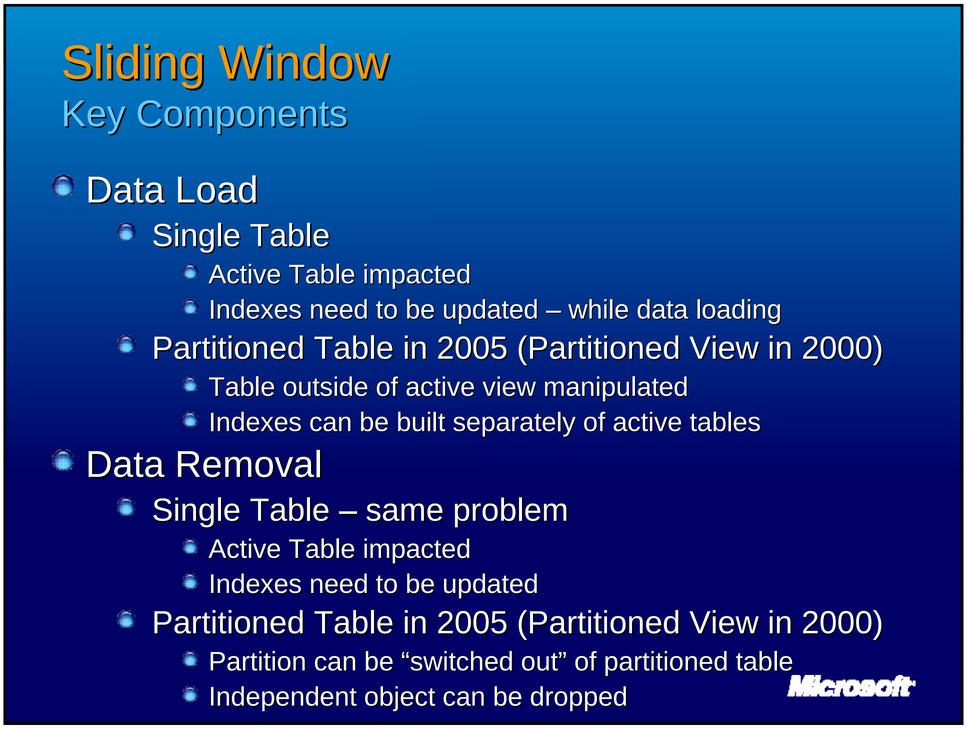
Data:	Total	Used	Free
CreditPartitionedDataPrimary	1000MB	188.62MB	811.38MB
CreditPartitionedSFGFile1	200MB	65.06MB	134.94MB
CreditPartitionedSFGFile2	200MB	65.06MB	134.94MB
CreditPartitionedSFGFile3	200MB	65.06MB	134.94MB
CreditPartitionedSFGFile4	200MB	65.06MB	134.94MB
CreditPartitionedFFGFile1	200MB	55.06MB	144.94MB
CreditPartitionedFFGFile2	200MB	117.75MB	82.25MB
CreditPartitionedFFGFile3	200MB	70.69MB	129.31MB
CreditPartitionedFFGFile4	200MB	17.06MB	182.94MB
ChargesFY04Staging	550MB	0.06MB	549.94MB
Transaction log space:	2419.99MB	24.49MB	2395.5MB

Total ▨ Used ■ Free



Key Differences

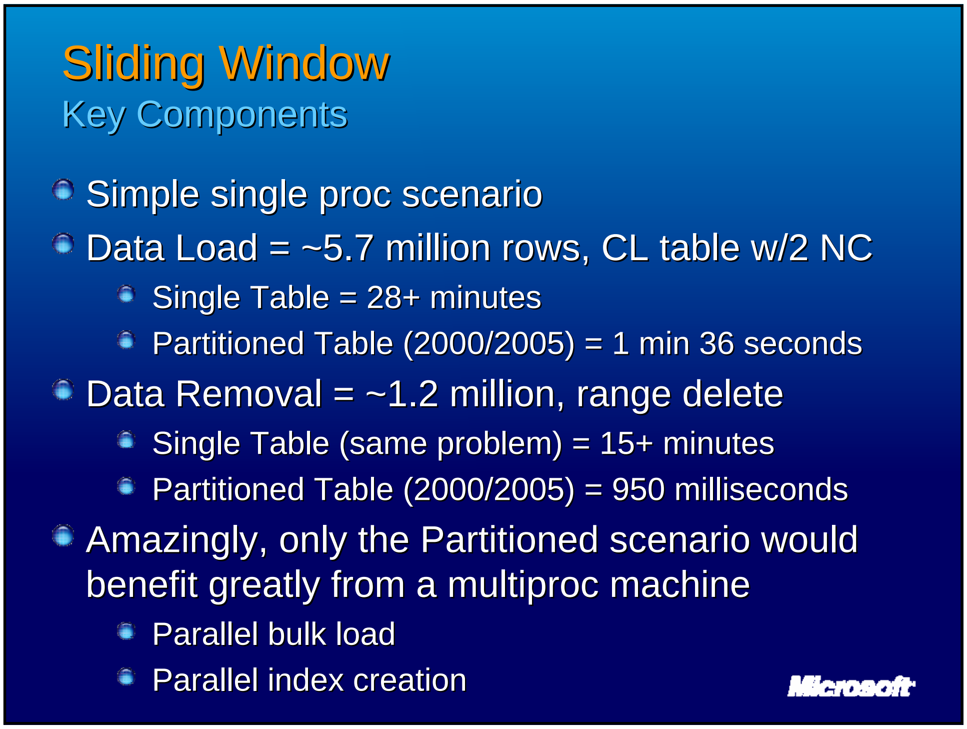
- Single Filegroup is easier to create/administer
- CAN perform file/filegroup backups however, no guarantee of where data lives so all files/filegroups must be backed up more frequently v. frequently backing up ONLY the active partition
- If a file (within a filegroup) becomes damaged the ENTIRE filegroup will be taken OFFLINE
- Cannot manipulate data except at the table level – no concept of data separation or partitions
- Partitioned Table is ORDERS OF MAGNITUDE faster on Rolling Range/Sliding Window operations



Sliding Window

Key Components

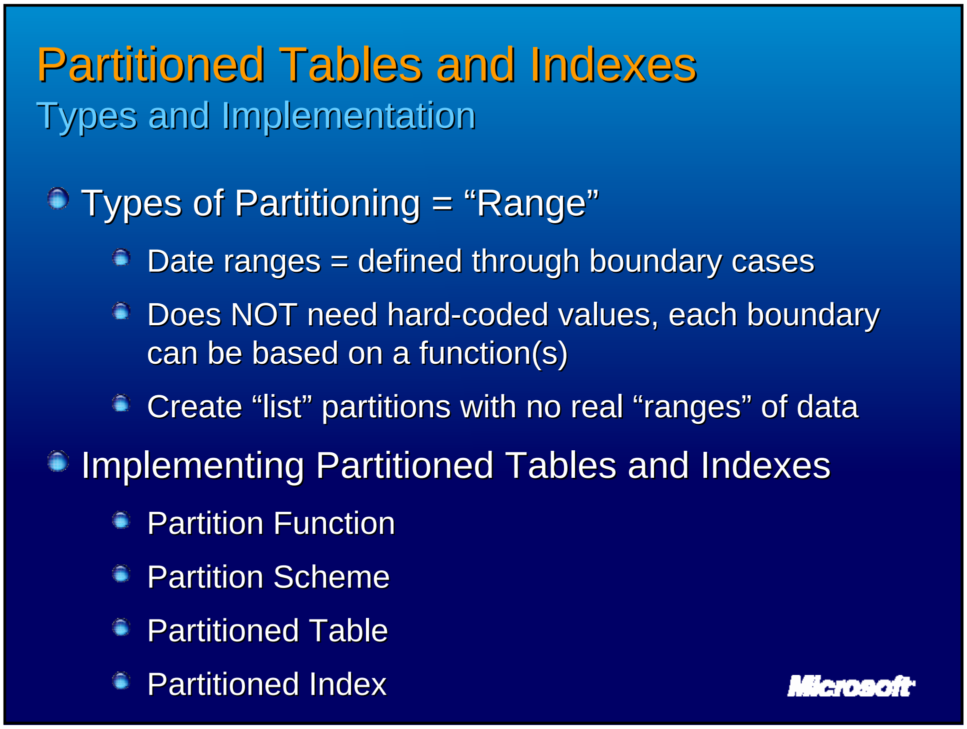
- Data Load
  - Single Table
    - Active Table impacted
    - Indexes need to be updated – while data loading
  - Partitioned Table in 2005 (Partitioned View in 2000)
    - Table outside of active view manipulated
    - Indexes can be built separately of active tables
- Data Removal
  - Single Table – same problem
    - Active Table impacted
    - Indexes need to be updated
  - Partitioned Table in 2005 (Partitioned View in 2000)
    - Partition can be “switched out” of partitioned table
    - Independent object can be dropped



Sliding Window

Key Components

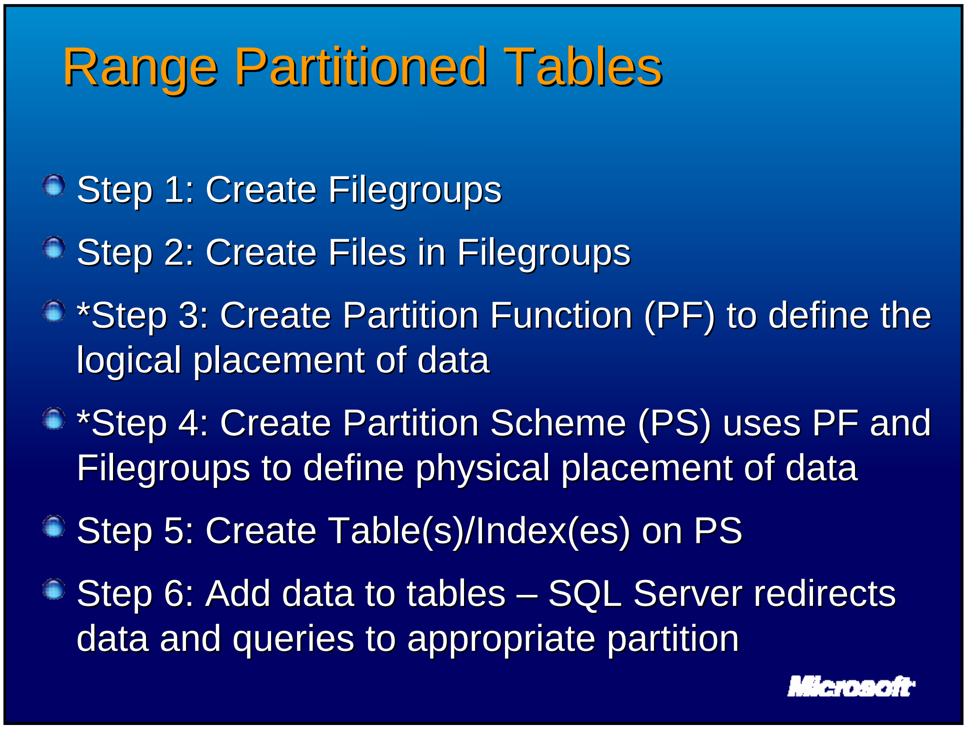
- Simple single proc scenario
- Data Load = ~5.7 million rows, CL table w/2 NC
  - Single Table = 28+ minutes
  - Partitioned Table (2000/2005) = 1 min 36 seconds
- Data Removal = ~1.2 million, range delete
  - Single Table (same problem) = 15+ minutes
  - Partitioned Table (2000/2005) = 950 milliseconds
- Amazingly, only the Partitioned scenario would benefit greatly from a multiproc machine
  - Parallel bulk load
  - Parallel index creation



Partitioned Tables and Indexes

Types and Implementation

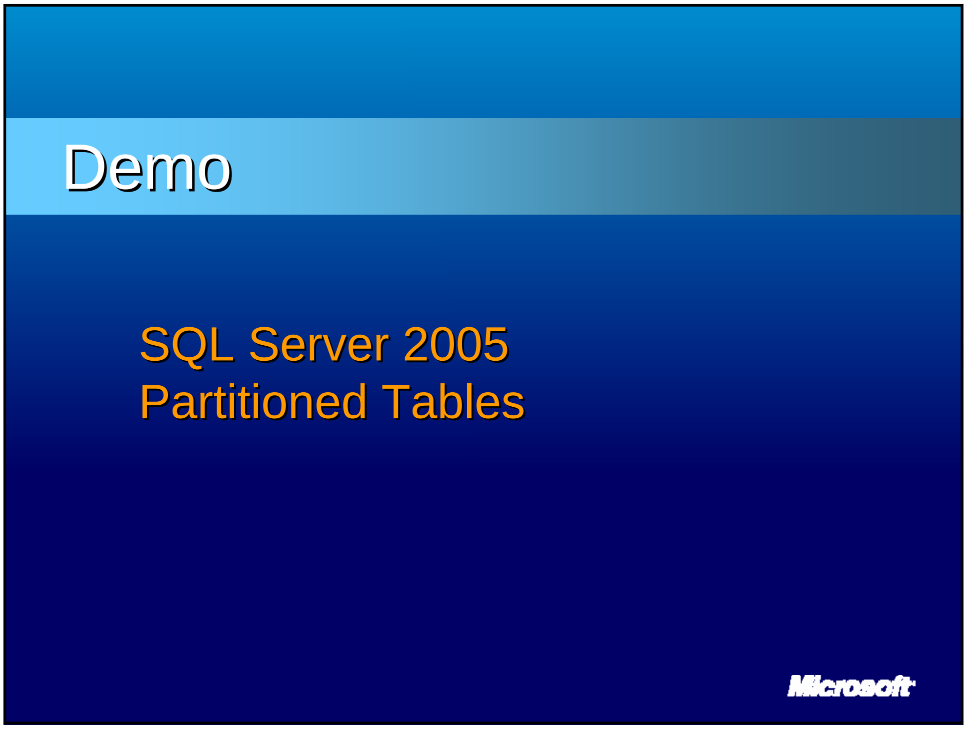
- Types of Partitioning = “Range”
  - Date ranges = defined through boundary cases
  - Does NOT need hard-coded values, each boundary can be based on a function(s)
  - Create “list” partitions with no real “ranges” of data
- Implementing Partitioned Tables and Indexes
  - Partition Function
  - Partition Scheme
  - Partitioned Table
  - Partitioned Index



Range Partitioned Tables

- Step 1: Create Filegroups
- Step 2: Create Files in Filegroups
- *Step 3: Create Partition Function (PF) to define the logical placement of data
- *Step 4: Create Partition Scheme (PS) uses PF and Filegroups to define physical placement of data
- Step 5: Create Table(s)/Index(es) on PS
- Step 6: Add data to tables – SQL Server redirects data and queries to appropriate partition

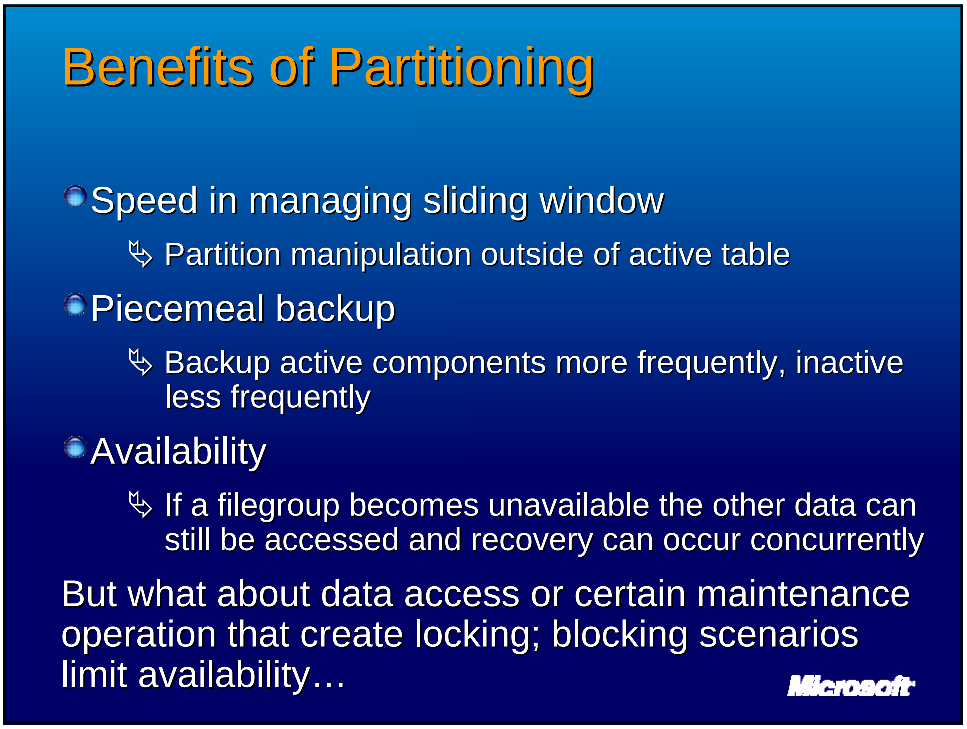
Microsoft



Demo

SQL Server 2005
Partitioned Tables

Microsoft

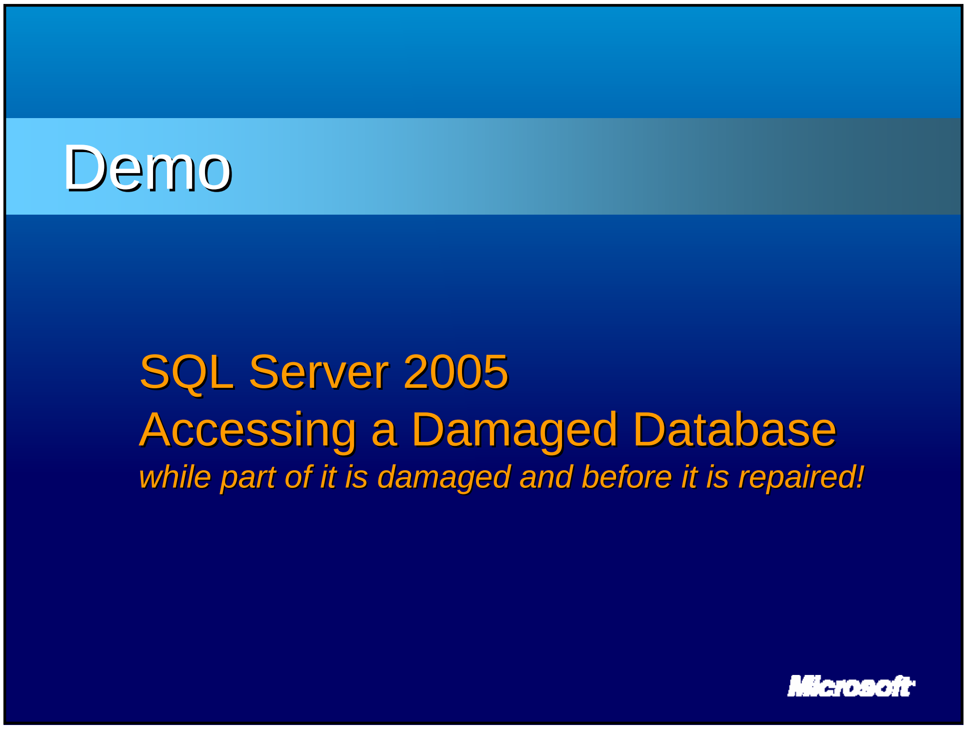


Benefits of Partitioning

- Speed in managing sliding window
  - ↳ Partition manipulation outside of active table
- Piecemeal backup
  - ↳ Backup active components more frequently, inactive less frequently
- Availability
  - ↳ If a filegroup becomes unavailable the other data can still be accessed and recovery can occur concurrently

But what about data access or certain maintenance operation that create locking; blocking scenarios limit availability…

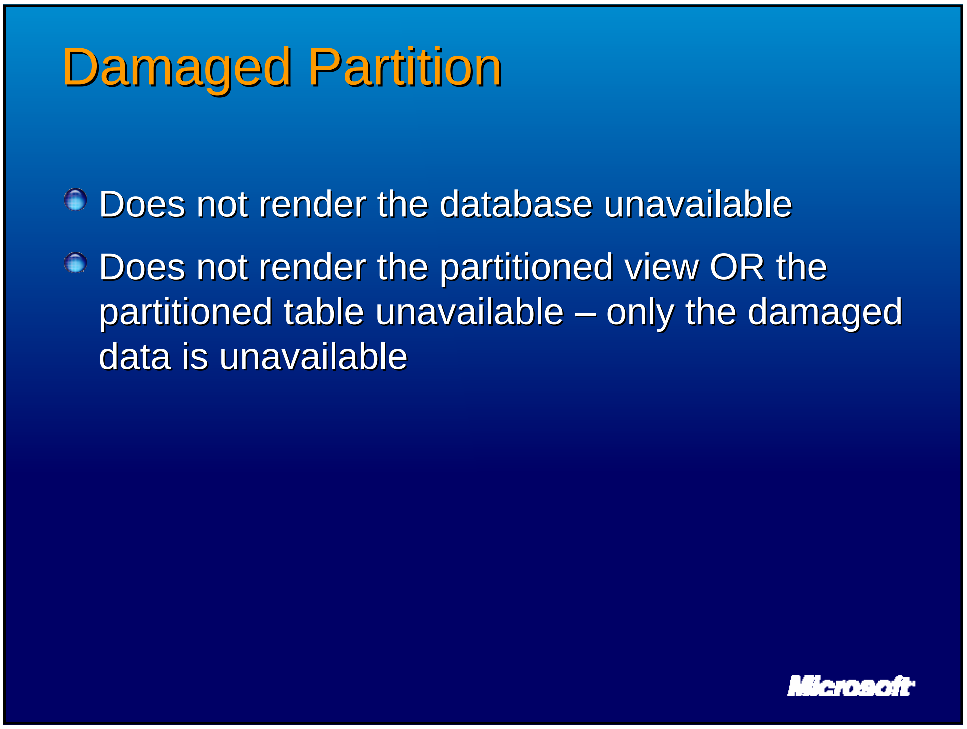
Microsoft



Demo

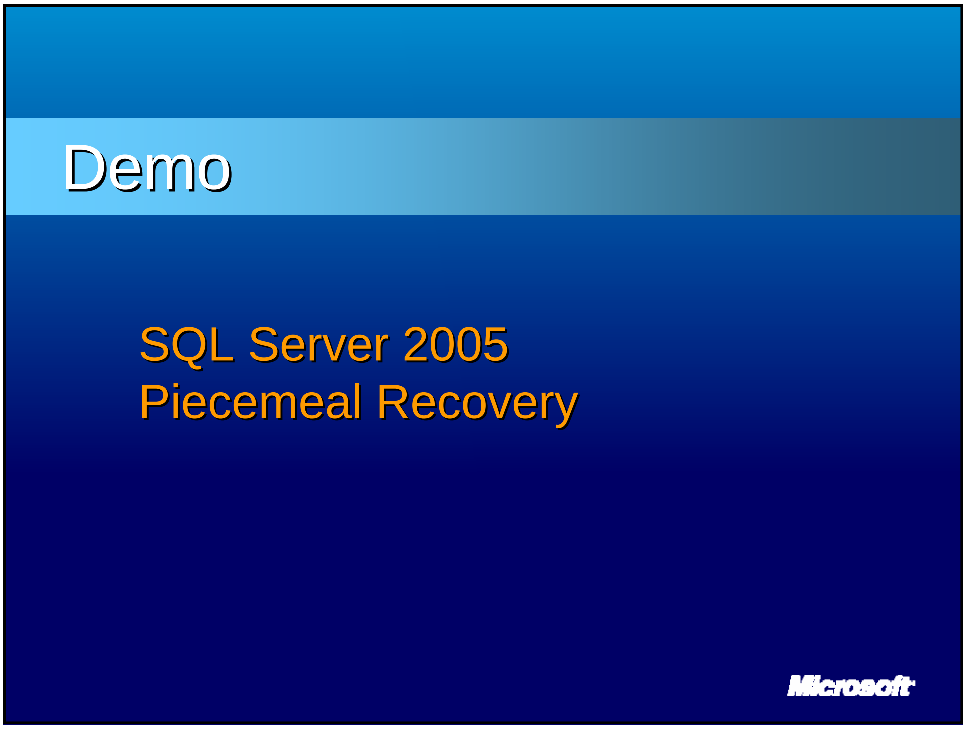
SQL Server 2005
Accessing a Damaged Database
while part of it is damaged and before it is repaired!

Microsoft



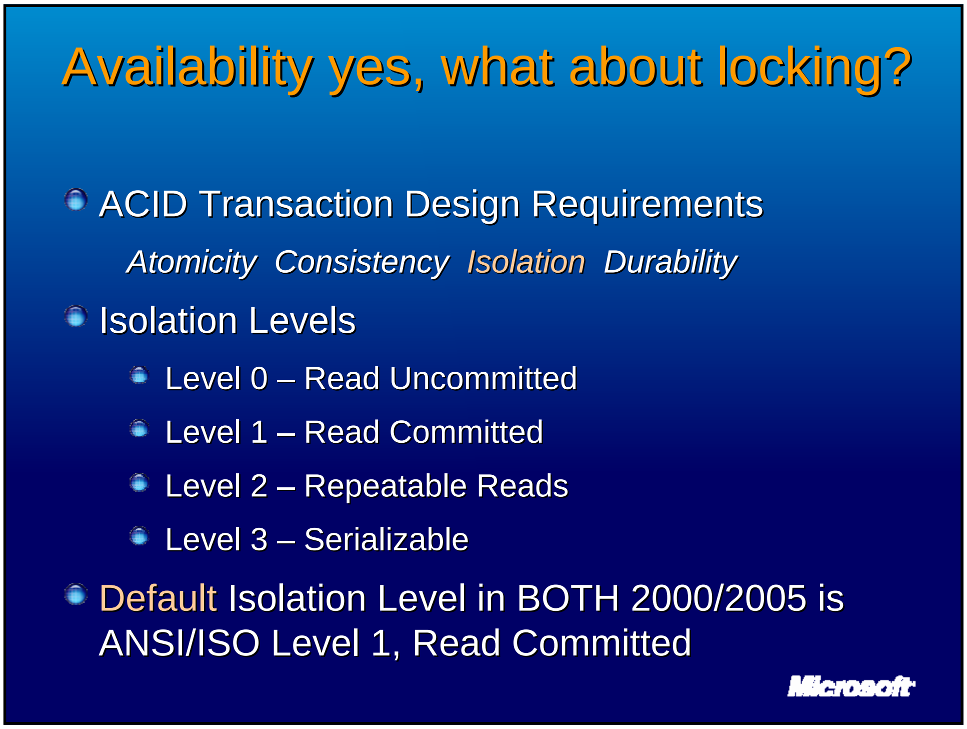
Damaged Partition

- Does not render the database unavailable
- Does not render the partitioned view OR the partitioned table unavailable – only the damaged data is unavailable



Demo

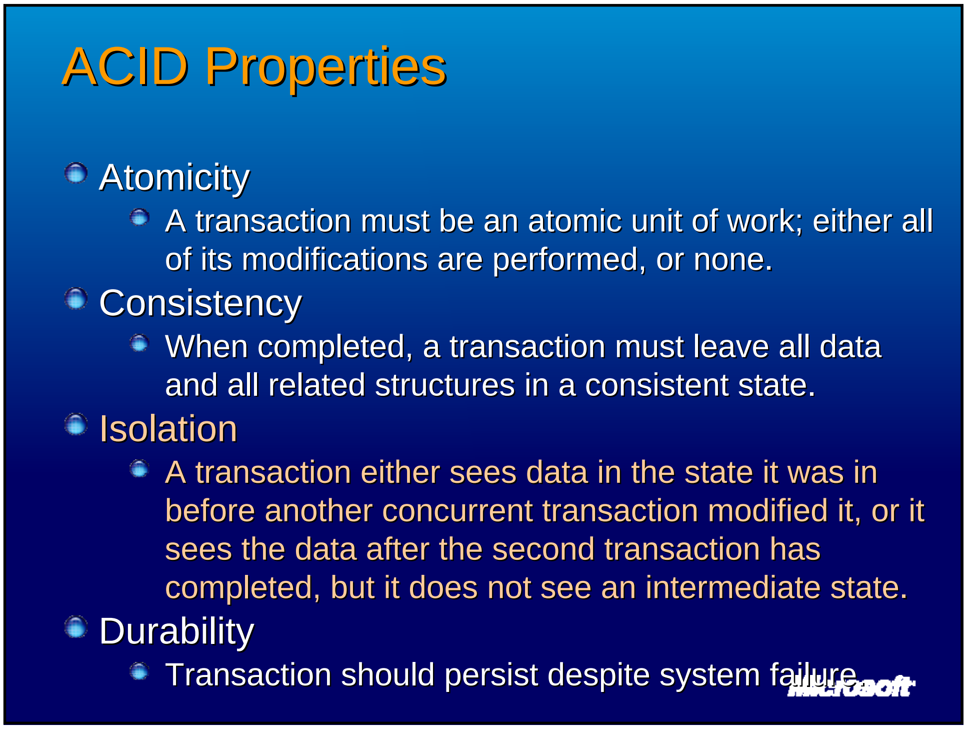
SQL Server 2005
Piecemeal Recovery



Availability yes, what about locking?

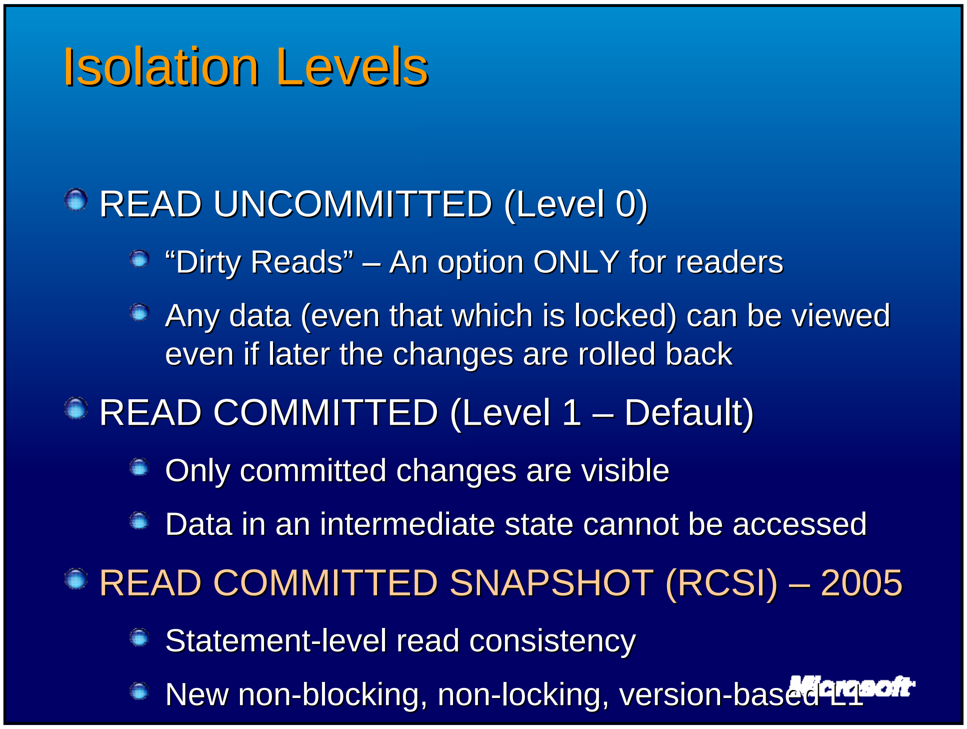
- ACID Transaction Design Requirements

  Atomicity Consistency Isolation Durability
- Isolation Levels
  - Level 0 – Read Uncommitted
  - Level 1 – Read Committed
  - Level 2 – Repeatable Reads
  - Level 3 – Serializable
- Default Isolation Level in BOTH 2000/2005 is ANSI/ISO Level 1, Read Committed



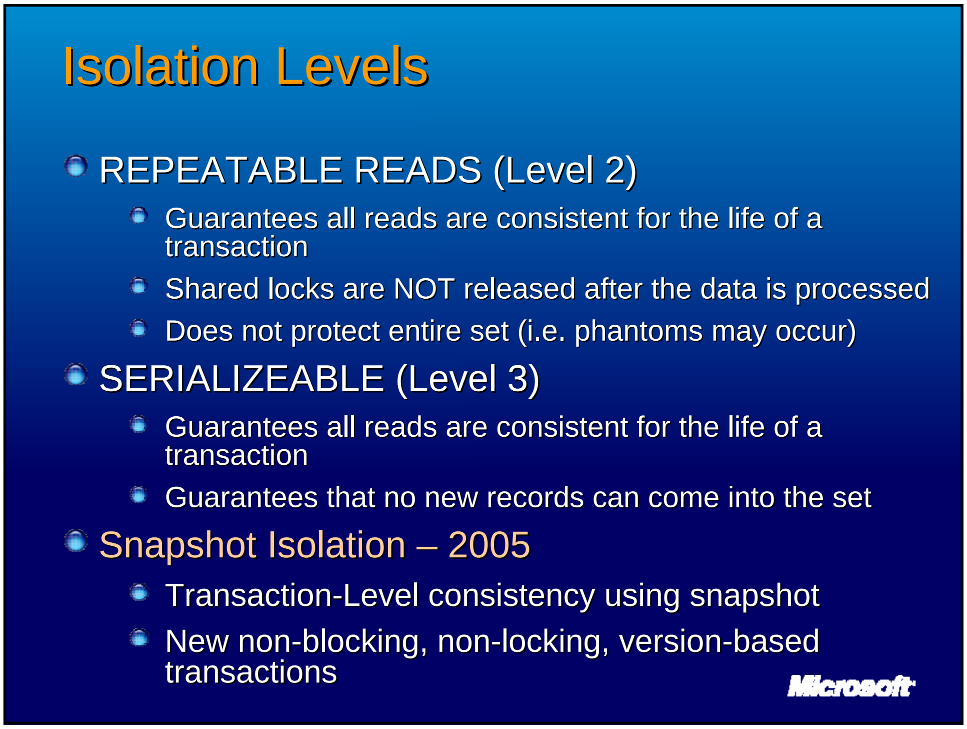
ACID Properties

- Atomicity
  - A transaction must be an atomic unit of work; either all of its modifications are performed, or none.
- Consistency
  - When completed, a transaction must leave all data and all related structures in a consistent state.
- Isolation
  - A transaction either sees data in the state it was in before another concurrent transaction modified it, or it sees the data after the second transaction has completed, but it does not see an intermediate state.
- Durability
  - Transaction should persist despite system failure.



Isolation Levels

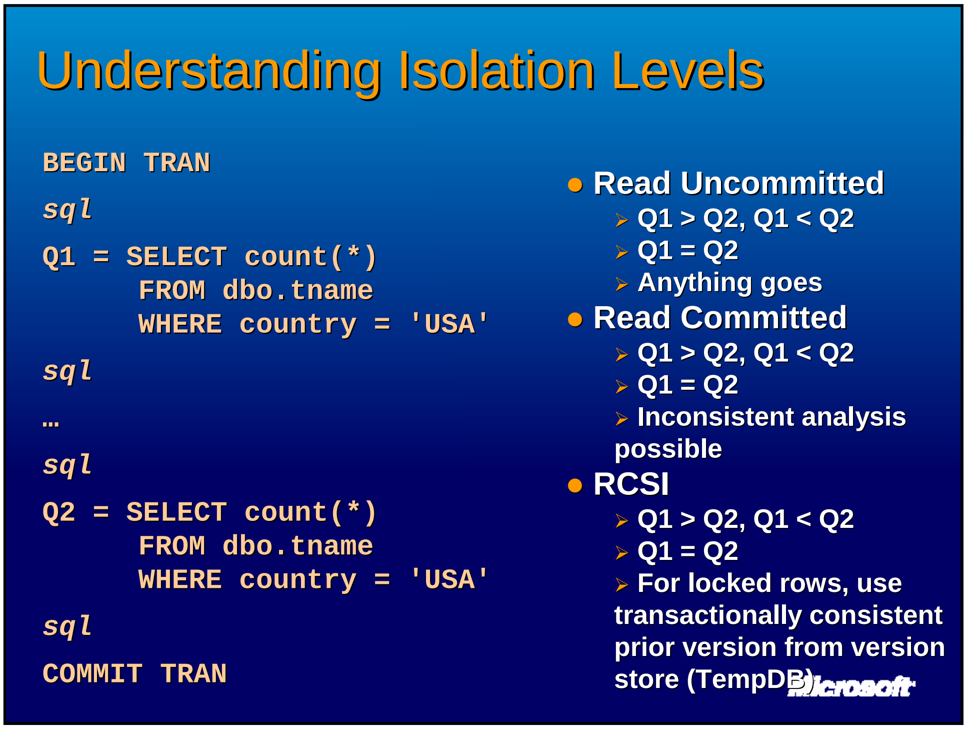
- READ UNCOMMITTED (Level 0)
  - “Dirty Reads” – An option ONLY for readers
  - Any data (even that which is locked) can be viewed even if later the changes are rolled back
- READ COMMITTED (Level 1 – Default)
  - Only committed changes are visible
  - Data in an intermediate state cannot be accessed
- READ COMMITTED SNAPSHOT (RCSI) – 2005
  - Statement-level read consistency
  - New non-blocking, non-locking, version-based L1



Isolation Levels

- REPEATABLE READS (Level 2)
  - Guarantees all reads are consistent for the life of a transaction
  - Shared locks are NOT released after the data is processed
  - Does not protect entire set (i.e. phantoms may occur)
- SERIALIZEABLE (Level 3)
  - Guarantees all reads are consistent for the life of a transaction
  - Guarantees that no new records can come into the set
- Snapshot Isolation – 2005
  - Transaction-Level consistency using snapshot
  - New non-blocking, non-locking, version-based transactions

This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.

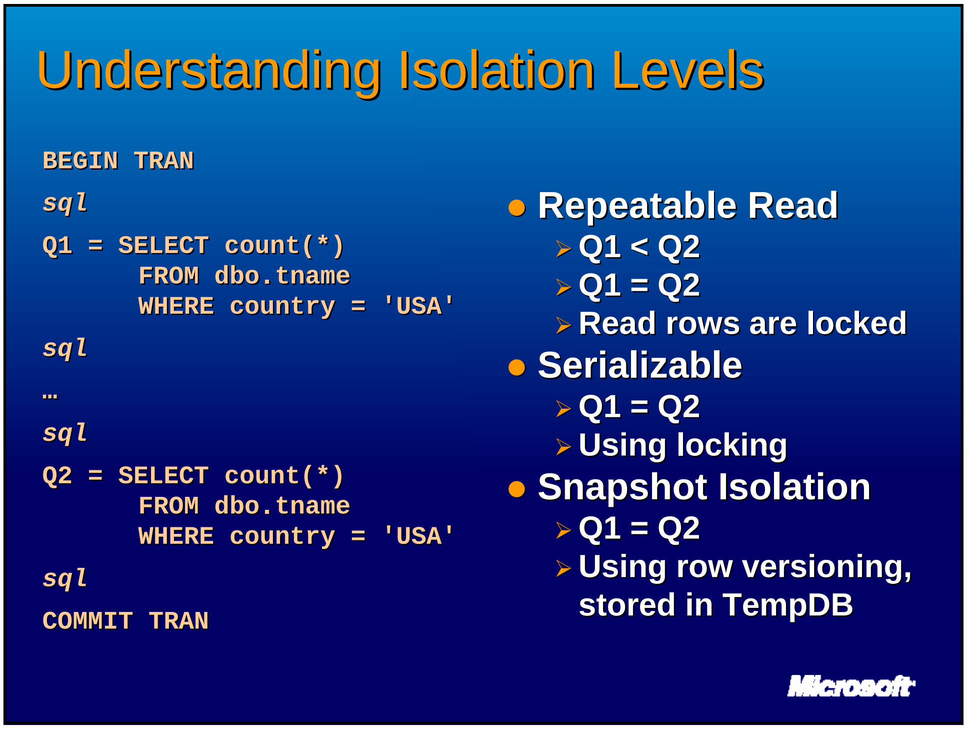


Understanding Isolation Levels

```
BEGIN TRAN
sql
Q1 = SELECT count(*)
       FROM dbo.tname
       WHERE country = 'USA'
sql
…
sql
Q2 = SELECT count(*)
       FROM dbo.tname
       WHERE country = 'USA'
sql
COMMIT TRAN
```

- Read Uncommitted
  - Q1 > Q2, Q1 < Q2
  - Q1 = Q2
  - Anything goes
- Read Committed
  - Q1 > Q2, Q1 < Q2
  - Q1 = Q2
  - Inconsistent analysis possible
- RCSI
  - Q1 > Q2, Q1 < Q2
  - Q1 = Q2
  - For locked rows, use transactionally consistent prior version from version store (TempDB)

Microsoft



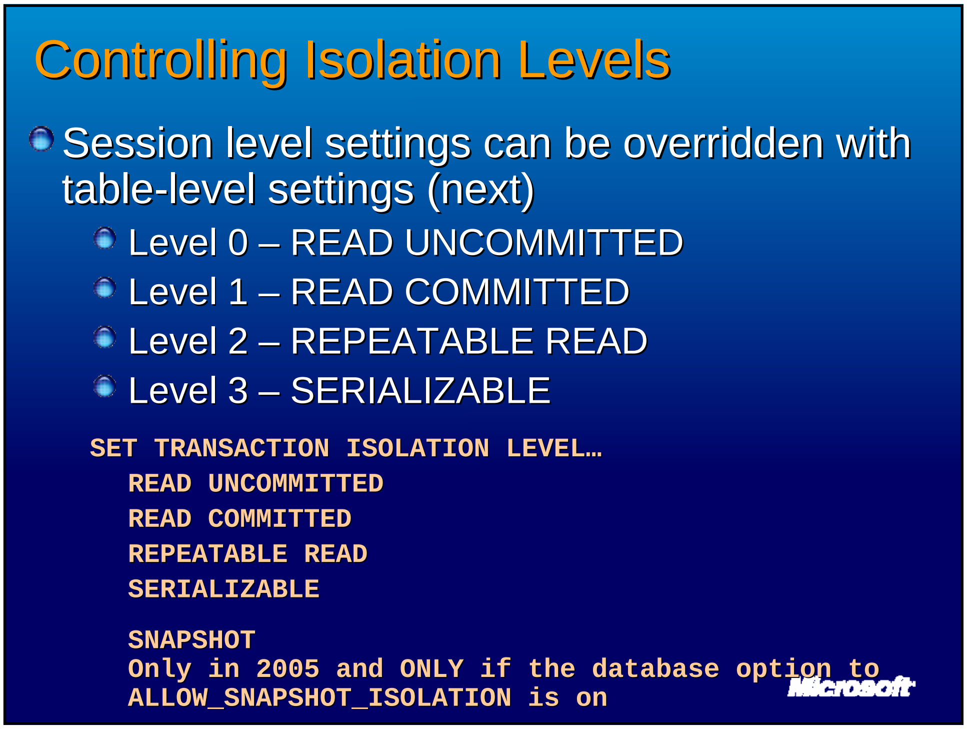
Understanding Isolation Levels

```
BEGIN TRAN
sql
Q1 = SELECT count(*)
       FROM dbo.tname
       WHERE country = 'USA'
sql
…
sql
Q2 = SELECT count(*)
       FROM dbo.tname
       WHERE country = 'USA'
sql
COMMIT TRAN
```

- Repeatable Read
  - Q1 < Q2
  - Q1 = Q2
  - Read rows are locked
- Serializable
  - Q1 = Q2
  - Using locking
- Snapshot Isolation
  - Q1 = Q2
  - Using row versioning, stored in TempDB

Microsoft

This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.

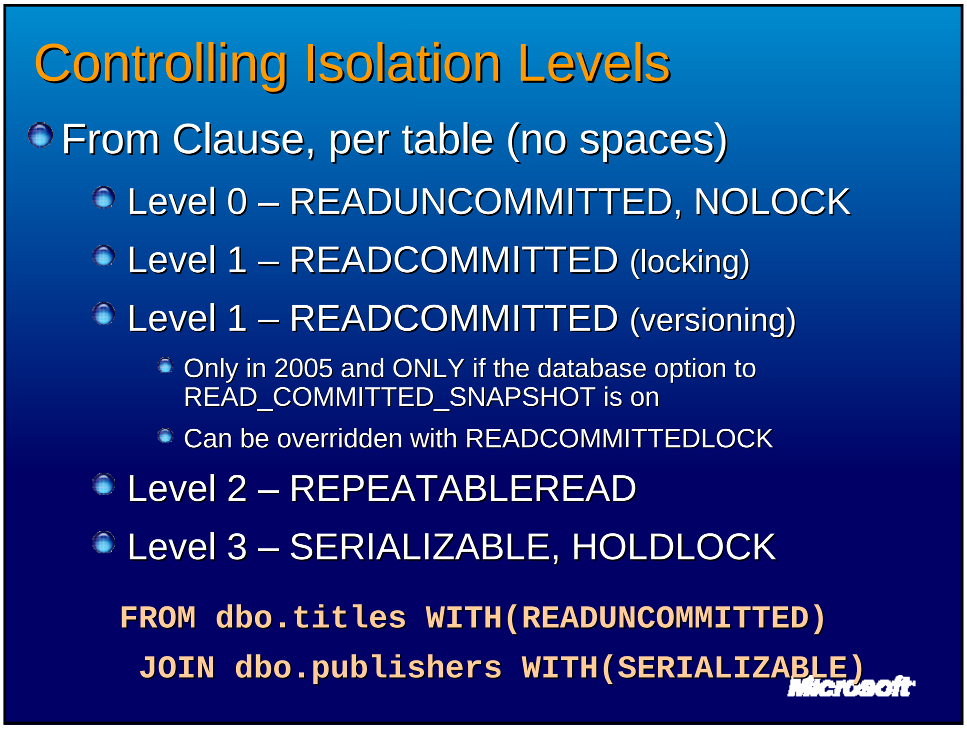


Controlling Isolation Levels

- Session level settings can be overridden with table-level settings (next)
  - Level 0 – READ UNCOMMITTED
  - Level 1 – READ COMMITTED
  - Level 2 – REPEATABLE READ
  - Level 3 – SERIALIZABLE

```
SET TRANSACTION ISOLATION LEVEL…
   READ UNCOMMITTED
   READ COMMITTED
   REPEATABLE READ
   SERIALIZABLE

   SNAPSHOT
   Only in 2005 and ONLY if the database option to
   ALLOW_SNAPSHOT_ISOLATION is on
```

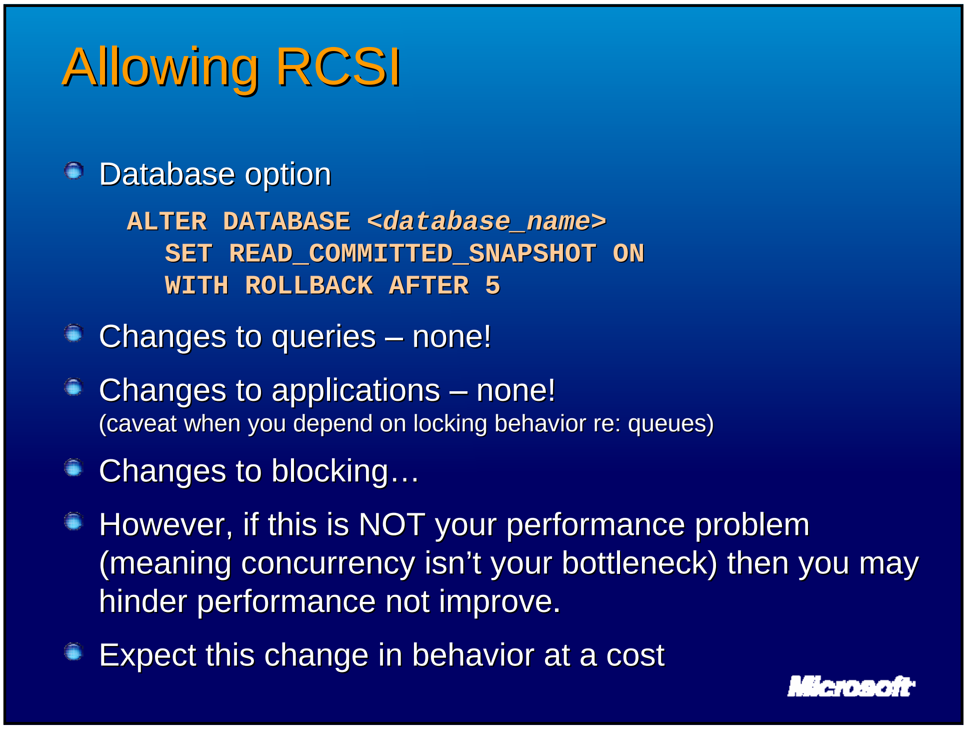


Controlling Isolation Levels

- From Clause, per table (no spaces)
  - Level 0 – READUNCOMMITTED, NOLOCK
  - Level 1 – READCOMMITTED (locking)
  - Level 1 – READCOMMITTED (versioning)
    - Only in 2005 and ONLY if the database option to READ_COMMITTED_SNAPSHOT is on
    - Can be overridden with READCOMMITTEDLOCK
  - Level 2 – REPEATABLEREAD
  - Level 3 – SERIALIZABLE, HOLDLOCK

```
FROM dbo.titles WITH(READUNCOMMITTED)
 JOIN dbo.publishers WITH(SERIALIZABLE)
```

This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.

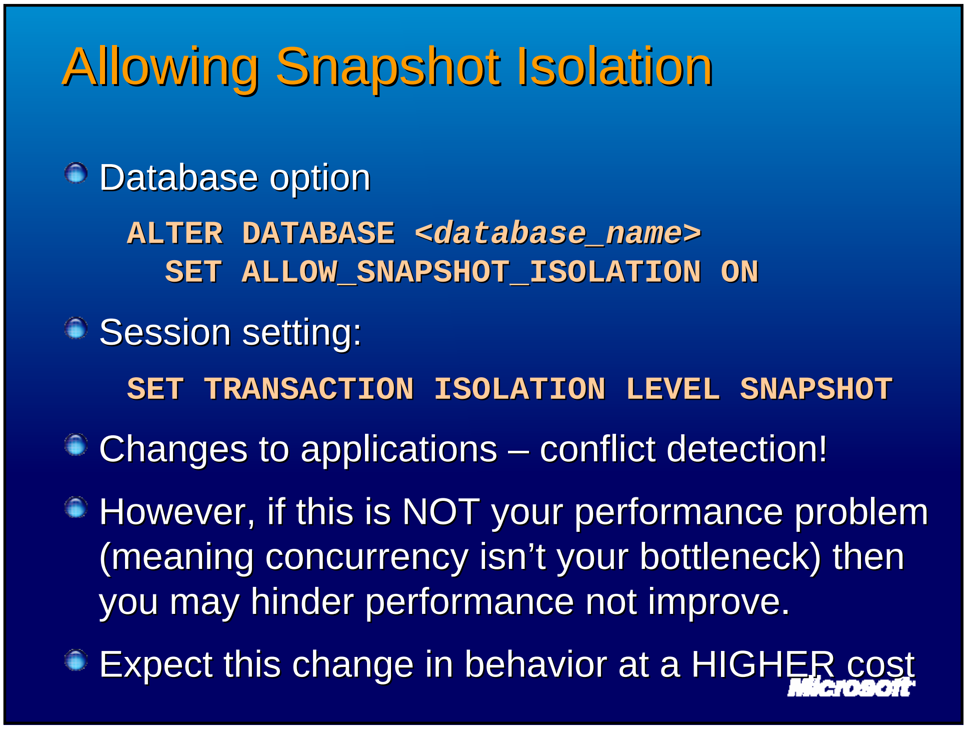


Allowing RCSI

- Database option

```
ALTER DATABASE <database_name>
   SET READ_COMMITTED_SNAPSHOT ON
   WITH ROLLBACK AFTER 5
```

- Changes to queries – none!
- Changes to applications – none!
  (caveat when you depend on locking behavior re: queues)
- Changes to blocking…
- However, if this is NOT your performance problem (meaning concurrency isn't your bottleneck) then you may hinder performance not improve.
- Expect this change in behavior at a cost



Allowing Snapshot Isolation

- Database option

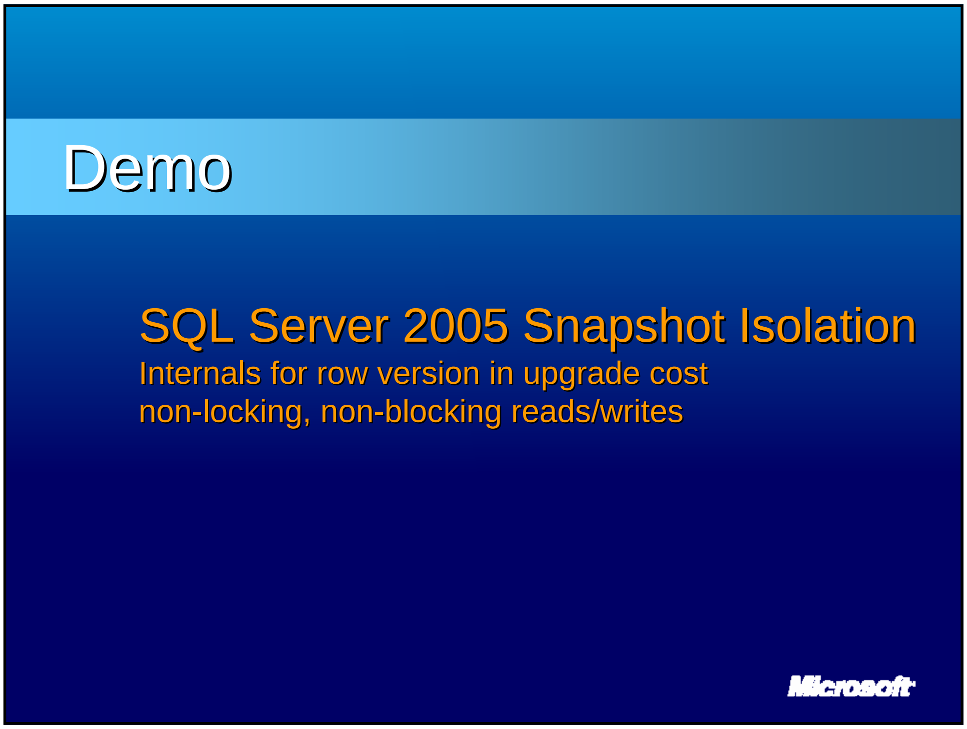
```
ALTER DATABASE <database_name>
   SET ALLOW_SNAPSHOT_ISOLATION ON
```

- Session setting:

```
SET TRANSACTION ISOLATION LEVEL SNAPSHOT
```

- Changes to applications – conflict detection!
- However, if this is NOT your performance problem (meaning concurrency isn't your bottleneck) then you may hinder performance not improve.
- Expect this change in behavior at a HIGHER cost

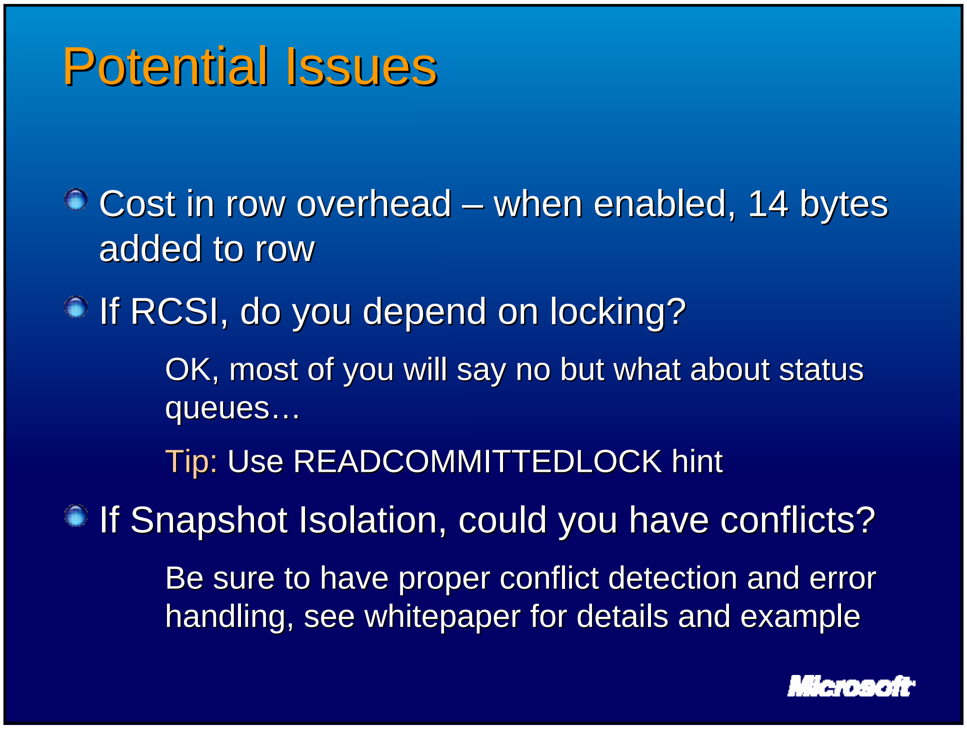
This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.



Demo

SQL Server 2005 Snapshot Isolation

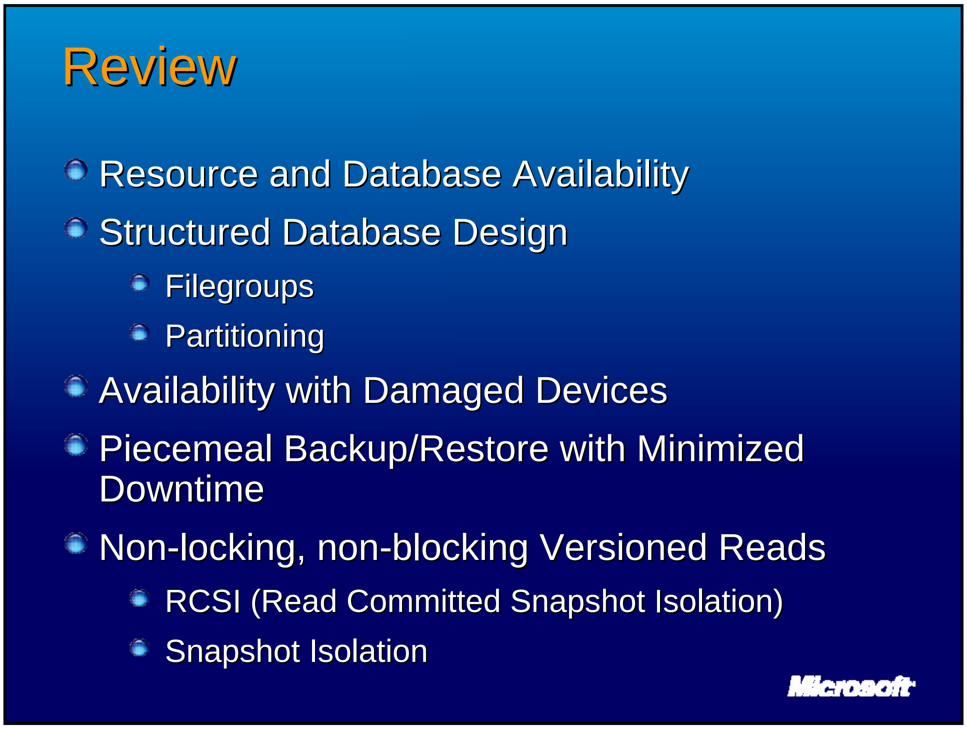
Internals for row version in upgrade cost
non-locking, non-blocking reads/writes



Potential Issues

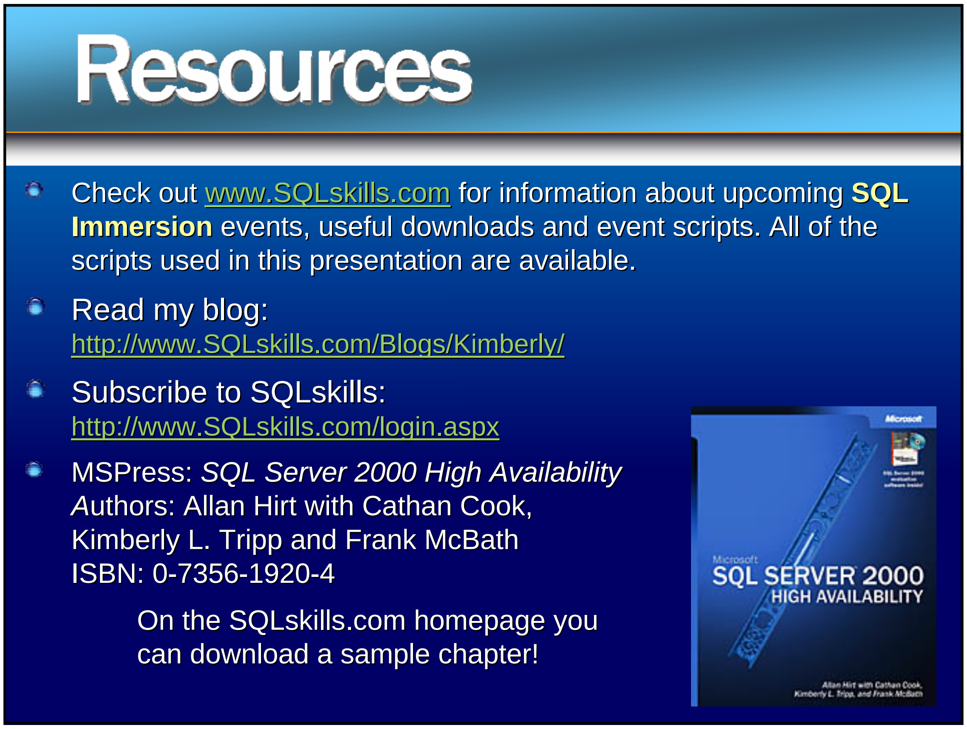
- Cost in row overhead – when enabled, 14 bytes added to row
- If RCSI, do you depend on locking?
  - OK, most of you will say no but what about status queues…
  - Tip: Use READCOMMITTEDLOCK hint
- If Snapshot Isolation, could you have conflicts?
  - Be sure to have proper conflict detection and error handling, see whitepaper for details and example

This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.



Review

- Resource and Database Availability
- Structured Database Design
  - Filegroups
  - Partitioning
- Availability with Damaged Devices
- Piecemeal Backup/Restore with Minimized Downtime
- Non-locking, non-blocking Versioned Reads
  - RCSI (Read Committed Snapshot Isolation)
  - Snapshot Isolation



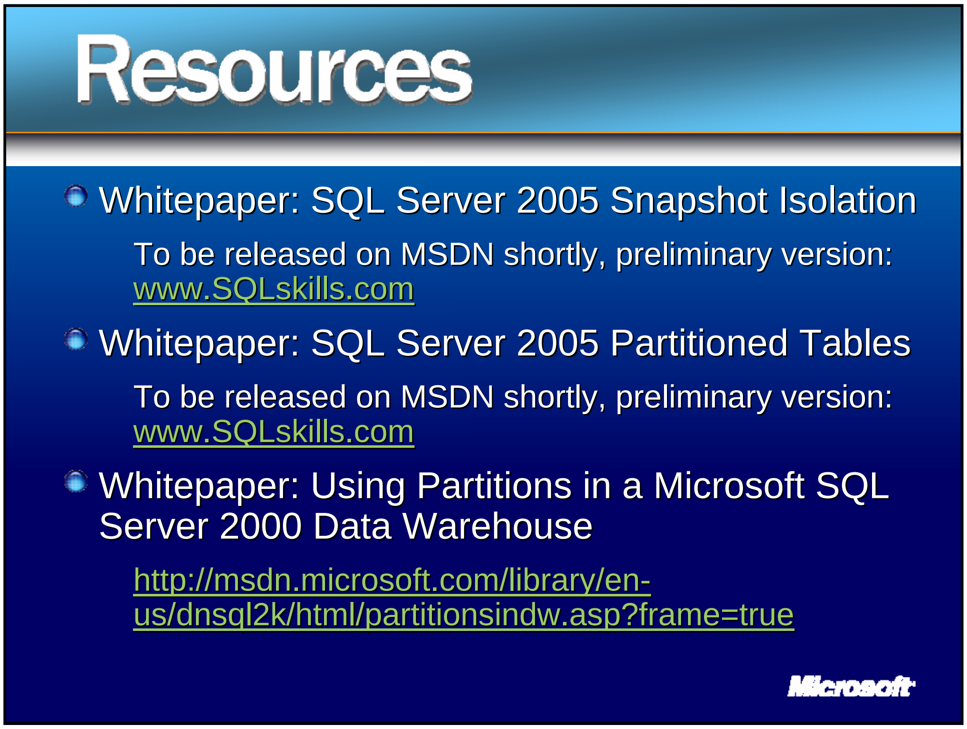
Resources

- Check out www.SQLskills.com for information about upcoming **SQL Immersion** events, useful downloads and event scripts. All of the scripts used in this presentation are available.
- Read my blog: http://www.SQLskills.com/Blogs/Kimberly/
- Subscribe to SQLskills: http://www.SQLskills.com/login.aspx
- MSPress: *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!

Microsoft
Microsoft SQL SERVER 2000
HIGH AVAILABILITY
Allan Hirt with Cathan Cook, Kimberly L. Tripp, and Frank McBath

This presentation is for informational purposes only. SYSolutions, Inc. makes no warranties, express or implied, in this summary.



Resources

- Whitepaper: SQL Server 2005 Snapshot Isolation

  To be released on MSDN shortly, preliminary version: www.SQLskills.com

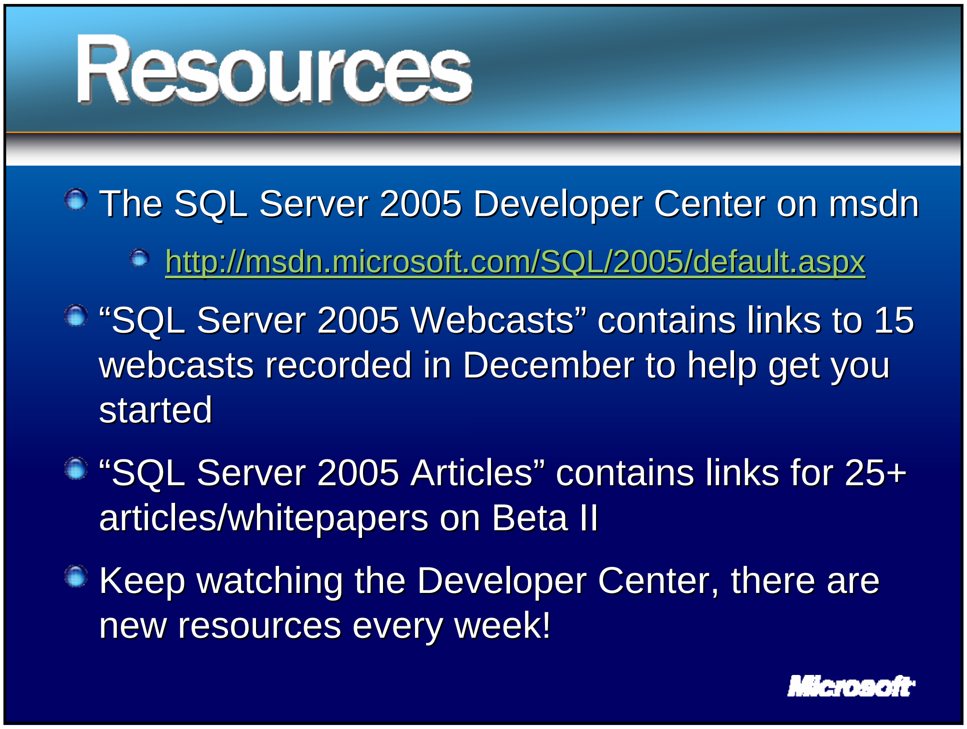
- Whitepaper: SQL Server 2005 Partitioned Tables

  To be released on MSDN shortly, preliminary version: www.SQLskills.com

- Whitepaper: Using Partitions in a Microsoft SQL Server 2000 Data Warehouse

  http://msdn.microsoft.com/library/en-us/dnsql2k/html/partitionsindw.asp?frame=true

Microsoft



Resources

- The SQL Server 2005 Developer Center on msdn
  - http://msdn.microsoft.com/SQL/2005/default.aspx
- “SQL Server 2005 Webcasts” contains links to 15 webcasts recorded in December to help get you started
- “SQL Server 2005 Articles” contains links for 25+ articles/whitepapers on Beta II
- Keep watching the Developer Center, there are new resources every week!

Microsoft

SQL skills

Please fill out your evaluation
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