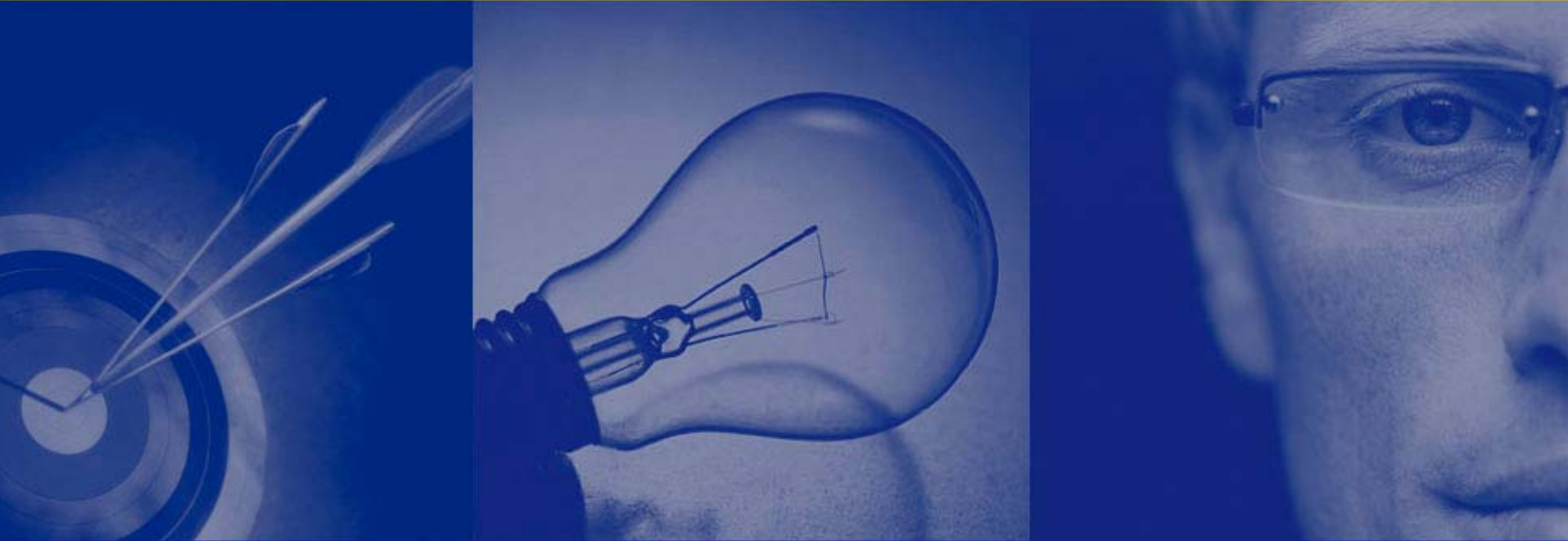


# Top 10 Most Common Mistakes on Microsoft SQL Server



Presented by Kevin Kline  
SQL Server MVP since 2004

# Agenda

- Speaker Bio
- About Quest Software
- Accolades & Awards
- The Top 10 Most Common Mistakes on Microsoft SQL Server
- Call to Action
- Q & A

# Kevin Kline



- *Technical Strategy Manager for SQL Server Solutions, Quest Software Inc.*
- Microsoft SQL Server MVP
- President of the International SQL Server User's Group – PASS
- SQL Server expert and author of the O'Reilly titles "SQL in a Nutshell" and "Transact-SQL Programming"

# About Quest Software

- Profitable & Growing
  - Over a half-billion in '06 revenues
  - Continued growth in '01 through '06 against sector trends
- Founded in 1987
- Public in 1999
  - Nasdaq: QSFT
- Customers: 18,000 Worldwide
  - Including 75 percent of the Fortune 500
- over 2,750 employees
  - Over 950 in R&D
  - Nearly a dozen R&D centers around the globe
- Headquarters: Aliso Viejo, CA

# Accolades & Awards

- TechTarget Product of the Year 2006 - LiteSpeed
- Best of Tech Ed award 2006 with Spotlight on SQL Server
- SQL Server Magazine, Platinum reader's choice award
- SQL Server Magazine Readers Choice Awards, winner in 11 categories, 2005
- No. 1 vendor in Distributed Data Management Facilities, IDC, 2005
- Microsoft TechEd Europe Best of Show Winner, 2005
- No. 1 vendor in Application Management Software, Gartner Dataquest, 2005
- Jolt Productivity Award Winner
- Network Computing Editor's Choice Winner
- No. 8 in the "Who's Who in Enterprise Software," Investor's Business Daily



# The Top 10 List

- Includes issues covering:
  - Database Design
  - Database Administration
  - Application Development
- Includes an explanation for each item on the list
- Includes resources to alleviate the mistake

# Placeholder for Polling Question #1

- How many servers are you currently responsible for managing:
  - 1-20
  - 21-50
  - 51-70
  - 71-100
  - More than 100 servers

## Placeholder for Polling Question #2

- In your day-to-day activities, which of the following require the most time and attention cause you the most aggravation:
  - Backup / Recovery / Replication
  - Finding and fixing performance problems
  - Managing database changes
  - Evaluating resource consumption / planning for future requirements
  - Tuning and optimizing your SQL Server environment
  - Job and Schedule maintenance
  - Performing other routine maintenance (i.e. deaggregating indexes, writing stored procedures, physical server maintenance)

## Alternate way to ask Polling Question #2

- In your day-to-day activities, which of the following causes you the most aggravation:
  - Backup / Recovery / Replication
  - Finding and fixing performance problems
  - Managing database changes
  - Evaluating resource consumption / planning for future requirements
  - Tuning and optimizing your SQL Server environment
  - Job and Schedule maintenance
  - Performing other routine maintenance (i.e. deaggregating indexes, writing stored procedures, physical server maintenance)

## 10. Unknown Scalability Requirements

- DBAs and Developers must understand the scalability requirements of their application and database design. Without it, the following problems occur:
  - Inadequate development testing
    - Often times, requirements don't include performance goals
    - Even more often, requirements don't include scalability goals
  - Performance shortcomings
  - Additional QA needed
    - Often, no or very little time to fix findings of QA process
  - What about the future?
    - No capacity planning or forecasting

## 9. Improperly Normalized Tables

- DBAs and Developers must perform a thorough database design phase. Without it, the following problems occur:
  - Duplicate data
  - Normalization performance anomalies, such as the classic Insert, Update, and Delete anomalies
  - No or poor indexing
  - No or poor key relationships
  - Bad or non-existent naming standards
  - Missed opportunities to denormalize

## 8. Indexing Issues

- DBAs and Developers must do a thorough job indexing their database. Without it, the following problems occur:
  - Too many page splits
    - Alleviated by using fill factor and sometimes pad\_indexes
  - Poor non-clustered indexing
    - Doesn't help with SQL transactions
    - Not applied to foreign key fields
    - Missed opportunity for covering indexes
    - Too many columns on composite indexes
  - Non-existent or bad clustered index
  - Non-existent or poorly selective primary key
- Many times, DBAs and Developers forget to update statistics on indexes

## 7. No Baselines or Benchmarks

- DBAs and Developers frequently have no idea what their server load is, nor can they tell you what it's peak load might be. Without this knowledge, the following problems occur:
  - Manage the user experience solely by call volume
  - Cannot meet SLAs
  - Cannot predict behavior
    - Will it be busy at the end of the week, month, or quarter?
    - Will it be busy after lunch? After a major holiday?
  - Don't know if where to start fixing problems:
    - Throw more hardware at it?
    - Tune SQL/app code?
  - May add additional apps/databases to a server that cannot sustain the combined load

## 6. Disks – Thinking Space but not IO

- DBAs and Developers frequently think about their disk subsystems only in terms of disk space not IO load. Without this knowledge, the following problems occur:
  - Inadequate fault tolerance
  - Insufficient IO:
    - OLTP requires high transactions/sec
    - OLAP requires high MB transfers/sec
  - Poor choice of RAID type
  - Not enough disk spindles
  - Poor use of controllers and/or channels
  - Let the SAN administrator tell them what they get

## 5. Error Notification

- DBAs and Developers don't take advantage of easy to use error notification built into SQL Server.

Without this feature, the following problems occur:

- Important schedule jobs fail without intervention
- Must rely upon end-users to notify of problems:
  - Squeaky wheel, but not most important problem, gets the grease
  - Miss problems that occur when users not on the system
- Valuable diagnostic and troubleshooting time wasted
- Must manually check all managed servers
- Impossible to be truly proactive

## 4. Backups

- DBAs and Developers do not test backups nor recoveries as they should. Without this activity, the following problems occur:
  - Not certain that backups are good:
    - Verified
    - Available
  - Not certain where to get all data, files, DLLs, etc for backup
  - Haven't tested a full, ground-up restore:
    - What if you have to reinstall everything?
    - The importance of recovery: the **Lost Job** scenario
  - Cannot actually perform a full restore on archived data

# 3. Query Tuning

- DBAs and Developers don't know how to tune a query. Without this proficiency, the following problems occur:
  - Don't know which tools to use:
    - PerfMon
    - SQL Profiler
    - Graphic Showplan and SET SHOWPLAN
    - SET STATISTICS IO
  - Don't know how to read a query plan:
    - Why is an index seek better than a table scan?
    - Why are logical reads better than physical reads?
  - Cannot suggest or try alternative plans that improve performance
  - Application will not scale beyond a few users
  - May spend more money on hardware to fix an “eternal” problem

## 2. Ignorance of Internals

- DBAs and Developers don't know the basics of the SQL Server internals, such as the storage engine and query engine. Without this knowledge, the following problems occur:
  - Write terrible queries and then place into production
    - Too much IO
    - Too much locking and/or deadlocking
    - Missed opportunities for indexes
  - Don't realize impact of *checkpointing* and *lazywriter*
    - Cause false alarms when checking performance
  - Miss opportunities to read from cache
    - E.G. Force recompiles of stored procedures
  - Miss the chance to utilize partitions and file group distribution for better IO

# 1. Cursors

- Ok for small, one-off applications with very limited number of rows
- Deprecated by Microsoft. Try to find it in SQL2005 documentation!
- DBAs and Developers don't know the basics of set theory versus row theory. Without this knowledge, the following problems occur:
  - Locking problems
  - Memory problems
    - Remember to deallocate those cursors
  - Iterations can take a long time

## Polling Question #3

- What best describes the type of environment for which you're responsible?
  - Production environment supporting business critical apps
  - Production environment supporting non-critical apps
  - Pre-Production environment (Test/Development)
  - All of the above



# Quest Solutions

- 10. Unknown Scalability Requirements
  - Capacity Manager for SQL Server
- 9. Improperly Normalized Tables
  - Toad Data Modeler
- 8. Indexing Issues
  - Toad Xpert for SQL Server
  - Performance Analysis for SQL Server
- 7. No Baselines or Benchmarks
  - Capacity Manager for SQL Server
  - Benchmark Factory
- 6. Disks – Thinking Space but not IO
  - Capacity Manager
  - Spotlight on SQL Server Enterprise
  - Performance Analysis for SQL Server

# Quest Solutions

- 5. Error Notification
  - Spotlight on SQL Server Enterprise
- 4. Backups
  - LiteSpeed for SQL Server
- 3. Query Tuning
  - Toad Xpert for SQL Server
  - Performance Analysis for SQL Server
  - Coefficient
- 2. Ignorance of Internals
  - Spotlight on SQL Server Enterprise
  - Performance Analysis for SQL Server
- 1. Cursors
  - Spotlight on SQL Server Enterprise
  - Performance Analysis for SQL Server

# Resources

- See my other webcasts on YouTube. Search for “Kevin Kline SQL”.
- Refer to my white papers at [www.quest.com/whitepapers](http://www.quest.com/whitepapers).
- Check out the fantastic webcasts at TechNet and MSDN:
  - <http://www.microsoft.com/technet/community/en-us/sql/webcasts.mspx>
  - <http://msdn2.microsoft.com/en-us/sql/default.aspx>
- Refer to learning resources at [www.sqlpass.org](http://www.sqlpass.org).



## Questions & Answers

Thank you for attending this session and the

More questions? Email me at [kevin.kline@quest.com](mailto:kevin.kline@quest.com)