

# 10 Things That Every DBA Should Know!

**John Morehouse**

Database Architect

Farm Credit Mid-America, Louisville, Kentucky



john@jmorehouse.com



@SqlRUs



<http://linkedin.com/in/sqlrus>



<http://www.sqlrus.com>

# Agenda

- Know Your Data!
- SELECT \*
- SET Based
- Auto Shrink / Auto Growth
- Transaction Isolation Levels
- Index Basics
- Backups & Restores
- Basic Monitoring
- Security
- Help Me!!



# Know Your Data!

## Pop Quiz!

You are tasked with adding a new column that stores an integer value, which the value will not be larger than **200**. What do you do hot shot?



# Know Your Data!

You are tasked with adding a new column that stores an integer value, which the value will not be larger than **200**. What do you do hot shot?

We see this:

```
8 ALTER TABLE dbo.Beaker  
9      ADD [theNumber] INT  
10
```

# Know Your Data!

You are tasked with adding a new column that stores an integer value, which the value will not be larger than **200**. What do you do hot shot?

We see this:

```
8 ALTER TABLE dbo.Beaker  
9     ADD [theNumber] INT  
10
```

We want this:

```
8 ALTER TABLE dbo.Beaker  
9     ADD [theNumber] TINYINT
```

# Know Your Data!

You are tasked with adding a new column that stores an integer value, which the value will not be larger than **200**. What do you do hot shot?

We see this:

```
8 ALTER TABLE dbo.Beaker  
9     ADD [theNumber] INT  
10
```

We want this:

```
8 ALTER TABLE dbo.Beaker  
9     ADD [theNumber] TINYINT
```



# Know Your Data!

## Space!!

Data type	Range	Storage
<b>bigint</b>	-2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807)	8 Bytes
<b>int</b>	-2^31 (-2,147,483,648) to 2^31-1 (2,147,483,647)	4 Bytes
<b>smallint</b>	-2^15 (-32,768) to 2^15-1 (32,767)	2 Bytes
<b>tinyint</b>	0 to 255	1 Byte

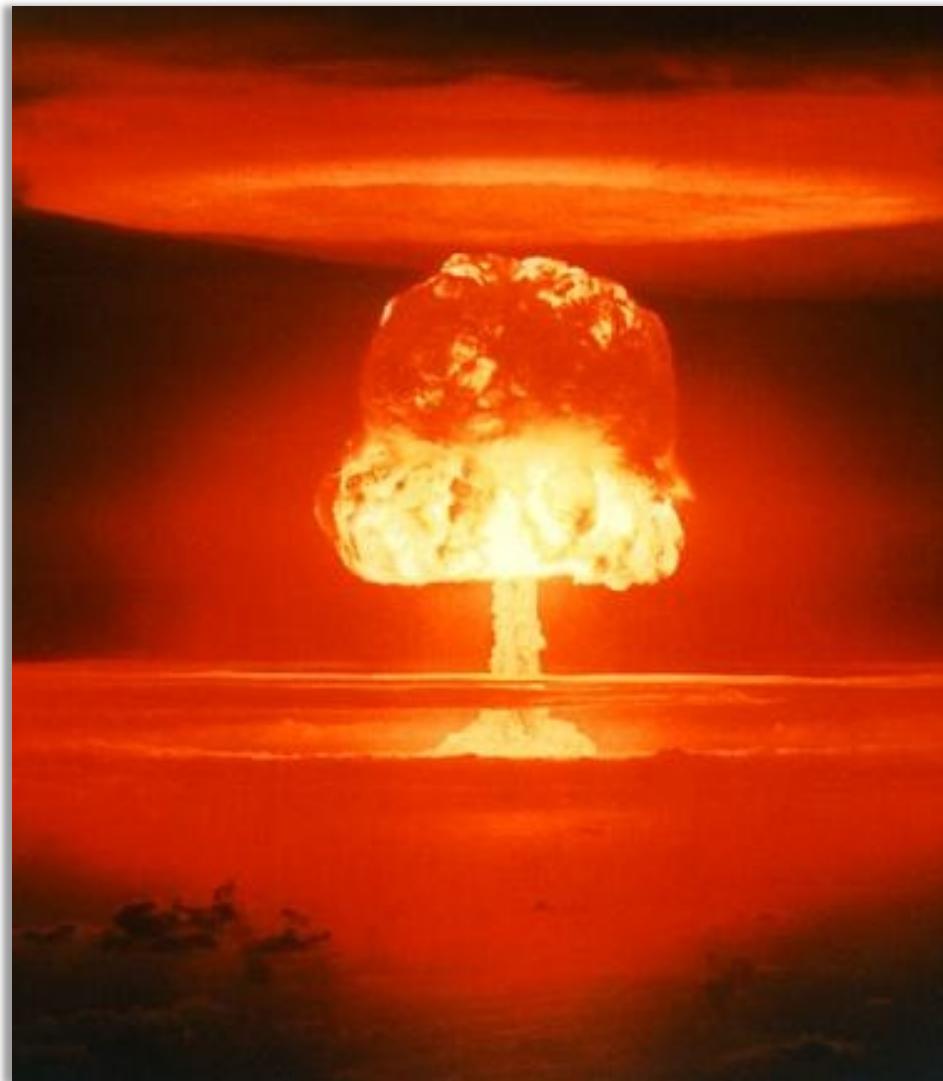
- Two usual culprits
  - NVARCHAR versus VARCHAR
  - INT versus TINYINT



# **Know Your Data!**

- **How do you choose?**
  - Think about the amount of space you want to use
  - It Depends!! (Standard DBA Answer)
- **The obvious**
  - Use string data types for string values
  - Use numeric data types for numbers
  - Exceptions (zip codes)
- **What to watch for**
  - UNIQUEIDENTIFIER (index killer)
  - DATE/DATETIME (watch your precision)

# **SELECT \***



# **SELECT \***

## **Why?**

- All the columns get sent across the wire
- Increased Memory Utilization
- Increased IO
- LAZY! (but we all do it!)
- DEMO



# SET Based Queries

- **Why do SET based?**
  - Simple. Faster performance.
- **Row By Agonizing Row (Bad RBAR!)**
  - Slower performance, more over head
  - CURSOR's not a good idea (it depends)
  - WHILE loops not a good idea (it depends)
- **Let SQL Server manage the relationships**
- **DEMO**



# Auto Shrink / Auto Growth

- **Auto Shrink**
  - Increases index fragmentation
  - Can't schedule it
- **Auto Growth**
  - Use specific space rather than percentage
  - Leave this on
- **Do not use Auto Shrink & Auto Growth at the same time!!**



# Transaction Isolation

**Q:** What is the default transaction isolation level for SQL Server?



# Transaction Isolation

**Q:** What is the default transaction isolation level for SQL Server?

**A: READ COMMITTED**

# Transaction Isolation

**Q:** What is the default transaction isolation level for SQL Server?

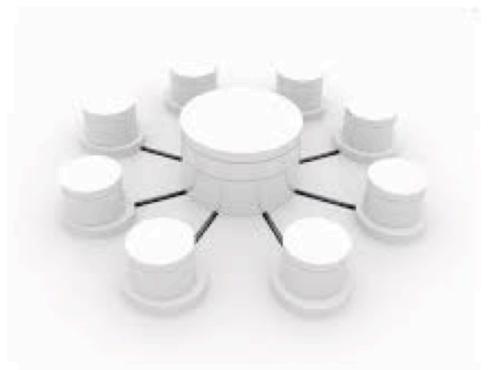
**A: READ COMMITTED**



# Transaction Isolation

## What is it?

- The various isolation levels tell SQL Server how “isolated” a **read** transaction(s) needs to be from other transactions
- **5 Types of isolation levels**
  - READUNCOMMITTED\*
  - READ COMMITTED\*
  - REPEATABLE READ
  - SNAPSHOT
  - SERIALIZABLE



# Transaction Isolation

- **READ COMMITTED**
  - Read only data that has been committed to the database
  - SQL Server Default
- **READ UNCOMMITTED**
  - Read data even if that data has been changed but not yet committed to the database
  - (NOLOCK) = READ UNCOMMITTED
- **DEMO**



# Transaction Isolation

## What's Wrong with this statement?

SET TRANSACTION ISOLATION LEVEL

READ COMMITTED

SELECT

[Customer\_ID],

[CustomerName],

FROM

[dbo].[Table1] (NOLOCK)

WHERE

[Customer\_ID] = @Customer\_ID



# Transaction Isolation

## What's Wrong with this statement?

```
SET TRANSACTION ISOLATION LEVEL  
READ COMMITTED
```

```
SELECT  
    [Customer_ID],  
    [CustomerName],  
FROM  
    [dbo].[Table1] (NOLOCK)  
WHERE  
    [Customer_ID] = @Customer_ID
```



# Indexes

- **Why do we use indexes?**
  - To find the data faster
- **What do I need to know?**
  - Indexes are partial copies of the data, so reuse when possible
  - Clustered vs Nonclustered
  - Know what indexes your queries are using
- **How do you see that?**



# Indexes

## Query Plans!!

SQLQuery1.sql...rehouse (55)\*

Query executed successfully.

Editor Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
select * from Person.Contact
```

SELECT  
Cost: 0 %      Clustered Index Scan (Clustered)  
[Contact].[PK\_Contact\_ContactID]  
Cost: 100 %

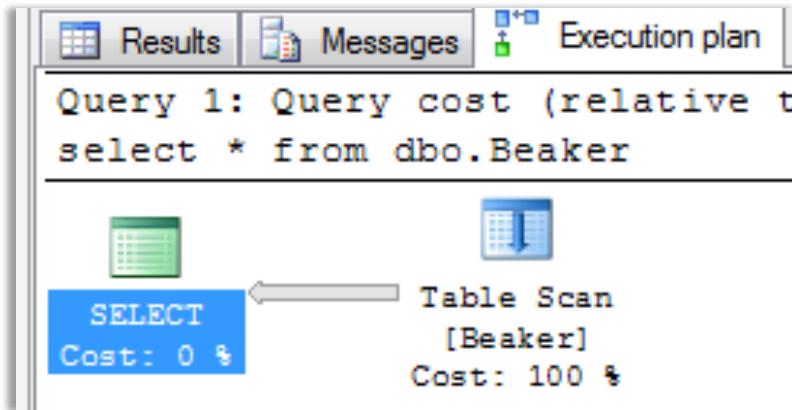
# Indexes

- **What are they?**
  - The plan that SQL Server uses to determine the best way to return the results for your query
- **What do you look for?**
  - Table Scans
  - Index Scan
  - Any type of bookmark (RID) look up / key look up
  - Index Seeks
- **Anything other than a **SEEK**, adjust the query/indexes to obtain a seek if possible**

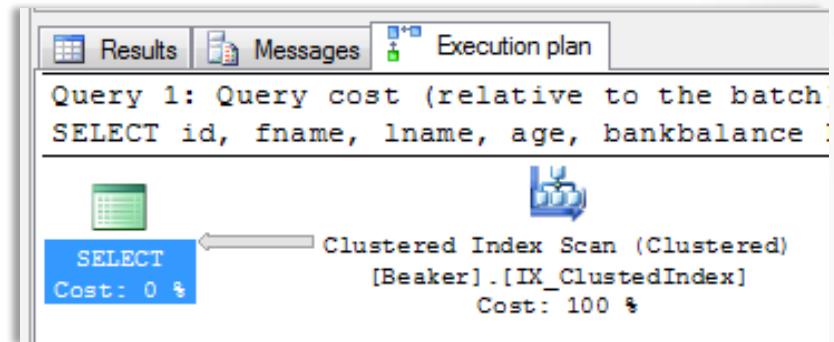


# Indexes – What to Watch For

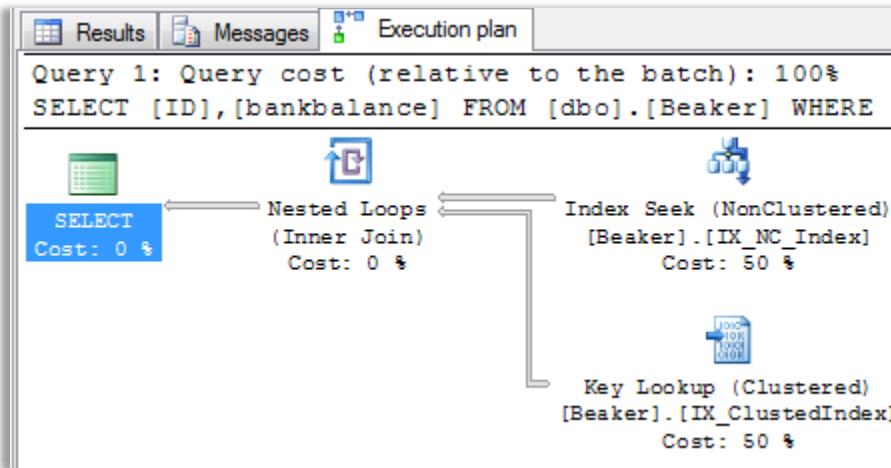
## Table Scans (Bad)



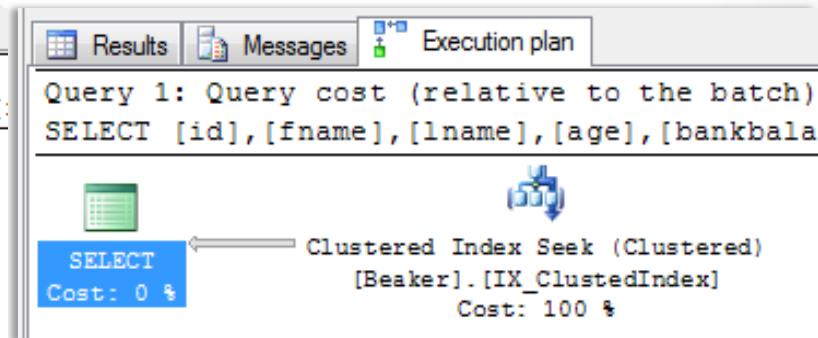
## Index Scans (Better)



## Key Lookup (Getting Better)



## Index Seeks (Best)



# Backup & Restores

## Fill In The Blank

A DBA is only as good as their last

---



# Backup & Restores

## Fill In The Blank

A DBA is only as good as their last  
**RESTORE.**



# Backup & Restores

- Recovery Models
  - SIMPLE\*
  - FULL\*
  - BULK-LOGGED



Recovery model	Description	Work loss exposure	Recover to point in time?
Simple	No log backups.  Automatically reclaims log space to keep space requirements small, essentially eliminating the need to manage the transaction log space.	Changes since the most recent backup are unprotected. In the event of a disaster, those changes must be redone.	Can recover only to the end of a backup.
Full	Requires log backups.  No work is lost due to a lost or damaged data file.  Can recover to an arbitrary point in time (for example, prior to application or user error).	Normally none.  If the tail of the log is damaged, changes since the most recent log backup must be redone. For more information, see <a href="#">Tail-Log Backups</a> .	Can recover to a specific point in time, assuming that your backups are complete up to that point in time. For more information, see <a href="#">Restoring a Database to a Point Within a Backup</a> .

Source: <http://msdn.microsoft.com/en-us/library/ms189275.aspx>

# Backup & Restores

- Types of Backups
  - **Full**
    - Enables a Point in Time restore (w/Tran Log Backups)
  - **Differential**
    - Everything since the last full backup
    - Can be done in the SIMPLE recovery model
  - **Transaction Log**
    - Backs up the transaction log
    - Enables point in time restores



# Backups & Restores

- **RECOVERY / NORECOVERY**
  - Understand the terms
- **Practice Your Restores**
  - Backups are only as good as the last **restore**
  - Practice! Practice!! Did I say practice?
- **DBCC CHECKDB!!!**
- **DEMO**



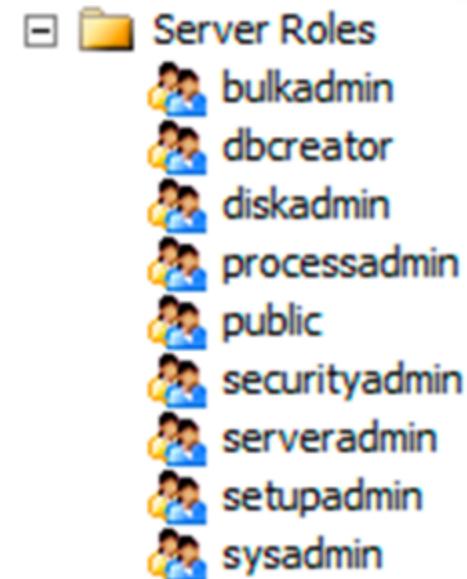
# Monitoring

- **Know the tools available**
  - They are FREE!!!
  - Profiler
  - Server Side Traces
  - Extended Events (above tools will be going away)
- **SQL Server Logs**
  - Server Logs
  - Error Logs
- **Standard Reports**



# Security

- **Logins are Mapped to Users**
  - Users are Mapped to Databases
- **SQL Server User versus Domain User**
  - Use Domain Groups (even better)
- **Database/Server Roles**
  - Use them wisely!
- **Be wary of ‘sysadmin’ & ‘securityadmin’**



# Asking For Help

- Sometimes you get stuck. Accept it.
- Books Online
- SQLServerCentral.com
- Stackoverflow.com
- Twitter (#sqlhelp)
  - Great's like:
    - Paul Randal
    - Kimberly Tripp
    - Brent Ozar
  - Google/Bing
- Blogs



# Summary

- Try to use the right data type, know thy data
- Watch your “Select \*’s”
- Use SET Based for everybody’s sanity
- Don’t use Auto shrink. Period
- Watch for the correct isolation level
- Know your indexes & query plans!
- Know your backups!
- Secure everything
- Know where to ask for help

# Resources

- **Idera Virtual Database:**

<http://www.idera.com/SQL-toolbox/SQL-virtual-database/>

- **Indexes/Query Plans:** <http://www.scarydba.com/resources/>

- **Data Types:**

[http://technet.microsoft.com/en-us/library/ms187752\(v=sql.100\).aspx](http://technet.microsoft.com/en-us/library/ms187752(v=sql.100).aspx)

- **Isolation Levels:**

[http://technet.microsoft.com/en-us/library/ms189122\(v=sql.105\).aspx](http://technet.microsoft.com/en-us/library/ms189122(v=sql.105).aspx)

- **Backups:**

<http://technet.microsoft.com/en-us/library/ms175477.aspx>

- **Autoshrink:**

<http://www.sqlskills.com/blogs/paul/auto-shrink-turn-it-off/>

**Please take a moment to  
thank all of the sponsors,  
speakers & volunteers  
who make these events  
happen!!**

# THANK YOU!!!!

**John Morehouse**



john@jmorehouse.com



@SqlRUs



<http://linkedin.com/in/sqlrus>



<http://www.sqlrus.com>