

SQL Server Internals

Gianluca Sartori

@spaghettidba



SQL Server Internals

Sponsors



November 26°, 2016

Organizers



UNIVERSITÀ DEGLI STUDI DI PARMA

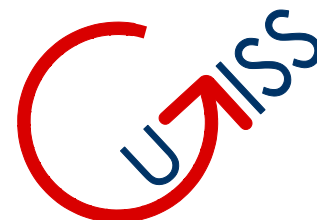


getlatestversion.it



dotNET {podcast}

ENGAGE
IT SERVICES



dan|ela
ma|visi
COMMUNICATION



#sqlsatParma
#sqlsat566

November 26°, 2016

Gianluca Sartori

Independent SQL Server consultant

SQL Server MVP, MCTS, MCITP, MCT

Works with SQL Server since version 7

DBA @ Scuderia Ferrari

Blog: spaghettidba.com

Twitter: [@spaghettidba](https://twitter.com/spaghettidba)



SQL Server Infernals

Agenda

- Best practices or Worst practices?
- What can go wrong?
 - Design
 - Development
 - Installation
 - Administration



Disclaimer:

- Not everything is black or white
- «It depends» is the most likely answer

There are edge cases when some of these worst practices are the only possible solution, or not such a bad idea...

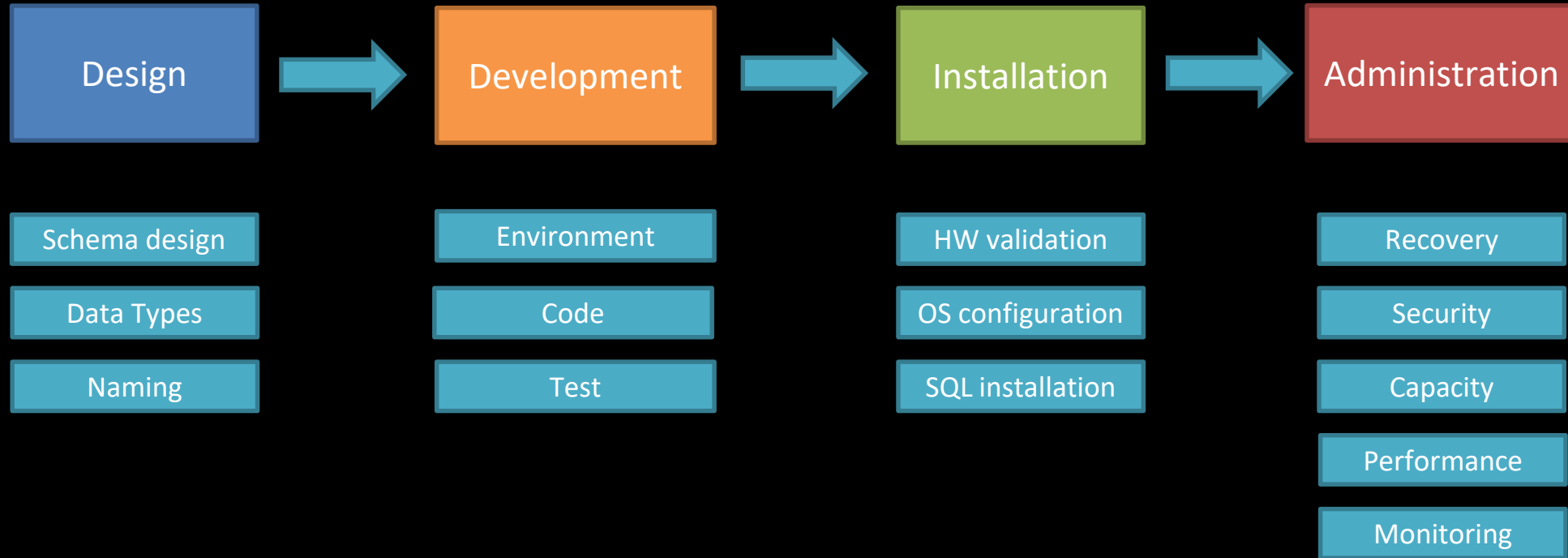


Best Practices vs. Worst Practices

- Why Best Practices are not enough
 - Too many
 - No time
 - Lack of experience
 - Not clear what happens if we don't follow them
- Why Worst Practices help
 - They show the mistakes to avoid
 - We can learn from someone else's mistakes



Worst Practices Areas



SQL Server Internals

- Worst Practices are sins that will put you in the SQL Server hell!!
- I will guide you through the circles, as Virgil did with Dante



CIRCLE 1:

Undernormalizers



SQL Server Internals

Schema Design

- Not normalizing the schema
 - 1NF:
A primary key, atomic attributes only
 - 2NF:
Every attribute depends on **the whole** key
 - 3NF:
Every attribute depends **only** on the key
*«The key, the whole key, nothing but the key,
so help me Codd»*



Clues of denormalization

- Repeating data ← *redundancies*
- Inconsistent data between tables ← *anomalies*
- Data separated by «,»
 - *Ex: john@gmail.com, john@business.com*
- Structured data in «notes» columns
- Columns with a numeric suffix
 - *Ex: Zone1, Zone2, Zone3 ...*



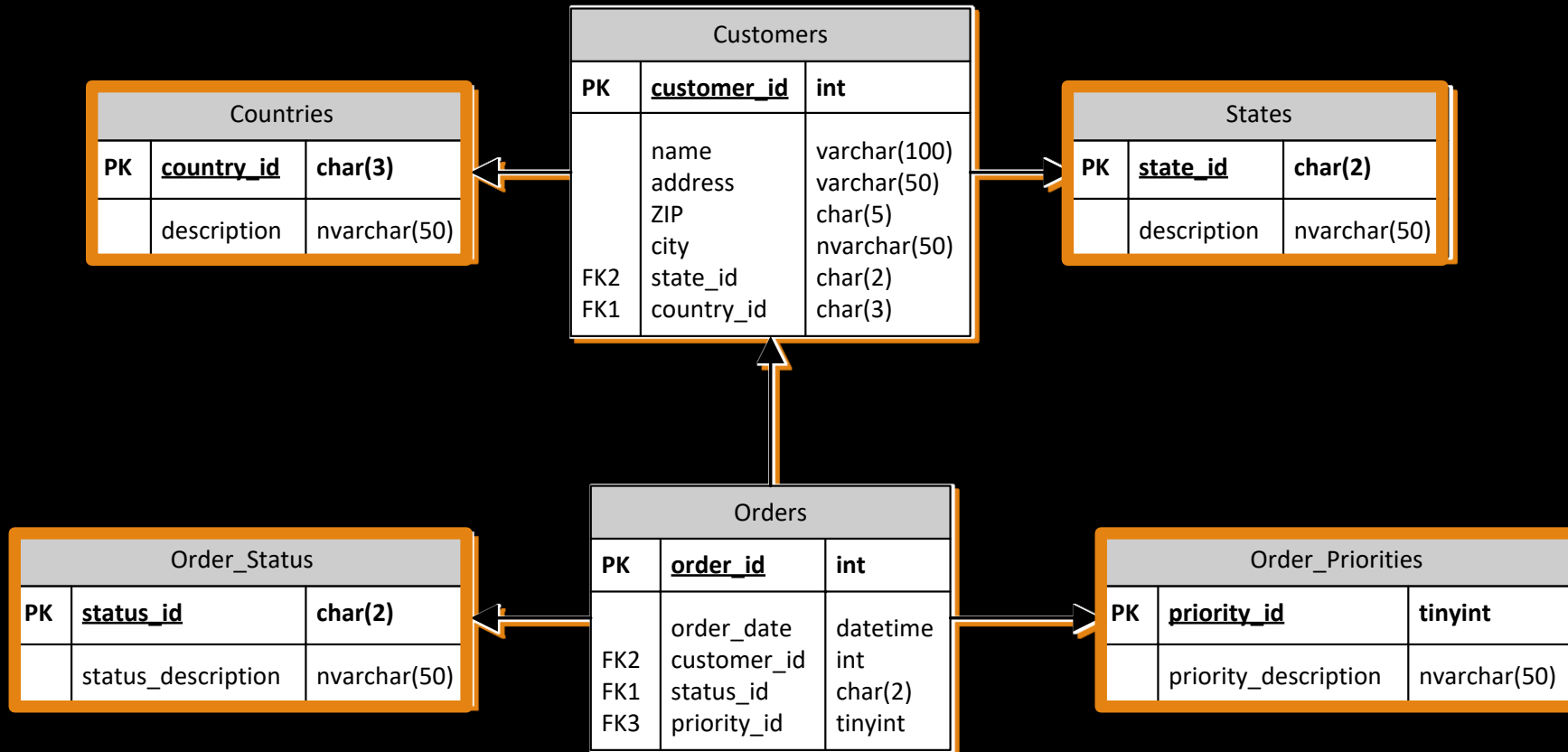
CIRCLE 2:

Generalizers



SQL Server Internals

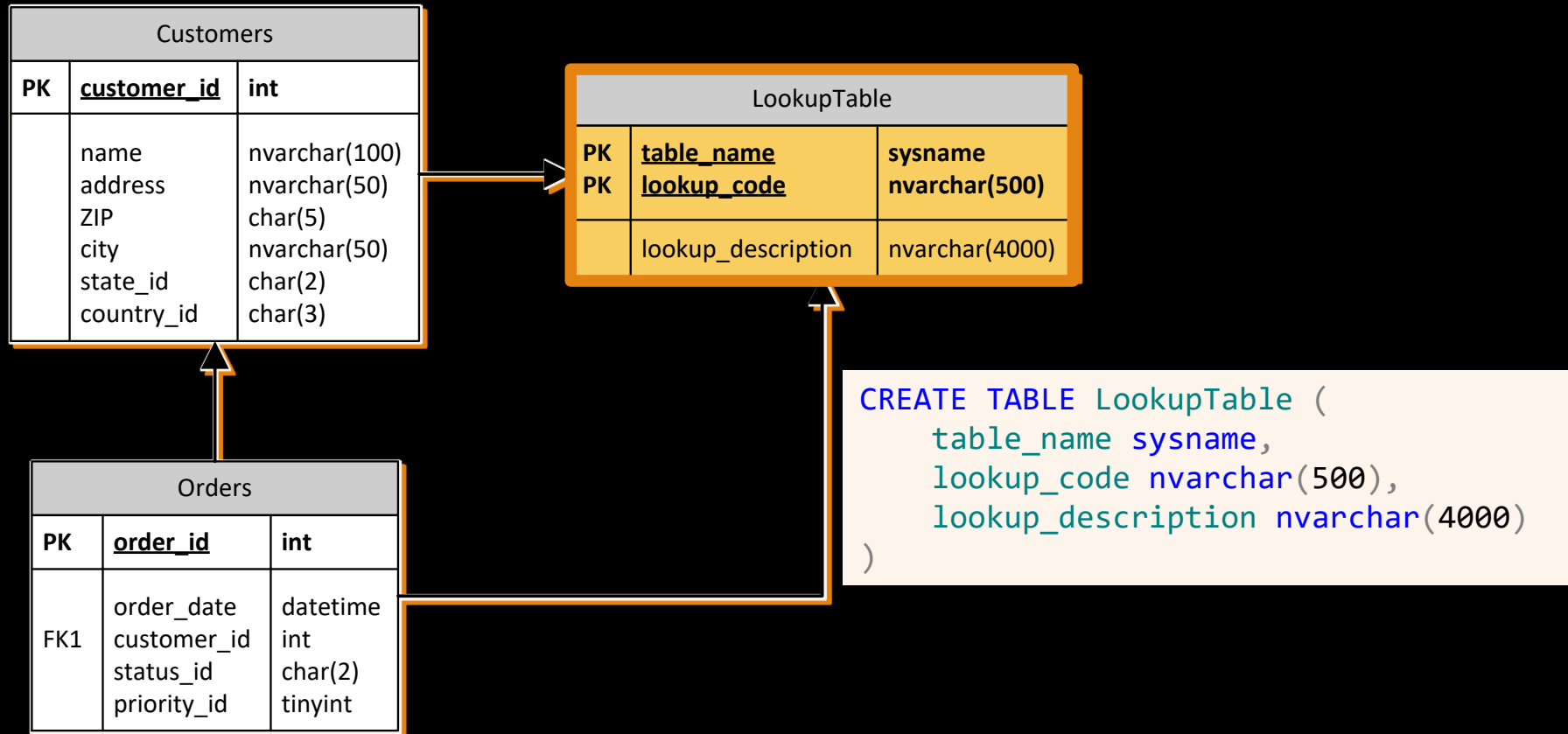
Lookup Tables



One lookup table for each attribute



OTLT: One True Lookup Table



One lookup table for all attributes

OTLT: One True Lookup Table

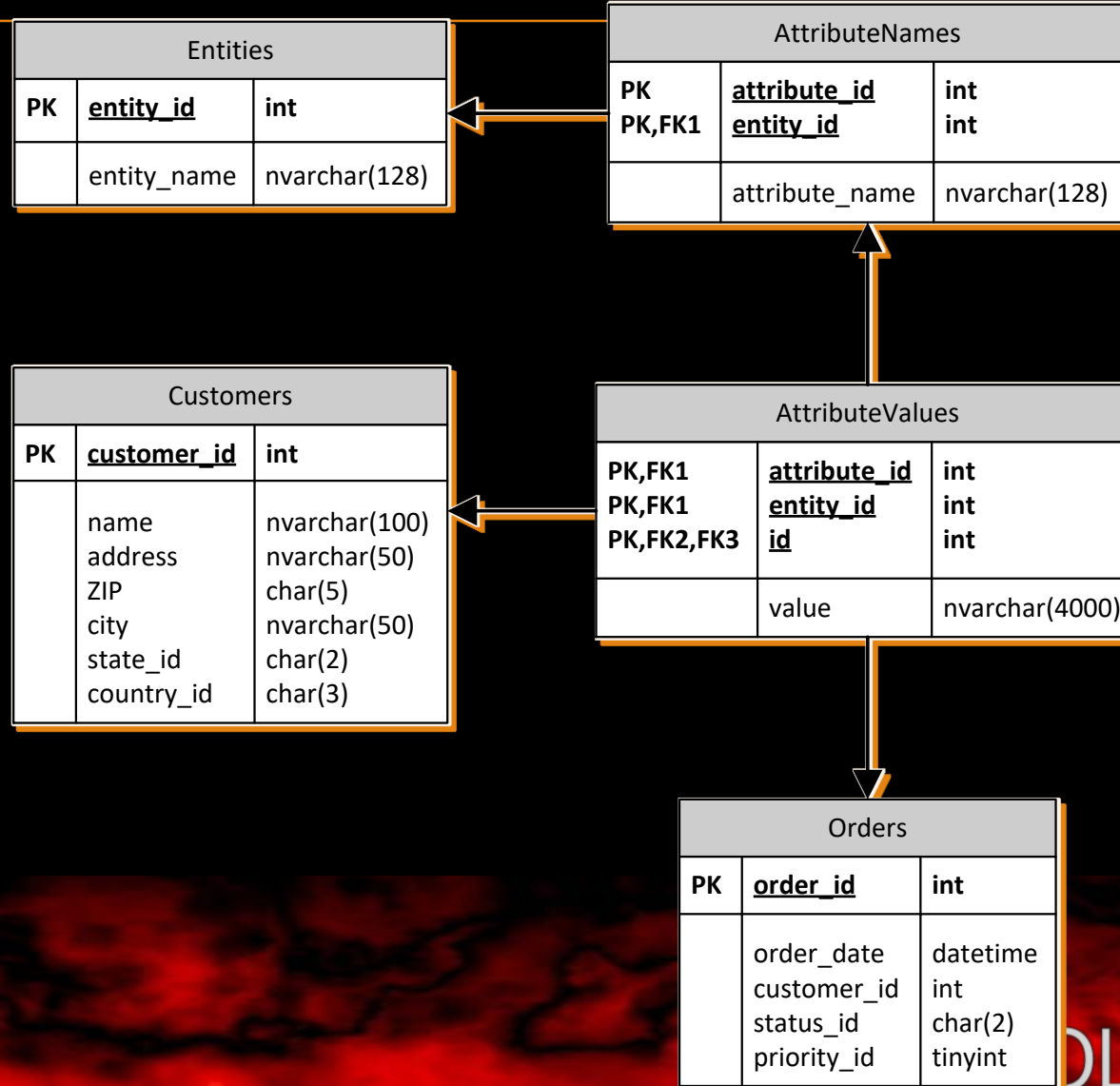
- No Foreign Keys
- Generic data types → nvarchar(SomeHighNumber)
Implicit Conversions, Incorrect Data, Huge memory grants...
- CHECK constraints may help to a point...

```
CHECK(  
    CASE  
        WHEN lookup_code = 'states'      AND lookup_code LIKE '[A-Z][A-Z]'      THEN 1  
        WHEN lookup_code = 'priorities'  AND lookup_code LIKE '[0-9]'          THEN 1  
        WHEN lookup_code = 'countries'   AND lookup_code LIKE '[0-9][0-9][0-9]'   THEN 1  
        WHEN lookup_code = 'status'      AND lookup_code LIKE '[A-Z][A-Z]'      THEN 1  
        ELSE 0  
    END = 1  
)
```

- Locking



EAV: Entity, Attribute, Value



EAV: Entity, Attribute, Value



Disadvantages:

- Generic data types → Ex: `varchar(4000)`
- No Foreign Keys
- No CHECK constraints
- Multiple accesses to the same table
 - *One access per attribute*

Advantages

- Dynamic schema: no need to alter the database
 - *Replication, distributed environments*



EAV: Entity, Attribute, Value

- Reporting is crazily hard.
- Writing to the EAV schema is a mess
- Workaround:
 - Reads: PIVOT / Crosstab
 - Writes: View + INSTEAD OF triggers
- Alternatives:
 - SPARSE columns
 - XML/JSON
 - Key-value store databases
 - Document-oriented databases



DEMO:

EAV Design



SQL Server Infernals

CIRCLE 3:

Shaky Typers



SQL Server Internals

Data type Worst Practices

- Numeric data types for non-numeric data
- Storing data as their human-readable representation
- Using deprecated data types
- Using larger data types “just in case”
- Using variable length data types for fixed size data
- Storing durations in date/datetime columns
- Getting Unicode wrong
- Using different data types for the same data in different tables





CIRCLE 4:

Anarchic Designers



SQL Server Internals

Chaos Belongs to Hell

- No Primary Key o surrogate keys only
«identity» is not the only possible key!
- No Foreign Keys
They're «awkward»
- No CHECK constraint
The application will guarantee consistency...
- Wrong data types
 - Data type is the 1° constraint on the data
- Use of NULL where not necessary
- Use of «dummy» data (ex: '.', 0)



The background is a dark, monochromatic illustration in shades of brown and black. It depicts a figure, possibly a prophet or a personification of a concept, standing on a rocky outcrop. The figure is wearing a long, flowing robe and has their back to the viewer, looking out over a vast, chaotic landscape. The landscape is filled with intricate, swirling patterns that resemble smoke, fire, or perhaps a complex network of roots and vines. The overall mood is somber and mysterious.

CIRCLE 5:

Inconsistent Baptists



SQL Server Internals

Damnation by Namification

- Hungarian Notation (AKA «tibbing»)
- Insanely short names
- Insanely long names
- Mixing languages
- Using the «sp_» prefix
- Using reserved words or illegal characters
- Using system generated constraint names
- No naming convention or multiple naming conventions



Hungary is a nice str_country



SQL Server Infernals



CIRCLE 6:

Environment Pollutors



SQL Server Internals

Pollutors will be prosecuted

INSTANT
DAMNATION!

- Developing in production
- Using the test environment for development
- Using a shared database for development
- No source control
- Developing with sysadmin privileges
- Developing on a different version/edition from production





CIRCLE 7:

Overly Optimistic Testers

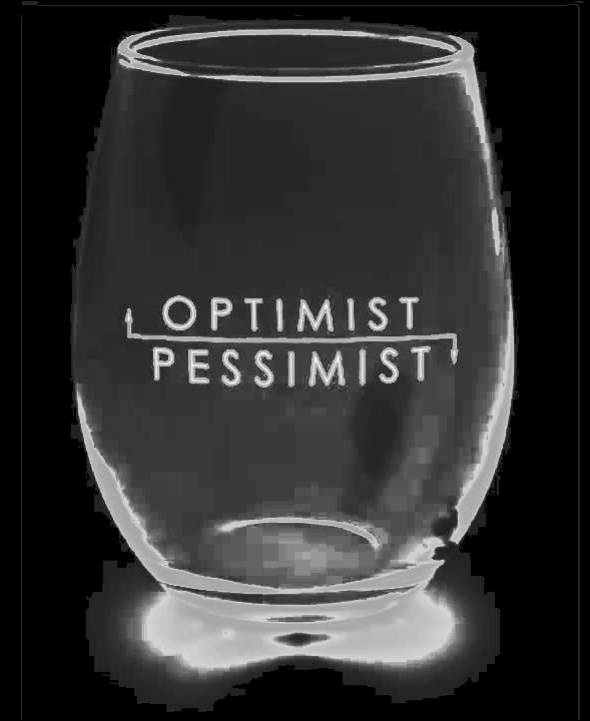


SQL Server Internals

Pessimists are Optimists with Experience

- Not testing all the code
 - Use meaningful data volumes*
- Testing in production
 - Can alter production data*
 - Interferes with production users*
- Testing in development environment
 - Useful at most for unit tests*

INSTANT
DAMNATION!



CIRCLE 8:

Indolent developers



SQL Server Internals

Development Worst Practices

- No transactions
- No error handling
 - @@ERROR is a thing of the past!*
- Wrong isolation levels
 - NOLOCK = no consistency!*
- `SELECT *`
- Dynamic SQL with concatenated params
- Code vulnerable to *SQL injection*
- No abstraction layer
 - Views, Functions, Stored Procedures*



It's all about laziness



SQL Server Internals

CIRCLE 9:

Stingy buyers



SQL Server Internals

HW Worst Practices

- Using inadequate or unbalanced HW
- Reusing decommissioned servers for new installations
 - Slower CPUs (license costs the same on fast CPUs)
 - Less RAM supported
- Planning storage with capacity in mind
 - Choosing the wrong RAID level



INSTANT
DAMNATION!



CIRCLE 10:

Next next finish installers



SQL Server Internals

Installation Worst Practices

- Installing accepting all the defaults
 - *Data files on the system drive*
 - *MAXDOP = 0*
 - *Max Server Memory = $+\infty$*
- Installing unused components
- Installing multiple services on the same machine
- Giving up easy wins on I/O
 - Partition misalignment
 - Using the default allocation unit (4Kb)



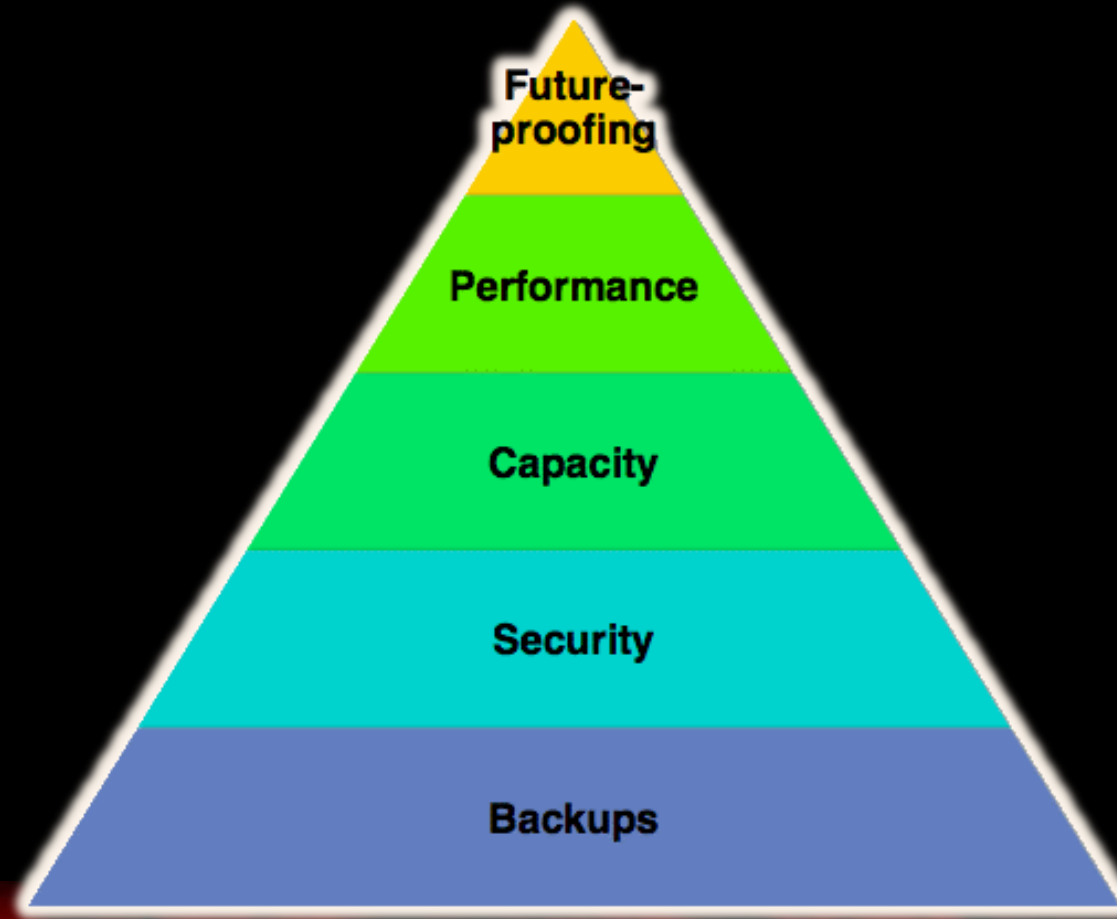
CIRCLE 11:

Careless caretakers



SQL Server Internals

What does a database need?



Backup and Recovery Worst Practices

- No backup
 - *With FULL recovery it's a timebomb*
 - *Ignoring RPO and RTO (it's not your decision!)*
- No test restores
- No consistency checks
 - *DBCC REPAIR_ALLOW_DATA_LOSS as default response to corruption*



**Our responsibility is to perform restores,
not backups!**



Security Worst Practices

- Too many sysadmins
- Everyone authenticating as 'sa'
- Using SQL Authentication
 - *Weak passwords*
 - 123
 - P4\$\$w0rd
 - Same as username
- No auditing on sensitive data
- Storing passwords in the database



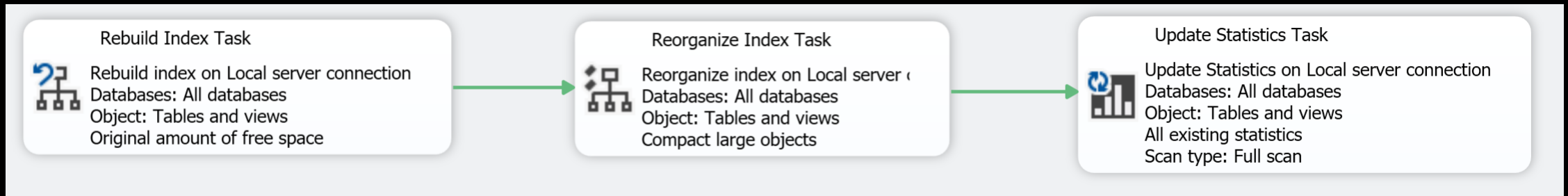
Capacity Management Worst Practices

- Not checking disk space
 - *No space left = database halted!*
 - *FULL recovery and no log backups?*
- Relying 100% on autogrow
- Autoshrink
- Autoclose
- Not presizing tempdb
 - Different file size = latching (and striping) penalty*



Maintenance Worst Practices

- Not maintaining indexes and statistics
- Obsessing over maintaining indexes and statistics
- Using catch-all maintenance plans



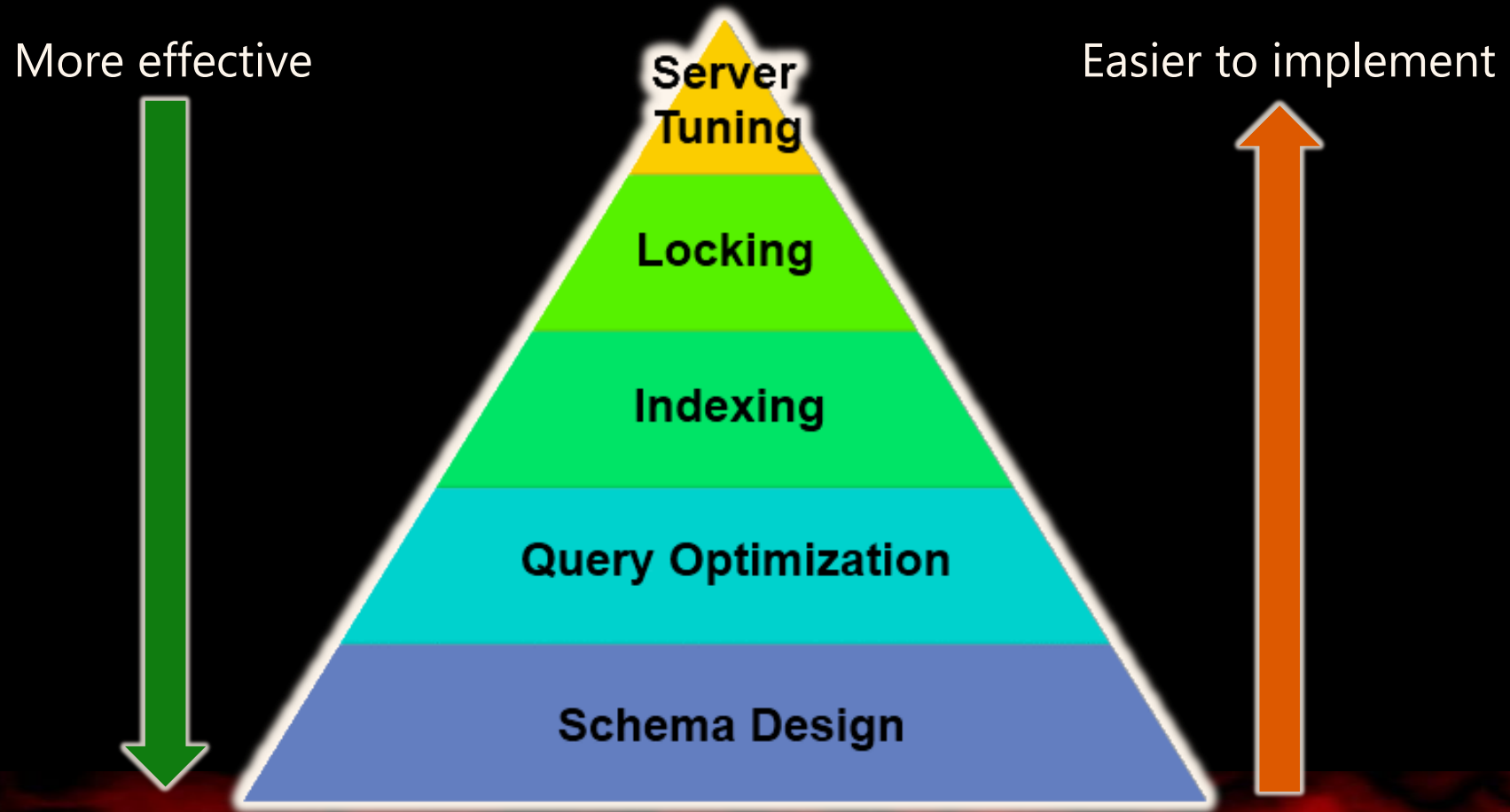
CIRCLE 12:

Performance Killers



SQL Server Internals

Performance Tuning



Query Optimization Worst Practices

RBAR: Row By Agonizing Row

- *Cursors*
- *WHILE loops*
- *App-side cursors*
- *Scalar and multi-statement functions*



Query Optimization Worst Practices

Views on views on views...

Might look like a brilliant idea at first (code re-use FTW!)

- You can end up losing control
- Unneeded multiple accesses to the same tables
- Unnecessary JOINS



Query Optimization Worst Practices

- One query to rule them all
The optimizer is good, not perfect
«divide et impera» delivers better performance
- DISTINCT in all queries
... because “who wants stinkin’ duplicates?”
- Query HINTs all over the place
Especially index hints



Indexing Worst Practices

- Accepting all suggestions from Tuning Advisor
- Duplicate indexes
- An index for each column
 - *Indexes are not for free!*
- Suboptimal Clustered Index
 - Unique
 - Small
 - Unchanging
 - Ever increasing or decreasing



NEWSEQUENTIALID()
NEWID()



Server Tuning Worst Practices

- «*Throwing HW*» at the problem
 - *A 2x faster machine might make RBAR code 2x faster*
 - *Using set-based code might make it 500x faster*
- Using «advanced» options without testing
 - *NT Fibers (lightweight pooling)*
 - *Priority Boost*



Resources

Detailed blog posts on spaghettidba.com

One post for each circle:

<https://spaghettidba.com/category/sql-server/sql-server-infernals/>



SQL Server Infernals

Resources

Free Tool:

Best Practices Analyzer

- Highlights configuration parameters that don't comply with best practices
- Highlights potential problems
- Offers recommendations

<http://www.microsoft.com/en-us/download/details.aspx?id=15289>



SQL Server Infernals



Questions?



SQL Server Internals



Thanks!



SQL Server Internals