

**NAV
TECH
DAYS
2017**

mibuso.com

C/AL, CODING FOR PERFORMANCE

JÖRG STRYK
STRYK SYSTEM IMPROVEMENT

JASMINKA THUNES
NXM BUSINESS SOLUTIONS

WHEN YOU ARE PASSIONATE ABOUT MICROSOFT DYNAMICS NAV | www.navtechdays.com

**NAV
TECH
DAYS
2017**
mibuso.com

Jasminka Thunes



ERP Software since 1997

Microsoft & Dynamics NAV 2002 – 2017

Application Developer

Escalation Engineer

Premier Field Engineer/Consultant

NxM Business Solutions 2017 -

Head of Development Dept.

Jörg Stryk – If your NAV sucks, it wasn't fixed by me!

MS Dynamics NAV (Navision) since 1997
Since version 1.2

MS SQL Server since 2003
Since version 2000

100% focus on **NAV/SQL Performance Optimization**

STRYK System Improvement (since 2006)

More than 500 projects in about 25 countries on 5 continents (December 2016)

Worldwide support of MS Dynamics Partners & Customers



Microsoft Most Valuable Professional (retired)
MVP MS Dynamics NAV 2007-2017

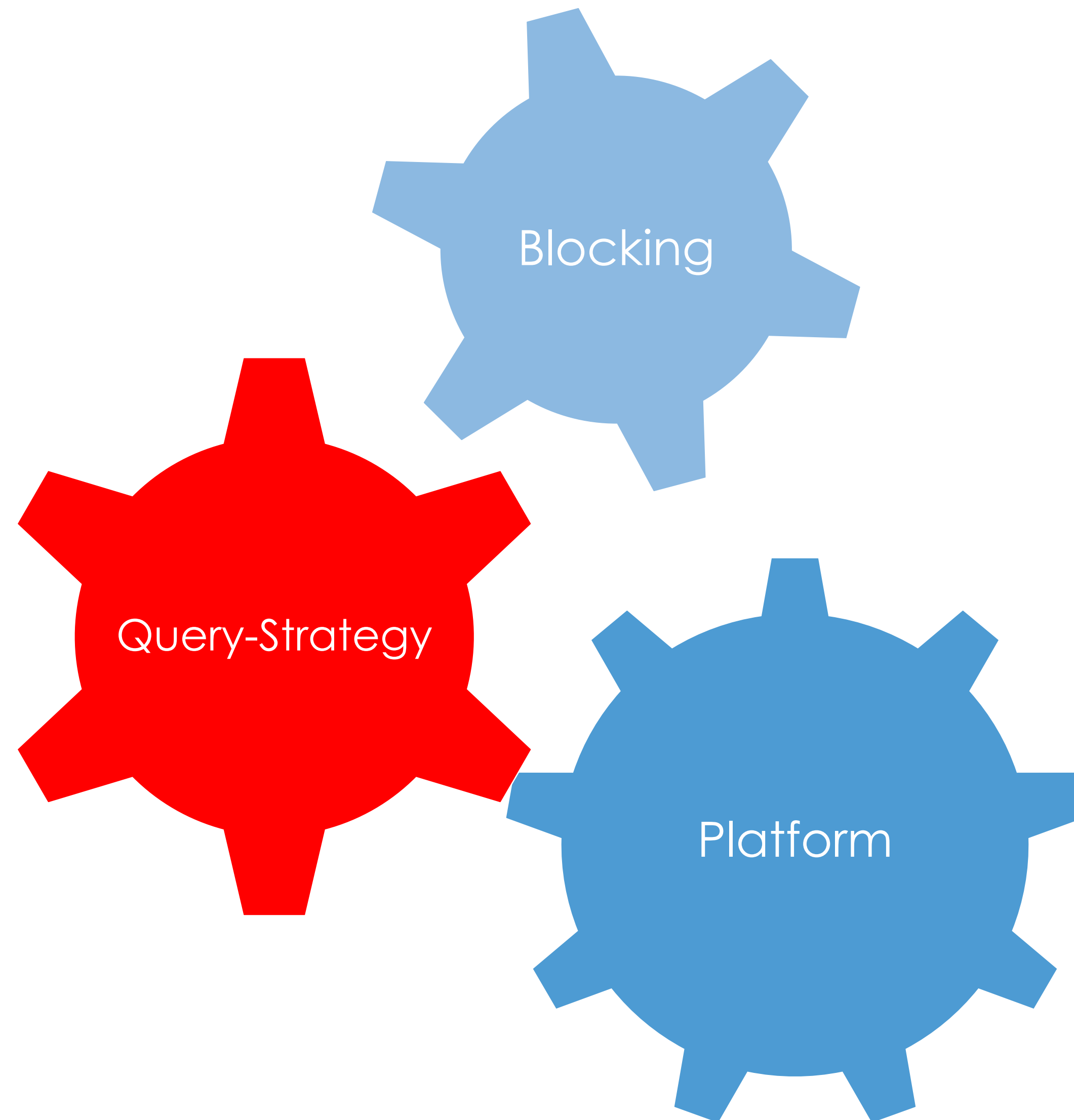
Book: “NAV/SQL Performance Field Guide”

ISBN 978-3-8370-1442-6

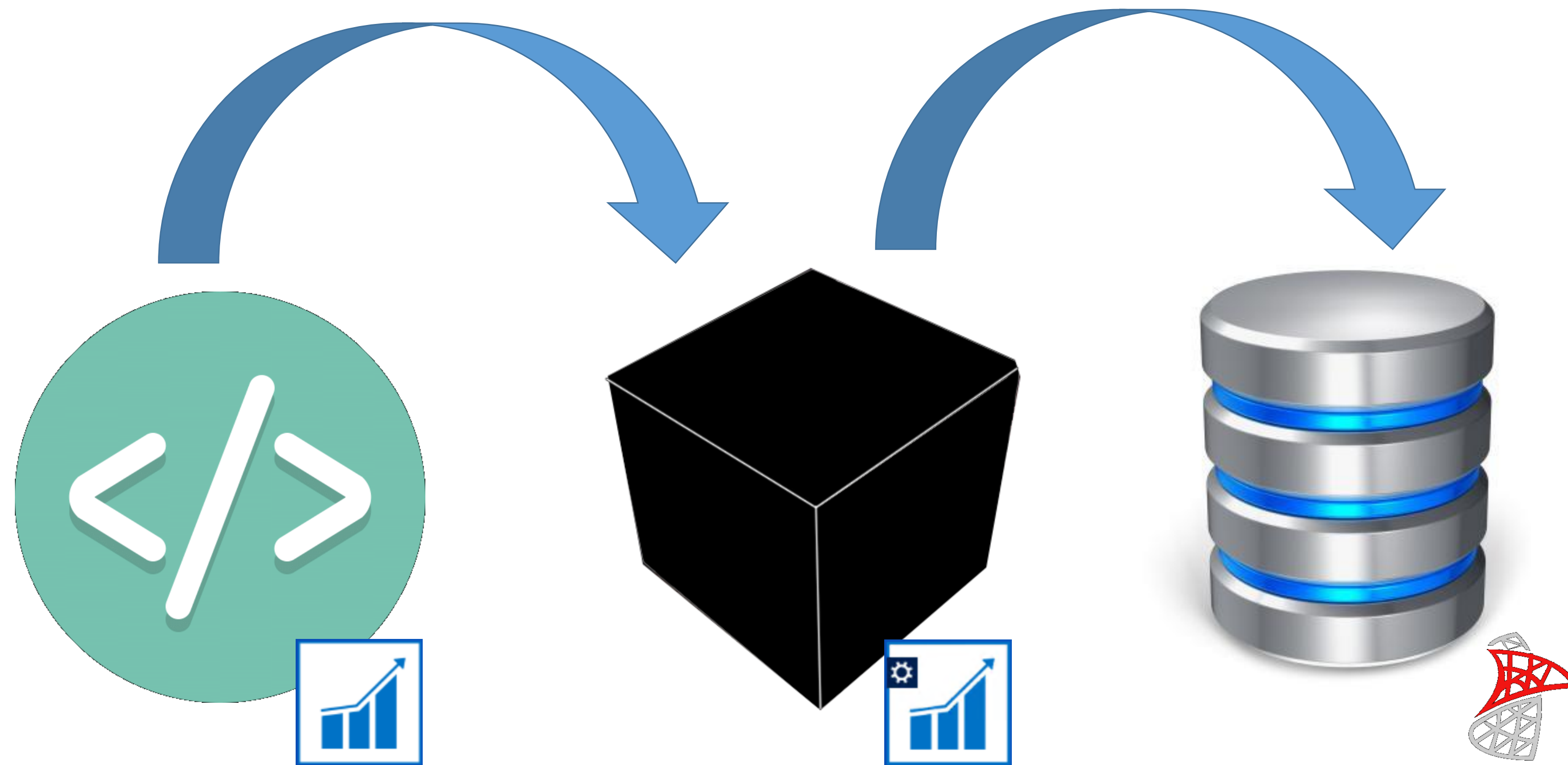
Software: “SSI Performance Toolbox”

<http://www.stryk.info/toolbox.html>

Performance Areas



Core Architecture



WHAT YOU WANT

WHAT YOU GET

C/AL Programming

- ✓ Algorithm Efficiency
- ✓ Data Retrieval Methods
- ✓ DO's & DON'Ts
- ✓ Be Mindful Of...
- ✓ Tools

Algorithm Efficiency

$O(n)$ a.k.a. **Big O**

Measures how well a computer algorithm scales as volume of data increases

➤ Not necessarily a measure of speed, but a measure of scalability

n = 1000

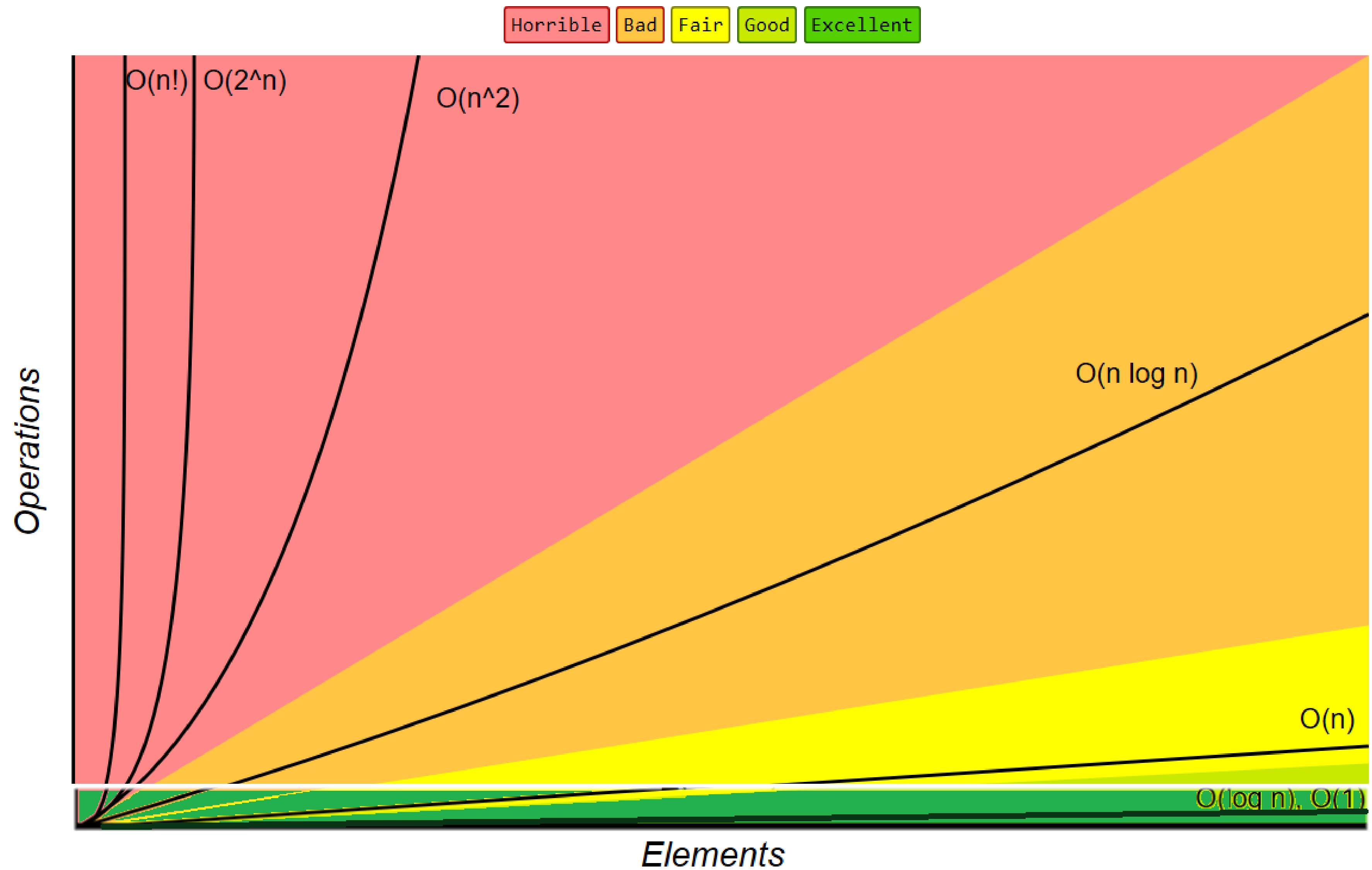
$O(1) = 1$
 $O(\log n) = 7$
 $O(N) = 1000$
 $O(n \log n) = 7000$
 $O(n^2) = 1\,000\,000$

n = 1 000 000

$O(1) = 1$
 $O(\log n) = 14$
 $O(N) = 1\,000\,000$
 $O(n \log n) = 14\,000\,0000$
 $O(n^2) = 1\,000\,000\,000\,000$

Big O

Big-O Complexity Chart



Algorithm Efficiency in (C/)AL

Minimize server roundtrips!

- Replace indented loops with queries, where feasible
- Use SETAUTOCALCFIELDS, where applicable
- Use CALCSUMS, where applicable
- Use MODIFYALL/DELETEALL, where applicable
- Code with iterations in mind

Data Retrieval Methods

- **FIND**
 - when requesting a set of data which may not be completely enumerated/read.
- **FINDSET**
 - when requesting a set of data that will be completely read
 - when modifying a set of data (w/ findset parameters)
- **FINDFIRST**
 - when requesting only a top (1) row
- **FINDLAST**
 - when requesting only the last row
- **ISEMPTY**
 - when you need to know **IF** the record exists, and not the values in the record

DO's & DON'Ts

```
IF RecA.FINDSET THEN  
REPEAT  
    MyRec."Entry No.":=  
        MyRec. "Entry No." + 1;  
    ....  
  
    MyRec.INSERT;  
  
UNTIL (RecA.NEXT = 0)
```

```
IF RecA.FINDSET THEN  
REPEAT  
    MyRec."Entry No.":=  
        MyRec. "Entry No." + 1;  
    ....  
  
    IF MyRec.INSERT THEN;  
  
UNTIL (RecA.NEXT = 0)
```

DO's & DON'Ts

```
MyRec.SETCURRENTKEY(FieldA,FieldB);  
MyRec.SETRANGE(FieldA,<filter>);  
IF MyRec.FIND(-) THEN  
REPEAT  
    MyRec2 := MyRec;  
    MyRec2.FieldA:= TmpRec.FieldX;  
    ....  
    MyRec2.MODIFY;  
UNTIL (MyRec.NEXT = 0);
```

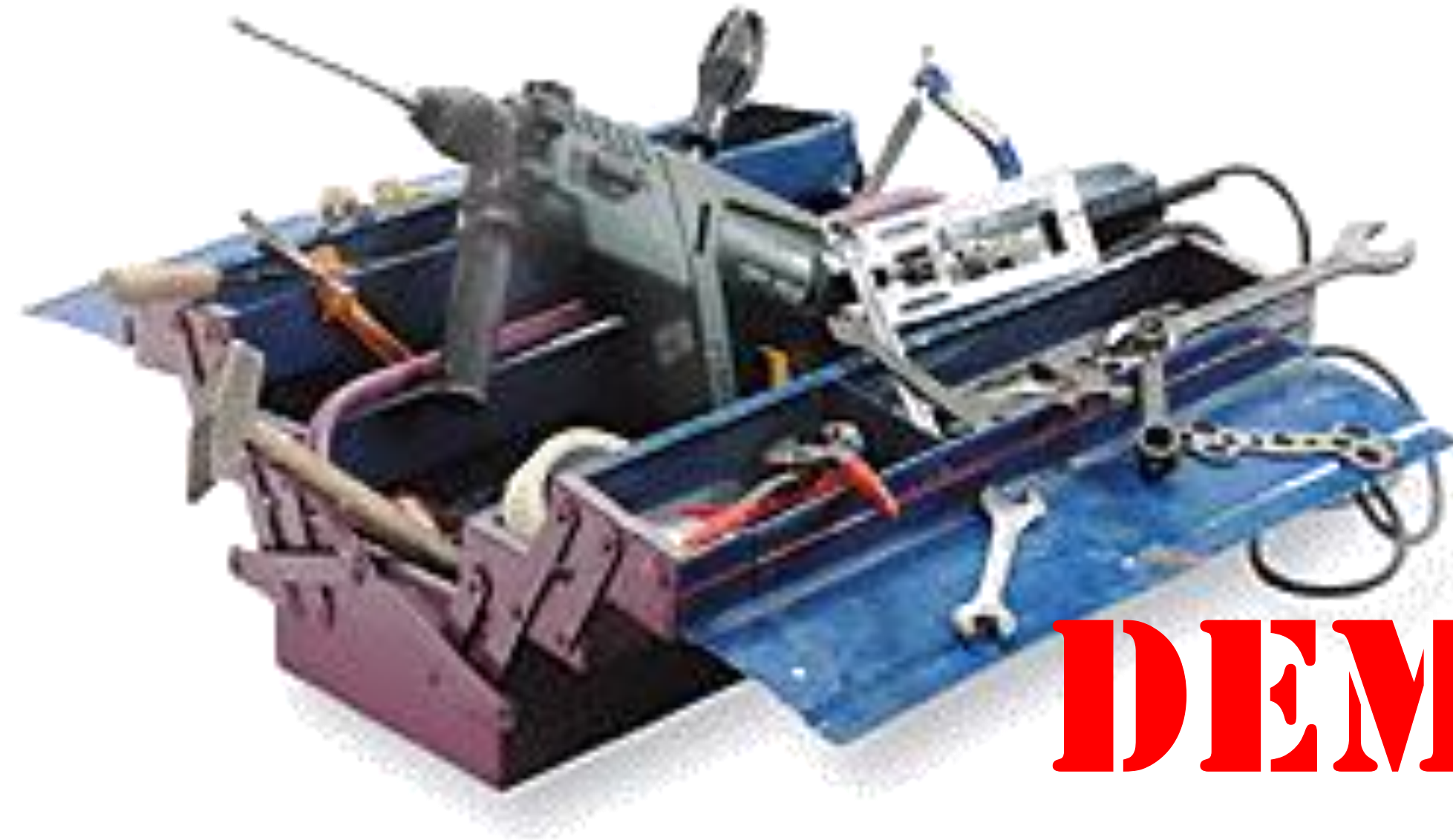
```
MyRec.SETCURRENTKEY(FieldA,FieldB);  
MyRec.SETRANGE(FieldA,<filter>);  
IF MyRec.FINDSET(TRUE,TRUE) THEN  
REPEAT  
    MyRec.FieldA:= TmpRec.FieldX;  
    ....  
  
    MyRec.MODIFY;  
UNTIL (MyRec.NEXT = 0);
```

DO's & DON'Ts

```
....  
WITH VE DO BEGIN  
  SETCURRENTKEY("Item Ledger Entry No.");  
  SETRANGE("Item Ledger Entry No.",ILE."Entry No.");  
  IF FINDSET THEN  
    REPEAT  
      SETRANGE("Document No.,"Document No.");  
      SETRANGE("Document Line No.,"Document Line No.");  
      ....  
      FINDLAST;  
      ....  
      SETRANGE("Document No.");  
      SETRANGE("Document Line No.");  
    UNTIL NEXT = 0;  
  END;  
....
```

```
myquery.OPEN;  
WHILE myquery.READ DO  
  BEGIN  
    ...  
  END;  
myquery.CLOSE;
```

C/AL Programming



DEMO

Example

Around 60 records in filtered selection

```
WITH ItemLedgEntry DO BEGIN
  SETCURRENTKEY( "Item No.", "Location Code", Open, "Variant Code", "Unit of Measure",
  Code", "Lot No.", "Serial No.");
  ...
  IF FIND('-') THEN BEGIN
    ...
    REPEAT
  ...
  SETRANGE("Item No.", "Item No.");
  IF FIND('-') THEN BEGIN
    SETRANGE("Location Code", LocCode);
    SETRANGE(Open, TRUE);
    IF FIND('-') THEN
      REPEAT
        SETRANGE("Variant Code", "Variant Code");
        IF FIND('-') THEN
          REPEAT
            SETRANGE("Lot No.", "Lot No.");
            IF FIND('-') THEN
              REPEAT
                ...
```

"Report", "5756", "59", "74"
"Report", "5756", "60", "74"
"Report", "5756", "61", "74"
"Report", "5756", "64", "74"
"Report", "5756", "65", "74"
"Report", "5756", "66", "74"
"Report", "5756", "67", "74"
...

No. of hits: 74

795	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	27
796	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	27
797	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	27
798	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	27
799	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	27
800	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	6
801	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	6
802	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	6
803	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	6
804	SELECT TOP (@0) "timestamp", "Entry No_", "Item No_", "Posting Date", "Entry Type", "Source No_", "Document No_", "Description", "Location Code", "Quantity", "Remaining Quantity", "Invoiced ...	6

Over 800 SQL statements

Over 37000 logical reads on table 32

Be Mindful Of...

- Subscribers
- Using MARK
- GUIDs

Application footprint

➤ Reports

Data set size

Client memory [\(Memory consumption on Web Client\)](#)

CAS Policy Switch (NetFx40_LegacySecurityPolicy) (memory leak vs. performance)

➤ .NET

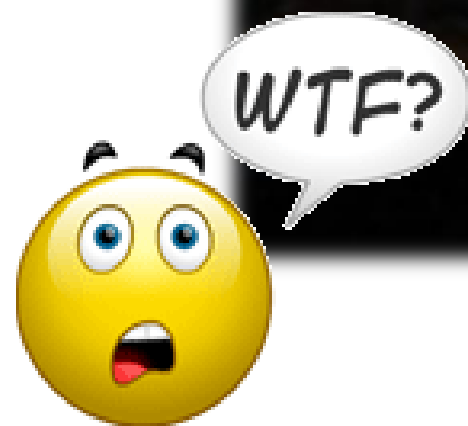
Tools

- Code Coverage
- NST Data Collector Sets
- SQL Side: Profiler/XEvents, Query Plans
- Resource Monitor

Indexing

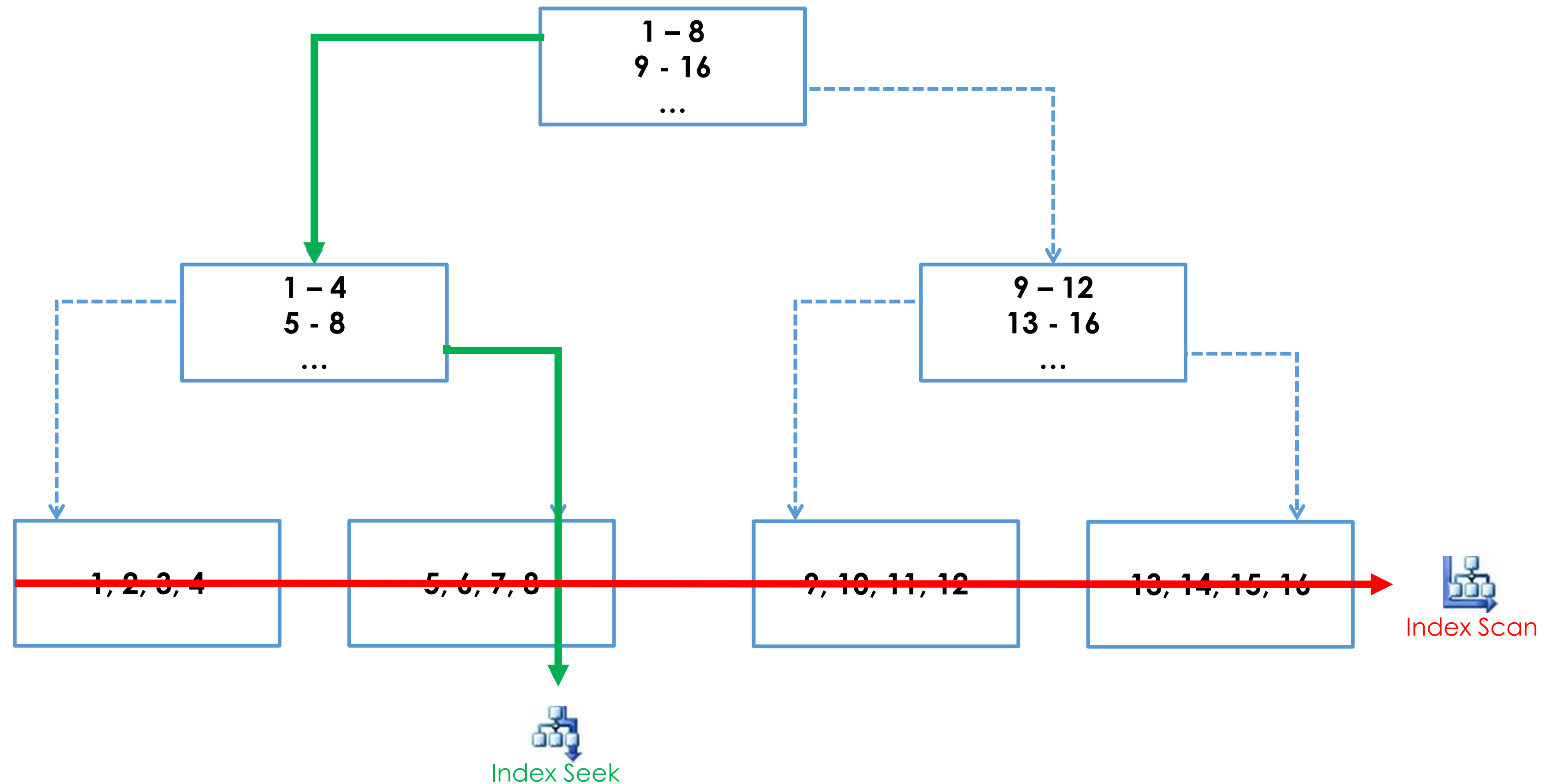
- ✓ Data Retrieval & Indexes
- ✓ From C/AL to SQL
- ✓ To SETCURRENTKEY or not to SETCURRENTKEY
- ✓ Data Density & Clustering
- ✓ Identifying & Resolving Problems
- ✓ Database Configuration & Maintenance

Data Retrieval

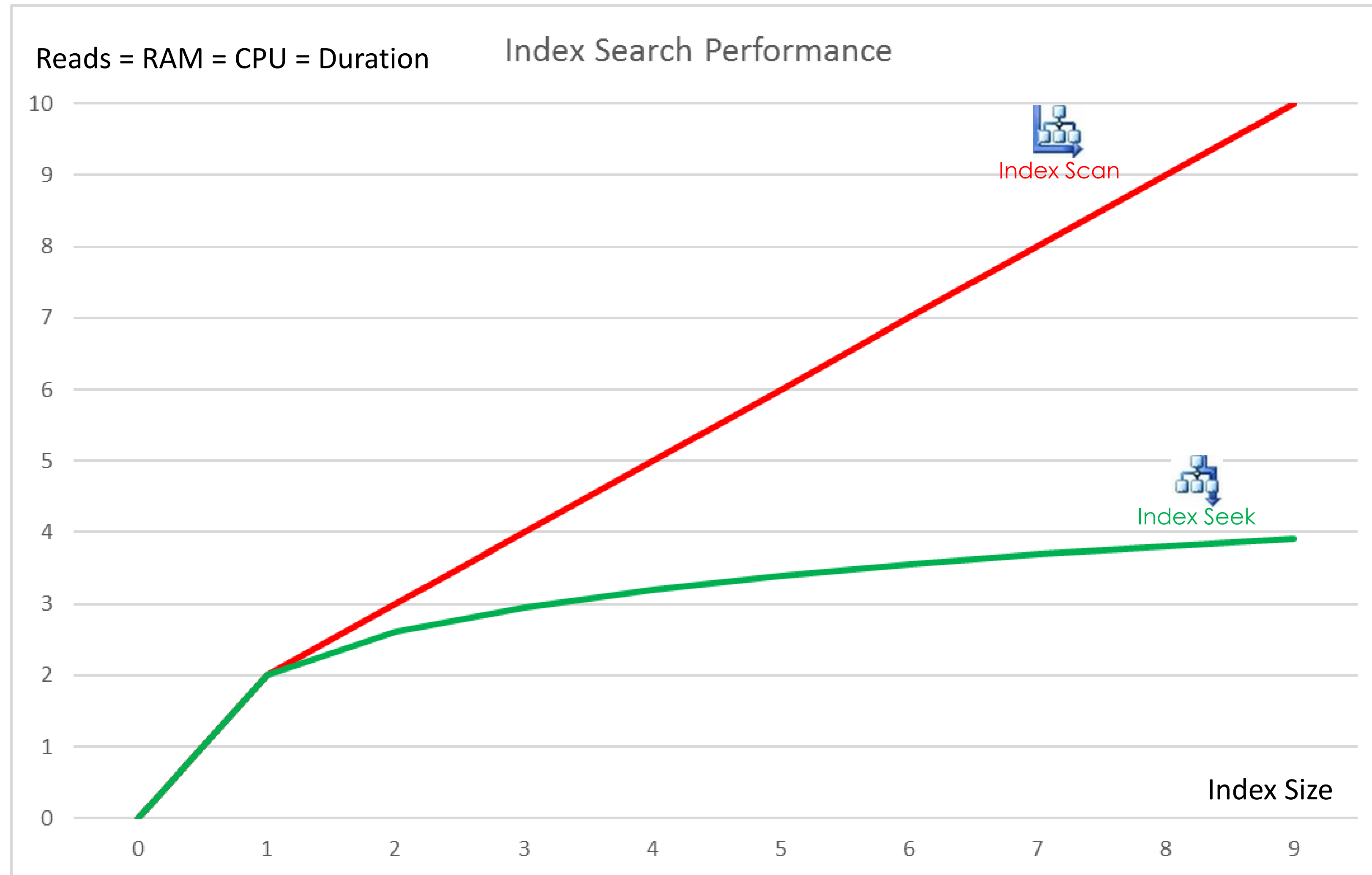


Data Retrieval

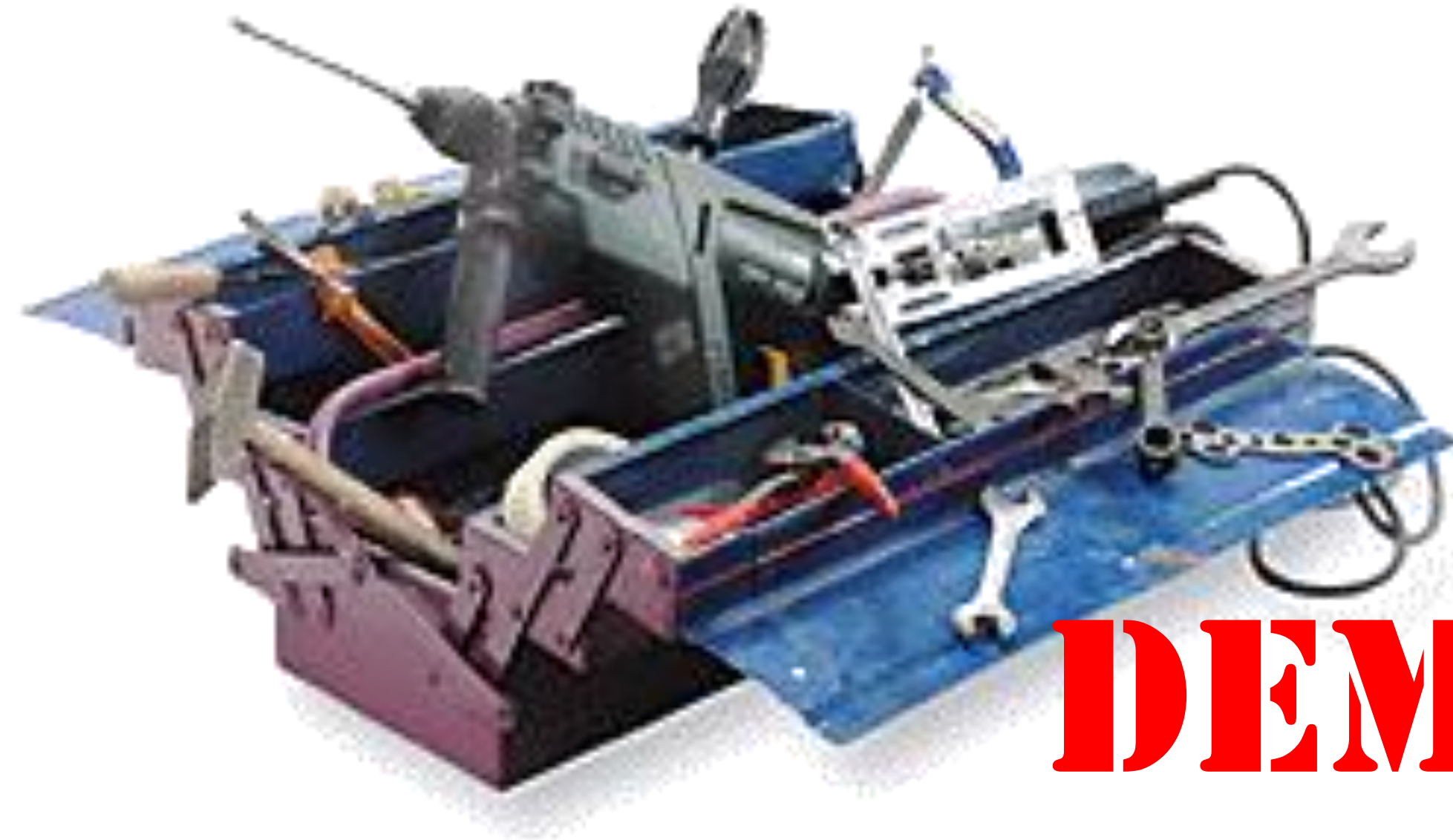
Balanced Tree



Data Retrieval



Queries, Indexes and Execution Plans



DEMO

Indexing_DataDensity_NAV2013.sql
ExpensiveQueries_MissingIndexes_generic.sql
ReScript_SSI_Indexes_generic.sql

Database Settings & Maintenance

- Auto Create Statistics = ENABLED
- Auto Update Statistics = ENABLED
- Auto Update Statistics Async. = ENABLED

- Traceflag 2371 (Dynamic Change-Thresholds) on large DB (before SQL 2016)

- Max. Degree of Parallelism = 2 (NAV 2013 and higher; older NAV incl. 2009 R2: MAXDOP = 1)

- Tempdb: 1 Datafile per (logical) CPU; no more than 8 files

Database Settings & Maintenance

- Once per Week:
 - 1. MP-Task: Index Rebuild
(User DB, Fragmentation > 30%, Online (EE), MAXDOP = 64, Free Space = 2%, Scan Type = Fast)
 - 2. MP-Task: Index Reorganize
(User DB, Fragmentation > 10%, Compact Large Objects = On, Scan Type = Fast)
 - 3. MP-Task: Update Statistics*
(User DB, All Statistics, Full Scan)
- Every Day:
 - MP-Task: T-SQL: EXEC sp_updatestats
(User DB)

And of course, though, not really related to THIS session, but mentioning it anyway, just to be complete:

- At least weekly, ideally daily:

MP-Task: Database Integrity Check
(All Databases, Include Indexes = On*, Physical Only = Off*, MAXDOP = 64)

*)depends on available time

Database Settings & Maintenance

- Query- and Index- Tuning
is necessary periodic
Database Maintenance!

C/AL Programming & Indexing

- ✓ Minimize Database Roundtrips
- ✓ Know your FIND
- ✓ Write scalable code
- ✓ Verify Code execution on NST and on SQL Server
- ✓ Have the right indexes
- ✓ Maintain the database periodically

Questions?

**NAV
TECH
DAYS
2017**

mibuso.com

Thank You!

NXM Business Solutions Jasminka Thunes



Right next to Santa
Claus



www.jasminkat.com



jth@nxm.no



Try LinkedIn

STRYK System Improvement Jörg A. Stryk



Wherever it takes



www.stryk.info



contact@stryk.info



[@navsqlperformance](https://www.facebook.com/navsqlperformance)