

Microsoft® SQL Server™ performance tuning for Microsoft Dynamics™ NAV

'TechNet Evening'

Introductions

Steven Renders
Microsoft Certified Trainer
Plataan
steven.renders@plataan.be
➔ Check Out: www.plataan.be

Overview

- ❑ Introduction
- ❑ Dynamics-NAV Architecture
- ❑ Differences Between the Server Options
- ❑ Recovery Model
- ❑ Security & Synchronisation process
- ❑ Sql Server Versions & Limitations
- ❑ How to conduct a Performance audit
- ❑ Hardware Recommendations
- ❑ The NDBCS Database Driver
- ❑ Using Find Statements

Overview

- Optimizing Dynamics NAV Indexes and SIFT Tables
- Usefull Tools & Scripts & DMV's
- Index Hinting
- Form design & performance
- Locking & Deadlocks
- Maintenance
 - Dynamics-NAV Native DB
 - Sql Server
- Q & A

Dynamics-NAV Architecture

□ The **Client**:

- basically responsible for the **user interface**.
- also responsible for executing all the **business logic**.
- **reads objects** from the database, **running** the objects, **controlling** their **behavior**.
- **Most** of the Navision application runs on the clients.

□ The **Server**:

- Controls the **number of users** that can connect.
- Controls **access** to the **data** through **locking**.
- Keeps track of all the read and write **transactions**.
- **Sends data** to each client, as requests are made.
- Performs all the **key-based filtering** & calculates the **SumIndexFields**.
- **Caches** data that can be requested again.

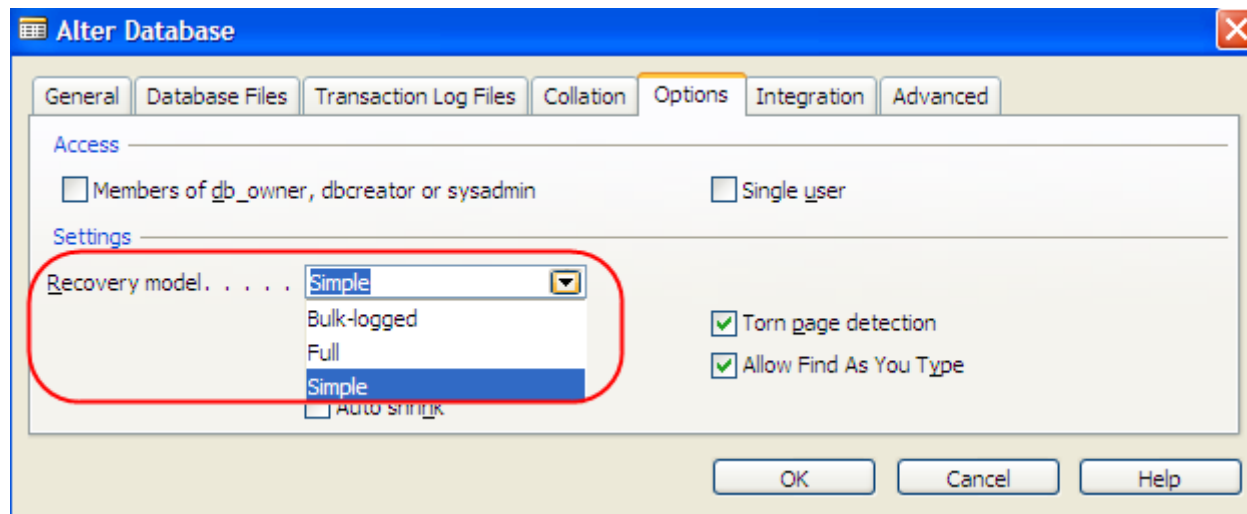
Differences Between the Server Options



- ❑ The way you **create** a database
- ❑ The **backup** facilities that are available
- ❑ The ability to access the data in the database with **third party tools**
- ❑ The way that **SIFT™** works
- ❑ Performance **monitoring**
- ❑ **Scalability**
- ❑ Multi-processor support

Recovery Model

- Determines what information is written to the **transaction log** ← recovery model you want to use in database:
 - Bulk-Logged
 - Full ← default option in Dynamics-NAV !
 - Simple



Security & Synchronisation

- **Whitepaper:** Security Synchronization in Microsoft Business Solutions-Navision 4.0 SP3:
 - When to Synchronize the Security System
 - Standard Security
 - Synchronizing the Standard Security Model
 - Enhanced Security
 - Synchronizing the Enhanced Security Model
 - Selecting the Security Model
 - After Changing the Security Model
 - Converting the Database
 - Attaching xp_ndo to SQL Server

Security & Synchronisation

Feature	Standard Security	Enhanced Security
Synchronization Performance	Fast	Slower If you have several companies and many users in the same database, the synchronization process will be slower with Enhanced Security.
Windows groups displayed	Local domain + forest of domains	Local domain only
Logins required in Navision	Windows groups and individual Windows users	Windows Groups + the members of each group and individual Windows users
Granularity of Synchronization	Entire security system	Entire security system and individual logins.
Automatic synchronization when you insert, modify or delete a Windows login or a database login in Navision.	Yes	No
Required Extended Stored Procedure	xp_ndo_enumusersids	xp_ndo_enumusergroups

Sql Server Versions & Limitations

Scalability and Performance					
Feature	Express	Workgroup	Standard	Enterprise	Comments
Number of CPUs	1	2	4	No Limit	Includes support for multicore processors.
RAM	1 gigabyte (GB)	3 GB	Operating system maximum	Operating system maximum	Memory limited to maximum supported by operating system.
64-bit Support	Windows on Windows (WOW)	WOW	✓	✓	
Database Size	4 GB	No Limit	No Limit	No Limit	
Partitioning				✓	Support for large-scale databases
Parallel Index Operations				✓	Parallel processing of indexing operations
Indexed Views				✓	Indexed view creation is supported in all editions. Indexed view matching by the query processor is supported only in Enterprise Edition.

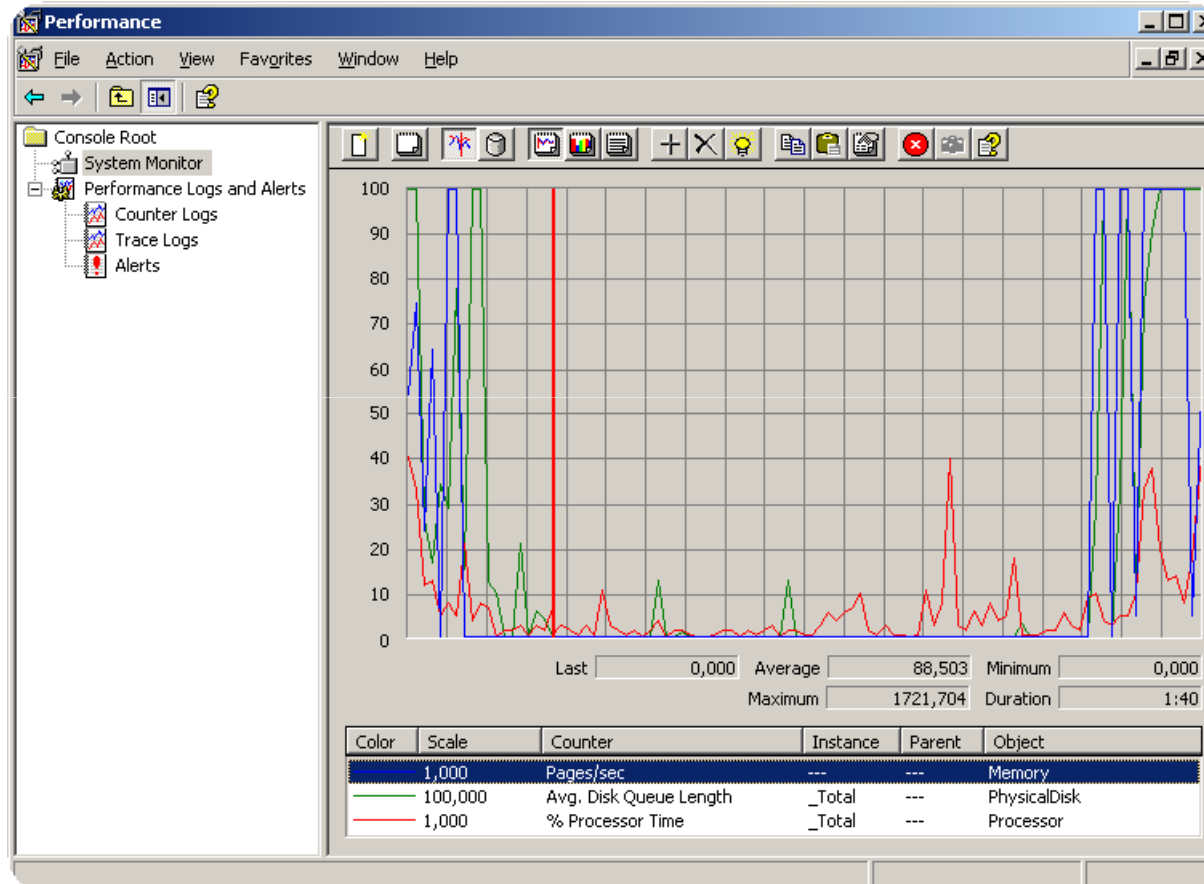
Sql Server Versions & Limitations

Manageability					
Feature	Express	Workgroup	Standard	Enterprise	Comments
Auto Tuning	✓	✓	✓	✓	Automatically tunes database for optimal performance.
Profiler	1	2	✓	✓	
SQL Server Management Studio Express	See Comments	✓	✓	✓	Easy-to-use graphical management tool available as a separate download or included with the SQL Server Express w/Advanced Services download
Management Studio		✓	✓	✓	Full management platform for SQL Server; includes Business Intelligence (BI) Development Studio.
Database Tuning Advisor			✓	✓	Automatically suggests enhancements to your database architecture to improve performance.
Serviceability Enhancements	✓	✓	✓	✓	Dynamic management views and reporting enhancements.
Full-text Search	See comments	✓	✓	✓	Available for SQL Server Express in the SQL Server Express w/ Advanced Services download
SQL Agent Job Scheduling Service		✓	✓	✓	

How to conduct a Performance audit

- Use **checklists**:
 - Server Hardware Performance Checklist
 - Operating System Performance Checklist
 - SQL Server Configuration Performance Checklist
 - Database Configuration Settings Performance Checklist
- Use **Performance Monitor** to Identify **Hardware Bottlenecks**
- Use **Session Monitor** to locate **clients** that cause problems
- Use **Client Monitor and/or Sql Profiler** to Identify Poorest Performing **Queries**
- **Use DMV's & DMF's !**

Performance/System Monitor

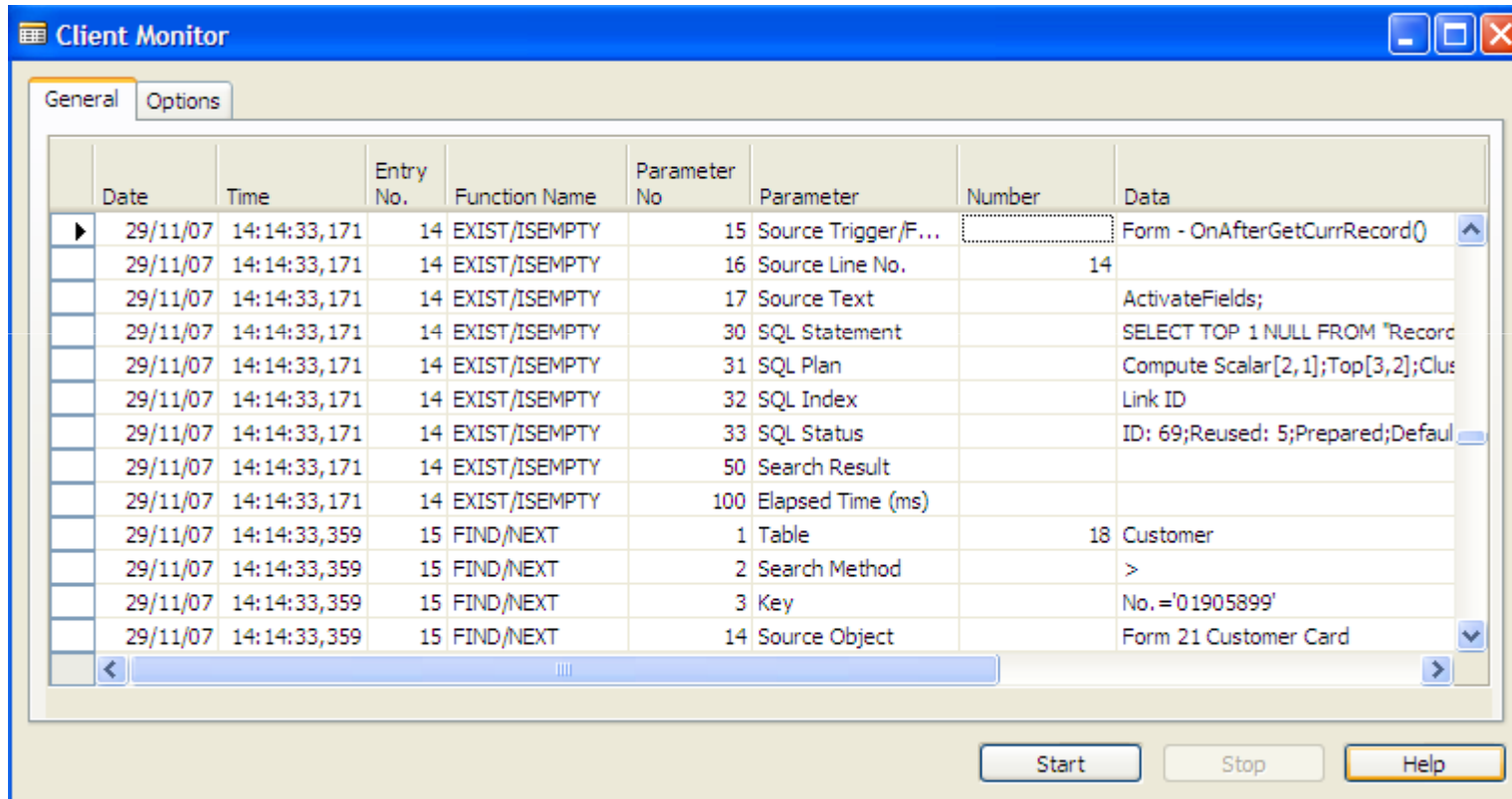


Interesting Counters

Object Name	Counter Name	Instances	Best Values	Recommendation (Best Values not met)
Memory	Available Mbytes	SQL Server TS Servers	>5MB	Add more memory Reserve less memory for SQL Server
	Pages/sec	SQL Server TS Servers	<25	Add more memory Reserve less memory for SQL Server
Physical Disk	Avg Disk Read Queue length	SQL Server Disks	<2	Change disk system
	Avg Disk Write Queue length	SQL Server Disks	<2	Change disk system
Processor	% Processor time	SQL Server TS Servers	0-80	Add more CPUs
System	Processor Queue Length	SQL Server TS Servers	<2	Add more CPUs
	Context Switches/sec	SQL Server TS Servers (Multi Processors)	<8000	Set Affinity Mask
Network Interface	Output queue length	SQL Server TS Servers	<2	Increase network capacity
SQL Server Access Methods	Full Scans/sec	SQL Server		Review Navision C/AL Code
	Page Splits/sec	SQL Server	0	Defrag SQL Server Indexes Review Navision C/AL Keys
SQL Server Buffer Manager	Buffer Cache Hit Ratio	SQL Server	>90	Add more memory
SQL Server Databases	Log Growths	SQL Server	0 (during peak times)	Increase and set the size of the transaction log
SQL Server General Statistics	User Connections	SQL Server		
SQL Server Locks	Lock Requests/sec	SQL Server		Review Navision C/AL Code
	Lock Wait/sec	SQL Server		Review Navision C/AL Code

Client Monitor

□ DEMO

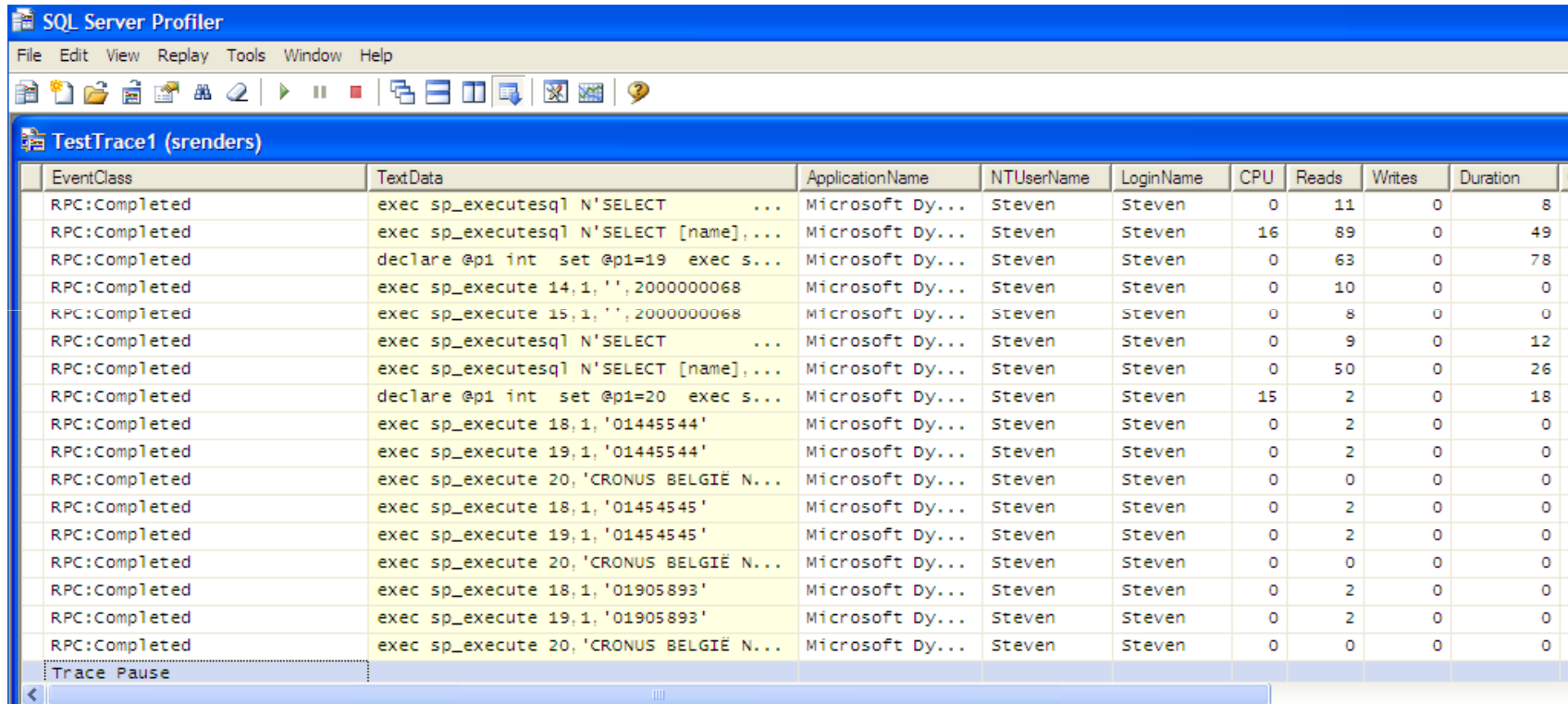


The screenshot shows the 'Client Monitor' application window with a table of system events. The table has columns for Date, Time, Entry No., Function Name, Parameter No., Parameter, Number, and Data. The events are grouped by date and time, showing various system functions like EXIST/ISEMPTY, FIND/NEXT, and SQL operations.

Date	Time	Entry No.	Function Name	Parameter No.	Parameter	Number	Data
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	15	Source Trigger/F...		Form - OnAfterGetCurrRecord()
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	16	Source Line No.	14	
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	17	Source Text		ActivateFields;
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	30	SQL Statement		SELECT TOP 1 NULL FROM "Record
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	31	SQL Plan		Compute Scalar[2, 1];Top[3,2];Clus
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	32	SQL Index		Link ID
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	33	SQL Status		ID: 69;Reused: 5;Prepared;Defaul
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	50	Search Result		
29/11/07	14:14:33,171	14	EXIST/ISEMPTY	100	Elapsed Time (ms)		
29/11/07	14:14:33,359	15	FIND/NEXT	1	Table	18	Customer
29/11/07	14:14:33,359	15	FIND/NEXT	2	Search Method		>
29/11/07	14:14:33,359	15	FIND/NEXT	3	Key		No. ='01905899'
29/11/07	14:14:33,359	15	FIND/NEXT	14	Source Object		Form 21 Customer Card

SQL Profiler

□ DEMO



The screenshot shows the SQL Server Profiler interface. The main window displays a table of events for a trace named 'TestTrace1 (srenders)'. The table has columns for EventClass, TextData, ApplicationName, NTUserName, LoginName, CPU, Reads, Writes, and Duration. The events are all of type 'RPC:Completed' and represent various SQL statements being executed.

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration
RPC:Completed	exec sp_executesql N'SELECT ...	Microsoft Dy...	Steven	Steven	0	11	0	8
RPC:Completed	exec sp_executesql N'SELECT [name], ...	Microsoft Dy...	Steven	Steven	16	89	0	49
RPC:Completed	declare @p1 int set @p1=19 exec s...	Microsoft Dy...	Steven	Steven	0	63	0	78
RPC:Completed	exec sp_execute 14,1, '',2000000068	Microsoft Dy...	Steven	Steven	0	10	0	0
RPC:Completed	exec sp_execute 15,1, '',2000000068	Microsoft Dy...	Steven	Steven	0	8	0	0
RPC:Completed	exec sp_executesql N'SELECT ...	Microsoft Dy...	Steven	Steven	0	9	0	12
RPC:Completed	exec sp_executesql N'SELECT [name], ...	Microsoft Dy...	Steven	Steven	0	50	0	26
RPC:Completed	declare @p1 int set @p1=20 exec s...	Microsoft Dy...	Steven	Steven	15	2	0	18
RPC:Completed	exec sp_execute 18,1, '01445544'	Microsoft Dy...	Steven	Steven	0	2	0	0
RPC:Completed	exec sp_execute 19,1, '01445544'	Microsoft Dy...	Steven	Steven	0	2	0	0
RPC:Completed	exec sp_execute 20, 'CRONUS BELGIË N...	Microsoft Dy...	Steven	Steven	0	0	0	0
RPC:Completed	exec sp_execute 18,1, '01454545'	Microsoft Dy...	Steven	Steven	0	2	0	0
RPC:Completed	exec sp_execute 19,1, '01454545'	Microsoft Dy...	Steven	Steven	0	2	0	0
RPC:Completed	exec sp_execute 20, 'CRONUS BELGIË N...	Microsoft Dy...	Steven	Steven	0	0	0	0
RPC:Completed	exec sp_execute 18,1, '01905893'	Microsoft Dy...	Steven	Steven	0	2	0	0
RPC:Completed	exec sp_execute 19,1, '01905893'	Microsoft Dy...	Steven	Steven	0	2	0	0
RPC:Completed	exec sp_execute 20, 'CRONUS BELGIË N...	Microsoft Dy...	Steven	Steven	0	0	0	0

Setup a TEST environment

- Test on a **separate** machine and not on production system !
- **Copy** the production database to the test server:
 - SQL Server: use backup/restore functions in Enterprise Manager / Mngt-Studio
 - Microsoft Dynamics NAV Database Server: use server-based HotCopy.
- **Warming up** the server → to ensure that you get realistic measurements:
 - **execution plans** for most queries generated & ready for use.
 - most frequently used **data** is now available in **memory**.

Hardware Recommendations

□ CPU:

- #cpu's, speed, cache
- 64-bit

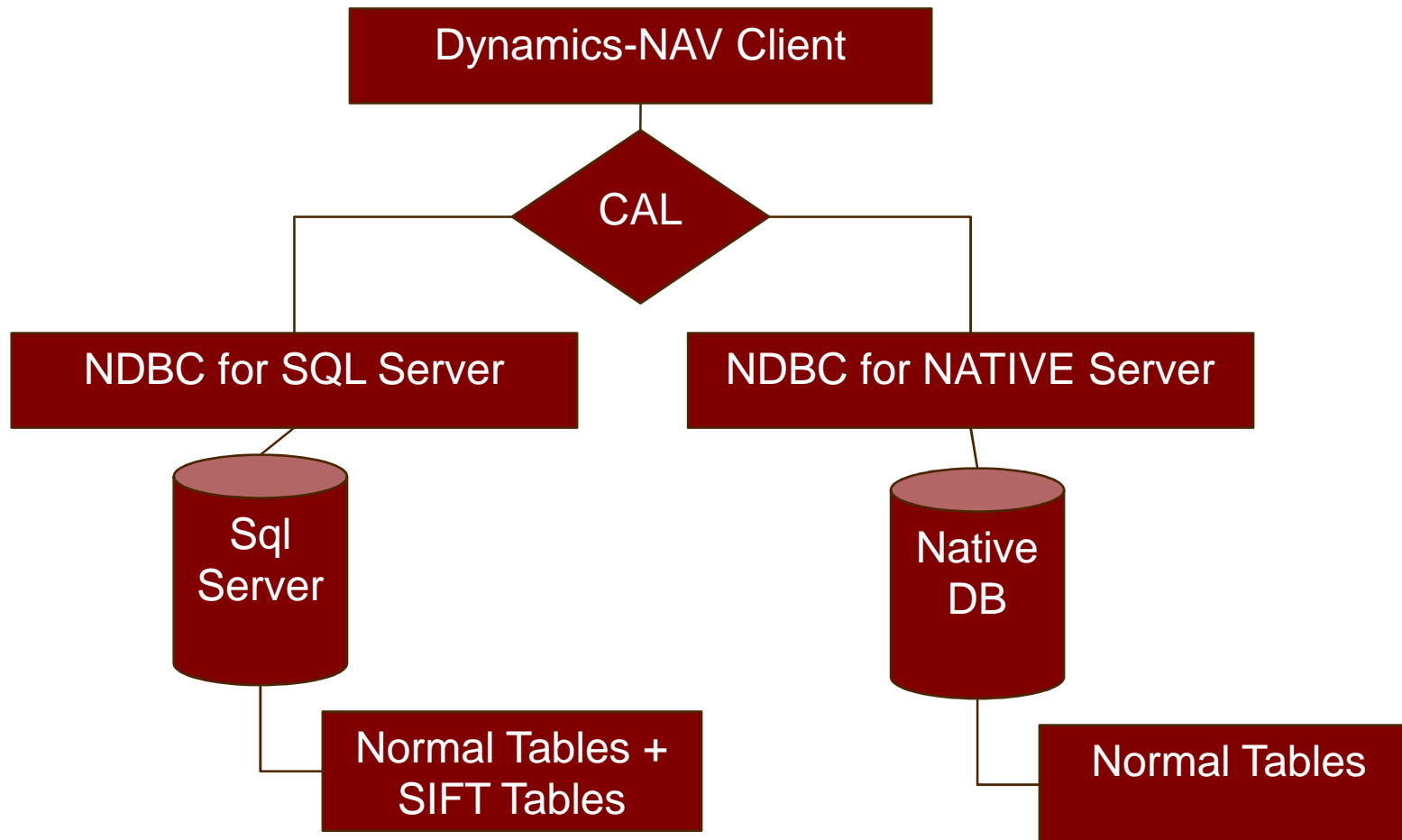
□ RAM MEMORY:

- As much as possible (64bit cpu)
- SQL Server prefers to stay in RAM as much as it can

□ HARD DISK(S):

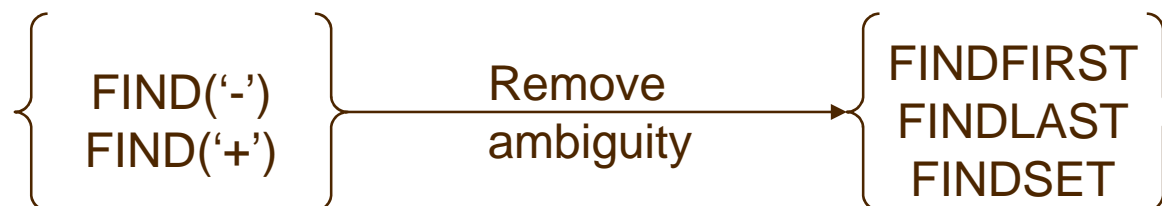
- **RAID**: Redundant Array of Independent Disks
 - Use RAID 1 or RAID 10, do not use RAID 5
- You must create **several database files** & store them in **separate disks**
- "When ever you double the amount of disks, you increase performance by 100%"

The NDBCS Database Driver



Using Find Statements

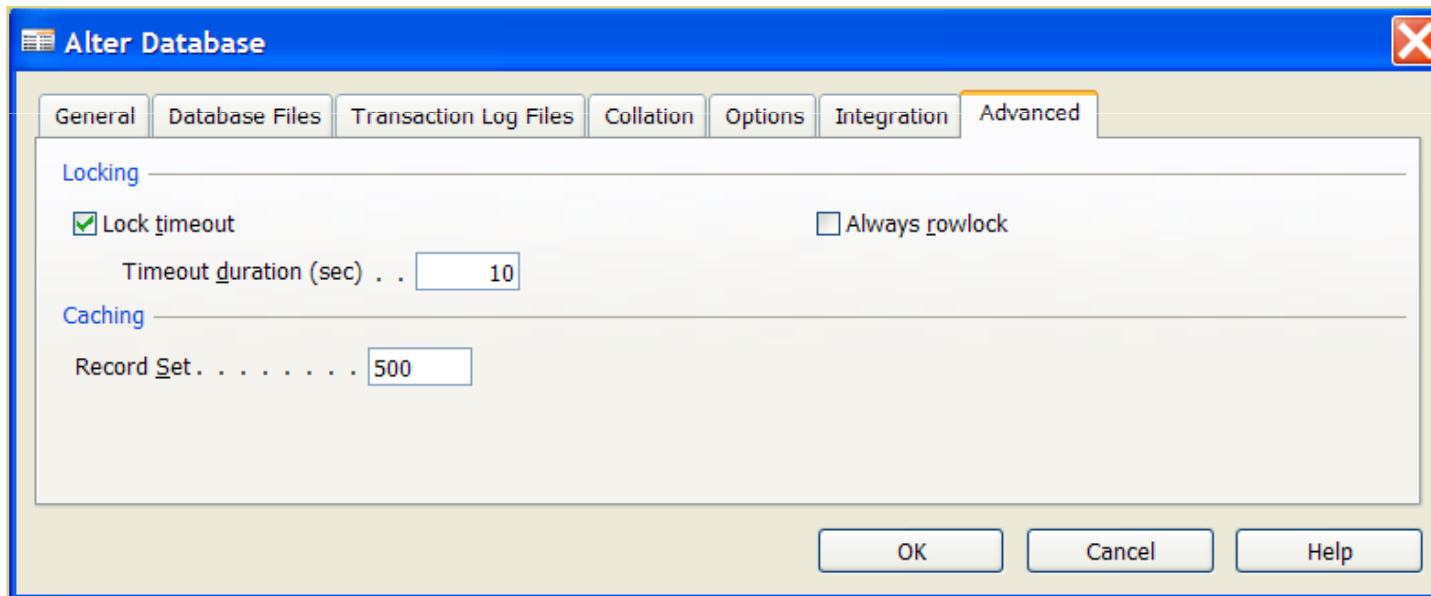
- Microsoft Dynamics NAV DB versus SQL Server
 - Microsoft Dynamics NAV Server returns **record by record**
 - Indexed Sequential Access Method (ISAM)
 - Sql Server return **Result Sets**
 - Sql Server option for Microsoft Dynamics NAV is designed to detect if it is **reading in a loop or reading single records**
 - Loops → Result Sets
 - Single records → Singleton query (Select TOP 1 ...)



- **[OK]:=FINDSET([ForUpdate][,UpdateKey]);**
 - Use this function to find a set of records in a table based on the current key and filter.
- **[ForUpdate]:**
 - FALSE if you don't intend to modify any records in set.
 - TRUE if you want to modify some records in the set.
 - If you set this parameter to TRUE, a LOCKTABLE is immediately performed on the table before the records are read.
- **[UpdateKey]:**
 - This only applies if ForUpdate is set to TRUE.
 - If you are going to modify any field value within the current key, set this parameter to TRUE.

FindSet

- ❑ **Optimize** loops
- ❑ **Reuse** of **cursors**
- ❑ Database property **RecordSet**: Sets amount of records retrieved in default recordset (cached)



Optimizing Microsoft Dynamics NAV Indexes and SIFT Tables

- In Microsoft Dynamics NAV, **indexes** are created for several purposes the most important of which are:
 - **Data retrieval:**
 - To quickly retrieve a result set based on a filter.
 - **Sorting:**
 - To display a result set in a specific order.
 - **SIFT** (Sum Index FlowField Technology):
 - SIFT is used to maintain pre-calculated sums for various columns.

- **All** the **indexes** in Microsoft Dynamics NAV are **unique**.
- A **primary** key → **unique clustered** index on SQL Server
- A **secondary** key → **unique non-clustered** index in SQL Server.
- Microsoft Dynamics NAV Database Server supports **SIFT** effortlessly.
 - In the SQL Server Option, when a SIFT field is defined on any index an **extra table** is created on SQL Server.
 - This table is **maintained** by **triggers** that have been placed on the source data table.

Sum Index Fields

- Every time you **insert/delete/update** data in a base table → all affected **SIFT tables** must also be **updated**.
- The SIFT **trigger** manages this procedure automatically → can take a long time → performance decrease !
- **Remember:** “Every single record that is inserted into a base table can cause hundreds of records to be updated in the SIFT tables !!!”

Sum Index Fields

- ❑ Have you **designed** your **SIFT indexes** optimally?
- ❑ Supporting **too many** SIFT indexes will affect performance.
- ❑ Unnecessary **date fields** in SIFT indexes of base table affects performance → create three times as many entries as an ordinary field.
- ❑ **Too many fields** in the SIFT indexes will also affect performance.
- ❑ The fields in the SIFT index that are **used most regularly** in queries must be positioned to the **left** in the SIFT index.
- ❑ Rule: the field that contains the greatest number of **unique values** must be placed on the **left** with the field that contains the second greatest number of unique values on its right and so on.
- ❑ Are there **too many SIFT levels**?
- ❑ If one of your SIFT tables becomes very large you might want to determine whether or not it should be **optimized**.

Recommendations

- **Eliminate** the **maintenance** of **indexes** that are only designed for **sorting** purposes.
- **Redesign** the indexes so that their **selectivity** becomes **higher** by putting **Boolean, Option** and **Date** fields towards the **end** of the index.
- Don't maintain SIFT indexes on **small/temporary** tables:
 - Sales Line, Purchase Line, Warehouse Activity Line, ...
- **AVOID** WHILE FIND(`-`) or WHILE FIND(`+`):
 - automatically disables the **read ahead mechanism**.
 - → use **REPEAT UNTIL NEXT** instead

Optimizing Microsoft Dynamics NAV Indexes and SIFT Tables

- ❑ **Minimize** the Number of Indexes
- ❑ Indexes on '**Hot**' Tables
- ❑ **Redesign** Indexes for Better Selectivity
- ❑ **SQLIndex** Key Property
- ❑ **Clustered** Key Property
- ❑ Small/Temporary Tables **SIFT Maintenance**
- ❑ Minimize Number of **SIFT Buckets**

Keys & Properties

Table 21 Cust. Ledger Entry - Keys

E. Key	SumIndexFields	KeyGroups	MaintainSQLIndex	MaintainSIFTIndex	SIFTLevels	Clustered	SQLIndex
✓ Entry No.			✓	✓		✓	
▶ ✓ Customer No.,Posting Date,Currency Code	Sales (LCY),Profit (LCY),Inv. D...		✓	✓	{Custome...		Customer No.,Posting Date,En...
Customer No.,Currency Code,Posting Date		Cust(Curr)	✓	✓			
✓ Document No.			✓	✓			
✓ External Document No.							
✓ Customer No.,Open,Positive,Due Date,Currency Code							Due Date,Entry No.
✓ Open,Due Date							
✓ Document Type,Customer No.,Posting Date,Currency Co							
✓ Salesperson Code,Posting Date							
✓ Closed by Entry No.							
✓ Transaction No.							
Customer No.,Open,Positive,Calculate Interest,Due Date							
Customer No.,Global Dimension 1 Code,Global Dimension							
Customer No.,Open,Global Dimension 1 Code,Global Dime							
Open,Global Dimension 1 Code,Global Dimension 2 Code,D							Due Date,Global Dimension 1 ...
Document Type,Customer No.,Global Dimension 1 Code,G							
✓ Customer No.,Applies-to ID,Open,Positive,Due Date							

Key - Properties

Property	Value
Enabled	<Yes>
Key	Customer No.,Posting Date,Currency Code
SumIndexFields	Sales (LCY),Profit (LCY),Inv. Discount (LCY)
KeyGroups	<>
MaintainSQLIndex	<Yes>
MaintainSIFTIndex	<Yes>
SIFTLevelsToMaintain	{Customer No.},{Customer No.,Posting Date:Year},{Custo...
Clustered	<No>
SQLIndex	Customer No.,Posting Date,Entry No.

Help

Usefull Tools & Scripts & DMV's



- ❑ The dynamic management views (DMVs) in SQL Server 2005 → **what's going on inside** SQL Server ?
- ❑ They are designed to be **used instead of system tables** and the various **functions** provided in SQL Server 2000.
- ❑ The Dynamic Management Views are actually composed of **both views** and **table-valued functions**.
- ❑ All are stored in the **sys** schema.
- ❑ They all start with **dm_** in the name.

Usefull Tools & Scripts & DMV's



- **12** categories:
- Common Language Runtime Related Dynamic Management Views
- **I/O** Related Dynamic Management Views and Functions
- Database Mirroring Related Dynamic Management Views
- Query Notifications Related Dynamic Management Views
- Database Related Dynamic Management Views
- Replication Related Dynamic Management Views
- **Execution** Related Dynamic Management Views and Functions
- Service Broker Related Dynamic Management Views
- Full-Text Search Related Dynamic Management Views
- SQL Operating System Related Dynamic Management Views
- **Index** Related Dynamic Management Views and Functions
- Transaction Related Dynamic Management Views and Functions

- **sys.dm_os_performance_counters**
 - all the counters that are pulled in PerfMon for the SQL Server category
- **sys.dm_db_index_physical_stats**
sys.dm_db_index_usage_stats
 - explain how much an index is used. It makes a great place for finding indexes that aren't needed.
- **sys.dm_db_missing_index_details,**
sys.dm_db_missing_index_group_stats,
sys.dm_db_missing_index_groups
 - identify indexes that are needed on tables.
 - index_group_stats is helpful because it identifies how many times the index could have been used
- **sys.dm_os_wait_stats**

Usefull Tools & Scripts & DMV's



- ❑ As you can see the “**sys.dm_os_performance_counters**” DMV allows you an **easy** method to get at SQL Server **performance counters** using a simple SELECT statement.
- ❑ You can use this DMV to develop a set of queries that you **periodically** run to monitor performance counters.
- ❑ By **routinely monitoring** performance counters, and **reviewing** the counters to see if they are similar to your last **counter snapshot**, you can quickly identify if your server is having **performance** issues.

Usefull Tools & Scripts & DMV's



- Database Resource Kit (PartnerSource)
 - Index Defrag Tool
 - Key Information Tool
 - Database Resource Kit.pdf
 - ...
- Tools_IndexQueries.sql
- SQL Server 2005 Performance Dashboard Reports
- ...

- The SQL Server 2005 **Performance Dashboard Reports** are **Reporting Services** report files designed to be used with the **Custom Reports feature** introduced in the **SP2** release of SQL Server **Management Studio**.
- Common performance problems that the dashboard reports may help to resolve include:
 - **CPU** bottlenecks (and what queries are consuming the most CPU)
 - **IO** bottlenecks (and what queries are performing the most IO).
 - **Index** recommendations generated by the query optimizer (missing indexes)
 - **Blocking**

SQL Server 2005 Performance Dashboard Reports



- ❑ The information captured in the reports is retrieved from SQL Server's **dynamic management views**.
- ❑ There is no additional tracing or data capture required, which means the **information is always available** and this is a **very inexpensive** means of monitoring your server.
- ❑ **Reporting Services is not required** to be installed to use the **Performance Dashboard Reports**.
- ❑ (DEMO)

Index Hinting

- ❑ On SQL Server, you can use index hinting to force the server to **use a particular index** when executing queries for **FINDFIRST, FINDLAST, FINDSET, FIND('-'), FIND('+'), FIND('=')** and **GET** statements.
- ❑ Index hinting can help avoid situations where SQL Server's Query Optimizer chooses an index access method that requires **many page reads** and generates **long-running queries** with response times that vary from **seconds** to several **minutes**.
 - Directing SQL Server to use a specific index can give **instant 'correct' query executions** with response times of **milliseconds**.

Index Hinting

- ❑ In Microsoft Dynamics NAV, index hinting is **turned on by default** and the application **automatically** uses this functionality to **improve performance**.
- ❑ If you need to **switch off** or **customize** index hinting to fit your implementation, you must **create** a SQL Server **table** to store the configuration **parameters**.
 - The parameters you enter into this table will determine some of the **behavior** of Microsoft Dynamics NAV when it is using this database.
- ❑ In the database create a table, owned by dbo:
CREATE TABLE [**\$ndo\$dbconfig**] (config VARCHAR(512)
NOT NULL)
GRANT **SELECT** ON [**\$ndo\$dbconfig**] TO **public**

Index Hinting

- The default value is IndexHint=Yes.
- You can disable index hinting at any level of granularity.
- There are two ways of using index hinting in your application:
 - You can leave index hinting turned on and disable it in specific places.
 - You can turn off index hinting and enable it in specific places.

Index Hinting Benefits

- Index hinting has been shown to optimize performance in the following scenarios:
 - Index hints **prevent** SQL Server from using an **out of date query** plan, such as a clustered index scan.
 - Index hints **prevent** SQL Server from **scanning** smaller tables and **escalating locks** to table locks.

Index Hinting Example

- In the following C/AL code, index hinting is turned on but SETCURRENTKEY has not been used:

```
GLEntry.SETRANGE("G/L Account No.", '2910');  
GLEntry.FINDSET;
```

- This will generate the following SQL query:

```
SELECT TOP 500 * FROM "W1403"."dbo"."CRONUS  
International Ltd_$G_L Entry" WITH  
(READUNCOMMITTED, INDEX("CRONUS International  
Ltd_$G_L Entry$0")) WHERE (("G_L Account  
No_" = @P1)) ORDER BY "Entry No_" ', '2910'
```

- Note that **without** a **SETCURRENTKEY**, Microsoft Dynamics NAV will hint the SQL index which corresponds to the **primary key** in the G/L Account table. This is **not the best key** to use for this query.

Index Hinting Example

- Conversely, in the following C/AL code, hinting is turned on and **SETCURRENTKEY** has been used:
GLEEntry.SETCURRENTKEY("G/L Account No.");
GLEEntry.SETRANGE("G/L Account No.", '2910');
GLEEntry.FINDSET;
- This will generate the following SQL query:
SELECT TOP 500 * **FROM** "W1403"."dbo"."CRONUS International Ltd_\$G_L Entry" WITH (READUNCOMMITTED, **INDEX("\$1")**) **WHERE** (("G_L Account No_"=@P1)) **ORDER BY** "G_L Account No_", "Posting Date", "Entry No_" ', '2910'
- Now, because the **C/AL** code **specifies which key** to use, Microsoft Dynamics NAV **hints** the corresponding index from the code, which ensures that the right **index** is always used.

Fill Factor

- ❑ When you create an index, the data in the table is stored in the data **pages** of the database according to the **order** of the **values** in the **indexed columns**.
- ❑ When new rows of data are **inserted** into the table or the values in the indexed columns are **changed**, SQL Server may have to **reorganize** the storage of the data in the table to make room for the new row and maintain the ordered storage of the data.
- ❑ When a new row is added to a **full index page**, SQL Server **moves** approximately **half** the **rows** to a **new page** to make room for the new row.
- ❑ This reorganization is known as a **page split**.
- ❑ Page splitting can **impair performance** and **fragment** the **storage** of the data in a table.

Fill Factor

- When **creating** an **index**, you can **specify** a **fill factor** to leave extra gaps and reserve a percentage of free space on each leaf level page of the index → **reduce** the potential for **page splits**.
- The fill factor value is a **%** from **0** to **100** that specifies how much to fill the data pages.
- **100%** means the **pages** will be **full** and will take the least amount of storage space.
 - This setting should be used only when there will be no changes to the data, for example, on a **read-only** table.
- **< 100%** leaves more empty space on the data pages, but requires more storage space.
 - This setting is more appropriate when there will be changes to the data in the table.

Form design & performance

- SIFT
 - Avoid FlowFields on List Forms ← calculated @runtime (even hidden ones)
 - Display on demand

- SourceTablePlacement property
 - Saved → First or Last

- Find As You Type feature

Locking & Deadlocks

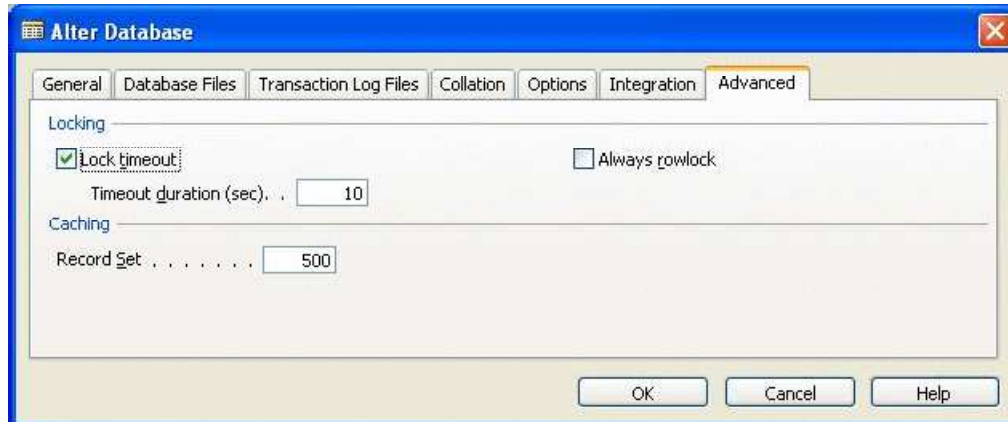
- ❑ Deadlocks
- ❑ LOCKTIMEOUT
- ❑ Minimizing the Duration of Locks
- ❑ Tools

- ❑ Tip: focus on **performance** before looking into locks because improving performance might minimize locking

Locking & Deadlocks

- A **deadlock** occurs when two or more transactions have a conflicting locking order and no deadlock can occur if the first lock the transactions place is on the same table.
- Always lock tables in the **same order**.
- Lock an agreed "**master resource**" **first**.

LOCKTIMEOUT



- ❑ you can also use the C/SIDE function **LOCKTIMEOUT** to temporarily enable or disable this property in the application
- ❑ **Always rowlock: ?**
 - By default this property is not selected and SQL Server uses its default locking behavior.
 - This can improve **performance** by allowing SQL Server to determine the best locking granularity

RecordLevelLocking

Navision Database Server:

```
IF Rec.FIND('-') THEN
  REPEAT
    UNTIL Rec.NEXT = 0;
  Rec.LOCKTABLE(TRUE,TRUE);
  IF Rec.FIND('-') THEN
    REPEAT
      Rec.MODIFY;
    UNTIL Rec.NEXT = 0;
```

SQL Server:

```
Rec.LOCKTABLE;
IF Rec.FIND('-') THEN
  REPEAT
    UNTIL Rec.NEXT = 0;
  IF Rec.FIND('-') THEN
    REPEAT
      Rec.MODIFY;
    UNTIL Rec.NEXT = 0;
```

RecordLevelLocking

```
IF Rec.RECORDLEVELLOCKING THEN
  Rec.LOCKTABLE;
IF Rec.FIND('-') THEN
  REPEAT
    UNTIL Rec.NEXT = 0;
IF NOT Rec.RECORDLEVELLOCKING THEN
  Rec.LOCKTABLE(TRUE,TRUE);
IF Rec.FIND('-') THEN
  REPEAT
    Rec.MODIFY;
  UNTIL Rec.NEXT = 0;
```

Maintenance

- **Updating SQL Server Statistics**
- **Index Fragmentation**
- **Index Defrag Tool**
- **Maintenance Plan !**
- **Optimalisation**

- (DEMO)

Performance Tuning

- Quick Wins
 - Upgrade to current version
 - Hardware upgrade
 - Operating System
 - SQL Server setup
-

- Time Consuming
 - Index & SIFT Alterations
 - Index Tuning
 - Code Changes

Resources available

- <http://www.microsoft.com/sql>
 - <http://www.microsoft.com/sql/prodinfo/features/compare-features.aspx>
- <http://www.sqlskills.com> (Kimberly Tripp)
- <http://www.sqljunkies.com>
- <http://www.sqlteam.com>
- <http://blogs.msdn.com/sqlblog>
- <http://sqlug.be>

- <http://plataan.typepad.com/microsoftdynamics>

- ...

Training available

- http://www.plataan.be/en/navision/training_calendar/MicrosoftDynamicsNavSQLServeroptionTraining.htm
- This two-day Dynamics-Nav course covers in detail the integration between Microsoft Dynamics-Nav and Microsoft SQL Server. The course is designed for technical consultants who are installing and implementing Dynamics-Nav with the Microsoft SQL Server option. This course deals with the integration, implementation, performance optimisation, maintenance, tools, locking and troubleshooting.

The End...

- Thank you for attendance and participation.



Steven Renders
Microsoft Certified Trainer
Plataan
steven.renders@plataan.be