

# MaxDB online class

A collection of MaxDB postings  
on <http://planetmysql.org>  
starting from Februar 2006

Last update: 24.04.06

## Table of Contents

Preface.....	4
How to contact us.....	4
The MaxDB series on planetmysql.org.....	5
What is MaxDB by MySQL?.....	5
How do MaxDB and MySQL compare?.....	5
A guide on using MaxDB.....	6
Next issue: Overview on MaxDB.....	6
MaxDB Overview.....	7
In this issue.....	7
TOP 5 Features of MaxDB: No Need for Reorganisation.....	8
TOP 5 Features of MaxDB: Backup and Recovery.....	10
TOP 5 Features of MaxDB: Standby Solutions.....	11
TOP 5 Features of MaxDB: Synchronization Manager.....	12
TOP 5 Features of MaxDB: Database snapshots.....	12
Getting MaxDB.....	13
MaxDB software components.....	13
In the next issue.....	14
Installation.....	15
Selecting a version .....	15
MaxDB Installation Manager .....	16
Connecting to MaxDB using the Database Manager (DBM) GUI .....	22
Issuing queries with SQL Studio.....	24
Shutting down with Database Manager.....	24
MaxDB series: DBM GUI and dbmcli, SQL Studio and sqlcli.....	26
In this issue.....	26
The Database Manager.....	26
System Architecture.....	27
DBMCLI Usage Example.....	28
The Database Manager GUI.....	29
Remote connections: ensure the X Server runs and no firewall blocks communication.....	30
Learning DBM command using the Reverse Engineering trick.....	30
The SQL Tools.....	31
Using SQL Studio.....	33
Pitfalls, error messages and searchable manual variants.....	34
Summary.....	34
In the next issue.....	35
MaxDB series: User concept, authorization and schemata.....	36

In this issue.....	36
Three different user types in MaxDB.....	36
“root”: The Database System Administrator (SYSDBA user).....	37
“Administrator”: Database Manager operators (DBM operator).....	38
Managing DBM Operators.....	39
Example: creating a start-stop user.....	39
The User Profile Container.....	42
“SQL users”: Database users of the user classes DBA, RESOURCE and STANDARD.....	43
Hands on: Managing two SQL users.....	44
In the next issue.....	46
MaxDB series: SQL users continued .....	48
In this issue.....	48
User groups - CREATE USERGROUP.....	48
Roles - CREATE ROLE.....	48
Dinosaurs - DOMAIN, SYSINFO, PERMLIMIT, TEMPLIMIT.....	49
In the next issue.....	49
MaxDB series: when do we publish the next article? .....	50
Can I recover a MaxDB backup on a different version and/or a different system?.....	50
MaxDB series: ...just SQL.....	51
In this issue.....	51
Data types for table columns.....	51
Check Constraints.....	53
Domains.....	53
Synonyms.....	53
Indexes.....	54
Sequences and SERIAL.....	54
Limiting result sets.....	55
Cursors and Recursion.....	56
In the next issue.....	58
MaxDB series: news but no series posting.....	59
MaxDB series: Transactions.....	60
In this issue.....	60
Subtransactions.....	60
Concurrency, Locking and Isolation levels.....	62
Implementation details.....	63
In the next issue.....	64

## Preface

This is a collection of selected MaxDB blog postings published on <http://planetmysql.org>. The collection contains postings from Februar 2006 until today. All postings of the series make a online class.

We are providing the original postings „as is“. Some of the postings have not undergone intensive technical-proof reading. Of course the authors tried their best not to make any mistakes. But we do not give any warranty that the materials are on the same high technical level as the MaxDB manual or the official SAP and MySQL class materials.

## How to contact us

We encourage you to ask questions on MaxDB in general and this document in particular. We will be glad to answer your questions. Please use one of the following ways to contact us.

Feedback on the online class and this document: <http://forums.mysql.com/list.php?41>

Sales contact: <http://www.mysql.com/company/contact/>

## [The MaxDB series on planetmysql.org](#)

Published: 03.02.2006

**Dear MySQL users, MaxDB users and friends,**

We welcome you to a series of MaxDB postings on [planetmysql.org](#).

Over the next weeks the MySQL MaxDB team will present [MaxDB](#) to you. We plan to write one posting per week, starting with today. Blog entries will be published every wednesday as our work constraints permit. All postings together might make a complete and comprehensive MaxDB course.

### **What is MaxDB by MySQL?**

[MySQL AB](#) delivers two enterprise-level database systems. The first database is the [MySQL Server](#). With more than 8 million active installations, it is the world's most popular open source database. The latest General Available (GA) Release is MySQL 5.0. The freely downloadable version is called [MySQL 5.0 Community Edition](#). Three different binary versions of the MySQL Server exist: Standard, Max and Debug. Do not mix up the Max version of the MySQL Server with MaxDB by MySQL.

MaxDB by MySQL is the second enterprise-level relational database management system delivered by MySQL AB. Formerly known as [SAP DB](#), MaxDB is the result of a [strategic alliance between MySQL and SAP](#) to jointly develop and market an open source database for SAP/R3 .

The database development started in 1977 as a research project at the technical University of Berlin. In the early 80s it became a database product that subsequently was owned by Nixdorf, Siemens Nixdorf, Software AG and today by [SAP AG](#). Along this line it has been named VDN, Reflex, Supra 2, DDB/4, Entire SQL-DB-Server and ADABAS. In 1997 SAP took over the software from Software AG and renamed it to SAP DB. In 2003 SAP AG and MySQL AB joined a partnership and re-branded the database system to MaxDB. - From this background on the MaxDB history you won't be surprised that the MaxDB database and the MySQL Server are two very different products.

### **How do MaxDB and MySQL compare?**

Let's check which MySQL 5.0 flagship features can be found in MaxDB. MaxDB has Stored Procedures, Triggers, Views, Precision Match and an Information Schema counterpart. These features have been available in MaxDB for a decade.

MySQL 5.1 will offer a new [Eventing and Scheduling feature](#). The service oriented architecture of MaxDB has introduced a scheduler in 7.5, which was released around 2004. Additionally MaxDB has an eventing feature that goes beyond temporal trigger of MySQL 5.1. MaxDB pro-actively warns the DBA if they have to perform some administrative tasks. For example MaxDB tells you if the database filling level falls below or goes above a certain fill-level. MaxDB offers built-in non-blocking hot backup. MaxDB is SAP certified, and, and, ...

On the other hand does MaxDB has no counterpart to the unique [Pluggable Storage Engine Architecture](#) that the MySQL Server offers for maximum flexibility. For example, there's no Archive Storage Engine for historical and audit data, that is compressed on-the-fly. MaxDB does not offer Partitioning, which will be available in MySQL 5.1 nor does it have a counterpart to the [MySQL Cluster](#) but MaxDB does support High Availability and Hot Standby setups.

The comparison could be easily extended. Depending on the focus of the writer, a comparison chart would show MaxDB ahead of MySQL or just the other way around with MySQL being better than MaxDB.

And features are not everything that counts. In the end only you can answer the question which system fulfills your needs. We encourage you to try out both systems! Both are enterprise-level, but have different optimizations and strengths, both are [dual-licensed](#) and thus, you can use them under the terms of the [Free Software GPL licence](#).

## A guide on using MaxDB

The MySQL MaxDB team hopes that we can create a guide that explains how to try out and use MaxDB in this article series. Note that we will not discuss how to use MaxDB with SAP applications. This series is about MaxDB when not used with SAP applications.

Before we start: we need your feedback! Every author needs feedback from the audience. All readers have different skills and expectations on an article. We do not know about your expectations. Please use the [MaxDB Forum](#) to tell us what you would like to see in the series.

## Next issue: Overview on MaxDB

We plan to publish the first issue of our series on Wednesday, the 8th of February, 2006. The posting will be about MaxDB in the present and the future, today's hottest MaxDB features, where to find more information on MaxDB, and we will try to give a first overview of the MaxDB software components.

Talk to you next Wednesday!

*Ulf Wendel for the MySQL MaxDB team*

## MaxDB Overview

Published: 09.02.2006

**Dear MySQL users, MaxDB users and friends,**

last week we announced a MaxDB series on [planetmysql.org](http://planetmysql.org) with weekly postings every wednesday. This is the first posting with “real” content, it’s the first time we do this for you. We spend a considerable amount of time on making it as good as possible, but we know it won’t be perfect. Please help us to improve in the future and use the [MaxDB Forum](#) to tell us more about your expectations on us and how we can improve. Everybody at [MySQL AB](#) is eager to learn about your needs.

The first “real” posting has become a little theoretical and less practical. The reason is that we wanted to explain why you should use MaxDB first and what the concepts of the system are. If you don’t know what the MaxDB software components do, you will have difficulties to learn how to use MaxDB.

### **In this issue**

In this first issue you will be presented a first overview of the MaxDB software, its components, today’s coolest features and the future of the product. No installation guide? No, not yet. You need to learn some terms and concepts of the MaxDB world first, before a step-by-step installation can be performed in the next issue on next wednesday.

- [Today and tomorrow: Quo vadis, MaxDB?](#)
- [TOP 5 Features of MaxDB: No Need for Reorganisation](#)
- [TOP 5 Features of MaxDB: Backup and Recovery](#)
- [TOP 5 Features of MaxDB: Standby Solutions](#)
- [TOP 5 Features of MaxDB: Synchronization Manager](#)
- [Getting MaxDB](#)
- [MaxDB software components](#)
- [In the next issue](#)

### **Today and tomorrow: Quo vadis, MaxDB?**

At end of the last year everybody looked back and reported about their achievements in 2005. [EnterpriseDB](#) announced that it had [70,000 downloads in six months](#) after the company’s public launch on May 23rd. Congratulations! People talked a lot about this new offer based on [PostgreSQL](#). MaxDB gets downloaded about 50,000 times per month but it gets much less attention. MaxDB users, do not hesitate to ask questions on the [MaxDB mailinglist](#), blog about your database, participate to the [freenode #maxdb](#) IRC Channel and show everybody that MaxDB is alive.

More than 120 employees of the [SAP AG](#) work on the MaxDB technology stack (see also [MaxDB and SAP liveCache](#) in the [SAP Developer Network](#)). MySQL AB actively supports the MaxDB development. Last year we announced to continue the Precompiler Development and we have more in the pipe.

**MaxDB is the world’s first SAP certified open source database** used in 3,500 SAP customer installations worldwide. Some figures show what it means to be SAP certified. SAP certified databases have passed several 10.000 quality assurance test cases, are able to run databases with more than 40,000 tables and views and can consist of terrabytes of data.

MaxDB stands for scalability and stability. Once you are SAP certified a tremendous pressure is on your shoulders. You cannot ask your customer to dump a database before an update and reimport it afterwards, because you changed some storage details. That is not acceptable in the SAP business world. This is disappointing for feature-hungry open source users, but relaxing for business users.

MaxDB is a secure investment. It has been a secure investment in the past, it is one today and it will be one tomorrow.

The database is distributed under the terms of the MySQL [dual-licence principle](#). You can use it for free if you go for the GPL way, buy a commercial licence for 1290 Euro per CPU and year or for 49 Euro per user. And you can also buy MaxDB for SAP Applications from MySQL - [contact us](#). A full set of Professional Services are available. Support, Training and Consulting are offered by MySQL and by SAP. Compare the prices before you spend too much money, money is always a short resource.

If you want to learn about the Total Cost of Ownership of MaxDB, [ask for our TCO papers](#) or read the articles [MaxDB reduces total cost of ownership: High Professionalism, Low Operating Costs](#) and [MaxDB in hosting environments: Secure Operation for Users and Developers](#) on [SAP INFO](#)

## TOP 5 Features of MaxDB: No Need for Reorganisation

MaxDB does not need any kind of reorganisation, similar to MySQL's InnoDB storage engine. There is no `OPTIMIZE TABLE`, no `VACUUM`, no `COMPRESS` or similar command available in MaxDB, because it is not needed. In other words:

- *no scheduled downtimes*
- *no slowdowns during reorganisation*
- *no impact on your business*
- *no DBA resources needed*

MaxDB has these advantages ever since. In fact, it has been designed with these goals in mind. No Need for Reorganisation is a very technical topic. Explaining the techniques and tricks of MaxDB requires some theoretical background. If you are not interested in reading about it, check the following the high-altitude list of benefits and skip the paragraph.

- no gaps between data records
- no unused space
- *no longer needed space gets freed and reused immediately*

To understand why MaxDB does not need any reorganization, you need to know some basics about how databases work internally. You need to know what a "database page", a "file" and a [B-Tree \(balanced tree\)](#). We will explain this, step by step. Continue to [TOP 5: Backup and Recovery](#) if you trust what we say.

### Some theoretical background

Internally a database has a different, more abstract view on the data it stores than you have. Databases usually translate a SQL table (a relation in the relational data model) into an internal file. A file is a collection of smaller units, of "database pages". Records of your SQL tables (tupels in te relational data model) are stored on the database pages. A page is a consequent chunk of storage. In case of MaxDB the size of a database page is 8kB. The size can be changed when recompiling MaxDB. It will be shown later how MaxDB does all required reorganizations of a page on the fly.

Don't get confused at this point: what a database calls a file internally, does not need to be the same as a file on your harddisk. Most database systems use a "file manager" component to decouple physical storage details from the higher level components of the database.

Databases could store records in arbitrary order on the database pages that belong to the file that represents the table to which the records belong. The result would be kind of an unsorted list of records. Unsorted lists as a data structure do not support search operations much. Searching for a record stored in an unsorted list requires scanning over all list elements in order to compare them to the searched value. Programmers and mathematicans use the term "[O](#)" ([Big O](#)) to describe the time complexity of an algorithm like a search operation. The search operation for a [list](#) has the time complexitiy  $O(n)$ .  $O(n)$  means, that the time is a function of  $n$ , which is the number of list elements. In simple words: if it takes 1 second to search a list with 1.000 elements, it will take 10 seconds to search a list with 10.000 elements and 100 seconds to search a list with 100.000 elements. Other data



structures exist that can be searched much quicker. B-Trees and [Hash tables \(hash maps\)](#) can be searched much faster. The time complexity for B-Tree search operations is  $O(\log n)$ . Is that any better than  $O(n)$  of the list? Yes, it is. See how we can search growing data sets with only little more time.  $O(\log 1.000)$  is 3, let's assume it's 3 seconds. Now, put time times as many entries as before in the tree:  $O(\log 10.000)$  is 4,  $O(\log 100.000)$  is 5. This means 3 seconds for 1.000 entries and only 5 seconds for 100.000. Hash tables can do even better with  $O(1)$ : 1 second to search a set of 1.000 entries, 1 second for 100.000. But unfortunately hash tables have other disadvantages which makes B-Trees the most commonly used data structure for search operations of a database.

MaxDB is using a variation of a B-Tree called [B\\*-Tree](#). B\*-Trees tend to have a little higher fill-level as other variations of B-Trees and their leaf nodes are chained in form of a linked list. But you do not need to worry about these details. For you it is important to remember that B-Trees are a very good data structure when you need fast search operations. B-Trees itself are just trees, but they are sorted and their branches are balanced. All branches have the same depth. We will unveil some more properties of B-Trees when we explain how indexes can speed up some of your queries.

## MaxDB does B-tree balancing on the fly

MaxDB performs all B-Tree balancing operations, all maintenance operations whenever they become necessary. This is a must, because a "degenerated" or "unbalanced" B-tree is no longer an efficient data structure for search operations. Balancing operations can become necessary when you add records to a table, alter records of a table or remove records from a table. MaxDB is using on B\*-tree for every table. All index nodes and leafes of the B\*-trees are made out of database pages. The leaf nodes of the tree hold your records. If you add a new record to your table, MaxDB will insert it into one of the leaf nodes of the B\*-tree that belongs to the table. As said, B-Trees are sorted and so goes your new record into a certain leaf node. It might happen that there is not enough space left on the leaf node, on the database page where your record is to be stored. In this case, MaxDB will add a new node to the tree which can make it necessary to rebalance a branch of the tree or the entire tree. Similar operations are performed automatically when you remove or alter records. MaxDB B\*-trees are always in an optimal shape for optimal performance.

## Update in place, sort by insertion, delete in place

The counterpart to balancing operations on the level of the B-trees are the operations update in place, sort by insertion and delete in place on the level of the database pages. To explain the operations one needs to know how MaxDB stores records on a data page. Each data page has a size of 8kb by default. A page is divided into three areas. Every data page starts with a list of unsorted data entries, followed by free space. At the end of the data page a sorted list of data entry pointers gets stored. The list of data entry pointers, the position list, is sorted in key order. That means the database can perform a fast sequential scan when it fetches records in ascending or descending key order.

When a data entry gets inserted into a page it gets appended to the end of the list of data entries on the page. The pointer to the newly added data entry gets inserted into the pointer list in a way that the pointer list remains sorted. Sorting is performed on the smaller of the two possible sort lists. MaxDB will not sort the data entries itself but it will sort the much smaller pointer list. Of course, sorting the smaller pointer list is faster than sorting the data entries. This is what MaxDB means by "sort by insertion".

However, if a page overflows or underflows during an insert and a B\*-tree balancing operation has to be performed, MaxDB will sort the data entries. Why? Having sorted data entries is of advantage when merging pages. Whenever possible MaxDB tries to handle updates of data entries on their current page. This is called "update in place". If a data entry gets updated and its overall length and its key value do not change, then MaxDB does not need to do anything but update the data entry itself. No changes to the position list are required because the position list is sorted by the key values and the key values have not changed. A simple update of the data entry in its current place is possible, it still fits into the existing slot.

More work needs to be done if the length of a data entry changes but the key value remains the same. If the size changes, then the slot that is occupied by the data entry is either too small and needs to be

increased or it is too large and should be shrunk in order to save space and avoid gaps. Resizing a slot means shifting forth or back all data entries that follow the resized entry. And, whenever the position of a data entry changes, MaxDB has to update the pointers in the position list.

If a data entry remains the same size but the key value gets changed, then there is little chance to handle this operation on page level. Each page, each leaf node in the tree, stores the records of a certain key value range. If the modified key value no longer matches the key value range of the original page, then the data entry has to be moved to another page which holds the matching key value range.

It is unlikely that the new key value can be kept on the original page. Therefore MaxDB does not try to handle an update with a key value change on a page, but transforms the update operation into a delete followed by an insert. Delete in place means that MaxDB will not juggle data entries on a page if an entry gets deleted. MaxDB will only update the position list, that is all.

## Compare it to the rivals

Check with your database experts how your system does it all and if you need to reorganize or optimize your storage system frequently to regain optimal performance. Some systems do need such optimizations, some need extra B-tree optimizations, some need it for the database pages, some for other areas and reasons. For example, EnterpriseDB (and Postgres) needs [Vacuuming](#) to remove already deleted records from the database files. As of Postgres 8.1 you can use a separate optional server process called the autovacuum daemon to automate this maintenance task. This is a possible solution, but not the best. The problem with it is that you have to do the house-keeping anyway at some point. You can stall it for a while, but your house will be soon a place where you do not want to live in anymore. Once you have reached this point you can spend a lot of time on the cleaning or you settle over to a new house. Both solutions will block you and require tremendous efforts. MaxDB does the house-keeping all the time. Do you want possible downtimes and peak-loads for cleaning up your database storage. Or do you want to do the work on a day-by-day basis and spend a little bit of time every day? MaxDB and InnoDB do it the day-by-day way. The result: no downtime, no need for reorganisation, always optimal storage, less DBA resources required.

## TOP 5 Features of MaxDB: Backup and Recovery

Backup and Recovery is full-fledged and feature-rich. All major backup strategies are supported. Integration into third-party tools and large-scale Backup systems is smooth.

- *Non-blocking, consistent Hot Backup*
- *Automatic Log Backup*
- *Support for consistent copies on the snapshots*
- *Im-/Export (SQL, CSV, ...): Loader tool*
- *Support for third-party tools*
- *GUI and CLI interfaces*
- *Functions to check and verify backups*

### Non-blocking, consistent Hot Backup

MaxDB can create data backups when the database is running. The backup operation does not block any running transactions. The use of extra server tasks ensures that the impact on other tasks running at the same time on the server is as small as possible. The data backup is transaction-consistent because it is based on a savepoint (checkpoint).

You can create full data backups and incremental data backups. Incremental backups are based on full backups. Once you have created a full backup, you can use incremental backups to backup the difference in the database between the last full or incremental backup. In most cases the delta is much smaller than a full backup.

## Automatic Log Backup

MaxDB can perform log backups automatically. Whenever a segment of the MaxDB log area has been filled an automatic log backup is done. The backup works asynchronously and does not block any other log writer activities. The size of the log segments can be reconfigured to enforce more or less frequent backup runs. In general you do not need to be too paranoid on this. If you used hardware-based mirroring or the MaxDB software-based log mirroring as we strongly recommend, then you do not need to worry too much about losing many log entries in case of a catastrophe (e.g. disk-failure). You do not need, because the log is mirrored and in most cases you still have one good copy left.

Full data backups, incremental data backups and automatic log backups together give you a wide variety of recovery options. In some cases you can even perform a recovery if you have lost an incremental data backup but you still have the log backups. We will go into the details in another issue, when we discuss how to make a backup

## Support for consistent copies on the snapshots

During the last years consistent copies of the filesystem have become popular. Specialized hardware and software can create a copy of very large filesystems extremely fast. Sometimes even faster than any other backup method. You can take a consistent copy of your MaxDB database while the database is running. Note that, if you do create a copy while the database is running, the database might perform an automatic recovery if you put the copy back in place and start the database again. There might have been transactions running when you did the copy. These unfinished transactions will be automatically rolledback when you restart the database.

A way to circumvent this is to create a consistent database snapshot inside the database before you do the copy. Once you have recovered the copy, you can revert the consistent database snapshot.

## More on Backup & Recovery

There is a lot more to say about backup and recovery features in MaxDB. The Loader, the MaxDB data import and export tool, has a new wizard mode as of 7.6.00. It makes using the Loader for backup as easy as using mysqldump, if not easier. MaxDB has interfaces to support third-party backup solutions, all administrative commands can be performed using CLI or a GUI and you can check and verify your backups. Don't worry that this all makes MaxDB overly complicated. It is not and you can start with easy to use wizards, if you want. This is still not all, but we guess you are eager to hear about the other TOP 5 features.

## TOP 5 Features of MaxDB: Standby Solutions

MaxDB offers two standby solutions a standby database and a hot standby solution. You do not need any extra software to set up a standby solution. MaxDB comes with all that you need. Both standby variants can be integrated in cluster solutions for automatic IP-switching and client redirection.

- *build-in*
- *standby database based on log-shipping*
- *hot standby solution*
- *integration into cluster setup possible*

### Standby database

Setting up a standby database using the GUI wizard is something you get taught on the second day of the [MaxDB Administration class](#) offered by MySQL. It takes 10 minutes to show the slides, 15 minutes to make all possible mistakes a beginner can make and another 5 minutes until the trainer comes along and corrects the problems. All in all it takes in the worst case it takes 30 minutes to set up.

The reason it is so easy is that there is no rocket science behind. To setup a standby database you need a full data backup of your original database. This full data backup is used to create a copy of your original database. After recovering the backup on your standby database you ship the log backup from the original database to your standby database to recover them. If the main system fails,

This process can be done manually using the GUI tools. With help of the CLI interfaces it can be automated. Third-party companies have created solutions that can manage many different standby databases for one production system. As the frequency of the log shipping determines how many minutes the standby database is “behind” the production systems, you can set up standby systems that hold an old copy of the data of the production system. In case of a logical mistake, for example an accidental `DROP`, you can have a standby system already in place that holds a copy of your data from before the mistake. You only need to know when the mistake was made to select the standby systems that still has a copy of the original data.

## Hot Standby

In an Hot Standby instance a MaxDB master and a slave share the log area. Each database runs on its own machine and is using his private data area. But the log area is shared. Immediately after the master has written log entries to the shared log area, the slave starts to replay the log. Therefore the slave is always only seconds behind the master. If the master fails, a cluster solution can perform the IP- and client switch over, while the slave replays the latest log entries before it can become the master.

## TOP 5 Features of MaxDB: Synchronization Manager

With version 7.6.00 the [Synchronization Manager](#) was introduced. The Synchronization Manager is a replication tool. It replicates tables from a master database to client databases. But the tool can do more: client databases can send messages back to the master even if they got disconnected for some time. This can be exactly what you are looking for. Guess you need to synchronize mobile devices of your field sales personell with a central database. The Synchronization Manager can do this for you if you run MaxDB, MinDB or - in the future - MySQL on the mobile devices. Your field sales can download customer details from the central database and upload changes to customer records or orders.

- *bi-directional replication*
- *conflict resolution*
- *asynchronous: replication continues after reconnect*
- *GUI-based admin tool*

Try it out, we have written an [article on how to install the Synchronization Manager](#).

## TOP 5 Features of MaxDB: Database snapshots

MaxDB can freeze in the current state of the database in a database snapshot. Upon creation of a snapshot the database marks all existing data as read-only. If you modify the data after the snapshot has been created, the modified data cannot overwrite the existing read-only data but gets stored in a different place. Whenever you want you can get back original state of the database you can revert the snapshot. Taking a snapshot and reverting to it are very fast operations as everything happens inside the database. Unlike with backup and recovery, you do not need to export and re-import data from an external medium.

- *database snapshots*
- *fast way to recover the original state of the database*

Snapshots are handy in many situations. For example you can use them for your training classes. You take a snapshot from your demo database before the class starts, allow the trainees to alter everything in it during the class and after the class you can revert the snapshot. As said, this is much faster than backup and recovery. Another example: you are doing an application upgrade. Before you install the

new version of the software, you create a database snapshot. If the software upgrade runs fine, you delete the snapshot. If it fails, you reinstall the old version of the software and revert the snapshot. Again, your benefit is the time needed to restore a previous state of the database.

I hope we could finally raise your interest in MaxDB. Try it out.

## Getting MaxDB

The latest General Available (GA) version of MaxDB is 7.6.00. The latest Build is 16 and the version string is 7.6.00.16. You can download the latest version of MaxDB 7.6.00 from <http://dev.mysql.com/downloads/maxdb/7.6.00.html>. MaxDB 7.5.00 is also a General Available (GA) version of MaxDB. If you start with MaxDB, you should go for 7.6.00. MaxDB 7.6.00 has many advantages over 7.5.00. For example SQL schema support, the Synchronization Tool, advanced monitoring and alerting and much more. Previous versions of MaxDB can be found in the MaxDB software archive on [http://dev.mysql.com/downloads/maxdb/archives/archive\\_index.html](http://dev.mysql.com/downloads/maxdb/archives/archive_index.html)

For your first steps with MaxDB you need to download one of the “MaxDB Installer” packages for your operating system. MaxDB 7.6.00 can be run on Windows, Linux, HP-UX, IBM AIX and Sun Solaris. If you have a Windows system available you should also Download the GUI administration tools “DBMGUI” and “SQL Studio”. We will provide detailed installation instructions for the software in the next issue of our series.

MaxDB documentation is available on <http://dev.mysql.com/doc/maxdb/index.html>. Note that the [Online Documentation](#) is about 7.6.00. The documentation does not have any notes how things have been done in previous versions. If you are using 7.5.00, get a copy of the 7.5.00 documentation from the [download archive](#). Many more places exist in the internet to learn about MaxDB and to discuss MaxDB questions:

- [MaxDB FAQ](#)
- [MaxDB mailing list](#)
- [MySQL MaxDB Forum](#) focussing mainly on MaxDB outside SAP environments
- [SAP Developer Network \(SDN\)](#) focussing mainly on MaxDB in SAP environments
- [MaxDB Wiki, mainly written by MaxDB developers](#)
- [SAP Info \(the SAP customer magazine\)](#), features MaxDB articles from time to time

## MaxDB software components

The first time you work with MaxDB you will be confronted with many new terms and software components that you might not know. The hearth of the MaxDB database is the kernel. The kernel does the SQL processing and manages the data and log area. You should never manipulate the data and log area on your own, unless the development support gave you detailed instructions. If the data and log area gets modified by any other software than the Kernel, you risk loosing all your data.

SQL commands can be send to the kernel using the sqlcli, SQL Studio or application clients that use one of the programming interfaces (JDBC, ODBC, Perl, PHP, Phyton, SQLDBC [C/C++]). The sqlcli is a command line interface (CLI) to enter SQL commands. It communicates directly to the kernel. The SQL Studio is a GUI application to run SQL commands. Currently the SQL Studio is only available on Windows, but you can expect new, updated GUI tools in future releases that will be available on more operating systems. .SQL Studio connects to the kernel through ODBC, similar to any other application client using ODBC. The ODBC driver in turn talks to the kernel.

Database Administrators can use the SQL tools (SQL Studio, sqlcli) to create users, tables, indexes and to grant access rights. Application programmers can use the SQL tools to inspect the structure of SQL object, test and optimize statements.

Administrative tasks can be performed with the Database Manager GUI (DBMGUI) application (available only on Windows) and the dbmcli command line tool. Both tools talk to the Database Manager component which in turn talks to the kernel. Administrative tasks are tasks like creating new database instances, creating backups or restoring backups.

All remote connections are handled by the X Server. No matter which tool you use to talk to the kernel (SQL Studio, sqlcli, DBMGUI, dbmcli), if the tool does not run on the same host as the kernel, all communication will be done through the X Server.

The following table gives a brief overview how the tools can be compared to tools of other systems.

	MaxDB	MySQL	Oracle	DB/2
<b>SQL Tool</b>	SQL Studio	MySQL Query Browser	SQL Plus	Command Center
<b>Administration Tool</b>	Database Manager GUI	MySQL Administrator	Enterprise Manager	Control Center

So, what do you need if you want to start with MaxDB on your own, before we finally present installation instructions and start with the practical parts in the next issue? You need the MaxDB Server software and - if you are on Windows - the SQL Studio and Database Manager GUI. On all other operating systems you have to use the command line interfaces (sqlcli, dbmcli). If you run the MaxDB server (kernel) on a different host than the GUI tools, then you have to make sure that your X Server is running.

## In the next issue

The next issue will give step-by-step instructions how to install MaxDB. We will try to publish the issue on next wednesday, but as you can see from this slightly delayed posting, we cannot promise that it will be available on wednesday.

We appreciate your comments. Use the [MaxDB Forum](#) to contact us.

*Ulf Wendel for the MySQL MaxDB team*

## Installation

Published: 22.02.2006

### **Hello MaxDB users, administrators, hobbyists and community!**

This, the third of our weekly articles on MaxDB, will cover the installation of the MaxDB server and associated tools. One of the features the team is most proud of is MaxDB's ease of installation and the speed with which it occurs. It's been estimated that a DBA with no prior knowledge of the system can complete a basic installation in less than 15 minutes. Please note that this does not include download time.

- [Selecting a version](#)
- [MaxDB Installation Manager](#)
- [xserver](#)
- [Connecting to MaxDB using the Database Manager \(DBM\) GUI](#)
- [Issuing queries with SQL Studio](#)
- [Shutting down with Database Manager](#)
- [In the next issue](#)

### **Selecting a version**

Before beginning the installation, one must decide which version of the software to install. If this is your first experience with MaxDB, I strongly recommend the newest and best. As of February 17, 2006, this is build 16 of version 7.6.00, or 7.6.00.16. If you are upgrading from an already-installed version, there are some factors to consider:

#### **Are you upgrading a production database?**

If you are running a production environment and intend to upgrade your database, you should first consider creating a testing environment. Although the MaxDB development team keeps stability and backward compatibility always at the forefront of their mind, not all situations can be accounted for. For safety's sake, we recommend replicating your production environment, upgrading the replication, and then shifting the load from the production machine to the backup system. To keep this document on course, however, we will not cover this process in this document.

The [MaxDB FAQ](#) has some [hints on upgrading](#), and the [MaxDB manual](#) has the details under the "Upgrade" section of the Glossary.

#### **Which version do you currently have installed?**

If you are running a production environment, you should consider upgrading to the most recent version of the branch you are currently running. For instance, if you are running a v7.5 database, we recommend upgrading to latest version on the [7.5 download page](#), and if you are running a 7.6 database, we recommend upgrading to the latest version on the [7.6 download page](#).

#### **Are there features that you need?**

If there are hot new features in the latest revision of the database that you simply cannot live without, you will want to get the latest revision. These can always be found at the official MaxDB release page, <http://www.mysql.com/products/maxdb/>

## Are you more comfortable with a GUI or CLI interface?

The installer package comes with both a GUI and CLI interface. The GUI is more user-friendly and provides an Interactive way to get familiar with the process of installing the software. The CLI version can be run as a Background Installation (without user interaction) and automated in batch or shell scripts. For those who use RPM-based GNU/Linux OSs such as Redhat, SuSE and Mandrake, there are also RPMs of 7.5 and 7.6 available at the official MaxDB page.

See also [MaxDB FAQ: Which choices for installations do I have?](#)

## Unofficial installation options

Since MaxDB is Free / Open Source Software (F/OSS), there are un-supported installation options to select from as well. Two of the most common are the Debian distribution of MaxDB 7.5 and the option of building from source.

- Debian GNU/Linux MaxDB packages (version 7.5 only)

```
$ apt-cache search maxdb | grep ^maxdb
maxdb-buildtools - Tools needed to build the MaxDB database system
maxdb-dbanalyzer - A performance monitoring tool for MaxDB databases
maxdb-dbmcli - The MaxDB database manager command line interface
maxdb-doc - Documentation of the MaxDB database system and all of its tools
maxdb-loadercli - A load/unload tool for MaxDB databases
maxdb-lserver - Loader server client of the MaxDB database package
maxdb-server - Communication server for the MaxDB database system
maxdb-server-7.5.00 - A transactional, industrial-strength SQL database
system
maxdb-server-dbg-7.5.00 - Debug versions of the database server kernel
maxdb-sqlcli - A command line query interface to MaxDB databases
maxdb-webtools - MaxDB webserver with WebDBM and WebSQL applications
```

- Building from source

Source (tarballs and zip files) can be found on the official MaxDB page. Note that servers built from source are not supported.

## MaxDB Installation Manager

The following illustrated introduction to MaxDB's installer is brought to you as a joint effort between Ulf Wendel (Windows XP) and myself (Ubuntu GNU/Linux).

Recommended system specifications for MaxDB are:

- 700M free disk space
- 256M memory
- Pentium II class processor operating at 300MHz

After downloading the [MaxDB package](#), unpack it and run SDBSETUP.

We will now walk through a typical installation procedure.

## Pre-install

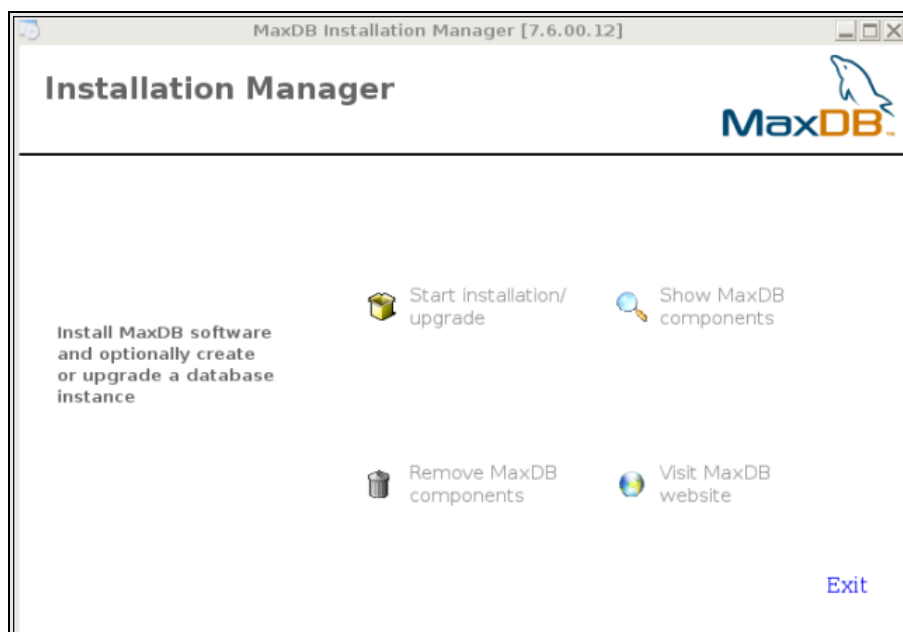
Start the timer!





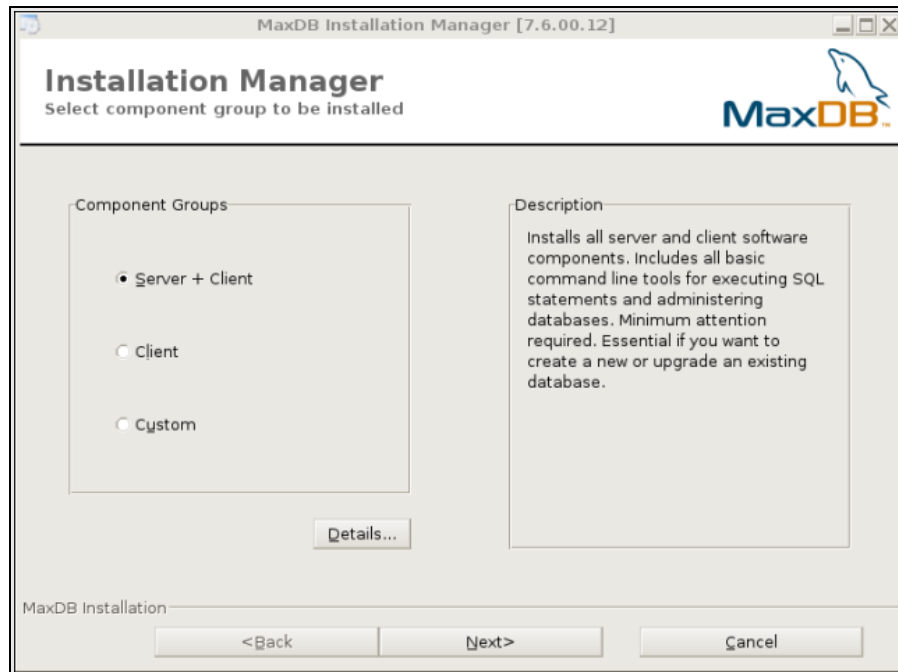
## Start Installation

Here, we select "Start installation/upgrade" in order to get the installation underway. You may select Show MaxDB components to display the packages available for installation and already installed. You may also delete installed components and visit the MaxDB website.



## Select Installation Profile

Here, we select Server + Client in order to install the MaxDB server along with the tools for daily usage, including the Synchronization Manager.



You can see here the client software that will be installed along with the server. Some key components are:

### SQLDBC

This is the C/C++ library for programmatic access to MaxDB.

### JDBC

Much like SQLDBC, but for the Java programming language.

### Loader

The Loader enables the Export from a MaxDB database instance to the file system or the Import of application data from the file system to a MaxDB database instance.

Because the Loader supports a great number of file formats it can often be used to import and export data instead of special application programs. Due to its specialization on MaxDB database instances the Loader offers maximum performance.

Furthermore, the Loader supports the direct transport of data from one database instance to another, which runs on a different hardware structure or a different operating system.

Optionally you can transport the entire data of a database instance, of a certain database user, the individual tables of a special database schema or only a number of qualified columns. Besides files, pipes and back-up tools by other providers are also supported as transport media.

The Loader is a command line tool and can be used on all operating systems supported by the database system.

## ODBC

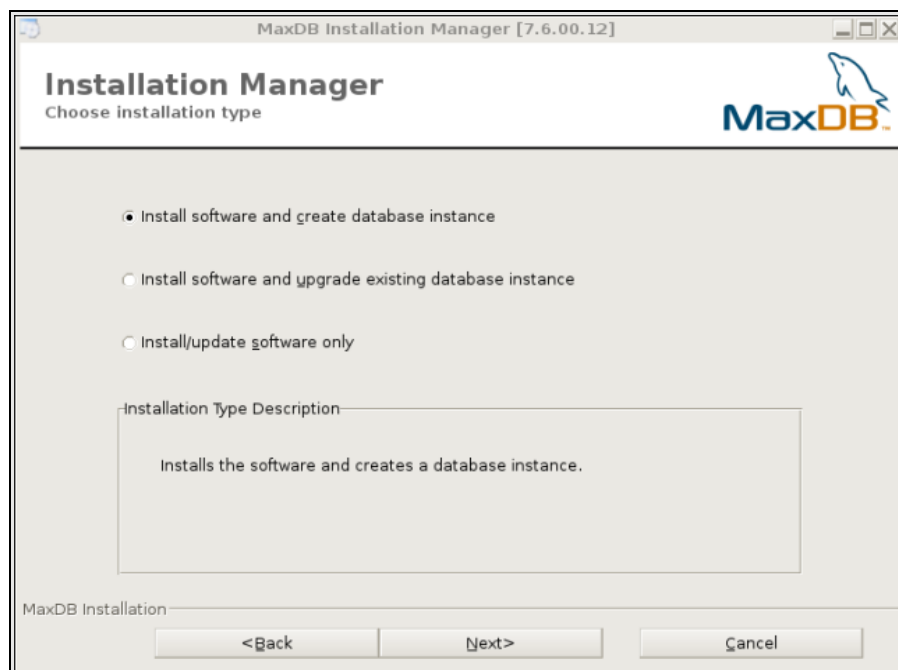
This is the Open Database Connectivity library, an industry standard API for database connectivity.

## Synchronization Manager

This tool synchronizes portions of databases using the Java Messaging Service. The database access layer is system-agnostic and based on JDBC. Example replication databases include MaxDB, MinDB, and to a lesser extent, MySQL.

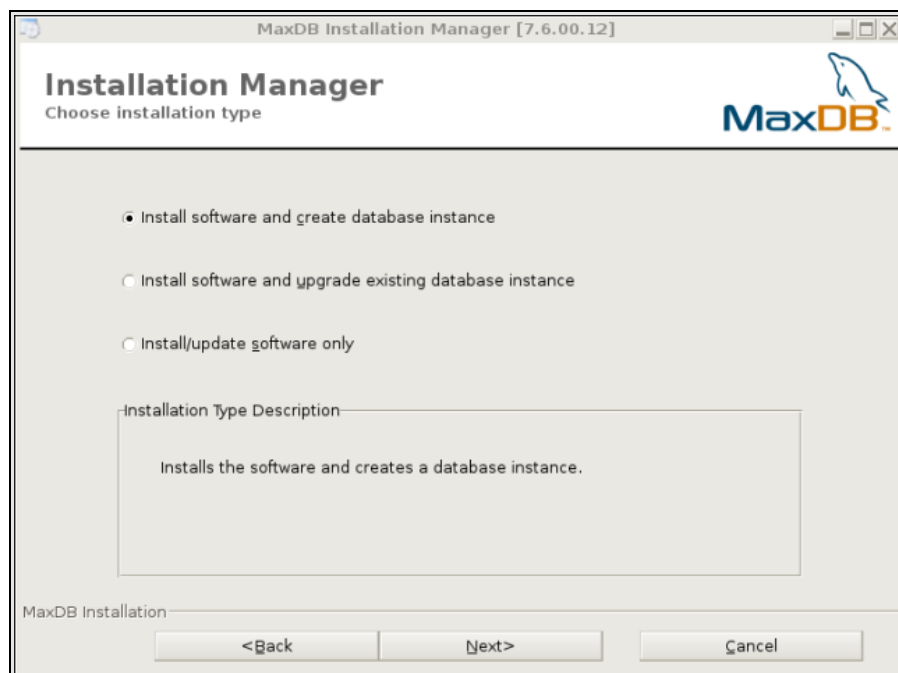
## Choose Installation Type

Select the top choice, "Install software and create database instance." The installer will handle all of the details for you and select reasonable defaults.



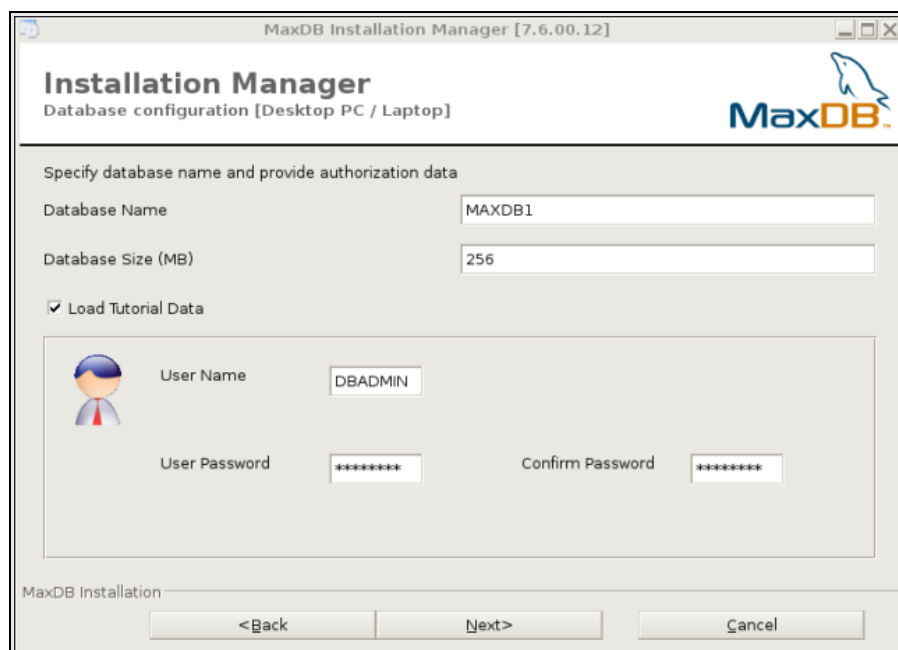
## Select Database Configuration

For this installation, chose PC / Laptop. This installs the most common pieces of software along with the database.



## Database Name / Auth configuration

Leave the database name, size and username values as they are. In future articles, it will be assumed that the password has been left as the default.

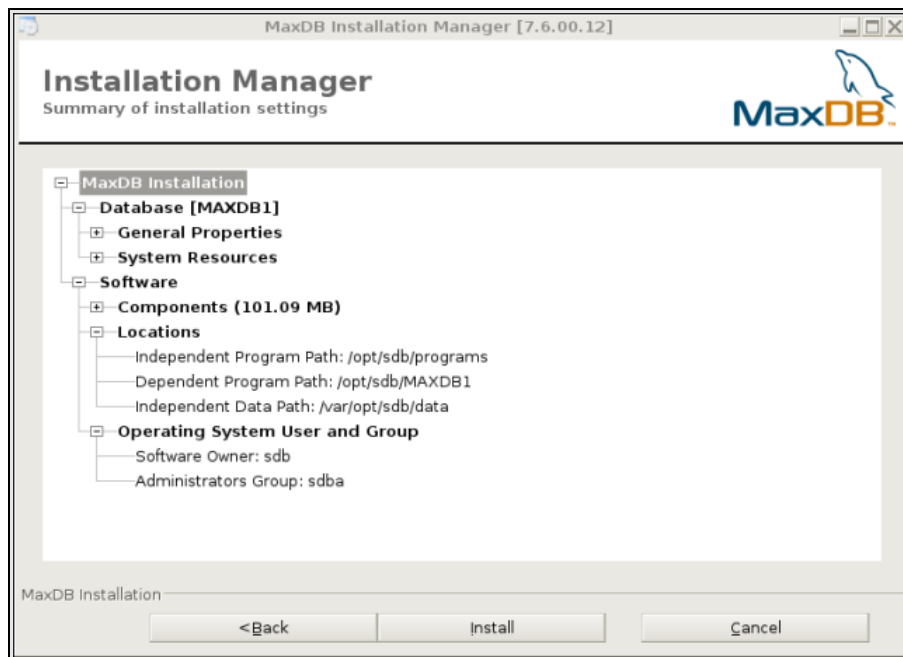


## Specify installation Location

The database software installation path and installation prefix should be left as their defaults (/opt/sdb/programs and /var/opt/sdb/data, respectively in this example); It will be assumed that default values are used throughout the articles.

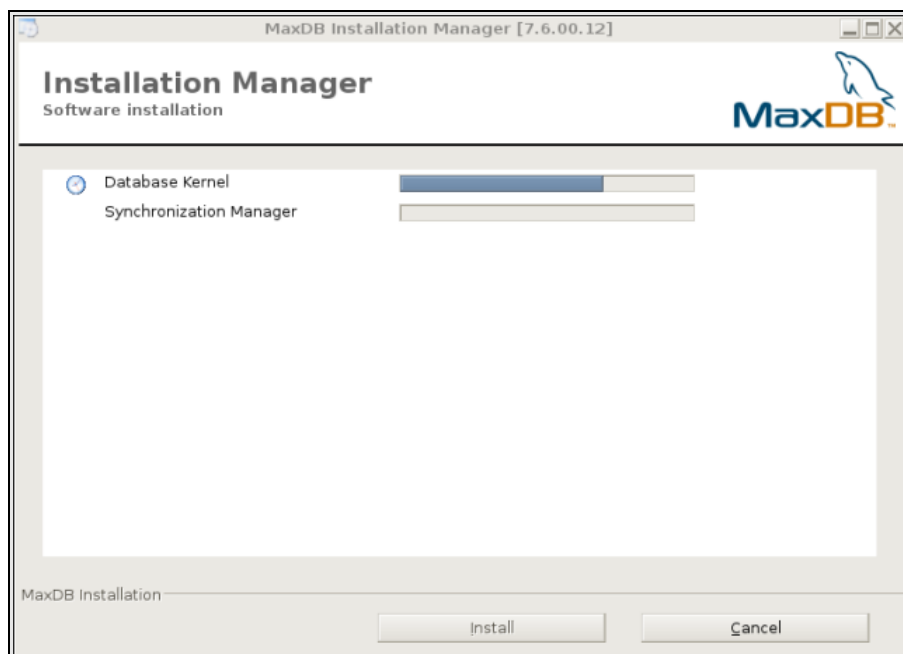
[View Summary](#)

You are presented with an outline of the installation overview. If the settings seem correct to you, continue with the installation by pressing the "Install" button.



## Installation progress

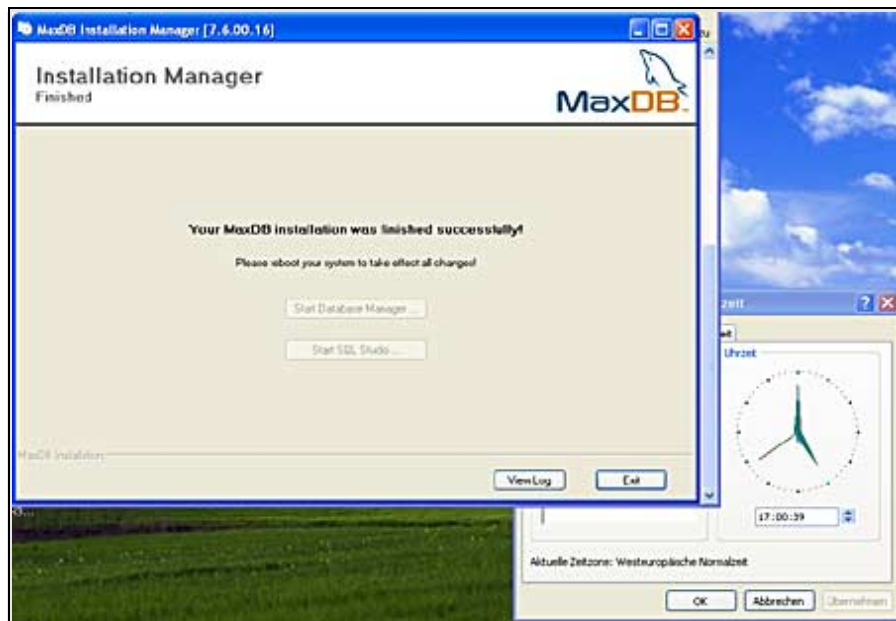
The installer proceeds to copy the files to the destinations previously selected. Patience.



## Complete!

Installation complete. It took us 4 minutes and 36 seconds to install an enterprise database and create a database instance. Less than 5 minutes for the installation, 15 minutes for installation of everything

plus backup. If you do it for the second time, you'll be able to: install the server software, create a database instance, install all GUI tools, setup automatic backup and do the first backup and issue the first SQL command within the 15 minutes limit that MySQL AB has defined for all it's products. Ease of use is our company philosophy.



## xserver

In order to connect to a remote MaxDB installation, you must start the xserver process. On Linux and other Unix-like systems, this can be done by issuing the `/opt/sdb/programs/bin/x_server` command. In a Windows environment, the program is `C:\Program Files\sdb\programs\bin\x_server.exe`

## Connecting to MaxDB using the Database Manager (DBM) GUI

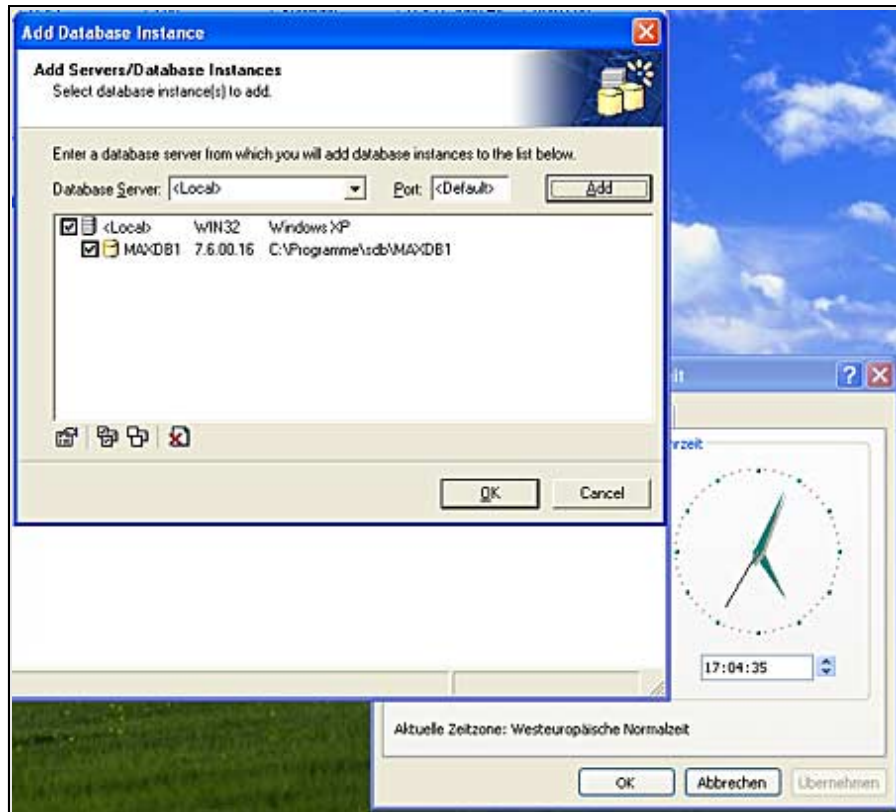
### Install GUI tools

You can download the MaxDB GUI tools, including DBM GUI and SQL Studio from the MaxDB home page: <http://mysql.com/products/maxdb>

The Command Line Interface (CLI) tools come bundled with the MaxDB server.

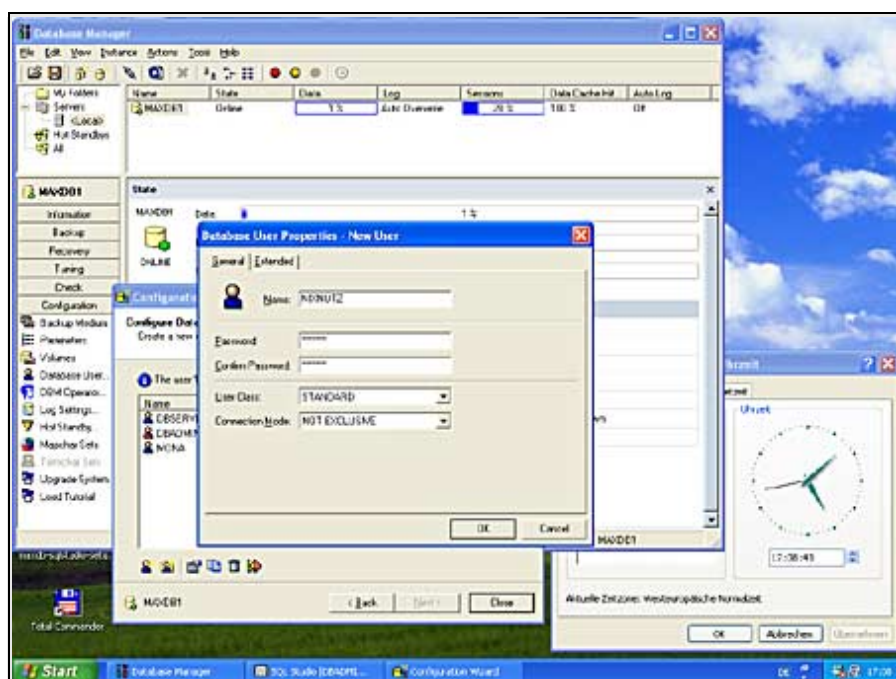
### Connect to MAXDB1 database instance

After installing the tools, you can connect to the MAXDB1 instance using Database Manager



## Easy administration

You can easily issue your daily DBA commands to your MaxDB databases using the user-friendly Database Manager. To do so, ensure that MaxDB is running. On Windows, you can start your MaxDB database by running the Database Manager (Start->All Programs->MySQL MaxDB->Database Manager). When the Database Manager starts, click the Add button in the Database Instance window. Select <Local> and <Default> from the Database Server and Port pulldown menus and click Add. Select the HOTELDB database and press the OK button.



Select HOTELDB from the top right window and select Instance->Connect from the pulldown menu. You will see that your database is in the "Offline" state. Click the green button in the top menu or select Actions->Online from the pulldown menu. Your database should now be in the online state.

## Issuing queries with SQL Studio

SQL Studio provides a simple interface to issuing queries to your MaxDB databases. You can point and click and view your results in an easy-to-understand spreadsheet layout.

Start SQL Studio (Start->All Programs->MySQL MaxDB->SQL Studio) and log in to HOTELDB on localhost. When SQL Studio starts, click Session->Connect from the pulldown menu. Enter 'localhost' in the Server text field and 'HOTELDB' in the Database field. Enter your database user and password in the appropriate fields.

Start SQL Studio (Start->All Programs->MySQL MaxDB->SQL Studio) and log in to HOTELDB on localhost. When SQL Studio starts, click Session->Connect from the pulldown menu. Enter 'localhost' in the Server text field and 'HOTELDB' in the Database field. Enter your database user and password in the appropriate fields.

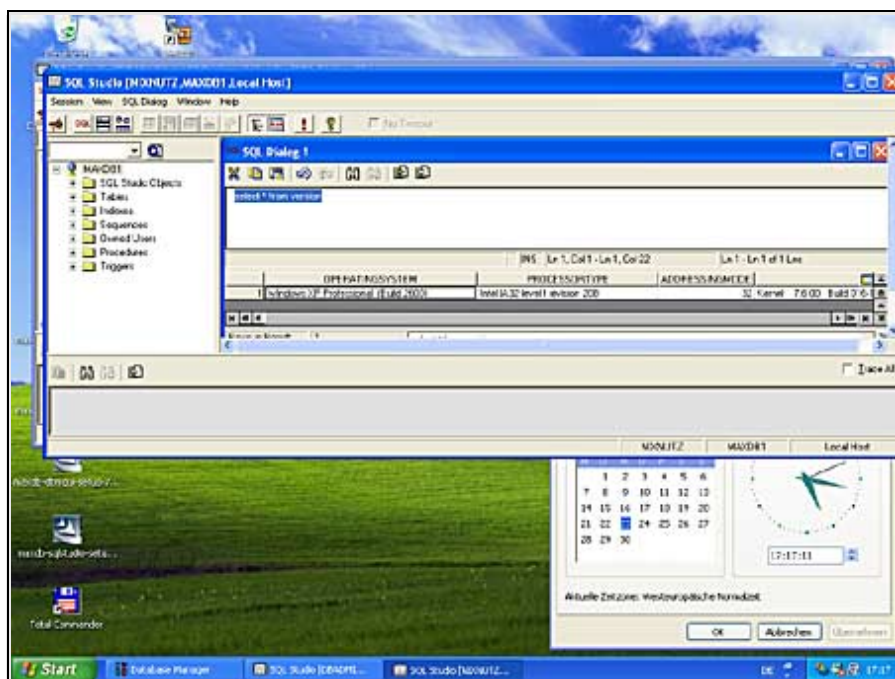
In the SQL Dialog 1 window, enter the following:

```
SELECT 'hello world' FROM DUAL
```

Now press F8 to issue the command to your database. You should be presented with a 1x1 result that says "hello world". Congratulations. You have just successfully issued your first SQL query to MaxDB.

You can also try this query:

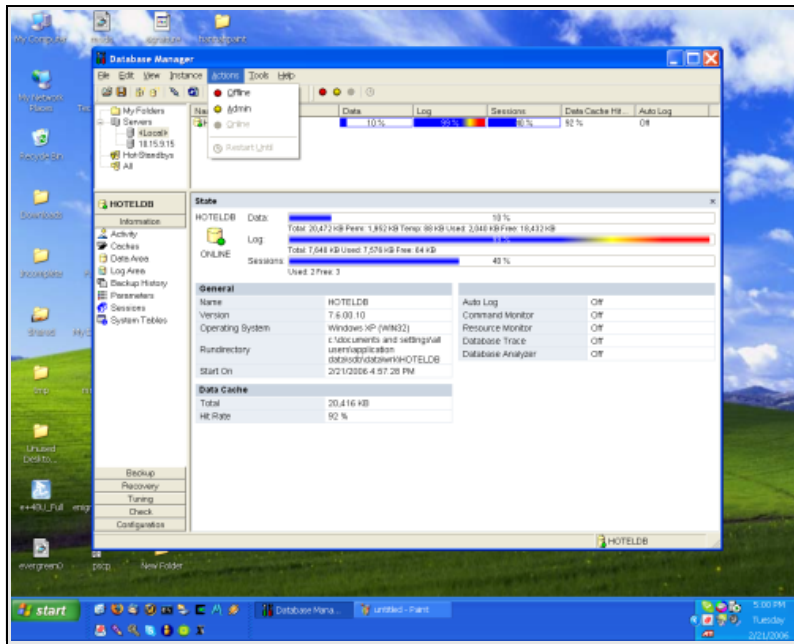
```
SELECT * FROM domain.versions
```



## Shutting down with Database Manager

Shutting down an instance with the Database Manager is as simple as clicking Actions->Offline from the menu.





## In the next issue

In the next issue, we'll describe in more detail the use of the Database Manager and SQL Studio. We will also present dbmcli and sqlcli, the command-line interfaces to issuing database commands and SQL queries to MaxDB.

We hope that you will join us in one week for this exciting and educational article. Until next time, we wish you happy hacking.

*C.J. Collier on behalf of the MaxDB team*

## MaxDB series: DBM GUI and dbmcli, SQL Studio and sqlcli

Published: 03.03.2006

**Dear MySQL users, MaxDB users and friends,**

this is the fourth posting in our MaxDB series. The series started about four weeks ago and it will very likely take the rest of the year to complete. All postings of the series together will make kind of an online course for MaxDB. Over the last postings we explained what [MaxDB](#) is, we explained about the outstanding features of MaxDB and we taught you how to perform a basic MaxDB installation that serves as a basis for all of the following lessons.

### **In this issue**

The today's MaxDB lesson is about the two most important tools of MaxDB. The first tool is the Database Manager GUI and it's commandline counterpart dbmcli. The Database Manager GUI is the main tool to perform administrative tasks like shutting down the database, performing backups or doing some basic monitoring of the health status of your MaxDB database instances.

- [The Database Manager](#)
- [System Architecture](#)
- [DBMCLI Usage Example](#)
- [The Database Manager GUI](#)
- [Remote connections: ensure the X Server runs and no firewall blocks communication](#)
- [The SQL Tools](#)
- [Learning DBM command using the Reverse Engineering trick](#)
- [Using SQL Studio](#)
- [Pitfalls, error messages and searchable manual variants](#)
- [Summary](#)
- [In the next issue](#)

The lesson will not discuss the Database Analyzer. The Database Analyzer is used for performance bottleneck-analysis. We described the tool in the DevZone article [MaxDB performance tuning primer](#) and we will get back to in a future posting. We also do not describe the Loader (Im-/Export tool) or any of the database utilities ( Event Dispatcher, XCONS, XUSER, X Server) today. But again, you can be sure we will discuss them later. For today the lesson is on the two most important tools for you to start with MaxDB.

Most of the components have their own logfile that you can check for errors. The logfile of X Server is named xserver\_<computer\_name>.prt.

### **The Database Manager**

MySQL users that try out MaxDB for the first time will be faced with a new term: database instance. You can compare a database instance roughly to a MySQL Server software installation with several databases. A MaxDB database instance is a copy of the version dependent software components of MaxDB - mainly the kernel and some interface libraries - together with the data and log files. In other, simplified words: a MaxDB database instance is a copy of the database kernel plus the database files managed by the software. The database instance can be divided on the SQL level into SQL schemata. One schemata inside a MaxDB database instance can be compared to one logical MySQL database. Remember that CREATE SCHEMA is a synonym for CREATE DATABASE as of MySQL 5.0.2.

Administrative actions are performed under MySQL partly using SQL-like commands, partly using tools, partly by modifying the server configuration file. MaxDB tries to use one unified tool for all administrative tasks. Although you'll find also SQL-like commands and tools in addition to the

Database Manager in MaxDB the Database Manager remains the main and central component to perform administrative tasks.

## System Architecture

Before we explain how to use the Database Manager, you need to learn some basics about the system architecture. The background knowledge of how the individual parts of the database work together helps you to get the big picture and to develop a deeper understanding of the technology. This in turn helps you to analyze and debug problems on your own.

Database Manager (DBM)	
Clients	Server
Database Manager GUI (DBMGUI): graphical user interface	Database Manager Server (DBM Server)
Database Manager CLI (DBMCLI): command line interface	
Database Manager RFC (DBMRFC): interface to the SAP system; only in SAP systems	

The Database Manager (DBM) is a server - client application. Client and server part together are called the Database Manager. The Database Manager GUI (DBM GUI) and the Database Manager CLI (DBMCLI) are two clients of the Database Manager Server (DBM Server). Having both a GUI and a CLI client is an advantage. You can use call the CLI tool DBMCLI from within your own administration scripts. Scripts can help you to automate standard tasks and ensure that things are always done in the same, documented way. The GUI interface is handy for beginners but also experts will profit from the built-in, easy to use wizards of the DBM GUI.

The Database Manager server receives commands from the clients and talks to the kernel in order to execute the commands. If the DBM client and the DBM server are on the same host, then the DBM client can connect directly to the DBM server. If the DBM client runs on a different host than the DBM server, then the X Server remote communication component (default port: 7210) will be connected from the DBM client to forward the request to the DBM server.

Local communication			
DBM Client ->		DBM Server ->	Kernel
Remote communication			
DBM Client ->	X Server ->	DBM Server ->	Kernel

What does this mean for you, why should you know? For example, you might want to know which components connect to each other, if your DBM GUI does not work as expected and reports communication error. If your DBM GUI connects to a remote DBM Server, if it connects to a remote database instance, then check all the components on the way. Ensure that the X Server is running (see below), check if you can establish a local DBM Server connection using a locally installed DBMCLI and check if the kernel is running.

Most of the components have their own logfile. You can check the logfiles for error messages to debug the chain of communication. The X Server protocol is named `xserver_<computer_name>` and is located in the directory `<independent_data_path>/wrk/`. The Independent Data directory (`<independent_data_path>`) contains data, configuration and run directories of database instances and applications that are version independent. The counterpart to the Independent Data directory is the Dependent Path (`<dependent_path>`). As you might expect, you will find version-dependent server software in the Dependent Path. All programs in these version-dependent directories are not executed directly by the user, but are called using the programs stored in the `<independent_program_path>` (See also [MaxDB Glossary: Database Directory](#)).

## DBMCLI Usage Example

Stay tuned, the lengthy explanations on the hardware footprint of MaxDB do not only explain how to debug your setup, but also lead us directly to the first usage example of the DBMCLI, the text-based command line DBM client. The DBMCLI is located in the installation path (<inst\_path>) of the database. If you did not change the default settings for locations during the MaxDB installation as we advised in our last article, the default location is /opt/sdb on Unix and C:\Program Files\sdb . Call the DBMCLI and check the help output.

```
$ /opt/sdb/programs/bin/dbmcli -help
usage: dbmcli [<options>] [[-c] <dbmserver -Command>]
  <options>:
    -u username,passwd (DBM user)
    -U [userkey] (default DBMUSR)
    -uUTL [username,passwd] (UTILITY session)
    -uSQL [username,passwd] (SQL session)
    -uSRV (Service session)
    -V (show version of server)
    -d dbname (set dbname)
    -R dbroot (set dbroot)
    -n node (name of servernode)
    -e encryption (encryption method)
        (use only with -n)
        (valid methods: 'SSL')
    -i inputfile (Default stdin)
    -ic inputfile (Default stdin)
    -o outputfile (Default stdout)
    -t protocolfile (writes subsequent protocol)
    -s (local mode, don't use with -n)
    -key (wait for any key on exit)
  <DBMServer-Command>:
    Everything after the options will be sent to DBMServer.
    For more information about the DBMServer-Commands use
    the DBMServer-Command help.
```

Connect to the MAXDB1 database instance that was created during the last lesson. Again, if you did not overrule any of the default settings as we suggested, you can use the user DBADMIN with the password DBADMIN to establish the connection of the DBMCLI client to the DBM server. Specify the username and password using the option -u. Use the option -d to specify the name of the database instance you want to connect to. If the database instance does not run on the same host as the DBMCLI tool, use -n to specify the hostname of the database instance.

```
$ /opt/sdb/programs/bin/dbmcli -u DBADMIN,DBADMIN -d MAXDB1
dbmcli on MAXDB1> help
nixnutz@linux:~> /opt/sdb/programs/bin/dbmcli -u DBADMIN,DBADMIN -d TEST
/opt/sdb/programs/bin/dbmcli on TEST>help
OK
apropos <command_name_part>
[...]
user_response <response>
version

---
dbmcli on MAXDB1> file_getlist
OK

keyname,mode,size,date,time,comment,filename
KNLDIAG ASCII 58646 20060228 181543 Database Messages
knldiag.classic
[...]
ANALYZER DIRECTORY 48 20051003 184601 DB Analyzer File
analyzer
```

Use the command “help” to display a list of all DBM commands and try “file\_getlist” to get a list of all database files including the configuration and protocol files that belong to the database instance MAXDB1 to which you are connected. As you can see the X Server protocol file is not listed, because the X Server is a version independent tool that does not belong to one database instance. However, we explained that the X Server protocol file is located in <independent\_data\_path>/wrk/. Use the DBM

command “dbm\_getpath” find out the setting for <independent\_data\_path> and check if you can find the X Server protocol file in that location.

```
dbmcli on TEST>help dbm_getpath
OK
dbm_getpath          [IndepDataPath|IndepProgPath]
---
dbmcli on TEST>help dbm_getpath IndepDataPath
OK
/var/opt/sdb/data
---
dbmcli on TEST> quit

$ ls -la /var/opt/sdb/data/wrk/xserver*
-rw-rw-rw- 1 sdb sdba 506 2006-02-28 17:27 /var/opt/sdb/data/wrk/xserver_linux.nitrace
-rw-r----- 1 sdb sdba 65536 2006-02-28 17:27 /var/opt/sdb/data/wrk/xserver_linux.prt
```

A list of all DBM commands is contained in the manual. As usual the easiest way to find the list is to use the [Glossary: DBM Command](#). Later in the course, we will give examples how to perform typical administrative tasks like backup and recovery using the DBMCLI. For now, please consult the manual.

Back to the debug example. Of course the DBM server does also have a Database Manager Protocol (DBMPRT). The DBMPRT is located in <independent\_data\_path>/wrk/<database\_name>/dbm.prt . The Database Manager Protocol is not only a great source of information when you debug connection problems. The support will ask you frequently for it to learn which administrative tasks have been executed. And - listen - you can use the DBMPRT to reverse engineer actions that have been performed in the DBM GUI. This is very handy as often you do not need only to know which DBM command to execute but also in which context and order to execute it.

```
$ ls -la /var/opt/sdb/data/wrk/MAXDB1/dbm.prt
-rw-rw---- 1 sdb sdba 13634 2006-02-28 18:26 /var/opt/sdb/data/wrk/MAXDB1/dbm.prt

$ tail /var/opt/sdb/data/wrk/MAXDB1/dbm.prt
[...]
2006-02-28 18:26:42 0x00001fa3 INF          226 DBMSrv   DBM Server client disconnect: PID 8098
on computer linux.site
```

## The Database Manager GUI

Let’s try out this reverse engineering hint. Download and install the DBM GUI from <http://dev.mysql.com/downloads/maxdb/7.6.00.html> if you have not done this already. Note that the DBM GUI is not part of the server download. It comes as an extra download. The same is true for the SQL Client “SQL Studio”.

Both applications are currently only available on Windows. Future versions of MaxDB will be delivered with new client tools that are based on Java and thus can be run on all operating systems that have a GUI and are able to run Java, which is basically true for all operating systems that MaxDB supports. Up to and including version 7.5.00 there are web based counterparts of the Windows GUI tools (DBM GUI and SQL Studio). The development of the Web DBM, the Web Server and the Web Server Manager has been discontinued. Open Source customers can try to install the 7.5.00 versions of the tool with 7.6.00 server. Commercial customers can consult the support team for more information.

The Database Manager GUI (DBM GUI) can be used to administrate multiple database instances. Start the DBM GUI and register your database instance “MAXDB1”. Make sure that the DBM GUI shows all of the four main screen areas. The four main screen areas are (clockwise from upper left to lower right): the directory structure of the database instances, the list of the database instances, the menu list of the current database instance and the output area. If your screen does not show these areas, use the “View” menu to reconfigure the software. Select “Database Instance... Add” in the output area to open a wizard to register your database instance “MAXDB1”. Alternatively you can use a right mouse click on “Servers” in the directory structure area (upper left) to invoke the wizard using the context menu.

Fill out the combo box “Database Server” of the wizard to specify the host on which your database instance “MAXDB1” runs. If the DBM GUI and the database instance are running on the same host, you do not need to alter the default value of <Local> for localhost/127.0.0.1 . If you want to connect to a remote host, specify the server name or the IP of the remote host, e.g. 192.168.0.2 . Click on “Add” to display a list of all MaxDB database instances running on the specified host, select the database you want to register at the DBM GUI and click OK.

## Remote connections: ensure the X Server runs and no firewall blocks communication

Remember that for all remote communication the X Server is used! The X Server must be running. On Windows the X Server has been installed as a service and will be started automatically. On Unix you must start the X Server manually (see below). By default the X Server is using port 7210 (TCP/IP). Make sure your firewall does not block this port. One way to configure an alternative port is to use the option -S of the x\_server program.

```
$ /opt/sdb/programs/bin/x_server -W -Y start
12902 XSERVER started, 'X32/LINUX 7.6.00 Build 012-121-102-632'

$ ps -ef | grep vserver
sdb      7747      1    0 12:41 pts/3      00:00:00 /opt/sdb/programs/pgm/vserver -W -Y start
root    7750     7739    0 12:41 pts/3      00:00:00 grep vserver
```

The “MAXDB1” database instance should now be visible in the Server list of the upper left screen area. Click on the host and select the database instance in the list of database instances. Connect to the database using the username DBADMIN and the password DBADMIN. As said above, we assume that you have not changed any of the default settings during the installation of the MaxDB software in the last lesson. Be warned that the DBM GUI might remember the username and password you have specified to connect the DBM Client (the DBM GUI) to the DBM Server if you have chosen this option when connecting to an instance. You can change this behaviour using the application menu: View - Options - Authorization.

You just managed to do the most complicated task in the DBM GUI. The rest should be self-explanatory. Most of the actions that you can perform are “secured” by using wizards. The wizards will prevent you from configuring settings that do not make sense or they will at least bail at you in most cases. In other words: it is “safe to play” with the DBM GUI when you try it out for the first time.

Your first exercise is to start the “MAXDB1” database instance. Select the MAXDB1 database instance in the instance list and start it using Actions - Online. A MaxDB database instance can be set in one of four operational states: Offline, Admin, Online and Standby. Unless you are working with a MaxDB Hot Standby setup, you will not work with the Standby state. SQL clients can only connect to the database when it is in Online state. Some, few administrative tasks can only be performed in Admin state. Admin state means that all SQL clients are disconnected but the kernel is still running. Once you have switched to Online state, switch to Admin, Offline and Online again. Don’t ask why, just do it.

- Online: SQL users can connect, administrative commands can be executed
- Admin: only administrative tasks can be executed
- Offline: not running

## Learning DBM command using the Reverse Engineering trick

At the beginning it has been shown that the DBM GUI is a client of the DBM Server and that the DBM Server writes a log file, a protocol file. The logfile is located in <independent\_data\_path>/wrk/<database\_name>/dbm.prt . It has been made the statement that this protocol file is a great help to learn about DBM commands as it can be used to reverse engineer DBM GUI commands. Let’s check.

```
$ tail -n 20 /var/opt/sdb/data/wrk/MAXDB1/dbm.prt
```

```

2006-03-02 14:38:50 0x00001f9a INF      216 DBMSrv  DBM Server client connection
established: PID 316 on computer 192.168.0.2
2006-03-02 14:38:50 0x00001f9a INF          0 DBM      command auto_extend show
2006-03-02 14:38:50 0x00001f9a INF          0 DBM      command auto_update_statistics show
2006-03-02 14:41:31 0x00001f9a INF          0 DBM      command db_online -f
2006-03-02 14:41:40 0x00001f9a INF          0 DBM      command db_execute SELECT
ID,CREATEDATE,USEDSPACE
2006-03-02 14:41:40 0x00001f9a INF          0 DBM      command auto_extend show
2006-03-02 14:41:40 0x00001f9a INF          0 DBM      command auto_update_statistics show
2006-03-02 14:41:45 0x00001f9a INF          0 DBM      command db_admin -f
2006-03-02 14:41:53 0x00001f9a INF          0 DBM      command auto_extend show
2006-03-02 14:41:53 0x00001f9a INF          0 DBM      command auto_update_statistics show
2006-03-02 14:41:55 0x00001f9a INF          0 DBM      command db_offline
2006-03-02 14:41:56 0x00001f9a INF          0 DBM      command auto_extend show
2006-03-02 14:41:56 0x00001f9a INF          0 DBM      command auto_update_statistics show
2006-03-02 14:42:03 0x00001f9a INF          0 DBM      command db_online -f
2006-03-02 14:42:10 0x00001f9a INF          0 DBM      command db_execute SELECT
ID,CREATEDATE,USEDSPACE
2006-03-02 14:42:10 0x00001f9a INF          0 DBM      command auto_extend show
2006-03-02 14:42:10 0x00001f9a INF          0 DBM      command auto_update_statistics show

```

The protocol file shows the commands that have been sent from the DBM GUI to the DBM server. It does not require expert knowledge see that commands `db_online`, `db_admin` and `db_offline` are used to alter the operational state of the database instance to Online, Admin and Offline. The commands `auto_extend` and `auto_update_statistics` are frequently called with the parameter `show`. Check the output area of the DBM GUI for the two columns Auto Extend and Auto Update Statistics. Again, it's not difficult to do the reverse engineering and find out where the DBM GUI gets the information to show the On/Off status information. Check what you have learned using DBMCLI. Make sure you end up with the database instance being in Online operational state, otherwise you cannot connect with the SQL tools in the next step.

```

$ /opt/sdb/programs/bin/dbmcli -u DBADMIN,DBADMIN -d MAXDB1
MAXDB1>auto_extend show
OK
OFF

---
MAXDB1>db_offline
OK

---
MAXDB1>help db_online
OK
db_online          [-f|-s|-t] [-u <yyyymmdd> <hhmmss>]

---
MAXDB1>apropos db_online
OK
db_online          [-f|-s|-t] [-u <yyyymmdd> <hhmmss>]

---
MAXDB1>db_online
OK

---

```

## The SQL Tools

MaxDB comes with three different SQL tools, much like with the DBM tools. SQL Studio is an ODBC based GUI application available on Windows to issue SQL queries. SQLCLI is the text-based command line variant and Web SQL is the third variant. As explained above, you should try not to use Web SQL or any of the other Web tools. Some of them are already deprecated and it could happen that Web SQL will also be deprecated in future versions.

The SQLCLI binary can be found in the same directory as the DBMCLI program. SQLCLI can be invoked with the same options as DBMCLI. The major difference in use is the syntax for commands. SQLCLI is using backslash commands. In order to quit SQLCLI you do not write quit but `\quit`. We will

not discuss any more details of SQLCLI. It is more convenient and comfortable to use SQL Studio or any other ODBC/JDBC based third-party GUI tool as your favourite SQL tool

```
$ /opt/sdb/programs/bin/sqlcli -h
SQLCLI version 7.6.0, the MaxDB interactive terminal.

Usage:
  sqlcli [<options>] ... [<command>]

Options for connecting to the database:
-n <database_server>[:<port>] name of the host on which the database
                           instance is installed(default: localhost)
-d <database_name>         name of the database instance
-u <user_name,password>   user name and password
-U <user_key>             xuser key

-S SQLMODE                SQL mode, possible values are
                           \"INTERNAL\", \"ORACLE\", \"ANSI\", or \"DB2\"
-y                        sets session timeout in seconds (0 means no timeout)
-z                        switches autocommit off
-r                        suppress usage of prepared statements

Input and output options:
-i <file_name>           use file <file_name> to input queries (default: stdin)
-o <file_name>           use file <file_name> for output (default: stdout)
-x                        run quietly (no messages, only query output)
-m                        activates the multi-line mode for query input
-c <separator>          use <separator> for separate commands read from
                           input files (default: '/')

Output format options:
-F <separator>          use <separator> as the field separator (default: '|')
-g <null_value>         use <null_value> whenever a field is NULL (default: '?')
-p <prefix>             use <prefix> as row prefix for printing (default: '|')
-P <suffix>            use <suffix> as row suffix for printing (default: '|')
-A                      switch to unaligned output mode.
-Q                      show each column on a separate line
-a                      do not print any header for SELECT commands
-f                      printout the SQL commands
-b MAXLENGTH           max. output length for long columns in
                           bytes/characters (default is 10)

General options:
-h                      show help, then exit
-v                      printout version information
-E <exit code>         exit sqlcli returning <exit code> in case of an error
-t                      printout debug information
-T <filename>         write a database interface trace (SQL trace)

For further information, type \"\\?\" (for internal commands) from
within SQLCLI or consult the SQLCLI documentation.

$ /opt/sdb/programs/bin/sqlcli -u DBADMIN,DBADMIN -d TEST

Welcome to the MaxDB interactive terminal.

Type:  \h for help with commands
       \q to quit

sqlcli=> \h

\? or \h[elp]  show help on internal slash commands
\q[uit]        quit SQLCLI
\c[onnect]    -n <database_server>[:<port>]
              -d <database_name>
              -u <user_name,password>
              connecting to the database
\di[sconnect] disconnecting from the database
\a[utocommit] [ON|OFF] switch autocommit mode on or off
\m[ode]       [INTERNAL|ORACLE|ANSI|DB2] change SQL mode
\ps           [ON|OFF] toggle the usage of prepared statements on or off
\o[utput]    <filename> send all query results to file
```



```

\i[input]      <filename> read input from the file <filename>

\p[rint]      print out the current query buffer
\r[eset]      reset (clear) the query buffer
\e[dit]       <filename> edit the query buffer (or file) with external editor
\g           send query buffer to server (and print out the results)

\al[ign]      [ON|OFF] toggle the aligned output on or off
\f[ieldsep]   <separator> use <separator> as the field separator
\s[tatus]     print out host, database, user etc.
\!<command>  execute <command> in shell

\dc [PATTERN] list columns
\de [PATTERN] list indices
\dp [PATTERN] list procedures
\ds [NAME]    list schemas
\dt [PATTERN] list tables
\du [NAME]    list users
\dv [PATTERN] list views

[PATTERN] = [OWNER.] [OBJECT NAME] eg. <username>.%A%

```

## Using SQL Studio

If you have not installed SQL Studio yet, go to the MaxDB download page, download the software and install it now. Remember that SQL Studio comes as an extra download just like DBM GUI. Both tools are not included in the server package.

Using SQL Studio is quite easy and does not require words to explain. All that must be said are some words about common pitfalls and useful functions that are not obvious to every beginner. The first thing to check when you are connected to a database instance are the connection properties. Below the result area and above the protocol area you can see four select boxes. The first box is used to turn auto commit on and off. If you turn auto commit on, MaxDB will implicitly issue a commit after every statement. Be careful with auto commit when you delete data or modify SQL objects. There is no way to undo your actions, there is no way to rollback an action.

With the second checkbox you can define the SQL mode that you would like to use. Similar to the MySQL Server, MaxDB supports different SQL modes. The SQL modes define which SQL language variant is accepted.

- INTERNAL: Internal MaxDB definition
- ANSI: ANSI standard in accordance with ANSI X3.135-1992, Entry SQL
- DB2: Definition DB2 Version 4
- ORACLE: Definition ORACLE 7

The SQL Studio checkbox allows you to connect to MaxDB using the SQL modes INTERNAL or ORACLE.

The third checkbox allows you to define the Isolation Level. MaxDB supports all Isolation Levels defined by the ANSI standard. You can choose between: not committed, committed, repeatable and serializable. The meaning of the levels and the phenomena that can occur are explained in the MaxDB manual, see [Glossary: Isolation Level](#). And if you are on MaxDB 7.6.00 you can set the current schema in checkbox on the right (MySQL: CREATE DATABASE = CREATE SCHEMA).

Once you have checked the connection properties you can execute your first query. Do something simple like: SELECT \* FROM VERSIONS. The database system contains a series of system tables with information about database objects and their connection to each other, statistics system tables, monitor system tables and loader system tables. When you compare this to MySQL, MaxDB system tables are the same as the tables contained in the INFORMATION SCHEMA of MySQL. To execute the query, move your cursor in the line of the query and press F8. Alternatively you can press the button with the red exclamation mark on it.

If you want to execute multiple statements at once, write it statement in its own line and put one line with two forward slashes ( // ) in between the statements. Double forward slashes in a new line are used to separate statements from each other and to mark the end of one statement.

## Pitfalls, error messages and searchable manual variants

Now comes a common pitfall when you use SQL Studio for the first time. Execute SELECT "Hello PlanetMySQL!" FROM DUAL. Note the double quotes around the string. SQL Studio complains about this statement:

```
---- Error -----
Auto Commit: On, SQL Mode: Internal, Isolation Level: Committed
Column not found;-4005 POS(9) Unknown column name:Hello PlanetMySQL!
SELECT \"Hello PlanetMySQL!\" FROM DUAL
```

The error message that appears in the protocol area tells you the reason. If you put a string in double MaxDB interprets it as special identifier. MaxDB distinguishes between simple and special identifiers. An identifier is something like a column name, a table name, or a function name. If you do not enclose an identifier in double quotes, MaxDB will ignore upper and lower case by using the upper case variant of the identifier. That means, that for example the table names Dual, DUal and dUal all refer to the table DUAL. But if you put double quotes around an identifier no conversion is made. That means the table name "Dual", "DUal" and "dUal" refer to three different tables and none of them refers to the table DUAL. So, in this case MaxDB believes you want to refer to the column "Hello PlanetMySQL!" which does not exist. Use single quotes to solve the problem: SELECT 'Hello PlanetMySQL!' FROM DUAL.

```
++++ Execute +++++
Auto Commit: On, SQL Mode: Internal, Isolation Level: Committed
SELECT 'Hello PlanetMySQL!' FROM DUAL
Statement successfully executed.
Execution Time: 20:10:06.744 - 20:10:06.763 (00.019 sec)
```

Whenever you get an error, MaxDB will show you the error code, e.g. -4005. Error codes are divided in certain number ranges. All errors from -4000 to -4999 are database kernel message errors. Error codes and number ranges are documented in the manual. The frontpage of the [online manual](#) shows a link "Messages" that you can use to navigate to a list of all error messages, number ranges and explanations how to solve the error. Searching the online manual is difficult, but there is a trick to search the manual. Press F1 in the SQL Studio or select SQL Studio Topics in the Help menu. The SQL Studio help contains the entire MaxDB manual. You can search this variant of the manual as easy as the Windows help/HTML help variant of the manual offered for download on <http://dev.mysql.com/doc/maxdb/index.html>. When you use the SQL Studio help, make sure that the version of the manual does match the version of the database instance you are connected to. If that is not the case, get a copy of the manual version that belongs to your database instance version. The SQL Studio packages is not as often updated as the server packages and the documentation packages. Be warned that the SQL Studio help might be outdated. Nevertheless, it is a good and powerful alternative to the HTML version of the manual.

## Summary

Looking back on all articles of the MaxDB series, you should have a working copy of MaxDB. You know how to start and stop it, how to use the administration tools and how to execute SQL queries using the SQL tools. What we left out is the topic of creating automatic startup and shutdown scripts on Unix. On Windows everything is done using services and has been configured automatically. Check the manual ([Glossary: Automatic Start](#)) on hints and examples for Unix start scripts.

For you it is time to look back. Use the DBM GUI and the SQL Studio to evaluate if MaxDB has all the features you need. If anything is missing, please use the [MaxDB forum](#) to tell us what MaxDB lacks.

## In the next issue

In the next issue we will discuss the user concept of MaxDB. Comments are always welcome. Enjoy!

*Ulf Wendel for the MySQL MaxDB team*

## MaxDB series: User concept, authorization and schemata

Published: 10.03.2006

### Dear MySQL users, MaxDB users and friends,

a new version of [MaxDB 7.5.00](#) has been released recently: [MaxDB 7.5.00.34](#). Two versions of MaxDB have the status of General Availability (GA). MaxDB 7.5.00 and MaxDB 7.6.00 can be used in production environments. Open Source users should use the latest release which is 7.6.00. Last weeks release of 7.5.00 Build 34 (7.5.00.34) is of interest for all 7.5.00 users. Check the [Changelog](#) if it shows “must-have” entries for you.

### In this issue

This article is part of a MaxDB series on PlanetMySQL that teaches you how to use MaxDB. We hope that we can write about 40 articles in 2006: roughly one per week, published on wednesdays if time permits. All articles together will make a complete online class.

Today's lesson is on the user concept of MaxDB and the related topic of SQL schemata. The user concept of MaxDB seems to have caused a lot of confusion in the past. One reason might be that in different versions of MaxDB different system users are available and that in some cases default usernames overlap with the name of user classes. Therefore it is important that you know about the basic concepts and the terms used in the [MaxDB documentation](#).

Closely related to the user concept is the topic of SQL schemata. A schemata is no more but a namespace for SQL objects. It would be hardly worth covering schemata if MaxDB would not implicitly create a schemata for every SQL user. This sounds odd if you do not have a long MaxDB background and if you do not know how it was handled before schemata have been introduced with MaxDB 7.6.00

Operating system level users are not discussed. On Unix, MaxDB will create a new operating system user “sdb” and a new operating system user group “sdba”. The new user will become the owner of the MaxDB software installation on the operating system level. This is done to implement extra security on the operating system level and to prevent other users from accidentally destroying the software installation.

- [Three different user types in MaxDB](#)
- [“root”: The Database System Administrator \(SYSDBA user\)](#)
- [“Administrator”: Database Manager operators \(DBM operator\)](#)
- [Managing DBM Operators](#)
- [Example: creating a start-stop user](#)
- [“SQL users”: Database users of the user classes DBA, RESOURCE and STANDARD](#)
- [Hands on: Managing two SQL users](#)
- [In the next issue](#)

### Three different user types in MaxDB

When talking about the user concept of MaxDB, I like to describe the three different user types as: root, administrator and SQL user. This comparison is far from being perfect. In fact, the comparison is even wrong in some ways. Nevertheless, the picture will be used. The terms root, administrator and SQL user are much easier to remember the official names for the three different user types. The official names of the different types of users are: Database System Administrator (SYSDBA user), Database Manager operators (DBM operator) and Database users.

*Simplified and not 100% correct!*

### **“root”: Database System Administrator (SYSDBA user)**

First user created during the database installation.

Has all rights of a SQL user plus some of an administrator

### **“Administrator”: Database Manager operators (DBM operator)**

Administrator who can use the DBM Clients (Database Manager GUI, DBMCLI, WebDBM)

### **“SQL user”: Database users**

SQL user who can only use SQL Tools like SQL Studio, SQLCLI (+ programming interfaces, Loader...)

Usually the SYSDBA (“root”) is the first database user that gets created during the creation of a database instance. The SYSDBA belongs to the group of database users (“SQL users”) but has also all rights of a Database Manager operator (DBM operator, “Administrator”) plus some more. Default names of the SYSDBA are DBADMIN and DBA. Every database instance has only one SYSDBA.

A user of the user type Database Manager operator (DBM operator) is kind of an administrator. Only a DBM operator can log in to the DBM Clients (DBM GUI, DBMCLI, WebDBM - see also [last week’s posting](#)) and perform administrative tasks like backup and recovery. You can create as many DBM operators as you want. A DBM operator is not meant to perform SQL queries.

Performing SQL queries is the task of the Database users (“SQL users”). A user of the user type database users can log in to the SQL tools (SQL Studio, SQLCLI, WebSQL), use a programming interface (JDBC, ODBC, PHP, Perl, ...) or the Loader to work with the SQL objects stored in the database. Database users cannot administrate the database instance, for example they cannot create a database backup.

## **“root”: The Database System Administrator (SYSDBA user)**

Every MaxDB database instance has exactly one Database System Administrator (SYSDBA user). The SYSDBA is the first database user that you create when you create a database instance with the DBM GUI or the Installation Manager. As of database version MaxDB 7.6.00, the default value for the name of the database system administrator is DBADMIN. Typical default passwords are DBADMIN and SECRET. MaxDB 7.5.00 has used the name DBA and the password DBA as a default. It will be shown later that DBA has been a confusing default name. SAP installations using MaxDB might use different default names and passwords, search the [manual](#) for “SAP Standard Users for Databases”.

The SYSDBA has the following properties:

- has all the server authorizations that a Database Manager operator (“Administrator”) can have
- has all the rights that a database user (“SQL user”) of the DBA database user class can have
- has owner rights over all users
- can create database users of any database user and Database Manager operators (DBM Operator)
- is the owner of a number of system tables

For beginners it is very convenient to work with the SYSDBA. The SYSDBA is the most privileged user of a database instance. He can perform administrative tasks as well as running SQL queries. Beginners do not need to worry about the details of the user concept and authorizations. Of course this is also very dangerous in some ways. In production environments you should restrict the use of the SYSDBA (“root”) to a bare minimum, just like you do with the Unix operating system user root. The simple reason is paranoia. Time will come you will make a mistake and do some harm because you have been using a privileged user for a operation that should have not been allowed to you. One day you might sit in front of your computer - like I did - and chat with your colleagues about the question what to do after you have run `rf -rf . in / as root`. Amazingly, the system continued running but of course I could not restart the computer any more with an empty disk.

You should never, ever forget the password of the SYSDBA. Of course there is no way to extract the password from the database files and you are unable to perform some actions if you do not know the

password. Simply never forget the password. Maybe you can put a note with it on your monitor or use one of the default passwords... Is that true? Discuss it in the forum. We will give the answer to this exercise in the next lesson.

The SYSDBA has owner rights over all users. All users means that he has owner rights over all users of the user type Database user ("SQL users") and all users of the user type DBM operator ("Administrators"). Being the owner of another user means that you manage the privileges of another user. The owner of a user can grant more rights and revoke rights from the user he owns, but the owner can also delete the user. Additionally to managing existing users, the SYSDBA can create new users of all types.

## “Administrator”: Database Manager operators (DBM operator)

Let's see what the manual says about a DBM operator ("Administrator"): "Database Manager operators (DBM operators) can log on to the Database Manager database tool and use its administration functions. Database Manager operators are not database users and consequently cannot log on to the query tools (such as SQL Studio) or the Loader. When you create a database instance with the Database Manager or the Installation Manager, the respective tool creates a first Database Manager operator that has the required DBM operator properties and all server authorizations. The default value for the name of the Database Manager operator is DBM."

Again, the installation wizards will create a user for you. If you followed the MaxDB series from the beginning and if you have done everything we described, then the Installation Manager that you used, has created a first DBM operator for you. The first DBM operator has been granted all server authorizations. The user has been given the user name DBM.

The default passwords that the installation wizards use are DBM and SECRET. In our series the Installation Manager was used to install MaxDB. This has caused a tiny pitfall: the Installation Manager does not use one of the before mentioned default passwords. The first database manager operator, the DBM user has been given the password that you have specified for the SYSDBA. We told you in the [Installation lesson](#) to use the user name DBADMIN and the password DBADMIN for the SYSDBA. Respectively, the first DBM user was given the password DBADMIN by the Installation Manager. You may check this now and try to log in to the DBM GUI and connect to your MAXDB1 database instance using the user name DBM and the password DBADMIN.

The manual continues to explain: "Database Manager operators can create, change or delete other Database Manager operators, if they have the respective server authorizations. If a Database Manager operator has the corresponding authorization, it can also display or change the DBM operator properties and server authorizations of another DBM operator."

This means that you can have multiple DBM operators. DBM operators can be given certain server rights, certain server authorizations. Strictly speaking there is a small semantical difference between server rights and server authorizations, but we will ignore this difference for simplicity and use the terms interchangeably. Every server right allows you to perform certain actions. One of the rights is the User management right. Every DBM operator that has this right can administrate the rights of all other DBM operators. Be careful, when you assign the User management right to someone. There is no hierarchy in the relationship of DBM operators. If you grant the User management right to a new DBM operator, the new DBM operator could revoke the User management right from the grantor.

The following server authorizations exist. Check the table carefully and ask your self why we said that the picture "SQL user" vs. "Administrator" is not perfect.

Permissions	ID in the Database Manager GUI	ID in the Database Manager CLI
Accessing SQL sessions	Access to SQL session	AccessSQL
Accessing utility sessions	Access to utility session	AccessUtility
Performing backups	Saving backups	Backup
Accessing database files (read-only)	Database file access (read	DBFileRead

	only)	
Displaying status information	Request status data	DBInfoRead
Starting the database instance	Start database instance	DBStart
Stopping the database instance	Stop database instance	DBStop
Executing the Loader Program	Execute the LOAD program	ExecLoad
Installation management	Installation management	InstallMgm
Loading the system tables	Load the system tables	LoadSysTab
Accessing database parameters (reading and writing; only those commands for changing parameters may be executed whose changes are checked with the default parameter file cserv.pcf)	Parameter access (checked write)	ParamCheckWrite
Accessing database parameters (reading and writing; all commands can be executed, even those for directly changing the parameter file without the changes being checked by the system)	Parameter access (read and write)	ParamFull
Accessing database parameters (read-only)	Parameter access (read only)	ParamRead
Restoring backups	Restoring backups	Recovery
Managing the Scheduler	SchedulerMgm	Scheduling DBM commands
Scheduling	Managing the shared memory	SharedMemoryAdmin
Executing operating system commands	Execute operating system commands	SystemCmd
Managing DBM operators	User management	UserMgm

## Managing DBM Operators

Before we dive into the details of the Database users ("SQL users") and schemata, we will take the opportunity to create a new DBM operator and assign some rights to the user. Managing DBM Operators in the DBM GUI is very easy. Open the DBM GUI and connect to your MAXDB1 database. Select Configuration - DBM Operator ... from the menu list of the current database instance or use the window menu Instance - Configuration - DBM Operator... to open a popup window for user management. That's it, it should not require a manual to learn how to use it.

The more experience you get and the more MaxDB installation you have to administrate, the more likely that you will use the text based DBMCLI and scripts for administration. Don't be much afraid of using the DBMCLI. Remember that DBMCLI and DBM GUI are clients of the DBM Server. The DBM Server writes a protocol file with all DBM commands that the clients send to the server. If you're looking for a specific DBM command, it can be helpful to use the DBM GUI first and then to reverse engineer the DBM commands you need for the DBMCLI based on the DBM Server protocol file. We described this in our [last posting](#).

### Example: creating a start-stop user

Let's start with a DBM command that shows a list of all existing DBM operators, similar to the list in the popup windows of the DBM GUI: `user_getall`. In addition to the user list we want to check the properties of one of the DBM operators using the DBM command `user_get`. To execute any of the two commands, you need the DBInfoRead server authorization. Log in to the DBMCLI using the SYSDBA user which has the username DBADMIN and the password DBADMIN if you did not overrule any of the defaults that we suggested in the class. The SYSDBA has all server authorizations that a DBM operator can have. This includes the required DBInfoRead server authorization.

```
$ /opt/sdb/programs/bin/dbmcli -u DBADMIN,DBADMIN -d MAXDB1
dbmcli on MAXDB1> user_getall
OK
dbm
dbadmin

---
dbmcli on MAXDB1> user_get dbm
OK
SERVERRIGHTS=DBInfoRead, SystemCmd, ExecLoad, UserMgm, DBFileRead, Backup, InstallMgm, LoadSysTab, Pa
ramCheckWrite, ParamFull, ParamRead, DBStart, DBStop, Recovery, AccessSQL, AccessUtility, SharedMemor
yMgm, SchedulerMgm, Scheduling, EvtDispMgm, EvtDisp
GUIRIGHTS=
SECONDPASSWORD=NO
DISABLED=NO
COMMENT=
USERTYPE=DBM

--
```

Create a new DBM operator called “DBM\_STARTSTOP” with the password “STARTSTOP\_PASSWORD” using the DBM command `user_create`. We will give this user the right to start and stop the MAXDB1 database instance in the following.

Quit the DBMCLI session and log in to the DBMCLI again using the newly created user. Try to execute `user_getlist`. The DBM server will not allow you to execute the command, because the newly created user has not been given any server right. The user DBM\_STARTSTOP does not have the server authorization DBInfoRead which is needed to execute the `user_getlist` command. MaxDB bails at you.

```
dbmcli on MAXDB1>user_create DBM_STARTSTOP,STARTSTOP_PASSWORD
OK

---
dbmcli on MAXDB1> quit
OK

---
$ /opt/sdb/programs/bin/dbmcli -u DBM_STARTSTOP,STARTSTOP_PASSWORD -d MAXDB1
dbmcli on MAXDB1> user_getall
ERR
-24937,ERR_MISSRIGHT: no server rights for this command

---
dbmcli on MAXDB1> quit
```

You can assign the “DBStart” and “DBStop” to the user DBM\_STARTSTOP by help of the command `user_put`. Use [Glossary - DBM Command](#) to navigate to the overview of all DBM commands and check the syntax of `user_put`. Please open the manual page now, you should learn how to look up the syntax and how to “read” the manual. The manual says the following.

#### Use

You can change the DBM operator properties and server authorizations of DBM operators and of the database system administrator.

#### Caution

See Conventions for User Names and Passwords.  
The database system automatically converts passwords for DBM operators into upper-case letters. [Note of the editor: possible pitfall]

#### Prerequisites

- You have the server authorization UserMgm.
- If you want to change the password of the database system administrator, the database instance must be in the ONLINE operational state.

#### Syntax



```
user_put    :: = |
:: = =
:: = =
:: = =
[...]
```

The manual continues to explain that property can be any of: PASSWORD, COMMENT, SECONDPASSWORD, DISABLED and SERVERRIGHTS. The idea behind SECONDPASSWORD is that you do not need to give the support your original password but you can temporarily assign a user a second password that you can give to the support engineers for the time of their work. Obviously, SERVERRIGHTS is the property needed to assign server rights.

A list of possible values for SERVERRIGHTS has been given above. If you want to grant not only one server right to a user using one user\_put call, you can specify a list of values. The list of rights has to be separated by commas. Every server right in the list must be preceded by a plus (+) or a minus sign (-). Putting a plus in front of the right means that you want to grant the right, a minus sign in front of the right means that you want to revoke it. This gives us:

```
$ /opt/sdb/programs/bin/dbmcli -u DBADMIN,DBADMIN -d MAXDB1
dbmcli on MAXDB1>user_put DBM_STARTSTOP SERVERRIGHTS=+DBStart,+DBStop
OK

---
dbmcli on MAXDB1>user_get DBM_STARTSTOP
OK
SERVERRIGHTS=DBStart,DBStop
GUIRIGHTS=
SECONDPASSWORD=NO
DISABLED=NO
COMMENT=
USERTYPE=

---
dbmcli on MAXDB1>user_getrights DBM_STARTSTOP SERVERRIGHTS
OK

DBInfoRead          -      Request status data
SystemCmd            -      Execute operating system commands
ExecLoad             -      Execute the LOAD program
UserMgm              -      User management
DBFileRead           -      Database file access (read only)
Backup               -      Saving backups
InstallMgm           -      Installation management
LoadSysTab           -      Load the system tables
ParamCheckWrite      -      Parameter access (checked write)
ParamFull            -      Parameter access (read and write)
ParamRead            -      Parameter access (read only)
DBStart              +      Start database instance
DBStop               +      Stop database instance
Recovery             -      Restoring backups
AccessSQL            -      Access to SQL session
AccessUtility        -      Access to utility session
SharedMemoryMgm     -      Shared memory management
SchedulerMgm         -      Scheduler management
Scheduling           -      Scheduler use
EvtDispMgm          -      Event Dispatcher management
EvtDisp              -      Event Dispatcher use
```

Quit the DBMCLI and log in to the DBM GUI using the user DBM\_STARTSTOP. Ask yourself why you are getting error messages, read the error messages carefully and you will find out. When you are done with your testing, use user\_delete to drop the DBM operator DBM\_STARTSTOP.

That's been a lot of "ugly", "difficult to understand" command line "hacking". **Always remember that you do not need to know about any of these commands, if you are using the DBM GUI.** The wizards of the DBM GUI will hide all this from you. Especially for beginners it is highly recommended to get started with the DBM GUI before you start writing DBMCLI scripts. If you want to start writing shell scripts right now, recall the help output of DBMCLI and the following options on input, output and protocol files:

```
$ /opt/sdb/programs/bin/dbmcli -h
```

```
[...]
-i    inputfile      (Default stdin)
-ic   inputfile      (Default stdin)
-o    outputfile     (Default stdout)
-t    protocolfile   (writes subsequent protocol)
[...]
```

## The User Profile Container

Usually you do not care where the database stores user informations. One situation when you do care is, when you remove database files or when your harddisk gets corrupted and database files are affected from the corruption. Removing database files without being 100% sure that you can remove them without doing any harm is like shooting yourself in your feet. It hurts and nobody will feel sorry for you. In the worst case, even the development support will not be able to help you. Never forget the SYSDBA password, never remove any database files unless you know what you are doing.

The user profile container is such a file that you should not modify or delete. The user profile container is the place where MaxDB stores the user informations of all DBM operators and the SYSDBA user. If the DBM operator and SYSDBA user informations would be stored in the database catalog, it would not be possible to connect to the DBM server if the database is in OFFLINE mode or corrupted. The user profile container does not store informations about the "SQL users". "SQL users" are kept inside the database in the database catalog.

Two copies of the user profile container exist. The original file is `/config.upc` and a copy exists in a `/dbm.upc`. We demonstrated in the [last lesson](#) how to find the actual values of `and` . MaxDB versions prior to 7.6.00 might use a different place to store the copy of the user profile container.

```
$ ls -la /var/opt/sdb/data/wrk/MAXDB1/dbm.upc
-rw-rw---- 1 sdb sdba 3072 2006-03-08 12:48 /var/opt/sdb/data/wrk/MAXDB1/dbm.upc

$ ls -la /var/opt/sdb/data/config/MAXDB1.upc
-rw-rw---- 1 sdb sdba 3072 2006-03-08 12:48 /var/opt/sdb/data/config/MAXDB1.upc

$ diff /var/opt/sdb/data/wrk/MAXDB1/dbm.upc /var/opt/sdb/data/config/MAXDB1.upc
```

As you can see, the files belong to the operating system user "sdb" and the operating system user group "sdba". This is important to protect the files from users who should not know about the database operators and it is required to ensure that the DBM server can read and write the two files. When you look into the file, you'll see that all contents are encrypted except the user names. This is helpful for debugging.

The database messages -24950 ERR\_USRFAIL, -24951 ERR\_USRREAD, -24952 ERR\_USRSAVE can indicate problems with the user profile container. Most likely -24950 ERR\_USRFAIL was caused by you, because you have used a wrong user name but it could also be a corrupted user profile container, which is likely the case for -24951 ERR\_USRREAD and -24952 ERR\_USRSAVE. If you get any of these messages start to debug the case:

1. Ensure the user exists, use DBMCLI and `user_get|user_getall` or DBM GUI to verify this
2. Ensure the user in question is a DBM operator or the SYSDBA, the user profile does not contain contain "SQL users"
3. Check the operating system level file permissions for both \*.upc files, they must be as shown above
4. Use `grep` to ensure the that the user is contained in both \*.upc files

If all fails and one of your \*.upc files is missing or corrupted, you might still able to rebuild it from the second one. When you loose the copy `/dbm.upc`, then the DBM server will re-create it based on the original in `/config.upc` . When you loose the original file `/config.upc`, it is slightly more complicated. Open a DBM session using DBMCLI or DBM GUI. You can still log in to the database, because MaxDB can check your credentials against the copy `/dbm.upc`. Switch the database in online mode and reload the system tables with `load_sysstab` . This will cause the original file to be recreated. Only

the SYSDBA and DOMAIN user will be recorded in the file. Not perfect, but you're "up again". Loosing both \*.upc-files is a very bad idea. Simply put the two directories on two different disks, keep your operating system backup up to date and never play with your database internal files and you will hardly ever deal with this problem.

## “SQL users”: Database users of the user classes DBA, RESOURCE and STANDARD

The counterpart to the DBM operators (“Administrators”), who are allowed to use the DBM GUI/DBMCLI/WebDBM for database administration, are the Database users (“SQL Users”). Database users can only log in to the SQL tools like SQL Studio and SQLCLI. Additionally they can use any other SQL clients like the Loader or a programming interface to connect to the database.

Three different classes of database users exist: DBA, RESOURCE and STANDARD. A user class implies certain rights. The rights are not bound to any specific SQL objects. The most powerful “SQL user” is a user of the class DBA (database administrator). Note that the term administrator in DBA (database administrator) refers only to the SQL world. It is not about managing the database system/database software. Unfortunately in some older versions of MaxDB the default user name of the SYSDBA was DBA! A user with the name DBA is not necessarily the same as a “SQL user” of the user class DBA. When you talk someone about “the DBA”, make sure you talk about the same things.

A database user of the user class DBA is the most privileged “SQL user”. A DBA can create other database users of the user classes RESOURCE and STANDARD. He can create all kind of database object (SQL objects) and grant of all or part of privileges for these database objects to other database users.

Next in row from most privileged to least privileged users comes a database user of the user class RESOURCE. A RESOURCE user can define data and database procedures. Data means tables, views, ... and so on. The user can grant privileges of these database objects to other users. But a RESOURCE user cannot create any users.

If your are a STANDARD user, you are restricted to accessing data and database procedures that were defined by other database users and for which the user has been given privileges. All you can create are views, synonyms and temporary tables. You cannot create users, you cannot create base tables nor can you grant privileges for any of the database objects you can access to other users. This means, you're basically restricted to work with the existing database objects.

The SYSDBA is the only user who can create database users of all classes. The SYSDBA can create DBA, RESOURCE and STANDARD users. Only the SYSDBA can create a DBA user. An existing DBA cannot create another DBA. Naturally, DBM operators (“Administrators”) also cannot manage any of the “SQL users”, including the DBA.

So, unlike to the world of DBM operators (“Administrators”) there is a clear hierarchy in managing “SQL users”. Among the DBM operators a user only needs the server right UserMgm to manage any of the DBM operators. This is not the case among the database users (“SQL users”). User management rights are bound to the user class. In order to manage a database user, you must have the owner rights of the user. The SYSDBA has owner rights on all “SQL users”. A DBA owns all users he created.

1. SYSDBA — manages —> DBA
2. DBA — manages and grants privileges for his objects to —> his RESOURCE and STANDARD users
3. RESOURCE — grants privileges for his objects to —> all other users
4. STANDARD — is at the end of the food chain —>

SQL users have more properties than “Administrators” and it is likely that you will more frequently adjust SQL users properties than the properties of a DBM operator or the SYSDBA.

Property	Description
General properties	
Name	Name of the database user

Password	Password of the database user
User Class	Database user class (DBA   RESOURCE   STANDARD)
Connection Mode	<ul style="list-style-type: none"> <li>• NOT EXCLUSIVE: allows the user to open several database sessions simultaneously</li> <li>• EXCLUSIVE: prevents the user from opening two different database sessions simultaneously</li> <li>• DISABLED: The database user may not open any database sessions</li> </ul>
Replication	REPLICATION: database user required for the synchronization with the Synchronization Manager
Extended properties	
Timeout	Session timeout value
Cost Warning	Specifies the estimated SELECT cost value beyond which the user receives a warning. If a warning is issued, the user is asked whether he or she really wants to execute the SQL statement
Cost Limit	Specifies the estimated SELECT cost value beyond which the SQL statement is not executed. This value must be greater than the Cost Warning value
Default code	<ul style="list-style-type: none"> <li>• DEFAULT: The default value for the code attribute is defined by the database parameter DEFAULT_CODE</li> <li>• ASCII: The default value for the code attribute is ASCII</li> <li>• UNICODE: The default value for the code attribute is UNICODE</li> </ul>

Managing SQL users with the DBM GUI is very easy. Log in to the DBM GUI and get connected to the MAXDB1 database instance. This time, use the DBM operator with the username DBM and the password DBADMIN. As usual, we assume that no changes to accounts and passwords have been and you are using exactly the setup that was created earlier in the class. Do not log in with the SYSDBA. If you do, we can't demonstrate what we plan to. Select Configuration - Database User... in the menu list of the database instance in the lower left corner of the window or use the menu Instance - Configuration - Database User... to open a dialog for SQL user management. If you did not use the SYSDBA user, an authentication dialog appears first. The reason is that a DBM operator is not allowed to manage SQL users, there is a clear line between both worlds. Only a DBA (database administrator) and the SYSDBA can manage SQL users. As there is no DBA account you could use to authenticate, use the SYSDBA (username: DBADMIN, password: DBADMIN) to log in to the SQL user management dialog. Using the GUI dialog is pretty much like using the GUI dialog for DBM operators. It's hardly possible not to understand how to use the GUI.

The only "surprise" is hidden behind the scenes. SQL users are managed using SQL commands. As a DBA cannot log in to the DBM GUI, he and you should get familiar with the corresponding SQL commands.

- CREATE USER, DROP USER
- ALTER USER, RENAME USER, GRANT USER
- ALTER PASSWORD
- GRANT, REVOKE

## Hands on: Managing two SQL users

After all this theoretical background information, it is time for exercises and examples. We will create two SQL users and grant them access to two tables. Log in to the SQL Studio and get connected to the example database MAXDB1. Ask yourself which is the only user account that you can use to log in to any of the SQL tools at the moment. Correct, the only user that can be used is the SYSDBA. Use the CREATE USER statement to create a new database user of the user class DBA with the username MYDBA and the password MYDBA.

```
::=
```

```
CREATE USER PASSWORD
[ ]
[TIMEOUT ]
[COSTWARNING ]
[COSTLIMIT ]
[[NOT] EXCLUSIVE]
[DEFAULTCODE ASCII]
[DEFAULTCODE UNICODE]
[ ]
| CREATE USER PASSWORD LIKE
| CREATE USER PASSWORD USERGROUP

::= DBA | RESOURCE | STANDARD
::= ENABLE CONNECT | DISABLE CONNECT
::=
```

Use the system table `USERS` to check if the user has been created and what default values have been used by MaxDB for user properties like `TIMEOUT` which have not been specified by you. A complete list of all system tables is contained in the manual, see also [Glossary - System Table](#). You can compare the system tables of MaxDB to a superset of the `SHOW` command and the `INFORMATION SCHEMA` in MySQL.

```
CREATE USER MYDBA PASSWORD MYDBA DBA
//
SELECT * FROM USERS ORDER BY USERMODE, OWNER, USERNAME
```

Open a second SQL Studio (Session - New SQL Studio). Log in to the second SQL Studio, using the newly created `MYDBA` and create a table called "MOVIES" with two columns. Give the first column the name `ID` and the data type `INTEGER`. Use the name `TITLE` and the data type `VARCHAR(30)` for the second column. Insert one record into the table.

```
CREATE TABLE MOVIES (ID INT, TITLE VARCHAR(30))
//
INSERT INTO MOVIES (ID, TITLE) VALUES ('1', 'Wargames')
```

Go back to the original SQL Studio running the SQL session of the `SYSDBA`. Execute exactly the same `CREATE TABLE` statement in the SQL session of the `SYSDBA`. Run `SELECT * FROM MOVIES`.

Why does the `CREATE TABLE` not fail and where is the record? Try the same `SELECT` statement to in the SQL session of `MYDBA`. The record is there and `AUTO_COMMIT` is turned on, so transactions and a missing `COMMIT` are not the cause. The solution to the riddle is, that the user `MYDBA` and the user `DBADMIN` are using two different namespaces. `MYDBA` is using a namespace called `MYDBA` and `DBADMIN` is using a namespace called `DBADMIN`. The correct SQL term is not namespace, but schema (recall: MaxDB schema = MySQL `CREATE DATABASE` = MySQL `CREATE SCHEMA` for recent versions).. That's all what a schema is about: a schema is a namespace for SQL objects like tables. MaxDB 7.6.00 implicitly creates a schema for every user. The name of the schema is identical to the name of the user. When the user opens a SQL session in SQL Studio, the current schema is set to the his schema. Check it with the help of the system table `SCHEMAS`:

```
SELECT * FROM SCHEMAS
//
SELECT CURRENT_SCHEMA FROM DUAL
```

What happened with the two tables is that `MYDBA` has created a table `MYDBA.MOVIES` and `DBADMIN` has created a table `DBADMIN.MOVIES`. . is called a fully qualified name. If you leave out the , MaxDB will use the current schema. The current schema is a property of your SQL session. You can set the current schema using `SET CURRENT_SCHEMA = .` When `DBADMIN` tries to access the table `MYDBA.MOVIES`, he will be told that `MYDBA.MOVIES` does not exist.

```
Auto Commit: On, SQL Mode: Internal, Isolation Level: Committed
Base table not found;-4004 POS(15) Unknown table name:MOVIES
SELECT * FROM MYDBA.MOVIES
```

The reason is that even the user DBADMIN (the SYSDBA in this case) does not have the permissions to access the SQL objects of other users. If you do not know the “trick” of using SELECT \* FROM SCHEMAS to get a list of all schemas, MaxDB does not even tell you that the schema MYDBA exists, unless you have been granted access to the schema. Only the owner of a SQL object - be it a schema or a table - can grant access to the object to other users. Even DBADMIN (the SYSDBA in this case), “root” as we called him before, has no way to workaroud this. That one reason why we said at the beginning the comparisons we made are not perfect.

The user MYDBA is the owner of the table MYDBA.MOVIES and he is the owner of the schema MYDBA. He can grant any other “SQL user” access to the schema or the table within it.

```
GRANT ALL ON MYDBA.MOVIES TO DBADMIN
```

Once this statement is committed, DBADMIN can access the table MYDBA.MOVIES. DBADMIN still cannot access any other tables of the schema MYDBA, but he can access the table MOVIES.

```
Auto Commit: On, SQL Mode: Internal, Isolation Level: Committed
SELECT * FROM MYDBA.MOVIES
Statement successfully executed.
Execution Time: 20:34:21.543 - 20:34:21.562 (00.019 sec)
```

In the example the user DBADMIN has been granted “ALL” permissions on the table MYDBA.MOVIES. Of course, there are much more fine-grained alternatives to “ALL”. The syntax diagram of the SQL statement GRANT shows the options. You can grant access to schema or to individual tables. If you want to give someone permissions to create new SQL objects inside a schema or to drop existing object, you give him the CREATEIN resp. DROPIN schema privileges. On the level of individual SQL object, for example tables, you can give someone the right to alter the table (ALTER), to remove records (DELETE), add an index to the table (INDEX), add new records (INSERT), add referential constraints (REFERENCES), allow him to read records from the table (SELECT), give a combined right for reading and updating (SELUPD) or just give him the allowance to update existing records of the table (UPDATE).

```
 ::= GRANT ,... TO ,... [WITH GRANT OPTION]
 | GRANT TO ,...
 | GRANT EXECUTE ON TO ,...
 | GRANT EXECUTE ON TO ,...
 | GRANT SELECT ON TO ,... [WITH GRANT OPTION]

 ::= | | | PUBLIC

priv_spec> ::= ALL [PRIV[ILEGES]] ON [TABLE] ,...
 | ,... ON [TABLE] ,...
 |

 ::=
 ALTER
 | DELETE
 | INDEX
 | INSERT
 | REFERENCES [(, ...)]
 | SELECT [(, ...)]
 | SELUPD [(, ...)]
 | UPDATE [(, ...)]

 ::= ON ,...
 ::= CREATEIN | DROPIN
```

The counterpart to the GRANT statement is REVOKE.

## In the next issue

Please apologize that we are late with this weeks posting. Again, we missed the magic wednesday. And we are forced to shift the remaining details on SQL users to the next issue. We are forced, because of a few days off and other commitments. Be sure, we continue to explain the concept and details of SQL users in the next lesson. Of course we encourage you to read the manual and find out yourself about the missing details.

Although we can't give you all details today, you are already armed with the necessary information to setup the users in your MaxDB installation. Always keep in mind that MaxDB 7.6.00 makes a clear cut between "SQL users" and "Administrators" - the only exception is the SYSDBA, kind of "root". Be warned that in MaxDB 7.5.00 you will find some more and different users. However, like with MySQL, we ask you to use the latest production versions of the database: 7.6.00 for MaxDB, 5.0 for MySQL.

Use the opportunity to comment to this article on the [MaxDB forum](#). Never hesitate to ask questions, the user concept is not always easy to understand. Talk to you next wednesday ... or thursday!

*Ulf Wendel for the MySQL MaxDB team*

## MaxDB series: SQL users continued

Published: 16.03.2006

### **Dear MySQL users, MaxDB users and friends,**

Several co-workers have spoken to me on the length of the MaxDB series postings. I fully agree that most postings are too long for a blog. But we do have a very eager plan to publish a complete online class. This is causing long postings and we can't make them much shorter. We try to structure the articles in a similar way like a web page to make reading and navigating easier. Every posting has a table of contents, is divided into sections and has a fixed structure. For offline reading, printing and as a reference we will soon publish a PDF document with all postings of the series.

The experiment to use the medium of a blog for a class will continue as long as the readers do not complain. But we will add a "read more" link to the fixed structure of every posting. That means, we will present only the beginning of a posting on PlanetMySQL and you have to click on a "read more" link for the full text which might get you to a different URL. The link will be added shortly after the "In this issue" section of the posting which shows the table of contents. However, this posting is short enough so that it will not use a "read more" link.

### **In this issue**

This issue is a follow-up on the last issue [MaxDB series: User concept, authorization and schemata](#). We continue to explain the user concept of MaxDB. MaxDB distinguishes between "Administrators" - as we called them - and "SQL users". The group of "Administrators" (Database Manager operators, DBM operators) and the "root" user (Database System Administrator, the SYSDBA) have been discussed in depth. The three different classes of "SQL users" (Databases users) have been introduced: DBA, STANDARD, RESOURCE. It has been shown how to grant privileges to an SQL user and what an SQL schema is. Detailed hands-on examples that show how to create users and how to use the SQL statements GRANT and REVOKE are left as an exercise to the reader.

You already learned everything you need to know about the user concept. What follows is only sugar.

- [User groups - CREATE USERGROUP](#)
- [Roles - CREATE ROLE](#)
- [Dinosaurs - DOMAIN, SYSINFO, PERMLIMIT, TEMPLIMIT](#)
- [In the next issue](#)

### **User groups - CREATE USERGROUP**

A user group is a tool to bundle related SQL objects, similar to a schema. The ownership of all SQL object that have been created by a certain member of a user group is given to the user group. All members of the user group are automatically owners of all SQL objects that have been created by any of the other group members. Privileges are not granted to individual users of the group but to the group itself and by this to all its members. Users who are not members of the user group that own a certain object cannot be given access to objects owned by the group.

You should favour the concept of schemas over that of user groups. There's no guarantee this feature will stay in MaxDB forever. Check the [manual](#) for details.

### **Roles - CREATE ROLE**

A role is like a mask that you can wear for a certain time. So far, we have described the relationship of SQL users and permissions as something unflexible. A user (or a user group) has certain permissions over certain objects and can't break out of this cage. With roles a user can break out of the cage.



Every SQL user of the user class DBA (including the SYSDBA) can define a role. A role can be granted permissions, very much like a user. Users can be granted permission to use a role. When users that have been granted access to a role change roles using SET ROLE during a database session, they will be given the permissions that have been assigned to the role. Using roles, a user can use different sets of permissions within one database session.

Be very careful with roles. One can compare roles to sudo/suexec and similar. This is a very mighty feature. You should only give selected and trustworthy users access to privileged roles.

## **Dinosaurs - DOMAIN, SYSINFO, PERMLIMIT, TEMPLIMIT**

All this is not enough for you? Well, you can scan the manual for details on the keywords DOMAIN, SYSINFO, PERMLIMIT and TEMPLIMIT. But be warned. These dinosaurs did already die out or might die out very soon. For example, you will find out that the DOMAIN user no longer exists and is only a schema in recent versions.

## **In the next issue**

This has been the first group of series postings. Similar to what I did at the beginning we will now discuss internally how to improve and what to change. As usual, we appreciate any feedback.

In the next issue we will discuss some interesting details of the MaxDB SQL syntax. But we will limit it to a very high-level discussion. We found that most users are more interested in the administration of MaxDB and only few of you need a SQL course. Most MaxDB users do have experiences with other databases and just want to know what is different and hot.

*Ulf Wendel for the MySQL MaxDB team*

## MaxDB series: when do we publish the next article?

Published: 24.03.2006

**Dear MySQL users, MaxDB users and friends,**

it happened what we always told you what might happen: we did not make it to write an article for you in this week. Of course, this is not the end of the series. It will continue, but other duties might force us in the future to skip a week again. As a small “excuse” for you, we have written an FAQ like entry.

### **Can I recover a MaxDB backup on a different version and/or a different system?**

It depends: the processor architecture and the MaxDB version must be compatible. Given these two main preconditions you can take a backup from one system and recover it on a different system. This is a common situation when you are planning to upgrade the database software and the server hardware in one step. In general, we do not recommend to do these two steps at once. The simple reason that we do not is, that you should never change two variables at once. This makes debugging extremely difficult, if it should become necessary.

**Precondition 1: compatible processor architecture.** The processor architecture on the source and the target systems must be compatible. For the experts among you: it is a question of little vs. big endian. Luckily, the manual has a chart that shows which systems are compatible for all of us that are not CPU experts. Use the glossary in the [MaxDB online manual](#) to check for “Database Copy”. You will find out some facts that might amaze if you are not a CPU expert. For example, you can take a backup on Win32 and recover it on Win64.

**Precondition 2: compatible MaxDB versions.** The MySQL MaxDB history starts with MaxDB 7.5.00, therefore we will not discuss hints for any older versions. For 7.5.00 and 7.6.00 the main precondition of compatibility is that both databases have the same version (7.x.x) in the first three places. The build level of the target instance version is equal to or higher than the build level of the source instance. This means, you can for example create a backup on 7.5.00 Build 23 (7.5.00.23) and recover it on 7.5.00 Build 34 (7.5.00.34), because the first three places of the version string are the same. But you cannot recover a 7.6.00 backup on a 7.5.00 server, nor is there any warranty that you can recover a 7.5.00 backup on a 7.6.00 server!

Additionally you have to take care which version of the DBM GUI you use to do the work. In the first three places, the version of the Database Manager GUI should be higher than or equal to the version of the database kernel. If it is equal, the build level (the last two figures) of the DBMGUI version can be lower than that of the database kernel.

Is that all you need to take care off? No! This is only a description how to find out if the backup files are compatible. Check the manual on hints and instructions how to perform a backup, what to do after a recovery (bad indexes) and what to take care of when you plan to upgrade from one MaxDB version to another.

We hope to have the promised SQL-highlights posting ready for you until next wednesday, happy hacking!

*Ulf Wendel for the MySQL MaxDB team*

## MaxDB series: ...just SQL

Published: 12.04.2006

### Dear MySQL users, MaxDB users and friends,

The previous “please wait” posting was caused by a new guideline on series postings. The new guideline says that postings must be written one week before they get published. The week between the writing and the publication is used for corrections. I’m writing these lines on April, 4th but the article won’t be published before Wednesday April, 12th. We hope that in the time between writing and publishing we can catch most of the Gotchas to further improve the quality of the series. However, this series will continue to be more like a collection of blog postings than a book, a magazine article or any other “official documentation”. We try to do our best, but unfortunately we cannot apply the same amount of quality assurance and proof-reading to the MaxDB series that we do for magazine articles, official documentation or official class materials.

### In this issue

In the past, we recognized that most MaxDB users do have some SQL knowledge. Hardly any of the typical MaxDB users seem to be beginners in the field of databases. Therefore SQL basics are not of interest. For example, most users who asked us SQL questions in the past did know what kind of joins exist, what the difference between a right join, left join and a full join are and how to define tables. Also, there seems to be no demand on describing the basics of views as many articles did when MySQL 5.0 introduced views to the MySQL server.

MaxDB users seem to be more interested in the limits and additional features over the SQL92 Entry Level compatible SQL dialect implemented by MaxDB. For this reason, we will often provide you only with brief overview information on some topics. If you need details, check the [SQL Tutorial](#) chapter in the [MaxDB documentation](#). Other online resources like the [SQL Course](#) and the [SQL Course 2](#) may also be worth reading if you want to learn the very basics. Those who like printed books better and prefer reading on a sunny balcony, can check the [book listing](#) on the [MySQL Developer Zone](#).

- [Data types for table columns](#)
- [Check Constraints](#)
- [Domains](#)
- [Synonyms](#)
- [Indexes](#)
- [Sequences and SERIAL](#)
- [Limiting result sets](#)
- [Cursors and Recursion](#)
- [In the next issue](#)

### Data types for table columns

Let us start the discussion of the MaxDB SQL dialect with something very basic: data types. In general you should always select the smallest data type for a column that can hold all values you want to store in the column. Try to keep your data as compact as possible to save costly disk-read operations and to hold the most records in as short a buffer cache as possible.

For comparisons with MySQL data types, check what mappings have been chosen for the MaxDB plugin of the [MySQL Migration Toolkit](#). The mappings are described in two german language blog postings. Although the texts are written in german, no german language knowledge is required to understand the tables given in the articles on mapping [strings](#), [BLOB/LONG](#), [numeric types](#) and [temporal types](#).

Datatype	Value range	Storage	Notes
----------	-------------	---------	-------

### Numeric types

SMALLINT	-32768 to 32767, 16-bit	5 bytes	Equal to FIXED(5, 0)
INT[EGER]	-2147483648 to 2147483647, 32-bit	7 bytes	Equal to FIXED(7, 0)
FLOAT(p), precision p (0 < p <= 38)	9. 999999999999999999 999999999999999999 999E +62 to 1E-64, 0.0, +1E-64 - +9. 999999999999999999 999999999999999999 999E +62	(p+1) DIV 2	Floating point numbers.  For MySQL users: try SUM(float_column / 3) * 3 or summing up small values. MaxDB FLOAT is MySQL Precision Math.
FIXED (p,s), 0 < p <= 38, s <= p	Depends on p and s, see FLOAT/SMALLINT/I NT	(p+1) DIV 2 + 2	Fixed point number.  MaxDB is using a packed decimal storage format for INTEGER, FLOAT and FIXED values. This means that if you store the same values in three columns of the three data types (with comparable precisions), you will get the same performance when doing comparisons.  For MySQL users: FIXED is for Precision Math.

### String types

CHAR[ACT ER](n), n < = 8000 (UNICODE: 4000)	<a href="#">see Manual</a>		UNICODE requires 2 bytes per character  Code attributes: ASCII, BINARY, UNICODE  Depending on n (<30, >=30) strings will be stored with a variable length
VARCHAR( n), n < = 8000 (UNICODE: 4000)	<a href="#">see Manual</a>		UNICODE requires 2 bytes per character  Code attributes: ASCII, BINARY, UNICODE
LONG [VARCHAR]		up to 2GB of bytes	Also known as BLOB, LOB, etc.  Code attributes: ASCII, BINARY, UNICODE

### Temporal types

TIME	ISO: '00:00:00' to '23:59:59'	9 bytes	Check the manual for database session settings and the database parameter DATE_TIME_FORMAT
DATE	ISO: '0001-01-01' to '9999-12-31'	9 bytes	Check the manual for database session settings and the database parameter DATE_TIME_FORMAT
TIMESTAMP	ISO: '0001-01-01 00:00:00.000000' to '9999-12-31 23:59:59.999999'	21 bytes	Check the manual for database session settings and the database parameter DATE_TIME_FORMAT  Supports Microseconds

### Other types

BOOLEAN	TRUE   FALSE   NULL	2 bytes	
---------	---------------------	---------	--

Check the manual for several aliases for data types, for example DECIMAL, NUMERIC, REAL and many more.

## Check Constraints

As you might have noticed there is no UNSIGNED INTEGER data type in MaxDB like in MySQL. It is not needed, because MaxDB supports check constraints. The check constraint syntax is accepted by MySQL and does not lead to any errors, but MySQL leaves it to the client to check the validity of values.

```
CREATE TABLE test_check_constraints (
  c_int_positive INTEGER CHECK c_int_positive >= 0,
  c_int_negative INTEGER CHECK c_int_negative < 0,
  c_enum CHAR(6) CHECK c_enum IN ('male', 'female')
)
//
INSERT INTO test_check_constraints(c_int_positive) VALUES (-1)

---- Error -----
Auto Commit: On, SQL Mode: Internal, Isolation Level: Committed
General error;300 POS(1) Integrity violation:C_INT_POSITIVE,TEST_CHECK_CONSTRAINTS,DBADMIN
INSERT INTO test_check_constraints(c_int_positive) VALUES (-1)
```

## Domains

In cases where you are using the same check constraints for multiple tables, consider defining a domain. A domain is a value range definition which consists of a data type definition, an optional default specification and an optional constraint definition. Using domains instead of individual definitions in each table helps you to ensure that certain columns always accept the same value ranges.

```
CREATE DOMAIN birthday_domain DATE DEFAULT DATE
  CONSTRAINT birthday_domain > '1880-01-01' AND birthday_domain < = DATE
//
CREATE TABLE people (
  first_name VARCHAR(64),
  last_name VARCHAR(64),
  birthday birthday_domain
)
//
INSERT INTO people(first_name, last_name, birthday) VALUES ('Albert', 'Einstein', '1879-03-14')

---- Error -----
Auto Commit: On, SQL Mode: Internal, Isolation Level: Committed
General error;300 POS(1) Integrity violation:BIRTHDAY_DOMAIN,PEOPLE,DBADMIN
INSERT INTO people(first_name, last_name, birthday) VALUES
('Albert', 'Einstein', '1879-03-14')
```

## Synonyms

MaxDB enables you to define an alternative name (a synonym) for a table. The synonym can be made visible to other users using the keyword PUBLIC. There is not counterpart to synonyms in MySQL. However, you can emulate the feature on MySQL using a view.

```
CREATE [PUBLIC] SYNONYM [<schema_name>]<synonym_name> FOR <table_name>
```

For example you could define an alternative name for the table people created in the last SQL example.

```
CREATE SYNONYM friends FOR people
//
SELECT * FROM friends
```

## Indexes

It seems less known that MaxDB features function-based indexes. Only user-defined functions can be used with a function-based index, but this is no severe restriction. It's simple to use a user-defined function as a wrapper for a build-in function as shown in the following example. The example gives a solution to a common pitfall for users with a MySQL background. MaxDB does always perform case sensitive string comparisons whereas old MySQL versions did case insensitive comparisons. As of version 4.1 of the MySQL server you can use a collation to control the comparison rules. However, MaxDB has nothing comparable: 'ALbert' = 'Albert' and 'ALbert' LIKE 'Albert' both evaluate to false.

```
++++ Execute ++++++

Auto Commit: On, SQL Mode: Internal, Isolation Level: Committed
SELECT 'Condition met, strings are the same' FROM DUAL WHERE ('ALbert' = 'Albert') OR
('ALbert' LIKE 'Albert')
Statement successfully executed. No Result
Execution Time: 17:00:43.071 - 17:00:43.080 (00.009 sec)
```

For a case insensitive search, you have to convert both operands to upper or lower cases, for example: `SELECT * FROM PEOPLE WHERE UPPER(first_name) = UPPER('ALbert')`. It would not be a big deal to use such a query if the query would still profit from an index on the column `first_name`. But this is not the case. A normal index on the column `first_name` cannot be used efficiently, because it stores the original value of the first name - 'Albert' - and not an upper case version of value - 'ALBERT'. All values stored in the index must be converted to upper case before the comparison against `UPPER('ALbert')` can be made. No extra conversion of index values would be needed if the results of the function `UPPER(first_name)` would have been stored in the index. This is what a function-based index does.

```
CREATE FUNCTION my_upper(string_value VARCHAR(64)) RETURNS VARCHAR(64) DETERMINISTIC AS
RETURN UPPER(string_value);
//
CREATE INDEX idx_my_upper_first_name ON people(my_upper(first_name))
//
SELECT COUNT(*) FROM people WHERE my_upper(first_name) = UPPER('ALbert')
```

Note: A bug in the current version of MaxDB 7.6 leads to a wrong cost value calculation for function-based indexes. The optimizer will not use any function-based indexes. The problem will be fixed in 7.6.00.26. You can expect to see a new MaxDB version during the next few weeks that contains the fix.

## Sequences and SERIAL

Sequences are number generators. For example, number generators can be used to create primary key values or to record the insert order of the rows of a table by adding an extra column to the table that gets filled with sequential numbers from a sequence.

```
<create_sequence_statement> ::= CREATE SEQUENCE [<schema_name>.]<sequence_name>
[INCREMENT BY <integer>] [START WITH <integer>]
[<MAXVALUE <integer> | NOMAXVALUE>] [<MINVALUE <integer> | NOMINVALUE>]
[CYCLE | NOCYCLE]
[CACHE <unsigned_integer> | NOCACHE]
[ORDER | NOORDER]
```

Let us assume you have a table of registered users in the database and you want to keep track of the order in which users have registered themselves. The username serves as a natural primary key for the table. The column registration\_timestamp contains a timestamp with the date of the registration. Although the timestamp includes microseconds it still could theoretically happen that two users register at exactly the same point in time. Therefore another column has been added which gets filled with sequential numbers from the sequence registered\_users\_seq. When doing the insert, <schema\_name>.<sequence\_name>.NEXTVAL gets used to retrieve the next number from the sequence. The current number - the last number returned for a <schema\_name>.<sequence\_name>.NEXTVAL call - can be retrieved using <schema\_name>.<sequence\_name>.CURRVAL. Note that you must specify the schema when calling the number generator.

```
CREATE SEQUENCE registered_users_seq
  START WITH 1
  INCREMENT BY 1
//
CREATE TABLE registered_users (
  username CHAR(32) NOT NULL,
  registration_timestamp TIMESTAMP NOT NULL,
  registration_order INT NOT NULL,
  PRIMARY KEY(username)
)
//
INSERT INTO registered_users(username, registration_timestamp, registration_order)
  VALUES ('nixnutz', TIMESTAMP, dbadmin.registered_users_seq.NEXTVAL)
//
SELECT dbadmin.registered_users_seq.CURRVAL FROM DUAL
//
SELECT * FROM registered_users
```

You cannot use <schema\_name>.<sequence\_name>.NEXTVAL for the default specification of a table column. But MaxDB has something that is roughly comparable to AUTO\_INCREMENT in MySQL: SERIAL. A column that has a DEFAULT SERIAL definition will be automatically assigned a sequential number if no other value is given for the column. Note that you can have only one DEFAULT SERIAL column definition per table and that you cannot update the values assigned by the database. This makes SERIAL especially useful for columns tracking the insert order of records as you cannot fake any other orders afterwards by updating the columns with DEFAULT SERIAL values.

```
DROP TABLE registered_users
//
CREATE TABLE registered_users (
  username CHAR(32) NOT NULL,
  registration_timestamp TIMESTAMP NOT NULL DEFAULT TIMESTAMP,
  registration_order INT NOT NULL DEFAULT SERIAL(10),
  PRIMARY KEY(username)
)
//
INSERT INTO registered_users(username) VALUES ('nixnutz')
//
SELECT* FROM registered_users
```

There is more to say about sequences and SERIAL. We strongly recommend that you check the [manual](#) for the details before you start using any of them intensively.

## Limiting result sets

Result sets can be limited to the first n results by adding a ROWNO predicate to the WHERE clause of a SELECT statement. ROWNO is in some ways similar to LIMIT in MySQL, but it is not as powerful as LIMIT. Using ROWNO you can only limit a result set to the first n records. You cannot use ROWNO to get a list of the 1th to the 20th records of the result set. In case you need this, use cursors instead. On the other hand you can add the ROWNO to the list of columns of the result set which has no counterpart in MySQL.

```
SELECT ROWNO, * FROM registered_users WHERE ROWNO < = 3
```

Plans exist to add full LIMIT (TOP) support to future versions of MaxDB.

## Cursors and Recursion

A cursor is a named result table. Functions exist to drop the named result table and to access its records in sequential and random order. The functions to access cursors are provided by the programming interfaces, for example JDBC, DBI and ODBC. But cursors can also be accessed using SQL commands. It is a bit difficult to demonstrate the use of the SQL commands. The SQL Studio uses ODBC to communicate with MaxDB. ODBC does not support the SQL commands. Therefore, we have wrapped a simple database procedure around the SQL commands for working with cursors. This way, the commands are run "inside" the server. Alternatively we could have used a non-ODBC client (e.g. sqlcli), but we assume that most readers are using some sort of ODBC-based GUI to access MaxDB.

When you are new to database procedures, pay special attention to all loops inside a procedure. Ensure that the loops will not run indefinitely but will terminate. Hint: \$RC is set not only by FETCH, but also by INSERT!

In the example a loop gets used to fetch the 1st, 3rd, 5th and so on user name from the table registered\_users to insert it into another table. Instead of calling FETCH NEXT twice to skip one record, you could calculate an offset value and use FETCH POS(offset) instead. Note that inside a database procedure you do not need to open a cursor explicitly. Please check the [manual](#) on further important syntax details.

```
CREATE DBPROC BASIC_CURSOR_EXAMPLE AS
  VAR username CHAR(64);

  DECLARE users_cursor CURSOR FOR
    SELECT username FROM DBADMIN.registered_users ORDER BY registration_order;

  FETCH NEXT users_cursor INTO :username;
  WHILE $RC = 0 DO BEGIN
    INSERT INTO DBADMIN.selected_users(username) VALUES (:username);
    FETCH NEXT users_cursor INTO :username;
    FETCH NEXT users_cursor INTO :username;
  END;

  CLOSE users_cursor;
```

The hottest aspects of MaxDB cursors is that they can be used for recursive queries. Recursion is particularly helpful to traverse tree structures. A classical example of a tree structure is the reporting structure of a company: Ulf reports to Patrik, Patrik reports to Kaj and Kaj reports to Marten. It is difficult to retrieve a list of all employees reporting to Kaj using classical SQL and you have to use a "trick" like "Nested Sets". With MaxDB, however, you can create a recursive cursor that builds an initial result set and adds new records to it until it does not find any further hits, then returning the entire result set to you.

```
DECLARE <result_table_name> CURSOR FOR WITH RECURSIVE <reference_name> (<alias_name>,...) AS
(<initial_select> UNION ALL <recursive_select>) <final_select>
```

The syntax of a recursive cursor reflects the three steps explained below. A recursive cursor consists of an initial select, a recursive select and a final select. When the cursor gets executed and the names result table gets built, the initial SELECT is executed first. We will show step by step how it works, and explain the complete example below.

```
DECLARE report_cur CURSOR FOR WITH RECURSIVE
  emp(emp_name) AS (SELECT employee FROM DBADMIN.employees WHERE boss = :boss_name
  UNION ALL SELECT employee FROM DBADMIN.employees INNER JOIN emp ON employees.boss =
  emp.emp_name)
  SELECT emp_name FROM emp ORDER BY emp_name
```

The example shows how to find all people reporting to Kaj. In the example, the initial select is used to retrieve all employees that report directly to Marten.



Original data set (table employees)		Result set of the initial select (reference name of the intermediate result set used during recursion)
Initial select: <b>SELECT employee FROM DBADMIN.employees WHERE boss = 'Kaj'</b>		
boss	employee	emp_name
NULL	Marten	Kaj
Marten	Kaj	
Kaj	Patrik	
Patrik	Ulf	

The initial select identifies Kaj as the person reporting directly to Marten and adds Kaj to an intermediate result set. The intermediate resultset has been given the reference name emp. This reference name can be used by the recursive query which is executed next. The recursive query joins intermediate resultset with the original data set and looks for all people reporting to Kaj. It finds Patrik. The UNION ALL part of the recursive cursor statement adds Patrik to the intermediate result. As there have been results, the execution of the recursive query is repeated. This time, the query looks for all people reporting to Patrik and adds Ulf. As there have been results, the execution of the recursive query is repeated, but there is nobody reporting to Ulf. No more records are added to the intermediate result set and MaxDB continues with the final step, the final select.

Intermediate result set (reference name: emp)		Intermediate result set (reference name: emp)
1st recursion: <b>SELECT employee FROM DBADMIN.employees INNER JOIN emp ON employees.boss = emp.emp_name</b>		
boss	employee	emp_name
NULL	Marten	Kaj
Marten	Kaj	Patrik
Kaj	Patrik	
Patrik	Ulf	

2nd recursion: <b>SELECT employee FROM DBADMIN.employees INNER JOIN emp ON employees.boss = emp.emp_name</b>		
boss	employee	emp_name
NULL	Marten	Kaj
Marten	Kaj	Patrik
Kaj	Patrik	Ulf
Patrik	Ulf	

The final select returns all records from the intermediate result set ordered by the column emp.

emp_name
Kaj
Ulf
Patrik

As mentioned above, the cursor examples are wrapped in a database procedure to allow you to run the commands even on ODBC-based GUI tools like SQL Studio.

```
CREATE TABLE employees (
  boss VARCHAR(64),
  employee VARCHAR(64)
)

INSERT INTO employees(boss, employee) VALUES (NULL, 'Marten')
//
INSERT INTO employees(boss, employee) VALUES ('Marten', 'Kaj')
```

```
//
INSERT INTO employees(boss, employee) VALUES ('Kaj', 'Patrik')
//
INSERT INTO employees(boss, employee) VALUES ('Patrik', 'Ulf')
//
CREATE TABLE reports_to(emp VARCHAR(64))
//
CREATE DBPROC report_structure(IN boss_name VARCHAR(64)) AS
  VAR emp_name VARCHAR(64);

  DECLARE report_cur CURSOR FOR WITH RECURSIVE
    emp(emp_name) AS (SELECT employee FROM DBADMIN.employees WHERE boss = :boss_name
    UNION ALL SELECT employee FROM DBADMIN.employees INNER JOIN emp ON employees.boss =
emp.emp_name)
  SELECT emp_name FROM emp ORDER BY emp_name;

  FETCH NEXT report_cur INTO :emp_name;
  WHILE $RC = 0 DO BEGIN
    INSERT INTO DBADMIN.reports_to(emp) values (:emp_name);
    FETCH NEXT report_cur INTO :emp_name;
  END;

  CLOSE report_cur;
//
DELETE FROM reports_to
//
CALL report_structure('Marten')
//
SELECT * FROM reports_to
```

## In the next issue

This issue has focused on several interesting details of the MaxDB SQL dialect. It might have looked like a “random” selection and in some ways it has been such. The next issue will also be on the SQL dialect, but it has only one topic: transactions. We will repeat the basic properties of transactions, explain about isolation levels, locking, subtransactions and say some words about deadlock detection.

Meanwhile you could check the manual for system triggers. System triggers are called after the start of a database instance... Check it out!

*Ulf Wendel for the MySQL MaxDB team*

## MaxDB series: news but no series posting

Published: 19.04.2006

### **Dear MySQL users, MaxDB users and friends,**

we have two news for you. The bad one: no MaxDB series posting this week. The good one: german language MaxDB web seminar tomorrow!

My last week has been dominated by preparing the upcoming MaxDB web seminar on MaxDB performance tuning. The web seminar will be held in german tomorrow Thursday, 21.04.2006 at 10 CEST (MEZ). You can still register for the seminar at <http://www.mysql.de/news-and-events/web-seminars/maxdb-performance.php>. Don't expect too much from the 45-minute talk. In 45-minutes one can't do much more than talk about the very basics and try to give you an overview on the topic. Due to the web seminar and other duties we did not make it to write the next MaxDB series posting on transactions.

We have more good news for you. My co-worker C.J. Collier is progressing well in his work to integrate MySQL as a source in the MaxDB Synchronization Manager. Scripts have been developed to create the so-called Shadow and Version tables. These tables contain the meta-data required by the Synch Service to push changes to other participants in the synchronization scenario. C.J. is currently developing the MySQL 5.x triggers which will be used to fill these tables.

No new MaxDB version is on the short-term horizon. But the Changelog of the next MaxDB 7.6 version might show some long-awaited changes, originally planned for MaxDB 7.7. MaxDB will implement data clustering features. Data clustering means that you can decide if all individual database pages of a table get striped over all configured volumes or if chunks of pages are striped. Striping of all pages is good if you have many concurrent high-qualified record accesses, because you are automatically striping the I/O load over all volumes and you are avoiding hot-spots. Grouping database pages in ordered chunks is an advantage if you have few concurrent sessions accessing the same data but you want to profit from sequential read performance and a better support of hardware-based prefetching. Having the freedom to choice on a per-table basis what kind of storage you prefer is a fantastic feature. Applications with few concurrent users and frequent scan operations should see notable performance improvements.

This is still not all... It looks as if some 7.7 optimizer improvements found their way into 7.6. Stay tuned, and join the MaxDB talks at the [MySQL User Conference](#) in the next week.

*Ulf Wendel for the MySQL MaxDB team*

## MaxDB series: Transactions

Published: 17.05.2006

### **Dear MySQL users, MaxDB users and friends,**

In this issue of the MaxDB series we will introduce you to some selected aspects of transactions. The topic of transactions seems pretty boring. But in fact it depends very much on the point of view you take. We have tried to put a light on the subject that shows some rather advanced aspects of transactions which are a little less often discussed in database classes. If you are new to the subject of transactions, make sure you read a introduction to the concept of transactions before you continue reading. The [MySQL Developer Zone](#) features a good introduction. The article [MySQL Tutorial Chapter 10: Using Transactions with InnoDB Tables](#) is an excerpt from the book "MySQL Tutorial - A concise introduction to the fundamentals of working with MySQL" written by Luke Welling and Laura Thomson.

### **In this issue**

- [Subtransactions](#)
- [Concurrency, Locking and Isolation levels](#)
- [Implementation details](#)
- [In the next issue](#)

All SQL commands that are executed in MaxDB are embedded in ACID-compliant transactions. MaxDB does not have different storage engines like the MySQL Server. There is "only one storage engine" in MaxDB and it is a transactional one, just like in many other databases such as Postgres, DB/2, Oracle, SQL Server and so on.

Whenever you open an SQL session, you open a transaction in MaxDB. Either you are running in autocommit mode, which is the default in SQL Studio, or you are using the SQL commands COMMIT and ROLLBACK to control the flow of a transaction. This should be no surprise to you, but have you ever heard of subtransactions?

### **Subtransactions**

Technically speaking subtransactions are a first step from a flat transaction model to a nested transaction model. In a flat model you cannot nest transactions into each other. You have to end a running transaction before you can start a new one. Starting a transaction within another transaction is not permitted. For most applications and in most transaction processing systems this is all what you need. As it is all you usually need and it is simpler to implement, most database systems offer a flat transaction model.

The transaction model of MaxDB is a bit more powerful and goes one step into the direction of nested transactions. A nested transaction model allows you to start a transaction within another transaction. This is particularly useful if you have long transactions where a rollback is very expensive.

Here is the basic idea. Say you are developing a travel shop that has one unique selling point: you do not sell only individual tickets but bundles of individually selected tickets. This means, your customer can book a ticket for their trip from Hamburg to Berlin, book a hotel reservation, book a ticket from Berlin to Munich another hotel and a flight back in one step, in one transaction. Booking all these tickets and hotels requires many individual steps in the booking transaction. Some of them might be expensive. What if all steps of the transaction but the flight booking can be successfully executed. With flat transactions you could do nothing but rollback the entire transaction and start from the very beginning. But with nested transactions you could wrap the flight booking in it's own subtransaction (nested transaction) within the booking transaction. If the flight booking subtransaction fails for some reason, the outer booking transaction can decide if the entire transaction should be rolled back or if the

deal should be closed without the flight or if another flight should be booked, etc. This extra of control might be exactly the feature that is needed to prevent the expensive rollback of the entire transaction.

In MaxDB you can start a nested transaction with the SUBTRANS statement. Make sure that you turn off the autocommit mode of your client before you try the following example. If you are using the SQL Studio, check the appropriate select box to turn the autocommit mode off or use the menu entry Session -> Settings -> Query Dialog to disable autocommit. In SQLCLI you can use “\a off” to disable autocommit.

```
//
// Turn off autocommit in advance!
//
BEGIN TRANSACTION
//
CREATE TABLE subtrans_test(id INT)
//
INSERT INTO subtrans_test(id) VALUES (1)
//
SUBTRANS BEGIN
//
INSERT INTO subtrans_test(id) VALUES (2)
//
SELECT * FROM subtrans_test
//
SUBTRANS ROLLBACK
//
SELECT * FROM subtrans_test
//
ROLLBACK
```

The example above shows one transaction with a nested subtransaction. In the outer transaction a table “subtrans\_test” gets created and one record gets inserted into the table. After that a subtransaction gets started with SUBTRANS BEGIN. Within the sub-transaction another record gets inserted and the result is made visible with the SELECT. Then the subtransaction is rolled back using SUBTRANS ROLLBACK, a select gets executed to verify that the table “subtrans\_test” contains only one record and the effects of the subtransaction have been undone, before the entire (outer) transaction gets rolled back.

```
//
// Turn off autocommit in advance!
//
BEGIN TRANSACTION
//
CREATE TABLE subtrans_test(id INT)
//
INSERT INTO subtrans_test(id) VALUES (1)
//
SUBTRANS BEGIN
//
INSERT INTO subtrans_test(id) VALUES (2)
//
SELECT * FROM subtrans_test
//
SUBTRANS BEGIN
//
INSERT INTO subtrans_test(id) VALUES (3)
//
SUBTRANS END
//
SUBTRANS ROLLBACK
//
SELECT * FROM subtrans_test
//
ROLLBACK
```

It is allowed to nest several subtransactions. You can start a subtransaction from within another subtransaction. However, note that if you nest subtransactions, the outermost SUBTRANS ROLLBACK or SUBTRANS END determines if any of the actions made by all the nested subtransactions are rolled back or committed.

Before you start using subtransactions, for example to isolate the effects of stored procedures, check the [MaxDB manual](#) for more details. There are some details you should know about. One detail is that subtransactions do not affect locks assigned to the (outer) transaction.

Locks? Yes. Locks play an important role in databases to synchronize concurrent accesses to shared resources. Such resources can be any kind of memory areas used by the database server like the I/O buffer cache or SQL objects like tables during a transaction. The less locks need to be set, the less work needs to be done and the lower is the propability that one transaction has to wait for another transaction to release a lock on an SQL object before the transaction can continue its work. In other words: the number of locks has a direct influence on concurrency and parallelism.

## Concurrency, Locking and Isolation levels

Which locks are implicitly set during the execution of a transaction can be controlled with the Isolation levels. MaxDB supports all four ANSI SQL Isolation levels: (1) READ UNCOMMITTED, (2) READ COMMITTED, (3) REPEATABLE READ and (4) SERIALIZABLE. The list of Isolation levels is sorted by the number of locks that MaxDB sets. This means, that the fewest locks are set by the database when using the level READ UNCOMMITTED. Therefore, the most probability of collisions and waits is the lowest when you use that level and the parallelism is likely to be the best. But the problem is that three phenonema can occur in all Isolation levels but SERIALIZABLE. The three phenonema are Dirty Read, Non Repeatable Read and Phantom. In general you do not want to see any of them happen in your applications. Now you are in trouble to decide if you want to go for the Isolation level SERIALIZABLE to avoid the unwanted phenomena and loose some degree of concurrency or if you can accept that some phenomena can occur.

Dirty Read describes a situation when a transaction T1 changes a row and T2 reads the row before T1 ends with COMMIT. If T1 performs a ROLLBACK in this case, then T2 has read a row that never existed.

Transaction T1	Transaction T2
INSERT INTO table1 (column1) VALUES (3)	
	SELECT * FROM table1
ROLLBACK, removes the newly inserted row from table1	continues to use a record that never existed

A Non Repeatable Read can happen in the following situation. Transaction T1 reads a row. Then transaction T2 modifies that row and ends with COMMIT. If T1 then reads the row again, it either gets the modified row or a message indicating that the row no longer exists.

Transaction T1	Transaction T2
SELECT * FROM table1 WHERE column1 = 3	
	DELETE FROM table1 WHERE column1 = 3
	COMMIT
Row no longer exists: UPDATE table1 SET column2 = 'a' WHERE column1 = 3	

A Phantom is a new record that should not be visible to a transaction. Transaction T1 executes a query and retrieves a result set of n rows. Then Transaction T2 creates at least one new record that meets the search conditions used to buld the result set retrieved by T1. If T1 executes the search again, the new record created in the transaction T2 becomes visible to T1. This breaks the Isolation rule of ACID-transactions.

Transaction T1	Transaction T2
SELECT column2 FROM table1 WHERE column1 = 3, returns: 1, 2, 3	
	INSERT INTO table1(column2, column1) VALUES (4, 3)
	COMMIT
SELECT column2 FROM table1 WHERE column1 = 3, returns: 1, 2, 3 plus the Phantom 4	

Again, more informations on the Isolation Level and the Locks that are implicitly set can be found in the manual. Use the Glossary to navigate to the related manual pages. Every database user should know the basic properties of transactions, isolation levels and locks. No matter if you are a database administrator or a database software developer. Once you are familiar with the properties from a users standpoint, you should continue to dive into the subject and try to understand how you database system has implemented transactions internally.

## Implementation details

Databases are writing to types of log entries for transaction. One log is called redo-log and the other one is called undo-log. When a database user starts a transaction and executes some commands, the database does not know if the user will end the transaction with commit or rollback. If he ends it with commit, then all the changes made in the course of the transaction have to be made durable. If the transaction gets interrupted for some reason, be it that the client disconnects before sending a commit, the system crashes or the user sends an explicit rollback, then all changes made by the transaction need to be undone.

This makes clear why a database needs undo-log records. Undo-log records are also referred to as before-images, because they store the state of the database before a transaction has changed it. For example, if a transaction changes the value of a column from 3 to 4, then the database writes an undo-record to remember the overwritten value of the modified column. If the transaction gets rolled back, the database looks up the before-image of the modified column and sets the column value back from 4 to its original value of 3. Undo informations are stored by MaxDB on the data volumes.

Storing the undo informations on the data volumes is beneficial, if you are loosing all log volumes due to a hardware outage. In that case, MaxDB can undo all unfinished transaction and restore a transaction consistent state on the data volumes without the log volumes. If you are sure, you understood the meaning of undo records, ask yourself how it *\*theoretically\** can happen that during the execution of a DELETE or UPDATE operation the usage level on the data volumes can temporarily increase instead of being reduced. The answer is that in the worst case the Garbage Collector (Database parameter `_MAXGARBAGE_COL`) cannot remove undo-log records as fast as you are implicitly creating new ones. As a result the usage level on the data volumes increases and in the worst case, the volumes will run full. However, I've never seen this in real-life applications!

Undo-log records are important in case of a rollback, but what do you need a redo log for? The D in ACID stands for Durability. Durability means that after the successful execution of COMMIT all modifications that are made by the transaction must be stored in a persistent, durable medium. MaxDB could write the modified records to places where they are stored. But this would result in a lot of random write operations spreaded over the data volumes. Random write operations are slower than simple sequential write operations on a log file.

In the times of 64bit computers and affordable prices for RAM, most database servers will use huge I/O buffer with hit rates above 98%. That means that most read and write operations will be done in main memory. And it is likely that a modification that is made in the cache will soon be overwritten again. Therefore it is great if you do not need to flush all modified pages from the buffer to the disk in case of a COMMIT, but if you simply can write a sequential redo log. Then, from time to time, you write the modified pages to the data volumes to get a transaction consistent state on the data volumes.

Writing flushing all modified pages from the buffer cache to disk is called a savepoint in MaxDB. Savepoints are written in regular intervals (see also Database parameter `_RESTART_TIME`). If the database server crashes in the middle of two savepoints, then it will replay all redo log entries to the crash date to recover the state of the database when the crash occurred.

For you all this means, that you should use the fastest disks you can afford for the log volume and you should always mirror the log volumes with the redo log entries. The disks must be fast, because this influences how long a transaction has to wait for a COMMIT to be completed. Mirroring should be used to reduce the risk of a hardware failure. Remember that if you loose the log volumes due to a crash, you loose all the changes made since the last savepoint. By default this can be upto some 10 minutes of work.

To further speed up operations, MaxDB does not write every redo record immediately to disk. Instead all records are first written into a main memory log queue. There can be upto `MAX_LOG_QUEUE_COUNT` log queues in MaxDB 7.6. The actual number is calculated automatically or can be defined using the database parameter `LOG_QUEUE_COUNT`. Every log queue has a size of `LOG_IO_QUEUE` pages (8kb page size). Log queue entries are written two disk by dedicated log queue writer tasks. If possible, MaxDB tries to bundle write operations and redo records into larger blocks. Writing larger blocks is usually much faster than writing individual records and you have the chance that you can write the records of several transactions to disk with only one physical I/O operation.

Therefore, every MaxDB database administrator should not only take extra care of the log volumes, put them on the fastest disks that he can afford but he should also have an eye on monitoring informations and the values of LOG QUEUE OVERFLOWS and LOG QUEUE MAX USED ENTRIES. The monitoring can be done using the Database Analyzer, the Information screen of the Database Manager GUI or by checking the assorted LOG\* system tables, for example the table LOGQUESTATISTICS. This gives you another interesting information in the column WAITCOUNT which describes how may transactions are currently waiting for a COMMIT/ROLLBACK to be completed.

## In the next issue

In the next issue of the MaxDB series we will have a short look at some locking details. Stay tuned and happy hacking!

*Ulf Wendel for the MySQL MaxDB team*