



Information for BI Consultants

BI with MaxDB

Date: 14.10. 2008

Version: MaxDB Version 7.6.01 – 7.6.05

Author: Christiane Hienger (SAP AG)
Torsten Pfeiffer (SAP AG)

Copyright

©Copyright 2007 SAP AG
All rights reserved

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned and associated logos displayed are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Einige von der SAP AG und deren Vertriebspartnern vertriebene Softwareprodukte können Softwarekomponenten umfassen, die Eigentum anderer Softwarehersteller sind.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge und andere in diesem Dokument erwähnte SAP-Produkte und Services sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und in mehreren anderen Ländern weltweit. Alle anderen in diesem Dokument erwähnten Namen von Produkten und Services sowie die damit verbundenen Firmenlogos sind Marken der jeweiligen Unternehmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.

Die in diesem Dokument enthaltenen Informationen sind Eigentum von SAP. Dieses Dokument ist eine Vorabversion und unterliegt nicht Ihrer Lizenzvereinbarung oder einer anderen Vereinbarung mit SAP. Dieses Dokument enthält nur vorgesehene Strategien, Entwicklungen und Funktionen des SAP®-Produkts und ist für SAP nicht bindend, einen bestimmten Geschäftsweg, eine Produktstrategie bzw. -entwicklung einzuschlagen. SAP übernimmt keine Verantwortung für Fehler oder Auslassungen in diesen Materialien. SAP garantiert nicht die Richtigkeit oder Vollständigkeit der Informationen, Texte, Grafiken, Links oder anderer in diesen Materialien enthaltenen Elemente. Diese Publikation wird ohne jegliche Gewähr, weder ausdrücklich noch stillschweigend, bereitgestellt. Dies gilt u. a., aber nicht ausschließlich, hinsichtlich der Gewährleistung der Marktgängigkeit und der Eignung für einen bestimmten Zweck sowie für die Gewährleistung der Nichtverletzung geltenden Rechts.

SAP übernimmt keine Haftung für Schäden jeglicher Art, einschließlich und ohne Einschränkung für direkte, spezielle, indirekte oder Folgeschäden im Zusammenhang mit der Verwendung dieser Unterlagen. Diese Einschränkung gilt nicht bei Vorsatz oder grober Fahrlässigkeit.

Die gesetzliche Haftung bei Personenschäden oder die Produkthaftung bleibt unberührt. Die Informationen, auf die Sie möglicherweise über die in diesem Material enthaltenen Hotlinks zugreifen, unterliegen nicht dem Einfluss von SAP, und SAP unterstützt nicht die Nutzung von Internetseiten Dritter durch Sie und gibt keinerlei Gewährleistungen oder Zusagen über Internetseiten Dritter ab.

Alle Rechte vorbehalten."

Table of Contents

1	PURPOSE OF THIS DOCUMENT	5
2	MAXDB BW FEATURE PACK	5
2.1	DATA STORAGE WITHOUT BW FEATURE PACK	5
2.2	DATA STORAGE WITH MAXDB BW FEATURE PACKS	6
2.2.1	<i>Technical Information about Virtual Key</i>	6
2.2.2	<i>Efficient Data Storage (Table Attribute Cluster)</i>	7
2.2.3	<i>Space-Saving Data Storage (Table Attribute PACKED)</i>	7
2.3	JOIN PROCESSING	8
2.3.1	<i>Hash Join</i>	8
2.3.2	<i>Star Join</i>	8
3	SELECTION OF THE DEDICATED DATA MODEL	9
3.1	KEY FIGURE MODEL:.....	9
3.2	ACCOUNT MODEL:	9
3.3	KEY FIGURE MODEL OR ACCOUNT MODEL?.....	10
4	DATA MODELLING	11
4.1	COMMON RECOMMENDATIONS	11
4.2	MULTI PROVIDER	12
5	CUBE ADMINISTRATION.....	12
5.1	DATA LOAD	12
5.2	BI COMPRESSION	13
6	QUERY-ANALYSIS AND PERFORMANCE TUNING.....	13
6.1	HOW TO USE THE TABLE ATTRIBUTE CLUSTER	13
6.2	MASTER DATA TABLES	15
7	INFORMATION MATERIAL	15
7.1	MAXDB BW FEATURE PACK.....	15
7.2	MAXDB GENERAL.....	16
7.3	MAXDB MAILING-LIST	16

1 Purpose of this Document

This document describes the functionality of the MaxDB BW Feature Pack and its consequences for the operation of SAP BI systems with MaxDB.

Here you can find recommendations for BI consultants to consider when choosing the data model, during data modeling, and cube administration. There are also hints about query optimization and performance analysis.

This is the first version of this document and will be updated if necessary.

2 MaxDB BW Feature Pack

Since MaxDB Version 7.6.01 the MaxDB Feature Pack is available. A new installation of BI with MaxDB 7.6.01 implicitly activates the functionality of the BW Feature Pack.

If customers already use BI with MaxDB version lower than 7.6.01 they must upgrade to a MaxDB version greater or equal than 7.6.01.

Also see note:

[983845: Using MaxDB 7.6 for BW 3.x / BI 7.x](#) .

We recommend executing a SAVE DATA followed by a RESTORE DATA before converting the fact tables, especially if the target MaxDB Version is lower than 7.6.04.

2.1 Data Storage without BW Feature Pack

To make it easier to understand the specifics of the BW Feature Pack, here is a short introduction to the default data storage of MaxDB, if used without the functionality of the BW Feature Pack.

MaxDB stores data on the data volumes as 8 KB pages. MaxDB does not have table spaces. All space management is fully automated and independent from schema and data model. You do not have to assign storage locations to schema objects.

All data is processed in the data cache. Every 10 minutes a savepoint ensures that modified data is flushed to disk. The MaxDB data pages are distributed equally on all volumes (striped). A MaxDB primary key is clustered with the data and thus defines the sorting sequence.

BI systems running on a MaxDB version lower than 7.6.01 use this default data storage model.

2.2 Data Storage with MaxDB BW Feature Packs

With the MaxDB BW Feature Pack, the internal data storage model is changed. Interrelated data is now stored continuously on disk, sorted by a time characteristic. This feature accelerates the I/O accesses.

This can be reached:

- By defining a virtual key on the fact table. This virtual key ensures that rows with the same time characteristic are stored on the same disk area.
- With an efficient data storage on the disk data area with the table attribute CLUSTER
- By using the table attribute PACKED to reduce space on storage

2.2.1 Technical Information about Virtual Key

The virtual key feature lets you configure a key on non-UNIQUE columns. A virtual key consists of a non-UNIQUE key column (so called cluster key) in combination with a unique serial number (SEQUENCE). If the table has a virtual key it is called a logical clustered table. This feature is comparable to a clustered index used with other DBMSs.

Virtual keys can only be created on fact tables, which have the table attribute CLUSTER. The cluster key is created from a time characteristic. This ensures that records with the same time characteristic are stored in the same storage area.

PSA tables with the table attribute CLUSTER (as of BI 7.0. Support-Package 16) do not need a virtual key. PSA tables have a primary key, which is also the sorting criteria.

2.2.2 Efficient Data Storage (Table Attribute Cluster)

Fact tables and, as of BI Release 7.0, additional PSA tables are created with the table attribute CLUSTER. With this CLUSTER attribute data the 8KB pages are no longer stored equally across all data volumes. Instead, if possible the data is stored in larger continuous blocks of 8 KB pages on the data volumes. In an extreme case, a whole table might be stored completely and ordered on one single volume.

Benefit:

- The data can be continuously read again during read access.
- Larger segments of a table are read with one I/O access due to the read ahead facilities of today's storage systems. The result is significantly better I/O access time – for example, during sequential reads as large sectors are read and subsequent reads already hit the data in the cache.

If a table has the table attribute CLUSTER, all indexes of this table also have the attribute CLUSTER.

Comment:

To preserve the cluster of a table during the backup or recovery process you

have to make sure that the block size in the media definition is equal the value of parameter DATA_IO_BLOCK_COUNT or a multiple of DATA_IO_BLOCK_COUNT.

2.2.3 Space-Saving Data Storage (Table Attribute PACKED)

Fact tables have the table attribute PACKED. The integer values are compressed.

Benefit:

- Data needs less disk space
- This means fewer I/O accesses and fewer cache swaps

2.3 Join Processing

With the BW Feature Pack MaxDB supports Hash- and Star-Join in a wider range.

2.3.1 Hash Join

Apart from the Nested-Loop-Join, the MaxDB BW Feature Pack also supports the Hash-Join. The memory to be used for Hash-Joins is configured with the MaxDB parameters MAX_HASHTABLE_MEMORY and MAX_SINGLE_HASHTABLE_SIZE.

Our experience in BI systems shows that after the installation the default settings of both parameters are often too small. You can find recommendations for the initial configuration of both parameters in note:

[814704: MaxDB Version 7.6 parameter settings for OLTP/BW](#)

You can achieve optimal behavior of complex queries if all tables – except the fact table – are processed with the Hash-Join algorithm. You can check this with the EXPLAIN plan. If large tables are involved in your query, you might have to further increase the initial values of the parameters MAX_HASHTABLE_MEMORY and MAX_SINGLE_HASHTABLE_SIZE.

2.3.2 Star Join

MaxDB Star-Join is based on the SAP BI-Data model. The optimization is done on the dimension level and not on the table level. This process requires a labelling of the fact tables in the database catalog. Use note

[976930: FACT attribute missing for fact tables on database](#)

and execute the report RSDU_SET_FACT_ATTR_ADA. This report takes only a few seconds and ensures that the fact tables are labelled as fact tables in the database catalog. You can use this report at any time to label the fact tables.

3 Selection of the Dedicated Data Model

Consider the following information when you select the data model in a MaxDB system environment.

3.1 Key Figure Model:

You want to assign several key ratios (sales price, production price, price) with identical units (currency *EUR*) to a characteristic (Product *P1*).

The key figure model includes all key figures in one data record of the fact table.

Product	Sales Price	Production Price	Price
P1	100	50	75

- Doing this, the key figure model creates wider cubes on database level than the account model - and MaxDB implicitly compresses unused key figures (zero values => 4 bit), you can require less additional space.
- It is an advantage to have coherent key figures in one record of the fact table

3.2 Account Model:

On the other hand in an account model you add a new characteristic, which describes a key figure.

Product	Price type	Price
P1	Sales Price	100
P1	Production Price	50
P1	Price	75

On the database level an account model means that:

- The key figure type can be described by a new characteristic in an additional dimension table.
- For each record on transaction level there are several records created in the fact table(s).

3.3 Key Figure Model or Account Model?

Using an account model leads to significantly more records in the fact table compared to the key figure model.

Since each dimension has a secondary index, the indexes in an account model can get very large. Note that for a fact table the total of the sizes of its indexes can be larger than the size of the fact table itself. This means that index maintenance is costly.

MaxDB does utilize bit-mapped indexes. If a query has predicates on multiple dimensions the MaxDB optimizer can only evaluate this criteria before accessing the fact table, if similar multiple indexes exist on the fact table. This is generally not the case. In other words, such multiple indexes have to be created explicitly during the performance analysis and tuning phase.

Recommendation:

To help you decide which data model should be used with MaxDB, the recommendations in the SAP documentation are basically valid. But also note the MaxDB distinctions listed above in borderline cases. See the following example:

Example:

You have designed 100 key figures, but the customer is using only 10 key figures. The residual 90 figures are filled with zero values.

Key figure model:

With MaxDB in the key figure model those unused key figures use only 45 bytes.

Account model:

If the key figures with value zero have been eliminated during transformation, the fact table will have 9 additional records for each characteristics combination. If it is necessary to maintain the zero values, the fact table has 99 additional records for each characteristic combination.

When using MaxDB for this case, choose the key figure model.

4 Data Modelling

4.1 Common Recommendations

By default all data of the fact table will be stored sorted by time, along to a correspondingly defined cluster key.

For the following cases a cluster key on a non time characteristic makes sense:

- For very large aggregates with no time dimension defined
- If the time characteristic is not used in the reporting

Use the following SAP note to change the cluster key:

[1118095: Clusterung der Faktentabelle](#)

Caution:

If you want to use a cluster key other than the default cluster key, the affected cube can only be activated after the cluster key has been modified.

Time characteristics must not be designed as key figures. Those key figure columns in the fact table would be defined as type CHAR. Columns of type CHAR cannot be packed.

Important:

We generally recommend designing small dimensions for BI. Dimension tables should have as few entries as possible. High cardinality characteristics should be

stored in different dimensions. In contrary, MaxDB profits from queries that contain selection criteria from only a few different dimensions. So if multiple characteristics that represent the most typical selection(s) can be bound to one dimension, this is still advantageous even if it results in a bigger dimension.

However, if strongly qualifying characteristics are stored in different dimensions, an optimization can be reached by a multiple index on the corresponding dimension columns on the fact table. This task should only be used in exceptional cases, because additional indexes lead to increased resource consumption.

4.2 Multi Provider

You should also consider the usage of multi provider, e.g. annual cubes.

From the database viewpoint this has two benefits:

- Smaller tables and indexes. This has a positive effect on the loading of data.
- The definition of multi provider enables concurrent execution of multi-provider queries across several work processes.

5 Cube Administration

5.1 Data Load

For data load with MaxDB version 7.6.01 – 7.6.03, take care that records with identical or consecutive time characteristics should be loaded coherently. This lets you effectively cluster the fact table.

As of MaxDB Version 7.6.04 onwards this can be disregarded as the database kernel automatically performs optimal clustering.

To avoid data cache swaps during data load, make sure that the size of the data packages is not too large. Unfortunately it is not possible to provide a formula for calculation of the package size. Instead, the size of the package and therefore the number of processed records depends on the width of a cube in terms of its columns and on the size of the data cache (MaxDB parameter CACHE_SIZE).

5.2 BI Compression

The BI compression aims to reduce the number of records in the fact table of an info cube. If using MaxDB such compression is only worthwhile if the number of records can be reduced significantly.

If the customer decides to use BI compression, the compression should be applied at regular intervals in future.

You can find a lot of SAP notes about compression especially with BI 7.0.

6 Query-Analysis and Performance Tuning

6.1 How to Use the Table Attribute CLUSTER

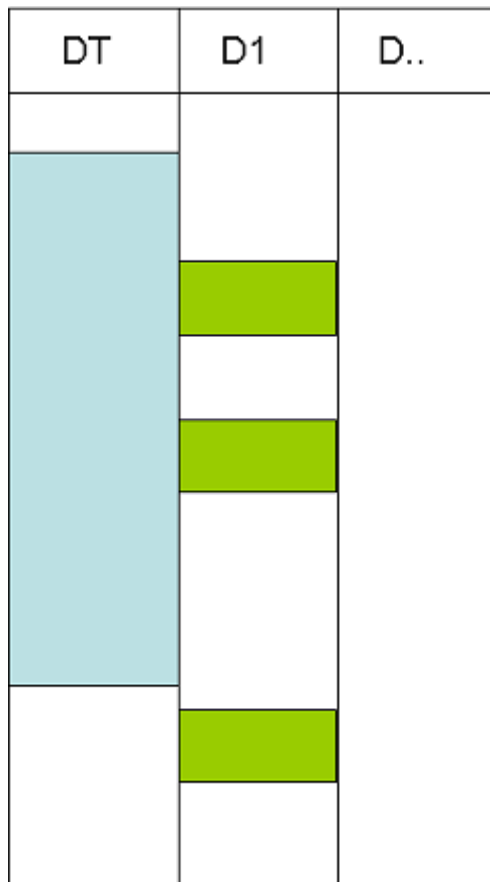
Having the table attribute CLUSTER, all data of the fact table is stored in large coherent blocks, which are sorted by time criteria. To achieve good performance you must have qualifications or predicates based on time criteria to assure that the database can read a coherent area of the fact table also coherently from disk. Due to the cluster feature the database needs only a few I/O accesses for a large amount of data.

If you qualify records with non-time characteristics on the fact table, this cluster feature cannot be used - that is, the data is read in 8 KB packages.

If large sections of data are already stored in the data cache, clustering has less or even no benefit.

In MaxDB Versions 7.6.01 – 7.6.03 the cost calculation of the optimizer is not I/O optimized and does not take account of clustering. The database might not select the data of the fact table by using the time characteristic, but instead uses another dimension.

The following picture shows this:



The blue area (DT) shows a coherent area of the fact table on the disk sorted by time criteria. The green areas (D1) shows 8 KB pages with records that are selected by a qualification on Dimension 1 (D1).

If the fact table is accessed by using dimension 1 the system must read less data, but for each page one I/O is necessary. The access using time dimension (DT) on the fact table obviously has to read more data. But as this data is stored coherently on the disk, all data can be read with less I/O.

With MaxDB versions 7.6.01 – 7.6.03 this case can be optimized by creating a multiple index on the fact table using columns DT and D1. This should be done only in exceptional cases, because additional indexes lead to an increase in resource consumption.

As of MaxDB version 7.6.04 the administrator has the option to set a table attribute that triggers the optimizer to select access via the clustered key (for example, DT).

This new table attribute should be set for large cubes.

You can find more information about this in note:

[1118094: Parametrisierung der Kostenfunktion für BI-Queries](#)

6.2 Master Data Tables

Master data tables (namespace /X., /Y..., /S...) are sometimes rather small tables but do not necessarily have indexes on all those attributes used for qualification. MaxDB uses statistical information on those attributes to find the best access path only if indexes for those attributes exist. It is important to have indexes for MaxDB

Recommendation:

Create an index for attributes even on small master data tables, if those attributes are used for qualification.

7 Information Material

7.1 MaxDB BW Feature Pack

You can find information about I/O Optimization in SDN here:

[Kernel Performance - Accelerating IO](#)

You can find information about join optimization in SDN here:

[Kernel Performance - Join Processing](#)

In SDN MaxDB Wiki an area about MaxDB and BI systems is under construction

<https://wiki.sdn.sap.com/wiki/x/W0o>

Questions and Answers to BW Feature Packs are here:

[1040431: MaxDB BW Feature Pack](#)

7.2 MaxDB general

You can find information about MaxDB in SDN here:

<https://www.sdn.sap.com/irj/sdn/maxdb>

7.3 MaxDB Mailing-List

You can put your questions about MaxDB and BI to the MaxDB forum here:

[MaxDB Forum](#)