

Performance Tuning in SAP BI 7.0



Applies to:

SAP Net Weaver BW. For more information, visit the [EDW homepage](#).

Summary

Detailed description of performance tuning at the back end level and front end level with example

Author: Adlin Sundararaj

Company: Accenture Services Pvt. Ltd.

Created on: 15 April 2011

Author Bio



Adlin Sundararaj is working as SAP-BI Consultant in Accenture Services Private Ltd and has experience of working on implementation and maintenance projects.

Table of Contents

Introduction	3
Factors Affecting the Performance at the Backend	3
Data Load Performance.....	4
Upload Sequence.....	4
PSA Partition Size.....	4
Parallelizing Upload	4
Export DataSource.....	5
Transformation Rules.....	5
Factors Affecting the Performance at the Frontend	6
Query Performance.....	6
Aggregates.....	6
Step by step procedure	7
Step 1:	7
Step 2:	7
Step 3:	8
Step 4:	8
Step 5:	9
Step 6:	9
OLAP Cache	10
Step to change OLAP cache settings:-	10
Read Mode.....	11
Step to set the read mode:.....	11
Related Content.....	12
Disclaimer and Liability Notice.....	13

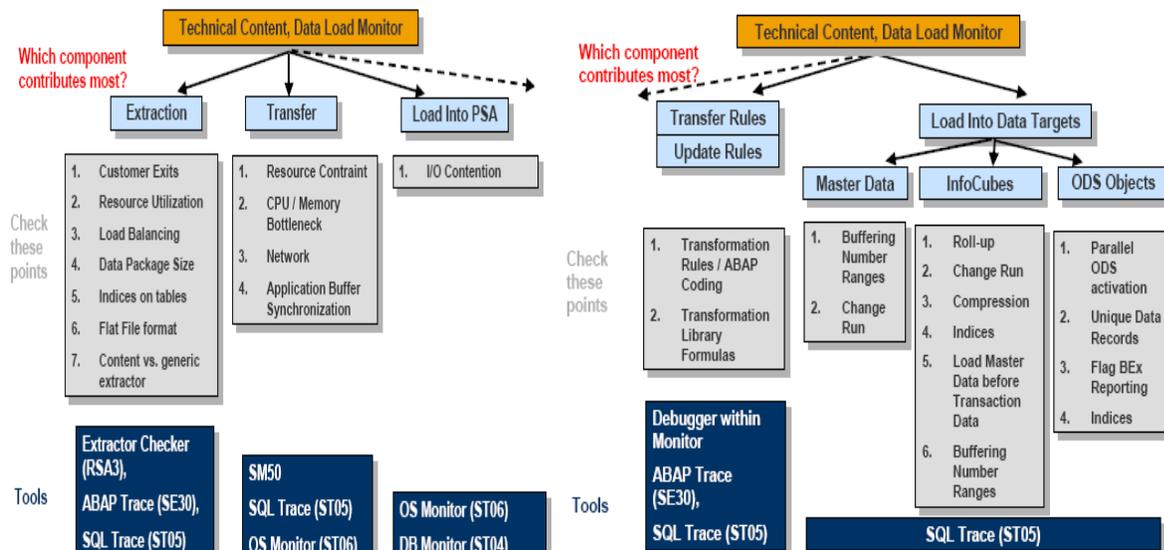
Introduction

The performance aspects of a data warehouse project should be of primary consideration, as performance is directly related to the degree of user acceptance of the solution, as well as the overall effectiveness of its analytical capabilities.

- Every project phase should be reviewed subject to performance: beginning with the data model, through the fine-tuning of the database, and ending with proactive operational maintenance mode (e.g. review aggregates, build new ones, delete unused ones, and find problem queries) as an ongoing cycle.
- Performance tuning in BW can be done mainly in three different stages: modeling, extraction and reporting.

Factors Affecting the Performance at the Backend

The following picture shows how to approach performance problems at the Back end level.



- The top item shows the central point of entry for detecting performance issues. The area which contributes most to the long runtime can be identified here; this is shown in the second level.
- Check all items in the boxes below the problem area.
- The bottom line tells you which tools are available to find the problem.

Data Load Performance

This section describes the factors which affect the performance during data load.

Upload Sequence

Issue

- The sequence of the load processes is usually defined within process chains or event chains. This sequence can have a significant impact on the load performance.
- The master data load creates all SIDs and populates the master data tables (attributes and/or texts). If the SIDs do not exist when transaction data is loaded, these tables have to be populated during the transaction data load, which slows down the overall process.

Solution

- Always load master data before transaction data. The transaction data load will be improved, as all master data SIDs are created prior to the transaction data load, thus precluding the system from creating the SIDs at the time of load.
- If you want to replace existing data in the DataTarget completely, first delete the data (in PSA and/or DataTarget) and load afterwards. Small (or empty) PSA tables improve PSA read times. Small (or empty) InfoCubes improve deletion, compression time and aggregation time (thus affecting data availability).

PSA Partition Size

Issue

- PSA tables are partitioned automatically. In transaction RSCUSTV6 the size of each PSA partition can be defined. This size defines the number of records that must be exceeded to create a new PSA partition. One request is contained in one partition, even if its size exceeds the user-defined PSA size; several packages can be stored within one partition.
- The PSA is partitioned to enable fast deletion (DDL statement DROP PARTITION). Packages are not deleted physically until all packages in the same partition can be deleted.

Solution

- Set the PSA partition size according to the expected package sizes. If you expect many small packages, set the partition size to a rather small value, so that these partitions can be deleted quickly.

Parallelizing Upload

Issue

- Parallel processing is automatically initiated during an extraction from an SAP system; the settings for the data package size directly influence the number of data packages that are likely to be sent in parallel. Moreover, a setting in the InfoPackage defines the degree of parallelism of updating to PSA and Data Targets. Thus, data load performance is fully scalable.
- You can start several processes manually in parallel: InfoPackages, Process Chains.

Solution

Use parallelism for uploading if possible (i.e. no resource restrictions):

- Flat files: Split files for multiple InfoPackages, enable parallel PSA → DataTarget load process.
- SAP source system: Create several InfoPackages for the same or different Data Sources and then schedule them in parallel.
- Data load to several Data Targets: Use different InfoPackages (at least one for each DataTarget).

If the data load takes place when there is little other activity in the BW system, then optimal results are likely to be observed if the number of (roughly equally-sized) InfoPackages is equivalent to the number of CPUs in the application servers in the logon group handling the data load.

Export DataSource

Issue

- The Export DataSource (or Data Mart interface) enables the data population of InfoCubes and DSO Objects out of other InfoCubes.
- The read operations of the export DataSource are single threaded (i.e. sequential). Note that during the read operations – dependent on the complexity of the source InfoCube – the initial time before data is retrieved (i.e. parsing, reading, sorting) can be significant.
- The posting to a subsequent DataTarget can be parallelized by ROIDOCPRMS settings for the “myself” system. But note that several Data Targets cannot be populated in parallel; there is only parallelism within one DataTarget.

Solution

- Use InfoPackages with disjoint selection criteria to parallelize the data export.
- Complex database selections can be split to several less complex requests. This should help for Export Data Sources reading many tables.
- If the population of InfoCubes out of other InfoCubes via Export DataSource takes up too much time, try to load the InfoCubes from PSA.

Transformation Rules

Issue

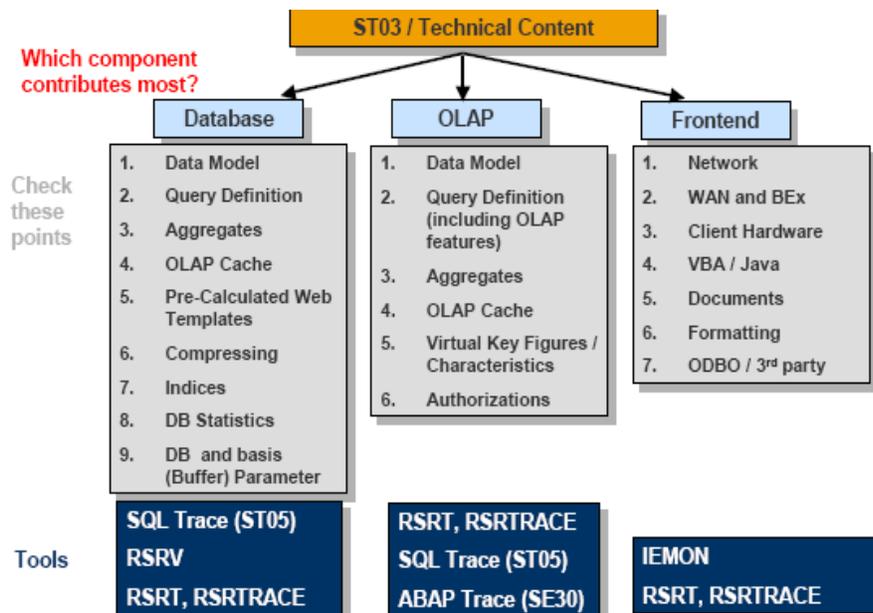
- Transformation rules are transfer rules and update rules. Start routines enable you to manipulate whole data packages (database array operations) instead of changing record-by-record.
- Standard functionalities are one-to-one-mapping, reading master data, using transformation rules (in BW 3.x) and providing own ABAP coding.

Solution

- In general it is preferable to apply transformations as early as possible in order to reuse the data for several targets.
- Better use transfer rules (ONE transformation) if you have to transform data for example to several Data Targets (the same transformation for EACH DataTarget). Technical transformations could even be done outside BW (e.g., in the ETL tool).
- If you discover a lot of time spent in transformation rules (e.g., via Data Load Monitor), check and improve the coding in the transformation.

Factors Affecting the Performance at the Frontend

The following picture shows how to approach performance problems at the Back end level.



- The top item shows the central point of entry for detecting performance issues. The area which contributes most to the long runtime can be identified here; this is shown in the second level.
- Check all items in the boxes below the problem area.
- The bottom line tells you which tools are available to find the problem.

Query Performance

The Query performance can be improved by the below three options.

Aggregates

- Aggregates are mini-cubes built on top of InfoCubes. An aggregate is a highly summarized version of the data in the underlying InfoCube.
- During query execution, if applicable, an aggregate is hit before the InfoCube, and this serves to improve performance.
- Users of aggregates (especially first-timers), tend to think of it as a silver bullet for the **query is taking forever to run** problem. However, the sole rationale for using aggregates shouldn't be that a query is taking too long to run.
- Unless the planning and design process is properly followed, aggregates may degrade performance rather than improve it.
- An example case of this is frequent change to master data attributes. Keeping these updates to your InfoCube synchronized with the corresponding aggregate(s) involves frequent change runs, which take longer and longer as the number of aggregates

increases. The interface to build the summary tables (aggregates) is intuitive and easy to master, but few are taking real advantage of them.

Step by step procedure

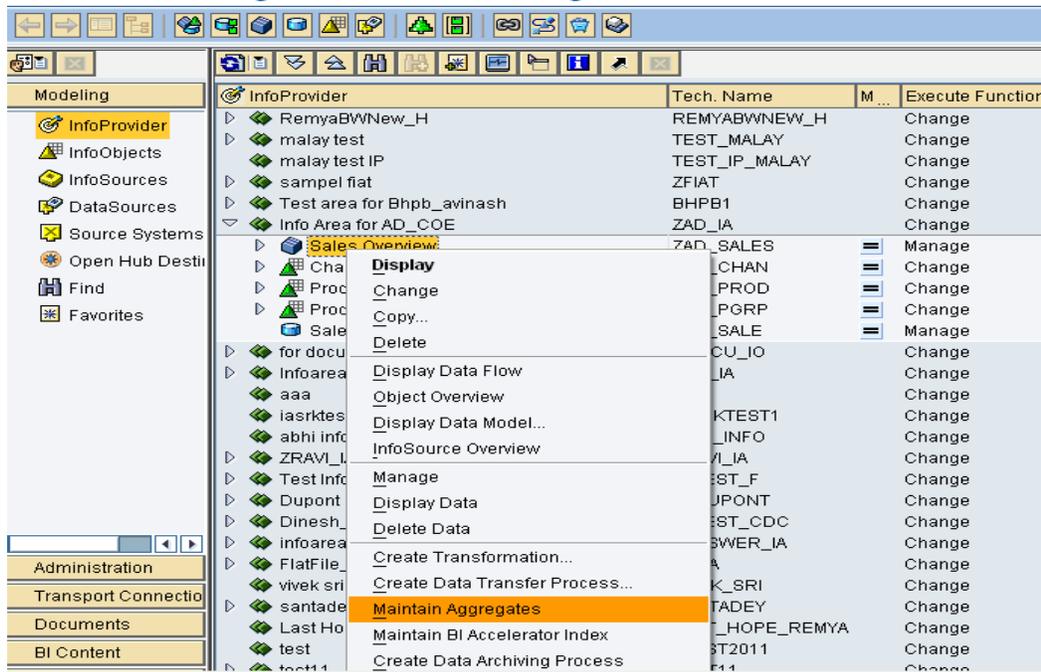
Step 1:

Drill down to the InfoCube on which you want to create an aggregate.

Highlight this InfoCube;

Then from the context menu (right mouse click), select → Maintain Aggregates.

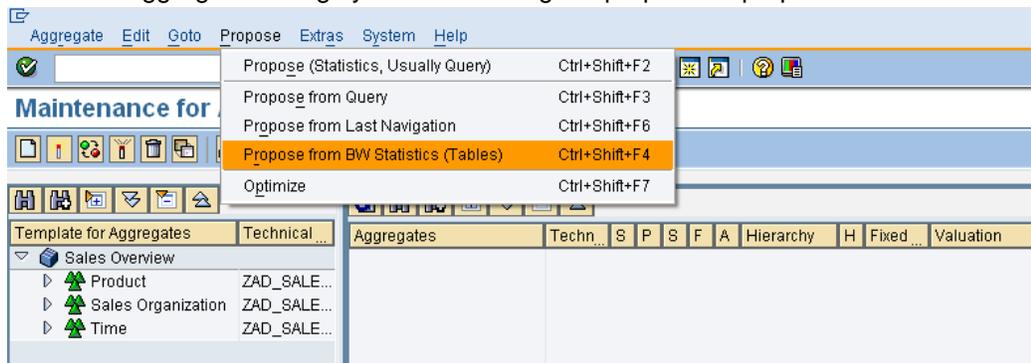
Data Warehousing Workbench: Modeling



Step 2:

Displays the menu path that takes you to aggregates creation

To create aggregates using system statistics go to propose → propose from BW statistics.

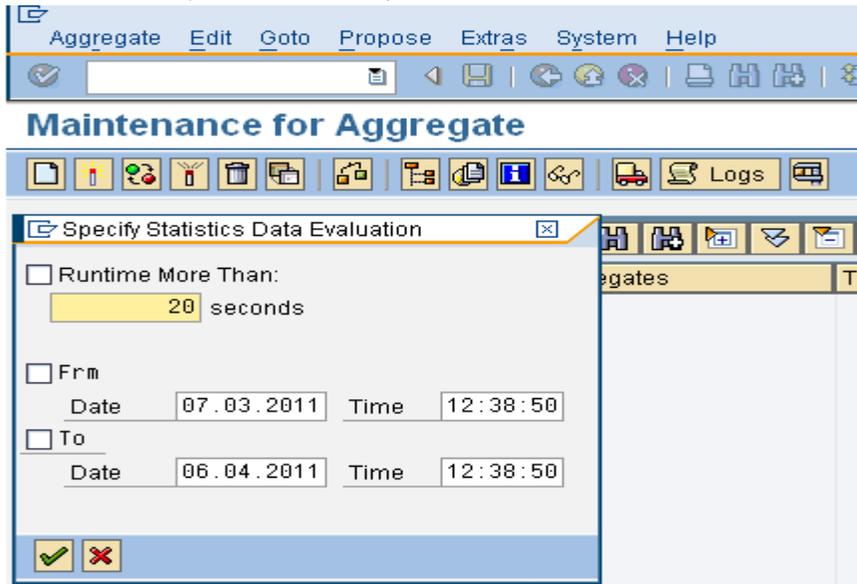


Step 3:

A pop up window will open to specify Statistics data evaluation

Select the run time of queries to be analyzed (e.g. 20sec)

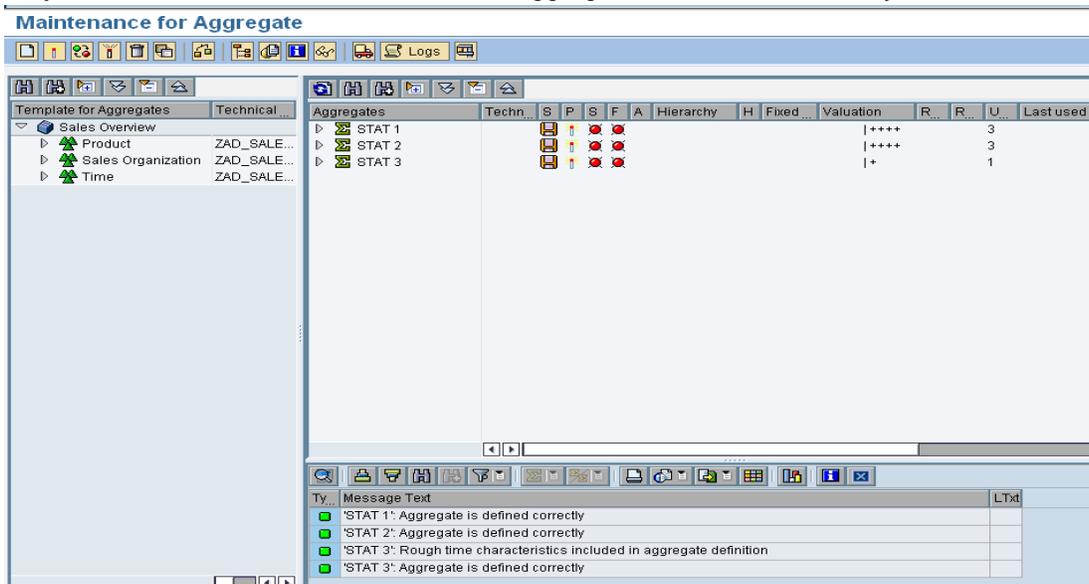
Select the time period to be analyzed



Step 4:

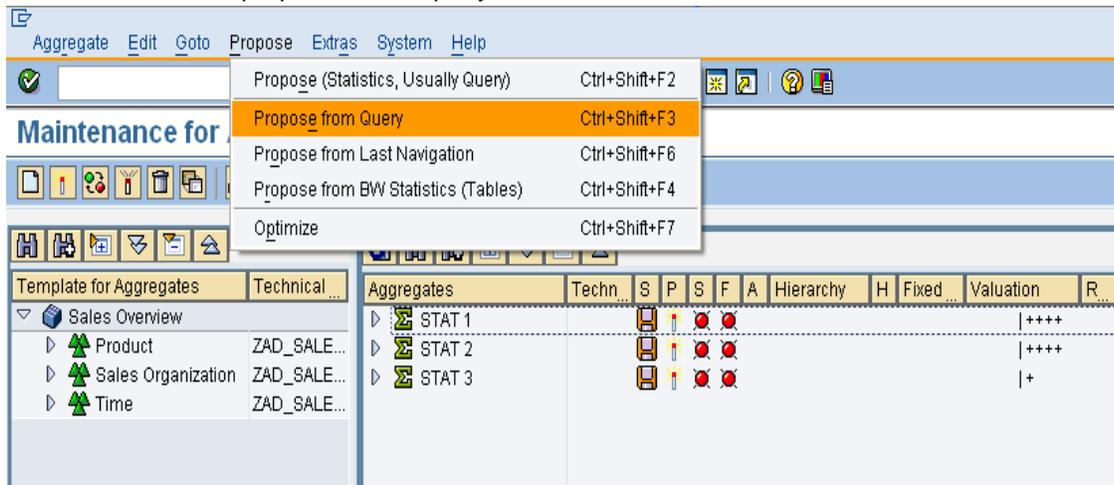
In our example, 3 aggregates are created as shown below.

As you can see in the below status bar, 'the aggregates are defined correctly'



Step 5:

We can also create proposals from query as shown below.



We can create proposals for aggregates on individual queries, select the query on which you want to create proposal.

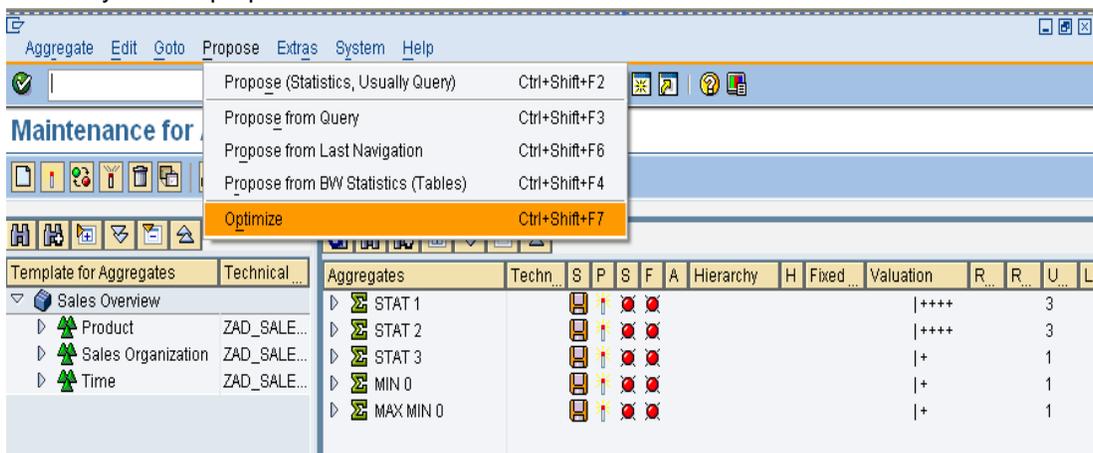
In our example, I have selected ZAD_SALES_2007 query.



Step 6:

Reduce the overlapping proposals by optimizing them as shown below.

This may reduce proposals from 99 to less than a dozen.



After optimization the result is like

Maintenance for Aggregate

Aggregates	Techn	S	P	S	F	A	Hierarchy	H	Fixed	Valuation	R	R	U	Last used
STAT 1										++++			3	
STAT 3										+			1	

We want high valuation and high usage. This indicates high reduction of records in aggregates and high benefits to users.

And thus our aggregates are maintained for InfoCube.

Note:

An InfoCube can contain more than one aggregate, but usually only one aggregate can be used in each query step. If several aggregates are created for an InfoCube, they are checked by the OLAP processor when a query is run. During this check, the OLAP processor reads the InfoCube aggregates and selects the aggregate that is suitable for the query definition.

OLAP Cache

- The OLAP cache is architected to store query result sets as highly compressed cluster data and to give all users on an application server access to those result sets.
- If the same query (or a subset) is then executed by another user, the subsequent query request can be filled by accessing the result set already stored in the OLAP cache.
- OLAP Cache has a system default of 100 MB for local and 200 MB for global cache. This may be too low for a system that can be optimized via broadcaster.

Step to change OLAP cache settings:-

Go to transaction code RSCUSTV14 and edit the settings

Change View "OLAP: Cache Parameters": Details

Customizing ID: Business Information Warehouse

OLAP: Cache Parameters

Cache Inactive

Local Size MB: 100

Global Size MB: 200

Persistence Mode: Flat File

Flat File Name: BW_OLAP_CACHE

ComprehensiveFile: BW_OLAP_CACHE_SPAN

- Persistence mode – is alternative cache storage medium when cache memory is exhausted.
- Flat File Name – application server specific logical file name
- Comprehensive File – cross application server specific logical file name

Read Mode

Read mode determines the manner in which data are returned by the OLAP processor when you execute a query. SAP provides three read options. These are:

(H) – Query to read, when you read or expand hierarchies: In H mode, the detailed information within the nodes and sub-nodes of a hierarchy are read on demand (i.e., when you expand a hierarchy node). The amount of data transferred to the OLAP processor is the least of all the three categories. The downside is that the number of database reads increases as you keep expanding hierarchy nodes. SAP recommends setting queries to the H read mode. This is based on the empirical evidence of the relative performance of queries set to each of these modes.

(X) – Query to read data during navigation: In this mode, data are requested by the OLAP processor as dictated by every navigational step. The difference between X mode and H is that in the case of hierarchies, all the data on a leaf level are completely retrieved by the OLAP processor.

(A) – Query to read all data at once: There is only one read process in this mode. When you execute the query in the Business Explorer, all data in the main memory area of the OLAP processor that is needed for all possible navigational steps of this query is read. During navigation, all new navigational states are aggregated and calculated from the data from the main memory.

Step to set the read mode:

Go to transaction RSDIPROP; specify the info provider name and change.

InfoProvider: Edit Properties

You use this transaction to edit performance, data load and query properties of an InfoProvider

InfoProvider: ZAD_SALES
Description:

Editing Functions: Display, Change

Now edit the properties as shown below.

Read Mode decides how many records are read during navigation.

InfoProvider: Edit Properties

InfoProvider: ZAD_SALES
Description: Sales Overview

Query / Cache | Roll Up | DB Performance

Read Mode: H Query to Read When You Navigate or Expand Hierarchies
Cache Mode: 1 Main Memory Cache Without Swapping
Persistence Mode: 0 Inactive
Data Integrity:

Delta Cache
Delta Cache Class: CL_RSD_DC_SUPPORT_INFOCUBE

System Default

Related Content

http://www.comeritinc.com/UserFiles/file/Berg_NW2010_EDW_v4.pdf

<http://www.comeritinc.com/UserFiles/file/tips%20tricks%20to%20speed%20%20NW%20BI%20%202009.ppt>

<http://searchsap.techtarget.com/answer/Performance-tuning-in-SAP-BW>

For more information, visit the [EDW homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.