

How to Create Secondary Index on DSO's in SAP BW/BI 7.x



Applies to:

SAP BW/BI 7.x. For More Information visit the [Business Intelligence homepage](#).

Summary

This tutorial will give you an understanding on steps used to create indexes on Data Store Objects in SAP BI to improve performance of queries built on top of DSO's for operational reporting and also improve performance of select queries (lookup's) on non key fields of DSO mostly in the Harmonization layer.

Author: Vijay Kumar Nagaraj

Company: Accenture

Created on: 26 October 2011

Author Bio



Vijay Kumar Nagaraj is a Certified SAP BW consultant with 3+ years of experience in SAP BW/BI and SAP BI ABAP.

Table of Contents

Overview	3
Indexes in Contrast with DSO.....	4
Steps to create Indexes on DSO	5
Pros and Cons of Secondary Indexes on DSO	8
Pros:	8
Cons:.....	8
Related Content.....	9
Disclaimer and Liability Notice.....	10

Overview

A **database index** is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes.

An index can be considered a copy of a database table that has been reduced to certain fields. This copy is always in sorted form. Sorting provides faster access to the data records of the table, for example using a binary search. The index also contains a pointer to the corresponding record of the actual table so that the fields not contained in the index can also be read.

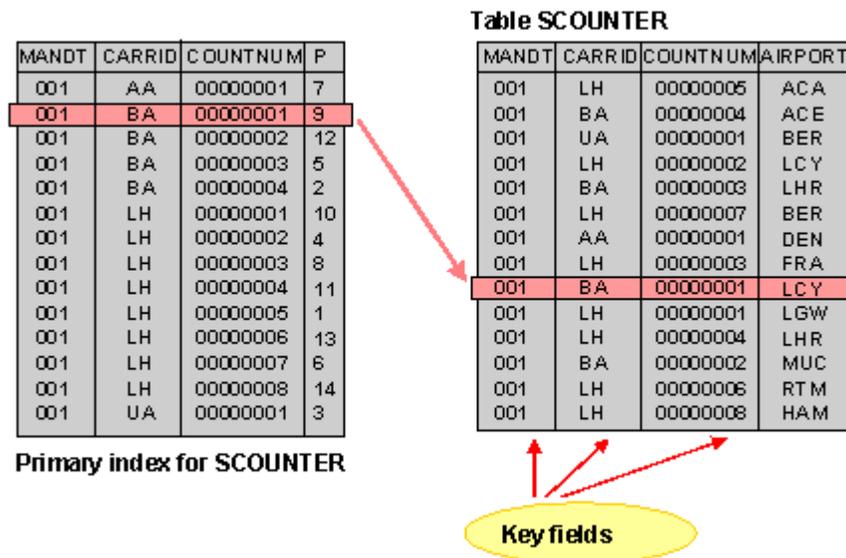
The optimizer of the database system decides whether an index should be used for a concrete table access.

There are two basic categories of indexes in general:

- 1) Primary Index
- 2) Secondary Index

- 1) Primary index usually contains all the key fields of a database table and a pointer to the original data records as shown in the figure below

Figure 1: Primary Index



- 2) A secondary index, simply, is a way to efficiently access records in a database by means of some piece of information other than the usual (primary) key. In other words, secondary index will be necessary if a table is accessed in a way that does not take advantage of the sorting of the primary index for the access.

As an example of how secondary indexes might be used, consider a database containing a list of students at a college, each of whom has a unique student ID number. A typical database would use the student ID number as the key; however, one might also reasonably want to be able to look up students by last name. To do this, one would construct a secondary index in which the secondary key was this last name.

Indexes in Contrast with DSO

All the key fields of a DSO form a Primary Index for that DSO. So, all the lookups and queries which access DSO based on these key fields shall make use of Primary Indexes which helps improve the read performance.

However, SAP has provided an option to create Secondary Indexes on a DSO which is the main topic of our discussion. Here we can drag all the characteristics/ Key figures we wish to be a part of secondary index. This is shown in the screen shot below:

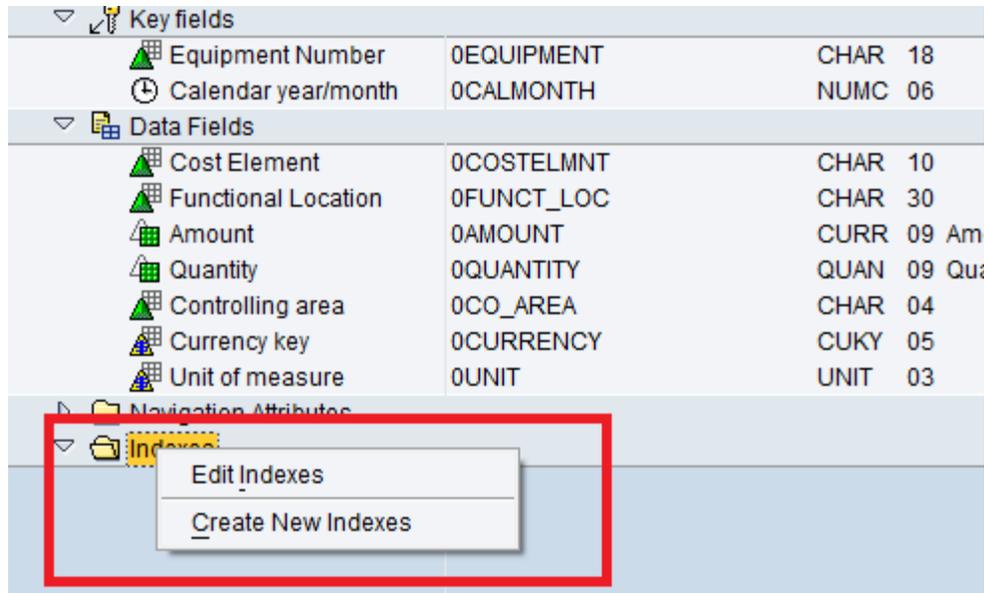
Figure 2: Test DSO

DataStore Object	Techn. name /value	Fu...	O...	Data ...	L	K
Test Indexes	ZTESTDSO					
Object Information						
Version	◆ New					
Save	⊖ Not saved					
Object Status	📁					
Settings						
Key fields						
Equipment Number	0EQUIPMENT			CHAR	18	
Calendar year/month	0CALMONTH			NUMC	06	
Data Fields						
Cost Element	0COSTELMNT			CHAR	10	
Functional Location	0FUNCT_LOC			CHAR	30	
Amount	0AMOUNT			CURR	09	Am
Quantity	0QUANTITY			QUAN	09	Qua
Controlling area	0CO_AREA			CHAR	04	
Currency key	0CURRENCY			CUKY	05	
Unit of measure	0UNIT			UNIT	03	
Navigation Attributes						
Indexes						

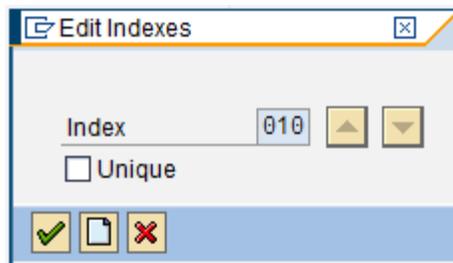
Steps to create Indexes on DSO

Step 1: Open Data warehousing workbench (RSA1) and open your DSO in change mode to arrive at DSO maintenance screen. We are using a test DSO in our example (Refer Fig 2 above).

Step 2: Right Click on the folder named "Indexes" in the DSO maintenance screen, in the context menu you get two options as shown in the below screen. Select "**Create New Indexes**".



Step 3: After selecting Create New Indexes option, you get a pop up screen as shown below. Click on OK button to create an Index. You can create more than one Index using the create button in the middle.



Note: Always select unique check box if you are sure that the characteristics you choose as a part of secondary Index will have unique values for every record in the DSO.

Step 4: Notice that a new Index (non-unique) folder is created just below Index folder. Now you can drag and drop all the required characteristics that you wish to be a part of secondary Index in the new Index folder.

▼	🔑	Key fields		
	🏠	Equipment Number	0EQUIPMENT	CHAR 18
	🕒	Calendar year/month	0CALMONTH	NUMC 06
▼	📊	Data Fields		
	🏠	Cost Element	0COSTELMNT	CHAR 10
	🏠	Functional Location	0FUNCT_LOC	CHAR 30
	📊	Amount	0AMOUNT	CURR 09 Am
	📊	Quantity	0QUANTITY	QUAN 09 Qu
	🏠	Controlling area	0CO_AREA	CHAR 04
	🏠	Currency key	0CURRENCY	CUKY 05
	🏠	Unit of measure	0UNIT	UNIT 03
▶	📁	Navigation Attributes		
▼	📁	Indexes		
	▼	📁 010 (non-unique)		

Note: It is always recommended to use Characteristic Info objects as a part of secondary index though we can create Indexes on Key Figure Info objects.

Step 5: Drag and drop the required characteristics in the Indexes folder. The screen below shows three characteristics are added as a part of secondary indexes (0COSTELEMNT, 0CO_AREA and 0FUNCT_LOC).

▼ 🔑	Key fields			
▲	Equipment Number	0EQUIPMENT	CHAR	18
🕒	Calendar year/month	0CALMONTH	NUMC	06
▼ 📄	Data Fields			
▲	Cost Element	0COSTELEMNT	CHAR	10
▲	Functional Location	0FUNCT_LOC	CHAR	30
▲	Amount	0AMOUNT	CURR	09 Am
▲	Quantity	0QUANTITY	QUAN	09 Quz
▲	Controlling area	0CO_AREA	CHAR	04
🌐	Currency key	0CURRENCY	CUKY	05
📏	Unit of measure	0UNIT	UNIT	03
▶ 📁	Navigation Attributes			
▼ 📁	Indexes			
▼ 📁	010 (non-unique)			
▲	Cost Element	0COSTELEMNT	CHAR	10
▲	Controlling area	0CO_AREA	CHAR	04
▲	Functional Location	0FUNCT_LOC	CHAR	30

Step 6: Now that you have added your characteristics as secondary Index next step is to generate Secondary Indexes. For this SAP has provided standard Function Modules for creation and deletion of Indexes as mentioned below.

Create Index: RSSM_PROCESS_ODS_CREA_INDEXES

Drop Index: RSSM_PROCESS_ODS_DROP_INDEXES

Execute these FM's manually and give the corresponding DSO names as inputs for creation and deletion of Indexes respectively.

It is recommended to add these two Function Modules to your process chains using **custom process types** or create two ABAP programs that call these FM's and add these two **ABAP program process type's** to the process chain before and after the data load to the DSO.

Pros and Cons of Secondary Indexes on DSO

Pros:

- 1) Improved Query response time for queries created on DSO provided they are using characteristics as filters which are non-key fields of the DSO and are added as a part of secondary Index.
- 2) Lookup performance is improved when doing a lookup on DSO using SELECT Statement which uses non-key fields in the where condition such as

```
SELECT *  
  FROM DSO  
 WHERE 0costelemnt = 'x'  
 AND    0funct_loc = 'x'  
 AND    0co_area = 'x'.
```

In the above Select Statement characteristics 0costelemnt, 0funct_loc and 0co_area are non-key fields of testdso, if we add all the three characteristics to the secondary index, the above Select statement will perform an optimized read on the database based on the given selection conditions.

Cons:

- 1) Additional indexes can also place a load on the system since they must be adjusted each time the table contents change. Each additional index therefore slows down the insertion of records in the table, (in order to avoid this we can drop the Indexes before the start of data load to the DSO then later recreate it).
- 2) Two or more indexes on a table should contain as few fields in common as possible. If two indexes on a table have a large number of common fields, this could make it more difficult for the database optimizer to choose the most selective index.
- 3) There is a constraint on the number of Indexes we can create on a DSO, therefore, try to use fewer Indexes as possible on the DSO.

Related Content

[Indexes](#)

[Secondary Indexes](#)

[Difference between primary and secondary Index](#)

For more information, visit the [Business Intelligence 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.