SANA FOR SAP ECC

IMPROVE PERFORMANCE OF PRODUCT AND CUSTOMER INDEXING

sana
The shortcut to e-commerce

INTRODUCTION

Sana uses the **Product import** task in Sana Admin to retrieve material (product) data from SAP ECC to build and update the catalog of a webstore, and the **Customer import** task to retrieve and update customer information.

When you change material or customer data in SAP ECC, the value of the **Last Date/Time Modified** field of the material or customer will be updated with the current date and time. Based on this, Sana checks whether some material or customer data has been changed in SAP ECC since last indexing and it updates the product or customer index by synchronizing material and customer changes between SAP ECC and Sana. When products and customers are indexed by Sana, material and customer changes you have made in SAP ECC will be available in your webstore.

PRODUCT AND CUSTOMER INDEXING BOTTLENECK

In SAP ECC data changes are stored in the **CDHDR** table (Change Document Header). This is a standard SAP ECC table and it stores a lot of data. The keys in this table are object class - material and customer, and material and customer numbers.

Field	Key	Ini...	Data element	Data Type	Length	Deci...	Short Description
MANDANT	☑	☑	MANDT	CLNT	3	0	Client
OBJECTCLAS	☑	☑	CDOBJECTCL	CHAR	15	0	Object class
OBJECTID	☑	☑	CDOBJECTV	CHAR	90	0	Object value
CHANGENR	☑	☑	CDCHANGENR	CHAR	10	0	Document change number
USERNAME	☐	☐	CDUSERNAME	CHAR	12	0	User name of the person responsible in change document
UDATE	☐	☐	CDDATUM	DATS	8	0	Creation date of the change document
UTIME	☐	☐	CDUZEIT	TIMS	6	0	Time changed
TCODE	☐	☐	CDTCODE	CHAR	20	0	Transaction in which a change was made
PLANCHNGNR	☐	☐	PLANCHNGNR	CHAR	12	0	Planned change number
ACT_CHNGNO	☐	☐	CD_CHNGNO	CHAR	10	0	Change number of the document created by this change
WAS_PLANND	☐	☐	CD_PLANNED	CHAR	1	0	Flag that changes were generated from planned changes
CHANGE_IND	☐	☐	CDCHNGINDH	CHAR	1	0	Application object change type (U, I, E, D)
LANGU	☐	☐	LANGU	LANG	1	0	Language Key
VERSION	☐	☐	CHAR3	CHAR	3	0	3-Byte field

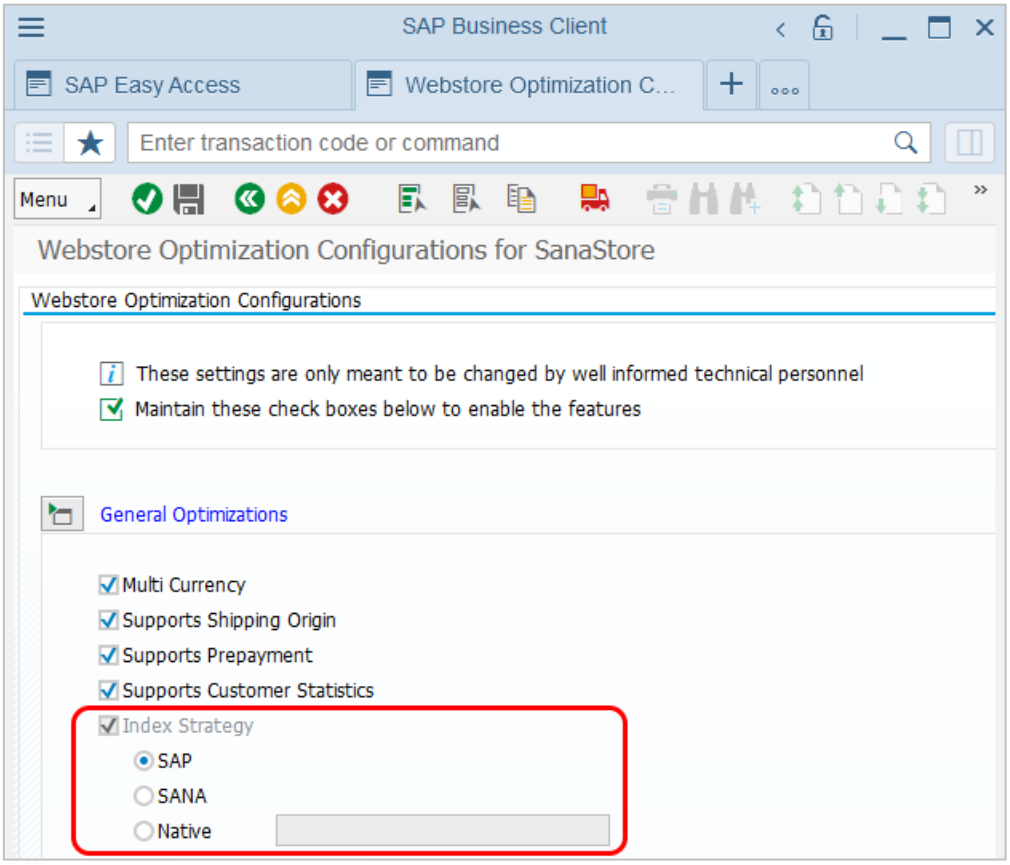
Sana retrieves material and customer changes from the **CDHDR** table in SAP ECC based on the **UDATE** (Update date) and **UTIME** (Update time) fields. Since **UDATE** and **UTIME** are non-key fields, fetching data from a query based on these fields can be slow or even lead to a timeout depending on the amount of data. If you need to change some data of a large number of materials or customers in SAP ECC at once, it can take a lot of time for Sana to update the product or customer index.

Some companies use the SAP system for many years and they have millions of records in the **CDHDR** table which makes it difficult for Sana to access this table and check for material and customer data changes there.

INDEXING STRATEGIES

Sana for SAP ECC has three indexing strategies which can be tested on a customer's environment to find the most suitable one, if you encounter performance issues with product and customer indexing.

You can change the indexing strategy in the **Webstore Optimization** window in SAP. You can access **Webstore Optimization** either from the main window of the Sana Commerce solution (**/n/sanaecom/webstore**), or by calling the transaction **/n/sanaecom/optimize**.



- **SAP** – the default indexing strategy which retrieves product and customer changes directly from the standard SAP table **CDHDR**.
- **SANA** – copies product and customer changes from the standard SAP table **CDHDR** to the custom Sana table. Sana checks product and customer changes not in the standard SAP table **CDHDR** but in the custom table. You can schedule a background job for how long the product and customer changes must be stored in the custom Sana table and clean up the history automatically.
- **Native** – retrieves product and customer changes from the database view. You can create a view in your SAP database to store product and customer changes there. In this case direct database calls are made to retrieve product and customer changes from the database view which improves performance.

We recommend using the default **SAP** indexing strategy. If you encounter some performance issues with product and customer indexing, first we recommend to consider the actions listed below to improve indexing performance of the default **SAP** indexing strategy.

IMPROVE SAP INDEXING STRATEGY

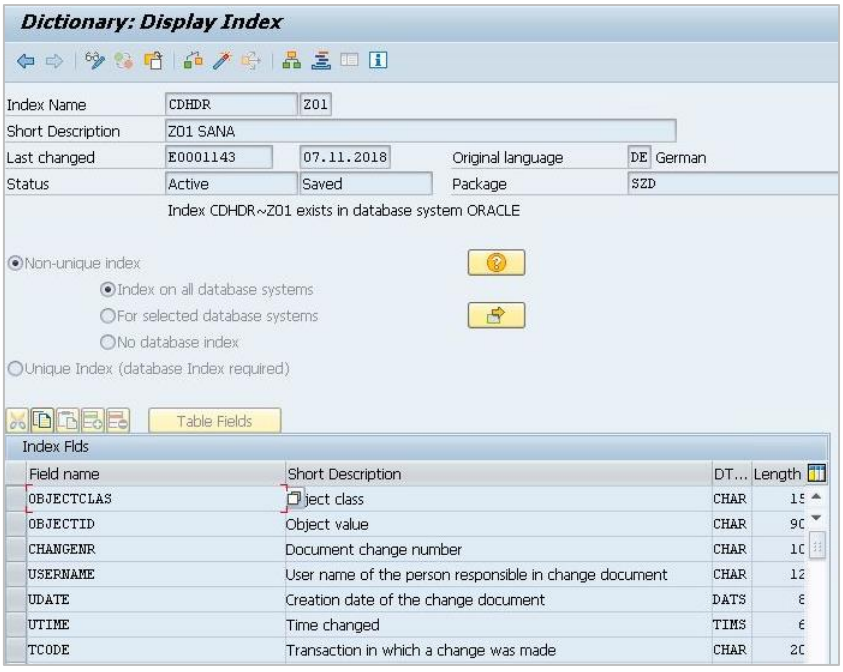
Recommendations listed below are applicable to the **SAP** indexing strategy which retrieves product and customer changes directly from the standard SAP ECC **CDHDR** table.

1. Create an index for the **UDATE** and **UTIME** fields of the **CDHDR** table. Contact your Sana project manager or Sana partner for more details.
2. Implement a **BADI**. Contact your Sana project manager or Sana partner for more details.
3. Reduce the number of records in the **CDHDR** table.
4. Increase system resources.

OPTION 1: CREATE INDEX FOR THE CDHDR TABLE

Indexing the **CDHDR** table or archiving the old records can significantly improve performance of the queries based on the **UDATE** (Update date) and **UTIME** (Update time) fields. This approach does not require any code changes or custom implementations.

You can ask the BASIS team to create an index as shown on the screenshot below for the **CDHDR** table or archive the old records in the table.



Advantages

1. This is the standard SAP approach.
2. No code changes and custom implementations.

OPTION 2: IMPLEMENT A BADI

If you cannot create index or optimize the **CDHDR** table, you can think about implementation of the **BADI_MATERIAL_CHECK** BADI to improve performance of products indexing, but this solution requires customizations both in SAP and the Sana add-on for SAP.

You must implement a BADI and improve the **GetProducts** API method:

1. The method **BADI_MATERIAL_CHECK~CHECK_DATA** should include the logic to update the **/SANAECOM/MARA** table with the current timestamp.

2. The method **/SANAECOM/CL_APP_GETPRODUCT~ GET_UPDATED_MATERIALS** should be overridden with the new logic to retrieve data from the **/SANAECOM/MARA** table instead of the standard SAP table **CDHDR**.

Advantages

1. There is no dependency on the standard SAP table **CDHDR**.

Disadvantages

1. The standard logic of the Sana add-on for SAP ECC must be changed.
2. It requires customizations in the customer's system which cannot be included in the standard product.
3. If you use the external system to update materials, for example "IDoc", you must check whether this approach will work for you.
4. Some material changes may not be retrieved as when using the **CDHDR** table.

OPTION 3: ARCHIVE CDHDR TABLE RECORDS

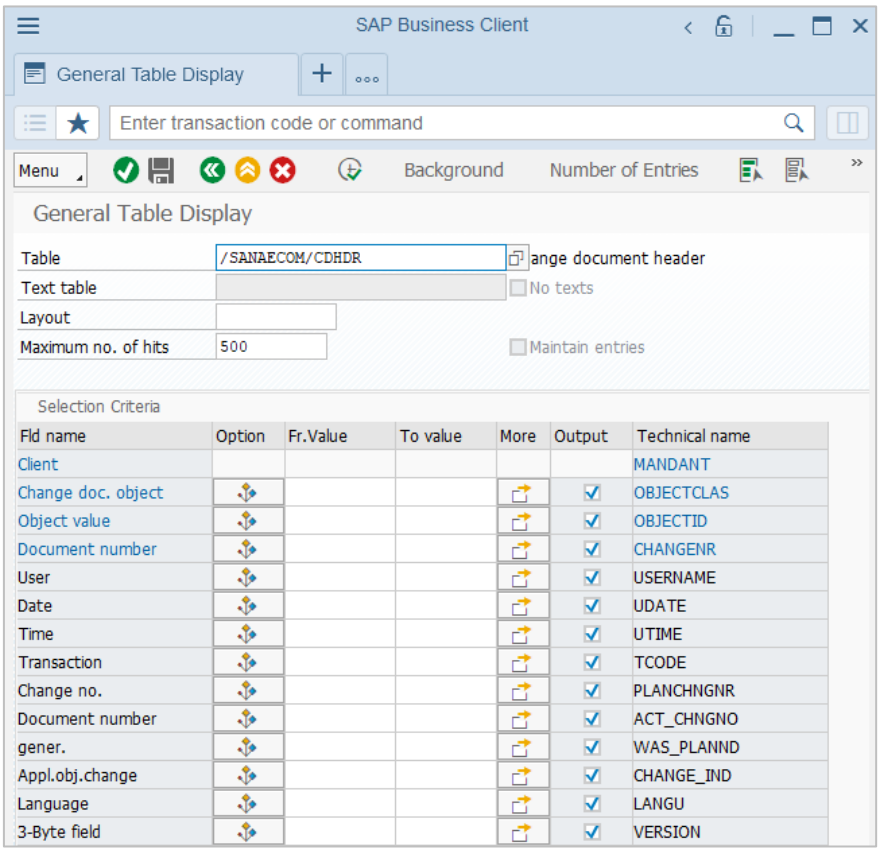
You can archive the records in the **CDHDR** table to improve record search performance. You are free to decide how many records to store in the **CDHDR** table.

OPTION 4: INCREASE SYSTEM RESOURSES

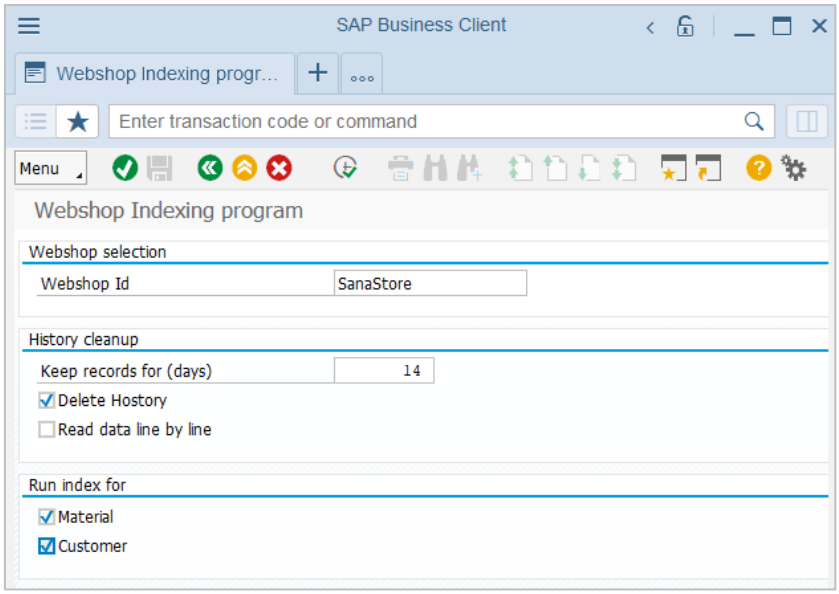
You can also think about increasing system resources to improve performance.

SANA INDEXING STRATEGY

When using the **SANA** indexing strategy, it copies product and customer changes from the standard SAP table **CDHDR** to the custom Sana table **/SANAECOM/CDHDR**. When updating the product and customer indexes, Sana checks product and customer changes not in the standard SAP table **CDHDR** but in the custom Sana table.



Sana provides the **Webshop Indexing** program in SAP which you can access by calling the transaction code **/n/sanaecom/index**.



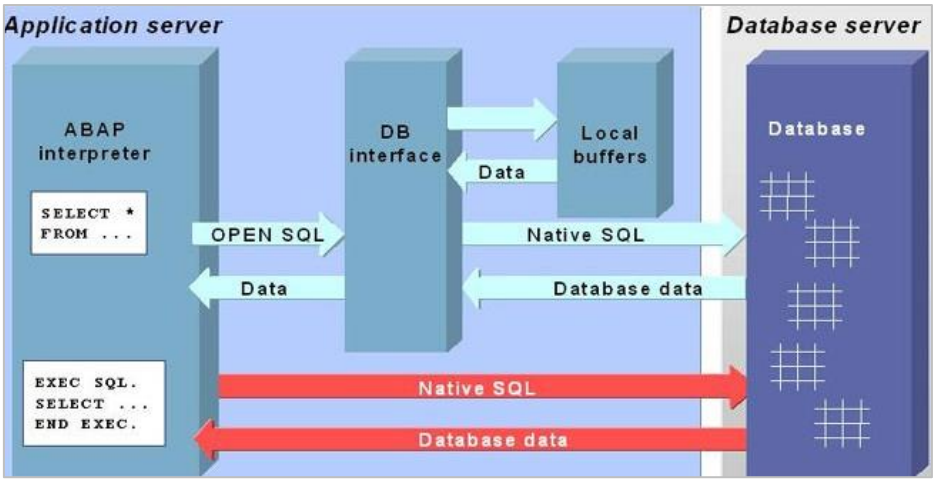
There you can define how many days to keep material and customer changes for Sana indexing in the custom Sana table and automatically delete the history.

You can create a background job to run this Sana program automatically according to the schedule.

NATIVE INDEXING STRATEGY

Using the **Native** indexing strategy Sana retrieves product and customer changes from the SAP database view. You can create a view in your SAP database to store product and customer changes there. You can use this approach for any SAP database supported by Sana. In this manual Microsoft SQL database is used as an example.

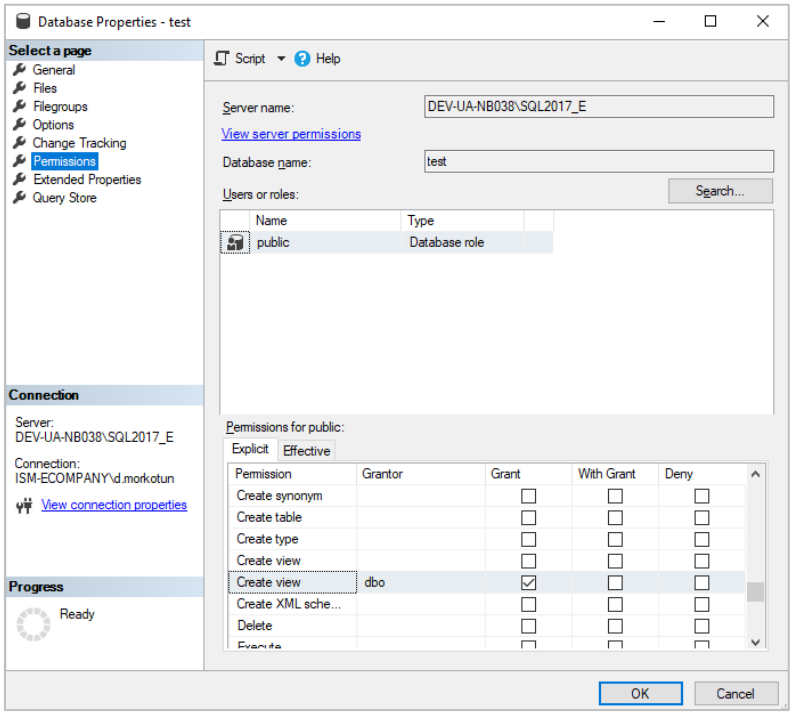
The **Native** indexing strategy uses a database cursor with Native SQL. It uses a direct database call producing a significant performance improvement. Native SQL allows you to use database-specific SQL statements in an ABAP program. This means that you can use database tables that are not administered by ABAP dictionary, and therefore integrate data that is not a part of the SAP system.



To use the **Native** indexing strategy, you must follow the steps below.

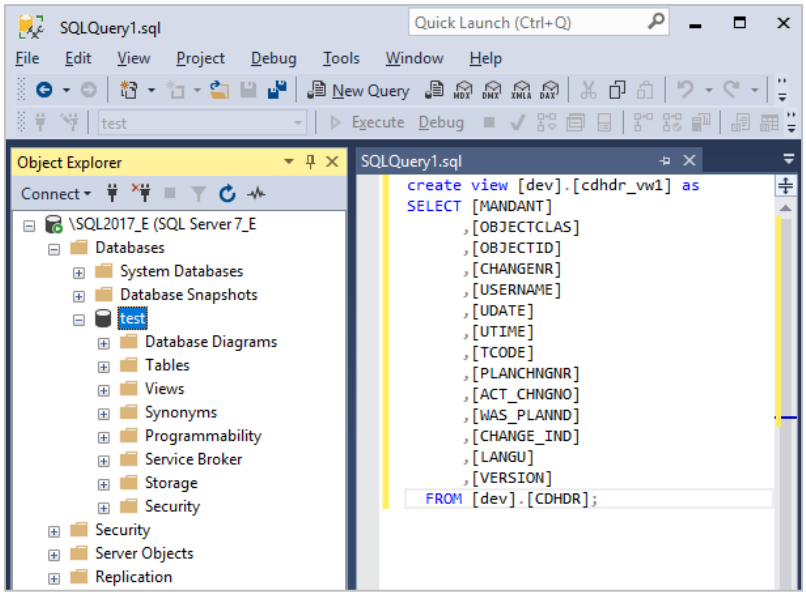
CREATE A DATABASE VIEW FOR CDHDR TABLE

Step 1: Log in to your SAP database and check if you have permissions to create a view.



Step 2: Select your database and create a query for it. You must create a view in your SAP database with the same fields as in your **CDHDR** SAP table.

```
create view [dev].[cdhdr_vw1] as
SELECT [MANDANT]
      ,[OBJECTCLAS]
      ,[OBJECTID]
      ,[CHANGENR]
      ,[USERNAME]
      ,[UDATE]
      ,[UTIME]
      ,[TCODE]
      ,[PLANCHNGNR]
      ,[ACT_CHNGNO]
      ,[WAS_PLANND]
      ,[CHANGE_IND]
      ,[LANGU]
      ,[VERSION]
  FROM [dev].[CDHDR];
```



Step 3: Check if the database view has been created and populated with data.

ENABLE NATIVE INDEXING IN SAP

Step 1: In SAP open the **Webstore Optimization** window either from the main window of the Sana Commerce solution (**/n/sanaecom/webstore**), or by calling the transaction **/n/sanaecom/optimize**.

Step 2: Select the **Native** indexing strategy and enter the name of the database view you created.