

**Andrej Brassas** <andrej.brassas@gmail.com>

SAP - MixCont interface test - table definition2 meddelanden

Pieter Markus <Pieter.Markus@provide.nl>

15 januari 2014 00:03

Till: "stas.brassas@mixcont.com" <stas.brassas@mixcont.com>, "Andrej Brassas (andrej.brassas@mixcont.com)" <andrej.brassas@mixcont.com>

Kopia: "'Peter Boon' (peter.boon@qewrubber.com)" <peter.boon@qewrubber.com>, "'Cor Gringhuis' (cor.gringhuis@qewrubber.com)" <cor.gringhuis@qewrubber.com>, "Reyer Sneller (rsneller@telfort.nl)" <rsneller@telfort.nl>

Hello Stas / Andrei,

During my tests with the SAP <-> MixCont interface I came across some issues I would like to share. I noticed that in the interface tables, a number of primary and foreign keys are defined. For example, on the 'vendor' table the column 'VendorNo' is marked as Primary Key. In the ItemPrice table, the 'ItemCode' column is defined as primary key. In my opinion, this is not correct, because for example a vendor record can be listed in this table multiple times, but with a different 'dateExport' value. The same is true for the ItemPrice record. Next, the ItemPrice table contains a foreign key to the 'raw_material' table. It is not necessarily true to say that each raw material item that is listed in SAP with a price, is also listed in the 'Raw_material' table from MixCont.

For logging and tracing purpose, I think we should keep a history of records in the interface tables which could give errors if we stick to the 'functional' primary key (vendorNo, ItemCode, etc.). That is the reason I added the 'ID' column in the interface specification and marked that as being the primary key. Not for enforcing relations between tables in the interface database, but to ensure that at least all records have a primary key to be uniquely identified.

In my testenvironment I changed the primary and foreign keys, I haven't changed the datamodel of these tables in the MCSap database at the MixCont server. I am not sure I have the permissions to do this, but I can check tomorrow.

What's your opinion on the use of the primary and foreign keys?

Best regards,

Pieter Markus CPIM CIRM, Business Unit Manager

Provide Business Solutions B.V. | De Dompelaar 1-D | 3454 XZ De Meern | The Netherlands | T. +31 (0)30 6669600 | CC 11033102 | www.provide.nl

Andrej Brassas <andrej.brassas@mixcont.com>

15 januari 2014 14:52

Till: Pieter Markus <Pieter.Markus@provide.nl>, peter.boon@qewrubber.com, cor.gringhuis@qewrubber.com, "S.Brassas" <stas.brassas@mixcont.com>

Hello Pieter,

"During my tests with the SAP <-> MixCont interface I came across some issues I would like to share. I noticed that in the interface tables, a number of primary and foreign keys are defined. For example, on the 'vendor' table the column 'VendorNo' is marked as Primary Key. In the ItemPrice table, the 'ItemCode' column is defined as primary key. In my opinion, this is not correct, because for example a vendor record can be listed in this table multiple times, but with a different 'dateExport' value. The same is true for the ItemPrice record."

Could you please name any advantages of multiple storing of each vendor (storing of old data) because in our understanding it means troubles. Data redundancy is the thing we always try to avoid to enhance the reliability.

Example: MixCont performs an export to "raw_material_vendor" table. In case as it is now namely only one instance of each vendor is allowed to be listed in the "vendor" table. Our program goes through each of the rows in the "vendor" table, updates the row in the "raw_material_vendor" table if it already exists or inserts row if missing. This results into a very fast & reliable export process.

If the "vendor" table contains same vendor multiple times it means that we on our side have to make filtering logic's to prevent exporting a wrong/old instance of a vendor.

Another very important issue is related to the "vendor_contact" table. As the database design is now, each vendor can have "0 to many" vendor contacts, how shall we bind vendor with it's contacts if same vendor is stored multiple times in the "vendor" table and the same vendor contact is stored multiple times in the "vendor_contact" table, in this case the binding is only possible with help of "dateExport" which means unneeded complexity.

Having vendor contacts in table vendor is not so good either because different vendor have different amount of contacts which means that we must have fixed amount of reserved columns.

The same opinion regarding multiple storing we have about the "item_price" table.

It is not necessarily true to say that each raw material item that is listed in SAP with a price, is also listed in the 'Raw_material' table from MixCont.

As we understand, there cannot be "itemCodes" in SAP which are not present in MCCompound?

It was desired that raw materials are created only in MCCompound program. If our statement is correct there won't be any problems of exporting to the "item_price" table.

For logging and tracing purpose, I think we should keep a history of records in the interface tables which could give errors if we stick to the 'functional' primary key (vendorNo, ItemCode, etc.).

We think if we keep to the multiple storing model (without using primary, foreign keys) this would mean a database without any structure which also means many troubles and complexity. If we just think of "recipe_raw_material" table just after lets say 10 exports the table will contain 500000 records instead of 50000 with a rational database design.

For logging and tracking purpose we can create a clone of each table but without any primary/foreign key constraints (only id for each row), for example for logging purpose of table "vendor" we create a table named "vendor_log", so each time the data has to be exported to the "vendor" table it will be exported both to "vendor" & to the "vendor_log" tables. In this case we have a rational database design and also tracking possibilities.

Best Regards

Andrei Brassas
MixCont AB

[Citerad text är dold]