**Doc for One Web**

**Database Documentation**

**For**

**DOC41WEB**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DOCUMENT HISTORY | | | | |
| Version No. | Change Date | Change Request No. | Change Description | Author |
| 1.0.0 | 24.09.2019 |  | Initial Version | Ingeborg Menge |

Content

[1 General 4](#_Toc20233028)

[1.1 Participants 4](#_Toc20233029)

[1.2 JIRA concepts 4](#_Toc20233030)

[1.3 Subversion 4](#_Toc20233031)

[1.4 Helpful Documents 4](#_Toc20233032)

[2 Databases 5](#_Toc20233033)

[2.1 Standard Development 5](#_Toc20233034)

[2.2 Database Rules 5](#_Toc20233035)

[2.2.1 Foundation columns 6](#_Toc20233036)

[2.2.2 Naming Rules 6](#_Toc20233037)

[2.2.3 Indexes and Constraints 7](#_Toc20233038)

[2.2.4 Trigger XXX\_CDT 7](#_Toc20233039)

[2.3 Documentation Files 8](#_Toc20233040)

[2.3.1 DOC41WEB\_MGRSchema.sql 8](#_Toc20233041)

[2.3.2 DOC41WEB\_MGRSchema-Alter.sql 13](#_Toc20233042)

[2.3.3 Folder cmn 14](#_Toc20233043)

[2.3.4 Folder usermgmt 14](#_Toc20233044)

[2.3.5 Foundation Files 14](#_Toc20233045)

[2.4 Table Prefix 14](#_Toc20233046)

[2.5 Deployment on Production 15](#_Toc20233047)

[2.6 Translations 15](#_Toc20233048)

[2.6.1 CMNTranslationDepoly\_DOC41WEB\_QA\_DEV.chn 15](#_Toc20233049)

[2.6.2 CMNTranslationDepoly\_DOC41WEB\_QA\_PROD.chn 15](#_Toc20233050)

[2.6.3 CMNTranslationDepolyonly\_Initial\_DOC41WEB\_DEV\_QA.chn 15](#_Toc20233051)

[2.7 Displaytexts 15](#_Toc20233052)

# General

This document describes Doc4One Web also written DOC41WEB

DOC41WEB is an old portal and it is running smoothly.

The DOC41WEB database is an Oracle database, current version 12.2.

## Participants

DOC41WEB Project Management (Leader for Business Services is Frank Wiesen IMWIF)

DOC41WEB Development/Support Team (QuinScape Managed Service)

DOC41WEB Database Team QuinScape Managed Service)

eComOBA Oracle Administration

eCommerce Platforms as owner of the DOC41WEB and the database

## JIRA concepts

JIRA is a tool used for bug tracking, ticket tracking, and project management.

This document does not describe the functionality of JIRA. In case further information is required, use the [following Link](http://connections.bayer.com/wikis/home?lang=de-de#!/wiki/We9ce959ebde8_4158_8677_aa131bdf4946) for JIRA description or the online help function.

Only main components important for the release management process are brief described.

Project name BORDERS

## Subversion

Subversion is a tool to store files with versions. You can tag a version to remember which versions are together deployed.

We use Subversion to store the files used by the database release management. You will find all files described below in the Subversion repository.

Use the [following link](https://by-stsvn.bayer-ag.com/svn/DOC41WEBUI/trunk/doc41web) for the Subversion repository for DOC41WEB.

## Helpful Documents

These documents have some information about Foundation and Chains.

You find it in the foundation Subversion repository

../foundation/doc/Chainprozesse-Benutzerhandbuch.docx - Develop Chains (in German)

../foundation/doc/Chainprocess\_Manual.docx - Develop Chains (in English)

../foundation/doc/Interfaces Chains und DB Vertretung.docx - Holiday Backup for Chains and DB

../foundation/doc/Interfaces Chains Lev.docx - Short overview for Leverkusen

../foundation/doc/Interfaces Chains Berlin.docx - Short overview for Berlin

../DOC41WEB/chains/doc/DOC41WEB-Chains.docx - DOC41WEB Chains

Unfortunately, only in German available.

../foundation/doc/Foundation DB Konzept.pptx - Basics for Foundation DBs

../foundation/doc/Chains.pptx - Basics for Chains

# Databases

DOC41WEB use two schemata on the eCommerce Platforms databases GBOD, GBOQ and GBOP. There are always a DOC41WEB\_FDT schema for the foundation tables and a DOC41WEB\_MGR schema for the DOC41WEB specific tables.

| **Schema@database** | **Use Case** | **Appr.** |
| --- | --- | --- |
| DOC41WEB\_FDT@GBOD  DOC41WEB\_MGR@GBOD | Development System | D |
| DOC41WEB\_FDT@GBOQ  DOC41WEB\_MGR@GBOQ | Test System | Q |
| DOC41WEB\_FDT@GBOP  DOC41WEB\_MGR@GBOP | Production System | P |

Be aware that an error in using a schema could affect the whole database. E.g. long running SQL statements can have time effects of the SQL in other schemas in the same database.

## Standard Development

Normally the changes were developed at Development System and module tests were done.

If the test were successful, the changes were transported to Q.

If the test were successful, the files were committed to Subversion.

The changes were transported to Production System on the release date or if possible one day before release date.

## Database Rules

This are global rules that helps me to avoid mistakes. This are only proposals.

I avoid using materialized view because they have always a problem with the time for updating. Normal views can be helpful, but I try to use it rare and they are marked with prefix V\_. There could be problems if the view must be recompiled.

Tablespace DOC41WEB\_DAT is used for all object except indexes.

Tablespace DOC41WEB\_IDX is used for indexes.

Grant the object to machine user MXDOC41WEB

Example:

GRANT DELETE, INSERT, SELECT, UPDATE ON DOC41WEB\_FDT.COUNTRY TO MXDOC41WEB WITH GRANT OPTION;

### Foundation columns

DOC41WEB use the foundation as base and so every table needs the foundation base columns. These columns are always the first in the table with one difference. Standard foundation has the OBJECT\_ID as primary key.

|  |  |
| --- | --- |
| OBJECT\_ID | A unique key for every record in the schema. **Filled automatically** via trigger with a database sequence. Should not be filled by files or programs. |
| CREATED | The date when the record was created. **Filled automatically** via trigger. Should not be filled by files or programs. |
| CHANGED | The date when the record was last changed. **Filled automatically** via trigger. Should not be filled by files or programs. |
| CREATEDBY | Who has created the record? Filled by files or programs. |
| CHANGEDBY | Who has last changed the record? Filled by files or programs. |
| CREATEDON | From which server was the record created? Filled by files or programs. |
| CHANGEDON | From which server was the record changed? Filled by files or programs. |
| OBJECTSTATE\_ID | State of the record. At the moment 0 = deleted; 1 = active; 2 = inactive. State 2 was only set by the program.  They could be tables without OBJECTSTATE. |

CREATEDBY and CHANGEDBY were filled with the CWID if the record was created by users. It was filled with \_xxx\_ if it was created by a Chain Tool job.

The xxx could have different values, e.g. \_INIT\_ for initial import jobs, \_SAP\_ for SAP import jobs.

### Naming Rules

For DOC41WEB I decide to use singular form for names. E.g. a table is called CUSTOMER and not CUSTOMERS.

All table names have a prefix (see Table Prefix).

Don’t use simple names for database objects. E.g. never NAME, always PRODUCT\_NAME, CATALOGUE\_NAME. Never FILE or FILENAME, always IMAGE\_FILE or IMAGE\_FILENAME. This avoids problems with Oracle reserved words.

The suffix \_ID is used only for internal database references.

E.g. CUSTOMER\_ID is the reference to OBJECT\_ID of SAP\_CUSTOMER table.

OBJECTSTATE\_ID is the reference to OBJECTSTATE\_ID of DOC41WEB\_FDT.OBJECSTATE table.

For idents like PRODUCT ID use PRODUCT\_CODE or PRODUCT\_KEY.

Try to use the same column name for the same thing and the same precision. E.g. always PRODUCT\_CODE and not PRODUCTCODE, PROD\_CODE, PRD\_CD.

Use \_ to separate words. E.g. PRODUCT\_CODE and not PRODUCTCODE.

Booleans would be used as NUMBER(1) with the values 0 = false, 1 = true and with the prefix IS\_ or HAS\_ or ALLOW\_ or WARN\_ to find them easy.

Use a suffix for trigger name to show the kind of trigger. Exception foundation \_CDT trigger

\_A for AFTER trigger

\_B for BEFORE trigger

I for INSERT trigger

U for UPDATE trigger

D for DELETE trigger

Example:

UPS\_PARTNER\_BIU

BEFORE INSERT OR UPDATE ON

DOC41WEB2\_MGR.UPS\_PARTNER

FOR EACH ROW

DECLARE

At the moment only \_CDT trigger exists.

### Indexes and Constraints

Use prefix PK\_ for primary keys.

Use prefix UK\_ for unique keys.

Use prefix FK\_ for foreign keys.

Use prefix CK\_ for check constrains.

For primary and unique keys, an index with the same name is always created.

The OBJECT\_ID is always part of the primary key.

Try to create a unique key beside the object key with the columns special for the table.

Use the table name after the prefix if the table name is not too long. Maximum length is 32 character.

For every foreign key relation, a foreign key must be created to avoid orphan records.

### Trigger XXX\_CDT

Every new table get the trigger XXX\_CDT to set the foundation columns (XXX = table name). This is recommended.

Example:

CREATE OR REPLACE TRIGGER DOC41WEB\_MGR.MD\_SYSPARAM\_CDT

BEFORE INSERT OR UPDATE ON DOC41WEB\_MGR.MD\_SYSPARAM

FOR EACH ROW

DECLARE

-- declare variables...

BEGIN

-- protect the create columns

IF (INSERTING) THEN

:new.created := DOC41WEB\_FDT.get\_Normalized\_Time();

IF (:new.changedBy IS NULL) THEN

:new.changedBy := :new.createdBy;

END IF;

END IF;

IF (UPDATING) THEN

:new.created := :old.created;

:new.createdBy := :old.createdBy;

:new.createdOn := :old.createdOn;

END IF;

:new.changed := DOC41WEB\_FDT.get\_Normalized\_Time();

IF (INSERTING AND :new.object\_Id IS NULL) THEN

:new.object\_Id := DOC41WEB\_FDT.get\_Next\_Oid();

ELSIF (UPDATING) THEN

:new.object\_Id := :old.object\_Id;

END IF;

END;

/

The name of the table must be changed.

## Documentation Files

All changes in the database must be documented in so called Alter-Steps. Every Alter-Step documented one change phase and can be executed separately.

It is essential that the Alter-Step executed on Q System is at the end of development. This guarantees that all the schemas were equal.

It is not allowed to make changes (only temporary for testing) without document it in the files.

It is better to use ALTER statement than DROP and CREATE.

### DOC41WEB\_MGRSchema.sql

These file documents the current schema on Q.

It is created with help of the Toad Professional Tool.

e.g.

--

-- Create Schema Script

-- Database Version : 11.2.0.3.0

-- Database Compatible Level : 11.2.0.3.0

-- Toad Version : 11.5.0.56

-- DB Connect String : GBOQ

-- Schema : DOC41WEB\_MGR

-- Script Created by : DOC41WEB\_MGR

-- Script Created at : 11.11.2013 18:04:48

-- Physical Location :

-- Notes :

--

-- Object Counts:

-- Indexes: 13 Columns: 17

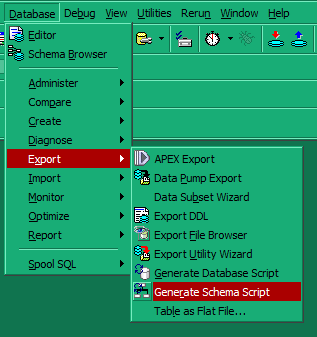
-- Object Privileges: 7

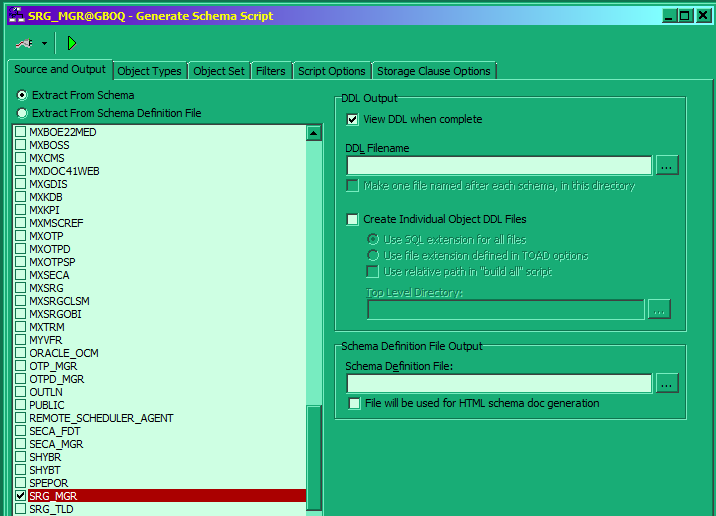
-- Tables: 7 Columns: 82 Constraints: 19

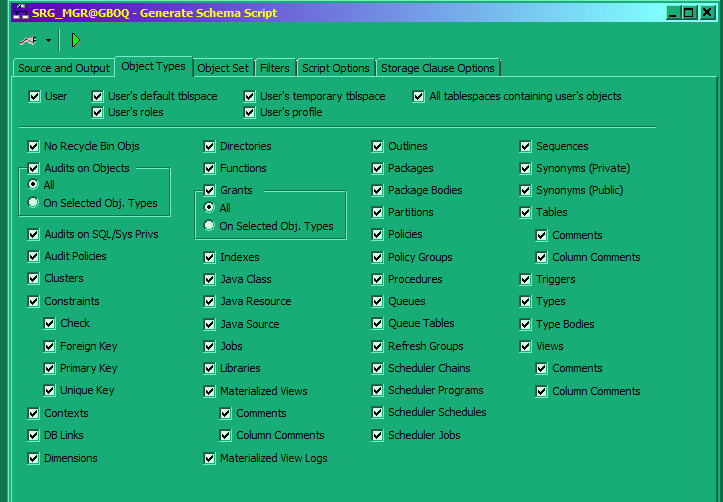
-- Triggers: 7

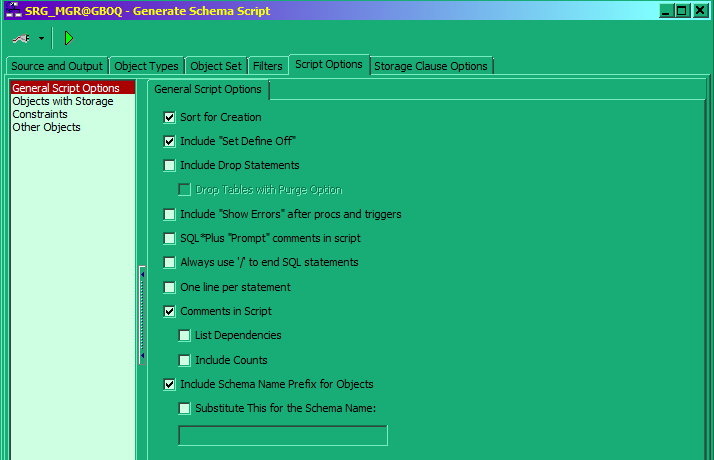
It would be created before a new version goes on production.

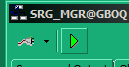
It is very easy if you can use Toad Professional Tool.



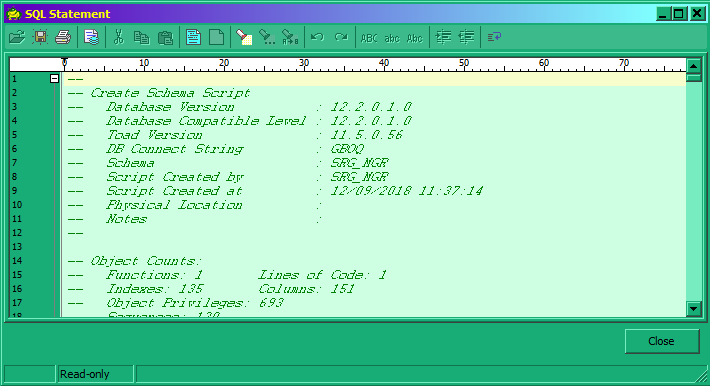




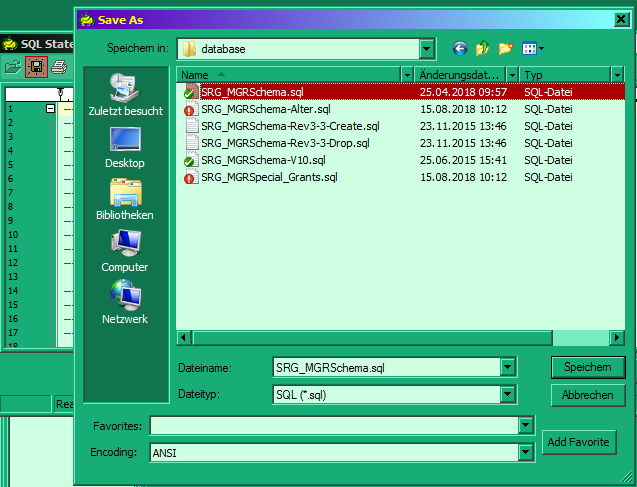




Answer always OK



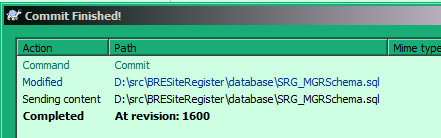
You get the file and you can save it.



After you save it, you can close all screens in Toad.

Last Step after testing is completed commits the current version in Subversion.





### DOC41WEB\_MGRSchema-Alter.sql

This is the main file to document Alter-Steps.

The current Alter-Step on a schema could be seen in the table DOC41WEB\_FDT.VERSIONS

Every new Alter-Step starts with a comment and ends with the update of this entry. Put all changes for the same release in one alter step. But only the changes for ONE release. Next release means next alter step. Sometimes a release could have more the one name.

It makes it easier the document all JIRA that changes the database for this release.

At the moment there are no alter steps.

e.g.

--------------------------------------

-- Alter-Script: CVS v1.0 -> v1.1 --

--------------------------------------

…

UPDATE DOC41WEB\_FDT.VERSIONS SET subVersion = 1 WHERE ( module = 'DOC41WEB' ) AND ( subVersion = 0 );

COMMIT WORK;

Copy the Update and the headlines and change the versions.

UPDATE DOC41WEB\_FDT.VERSIONS SET subVersion = 1 WHERE ( module = 'DOC41WEB' ) AND ( subVersion = 0 );

COMMIT WORK;

--------------------------------------

-- Alter-Script: CVS v1.1 -> v1.2 --

--------------------------------------

-- DOC41WEB-2

UPDATE DOC41WEB\_FDT.VERSIONS SET subVersion = 2 WHERE ( module = 'DOC41WEB' ) AND ( subVersion = 1 );

COMMIT WORK;

--------------------------------------

-- Alter-Script: CVS v1.2 -> v1.3 --

--------------------------------------

Now the step would be filled with the changed

This script could be copied and executed in Toad on Q.

After successful testing on Q it would be executed without changes on P.

### Folder cmn

Main files are

CMNSchema.sql

CMNSchema-Alter.sql

Both files rooted in the foundation-web repository (foundation-web/database/cmn).

If you want to update the foundation cmn part copy the current CMNSchema.sql to DOC41WEB.

Copy the new alter steps from foundation-web file to the DOC41WEB CMNSchema-Alter.sql. Change e.g. table space names to the DOC41WEB table space names and connect user name to DOC41WEB connect user name (MXDOC41WEB2).

In the subfolder init you find files for schema initialization.

### Folder usermgmt

Main files are

USERMGMTSchema.sql

USERMGMTSchema-Alter.sql

Both files rooted in the foundation-web repository (foundation-web/database/usermgmt).

If you want to update the foundation usermgmt part copy the current USERMGMTSchema.sql to DOC41WEB.

Copy the new alter steps from foundation-web file to the DOC41WEB USERMGMTSchema-Alter.sql. Change e.g. table space names to the DOC41WEB table space names and connect user name to DOC41WEB connect user name (MXDOC41WEB2).

In the subfolder init you find files for schema initialization.

### Foundation Files

In the subfolder init you find files for schema initialization.

## Table Prefix

|  |  |
| --- | --- |
| Prefix | Description |
| D41\_ | Doc for One tables |
| IM\_ | Internal monitoring tables |
| MD\_ | Master data tables |
| SAP\_ | SAP tables (data related to or from SAP) |

## Deployment on Production

After all issues of the release have been successfully tested and the documentation is up to date (The files are committed on Subversion) the release can be deployed on production.

The deployment should be done latest to the planned release date.

## Translations

In DOC41WEB the translations could be maintained on QA stage.

### CMNTranslationDepoly\_DOC41WEB\_QA\_DEV.chn

All translations would be copied from QA stage to D stage.

Translations on D stage which are not on the QA stage would be deleted.

After all, the same translations should be on both stages.

### CMNTranslationDepoly\_DOC41WEB\_QA\_PROD.chn

All translations would be copied from QA stage to prod stage.

Translations on prod stage which are not on the QA stage would be deleted.

After all, the same translations should be on both stages.

### CMNTranslationDepolyonly\_Initial\_DOC41WEB\_DEV\_QA.chn

**Only for initial load - because QA is the leading system.**

The NEW\_DEVELOPMENT like in BOE isn’t implemented yet.

All translations would be copied from D stage to QA stage.

Translations on QA stage which are not on the D stage would be deleted.

After all, the same translations should be on both stages.

## Displaytexts

In DOC41WEB the displaytexts could be maintained in ../database/cmn/DISPLAYTEXT\_ADD.csv.

Run ../chains/resources/chain&DOC41WEB/dba/init\_add.chn to insert new displaytexts values. Update isn’t possible on this way.