Visma Case

DATABASE DESCRIPTION

VISMA IMS

VISMA | IMS

Databasebeskrivelse Visma Case

September 2021

Visma IMS

Tlf.: +45 3174 0009

E-mail: booking@ims.dk

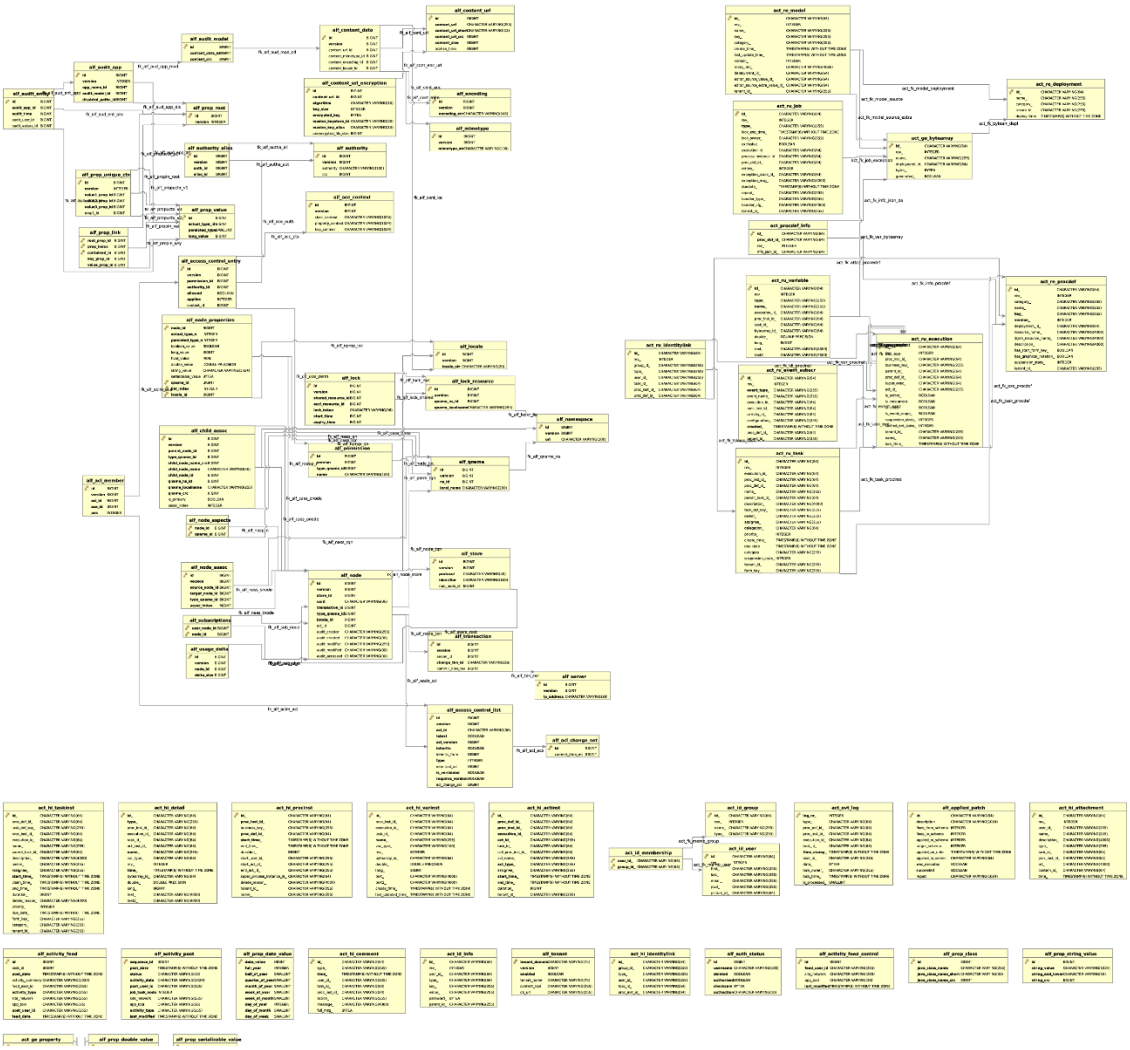
Web: www.ims.dk

Indhold

Database E/R diagram

Useful links:

http://www.lionsgatesoft.com/understanding-alfresco-content-data-model/



The information is based on Alfresco Community Edition 5.2 with meta database on PostgreSQL 9.4.12.

There are 71 entities in the data model.

The data model consists of the following modules (with main tables):

Core Content Module

alf_namespace

alf_qname

alf_node

alf_node_properties

alf_node_assoc

alf_node_aspects

alf_child_assoc

alf_store

alf_transaction

alf_subscription

alf_locale

Access Control Module

alf_access_control_entry

alf_authority

alf_authority_alias

alf_ace_context

alf_permission

alf_acl_member

alf_access_control_list

Auditing and Content Data Module

alf_audit_model

alf_audit_app

alf_audit_entry

alf_content_data

alf_content_url_encryption

alf_content_url

alf_encoding

alf_mimetype

Activity / Workflow Module

act_ru_execution

act_ru_task

act_ru_variable

act_ru_event_subscr

act_ru_identitylink

act_ru_job

act_re_model

act_re_procdef

act_re_deployment

Notifications module

alf_activity_feed

alf_activity_post

Patches module

alf_applied_patch

Multi-tenancy module

alf_tenant

Other Orphan Tables

act_hi_taskinst

act_hi_detail

act_hi_procinst

act_hi_varinst

act_hi_actinst

act_hi_attachment

act_id_membership

act_id_group

act_id_user

alf_auth_status

Database schema description

This page provides description of some important database tables.

For the reference you can use the **Alfresco for Administrators** book by V. Pal, Packt Publishing

Schema of the alf_namespace table

This table stores information of all the namespaces used for content model in the Alfresco repository.

- id: This is the unique database ID for each namespace. This ID is referenced in other tables
- uri: This the namespace value in URI format

Schema of the alf_qname table

This table stores information of all the qualified names (qnames) used for content model in the Alfresco repository. The qualified names reference node types, aspects and properties as well as association types.
Each qualified name consists of a namespace and a name.

- id: This is the unique database ID for each qualified name. This ID is reference in other tables.
- ns_id: This ID refers to the unique ID from the alf_namespace table. This ID signifies to which namespace this qualified name belongs to.
- local_name: This is the name value for the qualified name

Schema of the alf_node table

This table is the main primary reference of the node. All the node entries are present in this table.

It contains primary node information like UUID, created date, modified date, creator, transaction ID , permission information, and so on.

Here are the details about some of the columns of this table.

- id: This is the unique database ID for each node in Alfresco. This same ID is being referenced in other tables
- store_id: This ID refers to the unique ID from the alf_store table. This ID signifies which store this node belongs to, like workspacesStore, archivespacesStore, or usersStore, and so on.
- uuid: This is the unique node ID which is used in all the services and interfaces that refer to this node. This UUID never changes for a node as long as a node is present in the Alfresco system.
- transaction_id: This refers to the ID column in the alf_transaction table. Any write operation in Alfresco is considered to be processed within a transaction. Every node is part of a transaction. If you do a batch operation, there is a possibility that all nodes will have the same transaction ID. While indexing in Alfresco, based on these transaction IDs, all nodes are fetched from the database and indexed in Solr. So, this transaction ID is also a very important column.
- type_qname_id: This column is the foreign key reference ID from the alf_ qname table. It defines the type of node like content, folder, person, and so on.
- acl_id: This ID refers to the unique ID in the alf_access_control_list table related to permission. Permission in Alfresco can be considered a map; all this information is stored in a different table structure. acl_id will lead to all permissions available on this node.
- auditor_creator: This column captures information about the user who created this node.
- auditor_created: This column captures the time this node was created in the system.
- auditor_modified: This column captures the time this node was modified.

Schema of the alf_node_properties table

This table stores metadata information about the node like its name, description, and content store path ID.

All the node properties are stored in this table.

Here is the list of a few important columns in this table:

- node_id: This column refers to the unique database ID of the node defined in the alf_node table.
- actual_type_n: This column defines the property type.
- boolean_value: This column stores the values of any Boolean value metadata of node.
- string_value: This column stores the values of any string valued metadata of the node.
- long_value: If the metadata value is referring to some other table, the ID of that table will be in this column. For example, the binary file location of an asset is captured in different tables. The ID referring to that entry is stored in this column.
- qname_id: This column stores the namespace and property name referring to its ID in the alf_qname table.

Schema of the alf_child_assoc table

This table stores all the parent-child association information. With this table, you can identify the child and parent of any node.

- parent_node_id: This column stores the unique node database ID referring to the ID column of the alf_node table. As the name suggests, this would represent the parent node ID.
- child_node_id: This column stores the unique node database ID referring to the ID column of the alf_node table. This provides the node information of the child under the associated parent node ID.
- child_node_name: This column stores the names of the children nodes.
- type_qname_id: This column specifies the type of the association.
- qname_ns_id: This column refers to the ID column of the alf_namespace table.
- is_primary: This column indicates whether the parent association is primary for the underlying child node.

Schema of the alf_node_assoc table

This table stores all the peer-to-peer association information.

- source_node_id: This column stores the unique node database ID referring to the ID column of the alf_node table. This column represents the source node ID.
- target_node_id: This column stores the unique node database ID referring to the ID column of the alf_node table. This provides the node information of the target node ID.
- type_qname_id: This column specifies the type of the association.

Schema of the alf_content_data table

This table stores the mapping of node and binary file location.

All the required information related to binary content is stored in this table.

- id: This column stores the unique row ID; this ID is being used to map the node and binary content. The alf_node_properties file defining the cm: content property will have this unique ID.
- content_url_id: This refers to the ID in the alf_content_url table. This ID will lead to the actual path of content in the filesystem.

- content_mimetype_id: This column specifies the mime-type of the content, which will be required for read-write operations on content.
- content_encoding_id: This column specifies the encoding of content.

Schema of the alf_content_url table

This table has the actual filesytem path information about the binary file.

Here is the information about each of the columns:

- id: This column stores the unique row ID. This same ID is a reference point to fetch the URL information.
- content_url: This contains the actual filesystem location of the binary file under the contentstore directory. A sample entry would be store://2015/2/10/20/30/xxxx-xxxx-xxx.bin.
- content_size: This stores the specified size of the content.
- orphan_time: This column is very important. When content is permanently deleted from the Alfresco system, this column will have orphan time. Based on this time, the cleanup process cleans up all the orphaned content from the file system.

Schema of the alf_node_aspects table

This table contains information of the node aspects.

- node_id: This column stores the unique node database ID referring to the ID column of the alf_node table.
- qname_id: This column stores the namespace and aspect name referring to its ID in the alf_qname table.