

# Alfresco Repository

Database



# Introduction

- The Alfresco database is normally something that you should not directly manipulate
- But, sometimes it is good to know a little bit about it when you need to do debugging and track down problems
- The database is usually created in a similar way to:
  - create database **alfresco** default character set utf8 collate utf8\_bin;
  - grant all on alfresco.\* to 'alfresco'@'localhost' identified by '**alfresco**' with grant option;
  - grant all on alfresco.\* to 'alfresco'@'localhost.localdomain' identified by 'alfresco' with grant option;
- The tables in the DB is created the first time you start the server

# Important Tables: ALF\_NODE

- This is the parent table for node metadata and many other tables refer to it with a foreign key
- Listing a couple rows from it looks like this:

[illegible]

# Important Tables: ALF\_NODE\_PROPERTIES

- This table contains all the properties that have been set as metadata for a particular node
- Listing the properties for a node can be done with the following SQL:
  - `select * from alf_node_properties where node_id=<id>`

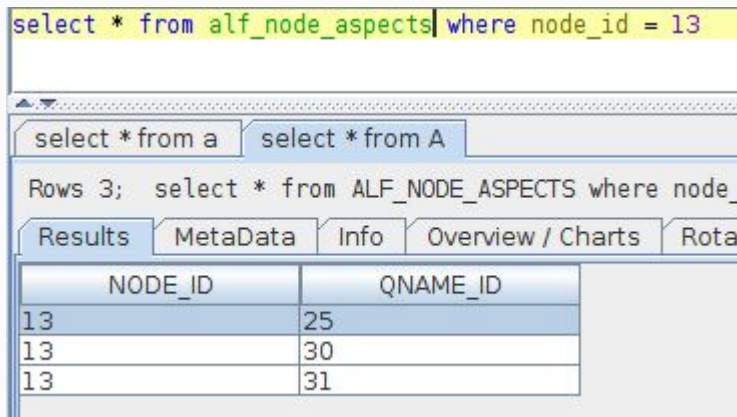
```
select * from alf_node_properties where node_id = 13
```

select * from a											
Rows 4; select * from alf_node_properties where node_id = 13											
Results MetaData Info Overview / Charts Rotated table											
NODE_ID	ACTUAL_TYPE_N	PERSISTED_TYP...	BOOLEA...	LONG...	FLOAT...	DOUBLE...	STRING_VALUE	SERIALIZABLE_VALUE	QNAME_ID	LIST_INDEX	LOCALE_ID
13	6	6	false	0	0	0	Company Home	<null>	26	-1	1
13	6	6	false	0	0	0	The company root space	<null>	27	-1	2
13	6	6	false	0	0	0	Company Home	<null>	28	-1	2
13	6	6	false	0	0	0	space-icon-default	<null>	29	-1	1

*Note that audit properties are not listed (i.e. creator, created etc), they are in ALF\_NODE*

# Important Tables: ALF\_NODE\_ASPECTS

- This table contains all aspects that have been applied to a node
- Listing the aspects for a node can be done with the following SQL:
  - `select * from alf_node_aspects where node_id=<id>`



The screenshot shows a database query tool interface. At the top, a SQL query is entered: `select * from alf_node_aspects where node_id = 13`. Below the query, there are tabs for 'select \* from a', 'select \* from A', and 'Rows 3; select \* from ALF\_NODE\_ASPECTS where node\_'. The 'Results' tab is selected, displaying a table with two columns: 'NODE\_ID' and 'QNAME\_ID'. The table contains three rows of data.

NODE_ID	QNAME_ID
13	25
13	30
13	31

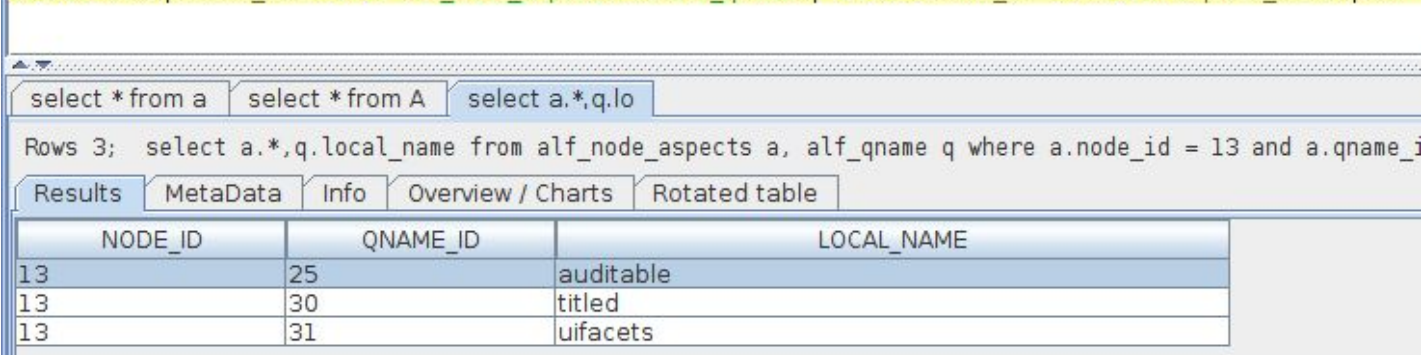
- This is not very helpful as we cannot see the names of the aspects, just their QNames
- We need to join with the ALF\_QNAME table to see the names

# Important Tables: ALF\_QNAME

- This table contains all the QName definitions and it is referred to from lots of the other tables
- For example, here is how to use it together the ALF\_NODE\_ASPECT table:

○ `select a.*,q.local_name from alf_node_aspects a, alf_qname q`

`select a.*,q.local_name from alf_node_aspects a, alf_qname q where a.node_id = 13 and a.qname_id = q.id`



select \* from a   select \* from A   select a.\*,q.lo

Rows 3; select a.\*,q.local\_name from alf\_node\_aspects a, alf\_qname q where a.node\_id = 13 and a.qname\_id = q.id

Results   MetaData   Info   Overview / Charts   Rotated table

NODE_ID	QNAME_ID	LOCAL_NAME
13	25	auditable
13	30	titled
13	31	uifacets

# Important Tables: ALF\_APPLIED\_PATCH

- This table contains information about all executed patches
- It keeps information about if they were successful or not and any error messages:

	ID	DESCRIPTION	FIXES_FROM_SC...	FIXES_TO_SCH...	A
ACT_RU_TASK	patch.db-V4.0-Activiti-task-id-indexes	patch.schemaUpgradeScript.description	0	6003	-1
ACT_RU_VARIABLE	patch.db-V4.1-increase-column-size-activiti	patch.increaseColumnSizeActiviti.description	0	6012	-1
ALF_ACCESS_CONTR...	patch.db-V4.1-remove-column-activiti	patch.removeColumnActiviti.description	0	6017	-1
ALF_ACCESS_CONTR...	patch.db-V4.2-upgrade-to-activiti-5.10	patch.upgradeToActiviti5-10.description	0	6018	-1
ALF_ACE_CONTEXT	patch.db-V4.1-rename-constraint-activiti	patch.renameConstraintActiviti.description	0	6020	-1
ALF_ACL_CHANGE_SE...	patch.db-V4.2-add-activiti-index-historic-activity	patch.addActivitiIndexHistoricActivity.description	0	6021	-1
ALF_ACL_MEMBER	patch.db-V4.2-upgrade-to-activiti-5.11	patch.upgradeToActiviti5-11.description	0	6022	-1
ALF_ACTIVITY_FEED	patch.db-V4.2-remove-old-index-act	patch.db-V4.2-remove-old-index-act.description	0	6027	-1
ALF_ACTIVITY_FEED...	patch.db-V4.2-upgrade-to-activiti-5.13	patch.upgradeToActiviti5-13.description	0	6028	-1
ALF_ACTIVITY_POST	patch.db-V5.0-upgrade-to-activiti-5.16.2	patch.db-V5.0-upgrade-to-activiti-5.16.2.description	0	8003	-1
ALF_APPLIED_PATCH	patch.db-V5.0-upgrade-to-activiti-5.16.4	patch.db-V5.0-upgrade-to-activiti-5.16.4.description	0	8008	-1
ALF_AUDIT_APP	patch.db-V5.0-remove-columns-after-upgrade-to-activ...	patch.db-V5.0-remove-columns-after-upgrade-to-activ...	0	9002	-1
ALF_AUDIT_ENTRY	patch.db-V5.0-update-activiti-default-timestamp-colu...	patch.db-V5.0-update-activiti-default-timestamp-colu...	0	9012	-1
ALF_AUDIT_MODEL	patch.db-V5.1-upgrade-to-activiti-5.19.0	patch.db-V5.1-upgrade-to-activiti-5.19.0	0	9013	-1
ALF_AUTHORITY	patch.db-V3.4-UsageTables	Manually executed script upgrade V3.4: Usage Tables	0	113	-1
ALF_AUTHORITY_ALIA	patch.db-V4.0-TenantTables	Manually executed script upgrade V4.0: Tenant Tables	0	6004	-1
ALF_AUTH_STATUS	patch.db-V3.4-jBPM-varinst-indexes	Manually executed script upgrade to add FK indexes f...	0	6016	-1
ALF_CHILD_ASSOC	patch.db-V4.1-AuthorizationTables	Manually executed script upgrade V4.1: Authorization...	0	6075	-1
ALF_CONTENT_DATA					

# Useful Queries: number of nodes (type)

- *Querying for number of nodes of a certain type*
- Let's say you wanted to find out how many documents have been stored in the repository so far
- Then you could use that with the following query:
  - `SELECT count (*)`
  - `FROM ALF_NODE n, ALF_QNAME q`
  - `WHERE n.TYPE_QNAME_ID=q.ID`
  - `AND q.LOCAL_NAME='content';`
- This query will search for all content nodes that have the type `cm:content` applied and count them



# Useful Queries: number of nodes (aspect)

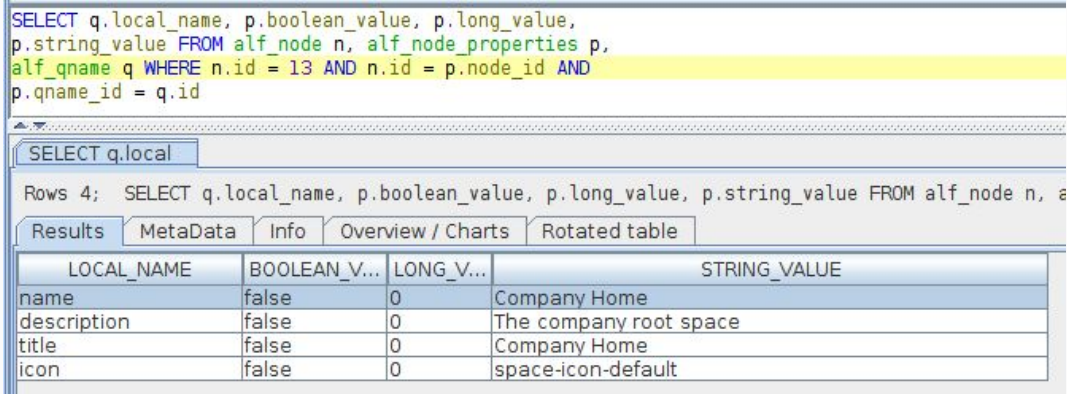
- *Querying for number of nodes with a certain aspect applied*
- Let's say you wanted to find out how many documents are currently being versioned in the repository
- Then you could use that with the following query:
  - `SELECT count(*)`
  - `FROM ALF_NODE n, ALF_NODE_ASPECTS a, ALF_QNAME q`
  - `WHERE n.ID = a.NODE_ID AND a.QNAME_ID=q.ID AND`
  - `q.LOCAL_NAME='versionable';`
- This query will search for all content nodes that have the aspect `cm:versionable` applied and count them

# Useful Queries: number of nodes (month)

- *Querying for number of nodes created in a particular month*
- If you wanted to build on the previous query and find out how many versioned documents nodes that have been created a particular month
- Then you could do that as follows:
  - `SELECT count(*) from ALF_NODE n, ALF_NODE_ASPECTS a,`
  - `ALF_QNAME q where n.ID = a.NODE_ID and a.QNAME_ID=q.ID and`
  - `q.LOCAL_NAME='versionable' and n.audit_created like`
  - `'2017-02%';`
- This will query for the number of versioned document nodes that have been stored in February 2017

# Useful Queries: properties for node

- Get the number of properties and their local names for a node:
  - `SELECT q.local_name, p.boolean_value, p.long_value, p.string_value  
FROM alf_node n, alf_node_properties p, alf_qname q  
WHERE n.id = <nodeid> AND n.id = p.node_id AND p.qname_id = q.id`



The screenshot shows a database query tool interface. At the top, a SQL query is entered in a text area. Below the query, a tabbed interface shows the 'Results' tab selected. The results are displayed in a table with four columns: LOCAL\_NAME, BOOLEAN\_VALUE, LONG\_VALUE, and STRING\_VALUE. The table contains four rows of data, representing properties of a node with ID 13. The properties are 'name', 'description', 'title', and 'icon'. The 'name' property has a boolean value of false and a string value of 'Company Home'. The 'description' property has a boolean value of false and a string value of 'The company root space'. The 'title' property has a boolean value of false and a string value of 'Company Home'. The 'icon' property has a boolean value of false and a string value of 'space-icon-default'.

```
SELECT q.local_name, p.boolean_value, p.long_value,  
p.string_value FROM alf_node n, alf_node_properties p,  
alf_qname q WHERE n.id = 13 AND n.id = p.node_id AND  
p.qname_id = q.id
```

Rows 4; SELECT q.local\_name, p.boolean\_value, p.long\_value, p.string\_value FROM alf\_node n, a

LOCAL_NAME	BOOLEAN_V...	LONG_V...	STRING_VALUE
name	false	0	Company Home
description	false	0	The company root space
title	false	0	Company Home
icon	false	0	space-icon-default