|  |  |
| --- | --- |
|  | The authorizationStatus field is not a modeled property and therefore is not indexed in InsightEngine. There is no easy way to force the indexing of this particular field as its not quite a ‘normal’ property and it just gets ignored even if we try to add it to a custom model - it stops being residual but will still not be indexed.  As an alternative and as the main goal is to have stats on user authorisation through Zeppelin I suggest a different approach:   * Configure a new jdbc interpreter in Zeppelin connecting it directly to the database (in my example below I named my interpreter pgalf * Configure your note to also use the new (pgalf) interpreter * Add the queries   I prepared and tested 2 queries that you can use and tweak for what they want that uses not only the authorizedStatus but also the enabled, locked and deleted attributes a user can have:   1. Lists all users and their status on each attribute:   %pgalf  select  p\_u.string\_value as username,  p\_e.boolean\_value as enabled,  p\_l.boolean\_value as accountLocked,  (case when auth.deleted IS NULL then false else auth.deleted end) as deleted,  (case when auth.authorized IS NULL then 'NEVER\_AUTHORIZED'  when auth.authorized = true then 'AUTHORIZED'  else 'DEAUTHORIZED' end) as authorizedStatus  from  alf\_node\_properties p\_u inner join alf\_qname q\_u on (q\_u.id=p\_u.qname\_id and q\_u.local\_name='username')  inner join alf\_node\_properties p\_e on (p\_e.node\_id=p\_u.node\_id) inner join alf\_qname q\_e on (q\_e.id=p\_e.qname\_id and q\_e.local\_name='enabled')  inner join alf\_node\_properties p\_l on (p\_l.node\_id=p\_u.node\_id) inner join alf\_qname q\_l on (q\_l.id=p\_l.qname\_id and q\_l.local\_name='accountLocked')  left join alf\_auth\_status auth on (p\_u.string\_value=auth.username)  inner join  (  select n.id from alf\_node n  inner join alf\_store s on (s.id=n.store\_id and s.protocol='user' and s.identifier='alfrescoUserStore')  inner join alf\_qname q on (q.id=n.type\_qname\_id and q.local\_name='user')  ) as user\_nodes on (p\_u.node\_id=user\_nodes.id)  2. Aggregate query that gives the overall stats per attribute:  %pgalf  select  count(username) as total\_users,  count(case when enabled=true then 1 end) as enabled\_users,  count(case when accountLocked=true then 1 end) as locked\_users,  count(case when deleted=true then 1 end) as deleted\_users,  count(case when authorizedStatus='NEVER\_AUTHORIZED' then 1 end) as never\_authorized\_users,  count(case when authorizedStatus='AUTHORIZED' then 1 end) as authorized\_users,  count(case when authorizedStatus='DEAUTHORIZED' then 1 end) as deauthorized\_users  from  (  select  p\_u.string\_value as username,  p\_e.boolean\_value as enabled,  p\_l.boolean\_value as accountLocked,  (case when auth.deleted IS NULL then false else auth.deleted end) as deleted,  (case when auth.authorized IS NULL then 'NEVER\_AUTHORIZED'  when auth.authorized = true then 'AUTHORIZED'  else 'DEAUTHORIZED' end) as authorizedStatus  from  alf\_node\_properties p\_u inner join alf\_qname q\_u on (q\_u.id=p\_u.qname\_id and q\_u.local\_name='username')  inner join alf\_node\_properties p\_e on (p\_e.node\_id=p\_u.node\_id) inner join alf\_qname q\_e on (q\_e.id=p\_e.qname\_id and q\_e.local\_name='enabled')  inner join alf\_node\_properties p\_l on (p\_l.node\_id=p\_u.node\_id) inner join alf\_qname q\_l on (q\_l.id=p\_l.qname\_id and q\_l.local\_name='accountLocked')  left join alf\_auth\_status auth on (p\_u.string\_value=auth.username)  inner join  (  select n.id from alf\_node n  inner join alf\_store s on (s.id=n.store\_id and s.protocol='user' and s.identifier='alfrescoUserStore')  inner join alf\_qname q on (q.id=n.type\_qname\_id and q.local\_name='user')  ) as user\_nodes on (p\_u.node\_id=user\_nodes.id)  ) users\_auth\_status  The end result looks like this:  Attached file  Also a quick note as I don’t know the end goal of checking the authorisation status, but the authorizationStatus is tied to ‘which users are authorised to be logged in within the alfresco enterprise licensed number of users’. On the other hand, the enabled field will tell you if the user has been deactivated by an admin or not - not sure which one the customer was aiming for so that’s why I included both in the queries |