

Le but l'exercice est d'automatiser automatiquement le tri d'une liste quelconque, et que la modification du tri d'un des enregistrements recalcule le tri des autres enregistrements

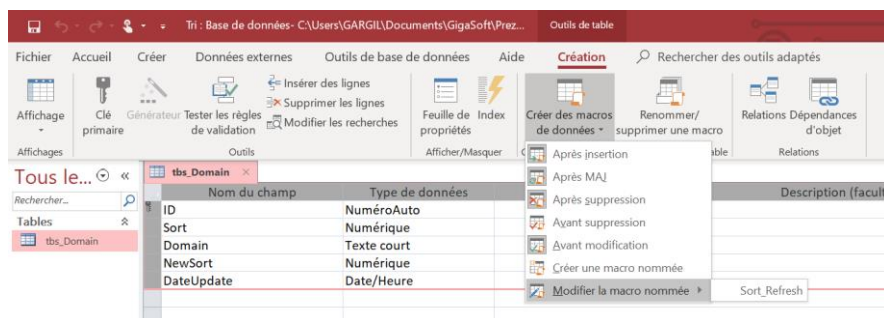
Par exemple, je dispose de cette liste (trié 1,2,3,4,5..)

Sort	Domain
1	Système
2	Organisation
3	Référentiel
4	Article
5	Partenaire
6	Finance
7	Commercial
8	To Be Defined

Si je modifie le tri de la 4^{ème} ligne de 4 en 2, l'ordre sera automatiquement recalculé

Sort	Domain
1	Système
3	Organisation
4	Référentiel
2	Article
5	Partenaire
6	Finance
7	Commercial
8	To Be Defined

Structure de la Table



La table doit comprendre les champs suivants

- ID : un champ avec un numéro automatique
- Domain: champ texte informatif de l'enregistrement
- Sort: entier long utilisé pour trier les enregistrements, défaut =0
- NewSort: entier long, défaut=0 ; un champ de calcul intermédiaire du nouveau tri
- DateUpdate : un champ qui se met à jour à chaque modification (sauf si le nouveau tri est mis à jour), c'est-à-dire un champ qui trace quel est le dernier enregistrement modifié

La table doit comprendre les macro de données suivantes :

<p>Après insertion : pour relancer le tri si le champ [Sort] est différent de Zéro, sinon, il faut affecter à ce champ la valeur du tri maximal +1</p>	<pre> Groupe : NewSort or SetSort Si [Sort]<>0 Alors ExécuterMacroDonnées Nom de macro tbs_Domain.Sort_Refresh Paramètres prmGoDown = [Sort]>MaxDom("Sort";"tbs_Domain";"ID<>" & [ID]) prmSortMin = [Sort] Sinon Rechercher un enregistrement dans tbs_Domain Condition Where = [MyTarget].[ID]=[tbs_Domain].[ID] Alias MyTarget ModifierEnregistrement Alias MyTarget DéfinirChamp Nom MyTarget.Sort Valeur = MaxDom("Sort";"tbs_Domain")+1 Terminer ModifierEnregistrement Fin Si Groupe Fin </pre>
<p>Après MAJ: pour recalculer le tri si le champ [Sort] a été mis à jour par l'utilisateur</p>	<pre> Groupe : NewSort Si Updated("Sort") Et Pas Updated("NewSort") Alors ExécuterMacroDonnées Nom de macro tbs_Domain.Sort_Refresh Paramètres prmGoDown = [Sort]>[Ancien].[Sort] prmSortMin = VraiFaux([Sort]>[Ancien].[Sort];[Ancien].[Sort];[Sort]) Fin Si Groupe Fin </pre>
<p>Après Suppression: pour relancer lancer le tri</p>	<pre> ExécuterMacroDonnées Nom de macro tbs_Domain.Sort_Refresh Paramètres prmGoDown = Vrai prmSortMin = [Ancien].[Sort] </pre>
<p>Avant Modification : pour mettre à jour le champ DateUpdate à chaque modification</p>	<pre> Groupe : Update if NewSort not changed Si Pas Updated("NewSort") Alors DéfinirChamp Nom DateUpdate Valeur = Maintenant() Fin Si Groupe Fin </pre>
<p>Sort_Refresh : la fonction centrale de recalcul du tri</p>	

<p>La fonction dispose de 2 arguments</p>	<table border="1"> <thead> <tr> <th colspan="2">Paramètres</th> </tr> <tr> <th>Nom</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>prmGoDown</td> <td>Description</td> </tr> <tr> <td>prmSortMin</td> <td>Description</td> </tr> </tbody> </table>	Paramètres		Nom	Description	prmGoDown	Description	prmSortMin	Description
Paramètres									
Nom	Description								
prmGoDown	Description								
prmSortMin	Description								
<p>Dans un premier temps on initialise un compteur pour chaque enregistrement (Loop1) on compte le nombre d'enregistrements (Loop2) au tri inférieur</p>	<pre> Pour chaque enregistrement dans tbs_Domain Condition Where = [Loop1].[Sort] >=[prmSortMin] Alias Loop1 DéfinirVarLocale Nom MyCounter Expression =0 Pour chaque enregistrement dans tbs_Domain Condition Where = [Loop2].[Sort] <[Loop1].[Sort] Alias Loop2 DéfinirVarLocale Nom MyCounter Expression =[MyCounter]+1 </pre>								
<p>Dans un second temps on incrémente le compteur (de Loop1) pour les enregistrements qui ont le même tri (Loop3) en prenant en compte si on a fait descendre ou monter le tri initial</p>	<pre> Si [prmGoDown] Alors Pour chaque enregistrement dans tbs_Domain Condition Where = [Loop3].[Sort]=[Loop1].[Sort] Et [Loop3].[DateUpdate] <=[Loop1].[DateUpdate] Alias Loop3 DéfinirVarLocale Nom MyCounter Expression =[MyCounter]+1 Sinon Pour chaque enregistrement dans tbs_Domain Condition Where = [Loop3].[Sort]=[Loop1].[Sort] Et [Loop3].[DateUpdate] >=[Loop1].[DateUpdate] Alias Loop3 DéfinirVarLocale Nom MyCounter Expression =[MyCounter]+1 </pre>								
<p>Enfin, on stocke la valeur du nouveau tri dans le champ temporaire</p>	<pre> ModifierEnregistrement Alias Loop1 DéfinirChamp Nom Loop1.NewSort Valeur =[Mycounter] Terminer ModifierEnregistrement </pre>								
<p>Ensuite on met à jour le tri et on réinitialise le champ de tri temporaire</p>	<pre> Pour chaque enregistrement dans tbs_Domain Condition Where = [MyTarget].[Sort] >=[prmSortMin] Alias MyTarget ModifierEnregistrement Alias MyTarget DéfinirChamp Nom MyTarget.Sort Valeur =[MyTarget].[NewSort] DéfinirChamp Nom MyTarget.NewSort Valeur =0 Terminer ModifierEnregistrement </pre>								