w.c.s. - Development #77302

backoffice: tableau de traitement & filtre, ne pas se baser sur les valeurs trouvées dans la table pour les champs item branchés sur un modèle de fiche

04 mai 2023 18:01 - Lauréline Guérin

Statut:	Fermé	**Début:**	04 mai 2023
Priorité:	Normal	**Echéance:**	
Assigné à:	Lauréline Guérin	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Un model de fiche A, avec des fiches,
Un model de fiche B, avec un champ item a qui pointe sur A, pas de fiches, table vide.

Aller sur le tableau de traitement, ajouter un filtre sur la colonne a, dans le select il n'y a aucune valeur, parce qu'on se base sur les valeurs trouvées dans la table b:

```
            options = self.formdef.data_class().select_distinct(
                [sql.get_field_id(filter_field), '%s_display' % sql.get_field_id(filter_figure)],
                clause=options_criterias,
            )
```

On pourrait, dans ce cas, renvoyer toutes les fiches A, peu importe ce qu'on trouve dans la table b.
(Peut-être seulement dans le mode autocomplete, pour éviter de surcharger la page avec 15000 options dans un select ?)

Demandes liées:

Lié à w.c.s. - Bug #78441: backoffice: tableau de traitement custom view avec...	**Fermé**	**13 juin 2023**

Révisions associées

Révision 66c154ad - 12 juin 2023 16:07 - Lauréline Guérin

backoffice: remove obsolete code (#77302)

Révision 3c4ef7cb - 12 juin 2023 16:07 - Lauréline Guérin

backoffice: custom view datasource, return all options (#77302)

Révision f9502ca0 - 12 juin 2023 16:07 - Lauréline Guérin

backoffice: fix display value of autocomplete item criteria (#77302)

Historique

#2 - 09 mai 2023 10:39 - Lauréline Guérin

Je suppose qu'à l'époque de l'introduction des tableaux de traitement et des filtres sur ceux-ci, il était pratique d'avoir dans les select les valeurs possibles, trouvées dans la table, pour éviter d'être tenté de filtrer sur une valeur inutilisées, ce qui aurait donné un tableau de traitement vide.

Mais avec l'apparition des vues personnalisées, et leur utilisation en tant que source de données, il devient impossible de créer une vue personnalisée avec une valeur qui existe dans la liste des items, mais qui n'est pas encore utilisée et donc non trouvée dans la table.
Or, on veut pouvoir définir des vues personnalisées en tout début de projet, alors que la table est vide.

Je pense qu'il n'est plus pertinent de se restreindre aux valeurs trouvées dans la table, encore plus avec l'apparition des opérateurs.
Exemple: on a les valeurs 1, 2, 3 utilisées, mais on veut un filtre != 4; impossible à définir, 4 n'étant pas utilisé.

Proposition:

1/ pour les champs item/items, toujours renvoyer l'ensemble des valeurs, quelle que soit leur utilisation (que le champ soit branché sur un modèle de fiche ou non)
2/ ne plus se baser sur le paramétrage du field autocomplete/liste pour avoir un select2 ou pas, mais à la place:
a/ si moins de 15 valeurs, select
b/ si plus de 15 valeurs, select2 (qui renvoie un max de 15 valeurs)
ceci pour éviter de surcharger la page avec un select contenant 15000 options.

#3 - 09 mai 2023 10:56 - Frédéric Péters

> 1/ pour les champs item/items, toujours renvoyer l'ensemble des valeurs, quelle que soit leur utilisation (que le champ soit branché sur un modèle de fiche ou non)

Pour revenir à l'historique de pourquoi taper dans la base plutôt que prendre les options de la source de données, c'est #35703, "par exemple une source chrono avec des événements qui ne sont plus disponibles"; je pense qu'il faut garder une différence de comportement entre ce qui est fiches et source de données générique.

> Mais avec l'apparition des vues personnalisées, et leur utilisation en tant que source de données, il devient impossible de créer une vue personnalisée avec une valeur qui existe dans la liste des items, mais qui n'est pas encore utilisée et donc non trouvée dans la table.

Il y a en fait un problème par le mélange de la même vue qui peut servir pour l'agent à filtrer à la volée (où il y a une utilité à voir uniquement les choix possibles) et qui doit servir à la conception de vues personnalisées. Sans résoudre toutes les situations on pourrait au moins résoudre le cas particuler de l'utilisation en source de données, qui est un cas où la vue est clairement identifiée comme telle. ?

#4 - 16 mai 2023 13:47 - Lauréline Guérin

> résoudre le cas particuler de l'utilisation en source de données

Ok, je fais ça pour ce cas seulement

#5 - 16 mai 2023 13:49 - Robot Gitea

- Statut changé de Nouveau à En cours

- Assigné à mis à Lauréline Guérin

Lauréline Guérin (lguerin) a ouvert une pull request sur Gitea concernant cette demande :

- URL : https://git.entrouvert.org/entrouvert/wcs/pulls/320
- Titre : WIP: backoffice: custom view datasource, return all options (#77302)
- Modifications : https://git.entrouvert.org/entrouvert/wcs/pulls/320/files

#6 - 16 mai 2023 15:16 - Lauréline Guérin

note: au passage, #72055 devient obsolète

#7 - 16 mai 2023 15:56 - Robot Gitea

- Statut changé de En cours à Solution proposée

#8 - 26 mai 2023 11:32 - Robot Gitea

- Statut changé de Solution proposée à Solution validée

Frédéric Péters (fpeters) a approuvé une pull request sur Gitea concernant cette demande :

- URL : https://git.entrouvert.org/entrouvert/wcs/pulls/320

#9 - 12 juin 2023 16:07 - Robot Gitea

- Statut changé de Solution validée à Résolu (à déployer)

Lauréline Guérin (lguerin) a mergé une pull request sur Gitea concernant cette demande :

- URL : https://git.entrouvert.org/entrouvert/wcs/pulls/320
- Titre : backoffice: custom view datasource, return all options (#77302)
- Modifications : https://git.entrouvert.org/entrouvert/wcs/pulls/320/files

#10 - 12 juin 2023 19:14 - Transition automatique

- Statut changé de Résolu (à déployer) à Solution déployée

#11 - 13 juin 2023 14:19 - Lauréline Guérin

- Lié à Bug #78441: backoffice: tableau de traitement custom view avec template & select2 ajouté

#12 - 13 août 2023 04:42 - Transition automatique

Automatic expiration