

OLAP / Business Intelligence pour Publik - Development #36930

rajouter les slugs dans la table des formdefs

15 octobre 2019 11:32 - Serghei Mihai

Statut:	Fermé	Début:	15 octobre 2019
Priorité:	Normal	Echéance:	
Assigné à:	Serghei Mihai	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Pour se baser dessus dans la construction des visualisations. Cela évite de se retrouver avec une sélection des formulaires différente de celle souhaitée.			
Demandes liées:			
Lié à BiJoe - Development #36931: utiliser les slugs des formdefs dans les vi...		Fermé	15 octobre 2019

Révisions associées

Révision d11d196f - 02 mars 2020 22:07 - Serghei Mihai

feeder: preseve categories and form names order (#36930)

Historique

#4 - 15 octobre 2019 11:38 - Serghei Mihai

- Lié à Development #36931: utiliser les slugs des formdefs dans les visualisations ajouté

#5 - 15 octobre 2019 13:03 - Benjamin Dauvergne

C'est loin d'être évident, c'est exactement le même problème que [#25982](#) qui a été fermé sans ouvrir de ticket pour continuer (on pourrait prendre celui-ci en le renommant): **si on change les valeurs de filtrage on va casser toutes les visualisations actuelles qui utilisent ces valeurs.**

Donc oui mais il faut traiter le cas général: utiliser un identifiant unique stable pour les énumérations (formdef, valeur d'itemfield, etc..) et ça ne concerne pas uniquement wcs-olap mais ça nécessite une migration de donnée coté bijoe vraiment compliquée à faire en commun avec une évolution dans wcs-olap, dont je parlais dans [#25982](#) :

- faire évoluer wcs-olap pour avoir cette valeur stable et l'utiliser partout à la place des id (ça veut dire plus de dimension comme

```
{
  'name': 'formdef',
  'label': 'formulaire',
  'join': ['formdef'],
  'type': 'integer',
  'value': 'formdef.id',
  'value_label': 'formdef.label',
  'order_by': 'formdef.label',
},
```

où on a 'value' == '<jointure>.id', mais plutôt :

```
{
  'name': 'formdef',
  'label': 'formulaire',
  'join': ['formdef'],
  'type': 'integer',
  'value': 'formdef.slug',
  'value_label': 'formdef.label',
  'order_by': 'formdef.label',
},
```

ça restera bancal en cas de changement de slug mais c'est quand même moins fréquent (et si on expose les visu dans l'admin Django on pourra toujours aller corriger assez simplement)

- ensuite faire une migration coté bijoe qui aurait cette tête (et je ne pense pas une migration Django plutôt un script) :
 - parcourir l'ensemble des visualisations,
 - trouver les valeurs des filtrages (des trucs genre @[1,2,3],
 - remplacer les id (c'est forcément des entiers) par la nouvelle valeur, via :

- prendre la dimension correspondant au filtre, parser 'value' en extraire la jointure (si pas jointure on est bon), on peut se baser sur join éventuellement (si y en a qu'une seule ça devrait aller sinon c'est plus compliqué)
- chercher la ligne <jointure>.id dans la table,
- calculer 'value' pour cette ligne, ça donne la nouvelle valeur

#6 - 15 octobre 2019 13:12 - Benjamin Dauvergne

Pour revenir au souci de Arles (je vais copier mon laïus dans un autre ticket finalement); il faut stabiliser les ids comme on l'a fait pour les items, i.e. reprendre la table précédente, et oui il faudra les slugs puisque les labels ne sont pas stables.

Ça irait bien d'avoir une nouvelle méthode create_ref_table qui produirait des tables à 3 colonnes :

- id : serial
- ref : text
- label : text

ref étant le truc stable à utiliser pour récupérer les anciens id dans la table précédente, après création on poserait le curseur de séquence sur le max(id) de l'ancienne table ; à l'insertion soit on trouve la valeur dans l'ancienne table et on conserve l'id sinon on ajoute une nouvelle ligne.

#7 - 21 octobre 2019 18:50 - Serghei Mihai

Benjamin Dauvergne a écrit :

C'est loin d'être évident, c'est exactement le même problème que [#25982](#) qui a été fermé sans ouvrir de ticket pour continuer (on pourrait prendre celui-ci en le renommant): **si on change les valeurs de filtrage on va casser toutes les visualisations actuelles qui utilisent ces valeurs.**

Elles sont déjà cassées car elles pointent vers des mauvais formulaires.

Donc rajouter un slug ou figer les id, je pense qu'on pourra zapper les migrations côté bijoe.

#8 - 21 octobre 2019 19:53 - Benjamin Dauvergne

Oui c'est peut-être cassé sur 50% des déploiements où l'ordre des formulaires a pu changer mais là ce sera cassé sur 100% des déploiements, garanti.

#9 - 25 novembre 2019 17:42 - Serghei Mihai

Premier jet avec l'ajout des méthodes qui créent des tables en y ajoutant une colonne ref et ensuite rajouter les nouvelles données.

Puis usage de ces méthodes pour stocker les categories et les formulaires. Pour les formulaires la déclaration des noms des ceux-ci et enregistrement de leurs données se font séparément.

#10 - 02 décembre 2019 09:38 - Serghei Mihai

- Fichier 0001-feeder-preseve-categories-and-form-names-order-36930.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Deuxième version avec ajout implicite de la clé primaire et de la référence. Cela facilitera la transition des relations basées sur les id à celles des références.

#11 - 23 décembre 2019 12:09 - Benjamin Dauvergne

Serghei Mihai a écrit :

Deuxième version avec ajout implicite de la clé primaire et de la référence. Cela facilitera la transition des relations basées sur les id à celles des références.

- tu pourrais ajouter deux méthode do_category_table() et do_formdef_table()
- Comme update_sequence_table_number() est factorisé, l'utiliser en fin d'insertion des valeurs dans les tables (ça n'a pas de conséquence particulière de ne pas le faire pour l'instant mais je trouve ça bien de conserver les invariants que n'importe qui espère d'une base postgres normale)
- ref devrait être unique (simplement rajouter unique après le type varchar)
- il me me semble qu'avec le changement de schéma des tables category et formdef la reprise bête et méchante à base "*" ne va pas marcher comme on veut, on aura jamais de valeur ref pour les valeurs existantes et donc on va avoir des doublons lors de la détection to_insert (ref étant nullable l'insert des vieilles valeurs dans la nouvelle table va marcher, mais ref sera nul pour toutes les lignes).

```
if self.prev_table_exists(name):
    # Insert data from previous table
    self.ex(
        'INSERT INTO {schema_temp}.{name} SELECT * FROM {schema}.{name}',
        ctx={'name': quote(name)})
```

```
)  
self.update_table_sequence_number(name)
```

Il va falloir prévoir de:

1. détecter que la table existante n'a pas de ref,
 2. faire des update spécifique dans ce cas en fonction du name
- On est en postgresql 9.6 on peut utiliser UPSERT pour se simplifier la vie (on aura automatiquement le suivi du name si jamais il change)

```
INSERT INTO category (ref, name) VALUES (...)  
ON CONFLICT (ref) DO UPDATE SET name = EXCLUDED.name;
```

et laissant la base maintenir l'id toute seule (plus besoin de faire enumerate)

#12 - 27 décembre 2019 16:31 - Serghei Mihai

- Fichier 0001-feeder-preseve-categories-and-form-names-order-36930.patch ajouté

Merci pour le tuyau UPSERT, j'ai appris une chose.
Nouveau patch avec tes remarques.

#18 - 14 janvier 2020 11:01 - Serghei Mihai

- Fichier 0001-feeder-preseve-categories-and-form-names-order-36930.patch ajouté

Nouveau patch faisant d'abord la mise à jour des données existantes se basant sur la référence.

#19 - 24 janvier 2020 12:04 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

Serghei Mihai a écrit :

Nouveau patch faisant d'abord la mise à jour des données existantes se basant sur la référence.

Ça ne va pas marcher, tu slugifies le titre du formulaire pour générer les valeurs de référence lors de la reprise de la table existante, il faut matcher sur le label, c'est la seule manière à peu près correcte de migrer l'état actuel vers l'état futur avec la colonne ref en tentant de conserver les id.

#20 - 03 février 2020 10:21 - Serghei Mihai

Je zappe exprès l'état actuel car du côté bijoe seuls les id comptent.

Il était question de passer sur toutes les instances de bijoe pour mettre à jour les id en fonction des libellés sélectionnés, mais ils sont déjà érronnées.

Donc autant partir de zéro.

#22 - 17 février 2020 10:48 - Benjamin Dauvergne

Ok mais alors vire complètement la récupération des données du schéma précédent et gère les items aussi; ça va effectivement péter 100% des visus (dans l'idée que ça sert à rien de péter à moitié maintenant pour péter à 100% plus tard).

#23 - 21 février 2020 14:59 - Serghei Mihai

- Fichier 0001-feeder-preseve-categories-and-form-names-order-36930.patch ajouté

- Statut changé de En cours à Solution proposée

Ok, récupération du schéma précédent viré.

Pas compris pour la gestion des items. Lequels?

#24 - 02 mars 2020 15:46 - Benjamin Dauvergne

Serghei Mihai a écrit :

Ok, récupération du schéma précédent viré.

Ok mais il manque les modifications au schéma (JSON) pour bijoe pour utiliser le slug ou le label comme valeur dans les filtres (dans dimensions[*].value), actuellement c'est category.id et formdef.id.

Pas compris pour la gestion des items. Lequels?

Oublions on fera un autre ticket.

#25 - 02 mars 2020 15:46 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#26 - 02 mars 2020 17:05 - Serghei Mihai

- Fichier 0001-feeder-preseve-categories-and-form-names-order-36930.patch ajouté

- Statut changé de En cours à Solution proposée

- Assigné à mis à Serghei Mihai

Traite moi de trouillard, mais je préfère d'abord passer ce patch, valider que ça tourne et ensuite attaquer les items (statuts et itemfields).

#27 - 02 mars 2020 17:08 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Ok; tu l'as fait tourner sur ton publik-devinst pour voir si bijoe te pond des <select> avec les bons identifiants désormais (tu devrais y voir les slugs des formulaires et des catégories comme <option.value>).

#28 - 02 mars 2020 22:17 - Serghei Mihai

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

Yep, en modifiant le type en string car sinon il y a tentative dans bijoe de conversion en entier.

```
commit d11d196fa887bbc97d04a052e4e4728ccf30efaf
Author: Serghei Mihai <smihai@entrouvert.com>
Date: Mon Nov 25 17:16:50 2019 +0100
```

```
feeder: preseve categories and form names order (#36930)
```

#29 - 07 mars 2020 18:15 - Frédéric Péters

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

Fichiers

0001-feeder-preseve-categories-and-form-names-order-36930.patch	7 ko	02 décembre 2019	Serghei Mihai
0001-feeder-preseve-categories-and-form-names-order-36930.patch	11,1 ko	27 décembre 2019	Serghei Mihai
0001-feeder-preseve-categories-and-form-names-order-36930.patch	11,2 ko	14 janvier 2020	Serghei Mihai
0001-feeder-preseve-categories-and-form-names-order-36930.patch	9,01 ko	21 février 2020	Serghei Mihai
0001-feeder-preseve-categories-and-form-names-order-36930.patch	10,3 ko	02 mars 2020	Serghei Mihai