

w.c.s. - Development #34179

concurrence de migrations (le migrate user_label ne fonctionne pas)

19 juin 2019 21:31 - Frédéric Péters

Statut:	Fermé	Début:	19 juin 2019
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		
Description			
Après mise à jour des recettes, sql_level est bien passé à 32 mais les tables/vues n'ont pas gagné de colonne user_label.			

Révisions associées

Révision 21040989 - 20 juin 2019 22:19 - Frédéric Péters

tests: add check for user_label migration (#34179)

Historique

#1 - 19 juin 2019 21:33 - Frédéric Péters

(mais la migration qui suit, qui ajoute la colonne last_update_time aux sessions, est bien passée)

#2 - 19 juin 2019 21:49 - Frédéric Péters

Ça semble n'avoir concerné que le SaaS de recette, sur des installations externes ça me semble tout à fait ok.

#4 - 20 juin 2019 11:28 - Thomas Noël

En première lecture, ça pourrait être dû à ça dans wcs/sql.py :

- pour l'ajout dans les formdata, on passe par : set_reindex('formdata', 'needed', conn=conn, cur=cur)
- et pour les vues, on fait : migrate_views(conn, cur)

Mais le set_reindex ne fait rien, en réalité. Il va juste lancer la migration "plus tard", au prochain cron.

Et donc, les vues ne sont pas ok. Sans doute que tripoter une formulaire va suffire à tout recréer.

#5 - 20 juin 2019 21:17 - Frédéric Péters

Mais le set_reindex ne fait rien, en réalité. Il va juste lancer la migration "plus tard", au prochain cron.

Je ne vois pas le rapport (le set_reindex est pour passer sur les formdata, ce qui n'est pas nécessaire ici); je creuserai davantage demain, en simulant les migrations dans les tests.

#6 - 21 juin 2019 09:43 - Thomas Noël

Frédéric Péters a écrit :

Mais le set_reindex ne fait rien, en réalité. Il va juste lancer la migration "plus tard", au prochain cron.

Je ne vois pas le rapport (le set_reindex est pour passer sur les formdata, ce qui n'est pas nécessaire ici); je creuserai davantage demain, en simulant les migrations dans les tests.

Effectivement ; j'ai été enduit d'erreur par le commentaire

```
...
# 31: add user_label to formdata
set_reindex('formdata', 'needed', conn=conn, cur=cur)
```

#7 - 27 juin 2019 23:15 - Frédéric Péters

- Statut changé de Nouveau à En cours

Sur les prods, sauf s'il y a eu manipulation manuelle, ça m'a l'air d'avoir fonctionné sans soucis (j'ai regardé le saas, rochefort et sictiam).

#8 - 27 juin 2019 23:18 - Frédéric Péters

Mmm, non, Thomas ?

#9 - 28 juin 2019 06:41 - Frédéric Péters

- Sujet changé de le migrate user_label ne fonctionne pas à concurrence de migrations (le migrate user_label ne fonctionne pas)

Ça n'apparaît que sur des déploiements redondants, l'idée ici c'est que :

- sur node1 la migration est jouée,
- pendant ce temps, sur node2, qui continue sa vie, une requête se trouve appeler do_formdef_tables, qui voit que le champ user_label existe "en trop", décide du coup d'appeler le redo_views,
- bardaf, c'est l'embarquée.

#10 - 28 juin 2019 09:45 - Thomas Noël

Frédéric Péters a écrit :

- bardaf, c'est l'embarquée.

Effectivement, analyse convaincante. Du coup, ça ne serait arrivé que sur user_label parce qu'on n'a pas ajouté de champ de ce type depuis la naissance des systèmes redondants ?

#11 - 28 juin 2019 09:53 - Frédéric Péters

Yep, le précédent,

```
ce94b90179 (Thomas NOEL 2018-05-01 12:22:19 +0200 2275) # 27: add last_jump_datetime in e
volutions tables
```

#12 - 21 décembre 2021 19:00 - Frédéric Péters

- Statut changé de En cours à Fermé

pendant ce temps, sur node2, qui continue sa vie, une requête se trouve appeler do_formdef_tables, qui voit que le champ user_label existe "en trop", décide du coup d'appeler le redo_views,

Ça ne peut (presque) plus avoir lieu dans la mesure où les modifications de schéma sont limitées au .store() du formdef, plus au simple accès à formdef.data_class(). ([#57118](#)).