

w.c.s. - Bug #3363

erreur SQL « column "f57_display" does not exist »

24 juillet 2013 11:51 - Thomas Noël

Statut:	Fermé	Début:	24 juillet 2013
Priorité:	Haut	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposé:			
Description			
A priori, sur le formulaire, un champ (le numéro 57) de type "liste multichoix" a été crée par erreur.			
Supprimé, puis remplacé par un champ "liste" normal.			
Et depuis :			
<pre>LINE 1: ...f55_display, f52, f22, f3, f4, f10, f11, f8, f57, f57_displa... ^) To: admin@entrouvert.com Exception: type = '<class 'psycopg2.ProgrammingError'>', value = 'column "f57_display" does not exist LINE 1: ...f55_display, f52, f22, f3, f4, f10, f11, f8, f57, f57_displa... ^ '</pre>			
Stack trace (most recent call first):			
<pre>File "/usr/lib/pymodules/python2.6/wcs/sql.py", line 230, in get_with_indexed_value 228 cls._table_name, 229 index) > 230 cur.execute(sql_statement, {'value': str(value)}) 231 objects = [] 232 while True: locals: index = 'user_id' sql_statement = 'SELECT id, user_id, user_hash, receipt_time, status, workflow_data, id_displ ay, workflow_roles, work flow_roles_array, f55, f55_display, f52, f22, f3, f4, f10, f11, f8, f57, f57_display, f9, f26, f27 , f32, f31, f30, f33, f34, f18, f36, f38, f38_display, f50, f50_display\n FROM formdata_26_demande_de_dispositif_d_assain\n WHERE user_id = %(value)s' cur = <cursor object at 0x29e6bf0; closed: 0> value = 118 ignore_errors = False x = ('workflow_roles_array', 'text[]') conn = <connection object at 0x207ca20; dsn: 'dbname=eservices host=localhost port=5432 user= auquotidien password=xxx xxxxx', closed: 0> cls = <class 'wcs.formdef.Assainissement-Demande-De-Raccordement-Au-Reseau'> File "/usr/lib/pymodules/python2.6/wcs/sql.py", line 95, in f 93 except psycopg2.Error: 94 get_connection().rollback() > 95 raise 96 return f 97 (...)</pre>			

Révisions associées

Révision 82cf4f12 - 29 juillet 2013 17:04 - Thomas Noël

sql: drop useless columns (#3363)

Historique

#1 - 24 juillet 2013 12:01 - Thomas Noël

Mon analyse...

Dans sql.py, on fait des "ALTER TABLE %s ADD COLUMN ..." si un champ n'existe plus, puis le %s_display correspondant si nécessaire.

Mais si le champ existe toujours, on ne fait rien du tout, et notamment on n'ajoute pas le xx_display s'il devait être ajouté.

C'est là :

```
139     for field in formdef.fields:
140         if 'f%s' % field.id not in existing_fields:
141             sql_type = SQL_TYPE_MAPPING.get(field.key, 'varchar')
142             if sql_type is None:
143                 continue
144             cur.execute(''ALTER TABLE %s ADD COLUMN %s %s'' % (
145                             table_name,
146                             'f%s' % field.id,
147                             sql_type))
148             if field.store_display_value:
149                 cur.execute(''ALTER TABLE %s ADD COLUMN %s varchar'' % (
150                             table_name, 'f%s_display' % field.id))
151
```

et

```
181     for field in formdef.fields:
182         if 'f%s' % field.id not in existing_fields:
183             sql_type = SQL_TYPE_MAPPING.get(field.key, 'varchar')
184             if sql_type is None:
185                 continue
186             cur.execute(''ALTER TABLE %s ADD COLUMN %s %s'' % (
187                             table_name,
188                             'f%s' % field.id,
189                             sql_type))
190             if field.store_display_value:
191                 cur.execute(''ALTER TABLE %s ADD COLUMN %s varchar'' % (
192                             table_name, 'f%s_display' % field.id))
```

#2 - 24 juillet 2013 14:40 - Thomas Noël

C'est bien ça, mais c'est même pire : il faut aussi faire un ALTER si le sql_type de fXX a été modifié.

#3 - 24 juillet 2013 15:11 - Thomas Noël

Solution en cours de codage : supprimer du SQL les champs qui ont été supprimés du formdef.

Je ne vois pas d'autre technique (alter type n'est pas garanti, et je ne trouve pas de quoi tester si le type doit être modifié pour forcer un drop+add).

#4 - 24 juillet 2013 17:50 - Thomas Noël

- Fichier wcs-sql-drop.patch ajouté

Un patch qui ajoute du DROP sur les formdata. Testé, ça marche...

Mais je ne suis pas bien à l'aise avec l'idée : une erreur de manip sur un champ dans un formdef va **vraiment** effacer immédiatement toutes les données du champ dans les demandes correspondantes...

#5 - 29 juillet 2013 12:36 - Benjamin Dauvergne

Je pense qu'ajouter une méthode FormDef.allocate_field_id() qui se base sur un compteur sauvegardé dans le FormDef pour remplacer les str(max([lax_int(x.id) for x in self.objectdef.fields]) + 1) un peu partout ça devrait suffire. Dans le cas du backend postgres on pourrait aussi utiliser un objet séquence¹ pour allouer cet id (mais je ne sais pas si actuellement la classe FormDef se spécialise quand on utilise le backend postgres).

1

```
CREATE SEQUENCE formdef_xx_field_id_seq;
```

```
SELECT nextval('formdef_xx_field_id_seq');
```

#6 - 29 juillet 2013 14:45 - Thomas Noël

- Fichier `wcs-sql-drop.patch` supprimé

#7 - 29 juillet 2013 14:49 - Thomas Noël

- Fichier `wcs-do_x_tables-with-drop.patch` ajouté

Patch que je propose de déployer maintenant en production sur Mtp. et Mez., qui gère bien les colonnes SQL dans les tables formdef et users. Testé à la main car Serghei ne m'a pas encore tout appris des tests en Python.

Note importante : ce patch pourra être retiré sans dommage quand on aura le patch qui évite les doublons sur les f<id> (patch à venir). Ce qui serait préférable, car les "DROP" du présent patch ne m'inspirent pas beaucoup -- mais j'ai pas d'autre solution rapide et efficace sous la main.

#8 - 29 juillet 2013 14:57 - Thomas Noël

- Fichier `wcs-formdef-id-counter.patch` ajouté

Autre **futur** patch, à tester car il ne couvre sans doute pas tout le périmètre du soucis, à ne pas basculer en production maintenant. Ce patch est dédié à remplacer le patch précédent, sur l'idée de Benjamin : chaque formdef a un compteur de id associé, qui ne diminue jamais. Ainsi on créé toujours des nouveaux champs.

A noter que c'est utile aussi pour les wcs en mode "pickle", ça évite de se retrouver avec d'anciennes valeurs d'un champs détruits sur un nouveau champ (manip pour reproduire le bug : détruire le dernier champ créé dans formulaire puis créer un autre champ : les valeurs de l'ancien champ détruit se retrouvent dans le nouveau, SQL ou pas).

Le patch provisoire contient une partie "pas géniale" dans `admin/settings.ptl` à cause de l'utilisation de FormDef stocké en XML pour la définition des champs utilisateurs. Il faudra aussi voir les autres partie où FormDef est utilisé, par exemple les formulaires de workflows, etc.

#9 - 29 juillet 2013 15:55 - Benjamin Dauvergne

Thomas Noël a écrit :

Patch que je propose de déployer maintenant en production sur Mtp. et Mez., qui gère bien les colonnes SQL dans les tables formdef et users. Testé à la main car Serghei ne m'a pas encore tout appris des tests en Python.

Note importante : ce patch pourra être retiré sans dommage quand on aura le patch qui évite les doublons sur les f<id> (patch à venir). Ce qui serait préférable, car les "DROP" du présent patch ne m'inspirent pas beaucoup -- mais j'ai pas d'autre solution rapide et efficace sous la main.

ok pour moi

#10 - 29 juillet 2013 15:58 - Benjamin Dauvergne

Thomas Noël a écrit :

Autre **futur** patch, à tester car il ne couvre sans doute pas tout le périmètre du soucis, à ne pas basculer en production maintenant. Ce patch est dédié à remplacer le patch précédent, sur l'idée de Benjamin : chaque formdef a un compteur de id associé, qui ne diminue jamais. Ainsi on créé toujours des nouveaux champs.

A noter que c'est utile aussi pour les wcs en mode "pickle", ça évite de se retrouver avec d'anciennes valeurs d'un champs détruits sur un nouveau champ (manip pour reproduire le bug : détruire le dernier champ créé dans formulaire puis créer un autre champ : les valeurs de l'ancien champ détruit se retrouvent dans le nouveau, SQL ou pas).

Le patch provisoire contient une partie "pas géniale" dans `admin/settings.ptl` à cause de l'utilisation de FormDef stocké en XML pour la définition des champs utilisateurs. Il faudra aussi voir les autres partie où FormDef est utilisé, par exemple les formulaires de workflows, etc.

Pour moi il manque un `self.store()` à la fin de `get_next_field_id()`.

#11 - 29 juillet 2013 16:56 - Thomas Noël

Benjamin Dauvergne a écrit :

Pour moi il manque un `self.store()` à la fin de `get_next_field_id()`.

Yep, merci pour le coup d'oeil. Je n'ai pas mis le store car la fonction `get_next_field_id()` est uniquement destinée à être utilisée lors de l'ajout d'un field, donc d'un store implicite... Mais c'est un get qui modifie la valeur, c'est finalement assez merdique comme fonction.

A la reflexion je me dis qu'il serait peut-être préférable, si ça colle dans le reste du code, de faire une méthode `formdef.add_field(field)` qui génère le `field.id`, fait le `fields.append(field)` puis le `formdef.store()`. Je regarderai un de ces quatre.

#12 - 29 juillet 2013 18:40 - Thomas Noël

Patch "DROP" poussé, commit [82cf4f12](#)

#13 - 29 juillet 2013 18:41 - Thomas Noël

Trace suite à ce patch :

Exception:

```
type = '<class 'psycogp2.ProgrammingError'>', value = 'column "f41" of relation "formdat
a_44_demande_de_stage_copie" does not exist
'
```

Stack trace (most recent call first):

```
File "/usr/lib/python2.6/wcs/sql.py", line 162, in do_formdef_tables
 160     # delete obsolete fields
 161     for field in (existing_fields - needed_fields):
> 162         cur.execute('ALTER TABLE %s DROP COLUMN %s' % (table_name, field))
 163
 164     conn.commit()
```

locals:

```
sql_type = 'date'
cur = <cursor object at 0x2bb6af8; closed: 0>
existing_fields = set(['f1', 'f2', 'f3', 'f4', 'f5', 'f6', 'f7', 'f8', 'f9', 'f32', '
id', 'f41', 'f40', 'fts', 'workflow_roles', 'user_id', 'f21', 'f1_display', 'f20',
'receipt_time', 'user_hash', 'id_display', 'f33', 'f27', 'f29', 'f26', 'status', 'f25',
'f43', 'f38', 'workflow_roles_array', 'f39', 'f13_display', 'workflow_data', 'f30', 'f31',
'f18', 'f19', 'f34', 'f35', 'f36', 'f37', 'f12', 'f13', 'f10', 'f11', 'f17'])
formdef = <FormDef "Demande d'emploi saisonnier" id:44>
field = 'f41'
needed_fields = set(['f1', 'f2', 'f3', 'f4', 'f5', 'f6', 'f7', 'f8', 'f32', 'id', 'f4
0', 'fts', 'workflow_roles', 'user_id', 'f21', 'f1_display', 'f33', 'receipt_time',
'user_hash', 'id_display', 'f29', 'status', 'f43', 'f38', 'workflow_roles_array',
'workflow_data', 'f30', 'f31', 'f18', 'f19', 'f34', 'f35', 'f36', 'f37', 'f12', 'f39',
'f10', 'f11', 'f17'])
x = ('f43',)
table_name = 'formdata_44_demande_de_stage_copie'
conn = <connection object at 0x304b1f0; dsn: 'dbname=eservices host=localhost port=54
32 user=auquotidien password=xxxxxxx', closed: 0>
```

Seule explication à cette heure : accès concurrent, le DROP a été déjà effectué entre temps..

Solution proposée : DROP COLUMN IF EXISTS

#14 - 30 juillet 2013 08:58 - Frédéric Péters

Je ne suis pas emballé, je verraï plutôt la modification en amont, pour ne pas réutiliser un id de champ déjà affecté. (→ garder un compteur au niveau du formdef et utiliser celui-ci plutôt que les "(max([lax_int(x.id) for x in self.objectdef.fields]) + 1" du moment).

En parallèle, garder ce code pour supprimer les colonnes correspondant à des champs supprimés, c'est bien.

Aussi, pour moi, les tests unitaires développés sur la partie SQL ont été très utiles.

#15 - 30 juillet 2013 09:18 - Frédéric Péters

- Fichier 0001-formdefs-keep-track-of-allocated-field-ids-3363.patch ajouté

Proof of concept, à véritablement relire et tester.

#16 - 30 juillet 2013 09:22 - Frédéric Péters

Et beh, j'avais pas fait gaffe aux échanges ici, il y a sans doute à merger entre mon patch et le précédent.

#17 - 30 juillet 2013 10:39 - Jérôme Schneider

Le commit <http://repos.entrouvert.org/wcs/git/commit/?id=e85781e4add0ae9cdde8ee6eb64004fad606948a> qui est déployé sur la recette de Montpellier ne tourne pas avec PostgreSQL < 9.0. Le soucis c'est que Montpellier tourne avec un pgsq 8.4.

Je vois deux solutions :

- demander à Ovea d'installer le backports de postgresSQL (9.1) avec les soucis de migration à prévoir
- faire un fix local à Montpellier qui vire le 'IF EXISTS' et ajoute un try except sur le drop column

J'attends vos retours, propositions, ... avant de bouger.

#18 - 30 juillet 2013 14:31 - Benjamin Dauvergne

Le try/except c'est suffisant si on est en mode autocommit, sinon il faut aussi faire un rollback dans le except je pense.

#19 - 30 juillet 2013 18:02 - Jérôme Schneider

- Fichier 0001_montpellier_fix_pgsql_84.sql ajouté

Pour moi il n'y a pas besoin de faire de rollback même lorsqu'on est pas en autocommit en tout cas sur une erreur : `psycpg2.ProgrammingError`.

Je viens de faire des tests avec une erreur '`psycpg2.ProgrammingError`' et un try / except pass et je n'ai pas eu de soucis. Il fait bien toutes les requêtes sauf celle là au moment du commit et il ne plante pas.

Je propose donc le patch suivant sachant que je ne pense le mettre que sur Montpellier et temporairement le temps d'avoir une solution propre.

edit : c'est déployé sur montpellier

#20 - 31 juillet 2013 10:14 - Benjamin Dauvergne

Ok avec tous tes tests ça me va. Je rejoins une demande récurrente de Fred, ce serait possible d'ajouter un test de non-régression (sachant que personne ne sait reproduire le problème, c'est un peu difficile je reconnais).

#21 - 31 juillet 2013 14:59 - Thomas Noël

En supposant postgresql > 8 on pourrait plutôt le laisser gérer l'exception, et faire un "pass" uniquement si la colonne n'existait pas ; dans le genre de : <http://stackoverflow.com/questions/12597465/how-to-add-column-if-not-exists-on-postgresql/12608570#12608570>

#22 - 07 août 2013 10:47 - Frédéric Péters

- Statut changé de Nouveau à En cours

- Assigné à changé de Thomas Noël à Frédéric Péters

- Fichier 0001-formdefs-keep-track-of-allocated-field-ids-3363.patch ajouté

Version actualisée de mon patch, qui reprend des éléments de celui de Thomas (`wcs-formdef-id-counter.patch`) ainsi qu'un nouveau test.

J'y ai par contre laissé le "... IF EXISTS" pour en avoir une exécution sur le Postgresql 8.4 de Jenkins, et regarder ce que ça donne.

Je pousse donc ce patch dans le dépôt.

#23 - 07 août 2013 11:03 - Frédéric Péters

https://jenkins.entrouvert.org/job/wcs/311/testReport/junit/tests/test_sql/test_sql_table_add_and_remove_fields/ pour l'erreur Jenkins.

#24 - 07 août 2013 11:13 - Frédéric Péters

Et ce qui est bien, c'est que ça ne marche pas :)

```
> cur.execute('ALTER TABLE %s DROP COLUMN %s' % (table_name, field))
E InternalError: current transaction is aborted, commands ignored until end of transaction b
lock
```

Et en fait, à relire, le "IF EXISTS", c'est juste une précaution prise par Thomas dans le premier patch ? Parce qu'a priori, comme la liste des colonnes existants est récupérée directement de Postgresql, il n'y aura jamais que des colonnes existantes là-dedans.

Je me permets donc de supprimer le "IF EXISTS" de manière générale.

#25 - 07 août 2013 11:59 - Jérôme Schneider

On Wed, Aug 07, 2013 at 11:13:07AM +0200, redmine@entrouvert.com wrote:

Et en fait, à relire, le "IF EXISTS", c'est juste une précaution prise par Thomas dans le premier patch ? Parce qu'a priori, comme la liste des colonnes existants est récupérée directement de Postgresql, il n'y aura jamais que des colonnes existantes là-dedans.

Je me permets donc de supprimer le "IF EXISTS" de manière générale.

Le IF EXISTS avait été ajouté dans un second patch de Thomas suite à une erreur avec le traceback avec le potentiel 'accès concurrent' :

Quixote Traceback (ProgrammingError: column "f41" of relation "formdata_44_demande_de_stage_copie" does not exist)

Ce bug c'était produit lors de l'application du patch initial

(d9eb4bd94be9ae1de5658a20e3934c90d876c1df).

C'est pour corriger ce traceback qu'il a ajouté le 'IF EXISTS' dans le commit e85781e4add0ae9cdde8ee6eb64004fad606948a.

#26 - 07 août 2013 12:04 - Frédéric Péters

Merci pour l'historique; je ne pense cependant pas la modification pertinente (l'ensemble des modifs se fait dans une seule transaction, l'accès concurrent je ne vois pas trop comment il peut avoir lieu).

#27 - 07 août 2013 17:44 - Thomas Noël

Frédéric Péters a écrit :

Merci pour l'historique; je ne pense cependant pas la modification pertinente (l'ensemble des modifs se fait dans une seule transaction, l'accès concurrent je ne vois pas trop comment il peut avoir lieu).

Ok avec ça, ce truc m'énervait aussi.

Reste que je n'ai pas d'explication à la trace que nous avons eu (un DROP avorté car la colonne n'existait pas...). Enigme qu'on pourra peut-être résoudre si le problème se reproduit, ce qui est une raison supplémentaire de virer le IF EXISTS.

#28 - 07 août 2013 17:55 - Jérôme Schneider

- Fichier 0001_montpellier_fix_pgsql_84.sql supprimé

#29 - 08 août 2013 09:26 - Frédéric Péters

- Statut changé de En cours à Résolu (à déployer)

#30 - 11 septembre 2013 10:31 - Thomas Noël

- Statut changé de Résolu (à déployer) à Fermé

#31 - 09 octobre 2013 10:51 - Thomas Noël

Le retour du problème (sur magglo-16, 08 Oct 2013 11:56:28) :

Exception:

```
type = '<class 'psycopg2.ProgrammingError'>', value = 'column "f16" of relation "formdata_48_contacter_le_service_de_la_col" already exists'
```

Stack trace (most recent call first):

```
File "/usr/lib/python2.6/dist-packages/wcs/sql.py", line 154, in do_formdef_tables
 152         table_name,
 153         'f%s' % field.id,
> 154         sql_type))
 155     if field.store_display_value:
 156         needed_fields.add('f%s_display' % field.id)
```

```
locals:
  sql_type = 'varchar'
  cur = <cursor object at 0x2aa9240; closed: 0>
  existing_fields = set(['status', 'f1', 'fts', 'workflow_roles', 'user_id', 'f5', 'f8', 'f14_display', 'f3', 'f15_display', 'workflow_roles_array', 'receipt_time', 'user_hash', 'id_display', 'f12', 'f4', 'f15', 'id', 'f14', 'workflow_data'])
  formdef = <FormDef "Signalement d'anomalie de collecte" id:48>
  field = <wcs.fields.StringField instance at 0x2b06dd0>
  needed_fields = set(['status', 'f1', 'fts', 'workflow_roles', 'user_id', 'f5', 'f8', 'f14_display', 'f3', 'f15_display', 'workflow_roles_array', 'receipt_time', 'user_hash', 'id_display', 'f12', 'f4', 'f15', 'f16', 'id', 'f14', 'workflow_data'])
  x = ('f12',)
  table_name = 'formdata_48_contacter_le_service_de_la_col'
  conn = <connection object at 0x3aaed30; dsn: 'dbname=auquotidien', closed: 0>
```

```
File "/usr/lib/python2.6/dist-packages/wcs/sql.py", line 95, in f
 93     except psycopg2.Error:
 94         get_connection().rollback()
> 95         raise
 96     return f
 97.
```

On voit que le existing_fields n'a pas vu le champ f16, qui a dû être créé entre temps par un autre process.

#32 - 24 octobre 2013 19:05 - Benjamin Dauvergne

- Statut changé de Fermé à En cours

Après re-discussion le problème est normal. Si deux transactions commence pour ajouter la même colonne, aucune des deux ne va voir l'action de l'autre pendant son exécution mais lors du commit l'une des deux sera exclu. Pour éviter ce problème il faut un lock explicite sur la table des tables ou bien utiliser le suffixe FOR UPDATE sur le select de la table des tables (ce sont des syntaxes existantes en 8.4).

#33 - 24 octobre 2013 19:07 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Après re-discussion le problème est normal. Si deux transactions commence pour ajouter la même colonne, aucune des deux ne va voir l'action de l'autre pendant son exécution mais lors du commit l'une des deux sera exclu. Pour éviter ce problème il faut un lock explicite sur la table des tables ou bien utiliser le suffixe FOR UPDATE sur le select de la table des tables (ce sont des syntaxes existantes en 8.4).

Remplacer table des tables par table des colonnes.

#34 - 24 octobre 2013 19:08 - Benjamin Dauvergne

Un LOCK information_schema.columns au début de la méthode do_formdef_tables() devrait suffire.

#35 - 03 janvier 2014 14:24 - Frédéric Péters

Pour mémoire, plutôt que l'improbable hypothèse de la modif concurrente, c'est peut-être une manifestation du cache du data_class (cf [#4060](#)).

#36 - 08 avril 2014 09:53 - Thomas Noël

- Statut changé de En cours à Résolu (à déployer)

Plus de soucis depuis 3 mois malgré de plus en plus de mises en prod psql.

#37 - 14 juin 2014 16:26 - Thomas Noël

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

#38 - 23 septembre 2014 14:22 - Thomas Noël

- Statut changé de Solution déployée à Fermé

Fichiers

wcs-do_x_tables-with-drop.patch	4,59 ko	29 juillet 2013	Thomas Noël
wcs-formdef-id-counter.patch	4,71 ko	29 juillet 2013	Thomas Noël
0001-formdefs-keep-track-of-allocated-field-ids-3363.patch	7,59 ko	30 juillet 2013	Frédéric Péters
0001-formdefs-keep-track-of-allocated-field-ids-3363.patch	10,1 ko	07 août 2013	Frédéric Péters