

Ne pas faire de CAST dans les jointures si la colonne peut contenir quelque chose qui n'est pas un entier

17 juin 2021 15:39 - Benjamin Dauvergne

Statut:	Nouveau	Début:	17 juin 2021
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:			
Patch proposed:	Non	Planning:	Non

Description

Dans #54943, la requête suivante :

```
SELECT formdata_125_inscription_copie.id, formdata_125_inscription_copie.user_id, formdata_125_inscription_copie.receipt_time, formdata_125_inscription_copie.status, formdata_125_inscription_copie.page_no, formdata_125_inscription_copie.anonymised, formdata_125_inscription_copie.workflow_data, formdata_125_inscription_copie.prefilling_data, formdata_125_inscription_copie.id_display, formdata_125_inscription_copie.workflow_roles, formdata_125_inscription_copie.workflow_roles_array, formdata_125_inscription_copie.concerned_roles_array, formdata_125_inscription_copie.actions_roles_array, formdata_125_inscription_copie.tracking_code, formdata_125_inscription_copie.backoffice_submission, formdata_125_inscription_copie.submission_context, formdata_125_inscription_copie.submission_agent_id, formdata_125_inscription_copie.submission_channel, formdata_125_inscription_copie.criticality_level, formdata_125_inscription_copie.last_update_time, formdata_125_inscription_copie.digest, formdata_125_inscription_copie.user_label, formdata_125_inscription_copie.f67, formdata_125_inscription_copie.f67_display, formdata_125_inscription_copie.f67_structured, formdata_125_inscription_copie.f20, formdata_125_inscription_copie.f20_display, formdata_125_inscription_copie.f20_structured, formdata_125_inscription_copie.f18, formdata_125_inscription_copie.f18_display, formdata_125_inscription_copie.f18_structured, formdata_125_inscription_copie.f24, formdata_125_inscription_copie.f77, formdata_125_inscription_copie.f77_display, formdata_125_inscription_copie.f77_structured, formdata_125_inscription_copie.f80, formdata_125_inscription_copie.f40, formdata_125_inscription_copie.f40_display, formdata_125_inscription_copie.f40_structured, formdata_125_inscription_copie.f41, formdata_125_inscription_copie.f41_display, formdata_125_inscription_copie.f41_structured, formdata_125_inscription_copie.f42, formdata_125_inscription_copie.f42_display, formdata_125_inscription_copie.f42_structured, formdata_125_inscription_copie.f78, formdata_125_inscription_copie.f26, formdata_125_inscription_copie.f26_display, formdata_125_inscription_copie.f26_structured, formdata_125_inscription_copie.fbo7aab9a7b_7465_444d_a466_3b992a987ddf, formdata_125_inscription_copie.fbo7aab9a7b_7465_444d_a466_3b992a987ddf_display, formdata_125_inscription_copie.fbo7aab9a7b_7465_444d_a466_3b992a987ddf_structured, formdata_125_inscription_copie.fbo7906d399_555b_45f4_ae15_53d9603375e5, formdata_125_inscription_copie.fbo974423e4_b4b4_42bb_94f5_39476e3cca5a, formdata_125_inscription_copie.fboe2038904_46ab_4fad_aa80_14dbcf688552, formdata_125_inscription_copie.fbo05ce9d97_75b7_4a84_80d8_b4b5e0d47230, formdata_125_inscription_copie.fbo04908fe_0590_4179_ae5f_3d1f61ba3c26, formdata_125_inscription_copie.fbo61cf6dbd_709c_4523_99d9_f649180ed048, formdata_125_inscription_copie.fbo61cf6dbd_709c_4523_99d9_f649180ed048_display, formdata_125_inscription_copie.fbo61cf6dbd_709c_4523_99d9_f649180ed048_structured, formdata_125_inscription_copie.fbo0b64ade2_2f41_4c04_94d3_f4e24f7a60cd, formdata_125_inscription_copie.fbo0b64ade2_2f41_4c04_94d3_f4e24f7a60cd_display, formdata_125_inscription_copie.fbo0b64ade2_2f41_4c04_94d3_f4e24f7a60cd_structured, formdata_125_inscription_copie.fboaf8b3c57_3b92_4562_aba7_ba8af0adec6f, formdata_125_inscription_copie.fbo3f70c788_7a0a_4e21_98cd_d171b19f679a, formdata_125_inscription_copie.fbo834fd902_9c12_417c_bca8_2a2523e30aaa, formdata_125_inscription_copie.fbo834fd902_9c12_417c_bca8_2a2523e30aaa_display, formdata_125_inscription_copie.fbo834fd902_9c12_417c_bca8_2a2523e30aaa_structured, formdata_125_inscription_copie.fbo70eb6d9b_68ca_42a5_9d6f_a9cf2e3f6f2d, formdata_125_inscription_copie.fboaf7ff58b_e98d_4119_8694_ce4c2c957999, formdata_125_inscription_copie.fbo78ce192b_5eaf_4d26_ba92_ee426f967291, formdata_125_inscription_copie.fbo4a1311dc_3fb4_4dac_a7fc_267f89762f87, formdata_125_inscription_copie.fbo4a1311dc_3fb4_4dac_a7fc_267f89762f87_display, formdata_125_inscription_copie.fbo4a1311dc_3fb4_4dac_a7fc_267f89762f87_structured, formdata_125_inscription_copie.fbof7027adc_8bc1_44c0_8518_37a780ecc643, formdata_125_inscription_copie.fbobddee6ea_dbec_4443_8664_b7f10eb3918d, formdata_125_inscription_copie.fbof230e5a3_1f5e_4a2a_9cc1_447e67fee2b1, users.f_last_name, users.f_first_name, users.f_phone, users.f_mobile, t139769934828264.f7, t139769934828264.f10, t139769934828264.f9, t139769934828264.f4\nFROM formdata_125_inscription_copie LEFT JOIN users ON (CAST(f
```

```
ormdata_125_inscription_copie.user_id AS INTEGER) = users.id) LEFT JOIN carddata_2_senior AS t139769934828264 ON (CAST(formdata_125_inscription_copie.fbo7aab9a7b_7465_444d_a466_3b992a987ddf AS INTEGER) = t139769934828264.id)\n                                WHERE formdata_125_inscription_copie.id IN (2066,2065,2064,2063,2060,2058,2056,2055,2054,2053,2052,2051,2050,2049,2048,2046,2045,2041,2038,2036) '\n    tables = ['formdata_125_inscription_copie', 'LEFT JOIN users ON (CAST(formdata_125_inscription_copie.user_id AS INTEGER) = users.id)', 'LEFT JOIN carddata_2_senior AS t139769934828264 ON (CAST(formdata_125_inscription_copie.fbo7aab9a7b_7465_444d_a466_3b992a987ddf AS INTEGER) = t139769934828264.id)']
```

génère une erreur SQL 'invalid input syntax for integer: ""' à cause des CAST (... AS INTEGER) je suppose.

Historique

#2 - 17 juin 2021 15:41 - Benjamin Dauvergne

À mon avis vu que SQL fait un cast explicite dans les comparaisons le CAST ici est inutile, on peut juste faire `machin.col1 = truc.id` où `truc.id` est un entier, `x` sera casté en entier si possible même si c'est une chaîne sinon ignoré.

#3 - 17 juin 2021 15:46 - Benjamin Dauvergne

Non je dis de la **bip** ça ne marche pas non plus, c'est la colonne entière qu'il faut convertir en chaîne, mais ce sera plus lent (à moins que postgresql optimise particulièrement ce cas mais j'en doute).

#4 - 17 juin 2021 15:48 - Benjamin Dauvergne

On peut implémenter une fonction pour gérer un "CAST or NULL",
<https://dba.stackexchange.com/questions/203934/postgresql-alternative-to-sql-server-s-try-cast-function>

#5 - 17 juin 2021 15:53 - Frédéric Péters

Cette description fonctionnelle du bug/ticket est-elle correcte : « Quand un tableau de traitement est configuré pour reprendre des colonnes issues de fiches liées (via des champs de type liste) une jointure est ajoutée à la requête SQL, mais celle-ci échoue si jamais le champ "liste" liant une fiche n'est pas obligatoire, que se trouvent alors dedans None pour certaines demandes » ?

#6 - 17 juin 2021 15:56 - Benjamin Dauvergne

Je pense oui, mais le souci c'est le CAST foireux c'est pas que la colonne soit vide. Il y a d'autres façon d'avoir une colonne non castable (pas forcément vide) qu'une liste optionnelle (changement de source de donnée).

#7 - 17 juin 2021 16:01 - Frédéric Péters

mais le souci c'est le CAST foireux

J'essaie de monter la description à un niveau moins technique, ce qui sera utile lors de l'écriture de tests.

Il y a d'autres façon d'avoir une colonne non castable (pas forcément vide) qu'une liste optionnelle (changement de source de donnée).

Genre on aurait un champ liste alimenté par une source de données externes et on modifierait le champ pour plutôt utiliser une source de données issue de fiches ? (Je serais assez pour interdire cette situation.)

#8 - 17 juin 2021 19:52 - Benjamin Dauvergne

Bon ici le champ¹ est obligatoire mais c'est un champ de workflow donc ça n'a d'impact (je me dis d'ailleurs qu'on devrait supprimer la checkbox pour les champs de workflow, ils sont forcément optionnels). En dehors d'empêcher ceci ou cela, serait-il possible de poser NULL plutôt qu'une chaîne vide quand on affecte la chaîne vide à un champ liste ?

Pour corriger le problème immédiatement j'ai changé les valeurs chaîne vide en NULL (#54943-11).

On pourrait poser définitivement que `id == ''` n'est pas autorisé (je ne sais pas si ça risque de casser quelque chose quelque part), éventuellement avec une migration pour normaliser les cas existants, et si on sent aventureux un

```
ALTER TABLE formdata_xxx DROP CONSTRAINT IF EXISTS fboxxxxx_not_empty;\nALTER TABLE formdata_xxx ADD CONSTRAINT fboxxxxx_not_empty CHECK (fboxxxxx <> '');
```

pour éviter que ça n'arrive.

¹<https://demarches-departement06.test.entrouvert.org/backoffice/workflows/105/backoffice-fields/fields/bo7aab9a7b-7465-444d-a466-3b992a987dd/>