

projekt kivitendo - Fehler #640

DBUpgrade fehler bei sql/Pg-upgrade2/oe_purchase_order_confirmation_order_types.sql

22.11.2023 14:54 - Cem Aydin

Status:	Erledigt	Beginn:	22.11.2023
Priorität:	Normal	Abgabedatum:	
Zugewiesen an:		% erledigt:	0%
Kategorie:		Geschätzter Aufwand:	0:00 Stunde
Zielversion:	3.9	Aufgewendete Zeit:	0:00 Stunde
Beschreibung Hallo, bei mir lokal kommt nach einem pull (master), beim db upgrade: Fehler! The database update/creation did not succeed. The file sql/Pg-upgrade2/oe_purchase_order_confirmation_order_types.sql containing the following query failed: ALTER TYPE order_types ADD VALUE IF NOT EXISTS 'purchase_order_confirmation' The error message was: FEHLER: ALTER TYPE ... ADD kann nicht in einem Transaktionsblock laufen All changes in that file have been reverted.: Es scheint, dass das problem möglicherweise bei älteren postgres versionen auftritt, gemäß kommentaren, nicht mehr ab PG V. 12.: https://stackoverflow.com/questions/53149484/error-alter-type-add-cannot-run-inside-a-transaction-block Lokal habe ich PG V. 11. Möglicherweise geht es auch, wenn man die query manuell ausführt (gemäß kommentaren im link).			

Zugehörige Revisionen

Revision 273ec927 - 24.11.2023 16:19 - Bernd Bleßmann

Doku: UPGRADE: Hinweis auf Probleme mit Upgrade-Skript und Postgres < 12

Refs #640 (redmine)

Historie

#1 - 22.11.2023 14:57 - Cem Aydin

- Datei Screenshot from 2023-11-22 14-56-27.png wurde hinzugefügt

Hmmm, manuell ausführen geht aber der fehler tritt trotzdem noch auf...

#2 - 22.11.2023 15:51 - Bernd Bleßmann

- Zielversion wurde auf 3.9 gesetzt

Hi - wollte auch gerade ein Ticket erstellen, da mit nicht ganz klar ist, wie wir damit umgehen sollen.

Ab PostgreSQL Version 12 geht das. Davor kann "ALTER TYPE ... ADD ..." nicht in einer Transaktion ausgeführt werden.

Du kannst die Query im Upgrade-Skript auskommentieren (damit es durchläuft und als gelaufen eingetragen wird) und dann direkt in der Datenbank die Query ausführen, damit es nicht in einer Transaktion läuft.

#3 - 22.11.2023 16:06 - Bernd Bleßmann

Die Lösungen für's Projekt wären meiner Meinung nach:

- PostgreSQL >= Version 12 verlangen; Doku und UPGRADE-Datei anpassen, evtl. dann auch noch im Installation-Check prüfen; Unterstützte Betriebssysteme in der Doku anpassen

und/oder

- Kommentar in der DB-Upgrade-Datei (evtl. eine Ausgabe beim Upgrade?) und Erläuterung in doc/UPGRADE, wie man dass dann händisch löst

#4 - 22.11.2023 16:16 - Cem Aydin

- Zielversion 3.9 wurde gelöscht

Du kannst die Query im Upgrade-Skript auskommentieren (damit es durchläuft und als gelaufen eingetragen wird) und dann direkt in der Datenbank die Query ausführen, damit es nicht in einer Transaktion läuft.

Ja, hab ich auch so gemacht, das hat funktioniert.

Bernd Bleßmann schrieb:

Die Lösungen für's Projekt wären meiner Meinung nach:

- PostgreSQL >= Version 12 verlangen; Doku und UPGRADE-Datei anpassen, evtl. dann auch noch im Installation-Check prüfen; Unterstützte Betriebssysteme in der Doku anpassen

und/oder

- Kommentar in der DB-Upgrade-Datei (evtl. eine Ausgabe beim Upgrade?) und Erläuterung in doc/UPGRADE, wie man dass dann händisch löst

Ja, denke ist beides möglich von mir aus. Das müsst ihr, beziehungsweise gemeinsam entschieden werden, was da das vorgehen ist. Wenn man offiziell PG V. 12 verlangt, wäre ja theoretisch der workaround trotzdem noch möglich.

Ist ja auch immer gut eine motivation zu haben zum upgraden :).

#5 - 22.11.2023 16:30 - Bernd Bleßmann

- Zielversion wurde auf 3.9 gesetzt

#6 - 29.11.2023 23:45 - Andreas Rudin

Damit das Datenbankupgradescript auch mit älteren Postgres-Versionen kompatibel ist, kann folgendes Script verwendet werden:

Statt:

```
ALTER TYPE order_types ADD VALUE IF NOT EXISTS 'purchase_order_confirmation';
```

Neu:

```
INSERT INTO pg_enum (  
    enumtypeid,  
    enumlabel,  
    enumsortorder  
)  
SELECT  
(SELECT oid FROM pg_type WHERE typename = 'order_types')::regtype::oid,  
'purchase_order_confirmation',  
(SELECT MAX + 1 FROM pg_enum WHERE enumtypeid = (SELECT oid FROM pg_type WHERE typename = 'order_types')::regtype);
```

Spricht etwas dagegen, dass das Upgradescript entsprechend geändert wird?

#7 - 30.11.2023 09:54 - Bernd Bleßmann

Andreas Rudin schrieb:

Damit das Datenbankupgradescript auch mit älteren Postgres-Versionen kompatibel ist, kann folgendes Script verwendet werden:

Statt:

```
ALTER TYPE order_types ADD VALUE IF NOT EXISTS 'purchase_order_confirmation';
```

Neu:

```
INSERT INTO pg_enum (  
    enumtypeid,  
    enumlabel,  
    enumsortorder  
)  
SELECT  
(SELECT oid FROM pg_type WHERE typename = 'order_types')::regtype::oid,  
'purchase_order_confirmation',  
(SELECT MAX + 1 FROM pg_enum WHERE enumtypeid = (SELECT oid FROM pg_type WHERE typename = 'order_types')::regtype);
```

Spricht etwas dagegen, dass das Upgradescript entsprechend geändert wird?

Hallo Andreas,

ja. Ich halte nicht so viel davon, im Postgres-System-Katalog direkt was zu ändern. Es wird ja einen Grund geben, warum das nicht geht. Oder andersrum: Ich kann nicht beurteilen, was da alles passieren kann.

Viele Grüße
Bernd

#8 - 05.03.2024 09:45 - Sven Schöling

PostgreSQL >= Version 12 verlangen; Doku und UPGRADE-Datei anpassen, evtl. dann auch noch im Installation-Check prüfen; Unterstützte Betriebssysteme in der Doku anpassen

Ich hab gerade nachgeschaut, Debian 10 (Buster) hat noch postgres 11. Das ist noch LTS Support und ist noch als empfohlen in der Dokumentation hinterlegt.

#9 - 05.03.2024 09:55 - Sven Schöling

- Status wurde von Neu zu Erledigt geändert

Besprochen im Team. Entscheidung:

Debian Buster (LTS bis 2024) wird nicht mehr offiziell unterstützt.

> ~~close, won't fix.~~
> neues Ticket um die Minmalanforderungen hochzusetzen in der Dokumentation.

Dateien

Screenshot from 2023-11-22 14-56-27.png	29,5 KB	22.11.2023	Cem Aydin
---	---------	------------	-----------