Appunti Postgres

Procedura di ripristino DB Postgres in caso di Esaurimento spazio su disco

fare riferimento al seguente link

https://www.endpoint.com/blog/2014/09/25/pgxlog-disk-space-problem-on-postgres

1 - aperto un terminale in \PostgreSQL\“n versione”\bin (shift + tasto dx –> apri terminale qui)

2 - eseguito il comando pg_controldata “path alla directory data di Postgres”

3 - fare riferimento alla riga “File WAL di REDO dell'ultimo checkpoint:” per vedere qual'è l'ultimo xlog correttamente salvato

4 - eseguire una copia della cartella pg_xlog di BKP

5 - cancellare tutti i file nella cartella pg_xlog tranne quello indicato al punto 3

6 - terminare tutti i processi relativi a postgres in task manager

7 - cancellare il file \PostgreSQL\“n versione”\data\postmaster.pid

8 - riavviare il servizio

Reimpostazione di una SEQUENCE in Postgres

```
ALTER TABLE fs12_data.rqm_request ALTER COLUMN prog_request DROP DEFAULT;

DROP SEQUENCE fs12_data.rqm_request_prog_request_seq;

CREATE SEQUENCE fs12_data.rqm_request_prog_request_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 3000
  CACHE 1;
ALTER TABLE fs12_data.rqm_request_prog_request_seq
  OWNER TO fs12_data;


ALTER TABLE fs12_data.rqm_request ADD COLUMN prog_request INTEGER;
ALTER TABLE fs12_data.rqm_request ALTER COLUMN prog_request SET NOT NULL;
ALTER TABLE fs12_data.rqm_request ALTER COLUMN prog_request SET DEFAULT
NEXTVAL('fs12_data.rqm_request_prog_request_seq'::regclass);
```

Velocizzare le Dipendenze di PostgreSQL

Questa funzione permette di semplificare, le generose dipendenze di PostgreSQL.

Per spiegare meglio “dipendenze”, vi pongo un esempio: avete presente quando dovete modificare il data type di un campo in una tabella ma non potete perché questa ha delle viste che dipendono da essa? Oppure, dovete modificare una vista ma non potete perché questa ha altre viste dipendenti. Insomma, credo di aver reso bene l'idea.

Prima di questa funzione, ho sempre fatto tutto a mano: salvato lo script della vista, eliminato la vista, modificato il campo e ricreato una vista. Diciamo che finché si tratta di una vista il problema non si pone, ma quando, come nel mio caso, ci sono più di 10 viste dipendenti dalla tabella?

Bene, qui entra in gioco la funzione di cui vi parlavo. Lo script non impedisce a PostgreSQL di generare dipendenze, perché queste sono una funzionalità efficace del software e ci aiutano a mantenere un prodotto funzionante, ma aiuta l'utente a risparmiare tempo nel ricreare tutte le viste (o altre dipendenze) a mano.

Dunque, la funzione che vi consiglio oggi salva, con un click, tutti gli script che ricreano le viste (o altre dipendenze) e con un altro le rigenera. Il tutto senza uscire dall'interfaccia di Pgadmin, poiché si tratta di una funzione che si crea nel database.

Installazione Se vi interessa, potete scaricare la cartella condivisa via mail ed eseguire lo script su Pgadmin, contenuto nel file: Edit_4_gw_save_restore_dependencies_psql.sql Questo non farà altro che creare: - una tabella, dove verranno salvate le dipendenze, chiamata: deps_saved_ddl - una funzione che salva (nella tabella) e cancella (dal database) le dipendenze di uno specifico oggetto: deps_save_and_drop_dependencies - una funzione che ripristina le dipendenze, di uno specifico oggetto, nel database (prendendole dalla tabella in cui le avete salvate con lo script precedente): deps_restore_dependencies

Se vi interessano altre informazioni potete anche leggere il file: ReadMe.txt, che avevo personalmente scritto.

Utilizzo Poniamo il caso che dobbiate modificare il data type di un campo in una tabella. PostgreSQL vi bloccherà dicendo che esistono una o più viste dipendenti.

Con il nuovo metodo, potrete semplicemente aprire il Query Tool e scrivere:

select deps_save_and_drop_dependencies('nome_schema','nome_tabella')

Ho scritto “nome_tabella”, ma ricordo che si può benissimo usare una vista come fonte. Successivamente eseguire lo script.

Dopo aver eseguito lo script, le dipendenze scompariranno dal database e saranno inserite nella tabella prima creata. Da qui l'utente può eseguire le modifiche che gli interessano: nel nostro esempio modificare il data type. Dopo la modifica l'utente dovrà tornare nel Query Tool e digitare:

select deps_restore_dependencies('nome_schema','nome_tabella')

Successivamente le dipendenze saranno ripristinare nel database e senza aver salvato nessuno script

in nessun foglio di Notepad.

Possibili Problemi Va specificato un possibile problema: la funzione non fa magie, ma riscrive semplicemente gli script che dipendono da un oggetto al posto vostro. Quindi se avete una tabella dove avete un campo character varying e necessitate di trasformarlo in un numeric, state bene attenti che le viste dipendenti non abbiano alcuna: “where a = b”, dove a è il campo che volete modificare (character varying) e b un altro campo character varying. Altrimenti, se avete fatto la modifica, dopo aver eseguito la funzione di ripristino delle dipendenze potreste beccarvi l'errore: ERROR: ERRORE: l'operatore non esiste: numeric = character varying.

C'è comunque non modo per risolverlo, basterà entrare nella tabella deps_saved_ddl, cercare la riga dove si trova lo script della vista incriminata e modificarla manualmente, ossia togliendo quella where o modificandola in modo da farla funzionare. Insomma nulla di impossibile ma fa sempre bene saperlo.