
Outil psql

Licence GNU FDL - Version 1.5

- ❖ Programme interactif en ligne de commande
- ❖ Fait partie de l'application PostgreSQL
- ❖ Très puissant, stable
- ❖ Permet d'exécuter des commandes SQL depuis le Shell
- ❖ Comportement identique sur toutes les plateformes
- ❖ Commandes interactives ou par fichiers





PSQL: Connection au serveur PostgreSQL

Forme générale:

```
psql [OPTIONS]... [DBNAME [USERNAME]]
```

Par exemple:

```
psql -h <host> -U <user> -d <database> -p <port>
```

ou

```
psql <database> <user>
```

ou si la base est sur localhost (socket Unix):

```
psql [-U <user>] <dbname>
```

Exécuter une commande SQL directement:

```
psql -d dbsig -c "SELECT * FROM my_table LIMIT 2"
```

Exécuter un fichier contenant des commandes:

```
psql -d dbsig -f sql_stream.sql
```

(possibilité de rajouter l'option -a)

Rediriger le résultat dans un fichier

```
psql -d dbsig -f sql_stream.sql -o /tmp/out.txt
```

```
cat /tmp/out.txt
```

PSQL: Lister les bases existantes

- ❖ Via la commande psql

```
psql <options> -l
```

- ❖ En mode interactif

```
\l
```

PSQL: Lister les tables de la base

- ❖ Lister les tables, séquences, index ou vue de la base :
`\d`
- ❖ Lister uniquement les tables de la base courante:
`\dt`
- ❖ Lister les colonnes d'une ou plusieurs table ou vue
`\d nom_de_la_table`
- ❖ Afficher des informations liées à une séquence ou index
`\d nom_de_la_sequence`

PSQL: Autres commandes en mode interactif

❖ Lister les utilisateurs de la base courante:

`\du`

❖ Lister les index de la base courante:

`\di`

❖ Lister les séquences de la base courante:

`\di`

❖ Lister les schémas de la base courante:

`\dn`



- ❖ Executer une commande Shell depuis psql
`\! <command shell>`
- ❖ Executer un fichier de commande SQL depuis psql:
`\i <path_filesystem_vers_fichier_SQL>`
- ❖ Rediriger la sortie des requêtes SQL vers un fichier:
`\o <path_filesystem de redirection>`

- ❖ options/aide du programme:

`psql --help`

- ❖ En mode interactif:

`\?` (commandes psql)

`\h [SQL Command]` (pour le langage SQL)

- ❖ Par défaut les résultats sont alignés
- ❖ Désactiver l'alignement des résultats:
 - ❖ `SELECT * FROM mytable LIMIT 2;`
 - ❖ `\a`
 - ❖ `SELECT * FROM mytable LIMIT 2;`
- ❖ Particulièrement indiqués sur des enregistrement volumineux (i.e: géométries polygonales...)
- ❖ Possibilité d'utiliser l'option `-A` lors de la connection `psql`

- ❖ Afficher l'encoding courant du client PSQL:

`\encoding`

- ❖ Modifier l'encoding du client PSQL:

`\encoding latin1`

- ❖ Valeurs courantes:

`SQL_ASCII`

`UTF8`

`LATIN1`

- ❖ Mesure le temps d'exécution d'une commande
- ❖ Activer/désactiver le timer

`\timing`

- ❖ Exécute la requête et indique le temps nécessaire

```
postgres=# select count(*) from spatial_ref_sys;
```

```
-[ RECORD 1 ]
```

```
count | 3749
```

```
postgres=# \timing
```

```
Timing is on.
```

```
postgres=# select count(*) from spatial_ref_sys;
```

```
-[ RECORD 1 ]
```

```
count | 3749
```

```
Time: 5.101 ms
```

- ❖ Lister les variables PostgreSQL:

```
show shared_buffers;  
show all;
```

- ❖ Changer les variables d'execution via SET
 \help set (pour l'aide sur la commande)

- ❖ Exemple: changer le search_path

```
set search_path to work, public, "$user";
```

- ❖ Se croire (à tort) dans un shell et lancer des commandes système
- ❖ Et son pendant: se croire (à tort) connecté à PostgreSQL et lancer des commandes SQL
- ❖ Faire un SELECT, sans utiliser LIMIT sur une grosse table (interrompre avec Ctrl+C (voire Ctrl+Z))
- ❖ Oublier le « ; » final dans une commande SQL (et ne pas comprendre pourquoi la requête ne s'exécute pas)
- ❖ Corollaire: ne pas respecter les fermetures de parenthèses ou guillemets