

# Bases de Données - TP 1

## EIDD

---

### Installation et requêtes basiques SQL

---

#### 1 Prérequis

Disposer d'une installation Linux sur son poste de travail pour laquelle on a les droits « root » ou sudoers (commande sudo).

#### 2 S'inscrire au cours sur Moodle

Si vous n'êtes pas déjà inscrit au cours sur Moodle, inscrivez-vous au cours intitulé «IE1BY020 Bases de Données » sur <https://moodle.u-paris.fr/>.

#### 3 Installation de PostgreSQL

Pour installer PostgreSQL suivez les instructions contenues dans le fichier :  
Installation\_PostgreSQL.txt

#### 4 Travailler sur le serveur

Après l'installation vous disposez du client psql pour interagir avec le serveur de la base de données.

##### Linux

Vous devez ensuite éventuellement lancer le serveur

```
$service postgresql status
```

Si cette commande renvoie : **down**

Alors, démarrer le serveur :

```
$sudo service postgresql start
```

Créer un utilisateur ayant pour identifiant votre login dans la base de données :

```
$sudo -u postgres createuser -s $(whoami)
```

Créer une base de données ayant pour nom votre login

```
$createdb $(whoami)
```

Ici le serveur est local, il écoute sur le port officiel (5432), il suffit de préciser la base sur laquelle vous souhaitez travailler, en exécutant la commande :

```
$psql -d base login
```

Dans votre cas :

```
$psql -d $(whoami) $(whoami)
```

Lors de la première connexion, changer votre mot de passe, tapez (en remplaçant login par votre login et mdp par votre nouveau mot de passe :

```
ma_base#ALTER USER login WITH PASSWORD 'mdp';
```

##### Windows

Sur Windows, le serveur est déjà lancé et psql est dans le menu de vos applications :  
PostgresNNN/SQL Script(psql)

*Serveur (localhost) : Touche [entrée]  
Database (postgres) : Touche [entrée]  
Port : 5432 : Touche [entrée]  
Utilisateur (postgres) : Touche [entrée]*

# Windows a créé par défaut un super utilisateur « postgres » avec le mot passe que vous avez dû saisir à l'installation.

*Mot de passe : Celui saisi lors de l'installation*

Créer ensuite une base de données, ne pas mettre les <, > simplement remplacer <login> par votre login:

*ma\_base#CREATE DATABASE <login>;*

Connectez-vous à votre base  
*ma\_base#\c <login>*

### **Linux et Windows**

Le prompt *ma\_base#* signifie que vous êtes connecté avec un super utilisateur à la base de données <ma\_base> qui devrait être votre login

Il est important de changer le schéma par défaut au début de chaque connexion à la base. En effet à la connexion vous êtes sur le schéma "public" de la base, accessible à tous.

Pour créer un schéma ne pas mettre les <, > simplement remplacer <login> par votre login:

*#create schema sch\_<login>;*

Vérifier la création du schéma  
*#\dn*

Pour changer le schéma par défaut :  
*#SET search\_path TO sch\_<login> ;*

Vérifiez le schéma courant :  
*#SHOW search\_path ;*

Après la commande set search\_path to sch\_login toute nouvelle table sera créé dans ce schéma.

PostgreSQL n'est (à priori) pas sensible à la casse des lettres (le fait que ce soit une majuscule ou une minuscule) dans les commandes. Vous pouvez utiliser la tabulation pour obtenir une complétion automatique. Voici quelques commandes psql utiles.

=====  
Commandes de bases de psql  
=====

- Démarrer : psql
- Lister le contenu du répertoire courant : \! ls
- Changer de répertoire (se déplacer dans le dossier TP1 par exemple) : \cd TP1 ou \!  
cd ...

Attention, il y a plusieurs commandes différentes sous Linux et Windows. Par exemple :

\! dir     sous Windows  
au lieu de  
\! ls     sous Linux

Mais vous pouvez toujours vous déplacer dans psql avec \cd

- Exécuter un fichier sql se trouvant dans le répertoire courant (c'est à dire qui est visible lorsqu'on fait \! ls) : \i TP1.sql
- Si le fichier TP1.sql que vous voulez charger n'est pas dans le répertoire courant, vous pouvez soit vous déplacer dans ce répertoire avec la commande \cd expliquée plus haut, soit entrer le chemin complet. Par exemple, si je suis dans un répertoire qui contient TP1 et que TP1 contient TP1.sql, je fais soit :  
`\i TP1/TP1.sql`  
soit je me déplace dans TP1 avant d'exécuter le fichier :  
`\cd TP1`  
`\i TP1.sql`
- Au lieu d'entrer vos requêtes dans le terminal, il vaut mieux les mettre dans un fichier tp1\_mysql.sql par exemple, puis exécuter le fichier dans psql :  
`\i tp1_mysql.sql`
- Si vous ne voulez pas que tout le fichier tp1\_mysql.sql soit exécuté (par exemple vous ne voulez plus voir s'afficher les réponses aux questions que vous avez déjà résolues), il suffit d'entourer tout le bloc de texte dans tp1\_mysql.sql à faire disparaître par /\* et \*/
- Afficher les tables en mémoire : \dt
- Afficher des informations sur une table "maTable" : \d+ maTable
- Supprimer toutes les tables en mémoire : drop schema public cascade;

\help pour obtenir de l'aide

\? donne toutes les commandes

\h ou \h mot\_cle donne une aide sur les commandes SQL

\i commandes.sql permet d'importer un fichier de commandes SQL. Ce script doit bien sûr se trouver dans le même répertoire que celui dans lequel vous étiez lorsque vous avez lancé PostgreSQL. Nous vous encourageons à écrire toutes vos requêtes dans un fichier pour garder une trace de votre travail.

\q pour quitter PostgreSQL

\l lister les bases de données

\c ma\_base pour se connecter à une base de données

\dn lister les schemas

\d pour obtenir de l'information sur vos tables (leur définition)

\d matable pour obtenir de l'information sur une table particulière

\echo message permet d'afficher un message, par exemple le numéro de la question à laquelle va répondre la requête ;

\e édite la dernière commande (mode vi ou emacs)

\! ls effectuer la commande ls (marche pour toute commande Unix)

Sur votre machine personnelle, vous pouvez également installer le client pgcli, qui ajoute par-dessus psql des fonctions un peu plus sympathiques : complétion automatique, coloration de la syntaxe etc. Cependant, psql étant le client officiel, il vous sera plus facile de trouver de l'aide sur internet concernant ce client.

\cd change de répertoire

flèche haut revenir à la commande précédente ;

CTRL-a aller au début de la ligne ; CTRL-e aller à la fin de la ligne ; CTRL-d effacer le caractère suivant ;

CTRL-k effacer tout jusqu'à la fin de la ligne.

## 5 Mise en place du TP

Pour chaque TP, créez un répertoire BD/TP<i>. Placez-vous dans ce répertoire puis lancez psql. Toujours dans ce répertoire, créez et éditez un script tp<i>\_mysql.sql qui contient les commandes demandées, plutôt que de taper les commandes directement sous psql. Pour chaque requête, mettre par exemple \echo Requête <i>: suivi de la requête SQL. Soignez bien l'indentation des requêtes pour qu'elles soient lisibles. En travaillant ainsi, vous gagnerez du temps, et aurez une solution de votre TP qui pourra vous être utile au moment des révisions.

## 6 La base de données Livraison

La base représente des magasins et les usines qui les approvisionnent, ainsi que les produits livrés aux magasins. Un même produit peut être fabriqué par différentes usines. Il y a quatre tables : La table **produit** donne la référence (unique) du produit, son nom, sa couleur et son poids en kilos. La table **usine** donne la référence (unique) de l'usine, son nom et la ville où elle est située. La table **magasin** donne la référence (unique) du magasin, son nom et la ville où il est situé. La table **provenance** indique la quantité livrée pour chaque produit commandé et livré à un magasin par une usine.

### Chargement de la base

Chargez d'abord le fichier **tp1-data.sql** qui se trouve sur Moodle. Pour cela, chargez sous psql ce fichier qui créera les tables et les remplira. Pour cela tapez dans psql :

Si vous n'avez pas encore créé de schéma pour gérer l'ensemble de vos TP :  
Créer un schéma (ne pas mettre les <, > simplement remplacer <login> par votre login) :  
*#create schema sch\_<login>;*

Vérifier la création du schéma  
*#\dn*

Pour changer le schéma par défaut :  
*#SET search\_path TO sch\_<login> ;*

Vérifiez le schéma courant :  
*#SHOW search\_path ;*

*Puis exécution du fichier*  
*\i chemin/vers/répertoire/travail/tp1-data.sql*

(Remplacez chemin/vers/répertoire/travail par l'emplacement adéquat dans votre système de fichiers).

Vous pouvez regarder ce qu'il y a dans le fichier tp1-data.sql pour avoir une idée de comment créer des tables et comment les remplir. Nous allons voir cela en détail plus tard.

Le contenu de toutes les tables :

**produit :**

ref_prod	nom_prod	couleur	poids
1	tabouret	rouge	5
2	evier	bleu	65
3	bureau	jaune	45
4	lampe a petrole	vert	15
5	ordinateur	rouge	10
6	telephone	bleu	8
7	tabouret	violet	1
8	evier	bleu	65
9	tabouret	orange	3
10	lampe halogene	rose	11
11	lampe a souder	noir	3
12	telephone	bleu	2

13	casse-noix	vert	1
14	casse-pied	marron	55
15	casse-oreille	violet	15

usine :

ref_usine	nom_usine	ville
109	martin	Nantes
189	leroux	Marseille
213	dupont	Bordeaux
402	peugeot	Toulouse
200	peugeot	Marseille
302	rover	Londres

magasin :

ref_mag	nom_mag	ville
14	Stock10	Paris
16	JaiTout	Marseille
18	EnGros	Bordeaux
20	PrixBas	Toulouse
22	BasPrix	Marseille
24	DuBon	Lyon
26	DuBeau	Toulouse
28	BasDeGamme	Dublin
30	PasCher	Lyon
32		Rennes
34	NULL	Metz
36		Nantes

provenance :

ref_prod	ref_usine	ref_mag	quantite
1	109	14	80
1	109	16	100
1	302	16	213
2	189	30	213
3	402	14	315
4	200	18	985
5	302	20	858
6	213	16	315
6	109	22	458
7	109	16	213
8	302	16	2000
9	189	30	175
10	402	14	100
11	109	16	750
11	302	16	100
12	189	30	315
12	200	16	589
12	189	22	213
13	402	14	499
14	109	18	213
15	189	20	1958
15	189	16	333

## 6.1 Exercice 1 – Requêtes

Affichez les informations suivantes à l'aide de requêtes dans PostgreSQL. Quand vous le pouvez, traduisez ensuite ces requêtes en algèbre relationnelle.

1. la définition de chacune des tables que vous venez de créer ;
2. le contenu de chaque table ;
3. les villes où il y a une usine ; Comment éviter les doublons ?
4. les noms et les couleurs de chaque produit ;
5. la référence et la quantité de chaque produit livré au magasin ayant pour référence 14 ;
6. la référence, le nom et la ville de toutes les usines de Marseille ;
7. les références des magasins qui sont approvisionnés par l'usine de référence 109 en produit de référence 1 ;
8. la référence et le nom de tous les produits rouges ;
9. le nom des magasins dans une ville qui commence avec la lettre 'L' ;
10. la référence et le nom de tous les produits dont le nom commence par "casse" ;
11. la référence des magasins auxquels on livre quelque chose. Faites la requête sans préciser DISTINCT puis avec ;
12. les noms et les couleurs des produits qui pèsent entre 15 et 45 ;
13. les noms des produits jaunes ou bleus qui pèsent moins que 20 ;
14. les noms des produits jaunes ou des produits bleus qui pèsent moins que 20 ;  
Quelle est la différence avec la requête précédente ?
15. les lampes et les produits qui pèsent plus que 30 ;
16. les références des magasins qui ont le nom "NULL" ;
17. les références des magasins qui n'ont pas de nom ;
18. les références des produits fabriqués par l'usine martin de Nantes ; répondez d'abord en utilisant deux requêtes successives (requêtes monotables), ensuite une seule (jointure de deux tables) ;
19. les noms des usines qui sont dans une ville où il y a aussi un magasin ;
20. les références et les quantités des produits livrés aux magasins dont le nom commence avec "P" ;
21. les noms des usines où on fabrique des ordinateurs ;
22. les noms des produits dont le poids correspond à une référence ; (par exemple tabouret)