

Bases de Données - TP 1

Licence 3 d'Informatique 2023-2024

Requêtes simples en SQL

1 S'inscrire au cours sur Moodle

Si vous n'êtes pas déjà inscrit au cours sur Moodle, inscrivez-vous au cours intitulé «**IF16Y020 Bases de Données**» sur <https://moodle.u-paris.fr/>.

2 Connexion au serveur PostgreSQL

Les outils de gestion de bases de données que nous allons utiliser dans ce cours - PostgreSQL <http://https://www.postgresql.org/>, “the world’s most advanced open source database” - se trouvent sur un serveur distant. Vous pouvez accéder à la base de données depuis les salles de TP ou bien de chez vous en vous connectant en SSH. Sous Linux, OSX (Mac) et en salles de TP, il vous suffit de lancer un terminal :

1. Connectez vous sur un poste de travail (éventuellement en tant qu’invité en salle de TP).
2. Lancez une fenêtre terminal.
3. Dans cette fenêtre, tapez

```
ssh login@nivose.informatique.univ-paris-diderot.fr
```

ou bien si vous êtes sur une machine de salle TP

```
ssh nivose
```

en remplaçant `login` par votre login afin de vous connecter au serveur.

4. Entrez ensuite votre mot de passe lorsque le serveur vous le demande et validez avec la touche Entrée. Vous êtes désormais connecté au serveur.

Sous Windows, il faut installer un logiciel capable de se connecter en SSH au serveur : PuTTY <http://www.putty.org/>. Après avoir téléchargé et exécuté PuTTY, procédez aux réglages suivants :

1. Dans la case “Host Name (or IP address)”, inscrivez :

```
login@nivose.informatique.univ-paris-diderot.fr
```

en remplaçant `login` par votre login.
2. Vérifiez que le champ “Port” contient bien 22 et que SSH est sélectionné dans le champ “Connection type”.
3. Cliquez sur le bouton “Save” pour ne pas avoir à retaper ces informations à chaque fois.
4. Cliquez enfin sur “Open”. Une fenêtre noire s’affiche et vous demande votre mot de passe. Entrez votre mot de passe. Vous êtes désormais connecté au serveur.

3 Travailler sur le serveur

Sur la machine `nivose.informatique.univ-paris-diderot.fr` vous disposez du *client* `psql` pour interagir avec le *serveur* de la base de données.¹ Si votre configuration n'est pas à jour, il est possible qu'il vous faille ajouter `/usr/local/pgsql/bin` à votre `PATH` :

— Dans le cas où vous n'utilisez pas `bash` par défaut, vous pouvez le lancer avec `bash`.

— Modifiez votre profil :

```
cat >> .bash_profile
export PATH=/usr/local/pgsql/bin:$PATH
suivi de Ctrl-D pour sortir de cat.
```

— Saisissez `source ~/.bash_profile` pour que la modification prenne effet.

Ici le serveur est *local*, il écoute sur port *officiel*, il suffit de préciser le catalogue sur lequel vous souhaitez travailler (`base_cours`), en exécutant la commande :

```
psql -d base_cours login
```

Remplacez évidemment `login` par votre login (le même que celui avec lequel vous avez accédé à *nivose*). Vous devez ensuite saisir le mot de passe par défaut : `urquinaona`. Vous êtes désormais connecté à la base de données. Vous avez chacun un nom d'utilisateur (votre login) et un schéma pour travailler (`projet_login`).

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 :

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

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 changer le schéma par défaut : `SET search_path TO projet_login ;`

Pour connaître le schéma courant : `SHOW search_path ;`

Après la commande `set search_path to projet_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 :

`\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

`\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)

1. 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
fleche 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 ;
\pset pager off désactive l'utilisation du paginateur pour les requêtes et les affichages de
l'aide de psql (confortable lorsque le nombre de résultats attendus pour certaines requêtes
est élevé).

```

Mise en place

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*sol.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.

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 Pour cela chargez d'abord le fichier `TP1.1.sql` qui se trouve sur Moodle. Utilisez ensuite éventuellement la commande `scp` pour importer le fichier sur votre répertoire personnel si vous travaillez depuis chez vous sur le serveur distant. Ensuite, chargez sous `psql` ce fichier qui créera les tables et les remplira. Pour cela tapez dans psql :

```
\i chemin/vers/répertoire/travail/TP1.1.sql
```

(Remplacez `chemin/vers/répertoire/travail` par l'emplacement adéquat dans votre système de fichiers.) Le fichier ayant déjà été chargé dans le répertoire `/ens/zielonka/BD`, vous pouvez également utiliser la commande suivante :

```
\i /ens/zielonka/BD/TP1.1.sql
```

Vous pouvez regarder ce qu'il y a dans le fichier `TP1.1.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

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 avec la 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)