

## Requêtes simples en SQL

---

### S'inscrire au cours sur DidEL

Si vous n'êtes pas déjà inscrit au cours sur DidEL, connectez-vous à <http://didel.script.univ-paris-diderot.fr/> et inscrivez-vous au cours intitulé « **BDL3INFO** ».

### Utilisation de psql

PostgreSQL ([www.postgresql.org](http://www.postgresql.org)) est installé sur la machine **nivose**. Pour y accéder, il faut donc vous connecter à nivose :  
`ssh nivose` (utiliser votre mot de passe Unix).

Pour pouvoir appeler PostgreSQL, il faut ajouter `/usr/local/pgsql/bin` à votre `PATH`. Le mieux est de l'ajouter lors de la première séance à votre fichier `.bash_profile` pour l'avoir de manière permanente :

```
cat >> .bash_profile
export PATH=/usr/local/pgsql/bin:$PATH
```

(suivi de `Ctrl-D` pour sortir de `cat`). Enfin, il faut sortir de **nivose** (`exit`) et se reconnecter pour que la modification prenne effet. Si vous n'utilisez pas `bash` par défaut, vous pouvez le lancer avec `bash`.

Vous avez chacun un nom d'utilisateur et une base de données : le nom d'utilisateur est votre login sous Unix, le nom de la base est **base**, et le mot de passe est **BD2013**. Pour entrer dans votre base, tapez :

```
psql nomUtilisateur nomBase
ou plus simplement
psql base
et donnez le mot de passe.
```

Pour changer votre mot de passe, tapez :  
`alter user votre_login with password 'NouveauPass';`  
Attention : n'utilisez pas le même mot de passe que pour votre login unix, et n'oubliez pas le nouveau mot de passe !

Pour obtenir de l'aide sur les commandes propres à PostgreSQL, taper  
`<votre login> => \?`

Pour obtenir de l'aide sur les commandes SQL :  
`<votre login> => \h`

Pour exécuter une liste de commandes SQL sauvegardée dans un fichier `commandes.sql` :  
`<votre login> => \i commandes.sql`

Pour quitter PostgreSQL, il faudra taper  
`<votre login> => \q`

`psql` n'est pas sensible à la casse des lettres (le fait que ce soit une majuscule ou une minuscule). Vous pouvez utiliser la tabulation pour obtenir une complétion automatique.

Quelques autres commandes `psql` utiles :

`\d` pour obtenir de l'information sur vos tables ;

`\d matable` pour obtenir de l'information sur une table appelée « matable » ;  
`\echo message` permet d'afficher un message (utilisé surtout dans les scripts) ;  
`\e` éditer la dernière commande (mode vi ou emacs) ;  
`\! ls` effectuer la commande `ls` (marche pour toute commande Unix) ;  
`\cd` change le répertoire courant ;  
`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.

## 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*.sql qui contient les commandes demandées, plutôt que de taper les commandes directement sous `psql`. Pour chaque exercice, taper `\echo Exercice 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.

Pour lancer un script contenu dans un fichier appelé TP1.sql, tapez `\i TP1.sql` sous `psql`. Ce script doit se trouver dans le même répertoire que celui dans lequel vous êtes lorsque vous avez lancé `psql`.

## Les tournages de films

La ville de Paris met à disposition du public de nombreuses bases de données. Nous allons utiliser la base de tournages de films.<sup>1</sup> Malheureusement les données sont difficilement exploitables en l'état car les concepteurs n'ont pas appliqué certaines règles de bonne gestion des bases de données.

Nous vous fournissons pour ce TP une version nettoyée de la base de données, mais il reste encore un nombre important de problèmes que nous nous efforcerons de mettre en lumière au cours des prochaines séances.

La base que nous vous fournissons est composée de 4 tables. La table **Film** qui contient les titres des films et une référence vers son réalisateur ; la table **Realisateur** qui contient le nom et le prénom de chaque réalisateur ; la table **Lieu** qui contient le type de lieu (rue, avenue, etc.), l'adresse (nom de la rue), l'arrondissement (code postal), ainsi que la latitude et la longitude lorsque celles-ci sont disponibles. Enfin, la table **Evenement** contient les tournages. On y trouve : une référence vers le film, une référence vers le lieu de tournage, le cadre (intérieur ou extérieur), et les dates de début et de fin du tournage. Un film peut bien sûr faire l'objet de nombreux événements.

## Chargement de la base

Pour charger la base vous pouvez taper la commande suivante depuis psql :

```
\i /ens/laplante/BD6/creer-tournage.sql
```

---

1. Disponible à : <http://opendata.paris.fr/explore/dataset/tournagesdefilmsparis2011>.

## Exercices

1. Quels sont les identifiants de la rue Watt dans le 13<sup>e</sup> arrondissement ? Afficher toutes les informations sur les lieux trouvés. Indication : il y a plusieurs variantes dans la base, ce qui rend les données difficilement exploitables. Pour trouver toutes les variantes vous pouvez utiliser la recherche de motif avec l'opérateur de comparaison de chaînes LIKE et le joker '%' qui signifie n'importe quelle chaîne y compris la chaîne vide. Par exemple, pour trouver tous les noms qui contiennent la lettre 'A' on écrira `WHERE nom LIKE '%A%'` dans la requête SQL.
2. Écrivez une requête qui retourne uniquement les identifiants (pas les autres attributs) des lieux de tournage rue Watt.
3. Quels films ont été tournés rue Watt ? Si on ne fait que des requêtes sur une table, on doit trouver d'abord dans la table **Evenement** l'identifiant du ou des films en utilisant les résultats de la question précédente. Puis trouver le ou les titres dans la table **Film**. (Nous verrons comment faire la même chose en une requête sur plusieurs tables lors du prochain TP.) Indication : si vous avez trouvé 3 identifiants de lieu, vous pouvez utiliser trois requêtes avec la clause `WHERE numlieu = ...`, ou vous pouvez obtenir tous les résultats en utilisant une clause `WHERE numlieu IN (... , ...)`.
4. Qui sont les réalisateurs des films tournés rue Watt ?
5. Toutes les adresses dans la base sont-elles tout en majuscules ? Indication : la fonction `UPPER(...)` convertit les chaînes en majuscules.
6. Trouver tous les lieux de tournages qui contiennent **IENA**. Remarquez-vous des anomalies qui rendent la base difficile à exploiter si on voulait trouver les tournages ayant eu lieu à cet endroit ?
7. On souhaite trouver tous les films tourné par Maïwenn. Combien de fois apparaît-elle dans la table **realisateurs** ? Indication : avec l'opérateur de comparaison LIKE, le joker '\_' est mis en correspondance avec exactement une lettre (n'importe laquelle).
8. Trouver tous les réalisateurs dont le prénom est égal à la chaîne vide. Constatez-vous des anomalies ?
9. Trouver TOUS les identifiants associés à Olivier Nakache et à Frédéric Berthe. (Cherchez bien.) Remarquez-vous des anomalies qui rendent la base difficile à exploiter si on voulait trouver tous leurs films ?
10. Trouver tous les lieux pour lesquels l'arrondissement n'est pas valué. Indication : le résultat n'est pas le même si on écrit `= NULL` ou `IS NULL`. Lequel des deux donne le bon résultat ?
11. Trouver les tournages qui ont duré 4 jours. Indication : le type **DATE** permet de faire des calculs simples sur les dates. Par exemple, `datedebut+2` retourne la date correspondant au surlendemain de `datedebut`. Taper `SELECT CURRENT_DATE;` retourne la date du jour. `SELECT CURRENT_DATE+14;` retourne la date dans 2 semaines. On peut aussi soustraire une **DATE** d'une autre **DATE** pour obtenir le nombre de jours entre les deux.
12. Afficher les tournages qui ont duré entre 3 et 7 jours. Indication : vous pouvez utiliser l'opérateur de comparaison `BETWEEN ... AND ...`.
13. Reprendre la requête précédente mais afficher : **numfilm**, **datedebut**, **datefin**, **nbjours**. Indication : utiliser un alias **AS nbjours** pour le résultat du calcul `datefin-datedebut`.
14. Reprendre la requête précédente mais trier sur (l'alias) **nbjours**. Indication : utiliser la clause `ORDER BY`.
15. Donner le nombre de tournages qui ont eu lieu sur 2 jours. Indication : utiliser `SELECT COUNT(*)` pour retourner le nombre de lignes trouvées.
16. Donner le nombre de jours du plus long tournage. Indication : utiliser la fonction d'agrégat **MAX** sur `datefin-datedebut`.
17. La dernière requête indique une anomalie dans les données, car les tournages ne durent jamais plus d'une semaine. Trouver tous les événements qui durent plus d'une semaine et afficher les résultats par ordre de nombre de jours. Tenter de donner une explication pour les anomalies constatées.

18. Tous les tournages ont-ils une date de début qui précède la date de fin ?

Nettoyer votre base en supprimant les tables créées à l'aide de l'ordre **DROP TABLE**. Vous devez les supprimer dans l'ordre suivant : **Evenement**, **Lieu**, **Film**, **Realisateur**.

## 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 commencer, chargez sous **psql** le fichier qui créera les tables et les remplira :

`\i /ens/laplante/BD6/TP1.sql`

Nous allons voir plus tard comment créer des tables et comment les remplir.

Voici le contenu de toutes les tables :

| Usine:   |                 |         |       | usine:    |            |           | provenance: |           |         |          |
|----------|-----------------|---------|-------|-----------|------------|-----------|-------------|-----------|---------|----------|
| ref_prod | nom_prod        | couleur | poids | ref_usine | nom_usine  | ville     | ref_prod    | ref_usine | ref_mag | quantite |
| 1        | tabouret        | rouge   | 5     | 109       | martin     | Nantes    | 1           | 109       | 14      | 80       |
| 2        | evier           | bleu    | 65    | 189       | leroux     | Marseille | 1           | 109       | 16      | 100      |
| 3        | bureau          | jaune   | 45    | 213       | dupont     | Bordeaux  | 1           | 302       | 16      | 213      |
| 4        | lampe a petrole | vert    | 15    | 402       | peugeot    | Toulouse  | 2           | 189       | 30      | 213      |
| 5        | ordinateur      | rouge   | 10    | 200       | peugeot    | Marseille | 3           | 402       | 14      | 315      |
| 6        | telephone       | bleu    | 8     | 302       | rover      | Londres   | 4           | 200       | 18      | 985      |
| 7        | tabouret        | violet  | 1     |           |            |           | 5           | 302       | 20      | 858      |
| 8        | evier           | bleu    | 65    | magasin:  |            |           | 6           | 213       | 16      | 315      |
| 9        | tabouret        | orange  | 3     | ref_mag   | nom_mag    | ville     | 6           | 109       | 22      | 458      |
| 10       | lampe halogene  | rose    | 11    |           |            |           | 7           | 109       | 16      | 213      |
| 11       | lampe a souder  | noir    | 3     | 14        | Stock10    | Paris     | 8           | 302       | 16      | 2000     |
| 12       | telephone       | bleu    | 2     | 16        | JaiTout    | Marseille | 9           | 189       | 30      | 175      |
| 13       | casse-noix      | vert    | 1     | 18        | EnGros     | Bordeaux  | 10          | 402       | 14      | 100      |
| 14       | casse-pied      | marron  | 55    | 20        | PrixBas    | Toulouse  | 11          | 109       | 16      | 750      |
| 15       | casse-oreille   | violet  | 15    | 22        | BasPrix    | Marseille | 11          | 302       | 16      | 100      |
|          |                 |         |       | 24        | DuBon      | Lyon      | 12          | 189       | 30      | 315      |
|          |                 |         |       | 26        | DuBeau     | Toulouse  | 12          | 200       | 16      | 589      |
|          |                 |         |       | 28        | BasDeGamme | Dublin    | 12          | 189       | 22      | 213      |
|          |                 |         |       | 30        | PasCher    | Lyon      | 13          | 402       | 14      | 499      |
|          |                 |         |       |           |            |           | 14          | 109       | 18      | 213      |
|          |                 |         |       |           |            |           | 15          | 189       | 20      | 1958     |
|          |                 |         |       |           |            |           | 15          | 189       | 16      | 333      |

## Exercices

Afficher :

- le contenu de chaque table ;
- les villes où il y a une usine ;
- les noms et les couleurs de chaque produit ;
- la référence et la quantité de chaque produit livré au magasin dont la référence est 14 ;
- la référence, le nom et la ville de toutes les usines de Marseille ;
- les références des magasins qui sont approvisionnés par l'usine de référence 109 en produit de référence 1 ;
- la référence et le nom de tous les produits rouges ;
- la référence et le nom de tous les produits dont le nom commence par « casse » ;
- la référence des magasins auxquels on livre quelque chose. Faites la requête sans préciser **DISTINCT** puis avec ;
- les noms des produits qui pèsent entre 15 et 45 ;
- les noms des produits jaunes ou bleus qui pèsent moins de 20 ;
- les noms des produits jaunes (sans restriction de poids), ou des produits bleus qui pèsent moins de 20 ;
- toutes les lampes, ainsi que les produits qui pèsent plus de 30.