

## TP n°4 : SQL - langage d'interrogation de BD

### Utilisation du SGBD PostGres

Pour commencer le TP :

1. **Si vous êtes à l'IUT**, ouvrir une fenêtre xterm. Placez vous dans le répertoire BDS1 puis tapez la commande suivante pour lancer **Postgres**  
`psql -h database-etudiants -d candidats`
2. **Si vous êtes chez et que vous vous connectez sur le serveur database-etudiants**, ouvrir une fenêtre xterm. Placez vous dans le répertoire BDS1 puis tapez la commande suivante pour lancer **Postgres**  
`psql -h database-etudiants.iut.univ-paris8.fr -d candidats -U votreLoginIUT -p 5432`
3. **Si vous travaillez en local sur votre PC**, ouvrir une fenêtre xterm. Placez vous dans le répertoire où vous avez sauvegardé les scripts SQL qu'il faudra exécuter (voir les exercices) puis tapez la commande suivante pour lancer **Postgres**  
`psql -h localhost -d votreLoginDeVotrePC`  
**ou**  
`psql -h localhost -d template1`

Quelques commandes utiles sous **PostgreSQL** :

commande	description
<code>\?</code>	help on slash commands
<code>\h</code>	Help on SQL commands
<code>\q</code>	quit
<code>\d <i>nom_table</i></code>	describe table
<code>\i <i>filename</i></code>	Execute filename

Sauvegarder toutes les commandes SQL tapées sous psql dans des fichiers **TP4\_VotreLogin.sql**, contenant les commandes que vous avez lancées et le résultat renvoyé par Postgres.

#### Exercice 1 : BD candidats

**Préambule pour ceux qui travaillent sur leur PC =>** charger le script `candidIUT.sql` et exécutez le dans un terminal par **`\i candidIUT.sql`**

Vérifier le contenu des tables **candidat**, **ville**, **baccalaureat**. Déduisez en le schéma relationnel.

- 1) Quels sont les noms et prénoms des candidats ?
- 2) Donne la liste des étudiants dont le nom est 'Chine'?
- 3) Quels sont les candidats ayant eu les 2 évaluations supérieures à 8?
- 4) Quelles sont les candidats ayant une évaluation valant 0 ou *null* ?
- 5) Quels sont les candidats qui n'ont eu qu'une évaluation (i.e. l'autre évaluation vaut *null*) ?
- 6) Quels sont les différents baccalauréats des candidats (sans doublons) ?
- 7) Quels sont les candidats qui ont un bac étranger (noté 'ETR') ?
- 8) Quelles sont les différentes villes (sans doublons) ?
- 9) Quelles sont les différentes villes des candidats (sans doublons) ?

- 10) Quelles sont les villes qui n'ont pas de candidats ?
- 11) Quels sont les bacs qui n'ont pas de candidats ?
- 12) Quels sont les candidats de la ville *L'HAY LES ROSES* ?

## Exercice 2 : Schéma soc

Avant de faire l'exercice 2, faites les commandes suivantes :

- connectez vous maintenant sur votre BD par la commande : **\c votreLogin**
- Téléchargez le script **usine.sql** sur Moodle et exécutez le par **\i usine.sql**
- **Attention**, les tables sont placées dans le schéma relationnel **soc** (il faudra préfixer le noms des tables par soc.)
- **vérifiez leur contenu** en tapant par exemple **select \* from soc.puf ;**

D'après le schéma relationnel vu lors du TD4 concernant des usines, des produits, des fournisseurs et la table associative PUF

Écrire en SQL les questions suivantes :

1. Quels sont les numéros, noms et villes de toutes les usines ?
2. Donner les numéros, noms et villes de toutes les usines situées à 'Londres'.
3. Quels sont les numéros des fournisseurs qui approvisionnent l'usine n°1 en produit n°1 ?
4. Quelles sont les noms et couleurs des produits livrés par le fournisseur n°1 ?
5. Quels sont les numéros des fournisseurs qui approvisionnent l'usine n°1 en un produit rouge ?
6. Quels sont les noms des fournisseurs qui approvisionnent une usine de Londres ou de Paris en un produit rouge ?
7. Quels sont les numéros des produits livrés à une usine de Londres par un fournisseur de Londres ?
8. Quels sont les numéros des produits livrés à une usine par un fournisseur de la même ville ?
9. Quels sont les noms des usines qui ont au moins un fournisseur qui n'est pas de la même ville ?
10. Quelles sont les fournisseurs qui approvisionnent à la fois l'usine n°1 et n°2 ?
11. Quels sont les noms des usines qui utilisent au moins un produit disponible chez le fournisseur n°3 (i.e. un produit qu'il livre mais pas nécessairement à cette usine) ?
12. Quelles sont les usines qui ne reçoivent aucun produit rouge d'un fournisseur londonien ?
13. Donner tous les triplets (VilleF, NP, VilleU) tels qu'un fournisseur de la première ville approvisionne une usine de la deuxième ville.
14. Même question qu'en 13) mais sans les triplets où les deux villes sont identiques.
15. Quelles sont les usines qui s'approvisionnent uniquement chez le fournisseur n°3 ?

## Exercice 3 : Jointure INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN et NATURAL JOIN

Activer le timer du requêteur SQL : **\timing**

Traiter les requêtes ci-dessous par jointure de colonne, puis par INNER JOIN, etc. et noter le temps de calcul nécessaire pour le requêteur :

1. Donner la liste des candidats (nom, prénom) avec l'intitulé de leur baccalauréat et le nom de la ville où ils habitent.
2. Quelles sont les différentes villes des candidats (sans doublons) ?
3. Pour chaque ville (trier selon l'ordre croissant), établir la liste des éventuels candidats.

## ANNEXE : Fonctions et opérateurs SQL pour le type chaîne

Fonction	Type renvoyé	Description	Exemple	Résultat
<i>chaîne</i>    <i>chaîne</i>	text	Concaténation de chaînes	'Post'    'greSQL'	PostgreSQL
<i>chaîne</i>    autre-que-chaîne ou autre-que-chaîne    <i>chaîne</i>	text	Concaténation de chaînes avec un argument non-chaîne	'Value: '    42	Value: 42
<b>length</b> ( <i>chaîne</i> )	int	Nombre de caractères de chaîne	length('jose')	4
<b>lower</b> ( <i>chaîne</i> )	text	Convertit une chaîne en minuscule	lower('TOM')	tom
<b>upper</b> ( <i>chaîne</i> )	text	Convertit une chaîne en majuscule	upper('tom')	TOM
<b>position</b> ( <i>substring</i> in chaîne)	int	Emplacement de la sous-chaîne indiquée	position('om' in 'Thomas')	3
<b>substring</b> ( <i>chaîne</i> [from int] [for int])	text	Extrait une sous-chaîne	substring('Thomas' from 2 for 3)	hom
<b>md5</b> ( <i>chaîne</i> )	text	Calcule la clé MD5 de chaîne et retourne le résultat en hexadécimal. <b>TRES UTILE POUR CODER MOT DE PASSE</b>	md5('abc')	900150983cd24fb0d6963f7d28e17f72
<b>replace</b> ( <i>chaîne</i> text, à partir de text, vers text)	text	Remplace dans chaîne toutes les occurrences de la sous-chaîne à partir de par la sous-chaîne vers.	replace('abcdefabcdef', 'cd', 'XX')	abXXefabXXef
<b>trim</b> ([leading   trailing   both] [ <i>caractères</i> ] from chaîne)	text	Supprime la plus grande chaîne qui ne contient que les caractères (une espace par défaut) à partir du début, de la fin ou des deux extrémités (respectivement leading, trailing,	trim(both 'x' from 'xTomxx')	Tom

Fonction	Type renvoyé	Description	Exemple	Résultat
		both) de la chaîne.		
<b>btrim</b> (chaîne text [, caractères text])	text	Supprime la chaîne la plus longue constituée uniquement de caractères issus de caractères (une espace par défaut) à partir du début et de la fin de chaîne.	btrim('xyxtrimyyx', 'xy')	trim