

ADMINISTRATION BDD - TP1 (Rappel sur l'interrogation des données et l'utilisation du langage SQL)

I. INSTALLATION MYSQL

Pour obtenir les dernières versions en cours de développement et celles de production, rendez-vous sur le site <http://www.mysql.com/downloads>, puis sélectionnez MySQL Community Server.

La première étape consiste à sélectionner votre plate-forme (qui va de Windows à Mac OS en passant par différentes éditions de Linux).

Pour pouvoir télécharger le logiciel et poster sur les forums officiels, vous devez vous enregistrer (si vous disposez déjà d'un compte Oracle, c'est le même). Dans le cas de Windows, le logiciel MySQL est fourni avec un fichier d'installation (MSI Installer) qu'il convient d'utiliser.

Après avoir accepté les termes de la licence, vous êtes invité à choisir le répertoire d'installation (par défaut Program Files\MySQL) et le type d'installation (par défaut Developer). Ensuite, une étape de mise à jour de composants et de vérification des prérequis s'opère, au cours de laquelle vous pouvez agir avant le récapitulatif des produits qui seront installés.

1. CONFIGURATION DU SERVEUR

La configuration de base inclut :

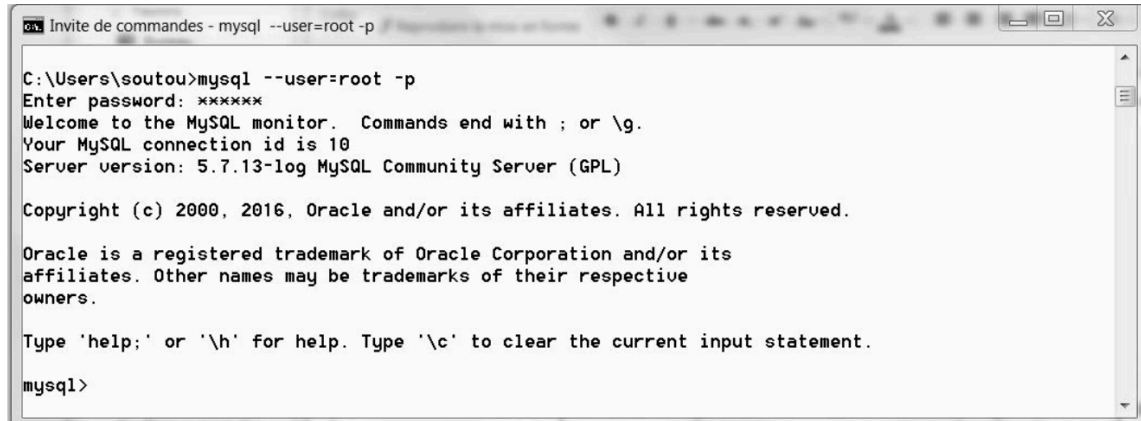
- le mot de passe de l'utilisateur root qui permet d'ajouter des nouveaux utilisateurs en leur accordant des rôles prédéfinis (administrateur, concepteur, importateur, sauvegarde, etc.) ;
- un service Windows dédié à MySQL (nommé par défaut MySQLxx, qu'il sera possible d'arrêter via le panneau de configuration) ;
- le port UDP d'écoute (par défaut 3306), vous pouvez ainsi ajouter une exception à votre pare-feu (option cochée par défaut). Suivant la version choisie, la configuration avancée vous permettra d'agir sur :
 - le type de votre serveur (machine de développement, serveur ou machine dédiée) ;
 - les répertoires qui contiendront les fichiers de trace ;
 - L'inclusion du chemin de l'exécutable mysql dans la variable d'environnement (path) ;
 - le mode comportemental du serveur par rapport à la syntaxe des instructions SQL ;
 - le type de base de données (multifonction, mode transactionnel ou pas) ;
 - nombre de connexions (15 par défaut) ;
 - le jeu de caractères (West European par exemple)

2. Démarrage de MySQL

Le tableau ci-dessous liste les commandes de base relatives au démarrage du serveur.

Dans une fenêtre de commande Windows, Linux (ou autre), lancez l'interface en ligne de commandes en connectant l'utilisateur root avec le mot de passe spécifié lors de l'installation :

```
mysql --user=root -p
```



Il vous est également possible de spécifier la machine et la base de données souhaitées.

```
mysql --user=util --host=localhost -p --database=boutil
```

3. Options de bases

Le tableau suivant liste les arguments pouvant être communiqués lors de la connexion

Option	Commentaire
--help ou -?	Affiche les options disponibles, l'état des variables d'environnement et rend la main.
--batch ou -B	Toute commande SQL peut être lancée dans la fenêtre de commandes système sans pour autant afficher l'invite. Les résultats (colonnes) sont séparés par des tabulations.
--database=nomBD ou -D nomBD	Sélection de la base de données à utiliser après la connexion.
--host=nomServeur ou -h nomServeur	Désignation du serveur.

--html ou -H	Formate le résultat des extractions en HTML.
--one-database ou -O	Restreint les instructions à la base de données spécifiée initialement.
-p	Demande le mot de passe sans l'employer en tant que paramètre.
--password=motdePasse	Transmission du mot de passe de l'utilisateur à connecter. Évitez le plus possible cette option et préférez-lui la précédente.
--prompt=parametre	Personnalise l'invite de commande (par défaut mysql>).
--silent ou -s	Configure le mode silence pour réduire les messages de MySQL.
--skip-column-names ou -N	N'écrit aucun en-tête de colonne pour les résultats d'extraction.
--table ou -t	Formate le résultat des extractions en tables à en-tête de colonne (par défaut dans le mode interactif).
--tee=cheminNomFichier	Copie la trace de toute la session dans le fichier spécifié.
--user=utilisateur ou -u utilisateur	Désigne l'utilisateur devant se connecter.
--verbose ou -v	Mode verbeux pour afficher davantage de messages du serveur.
--version ou -V	Affiche la version du serveur et rend la main.
--vertical ou -E	Affiche les résultats des extractions verticalement (non plus en lignes horizontales).
--xml ou -X	Formate le résultat des extractions en XML. Les noms de balises générées sont <resultset> pour la table résultat, <row> pour chaque ligne et <field> pour les colonnes.

Ces options peuvent se combiner en les séparant simplement par un espace (exemple : mysql --tee=D:\\dev\\sortiemysql.txt --database=bdsoutou va se connecter anonymement à la base bdsoutou en inscrivant le contenu de la trace de la session dans le fichier sortiemysql.txt situé dans le répertoire D:\\dev) dans le cas de windows.

1. L'interface de commande

L'interface en ligne de commande se lance grâce à l'exécutable mysql. Toutes ces commandes sont à entrer dans le prompt/terminal de MySQL une fois connecté au serveur.

Utiliser une base de données existante

```
USE nom_de_la_base;
```

Créer une base de données

```
CREATE DATABASE nom_de_la_base;
```

Liste des bases de données

```
SHOW DATABASES;
```

Supprimer une base de données

```
DROP DATABASE nom_de_la_base;
```

Importer une base de données située dans le répertoire courant (base de donnée exportée en .sql)

```
SOURCE fichier.sql;
```

III. Manipulation des données

Nous commençons cette section par rappeler comment créer une table.

La syntaxe générale pour créer une table est la suivante :

```
CREATE TABLE nom_de_la_table

(
    colonne1 type_donnees,
    colonne2 type_donnees,
    colonne3 type_donnees,
    colonne4 type_donnees
)
```

Dans cette requête, 4 colonnes ont été définies. Le mot-clé “type_donnees” sera à remplacer par un mot-clé pour définir le type de données (INT, DATE, TEXT ...). Pour chaque colonne, il est également possible de définir des options telles que (liste non-exhaustive) :

Exemple

Imaginons que l’on souhaite créer une table utilisateur, dans laquelle chaque ligne correspond à un utilisateur inscrit sur un site web. La requête pour créer cette table peut ressembler à ceci :

```
CREATE TABLE utilisateur
(
    id INT PRIMARY KEY NOT NULL,
    nom VARCHAR(100),
    prenom VARCHAR(100),
    email VARCHAR(255),
    date_naissance DATE,
    pays VARCHAR(255),
    ville VARCHAR(255),
    code_postal VARCHAR(5),
    nombre_achat INT
)
```

Voici des explications sur les colonnes créées :

- **id** : identifiant unique qui est utilisé comme clé primaire et qui n’est pas nulle
- **nom** : nom de l’utilisateur dans une colonne de type VARCHAR avec un maximum de 100 caractères au maximum
- **prenom** : idem mais pour le prénom
- **email** : adresse email enregistré sous 255 caractères au maximum
- **date_naissance** : date de naissance enregistré au format AAAA-MM-JJ (exemple : 1973-11-17)
- **pays** : nom du pays de l’utilisateur sous 255 caractères au maximum
- **ville** : idem pour la ville
- **code_postal** : 5 caractères du code postal
- **nombre_achat** : nombre d’achat de cet utilisateur sur le site

Pour plus d’exemple, veuillez consulter la page : <https://sql.sh/cours/create-table>

1. Présentation de la base de données

Une entreprise désire gérer son parc informatique à l’aide d’une base de données. Le bâtiment est composé de trois étages. Chaque étage possède son réseau (ou segment distinct) Ethernet. Ces réseaux traversent des salles équipées de postes de travail. Un poste de travail est une machine sur laquelle sont installés certains logiciels.

Quatre catégories de postes de travail sont recensées (stations Unix, terminaux X, PC Windows et PC NT). La base de données devra aussi décrire les installations de logiciels.

Les noms et types des colonnes sont les suivants :

Colonne	Commentaire	Type
indIP	trois premiers groupes IP (exemple : 130.120.80)	VARCHAR(11)
nomSegment	nom du segment	VARCHAR(20)
etage	étage du segment	TINYINT(1)
nSalle	numéro de la salle	VARCHAR(7)
nomSalle	nom de la salle	VARCHAR(20)
nbPoste	nombre de postes de travail dans la salle	TINYINT(2)
nPoste	code du poste de travail	VARCHAR(7)
nomPoste	nom du poste de travail	VARCHAR(20)
ad	dernier groupe de chiffres IP (exemple : 11)	VARCHAR(3)
typePoste	type du poste (UNIX, TX, PCWS, PCNT)	VARCHAR(9)
dateIns	date d'installation du logiciel sur le poste	dateTime
nLog	code du logiciel	VARCHAR(5)
nomLog	nom du logiciel	VARCHAR(20)
dateAch	date d'achat du logiciel	dateTime
version	version du logiciel	VARCHAR(7)
typeLog	type du logiciel (UNIX, TX, PCWS, PCNT)	VARCHAR(9)
prix	prix du logiciel	DECIMAL(6,2)
numIns	numéro séquentiel des installations	INTEGER(5)
dateIns	date d'installation du logiciel	TIMESTAMP
delai	intervalle entre achat et installation	SMALLINT
typeLP	types des logiciels et des postes	VARCHAR(9)
nomType	noms des types (Termiaux X, PC Windows...)	VARCHAR(20)

2. Travail à réaliser

Exercice 1 :

Écrire puis exécuter le script SQL (que vous appellerez **creParc.sql**) de création des tables avec leur clé primaire (en gras dans le schéma suivant) et les contraintes suivantes :

- Les noms des segments, des salles et des postes sont non nuls.
- Le domaine de valeurs de la colonne *ad* s'étend de 0 à 255.
- La colonne *prix* est supérieure ou égale à 0.
- La colonne *dateIns* est égale à la date du jour par défaut.

La composition des tables est la suivante :

Segment					
indIP	nomSegment	etage			
Salle					
nSalle	nomSalle	nbPoste	indIP		
Poste					
nPoste	nomPoste	indIP	ad	typePoste	nSalle
Logiciel					
nLog	nomLog	dateAch	version	typeLog	prix
Installer					
nPoste	nLog	numIns	dateIns	delai	
Types					
typeLP	nomType				

Exercice 2 : structure des tables

Écrire puis exécuter le script SQL (que vous appellerez **descParc.sql**) qui affiche la description de toutes ces tables (en utilisant des commandes **DESCRIBE**). Comparer le résultat obtenu avec le schéma ci-dessus.

Vous pouvez utiliser easyphp pour visualiser le schéma.

Exercice 3 : Destruction des tables

Écrire puis exécuter le script SQL de destruction des tables (que vous appellerez **dropParc.sql**). Lancer ce script puis celui de la création des tables à nouveau.

Exercice 4 : Interrogation des données

Les objectifs de ces exercices sont les :

- ⇒ de créer dynamiquement des tables et leurs données ;
- ⇒ d'écrire des requêtes monotables et multitable ;
- ⇒ de réaliser des modifications synchronisées ;
- ⇒ de composer des jointures et des divisions.

Écrire le script **créaDynamique.sql** permettant de créer les tables **Softs** et **PCSeuls** suivantes (en utilisant la directive **AS SELECT** de la commande **CREATE TABLE**). Vous ne poserez aucune contrainte sur ces tables. Penser à modifier le nom des colonnes.

Softs		
nomSoft	version	prix

PCSeuls					
nP	nomP	seg	ad	typeP	salle

La table **Softs** sera construite sur la base de tous les enregistrements de la table **Logiciel** que vous avez créée et alimentée précédemment. La table **PCSeuls** doit seulement contenir les enregistrements de la table **Poste**, qui sont de type 'PCWS' ou 'PCNT'. Vérifier :

```
SELECT * FROM Softs;
SELECT * FROM PCSeuls;
```

Requêtes monotables

Ecrire le script requêtes.sql permettant d'extraire, à l'aide d'instructions **SELECT**, les données suivantes :

- Type du poste 'p8'.
- Noms des logiciels 'UNIX'.
- Noms, adresses IP, numéros de salle des postes de type 'UNIX' ou 'PCWS'.
- Même requête pour les postes du segment '130.120.80' triés par numéros de salles décroissants.
- Numéros des logiciels installés sur le poste 'p6'.
- Numéros des postes qui hébergent le logiciel 'log1'.
- Noms et adresses IP complètes (ex : '130.120.80.01') des postes de type 'TX' (utiliser la fonction de concaténation).

Pour pouvoir traiter ces requêtes, penser à exécuter le script evolution.sql.

Vos erreurs peuvent apparaître (duplicaton des PK...), il sera donc possible que vous analysiez et/ou corrigiez vos données.

Fonctions et groupements

Pour chacune des requêtes suivantes :

- Pour chaque poste, le nombre de logiciels installés (en utilisant la table **Installer**).
- Pour chaque salle, le nombre de postes (à partir de la table **Poste**).
- Pour chaque logiciel, le nombre d'installations sur des postes différents.
- Moyenne des prix des logiciels 'UNIX'.
- Plus récente date d'achat d'un logiciel.
- Numéros des postes hébergeant 2 logiciels.
- Nombre de postes hébergeant 2 logiciels (utiliser la requête précédente en faisant un **SELECT** dans la clause **FROM**).

Requêtes multitables

Opérateurs ensemblistes

15 Types de postes non recensés dans le parc informatique (utiliser la table **Types**).

16 Types existant à la fois comme types de postes et de logiciels.

17 Types de postes de travail n'étant pas des types de logiciels.

Jointures procédurales

18 Adresses IP complètes des postes qui hébergent le logiciel 'log6'.

19 Adresses IP complètes des postes qui hébergent le logiciel de nom 'Oracle 8'.

20 Noms des segments possédant exactement trois postes de travail de type 'TX'.

21 Noms des salles où l'on peut trouver au moins un poste hébergeant le logiciel 'Oracle 6'. 22 Nom du logiciel acheté le plus récent (utiliser la requête 12).

Jointures relationnelles

Écrire les requêtes 18, 19, 20, 21 avec des jointures de la forme relationnelle. Numéroté ces nouvelles requêtes de 23 à 26.

27 Installations (nom segment, nom salle, adresse IP complète, nom logiciel, date d'installation) triées par segment, salle et adresse IP.

Modifications synchronisées

Écrire le script `modifSynchronisées.sql` pour ajouter les lignes suivantes dans la table `Installer` :

Installer				
<i>nPoste</i>	<i>nLog</i>	numIns	<i>dateIns</i>	<i>delai</i>
...
p2	log6	séquence...	<code>SYSDATE ()</code>	NULL
p8	log1		<code>SYSDATE ()</code>	NULL
p10	log1		<code>SYSDATE ()</code>	NULL

Écrire les requêtes `UPDATE` synchronisées de la forme suivante :

```
UPDATE table1 alias1
  SET colonne = (SELECT COUNT(*)
                  FROM table2 alias2
                  WHERE alias2.colonneA = alias1.colonneB...);
```

Pour mettre à jour automatiquement les colonnes rajoutées :

- **nbSalle** dans la table **Segment** (nombre de salles traversées par le segment) ;
- **nbPoste** dans la table **Segment** (nombre de postes du segment) ;
- **nbInstall** dans la table **Logiciel** (nombre d'installations du logiciel) ;
- **nbLog** dans la table **Poste** (nombre de logiciels installés par poste). Vérifier le contenu des tables modifiées (**Segment**, **Logiciel** et **Poste**).