ADMINISTRATION DE SYSTÈMES ET RÉSEAUX

ADMINISTRATION D'UN SGBD MYSQL EN LIGNE DE COMMANDE

Par : Bernard GIACOMONI

VERSION	OBJET	DATE
0	Création	26/12/2018

Table des matières

I.GÉNÉRALITÉS:

I.1.INTRODUCTION:

L'objectif de ce document est de donner un aperçu global des possibilités d'administration d'un serveurs MySQL à partir de clients MySQL en ligne de commande: vue d'ensemble du dispositif, utilité de l'administration en ligne de commandes, installation d'un serveur MySQL et d'un client MySQL en ligne, principales commandes en ligne, création de procédures (scripts) et automatisation des traitements.

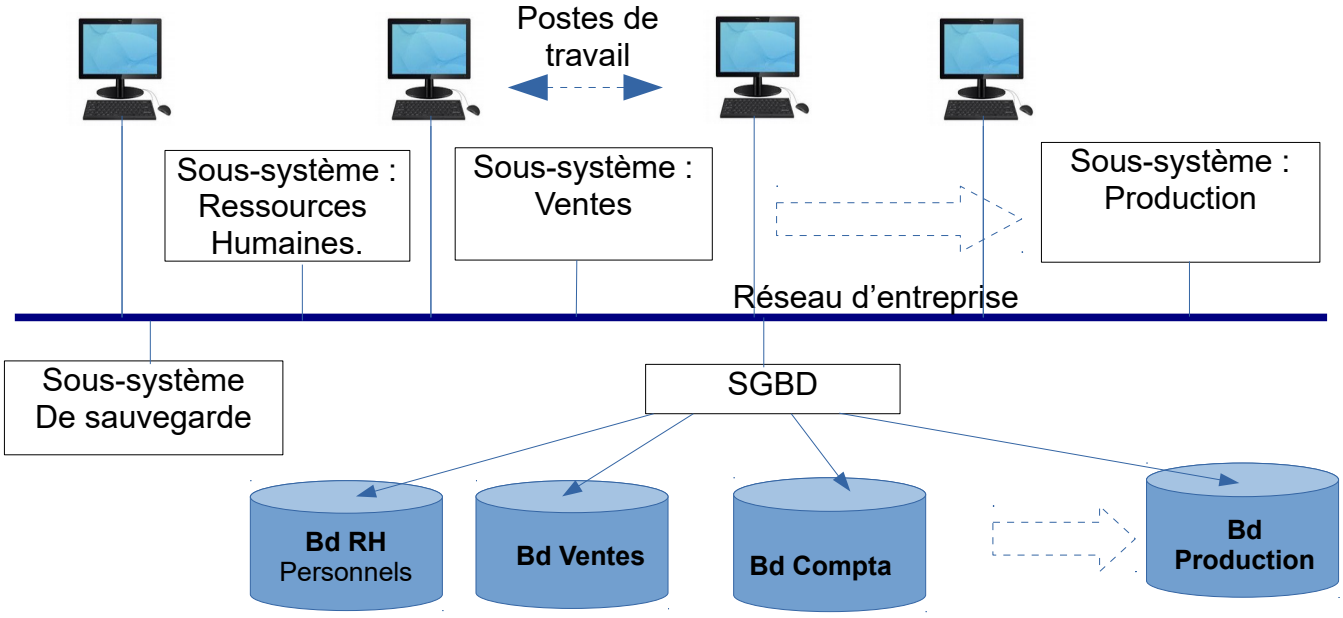
La compréhension de cet ouvrage exige un minimum de connaissances dans les domaines des bases de données, des systèmes de gestion de bases de données (SGBD) et des "langages de scripts".

Des exemples d'administration sous Windows et Linux sont fournis à la fin de l'ouvrage, ainsi que la description de dispositifs de test accessibles avec des moyens informatiques "grand public".

I.2.ACTIVITÉ D'ADMINISTRATION D'UN SGBD:

I.2.1.VUE D'ENSEMBLE

La plupart des SYSTÈMES DE TRAITEMENT INFORMATIQUES sont constitués de différents sous-systèmes (matériel et logiciels) bâtis autour d'un ou plusieurs Systèmes de Gestion de Bases de Données (SGBD), le tout étant relié par un RÉSEAU INFORMATIQUE D'ENTREPRISE. Les différentes bases de données hébergées par les SGBD ont pour rôle de stocker l'ensemble des données RÉMANENTES du système (c'est à dire celles qui subsistent après l'exécution des logiciels qui les produisent ou les utilisent). Ces données peuvent appartenir à différents domaines : administratif, commercial, scientifique , etc.



La fonction d'administration d'un système de traitement informatique englobe généralement l'administration des SGBD que ce système supporte. L'administration d'un SGBD comporte des activités appartenant aux 5 domaines suivants:

1. MAINTIEN EN CONDITION OPÉRATIONNELLE (Maintenance au sens large);
2. CONTRÔLE DES ÉVOLUTIONS du système;
3. ASSISTANCE AUX UTILISATEURS ET FORMATION;
4. SÉCURITÉ (du fonctionnement, des accès et des données);
5. QUALITÉ ET MÉTHODES.

Les paragraphes suivants détaillent les activités concernées par chacun de ces domaines.

I.2.2.MAINTIEN EN CONDITION OPÉRATIONNELLE (M.C.O):

- ***Maintenance des matériels et des capacités de stockage***: contrôle de l'état des systèmes de stockage, planification du remplacement des éléments défectueux ou vieillissants, maintien en état des capacités: défragmentation des unités, gestion des journaux de transactions pour éviter la saturation des disques, etc;
- ***Optimisation des performance en matière de volume de stockage et d'accès aux données*** : mise en place et utilisation d'outils de surveillance des temps de réponse et de diagnostics de dysfonctionnement ;
- ***Participation à la résolution des dysfonctionnements*** : diagnostic de pannes matérielles et systèmes, assistance des personnels de maintenance.

I.2.3.CONTRÔLE DES ÉVOLUTIONS DU SYSTÈME:

- ***Maîtrise des performances et de l'intégrité du système:*** l'administrateur doit être consulté lors que toute modification du système informatique afin d'en évaluer les conséquences sur les performances du système;
- ***Support des équipes de développement*** dans les domaines de la modélisation des bases de données, de l'optimisation des performances, de la sécurité et des tests;
- ***Contrôles des opérations de migration et de mise à jour*** : application des mises à jour préconisées par les fournisseurs, réalisation des migrations imposées en cas de modification de la configuration matérielle;
- ***Contrôle des imports et exports de données*** effectuées depuis ou vers d'autres systèmes.

I.2.4.ASSISTANCE AUX UTILISATEURS :

- ***Assistance et formation des utilisateurs du système*** pour l'utilisation de leurs postes de travail et les bonnes pratiques à respecter (en particulier du point de vue de la sécurité).

I.2.5.SÉCURITÉ:

- **Sécurité des accès:** définition et octroi des privilèges aux différents utilisateurs, imposition de bonnes pratiques de codage pour éviter les attaques par injection de données, application des mises à niveau de sécurité, prises de mesures contre les attaques par déni de services, etc ;
- ***Sauvegarde et récupération de données*** en cas de "crash" matériel ou logiciel ou d'attaques extérieures par la mise en place de solutions matérielles et logicielles

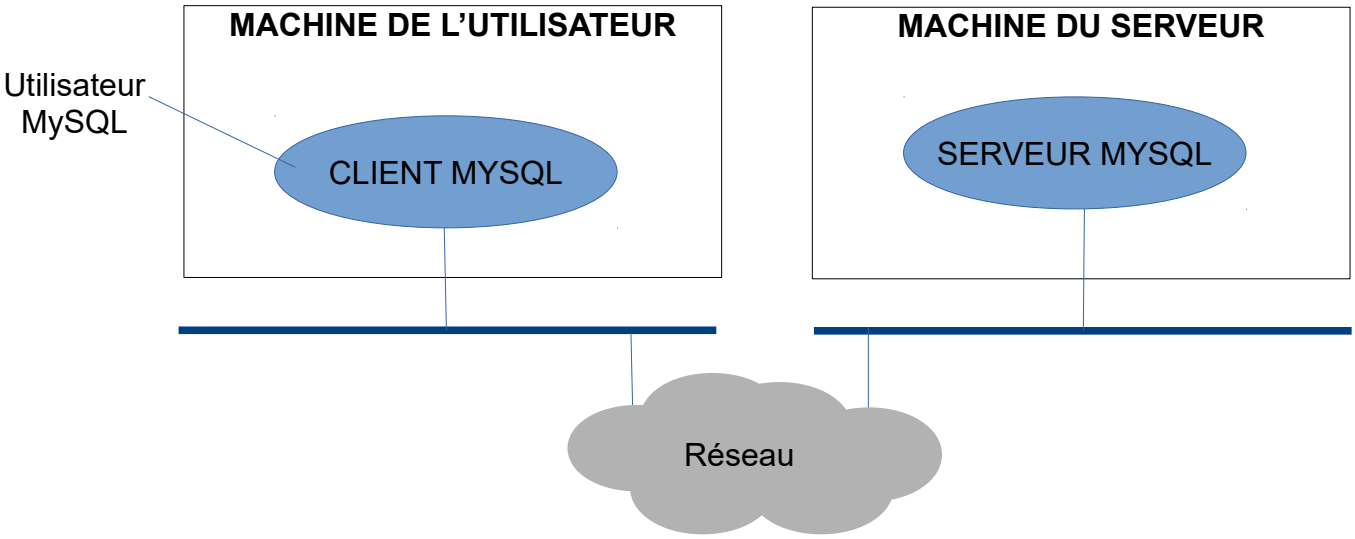
("clusterisation", miroir, etc) afin de pouvoir redémarrer la production le plus rapidement possible en cas de sinistre;

I.2.6.QUALITÉ ET MÉTHODES:

Mise en place et maintien en vigueur des règles à respecter lors des opérations de création, d'évolution, de maintenance ou d'utilisation des bases de données.

I.3.VUE D'ENSEMBLE DU DISPOSITIF:

L'utilisation d'un SERVEUR DE SGBD MyQSQL implique l'utilisation d'un CLIENT MYSQL sur la machine à partir de laquelle on veut utiliser le serveur:



L'image ci-dessus donne un aperçu des aspects matériels et architecturaux du problème:

- Les machines CLIENTS communiquent avec la machine SERVEUR par l'intermédiaire d'un RÉSEAU. Celui-ci peut être LOCAL ou ÉTENDU (web).
- CLIENT et SERVEUR peuvent aussi être hébergés par la même machine. Ils communiquent alors par la "boucle locale" (localhost).
- Le logiciel client peut proposer une interface d'utilisation GRAPHIQUE (comme phpMyAdmin ou MySQL Workbench) ou une interface en LIGNE DE COMMANDE activable depuis une invite de commande unix/linux ou CMD de WINDOWS.

REMARQUE : Utiliser une interface graphique) est certainement plus facile pour créer ou pour modifier ponctuellement une base de données. En revanche, une interface graphique ne permet pas, en général, d'AUTOMATISER des séquences de commandes d'administration représentant des algorithmes complexes:

EXEMPLE: pour mettre à jour dans la table catalogue de la BDD d'un site marchand tous les articles correspondant à un fournisseur donné, il n'est pas question d'utiliser une interface graphique pour modifier un à un les articles: il faut recourir soit à un LOGICIEL (php ou autre), soit à un FICHIER DE COMMANDES (appelé shell ou fichier batch suivant le système d'exploitation).
Les CLIENTS MYSQL en ligne de commande permettent d'utiliser le mécanisme des "scripts de commande" de leur machine hôte pour créer des PROCÉDURES D'ADMINISTRATION AUTOMATISÉES. Celles-ci sont en fait des programmes informatiques particuliers, écrits en langage de commande et pouvant de ce fait inclure des alternatives, des boucles, des entrées-sorties, etc. Ce type d'utilisation est surtout le fait des ADMINISTRATEURS de systèmes.

II.INSTALLATION DE MYSQL:

II.1.REMARQUE:

Nous ne présentons ici qu'une installation BASIQUE (sous DEBIAN ou UBUNTU et sous WINDOWS) pour permettre au lecteur de se constituer rapidement une plate-forme de test s'il veut expérimenter les mécanismes décrits.

Exemple de plate-forme de test avec une seule machine physique WINDOWS:

- Installation d'un client et d'un serveur MySQL dans une machine virtuelle UBUNTU (elle-même supportée par Virtual Box, VM Ware ou autre) ;
- Installation d'un client et d'un serveur MySQL sous une machine réelle Window supportant la machine virtuelle (déclarée "en pont" pour avoir accès directement au réseau local). Le problème de l'installation est traité plus en détail dans d'autres documents.

Exemple de plate-forme de test avec deux machines physiques WINDOWS:

- Installation d'un client et d'un serveur MySQL sur les deux machine réelle WINDOWS connectées au réseau local. Le problème de l'installation est traité plus en détail dans d'autres documents.

II.2.SOUS LINUX:

L'installation la plus facile recours à l'outil apt sous super utilisateur (ubuntu, debian). Avec l'outil apt, on installe les paquets correspondant au client, au serveur et aux dépendances.

```
>apt-get install mysql-server mysql-client libmysqlclient15-dev mysql-common
```

A la fin de cette installation, le serveur et le client MySQL sont disponibles. Le serveur peut être démarré ou arrêté par les commandes >/etc/init.d/mysql start ou >/etc/init.d/mysql stop.

II.3.SOUS WINDOWS:

II.3.1.INSTALLATION D'UN SERVEUR AUTONOME :

Depuis la page web: https://dev.mysql.com/downloads/windows/installer/8.0.html, télécharger le programme d'installation de la version libre MySQL Community Edition. choisir le fichier mysql-installer-web-community (16,3Mo) pour une installation en ligne.

La procédure, très simple, aboutit à l'installation du logiciel MySQ Installer Community qui permet d'installer et de paramétrer simplement un serveur MySQL, deux clients MYSQL en ligne de commande (MySQL Command Line Client et MySQL Command Line Client – unicode) ainsi que l'outil MySQL workbench (client graphique + d'autres fonctions).

Le serveur peut être installé comme un LOGICIEL ou comme un SERVICE WINDOWS. Dans ce dernier cas, qui est peut-être le plus pratique, il peut être démarré ou arrêté à partir du gestionnaire de tâches (onglet "services").

II.3.2.UTILISATION DU SERVEUR MYSQL D'UNE INSTALLATION WAMP :

Lorsqu'une distribution WAMP est déjà installé dans la machine hôte, il est possible d'utiliser le serveur MySQL compris dans cette distribution. Pour cela, il existe deux options :

II.3.2.1.SOIT ESSAYER D'UTILISER LE CLIENT MYSQL COMPRIS DANS CETTE DISTRIBUTION :

Pour cela, il suffit d'ajouter à la variable d'environnement %PATH % du système d'exploitation WINDOWS le chemin d'accès à l'application mysql.exe contenue dans le répertoire wamp local. Ce chemin d'accès est :

"c:\wamp\bin\mysql\mysql<Id. de version>\bin\"

où <Id de version> est l'identificateur de la version de mysql utilisée (par exemple : 5.6.12). Pour trouver ce numéro de version, démarrer wamp, puis cliquer sur l'icône WAMP dans la barre des tâches pour sélectionner "MySQL", puis "version".

Après redémarrage du système, la saisie d'une commande "mysql -u root -p" doit déclencher l'invite de saisie du mot de passe root du serveur MySQL de wamp, ce qui prouve que le client MySQL est bien activable en ligne de commande.

REMARQUE : WINDOW 10 refuse d'exécuter l'application mysql.exe de WAMP (message "cette application ne peut pas s'exécuter sur votre PC"). Il est parfois possible de passer outre en cliquant droit sur le fichier c:\wamp\bin\mysql\mysql<Id. de version>\bin\mysql.exe, puis en sélectionnant Propriétés → Compatibilité → lancer la résolution des problèmes de compatibilité, puis en choisissant la compatibilité windows 7 ou 8, mais ce n'est pas garanti.

II.3.2.2.SOIT INSTALLER UN CLIENT MYSQL AUTONOME :

Lorsqu'il est impossible d'exécuter le mysql.exe de WAMP sous l'OS installé (surtout WINDOWS 10) , une autre solution est d'installer un client autonome. Par exemple, on peut choisir sur la page :

https://downloads.mysql.com/archives/community/

de télécharger le fichier mysql-8.0.12-winx64.zip (Windows (x86, 64-bit), ZIP Archive). Il suffit alors de placer ce fichier dans le répertoire racine "c :" et de le décompresser. On obtient le répertoire :

c:\mysql-8.0.12-winx64

qui contient une application mysql.exe à l'emplacement c:\mysql-8.0.12-winx64\bin\mysql.exe. Il suffit alors d'ajouter à la variable d'environnement %PATH % du système d'exploitation WINDOWS le chemin d'accès à cette application mysql.exe. Ce chemin d'accès est :

c:\mysql-8.0.12-winx64\bin\

Comme précédemment, après redémarrage du système, la saisie d'une commande "mysql -u root -p" doit déclencher l'invite de saisie du mot de passe root du serveur MySQL de wamp, ce qui prouve que le client MySQL est bien activable en ligne de commande.

III.CONNEXION, AUTHENTIFICATION, IMPORT ET EXPORT:

III.1.INTRODUCTION:

La connexion, l'authentification des utilisateurs ainsi que l'import et l'export de bases de données sont les principales commandes d'administration "globales" d'un SGBD MySQL. En effet, elles permettent de:

- Gérer les accès aux bases de données (déclarer ou supprimer des utilisateurs, leur attribuer des privilèges d'accès)
- Connecter ou déconnecter un utilisateur);
- Effectuer la sauvegarde des bases de données et leur réinstallation après un "crash";

Ces commandes peuvent être saisies directement depuis l'invite de commande du système d'exploitation (shell linux ou invite CMD pour WINDOWS).
Pour bien comprendre la gestion des accès, il est nécessaire de connaître les mécanismes et les structures de données qui lui sont associés. Ces mécanismes sont présentés au paragraphe suivant.

REMARQUE : Les commandes en ligne de MySQL permettent également de créer, modifier ou supprimer des bases de données ou d'effectuer des opérations d'extraction de données. Cependant, ces commandes sont plutôt utilisées dans le but de DÉVELOPPER ou de FAIRE ÉVOLUER la structure ou le contenu d'une base de données, ce qui est plutôt du domaine du développement ou de l'exploitation.

III.2.NOTION D'UTILISATEUR ET AUTHENTIFICATION:

III.2.1.PRINCIPE:

Pour pouvoir utiliser un serveur MySQL, il faut d'abord établir une CONNEXION avec lui. L'intervalle entre la connexion d'un utilisateur et sa déconnexion est appelé SESSION D'UTILISATION du serveur.
Pour se connecter, il faut d'abord s'AUTHENTIFIER en tant qu'UTILISATEUR DÉCLARÉ de ce serveur. L'authentification se fait classiquement par la saisie d'un couple (nom d'utilisateur – mot de passe).

III.2.2.NOTION D'UTILISATEUR ET TABLE DES UTILISATEURS:

La liste des utilisateurs déclarés d'un serveur MySQL est conservée par celui-ci dans la table "user" de la base de données MySQL qui est implémentée par défaut dans tout serveur lors de son installation et tenue à jour par la suite au grès de la gestion des privilèges du SGBD.

Cette table possède les champs suivants:

User,
Password,

Select_priv, Insert_priv, Update_priv, Delete_priv, Create_priv
Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv
Grant_priv, References_priv, Index_priv, Alter_priv; Show_db_priv
Super_priv, Create_tmp_table_priv, Lock_tables_priv, Execute_priv
Repl_slave_priv, Repl_client_priv, Create_view_priv, Show_view_priv
Create_routine_priv, Alter_routine_priv, Create_user_priv, Event_priv
Trigger_priv, Create_tablespace_priv,

ssl_type, ssl_cipher, x509_issuer, x509_subject, max_questions, max_updates, max_connections, max_user_connections, plugin, authentication_string, password_expired,

Host.

- Les champs dont les noms se terminent par priv ont pour valeurs possibles Y/N (oui ou non). Ils enregistrent les PRIVILÈGES accordés à chaque utilisateur (exemple: Insert_priv = y signifie que l'utilisateur peut insérer des données dans la base de données qu'il utilise).
- Les champs marqués en italique permettent de paramétrer certains mécanismes du serveur (nombre de connexions simultanées, expiration de mot de passe, etc.).
- Les trois champs en caractères gras (User, Password et Host) sont ceux qui nous intéressent à ce niveau de l'étude: Ils enregistrent le nom d'un utilisateur, son mot de passe (crypté) et le nom ou l'adresse IP de la machine hôte de cet utilisateur. Si l'utilisateur est dans la machine du serveur, le champ host peut contenir "localhost", "127.0.0.1" (adresse locale en IPV4), "::1" (adresse locale en IPV6) ou encore "%".

EXEMPLE:

La table suivante enregistre 4 utilisateurs authentifiés par mots de passe cryptés.

- root est le "super utilisateur". Il peut opérer à partir de la machine du serveur (localhost);
- Admin est un utilisateur dédié à l'administration des bases de données. Il opère à partir du poste Ordi_Admin_BD;
- Compta est un utilisateur dédié à la comptabilité. Il opère à partir du poste Ordi_Compta_1;
- ***DevLog est un utilisateur dédié aux développeurs. Il opère à partir du poste d'adresse IP 192.168.1.77 du réseau local.***

User	Password	Champs privilèges et paramétrages	Host
root	*301B618C1D32D032B84DDEDDC51ECF6BE37CD69C	Tous privilèges	localhost
Admin	*F555574D7113306F427E428B9874CCB78391EFDD	Privilèges nécessaires à la fonction d'administrateur	Ordi_Admin_BD
Compta	*F555574D7113306F427E428B9874CCB78391EFDDY	Privilèges nécessaires à la comptabilité.	Ordi_Compta_1
DevLog	*29643012 5DDED7721966803CB67B494C0A761FC6		192.168.1.77

Le contenu des champs de cette table mysql.user sera modifié lorsqu'un gérera les utilisateurs (création, suppression d'un utilisateur, modification de ses privilèges ou de son mot de passe, etc.). nous verrons plus tard comment réaliser ces travaux.

III.3.PROCÉDURE DE CONNEXION:

III.3.1.COMMANDE DE CONNEXION:

La connexion à un serveur MySQL est toujours soumise à une authentification par un couple (utilisateur-mot de passe). La commande de connexion minimale est:

>mysql [-h <nom/adresse IP du serveur>] [-u <nom d'utilisateur>] -p<mot de passe>;

(Attention : pas d'espace après l'option -p)

Par défaut:

- <nom/adresse IP du serveur> est égal à "localhost" (on est dans la même machine physique);
- <nom d'utilisateur> est égal à "root".

Cette commande permet à un client MySQL de se connecter en local où à distance à un serveur MySQL. Dans le cas d'une connexion à distance, les machines connectées peuvent être physiques ou virtuelles. Elle doit être saisie directement dans l'invite de commande système de la machine (shell linux ou CMD WINDOWS).

Elle implique l'utilisation du compte utilisateur:

<nom d'utilisateur>@<nom/adresse IP du serveur>

qui doit être déclaré dans la table des utilisateurs du serveur (voir plus haut).

REMARQUE : *l'option -p est obligatoire même si l'accès est défini sans mot de passe. Si le mot de passe est absent après -p, celui-ci devra être saisi en ligne avant l'exécution de la commande.*

III.3.2.EXEMPLE:

Supposons que nous activions dans la machine d'adresse ip 192.168.1.101 la commande suivante:

>mysql -h 192.168.1.12 -u Compta -pMDP_Compta;

Cette commande permet de connecter l'utilisateur "Compta", muni du mot de passe "MDP_Compta" au serveur MySQL supporté dans la machine d'adresse IP 192.168.1.12. L'utilisateur Compta@192.168.1.101 doit être déclaré dans la machine serveur.

III.3.3.REMARQUES:

- *Pour se connecter à un serveur distant, il faut que dans celui-ci un utilisateur localisé dans la machine du client soit déclaré (si la machine du client a pour adresse ip 192.168.1.21 l'utilisateur à déclarer dans le serveur sera toto@192.168.1.21);*
- *Si le mot de passe n'est pas spécifié dans la commande, la saisie de celui-ci sera demandée par l'invite "enter password";*
- *Après connexion, c'est l'invite de commande de MySQL (→) qui apparaît à la place de l'invite de commande système.*
- *Toutes les commandes MySQL se termine par ";". Si le point virgule est absent, le invite de commande mysql (→) est renvoyé pour demander la terminaison de la commande;*

III.4.COMMANDE DE CONNEXION "ÉTENDUE":

L'option -e dans une commande de connexion permet d'envoyer au serveur une ou plusieurs commandes d'administration en même temps que la connexion. Son format est le suivant:

>mysql [-h <nom/ @IP>] [-u <utilisateur>] -p<mdp> -e "<liste de commandes MySQL>";

Dans la liste, les commandes doivent être terminées par " ;".

EXEMPLE:

+>mysql -h 192.168.1.12 -u root -pMDP -e "use bibliotheque; show tables;";

Permet de faire les actions suivantes:

- Se connecter au serveur MySQL situé dans la machine d'adresse IP 192.168.1.12 en tant qu'utilisateur "root" authentifié par le mot de passe "MdP";
- Sélectionner la base de données "bibliothèque" (use bibliotheque");
- Lister les tables de cette base de données sur l'écran ("show tables").

Nous verrons plus tard que c'est cette option qui permet d'intégrer des suites de commandes MySQL dans des fichiers batch systèmes sans nécessiter une connexion-déconnexion à chaque ligne.

III.5.DÉCONNEXION:

La commande exit: →exit; Permet de revenir à l'invite de commande du système hébergeur.

III.6.GESTION DES ACCÈS:

III.6.1.MODIFIER LE MOT DE PASSE D'UN UTILISATEUR

Afin de modifier le mot de passe d'un utilisateur du serveur MySQL, il suffit de lancer, **après s'être connecté au serveur** la commande suivante:

mysql>set password for <utilisateur>@<nom d'hôte>=password('<mot de passe>');

EXEMPLE: mysql>set password for root@localhost=password('%vFGh23%'); change le mot de passe root en '%vFGh23%'.

REMARQUE: *pour que le nouveau mot de passe soit pris en compte, il faut qu'il soit conforme aux métriques définies pour le serveur. Ces métriques sont définies par des variables systèmes qui peuvent être affichées par la commande:*

mysql>show variables like 'validate_password%';

Variable	**Signification et valeur**
validate_password.check_user_name	ON/OFF
validate_password.dictionary_file	Nom du fichier dictionnaire
validate_password.length	Longueur du mot de passe
validate_password.mixed_case_count	Nombre minimul de caractères miniscules et majuscules dans le MDP
validate_password.number_count	Nombre de chiffres dans le MDP
validate_password.policy	LOW/MEDIUM/HIGHT - En "low", seule la longueur du MDP est vérifiée; - En "hight", la vérification utilise le fichier dictionnaire.
validate_password.special_char_count	Nombre de caractères spéciaux

La valeur de ces variables peut être modifiée par la commande 'set':

mysql>set global validate_password_policy = HIGHT;

REMARQUE: Par défaut, MySQL crée l'utilisateur "root" sans mot de passe. Ceci peut convenir en phase de développement, mais peut s'avérer très dangereux par la suite. Il convient donc de créer un mot de passe root avant toute mise en service du serveur.

III.6.2.AJOUTER UN UTILISATEUR MYSQL:

Pour créer un utilisateur MySQL, il faut être connecté en tant qu'utilisateur root:

mysql -u root -p

La commande SQL de création d'utilisateur est:

mysql>create user '<nom_utilisateur>'@'<nom d'hote>' identified by '<mot_de_passe>';

Cette commande crée l'utilisateur 'nom_utilisateur' associé à l'hôte 'nom_hote'.

Il faut ensuite accorder des privilèges à l'utilisateur créé:

mysql> grant all privileges on '<*.* / '<nom_bibliotheque>.*' to 'kevin'@'192.168.1.77';

Il faut, enfin, mettre à jour les privilèges:

mysql>flush privileges;

REMARQUES:

- Ces deux commandes peuvent être fusionnées en une seule:

mysql> grant all privileges on <*.*/'nom_BD.*'> to '<nom_utilisateur>'@'<nom_hote>' identified by '<mot de passe>';

- La spécification 'on *.*' permet d'attribuer tous les droits sur toutes les BDD du serveur.

EXEMPLE:

mysql>create user 'kevin'@'<192.168.1.77>' identified by '%kev83%';
mysql> grant all privileges on 'bibiliotheque.*' to 'kevin'@'192.168.1.77';
mysql>flush privileges;

Ces commandes créent l'utilisateur kevin opérant depuis le poste d'IP 192.168.1.77 avec le mot de passe '%kev83%', puis lui attribuent tous les privilèges sur la bd 'bibliotheque'.

III.6.3.ACCORDER DES PRIVILÈGES PARTICULIERS A UN UTILISATEUR:

La commande suivante permet d'accorder des privilèges particuliers à un utilisateur (autres que "all privilèges"):

mysql>grant [<nom_privilege>] on ['<nom_BD>'.'<nom_table>'] to '<nom_utilisateur>'@'<nom_hote>';

Les noms de privilèges possibles sont: CREATE, SELECT, INSERT, UPDATE, DELETE, DROP...etc.

EXEMPLE:

mysql>grant CREATE on 'bibliotheque.*' to 'kevin'@'192.168.1.77';

Accorde le privilège CREATE (créer des tables) à l'utilisateur kevin'@'192.168.1.77.

III.6.4.RÉVOQUER DES PRIVILÈGES PARTICULIERS D'UN UTILISATEUR:

mysql>revoke [<nom_privilege>] on ['<nom_BD>'.'<nom_table>'] from '<nom_utilisateur>'@'<nom_hote>';

EXEMPLE:

mysql>revoke CREATE on 'bibliotheque.*' to 'kevin'@'192.168.1.77';

Supprime le privilège CREATE (créer des tables) à l'utilisateur kevin'@'192.168.1.77.

III.6.5.SUPPRIMER UN UTILISATEUR MYSQL:

Pour supprimer un utilisateur mysql, il faut être connecté en tant qu'utilisateur root, puis passer la commande:

mysql>drop user '<nom_utilisateur>'@'<nom d'hote>';

ATTENTION: il faut absolument donner le nom d'hôte (même si c'est "localhost"). Sinon, la commande renvoie une erreur.

III.7.IMPORT ET EXPORT DE BASES DE DONNÉES:

III.7.1.EXPORTER (SAUVEGARDER) DES BASES DE DONNÉE:

III.7.1.1.EXPORTER TOUTES LES BASES DE DONNÉES D'UN SERVEUR:

Exporter toutes les bases de donnée d'un serveur MySQL n'est possible qu'à partir d'un compte possédant le privilège global CREATE. L'export s'effectue dans un fichier au format *.sql. La commande d'exportation est:

>mysqldump –all-databases –h <nom/@ip serveur> -u root -p > serveur.sql

EXEMPLE:

>mysqldump –all-databases – h 192.168.1.12 -u root -p > serveur_12.sql

Exporte toutes les base de données du serveur MySQL hébergé par la machine d'IP 192.168.1.12 dans le fichier serveur_12.sql du client local.

III.7.1.2.EXPORTER UNE DES BASES DE DONNÉES D'UN SERVEUR:

Pour exporter une seule base du serveur, il faut préciser son nom et retirer l'option –all-databases:

mysqldump –h <nom/@ip serveur> -u root -p bibliothèque > bibliotheque.sql

EXEMPLE:

>mysqldump – h 192.168.1.12 -u root -p bibliotheque > bibliotheque.sql

Exporte (sauvegarde) la base de données "bibliotheque" du serveur MySQL hébergé par la machine d'IP 192.168.1.12 dans le fichier bibliotheque.sql du serveur local.

III.7.1.3.ÉVITER LA SAISIE MANUELLE DU MOT DE PASSE:

Dans les deux cas précédents, la commande est interrompue dans son déroulement par la demande en ligne du mot de passe, ce qui est gênant dans le cas ou la commande est intégrée à un fichier de commandes. Pour éviter cette interruption, il suffit de changer l'option -p en –password=<mot_de_passe>

EXEMPLE:

>mysqldump –all-databases -u root –password=motdepasseroot > serveur.sql

III.7.2.IMPORTER UNE BASE DE DONNÉES:

III.7.2.1.DEPUIS LE FICHIER DE SAUVEGARDE D'UNE BD:

Importer une base de donnée dans un serveur MySQL n'est possible qu'à partir d'un compte possédant le privilège global CREATE.
L'import s'effectue à partir d'un fichier au format *.sql (par exemple, un fichier sql exporté depuis une base de donnée). La commande d'importation est:

>mysql -h <nom/@ip serveur> -u <nom_utilisateur> -p <base_a_importer> <
<base_exportee.sql>

EXEMPLE:

>mysql -h 192.168.1.101 -u root -p bibliotheque <bibliotheque.sql

III.7.2.2.DEPUIS LE FICHIER DE SAUVEGARDE D'UN SERVEUR:

Lorsque l'ensemble des fichiers d'un serveur a été exporté dans un fichier sql, il est possible, grâce à l'option "--one-database", de n'importer qu'une de ces BD à partir de ce fichier:

>mysql --one-database <nom_base_a_importer> < <fichier_sauvegarde_serveur.sql>

EXEMPLE:

>mysql --one-database bibliotheque < svg_serveur_181117

Importe la base de données "bibliotheque" depuis le fichier de sauvegarde svg_serveur_181117.

III.7.2.3.ÉVITER LA SAISIE MANUELLE DU MOT DE PASSE:

Dans les deux cas précédents, la commande est interrompue dans son déroulement par la demande en ligne du mot de passe. Pour éviter cette interruption, il suffit de changer l'option -p en "–password=<mot_de_passe>".

EXEMPLE:

>mysql -h 192.168.1.101 -u root -password=motdepasse bibliotheque <bibliotheque.sql

IV.PRINCIPALES COMMANDES SQL EN LIGNE:

IV.1.INTRODUCTION:

Les commandes étudiées ici sont celles qui permettent de gérer la structure et le contenu des bases de données supportées par un SGBD MySQL. Ce sont en fait des requêtes SQL. Elles doivent être saisie dans l'invite de commande du client MySQL (mysql >), **après s'être connecté au serveur** (mysql -h ** - u *** -p****).

En général, si la requête a abouti, le serveur répond par un message ressemblant à ce qui suit:

→ Query OK, 151 rows affected (0.73 sec)

REMARQUE: Nous avons vu plus haut que l'option -e dans la commande MySQL permet également de présenter ces commandes directement avec la connexion.

EXEMPLE:

>mysql -u root – pmypass -e 'use nom_base; select nom_au from auteurs;' ;

permet de se connecter au MySQL local et d'expédier en même temps vers le serveur les requêtes SQL:

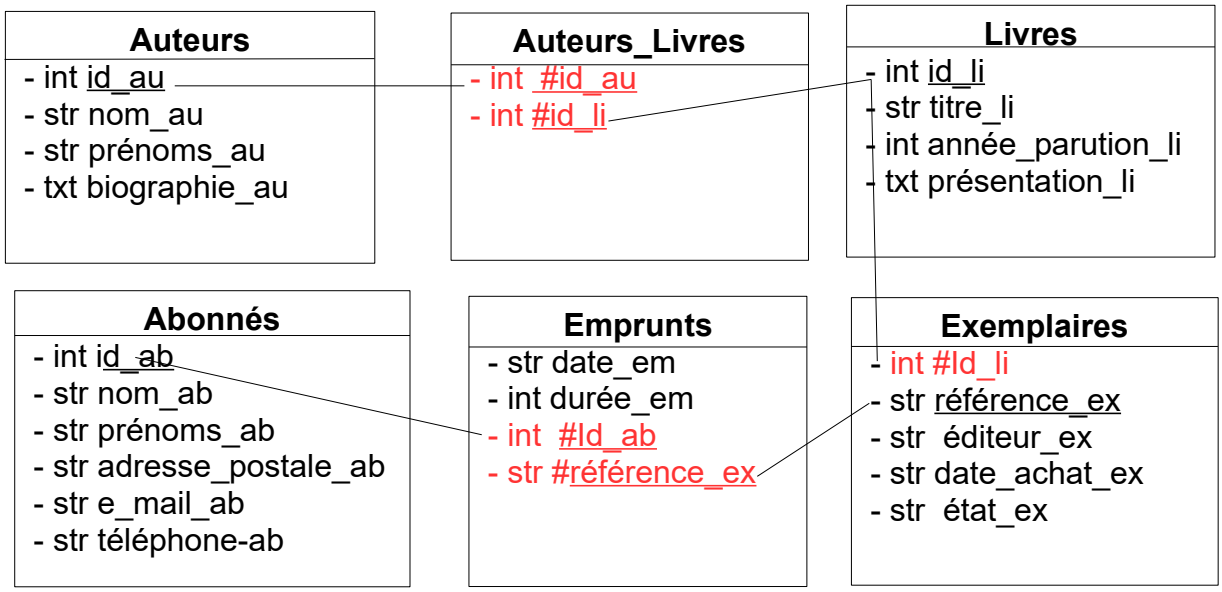
- USE nom_base;
- SELECT nom_au FROM auteurs;

Qui pourraient être saisies directement en manuel sous l'invite de commande mysql> après connexion.

IV.2.GESTION DES BASES DE DONNÉES:

IV.2.1.INTRODUCTION :

Dans les chapitres qui suivent, les exemples est basé sur l'interrogation d'un SGBD MySQL dans lequel figure la base de données suivante (base de données "bibliotheque"):



Cette base de données comprend donc 6 tables :

- La table Auteurs, qui stocke les données relatives à chacun des auteurs;
- La table Livres, qui stocke lers données relatives à chacun des ouvrages disponibles ;
- La table Exemplaires, qui enregistre les exemplaires de livres que possède la bibliothèque ;
- La table Abonnés qui renferme les "comptes abonnées" des clients de la bibliothèque ;
- La table Auteurs_Livres qui permet de faire correspondre à chaque livre son ou ses auteurs ;
- La table Emprunts qui permet de connaître, pour chaque exemplaire, s'il est en cours d'emprunt et l'abonné qui l'a emprunté.

IV.2.2.LISTER L'ENSEMBLE DES BASES DE DONNÉES D'UN SERVEUR

En SQL, la requête SHOW DATABASES permet d'afficher les noms des bases de données du SGBD MySQL. Cette requête peut être saisie:

- Soit dans l'invite de commande mysql, après connexion: mysql>show databases;
- Soit intégrée dans une commande de connexion: >mysql -u root -pMdP -e "show databases;"

Le résultat affiché sera de la forme suivante (information_schema et mysql sont des BD utilisées par le SGBD pour sa gestion et créées à l'installation de celui-ci):

```
+------------------------+
| DATABASE               |
+------------------------+
| information_schema     |
| BD_a                   |
| . . . . .              |
| BD_n                   |
| mysql                  |
+------------------------+
```

REMARQUE: les options d'affichage -B (suppression du formatage en colonne) et -s (suppression de l'entête de colonne) de la commande mysql permettent d'afficher les noms de bases de données ligne par ligne sans habillage :

mysql -u root --password=mot_de_passe -e "show databases;" -B -s;

Aboutit à la sortie ligne par ligne suivant :

```
information_schema
BDD_a
BDD_b
. . .
BD_n
mysql
```

présentation plus intéressante dans le cadre de l'écriture d'une procédure automatique.

IV.2.3.SUPPRIMER UNE BASE DE DONNÉES EXISTANTE:

Pour supprimer une base de données, il faut être connecté au serveur en tant qu'utilisateur possédant le privilège de suppression de cette BD, puis saisir la requête SQL de suppression:

→ drop database if exists <nom_base_a_supprimer>;

EXEMPLE : → drop database if exists bibliotheque; supprime la BDD bibliotheque.

IV.2.4.LISTER LES TABLES D'UNE BASE DE DONNÉES:

Une fois connecté au serveur en tant qu'utilisateur de cette BD, saisir les commandes:

→ use <nom_base_de données>;

→ show tables;

ou encore, directement sous l'invite de commande système:

>mysql -u <nom_utilisateur> -p -e "<use nom_base_de données>; show tables;";

EXEMPLE: >mysql -u compta -p -e "use bibliotheque; show tables;";

aboutit à l'affichage suivant :

```
+------------------------+
| Tables_in_bibliotheque |
+------------------------+
| abonnes                |
| auteurs                |
| auteurs_livres         |
| emprunts               |
| exemplaires            |
| livres                 |
+------------------------+
```

IV.2.5.AFFICHER LA LISTE DES CHAMPS D'UNE TABLE:

Une fois connecté au serveur en tant qu'utilisateur de cette BD, saisir les commandes:
→ use <nom_base_de données>;
→ show tables;
ou encore, directement sous l'invite de commande système:

>mysql -u <nom_utilisateur> -p -e "use <nom_bd>; describe <nom _table>;";

EXEMPLE: > mysql -u root -p -e "use bibliotheque;describe auteurs;"

Le résultat ressemble à ce qui suit:

```
+--------------+-----------------+------+-----+---------+----------------+
| Field        | Type            | Null | Key | Default | Extra          |
+--------------+-----------------+------+-----+---------+----------------+
| id_au        | int(8) unsigned | NO   | PRI | NULL    | auto_increment |
| nom_au       | varchar(100)    | NO   |     | NULL    |                |
| prenoms_au   | varchar(100)    | NO   |     | NULL    |                |
| biographie_au | text           | NO   |     | NULL    |                |
+--------------+-----------------+------+-----+---------+----------------+
```

IV.2.6.AUTRES COMMANDES:

Se référer à la documentation SQL.

V.CRÉATION DE PROCÉDURES AUTOMATIQUES:

V.1.INTRODUCTION :

Pour créer des procédures automatiques d'administration d'un serveur mysql, la solution consiste à lancer des commandes MySQL (mysql ou mysqldump) à partir de fichiers de commandes systèmes (fichiers batch pour windows, script shells sous linux). Le langage de commande en ligne différant notablement entre windows et linux, une présentation est faite pour chacun de ces systèmes.

V.2.CRÉATION DE PROCÉDURES AUTOMATIQUES SOUS WINDOWS :

V.2.1.RAPPEL : LES FICHIER BATCH SOUS WINDOWS :

Le terme anglais "batch" peut être traduit par "lot" en français. L'appellation française "fichier batch" provient de "batch processing files" que l'on peut traduire plus justement par "fichiers de traitement par lots".

Ces fichiers textes munis de l'extension ".bat" (l'extension ".cmd" est aussi reconnue pour les windows récents), contiennent des listes de commandes systèmes (commandes de script windows), chacune d'entre elles occupant une ligne.

Ces fichiers peuvent être exécutés automatiquement :

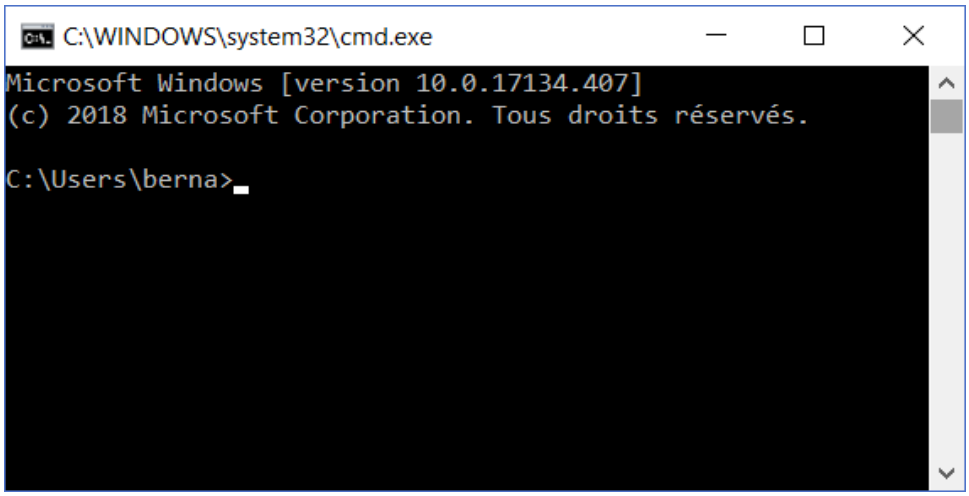
- Soit en effectuant un "double clic" sur leur nom dans une fenêtre d'affichage graphique, exactement comme pour un fichier ".exe";
- Soit par l'intermédiaire de l'invite de commande CMD (exécuter→cmd). C'est ce dernier cas qui nous intéresse dans le cadre de l'administration en ligne de commande.

Lors de leur exécution, les commandes qu'ils renferment sont exécutées lignes par ligne (lot par lot), sauf si la commande en cours d'exécution déclenche une rupture de la séquence d'exécution (commandes goto, if, for, etc.).

Il n'est pas possible, dans le cadre de ce document, de faire une présentation complété du langage de commandes de WINDOWS ni de la création de fichiers BATCH à partir de ces commandes. Nous nous contenterons de présenter quelques exemples qui permettrons au lecteur de faire une révision rapide de ces notions.

V.2.2.RAPPEL -OUVERTURE ET UTILISATION D'UNE "FENÊTRE DE COMMANDE":

Pour interagir en ligne de commande avec le système WINDOWS il est nécessaire d'ouvrir une fenêtre de commande (Pour cela, faire : exécuter → cmd). Ceci revient à lancer l'application C:\\WINDOWS\system32\cmd.exe.



L'invite de commande c:\Users\<nom utilisateur> apparaît (Ici, l'utilisateur est "berna", mais tout dépend de l'utilisateur sous lequel on s'est connecté en début de session windows).

REMARQUE : L'application cmd.exe qui gère cette fenêtre est en fait un INTERPRÉTEUR DE COMMANDES qui accepte un langage de commande très proche de celui de l'ancien système MS-DOS, prédécesseur de WINDOWS dans les machines Microsoft. Cette disposition, sans doute imposée pour conserver la compatibilité ascendante MS-DOS → WINDOWS a été conservée depuis car un langage de commande en ligne est un outil presque indispensable pour l'administration système. De ce fait, par abus de langage, on parle encore souvent de "commandes MS-DOS" et de "fenêtre MS-DOS".

V.2.3.CRÉATION D'UN RÉPERTOIRE DE TEST :

Nous allons créer, dans le répertoire de l'utilisateur, un sous-répertoire TestCMD qui contiendra les fichiers nécessaires à l'étude des différents exemples et dans lequel les commandes seront lancées. Pour cela, saisir dans l'invite de commandes:

C:\Users\berna> md TestCMD

Puis se placer dans ce nouveau répertoire :

C:\Users\berna> cd TestCMD

L'invite de commande dans la fenêtre devrait alors ressembler à :

C:\Users\berna\TestCMD>

V.2.4.PREMIER EXEMPLE : AFFICHAGE DES DONNÉES D'UN FICHIER TEXTE:

V.2.4.1.A.CRÉATION D'UN FICHIER DE TEST :

Nous allons d'abord créer un fichier de texte "semaine.txt" contenant les noms des 7 jours de la semaine, écrits sur sept lignes consécutives (avec une majuscule au début de chaque mot) :

```
Lundi
Mardi
Mercredi
Jeudi
Vendredi
Samedi
Dimanche
```

Nous utiliserons pour cela l'éditeur "bloc note" intégré dans les systèmes WINDOWS, que l'on peut lancer par la commande :

```
C:\Users\berna\TestCMD> notepad semaine.txt
```

Saisissons les jours de la semaine (Lundi, Mardi, etc.) ligne à ligne, puis, sauvegardons son contenu et fermons le bloc note.

V.2.4.1.B.CRÉATION D'UN FICHIER BATCH TRAITANT LE CONTENU DU FICHIER TEST:

Créons alors un fichier batch "AfficheFichierSemaine.bat" vide. Nous pouvons le faire à partir de l'éditeur "bloc note" ou bien utiliser la commande en ligne :

```
C:\Users\berna\TestCMD>echo "" > AfficheFichierSemaine.bat
```

NOTE : *La commande echo "" engendre une chaîne de caractère nulle "" qui est redirigée (par la commande de redirection de flux ">") dans le fichier Commandes.bat, après création de ce fichier.*

V.2.4.1.C.SAISIE DU CONTENU DU FICHIER BATCH :

Nous utiliserons pour cela l'éditeur Bloc Note (notepad) Pour éditer le fichier commande.bat, saisissons la commande :

```
C:\Users\berna\TestCMD>notepad AfficheFichierSemaine.bat
```

et plaçons dans ce fichier le contenu suivant :

```
echo off
REM Affichage du contenu du fichier semaine.txt (Lundi, Mardi, etc...)
REM Paramètre d'entrée: aucun
type semaine.txt
```

La commande echo off a uniquement pour but d'empêcher l'affichage des deux lignes de commentaires (REM). A l'exécution, la commande "type" aura pour effet d'afficher ligne par ligne le contenu du fichier texte semaine.txt que nous avons crée précédemment. Sauvegardons ce contenu et fermons le bloc note.

V.2.4.1.D.EXÉCUTION DU FICHIER BATCH :

Pour cela, saisissons dans l'invite de commande le nom du fichier batch :

C:\Users\berna\TestCMD\AfficheFichierSemaine.bat

Ceci va déclencher l'exécution de Commandes.bat, qui va afficher tel quel le contenu du fichier semaine.txt :

```
Lundi
Mardi
Mercredi
Jeudi
Vendredi
Samedi
Dimanche
```

L'exécution du fichier batch a donc le même résultat que l'exécution de la commande qu'il renferme.

V.2.4.1.E.MODIFICATION DU FICHIER BATCH POUR SÉLECTIONNER LES SORTIES:

Modifions alors le contenu du fichier AfficheFichierSemaine.bat :

```
echo off
REM Affichage du contenu du fichier semaine.txt (Lundi, Mardi, etc...)
REM Paramètre d'entrée: Aucun
type semaine.txt | find "a"
```

INDICATIONS:

- La commande type permet d'afficher ligne par ligne le fichier texte qui lui est spécifié en argument (ici : "semaine.txt"). En sortie de type nous aurons donc les noms des sept jours de la semaine ;
- Le signe "|" (commande "pipeline") permet d'enchaîner deux commandes, la sortie de la première devenant l'entrée de la seconde. De ce fait, le pipe line va présenter en entrée de la commande find les sorties de la commande type (les sept jours de la semaine) ;
- La directive "find" agit comme un filtre pour les chaînes de caractères qu'on lui présente en entrée : elle ne laisse passer que les chaînes qui contiennent la sous-chaîne qu'on lui a spécifiée en argument. Ici , elle ne laissera passer que les chaînes contenant la sous-chaine "a".

De ce fait, le résultat de cette commande composée doit être l'affichage des jours de la semaine comprenant un "a". Lançons alors l'exécution de ce fichier batch :

C:\Users\berna\TestCMD\AfficheFichierSemaine.bat

Comme prévu, on obtient uniquement les noms contenant un "a":

```
Mardi
Samedi
Dimanche
```

V.2.4.1.F.UTILISATION D'UN PARAMÈTRE D'ENTRÉE :

Modifions encore le contenu du fichier AfficheFichierSemaine.bat afin d'utiliser un paramètre d'entrée pour définir la chaîne de sélection des noms de jours:

```
echo off
REM Affichage du contenu du fichier semaine.txt (Lundi, Mardi, etc...)
REM Paramètre d'entrée: <chaîne à rechercher dans les noms de jours>
type semaine.txt | find "%1"
```

La différence avec le script précédent est que la valeur de la chaîne de sélection qui suit la clause FIND est définie par la valeur d'un paramètre d'entrée de la procédure. Ici, c'est le premier paramètre, dont la valeur est accessible par "%1" (la valeur du deuxième paramètre le serait par %2, et ainsi de suite).
Lançons alors l'exécution avec le paramètre "M" :

C:\Users\berna\TestCMD\AfficheFichierSemaine.bat M

On obtient alors l'affichage :

```
Mardi
Mercredi
```

C'est à dire tous les noms de jours contenant un "M".

V.2.4.1.G.CONCLUSION :

- Finalement, la saisie de la commande type semaine.txt | find "<chaîne>" est remplacée par la saisie du nom du fichier batch renfermant cette commande: ceci est déjà avantageux lorsque la commande remplacée est complexe (plus longue que le nom du fichier de commande) ;
- De plus, le fichier batch une fois créé peut être réutilisé un nombre de fois illimité, permettant ainsi de créer de véritables "tâches d'administration" lançables à chaque besoin ;
- La possibilité d'utiliser des paramètres d'entrée donne également une grande capacité d'adaptation aux procédures par rapport aux commandes en ligne.

V.2.5.DEUXIÈME EXEMPLE: AFFICHAGE DES DONNÉES D'UN FICHIER TEXTE GRÂCE A UNE BOUCLE FOR:

V.2.5.1.PRÉSENTATION :

Dans l'exemple précédent, nous avons affiché les lignes de texte contenues dans le fichier semaine.txt (les jours de la semaine) en utilisant la commande "type" (>type semaine.txt). Cette commande ne permet pas d'afficher autre chose que le contenu du fichier, ligne par ligne. Supposons maintenant que nous voulions afficher avec chaque jour de la semaine son numéro d'ordre dans la semaine :

Exemple :

```
1-Lundi
2-Mardi
etc...
```

Dans ce cas, nous ne pouvons pas nous contenter de la commande type. Il faudra utiliser une structure itérative (une boucle) de la forme :

```
FOR <options> <Variable d'occurence> IN <ensembles à parcourir> DO (
        <Commande batch>
        - - - - - - - - - - - - - -
        <Commande batch>
)
```

V.2.5.2.FICHIER BATCH :

```
echo off

REM Affichage des jours contenus dans le fichier semaine.txt (Lundi, Mardi, etc...)
REM accompagnés de leur numéro d'ordre dans la semaine
REM Paramètre d'entrée: aucun
set /a Nb=1
for /f "delims=" %%i in (semaine.txt) do (
        call :incr %%i
)
goto :fin

REM Fonction d'incrémentation du numéro d'ordre
REM Paramètres d'entrée: <nom du jour>
:incr
echo %Nb%-%1
set /a Nb=%Nb%+1
goto :eof

:fin
```

V.2.6.TROISIÈME EXEMPLE-AFFICHAGE D'INFORMATIONS SUR LES CARTES RÉSEAU D'UNE MACHINE WINDOWS:

V.2.6.1.PRÉSENTATION :

Ce deuxième exemple exploite les informations données par la commande batch ipconfig /all sur la configuration des entrées-sorties réseau de la machine hôte.
Cette commande permet d'afficher de nombreuses données sur les liaisons réseau de la machine locale (adresses IP V4 et V6, adresses MAC, masques de sous-réseau, etc.). Par exemple :

```
C:\Users\berna\TestsCMD>ipconfig /all

Configuration IP de Windows

   Nom de l'hôte . . . . . . . . . . : PC-BUREAU
   Suffixe DNS principal . . . . . . :
   Type de noeud. . . . . . . . . .  : Hybride
   Routage IP activé . . . . . . . . : Non
   Proxy WINS activé . . . . . . . . : Non
   Liste de recherche du suffixe DNS.: home

Carte Ethernet Ethernet :

   Suffixe DNS propre à la connexion. . . : home
   Description. . . . . . . . . . . . . . : Realtek PCIe GBE Family Controller
   Adresse physique . . . . . . . . . . . : 98-EE-CB-63-85-3C
   DHCP activé. . . . . . . . . . . . . . : Oui
   Configuration automatique activée. . . : Oui
   Adresse IPv6. . . . . . . . . . . . . .: 2a01:cb1c:8014:8e00:54fb:12fb:3e82:eb62(préféré)
   Adresse IPv6 temporaire . . . . . . . .: 2a01:cb1c:8014:8e00:51f3:9ed8:3683:1aa(préféré)
   Adresse IPv6 de liaison locale. . . . .: fe80::54fb:12fb:3e82:eb62%17(préféré)
   Adresse IPv4. . . . . . . . . . . . . .: 192.168.1.12(préféré)
   Masque de sous-réseau. . . . . . . . . : 255.255.255.0
   Bail obtenu. . . . . . . . . . . . . . : lundi 31 décembre 2018 07:43:53
   Bail expirant. . . . . . . . . . . . . : samedi 5 janvier 2019 14:06:39
   Passerelle par défaut. . . . . . . . . : fe80::a3e:5dff:feda:39e%17
                                            192.168.1.1
   Serveur DHCP . . . . . . . . . . . . . : 192.168.1.1
   IAID DHCPv6 . . . . . . . . . . . . . : 43577035
   DUID de client DHCPv6. . . . . . . . . : 00-01-00-01-20-DE-32-D9-98-EE-CB-63-85-3C
   Serveurs DNS. . .  . . . . . . . . . . : fe80::a3e:5dff:feda:39e%17
                                            192.168.1.1
   NetBIOS sur Tcpip. . . . . . . . . . . : Activé
   Liste de recherche de suffixes DNS propres à la connexion :
```

```
                                home

Carte Ethernet VirtualBox Host-Only Network #3 :

   Suffixe DNS propre à la connexion. . . :
   Description. . . . . . . . . . . . . . : VirtualBox Host-Only Ethernet Adapter #3
   Adresse physique . . . . . . . . . . . : 0A-00-27-00-00-10
   DHCP activé. . . . . . . . . . . . . . : Non
   Configuration automatique activée. . . : Oui
   Adresse IPv6 de liaison locale. . . . .: fe80::1835:e674:e50f:d5c2%16(préféré)
   Adresse IPv4. . . . . . . . . . . . . .: 192.168.56.1(préféré)
   Masque de sous-réseau. . . . . . . . . : 255.255.255.0
   Passerelle par défaut. . . . . . . . . :
   IAID DHCPv6 . . . . . . . . . . . : 302645287
   DUID de client DHCPv6. . . . . . . . : 00-01-00-01-20-DE-32-D9-98-EE-CB-63-85-3C
   Serveurs DNS. . .  . . . . . . . . . . : fec0:0:0:ffff::1%1
                                fec0:0:0:ffff::2%1
                                fec0:0:0:ffff::3%1
   NetBIOS sur Tcpip. . . . . . . . . . . : Activé
```

Le fichier batch que nous allons réaliser permet d'affiches, au choix, pour toutes les cartes réseau installées:

- La description de ces cartes réseau;
- L'adresse IPV4 de la machine ;
- Le masque de sous-réseau
- La passerelle par défaut.

Cet exemple fait usage de l'instruction alternative if et d'une boucle d'itération for.

V.2.6.2.FICHIER BATCH :

```
echo off
REM Affichage d'informations sur les connexions réseau installées sur la machine.
REM Paramètre d'entrée: aucun

REM Affichage des différents types d'informations sélectionnables
echo "Pour afficher la description des cartes réseau installées, saisir 1"
echo "Pour afficher l'adresse IPV4 de la machine, saisir 2"
echo "Pour afficher le masque de sous-reseau, saisir 3"
echo "Pour afficher la passerelle par defaut, saisir 4"

REM Saisie de l'information à afficher
set /p input=Saisir l'option choisie:

REM Définition de la sous-chaîne permettant de sélectionner dans le flux de sortie de
REM la commande ipconfig /all les informations désirées
if %input% == 1 (
        set string=Description
) else if  %input% == 2 (
        set string=IPv4
) else if %input% == 3 (
        set string=Masque
) else if %input% == 4 (
        set string=Passerelle
) else (
        set string=""
)

REM Boucle parcourant ligne par ligne les sorties de la commande ipconfig.old
for /f "delims=" %%i in ('ipconfig /all') do (
        echo %%i | find "Carte"
        echo %%i | find "%string%"
)
```

Commentaires :

- Après avoir affiché le "mode d'emploi" du fichier batch (définition des différentes options d'affichage), on initialise la variable "input" avec la valeur saisie par l'utilisateur (1, 2, 3 ou 4) ;
- Puis, en fonction de la valeur saisie, on initialise la variable "string" avec une chaîne de caractères caractérisant les données à afficher (Description, IPv4, Masque, Passerelle ou " "). Ce traitement est réalisé par la structure alternative if … else if …else if … else ...

- Enfin, on utilise une structure de boucle for pour afficher les informations désirées pour chaque carte réseau.

V.2.7.QUATRIÈME EXEMPLE-AFFICHAGE DES TABLES D'UNE BASE DE DONNÉES :

V.2.7.1.PRÉSENTATION :

L'objectif de ce troisième exemple est de créer une procédure batch permettant d'afficher la description d'une table contenue dans une base de données MySQL hébergée par un serveur MySQL local. Les paramètres de connexion au SGBD, le nom de la base de données et le nom de la table seront saisis en ligne.

V.2.7.2.FICHIER BATCH :

```
echo off
REM ************************************************************************
REM Affichage des tables d'une BDD dont le nom contient une chaîne donnée
REM Paramètres d'entrée:  aucun
REM ************************************************************************

REM Saisie du nom de la base de données et des éléments de connexion
REM ------------------------------------------------------------------

SETLOCAL
set /p BDD=Saisir le nom de la base de donnees:
set /p USER=Saisir votre nom d'utilisateur:
set /p MDP=Saisir votre mot de passe:

REM Boucle de travail
REM -----------------
:BOUCLE
	REM Récupération et affichage des noms des tables de la BDD
	REM ----------------------------------------
	mysql -u %USER% -p%MDP% -e "use bibliotheque; show tables;" -B -s

	REM Saisie du nom de la table à afficher et d'une chaîne de sélection des entrées
	REM Si cette chaîne n'est pas saisie, toutes les entrées sont affichées
	REM ------------------------------------------------------------------------------
	set /p TABLE=Saisir le nom de la table a afficher:
	set /p CHR=Saisir la chaine de recherche ("" pour aucune):

	REM Récupération des entrées à afficher
	REM -----------------------------------
	if %CHR% EQU "" (
		mysql -u %USER% -p%MDP% -e "use %BDD%; select * from %TABLE%;" > XXXX.txt
	) else (
		mysql -u %USER% -p%MDP% -e "use %BDD%; select * from %TABLE%;" | find
		"%CHR%" > XXXX.txt
	)

	REM Affichage des entrées sélectionnées
	REM -----------------------------------
	echo ************************************************************************
	for /f "delims=" %%i in ( 'type XXXX.txt' ) do (
		echo %%i
		echo  ------------------------------------------------------------
```

```
	)
	echo ************************************************************************
	del XXXX.txt

	REM Test d'arrèt du logiciel
	REM ------------------------
	set /p FIN=Afficher une autre table ? (o/n):
	if %FIN% EQU o (
		goto :BOUCLE
	)
REM Fin de boucle

set "USER="
set "MDP="
```

VI.SOLUTIONS POUR SE CONNECTER AU SGBD DANS UNE PROCÉDURE D'ADMINISTRATION AUTOMATIQUE :

VI.1.POSITION DU PROBLÈME:

Lorsqu'on veut créer une procédure d'administration AUTOMATISÉE d'un SGBD MySQL (script shell Linux ou batch Windows), un des problème qui se pose est la fourniture des identificateurs de connexion au moment de l'exécution des commandes mysql: comme il n'est pas possible dans ce cas d'opter pour la saisie en ligne des identifiants (qui stopperait l'exécution de la procédure), les solutions suivantes sont disponibles:

VI.2.SOLUTION N°1: LIBELLER LES IDENTIFICATEURS DANS LES COMMANDES:

Il est possible de libeller le login et le mot de passe dans la commande grâce aux options -u et -p (par exemple, pour se connecter : >mysql -u Compta -p%mdpcompta%). Cependant, cette solution est évidemment très dangereuse du point de vue de la sécurité puisque ces identifiants seront accessibles en clair dans toutes ces procédures.

VI.3.SOLUTION N°2: DÉCLARER DES "GROUPES" DANS LE FICHIER MY.CNF:

Il existe aussi la possibilité de créer des "groupes" dans le fichier my.cnf sous Linux (/etc/mysql/my.cnf) ou dans le fichier my.ini sous Windows (à la racine du répertoire d'installation du serveur). Un groupe se déclare de la manière suivante :

```
[<nom du groupe>]
user = <nom d'utilisateur>
password = "<mot de passe>"
```

Ce groupe peut alors être utilisé dans un script de connexion de la manière suivante:

```
>---------------------------------------------
>mysql –defaults-group-suffix=<nom du groupe>
>---------------------------------------------
```

Cependant, ceci revient toujours à introduire dans un fichier qui doit rester accessible à tous les utilisateurs des identifiants de connexion sous une forme non chiffrée.

Ces deux solutions présentent donc des défauts de sécurité trop importants pour être utilisées de manière professionnelle.

VI.4.SOLUTION N°3: UTILISER UN FICHIER CRYPTÉ POUR SAUVEGARDER LES IDENTIFIANTS DE CONNEXION:

En revanche, l'option –login-path de la commande mysql permet d'utiliser le contenu d'un fichier particulier pour réaliser la connexion avec le SGBD.

Ce nouveau fichier de configuration permet, comme .my.cnf ou my.ini, de créer des "groupes" renfermant la définition des paramètres host, user et password. Comme il n'est utilisé que pour la connexion d'un utilisateur particulier, il peut être chiffré, ce qui évite de faire figurer un mot de passe dans un script ou dans un fichier .my.cnf ou my.ini qu'il n'est pas possible de chiffrer.
Ce nouveau fichier (nommé .mylogin.cnf) peut être créé dans linux ou windows par l'utilitaire **mysql_config_editor**. On le trouve:

- Sous LINUX, dans le répertoire racine de l'utilisateur qui l'a créé ;
- Sous WINDOWS, dans le répertoire %APPDATA%\MySQL.

Ce fichier doit lui-même être protégé contre tous accès malveillants, car son cryptage n'est pas une garantie contre un hacker expérimenté.

VI.5.SYNTAXE DES COMMANDES:

VI.5.1.CRÉATION D'UN GROUPE :

La création d'un groupe obéît à la syntaxe suivante:

```
>mysql_config_editor set -G <nom du groupe>  -u <nom de l'utilisateur>
        -h <Adresse IP/nom d'hôte> -P <numero de port (ex: 3306)> -p<mot de passe>
```

Si le mot de passe n'est pas fourni, le système le demandera en ligne.
Cette commande provoque soit la création d'un fichier .mylogin.cnf renfermant le groupe:

```
[<nom du groupe>]
host :          <Adresse IP/nom d'hôte>
user :          <nom de l'utilisateur>
password :      <mot de passe>
port :          <numéro de port>
```

Soit l'ajout de ce nouveau groupe aux groupes existants.

VI.5.2.MODIFICATION D'UN GROUPE:

On utilise la même commande que pour la création.

VI.5.3.SUPPRESSION D'UN GROUPE:

Pour supprimer un groupe existant, utiliser la syntaxe suivante:

```
>mysql_config_editor remove -G <nom du groupe à supprimer>
```

VI.5.4.UTILISATION DANS UNE COMMANDE MySQL EN LIGNE:

Pour se connecter en utilisant un groupe existant, utiliser la syntaxe suivante:

```
>mysql --login-path=<nom du groupe à utiliser>
```

VI.5.5.VISUALISATION DES GROUPES EXISTANTS:

```
>mysql_config_editor print --all
```

VI.6.EXEMPLE D'UTILISATION

VI.6.1.CRÉATION DU FICHIER ".mylogin.cnf":

Supposons que les identificateurs de l'utilisateur en cours soient "admin" et "%admin %". Nous allons créer le groupe de paramètres "grp1", défini comme suit:

host : 127.0.0.1 (utilisateur local)
user : admin
password : %admin%
port : 3306

Pour cela, nous utiliserons la commande suivante:

>mysql_config_editor set -G grp1 -u admin -h 127.0.0.1 -P 3306 -p

Le mot de passe de l'utilisateur n'étant pas renseigné (option -p), le système va demander ce mot de passe:

>enter password:

La saisie d'un mot de passe entraîne la création (s'il n'existe pas) d'un fichier .mylogin.cnf à la racine du répertoire de l'utilisateur (sous Linux) ou dans le répertoire %appdata%\mysql (sous Windows) et l'ajout dans ce fichier de la déclaration du groupe grp1 :

```
[grp1]
host :      127.0.0.1
user :      admin
password :  %admin%
port :      3306
```

VI.6.2.CONNEXION EN UTILISANT CE FICHIER:

Pour se connecter à MySQL, il suffit maintenant de faire référence au groupe grp1 grâce à l'option "–login-path=" de la commande mysql:

>mysql --login-path=grp1

MyQL se connecte et répond alors avec ses informations d'accueil

>Welcome to the MySQL monitor. Commands end with ; or \g.
>Your MySQL connection id is 34096
> etc …..
mysql>

VI.6.3.AFFICHAGE DES GROUPES EXISTANTS :

Il est alors possible de vérifier que le groupe grp1 a bien été créé en saisissant:

>mysql_config_editor print –all ;

Ceci déclenche l'affichage du contenu du fichier .mylogin.cnf :

```
[grp1]
user = admin
password = *******
host = 127.0.0.1
port = 3306
```

VI.6.4.SUPPRESSION DU GROUPE:

C'est la commande :

```
>mysql_config_editor remove -G grp1
```

Qui supprime le groupe "grp1" dans le fichier .mylogin.cnf.

VII.SOLUTIONS POUR SE CONNECTER AU SGBD DANS UNE PROCÉDURE D'ADMINISTRATION AUTOMATIQUE :

VII.1.POSITION DU PROBLÈME:

Lorsqu'on veut créer une procédure d'administration AUTOMATISÉE d'un SGBD MySQL (script shell Linux ou batch Windows), un des problème qui se pose est la fourniture des identificateurs de connexion au moment de l'exécution des commandes mysql: comme il n'est pas possible dans ce cas d'opter pour la saisie en ligne des identifiants (qui stopperait l'exécution de la procédure), les solutions suivantes sont disponibles:

VII.2.SOLUTION N°1: LIBELLER LES IDENTIFICATEURS DANS LES COMMANDES:

Il est possible de libeller le login et le mot de passe dans la commande grâce aux options -u et -p (par exemple, pour se connecter : >mysql -u Compta -p%mdpcompta%). Cependant, cette solution est évidemment très dangereuse du point de vue de la sécurité puisque ces identifiants seront accessibles en clair dans toutes ces procédures.

VII.3.SOLUTION N°2: DÉCLARER DES "GROUPES" DANS LE FICHIER MY.CNF:

Il existe aussi la possibilité de créer des "groupes" dans le fichier my.cnf sous Linux (/etc/mysql/my.cnf) ou dans le fichier my.ini sous Windows (à la racine du répertoire d'installation du serveur). Un groupe se déclare de la manière suivante :

```
[<nom du groupe>]
user = <nom d'utilisateur>
password = "<mot de passe>"
```

Ce groupe peut alors être utilisé dans un script de connexion de la manière suivante:

```
>---------------------------------------------
>mysql –defaults-group-suffix=<nom du groupe>
>---------------------------------------------
```

Cependant, ceci revient toujours à introduire dans un fichier qui doit rester accessible à tous les utilisateurs des identifiants de connexion sous une forme non chiffrée.

Ces deux solutions présentent donc des défauts de sécurité trop importants pour être utilisées de manière professionnelle.

VII.4.SOLUTION N°3: UTILISER UN FICHIER CRYPTÉ POUR SAUVEGARDER LES IDENTIFIANTS DE CONNEXION:

En revanche, l'option –login-path de la commande mysql permet d'utiliser le contenu d'un fichier particulier pour réaliser la connexion avec le SGBD.

Ce nouveau fichier de configuration permet, comme .my.cnf ou my.ini, de créer des "groupes" renfermant la définition des paramètres host, user et password. Comme il n'est utilisé que pour la connexion d'un utilisateur particulier, il peut être chiffré, ce qui évite de faire figurer un mot de passe dans un script ou dans un fichier .my.cnf ou my.ini qu'il n'est pas possible de chiffrer.
Ce nouveau fichier (nommé .mylogin.cnf) peut être créé dans linux ou windows par l'utilitaire **mysql_config_editor**. On le trouve:

- Sous LINUX, dans le répertoire racine de l'utilisateur qui l'a créé ;
- Sous WINDOWS, dans le répertoire %APPDATA%\MySQL.

Ce fichier doit lui-même être protégé contre tous accès malveillants, car son cryptage n'est pas une garantie contre un hacker expérimenté.

VII.5.SYNTAXE DES COMMANDES:

VII.5.1.CRÉATION D'UN GROUPE :

La création d'un groupe obéît à la syntaxe suivante:

>mysql_config_editor set -G <nom du groupe> -u <nom de l'utilisateur>
-h <Adresse IP/nom d'hôte> -P <numero de port (ex: 3306)> -p<mot de passe>

Si le mot de passe n'est pas fourni, le système le demandera en ligne.

Cette commande provoque soit la création d'un fichier .mylogin.cnf renfermant le groupe:

```
[<nom du groupe>]
host :          <Adresse IP/nom d'hôte>
user :          <nom de l'utilisateur>
password :      <mot de passe>
port :          <numéro de port>
```

Soit l'ajout de ce nouveau groupe aux groupes existants.

VII.5.2.MODIFICATION D'UN GROUPE:

On utilise la même commande que pour la création.

VII.5.3.SUPPRESSION D'UN GROUPE:

Pour supprimer un groupe existant, utiliser la syntaxe suivante:

>mysql_config_editor remove -G <nom du groupe à supprimer>

VII.5.4.UTILISATION DANS UNE COMMANDE MySQL EN LIGNE:

Pour se connecter en utilisant un groupe existant, utiliser la syntaxe suivante:

>mysql --login-path=<nom du groupe à utiliser>

VII.5.5.VISUALISATION DES GROUPES EXISTANTS:

>mysql_config_editor print --all

VII.6.EXEMPLE D'UTILISATION

VII.6.1.CRÉATION DU FICHIER ".mylogin.cnf":

Supposons que les identificateurs de l'utilisateur en cours soient "admin" et "%admin %". Nous allons créer le groupe de paramètres "grp1", défini comme suit:

host : 127.0.0.1 (utilisateur local)
user : admin
password : %admin%
port : 3306

Pour cela, nous utiliserons la commande suivante:

>mysql_config_editor set -G grp1 -u admin -h 127.0.0.1 -P 3306 -p

Le mot de passe de l'utilisateur n'étant pas renseigné (option -p), le système va demander ce mot de passe:

>enter password:

La saisie d'un mot de passe entraîne la création (s'il n'existe pas) d'un fichier .mylogin.cnf à la racine du répertoire de l'utilisateur (sous Linux) ou dans le répertoire %appdata%\mysql (sous Windows) et l'ajout dans ce fichier de la déclaration du groupe grp1 :

```
[grp1]
host :      127.0.0.1
user :      admin
password :  %admin%
port :      3306
```

VII.6.2.CONNEXION EN UTILISANT CE FICHIER:

Pour se connecter à MySQL, il suffit maintenant de faire référence au groupe grp1 grâce à l'option "–login-path=" de la commande mysql:

>mysql --login-path=grp1

MyQL se connecte et répond alors avec ses informations d'accueil

>Welcome to the MySQL monitor. Commands end with ; or \g.
>Your MySQL connection id is 34096
> etc …..
mysql>

VII.6.3.AFFICHAGE DES GROUPES EXISTANTS :

Il est alors possible de vérifier que le groupe grp1 a bien été créé en saisissant:

>mysql_config_editor print –all ;

Ceci déclenche l'affichage du contenu du fichier .mylogin.cnf :

```
[grp1]
user = admin
password = *******
host = 127.0.0.1
port = 3306
```

VII.6.4.SUPPRESSION DU GROUPE:

C'est la commande :

>mysql_config_editor remove -G grp1

Qui supprime le groupe "grp1" dans le fichier .mylogin.cnf.

VII.7.SÉCURITÉ:

Bien que le fichier .mylogin.cnf soit chiffré, la commande :
>my_print_defaults -s <nom du groupe>
permet d'afficher le groupe avec le mot de passe EN CLAIR .

Exemple :

>my_print_defaults -s gpr1
Affiche le groupe avec le mot de passe EN CLAIR :

```
--user=admin
--password=%admin %
--host=127.0.0.1
--port=3306
```

La protection des identifiants de connexion ne repose donc que sur le contrôle de l'accès de l'utilisateur du système (Linux ou Windows).