Version : **2024.01**

Dernière mise-à-jour : 2024/12/08 15:59

Topic 105: Shells and Shell Scripting

Contenu du Module

        - Opérateurs Arithmétiques
        - Opérateurs de Comparaison
        - Opérateurs Logiques
      - La commande let
        - Opérateurs Arithmétiques
        - Opérateurs de comparaison
        - Opérateurs Logiques
        - Opérateurs travaillant sur les bits
      - Structures de contrôle
        - If
        - case
          - Exemple
      - Boucles
        - for
        - while
        - Exemple
    - Scripts de Démarrage
      - LAB #1- Scripts de Démarrage
        - ~/.bash_profile
        - ~/.bashrc
  - Le Langage SQL
    - Installation de MariaDB
    - Démarrage du Serveur
      - Le Service MariaDB
      - Invocation Directe
    - Arrêt du Serveur
      - Le Service MariaDB
      - La Commande mysqladmin
    - Configuration
      - Le Client MySQL
        - Utilisation
        - Options
    - LAB #2 - Configuration de Base
    - LAB #3 - Le Langage SQL

La commande seq

La commande **seq** affiche une séquence de nombres du PREMIER au DERNIER par pas d'un INCREMENT. La commande prend la forme suivante :

- **seq** [options] PREMIER
- **seq** [options] PREMIER DERNIER
- **seq** [options] PREMIER INCREMENT DERNIER

Par exemple :

```
[root@centos7 ~]# seq 10
1
2
3
4
5
6
7
8
9
10
[root@centos7 ~]# seq 20 30
20
21
22
23
24
25
26
27
28
29
30
```

```
[root@centos7 ~]# seq 20 10 90
20
30
40
50
60
70
80
90
[root@centos7 ~]#
```

Options de la commande

A faire : Utilisez l'option **–help** de la commande **seq** pour visualiser les options de la commande.

Affichage des variables du shell

Une variable du shell peut être affichée grâce à la commande :

```
$ echo $VARIABLE [Entrée]
```

Les variables principales

Variable	Description
BASH	Le chemin complet du shell.
BASH_VERSION	La version du shell.
EUID	EUID de l'utilisateur courant.

Variable	Description
UID	UID de l'utilisateur courant.
PPID	Le PID du processus père.
PWD	Le répertoire courant.
OLDPWD	Le répertoire avant la dernière commande cd. Même chose que la commande **cd -**.
RANDOM	Un nombre aléatoire entre 0 et 32767
SECONDS	Le nombre de scondes écoules depuis le lancement du shell
LINES	Le nombre de lignes de l'écran.
COLUMNS	La largeur de l'écran.
HISTFILE	Le fichier historique
HISTFILESIZE	La taille du fichier historique
HISTSIZE	Le nombre de commandes mémorisées dans le fichier historique
HISTCMD	Le numéro de la commande courante dans l'historique
HISTCONTROL	**ignorespace** ou **ignoredups** ou **ignoreboth**
HOME	Le répertoire de connexion.
HOSTTYPE	Le type de machine.
OSTYPE	Le système d'exploitation.
MAIL	Le fichier contenant le courrier.
MAILCHECK	La fréquence de vérification du courrier en secondes.
PATH	Le chemin de recherche des commandes.
PROMPT_COMMAND	La commande exécutée avant chaque affichage du prompt.
PS1	Le prompt par défaut.
PS2	Le deuxième prompt par défaut
PS3	Le troisième prompt par défaut
PS4	Le quatrième prompt par défaut
SHELL	Le shell de préférence.
SHLVL	Le nombre d'instances du shell.
TMOUT	Le nombre de secondes moins 60 d'inactivité avant que le shell exécute la commande **exit**.

Les Variables de Régionalisation et d'Internationalisation

L'**Internationalisation**, aussi appelé **i18n** car il y a 18 lettres entre la lettre **I** et la lettre **n** dans le mot *Internationalization*, consiste à adapter un logiciel aux paramètres variant d'une région à l'autre :

- longueur des mots,
- accents,
- écriture de gauche à droite ou de droite à gauche,
- unité monétaire,
- styles typographiques et modèles rédactionnels,
- unités de mesures,
- affichage des dates et des heures,
- formats d'impression,
- format du clavier,
- etc ...

Le **Régionalisation**, aussi appelé **l10n** car il y a 10 lettres entre la lettre **L** et la lettre n du mot *Localisation*, consiste à modifier l'internalisation en fonction d'une région spécifique.

Le code pays complet prend la forme suivante : **langue-PAYS.jeu_de_caractères**. Par exemple, pour la langue anglaise les valeurs de langue-PAYS sont :

- en_GB = Great Britain,
- en_US = USA,
- en_AU = Australia,
- en_NZ = New Zealand,
- en_ZA = South Africa,
- en_CA = Canada.

Les variables système les plus importants contenant les informations concernant le régionalisation sont :

Variable	Description
LC_ALL	Avec une valeur non nulle, celle-ci prend le dessus sur la valeur de toutes les autres variables d'internationalisation
LANG	Fournit une valeur par défaut pour les variables d'environnement dont la valeur est nulle ou non définie.

Variable	Description
LC_CTYPE	Détermine les paramètres régionaux pour l'interprétation de séquence d'octets de données texte en caractères.

Par exemple :

```
[trainee@centos7 ~]$ echo $LC_ALL
en_GB.UTF-8
[trainee@centos7 ~]$ echo $LC_CTYPE

[trainee@centos7 ~]$ echo $LANG
en_GB.UTF-8

[trainee@centos7 ~]$ locale
LANG=en_GB.UTF-8
LC_CTYPE="en_GB.UTF-8"
LC_NUMERIC="en_GB.UTF-8"
LC_TIME="en_GB.UTF-8"
LC_COLLATE="en_GB.UTF-8"
LC_MONETARY="en_GB.UTF-8"
LC_MESSAGES="en_GB.UTF-8"
LC_PAPER="en_GB.UTF-8"
LC_NAME="en_GB.UTF-8"
LC_ADDRESS="en_GB.UTF-8"
LC_TELEPHONE="en_GB.UTF-8"
LC_MEASUREMENT="en_GB.UTF-8"
LC_IDENTIFICATION="en_GB.UTF-8"
LC_ALL=en_GB.UTF-8
```

Les variables spéciales

Variable	Description
$LINENO	Contient le numéro de la ligne courante du script ou de la fonction
$$	Contient le PID du shell en cours

Variable	Description
$PPID	Contient le PID du processus parent du shell en cours
$0	Contient le nom du script en cours tel que ce nom ait été saisi sur la ligne de commande
$1, $2 ...	Contient respectivement le premier argument, deuxième argument etc passés au script
$#	Contient le nombre d'arguments passés au script
$*	Contient l'ensemble des arguments passés au script
$@	Contient l'ensemble des arguments passés au script

La Commande env

La commande **env** envoie sur la sortie standard les valeurs des variables système de l'environnement de l'utilisateur qui l'invoque :

```
[trainee@centos7 ~]$ env
XDG_SESSION_ID=1
HOSTNAME=centos7.fenestros.loc
SELINUX_ROLE_REQUESTED=
TERM=xterm-256color
SHELL=/bin/bash
HISTSIZE=1000
SSH_CLIENT=10.0.2.2 33896 22
SELINUX_USE_CURRENT_RANGE=
SSH_TTY=/dev/pts/0
LC_ALL=en_GB.UTF-8
USER=trainee
LS_COLORS=rs=0:di=38;5;27:ln=38;5;51:mh=44;38;5;15:pi=40;38;5;11:so=38;5;13:do=38;5;5:bd=48;5;232;38;5;11:cd=48;5
;232;38;5;3:or=48;5;232;38;5;9:mi=05;48;5;232;38;5;15:su=48;5;196;38;5;15:sg=48;5;11;38;5;16:ca=48;5;196;38;5;226
:tw=48;5;10;38;5;16:ow=48;5;10;38;5;21:st=48;5;21;38;5;15:ex=38;5;34:*.tar=38;5;9:*.tgz=38;5;9:*.arc=38;5;9:*.arj
=38;5;9:*.taz=38;5;9:*.lha=38;5;9:*.lz4=38;5;9:*.lzh=38;5;9:*.lzma=38;5;9:*.tlz=38;5;9:*.txz=38;5;9:*.tzo=38;5;9:
*.t7z=38;5;9:*.zip=38;5;9:*.z=38;5;9:*.Z=38;5;9:*.dz=38;5;9:*.gz=38;5;9:*.lrz=38;5;9:*.lz=38;5;9:*.lzo=38;5;9:*.x
z=38;5;9:*.bz2=38;5;9:*.bz=38;5;9:*.tbz=38;5;9:*.tbz2=38;5;9:*.tz=38;5;9:*.deb=38;5;9:*.rpm=38;5;9:*.jar=38;5;9:*
.war=38;5;9:*.ear=38;5;9:*.sar=38;5;9:*.rar=38;5;9:*.alz=38;5;9:*.ace=38;5;9:*.zoo=38;5;9:*.cpio=38;5;9:*.7z=38;5
;9:*.rz=38;5;9:*.cab=38;5;9:*.jpg=38;5;13:*.jpeg=38;5;13:*.gif=38;5;13:*.bmp=38;5;13:*.pbm=38;5;13:*.pgm=38;5;13:
```

```
*.ppm=38;5;13:*.tga=38;5;13:*.xbm=38;5;13:*.xpm=38;5;13:*.tif=38;5;13:*.tiff=38;5;13:*.png=38;5;13:*.svg=38;5;13:
*.svgz=38;5;13:*.mng=38;5;13:*.pcx=38;5;13:*.mov=38;5;13:*.mpg=38;5;13:*.mpeg=38;5;13:*.m2v=38;5;13:*.mkv=38;5;13
:*.webm=38;5;13:*.ogm=38;5;13:*.mp4=38;5;13:*.m4v=38;5;13:*.mp4v=38;5;13:*.vob=38;5;13:*.qt=38;5;13:*.nuv=38;5;13
:*.wmv=38;5;13:*.asf=38;5;13:*.rm=38;5;13:*.rmvb=38;5;13:*.flc=38;5;13:*.avi=38;5;13:*.fli=38;5;13:*.flv=38;5;13:
*.gl=38;5;13:*.dl=38;5;13:*.xcf=38;5;13:*.xwd=38;5;13:*.yuv=38;5;13:*.cgm=38;5;13:*.emf=38;5;13:*.axv=38;5;13:*.a
nx=38;5;13:*.ogv=38;5;13:*.ogx=38;5;13:*.aac=38;5;45:*.au=38;5;45:*.flac=38;5;45:*.mid=38;5;45:*.midi=38;5;45:*.m
ka=38;5;45:*.mp3=38;5;45:*.mpc=38;5;45:*.ogg=38;5;45:*.ra=38;5;45:*.wav=38;5;45:*.axa=38;5;45:*.oga=38;5;45:*.spx
=38;5;45:*.xspf=38;5;45:
MAIL=/var/spool/mail/trainee
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/trainee/.local/bin:/home/trainee/bin
PWD=/home/trainee
LANG=fr_FR.UTF-8
SELINUX_LEVEL_REQUESTED=
HISTCONTROL=ignoredups
SHLVL=1
HOME=/home/trainee
LOGNAME=trainee
SSH_CONNECTION=10.0.2.2 33896 192.168.1.99 22
LESSOPEN=||/usr/bin/lesspipe.sh %s
XDG_RUNTIME_DIR=/run/user/1000
_=/usr/bin/env
OLDPWD=/home/trainee/training
```

La commande peut aussi être utilisée pour fixer une variable lors de l'exécution d'une commande. Par exemple, pour lancer **xterm** avec la variable EDITOR fixée à **vi** :

```
$ env EDITOR=vim xterm
```

Les Scripts Shell

Le but de la suite de cette unité est de vous amener au point où vous êtes capable de comprendre et de déchiffrer les scripts, notamment les scripts de démarrage ainsi que les scripts de contrôle des services.

Écrire des scripts compliqués est en dehors de la portée de cette unité car il nécessite une approche programmation qui ne peut être adressée que lors d'une formation dédiée à l'écriture des scripts.

Exécution

Un script shell est un fichier dont le contenu est lu en entrée standard par le shell. Le contenu du fichier est lu et exécuté d'une manière séquentielle. Afin qu'un script soit exécuté, il suffit qu'il puisse être lu au quel cas le script est exécuté par un shell fils soit en l'appelant en argument à l'appel du shell :

/bin/bash myscript

soit en redirigeant son entrée standard :

/bin/bash < myscript

Dans le cas où le droit d'exécution est positionné sur le fichier script et à condition que celui-ci se trouve dans un répertoire spécifié dans le PATH de l'utilisateur qui le lance, le script peut être lancé en l'appelant simplement par son nom :

myscript

Pour lancer le script sans qu'il soit dans un répertoire du PATH, il convient de se placer dans le répertoire contenant le script et de le lancer ainsi :

./myscript

Dans le cas où le script doit être exécuté par le shell courant, dans les mêmes conditions que l'exemple précédent, et non par un shell fils, il convient de le lancer ainsi :

. myscript

Dans un script il est fortement conseillé d'inclure des commentaires. Les commentaires permettent à d'autres personnes de comprendre le script. Toute ligne de commentaire commence avec le caractère **#**.

Il existe aussi un **pseudo commentaire** qui est placé au début du script. Ce pseudo commentaire permet de stipuler quel shell doit être utilisé pour l'exécution du script. L'exécution du script est ainsi rendu indépendant du shell de l'utilisateur qui le lance. Le pseudo commentaire commence avec les

caractères **#!**. Chaque script commence donc par une ligne similaire à celle-ci :

```
#!/bin/sh
```

Puisque un script contient des lignes de commandes qui peuvent être saisies en shell intéractif, il est souvent issu d'une procédure manuelle. Afin de faciliter la création d'un script il existe une commande, **script**, qui permet d'enregistrer les textes sortis sur la sortie standard, y compris le prompt dans un fichier dénommé **typescript**. Afin d'illustrer l'utilisation de cette commande, saisissez la suite de commandes suivante :

```
[trainee@centos7 ~]$ script
Script started, file is typescript
[trainee@centos7 ~]$ pwd
/home/trainee
[trainee@centos7 ~]$ ls
aac  bca    Desktop    Downloads  fichier1  file   Music     Public     training    Videos  xyz
abc  codes  Documents  errorlog   fichier2  file1  Pictures  Templates  typescript  vitext
[trainee@centos7 ~]$ exit
exit
Script done, file is typescript
[trainee@centos7 ~]$ cat typescript
Script started on Tue 29 Nov 2016 03:58:33 CET
[trainee@centos7 ~]$ pwd
/home/trainee
[trainee@centos7 ~]$ ls
aac  bca    Desktop    Downloads  fichier1  file   Music     Public     training    Videos  xyz
abc  codes  Documents  errorlog   fichier2  file1  Pictures  Templates  typescript  vitext
[trainee@centos7 ~]$ exit
exit

Script done on Tue 29 Nov 2016 03:58:40 CET
```

Cette procédure peut être utilisée pour enregistrer une suite de commandes longues et compliquées afin d'écrire un script.

Pour illustrer l'écriture et l'exécution d'un script, éditez le fichier **myscript** avec **vi** :

```
$ vi myscript [Entrée]
```

Éditez votre fichier ainsi :

```
pwd
ls
```

Sauvegardez votre fichier. Lancez ensuite votre script en passant le nom du fichier en argument à /bin/bash :

```
[trainee@centos7 ~]$ vi myscript
[trainee@centos7 ~]$ /bin/bash myscript
/home/trainee
aac  codes  Downloads  fichier2  myscript    Public       typescript  xyz
abc  Desktop    errorlog   file      Music    Templates  Videos
bca  Documents  fichier1   file1     Pictures    training   vitext
```

Lancez ensuite le script en redirigeant son entrée standard :

```
[trainee@centos7 ~]$ /bin/bash < myscript
/home/trainee
aac  codes  Downloads  fichier2  myscript    Public       typescript  xyz
abc  Desktop    errorlog   file      Music    Templates  Videos
bca  Documents  fichier1   file1     Pictures    training   vitext
```

Pour lancer le script en l'appelant simplement par son nom, son chemin doit être inclus dans votre PATH:

```
[trainee@centos7 ~]$ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/trainee/.local/bin:/home/trainee/bin
```

Dans le cas de RHEL/CentOS, même si PATH contient $HOME/bin, le répertoire n'existe pas :

```
[trainee@centos7 ~]$ ls
aac  codes      Downloads  fichier2  myscript  Public     typescript  xyz
```

```
abc  Desktop    errorlog   file      Music      Templates  Videos
bca  Documents  fichier1   file1     Pictures   training   vitext
```

Créez donc ce répertoire :

```
[trainee@centos7 ~]$ mkdir bin
```

Ensuite déplacez votre script dans ce répertoire et rendez-le exécutable pour votre utilisateur :

```
[trainee@centos7 ~]$ mv myscript ~/bin
[trainee@centos7 ~]$ chmod u+x ~/bin/myscript
```

Exécutez maintenant votre script en l'appelant par son nom à partir du répertoire **/tmp** :

```
[trainee@centos7 tmp]$ myscript
/tmp
hsperfdata_root  systemd-private-e526abcf335b40949dfc725f28456502-cups.service-u0xGiL
```

Placez-vous dans le répertoire contenant le script et saisissez les commandes suivantes :

- ./myscript
- . myscript

```
[trainee@centos7 tmp]$ cd ~/bin
[trainee@centos7 bin]$ ./myscript
/home/trainee/bin
myscript
[trainee@centos7 bin]$ . myscript
/home/trainee/bin
myscript
```

A faire : Notez bien la différence entre les sorties de cette dernière commande et la précédente. Expliquez pourquoi.

La commande read

La commande **read** lit son entrée standard et affecte les mots saisis dans la ou les variable(s) passée(s) en argument. La séparation entre le contenu des variables est l'espace. Par conséquent il est intéressant de noter les exemples suivants :

```
[trainee@centos7 bin]$ read var1 var2 var3 var4
fenestros edu is great!
[trainee@centos7 bin]$ echo $var1
fenestros
[trainee@centos7 bin]$ echo $var2
edu
[trainee@centos7 bin]$ echo $var3
is
[trainee@centos7 bin]$ echo $var4
great!
```

Important: Notez que chaque champs a été placé dans une variable différente. Notez aussi que par convention les variables déclarées par des utilisateurs sont en miniscules afin de les distinguer des variables système qui sont en majuscules.

```
[trainee@centos7 bin]$ read var1 var2
fenestros edu is great!
[trainee@centos7 bin]$ echo $var1
fenestros
[trainee@centos7 bin]$ echo $var2
edu is great!
```

Important : Notez que dans le deuxième cas, le reste de la ligne après le mot *fenestros* est mis dans **$var2**.

Code de retour

La commande **read** renvoie un code de retour de **0** dans le cas où elle ne reçoit pas l'information **fin de fichier** matérialisée par les touches Ctrl+D.
Le contenu de la variable **var** peut être vide et la valeur du code de retour **0** grâce à l'utilisation de la touche Entrée :

```
[trainee@centos7 bin]$ read var
```

↵ Entrée

```
[trainee@centos7 bin]$ echo $?
0
[trainee@centos7 bin]$ echo $var

[trainee@centos7 bin]$
```

Le contenu de la variable **var** peut être vide et la valeur du code de retour **autre que 0** grâce à l'utilisation des touches Ctrl+D :

```
[trainee@centos7 bin]$ read var
```

Ctrl+D

```
[trainee@centos7 bin]$ echo $?
1
[trainee@centos7 bin]$ echo $var

[trainee@centos7 bin]$
```

La variable IFS

La variable IFS contient par défaut les caractères Espace, Tab et Entrée :

```
[trainee@centos7 bin]$ echo "$IFS" | od -c
0000000      \t  \n  \n
0000004
```

Important : La commande **od** (*Octal Dump*) renvoie le contenu d'un fichier ou de l'entrée standard au format octal. Ceci est utile afin de visualiser les caractères non-imprimables. L'option **-c** permet de sélectionner des caractères ASCII ou des backslash dans le fichier ou dans le contenu fourni à l'entrée standard.

La valeur de cette variable définit donc le séparateur de mots lors de la saisie des contenus des variables avec la commande **read**. La valeur de la variable **IFS** peut être modifiée :

```
[trainee@centos7 bin]$ OLDIFS="$IFS"
[trainee@centos7 bin]$ IFS=":"
[trainee@centos7 bin]$ echo "$IFS" | od -c
0000000   :  \n
0000002
```

De cette façon l'espace redevient un caractère normal :

```
[trainee@centos7 bin]$ read var1 var2 var3
fenestros:edu is:great!
[trainee@centos7 bin]$ echo $var1
fenestros
[trainee@centos7 bin]$ echo $var2
edu is
[trainee@centos7 bin]$ echo $var3
great!
```

Restaurez l'ancienne valeur de IFS avec la commande IFS=“$OLDIFS”

```
[trainee@centos7 bin]$ IFS="$OLDIFS"
[trainee@centos7 bin]$ echo "$IFS" | od -c
0000000      \t  \n  \n
0000004
```

La commande test

La commande **test** peut être utilisée avec deux syntaxes :

test *expression*

ou

[Espace*expression*Espace**]**

Tests de Fichiers

Test	Description
-f fichier	Retourne vrai si fichier est d'un type standard
-d fichier	Retourne vrai si fichier est d'un type répertoire
-r fichier	Retourne vrai si l'utilisateur peut lire fichier
-w fichier	Retourne vrai si l'utilisateur peut modifier fichier
-x fichier	Retourne vrai si l'utilisateur peut exécuter fichier
-e fichier	Retourne vrai si fichier existe
-s fichier	Retourne vrai si fichier n'est pas vide
fichier1 -nt fichier2	Retourne vrai si fichier1 est plus récent que fichier2
fichier1 -ot fichier2	Retourne vrai si fichier1 est plus ancien que fichier2
fichier1 -ef fichier2	Retourne vrai si fichier1 est identique à fichier2

Testez si le fichier **a100** est un fichier ordinaire :

```
[trainee@centos7 bin]$ cd ../training/
[trainee@centos7 training]$ test -f a100
[trainee@centos7 training]$ echo $?
0
[trainee@centos7 training]$ [ -f a100 ]
[trainee@centos7 training]$ echo $?
0
```

Testez si le fichier a101 existe :

```
[trainee@centos7 training]$ [ -f a101 ]
[trainee@centos7 training]$ echo $?
1
```

Testez si /home/trainee/training est un répertoire :

```
[trainee@centos7 training]$ [ -d /home/trainee/training ]
[trainee@centos7 training]$ echo $?
0
```

Tests de chaînes de caractère

Test	Description
-n chaîne	Retourne vrai si chaîne n'est pas de longueur 0
-z chaîne	Retourne vrai si chaîne est de longueur 0
string1 = string2	Retourne vrai si string1 est égale à string2
string1 != string2	Retourne vrai si string1 est différente de string2
string1	Retourne vrai si string1 n'est pas vide

Testez si les deux chaînes sont égales :

```
[trainee@centos7 training]$ string1="root"
```

```
[trainee@centos7 training]$ string2="fenestros"
[trainee@centos7 training]$ [ $string1 = $string2 ]
[trainee@centos7 training]$ echo $?
1
```

Testez si la string1 n'a pas de longueur 0 :

```
[trainee@centos7 training]$ [ -n $string1 ]
[trainee@centos7 training]$ echo $?
0
```

Testez si la string1 a une longueur de 0 :

```
[trainee@centos7 training]$ [ -z $string1 ]
[trainee@centos7 training]$ echo $?
1
```

Tests sur des nombres

Test	Description
value1 -eq value2	Retourne vrai si value1 est égale à value2
value1 -ne value2	Retourne vrai si value1 n'est pas égale à value2
value1 -lt value2	Retourne vrai si value1 est inférieure à value2
value1 -le value2	Retourne vrai si value1 est inférieur ou égale à value2
value1 -gt value2	Retourne vrai si value1 est supérieure à value2
value1 -ge value2	Retourne vrai si value1 est supérieure ou égale à value2

Comparez les deux nombres **value1** et **value2** :

```
[trainee@centos7 training]$ read value1
35
[trainee@centos7 training]$ read value2
23
```

```
[trainee@centos7 training]$ [ $value1 -lt $value2 ]
[trainee@centos7 training]$ echo $?
1
[trainee@centos7 training]$ [ $value2 -lt $value1 ]
[trainee@centos7 training]$ echo $?
0
[trainee@centos7 training]$ [ $value2 -eq $value1 ]
[trainee@centos7 training]$ echo $?
1
```

Les opérateurs

Test	Description
!expression	Retourne vrai si expression est fausse
expression1 -a expression2	Représente un et logique entre expression1 et expression2
expression1 -o expression2	Représente un ou logique entre expression1 et expression2
\(expression\)	Les parenthèses permettent de regrouper des expressions

Testez si $file n'est pas un répertoire :

```
[trainee@centos7 training]$ file=a100
[trainee@centos7 training]$ [ ! -d $file ]
[trainee@centos7 training]$ echo $?
0
```

Testez si $directory est un répertoire **et** si l'utilisateur à le droit de le traverser :

```
[trainee@centos7 training]$ directory=/usr
[trainee@centos7 training]$ [ -d $directory -a -x $directory ]
[trainee@centos7 training]$ echo $?
0
```

Testez si l'utilisateur peut écrire dans le fichier a100 **et** /usr est un répertoire **ou** /tmp est un répertoire :

```
[trainee@centos7 training]$ [ -w a100 -a \( -d /usr -o -d /tmp \) ]
[trainee@centos7 training]$ echo $?
0
```

Tests d'environnement utilisateur

Test	Description
-o option	Retourne vrai si l'option du shell “option” est activée

```
[trainee@centos7 training]$ [ -o allexport ]
[trainee@centos7 training]$ echo $?
1
```

La commande [[expression]]

La commande **[[**Espace**expression**Espace**]]** est une amélioration de la commande **test**. Les opérateurs de la commande test sont compatibles avec la commande [[expression]] sauf **-a** et **-o** qui sont remplacés par **&&** et **||** respectivement :

Test	Description
!expression	Retourne vrai si expression est fausse
expression1 && expression2	Représente un **et** logique entre expression1 et expression2
expression1 || expression2	Représente un **ou** logique entre expression1 et expression2
(expression)	Les parenthèses permettent de regrouper des expressions

D'autres opérateurs ont été ajoutés :

Test	Description
string = modele	Retourne vrai si chaîne correspond au modèle
string != modele	Retourne vrai si chaîne ne correspond pas au modèle
string1 < string2	Retourne vrai si string1 est lexicographiquement avant string2
string1 > string2	Retourne vrai si string1 est lexicographiquement après string2

Testez si l'utilisateur peut écrire dans le fichier a100 **et** /usr est un répertoire **ou** /tmp est un répertoire :

```
[trainee@centos7 training]$ [[ -w a100 && ( -d /usr || -d /tmp ) ]]
[trainee@centos7 training]$ echo $?
0
```

Opérateurs du shell

Opérateur	Description
Commande1 && Commande2	Commande 2 est exécutée si la première commande renvoie un code vrai
Commande1 || Commande2	Commande 2 est exécutée si la première commande renvoie un code faux

```
[trainee@centos7 training]$ [[ -d /root ]] && echo "The root directory exists"
The root directory exists
[trainee@centos7 training]$ [[ -d /root ]] || echo "The root directory exists"
[trainee@centos7 training]$
```

L'arithmétique

La commande expr

La commande **expr** prend la forme :

expr Espace value1 Espace *opérateur* Espace value2 Entrée

ou

expr Tab value1 Tab *opérateur* Tab value2 Entrée

ou

expr Espace chaîne Espace : Espace *expression_régulière* Entrée

ou

expr Tab chaîne Tab : Tab *expression_régulière* Entrée

Opérateurs Arithmétiques

Opérateur	Description
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo
\(\)	Parenthèses

Opérateurs de Comparaison

Opérateur	Description
\<	Inférieur
\<=	Inférieur ou égal
\>	Supérieur
\>=	Supérieur ou égal
=	égal
!=	inégal

Opérateurs Logiques

Opérateur	Description
|	ou logique
\&	et logique

Ajoutez 2 à la valeur de $x :

```
[trainee@centos7 training]$ x=2
[trainee@centos7 training]$ expr $x + 2
4
```

Si les espaces sont retirés, le résultat est tout autre :

```
[trainee@centos7 training]$ expr $x+2
2+2
```

Les opérateurs doivent être protégés :

```
[trainee@centos7 training]$ expr $x * 2
expr: syntax error
[trainee@centos7 training]$ expr $x \* 2
4
```

Mettez le résultat d'un calcul dans une variable :

```
[trainee@centos7 training]$ resultat=`expr $x + 10`
[trainee@centos7 training]$ echo $resultat
12
```

La commande let

La commande let est l'équivalent de la commande ((expression)). La commande ((expression)) est une amélioration de la commande **expr** :

- plus grand nombre d'opérateurs
- pas besoin d'espaces ou de tabulations entre les arguments
- pas besoin de préfixer les variables d'un $
- les caractères spéciaux du shell n'ont pas besoin d'être protégés
- les affectations se font dans la commande
- exécution plus rapide

Opérateurs Arithmétiques

Opérateur	Description
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo
^	Puissance

Opérateurs de comparaison

Opérateur	Description
<	Inférieur
<=	Inférieur ou égal
>	Supérieur
>=	Supérieur ou égal
==	égal
!=	inégal

Opérateurs Logiques

Opérateur	Description
&&	et logique
||	ou logique
!	négation logique

Opérateurs travaillant sur les bits

Opérateur	Description
~	négation binaire
>>	décalage binaire à droite
<<	décalage binaire à gauche

Opérateur	Description
&	et binaire
|	ou binaire
^	ou exclusif binaire

```
[trainee@centos7 training]$ x=2
[trainee@centos7 training]$ ((x=$x+10))
[trainee@centos7 training]$ echo $x
12
[trainee@centos7 training]$ ((x=$x+20))
[trainee@centos7 training]$ echo $x
32
```

Structures de contrôle

If

La syntaxe de la commande If est la suivante :

```
if condition
then
    commande(s)
else
    commande(s)
fi
```

ou :

```
if condition
then
    commande(s)
    commande(s)
```

```
fi
```

ou encore :

```
if condition
then
     commande(s)
elif condition
then
     commande(s)
elif condition
then
     commande(s)
else
    commande(s)

fi
```

Créez le script **user_check** suivant :

```
#!/bin/bash
if [ $# -ne 1 ] ; then
  echo "Mauvais nombre d'arguments"
  echo "Usage : $0 nom_utilisateur"
  exit 1
fi
if grep "^$1:" /etc/passwd > /dev/null
then
  echo "Utilisateur $1 est défini sur ce système"
else
  echo "Utilisateur $1 n'est pas défini sur ce système"
fi
exit 0
```

Testez-le :

```
[trainee@centos7 training]$ chmod 770 user_check
[trainee@centos7 training]$ ./user_check
Mauvais nombre d'arguments
Usage : ./user_check nom_utilisateur
[trainee@centos7 training]$ ./user_check root
Utilisateur root est défini sur ce système
[trainee@centos7 training]$ ./user_check mickey mouse
Mauvais nombre d'arguments
Usage : ./user_check nom_utilisateur
[trainee@centos7 training]$ ./user_check "mickey mouse"
Utilisateur mickey mouse n'est pas défini sur ce système
```

case

La syntaxe de la commande case est la suivante :

```
case $variable in
modele1) commande
   ...
   ;;
modele2) commande
   ...
   ;;
modele3 | modele4 | modele5 ) commande
   ...
   ;;
esac
```

Exemple

```
  case "$1" in
      start)
          start
          ;;
      stop)
          stop
          ;;
      restart|reload)
          stop
          start
          ;;
      status)
          status
          ;;
      *)
          echo $"Usage: $0 {start|stop|restart|status}"
          exit 1
esac
```

Important : L'exemple indique que dans le cas où le premier argument qui suit le nom du script contenant la clause **case** est **start**, la fonction *start* sera exécutée. La fonction *start* n'a pas besoin d'être définie dans **case** et est donc en règle générale définie en début de script. La même logique est appliquée dans le cas où le premier argument est **stop**, **restart** ou **reload** et **status**. Dans tous les autres cas, représentés par une étoile, **case** affichera la ligne **Usage: $0 {start|stop|restart|status}** où $0 est remplacé par le nom du script.

Boucles

for

La syntaxe de la commande for est la suivante :

```
for variable in liste_variables
do
    commande(s)
done
```

while

La syntaxe de la commande while est la suivante :

```
while condition
do
    commande(s)
done
```

Exemple

```
U=1
while [ $U -lt $MAX_ACCOUNTS ]
do
useradd fenestros"$U" -c fenestros"$U" -d /home/fenestros"$U" -g staff -G audio,fuse -s /bin/bash 2>/dev/null
useradd fenestros"$U"$ -g machines -s /dev/false -d /dev/null 2>/dev/null
echo "Compte fenestros$U créé"
let U=U+1
```

```
done
```

Scripts de Démarrage

Quand Bash est appelé en tant que shell de connexion, il exécute des scripts de démarrage dans l'ordre suivant :

- **/etc/profile**,
- **~/.bash_profile** ou **~/.bash_login** ou **~/.profile** selon la distribution,

Dans le cas de RHEL/CentOS, le système exécute le fichier **~/.bash_profile**.

Quand un shell de login se termine, Bash exécute le fichier **~/.bash_logout** si celui-ci existe.

Quand bash est appelé en tant que shell interactif qui n'est pas un shell de connexion, il exécute le script **~/.bashrc**.

LAB #1- Scripts de Démarrage

A faire : En utilisant vos connaissances acquises dans ce module, expliquez les scripts suivants ligne par ligne.

~/.bash_profile

```
[trainee@centos7 training]$ cat ~/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
```

```
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin

export PATH
```

~/.bashrc

```
[trainee@centos7 training]$ cat ~/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
```

Le Langage SQL

Installation de MariaDB

Pour installer MariaDB, utilisez yum :

```
[root@centos7 ~]# yum install mariadb mariadb-server
```

Important Vous pouvez également installer MariaDB à partir des sources disponibles sur le site **https://downloads.mariadb.org/**.

Démarrage du Serveur

Le serveur MariaDB peut être démarré par l'utilisation d'une de deux méthodes différentes.

Le Service MariaDB

Pour activer le service **mariadb** il convient d'utiliser la commande **systemctl** :

```
[root@centos7 ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to
/usr/lib/systemd/system/mariadb.service.
```

Le démarrage se fait également avec systemctl :

```
[root@centos7 ~]# systemctl start mariadb
[root@centos7 ~]# systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2017-10-31 12:34:38 CET; 5s ago
  Process: 9769 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
  Process: 9737 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
 Main PID: 9768 (mysqld_safe)
   CGroup: /system.slice/mariadb.service
           ├─9768 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─9930 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql...
```

```
Oct 31 12:34:35 centos7.fenestros.loc systemd[1]: Starting MariaDB database s...
Oct 31 12:34:35 centos7.fenestros.loc mariadb-prepare-db-dir[9737]: Database ...
Oct 31 12:34:36 centos7.fenestros.loc mysqld_safe[9768]: 171031 12:34:36 mysq...
Oct 31 12:34:36 centos7.fenestros.loc mysqld_safe[9768]: 171031 12:34:36 mysq...
Oct 31 12:34:38 centos7.fenestros.loc systemd[1]: Started MariaDB database se...
Hint: Some lines were ellipsized, use -l to show in full.
```

Le service mariadb appel un script appelé **mysqld_safe** qui lance le serveur et crée un journal d'erreur. Il relance le serveur en cas d'arrêt intempestif.

Invocation Directe

Il est aussi possible d'invoquer directement le binaire **mysqld** en spécifiant manuellement le fichier de configuration de MariaDB, le fichier d'erreurs ainsi que le nom de l'utilisateur. Par exemple :

```
# /usr/libexec/mysqld --defaults-file=/chemin/my.cnf --log-error=/chemin/nom_log --user=mysql &
```

Arrêt du Serveur

Le Service MariaDB

Il est possible d'arrêter le service mariadb avec la commande stop :

```
[root@centos7 ~]# systemctl stop mariadb
[root@centos7 ~]# systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Tue 2017-10-31 12:38:12 CET; 3s ago
  Process: 9769 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
  Process: 9768 ExecStart=/usr/bin/mysqld_safe --basedir=/usr (code=exited, status=0/SUCCESS)
  Process: 9737 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
```

```
 Main PID: 9768 (code=exited, status=0/SUCCESS)

Oct 31 12:34:35 centos7.fenestros.loc systemd[1]: Starting MariaDB database s...
Oct 31 12:34:35 centos7.fenestros.loc mariadb-prepare-db-dir[9737]: Database ...
Oct 31 12:34:36 centos7.fenestros.loc mysqld_safe[9768]: 171031 12:34:36 mysq...
Oct 31 12:34:36 centos7.fenestros.loc mysqld_safe[9768]: 171031 12:34:36 mysq...
Oct 31 12:34:38 centos7.fenestros.loc systemd[1]: Started MariaDB database se...
Oct 31 12:38:09 centos7.fenestros.loc systemd[1]: Stopping MariaDB database s...
Oct 31 12:38:12 centos7.fenestros.loc systemd[1]: Stopped MariaDB database se...
Hint: Some lines were ellipsized, use -l to show in full.
[root@centos7 ~]# systemctl start mariadb
[root@centos7 ~]# systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2017-10-31 12:38:25 CET; 1s ago
  Process: 11254 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
  Process: 11222 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
 Main PID: 11253 (mysqld_safe)
   CGroup: /system.slice/mariadb.service
           ├─11253 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─11420 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysq...

Oct 31 12:38:22 centos7.fenestros.loc systemd[1]: Starting MariaDB database s...
Oct 31 12:38:23 centos7.fenestros.loc mariadb-prepare-db-dir[11222]: Database...
Oct 31 12:38:23 centos7.fenestros.loc mysqld_safe[11253]: 171031 12:38:23 mys...
Oct 31 12:38:23 centos7.fenestros.loc mysqld_safe[11253]: 171031 12:38:23 mys...
Oct 31 12:38:25 centos7.fenestros.loc systemd[1]: Started MariaDB database se...
Hint: Some lines were ellipsized, use -l to show in full.
```

La Commande mysqladmin

La comande **mysqladmin** peut aussi être utilisée pour arrêter le serveur à condition que l'utilisateur qui l'invoque possède le privilège **shutdown**. Par exemple :

```
[root@centos7 ~]# /bin/mysqladmin -uroot -p shutdown
Enter password:
[root@centos7 ~]# systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Tue 2017-10-31 12:53:09 CET; 4s ago
  Process: 11254 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
  Process: 11253 ExecStart=/usr/bin/mysqld_safe --basedir=/usr (code=exited, status=0/SUCCESS)
  Process: 11222 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
 Main PID: 11253 (code=exited, status=0/SUCCESS)

Oct 31 12:38:22 centos7.fenestros.loc systemd[1]: Starting MariaDB database s...
Oct 31 12:38:23 centos7.fenestros.loc mariadb-prepare-db-dir[11222]: Database...
Oct 31 12:38:23 centos7.fenestros.loc mysqld_safe[11253]: 171031 12:38:23 mys...
Oct 31 12:38:23 centos7.fenestros.loc mysqld_safe[11253]: 171031 12:38:23 mys...
Oct 31 12:38:25 centos7.fenestros.loc systemd[1]: Started MariaDB database se...
Hint: Some lines were ellipsized, use -l to show in full.
```

Important - Le nom d'utilisateur peut être accoler ou non à l'option **-u**. Par exemple **-uroot** et **-u root** sont tous les deux correctes. Par contre, si vous souhaitez spécifier le mot de passe de l'utilisateur dans la ligne de commande, celui-ci **doit** être accoler à l'option **-p**.

Configuration

Votre première prise en mains de MariaDB doit débuter par la commande **mysql** :

Le Client MySQL

Utilisation

MariaDB dispose d'un outil client permettant de se connecter et d'envoyer des commandes SQL au serveur.

Pour démarrer une connexion au serveur MariaDB nous pouvons utiliser un C.L.I. sous Linux.

Les paramètres les plus courants sont :

```
$ mysql –u root –p Databasename [Entrée]
```

Où :

- **-u**
  - donne le nom de l'utilisateur. Si vous renoncez à l'option –u , le nom de login sera utilisé sous Unix/Linux et le nom ODBC sous Windows.
- **-p**
  - permet de saisir le mot de passe de MariaDB, cette option est obligatoire lorsque les utilisateurs MariaDB sont sécurisés, c'est-à-dire qu'ils ont un mot de passe.
- **-h**
  - permet de préciser le nom de l'hôte qui héberge MariaDB. Par défaut le port de communication est 3306 il est possible de changer ce port soit en configurant l'instance MariaDB soit en configurant le fichier **My.ini** sous Windows™ ou **my.cnf** sous Unix/Linux. (Attention: vérifiez qu'un par feu ne bloque pas ce port). Enfin vous devez vous assurer que le serveur MariaDB est configuré de sorte à autoriser les accès depuis votre ordinateur local.
- **- -protocol = name**
  - cette option permet de spécifier le protocole de communication à utiliser. Il est rarement nécessaire d'indiquer cette option, puisque MariaDB choisit le bon protocole par défaut. Lorsque le client mysql et le serveur MariaDB ne tournent pas sur le même ordinateur, le seul protocole réseau possible est le **TCP**, dans ce cas n'oubliez pas de préciser le paramètre –h.
- **-P n**
  - permet de préciser le port de communication utilisée pour se connecter à MariaDB. Cette option est effective que lorsque la communication

passe par TCP/IP.

- **- -default-character-set = nom**
  - cette option indique le jeu de caractères utilisés dans le cadre de communication entre mysql et le serveur MariaDB. En théorie, il s'agit du même jeu de caractères que celui utilisé par défaut dans la vie de commandes sous Windows ou la console sous Linux. Les jeux de caractères pris en charge par MariaDB sont notamment Latin 1 (ISO-8559-1), latin 2 (ISO-8559-2), UTF-8 (unicode) et cp850 (le jeu de caractères DOS pour l'Europe occidentale).

- Databasename
  - ce dernier paramètre indique à MariaDB le nom d'une base de données, ce qui permet de l'utiliser directement dès l'ouverture de MariaDB. Si vous désirez changer ici de base de données après sa connexion vous pouvez utiliser la commande SQL **USE <NOM_DB>. Exemple : $ mysql -u root -p -h server –protocol=tcp [Entrée] $ Password :** [Entrée]

Options

Dans la console MariaDB nous avons diverses options.

Abréviation	Commande	Signification
\c	clear	Annule une commande en cours de saisie
\h	help	Affiche la liste des commandes
\q	exit ou quit	Ferme la connexion à MariaDB. Sous Unix/Linux, il est possible utilisé le raccourci ctrl+D
\s	status	Affiche les informations de statut du serveur MariaDB
\T[f]	tee[filename]	Enregistre tous ce qui apparaît dans la fenêtre de commandes dans le fichier indiqué
L	notee	Ferme tee. Le protocole doit être repris à tout instant avec tee ou \T. Il n'est pas nécessaire de saisir de nouveau le nom de fichier.
\u db	use database	La base de données saisie devient la base de données par défaut
\. Fn	source file name	exécute les commandes SQL contenues dans le fichier spécifié. Les commandes doivent à séparer par des pointsvirgules.

NB : la fonction \c n'a aucun effet dans les chaînes de caractères(“” ou ' ').

Notez aussi que MariaDB se souvient des dernières commandes grâce aux touches **flèche vers le haut** et **flèche vers le bas**.

LAB #2 - Configuration de Base

Saisissez donc la commande **mysql** :

```
[root@centos7 ~]# systemctl start mariadb
[root@centos7 ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

pour visualiser la liste des bases de données par défaut :

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
7 rows in set (0.01 sec)

MariaDB [(none)]>
```

Ensuite changez de base de données avec la commande :

```
MariaDB [(none)]> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mysql]>
```

Afin de consulter les tables présentes dans la base, utilisez la commande :

```
MariaDB [mysql]> SHOW TABLES;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| ndb_binlog_index          |
| plugin                    |
| proc                      |
| procs_priv                |
| proxies_priv              |
| servers                   |
| slow_log                  |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
```

```
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
24 rows in set (0.00 sec)

MariaDB [mysql]>
```

Pour consulter une table spécifique, utilisez la commande **DESCRIBE** :

```
MariaDB [mysql]> DESCRIBE user;
+------------------------+-----------------------------------+------+-----+---------+-------+
| Field                  | Type                              | Null | Key | Default | Extra |
+------------------------+-----------------------------------+------+-----+---------+-------+
| Host                   | char(60)                          | NO   | PRI |         |       |
| User                   | char(16)                          | NO   | PRI |         |       |
| Password               | char(41)                          | NO   |     |         |       |
| Select_priv            | enum('N','Y')                     | NO   |     | N       |       |
| Insert_priv            | enum('N','Y')                     | NO   |     | N       |       |
| Update_priv            | enum('N','Y')                     | NO   |     | N       |       |
| Delete_priv            | enum('N','Y')                     | NO   |     | N       |       |
| Create_priv            | enum('N','Y')                     | NO   |     | N       |       |
| Drop_priv              | enum('N','Y')                     | NO   |     | N       |       |
| Reload_priv            | enum('N','Y')                     | NO   |     | N       |       |
| Shutdown_priv          | enum('N','Y')                     | NO   |     | N       |       |
| Process_priv           | enum('N','Y')                     | NO   |     | N       |       |
| File_priv              | enum('N','Y')                     | NO   |     | N       |       |
| Grant_priv             | enum('N','Y')                     | NO   |     | N       |       |
| References_priv        | enum('N','Y')                     | NO   |     | N       |       |
| Index_priv             | enum('N','Y')                     | NO   |     | N       |       |
| Alter_priv             | enum('N','Y')                     | NO   |     | N       |       |
| Show_db_priv           | enum('N','Y')                     | NO   |     | N       |       |
| Super_priv             | enum('N','Y')                     | NO   |     | N       |       |
| Create_tmp_table_priv  | enum('N','Y')                     | NO   |     | N       |       |
```

```
| Lock_tables_priv       | enum('N','Y')                          | NO   |     | N       |       |
| Execute_priv           | enum('N','Y')                          | NO   |     | N       |       |
| Repl_slave_priv        | enum('N','Y')                          | NO   |     | N       |       |
| Repl_client_priv       | enum('N','Y')                          | NO   |     | N       |       |
| Create_view_priv       | enum('N','Y')                          | NO   |     | N       |       |
| Show_view_priv         | enum('N','Y')                          | NO   |     | N       |       |
| Create_routine_priv    | enum('N','Y')                          | NO   |     | N       |       |
| Alter_routine_priv     | enum('N','Y')                          | NO   |     | N       |       |
| Create_user_priv       | enum('N','Y')                          | NO   |     | N       |       |
| Event_priv             | enum('N','Y')                          | NO   |     | N       |       |
| Trigger_priv           | enum('N','Y')                          | NO   |     | N       |       |
| Create_tablespace_priv | enum('N','Y')                          | NO   |     | N       |       |
| ssl_type               | enum('','ANY','X509','SPECIFIED')      | NO   |     |         |       |
| ssl_cipher             | blob                                   | NO   |     | NULL    |       |
| x509_issuer            | blob                                   | NO   |     | NULL    |       |
| x509_subject           | blob                                   | NO   |     | NULL    |       |
| max_questions          | int(11) unsigned                       | NO   |     | 0       |       |
| max_updates            | int(11) unsigned                       | NO   |     | 0       |       |
| max_connections        | int(11) unsigned                       | NO   |     | 0       |       |
| max_user_connections   | int(11)                                | NO   |     | 0       |       |
| plugin                 | char(64)                               | NO   |     |         |       |
| authentication_string  | text                                   | NO   |     | NULL    |       |
+------------------------+----------------------------------------+------+-----+---------+-------+
42 rows in set (0.00 sec)

MariaDB [mysql]>
```

Pour visualiser la liste des utilisateurs autorisés pour MariaDB, utilisez la commande suivante :

```
MariaDB [mysql]> SELECT host, user, password FROM user;
+-----------------------+-------+-----------+
| host                  | user  | password  |
+-----------------------+-------+-----------+
| localhost             | root  |           |
```

```
| centos7.fenestros.loc | root  |           |
| 127.0.0.1             | root  |           |
| ::1                   | root  |           |
+-----------------------+-------+-----------+
7 rows in set (0.00 sec)

MariaDB [mysql]>
```

Vous noterez que l'utilisateur root, l'administrateur de MariaDB, n'a pas de mot de passe !

Il faut par conséquence en définir un en urgence !

Pour sortir de l'administration de MariaDB, utilisez la commande **exit** :

```
MariaDB [mysql]> exit
Bye
[root@centos7 ~]#
```

Pour définir le mot de passe **fenestros** pour root, il convient de saisir la commande suivante :

```
[root@centos7 ~]# mysqladmin -u root password fenestros
```

Lors de la prochaine tentative de connexion en tant que root, vous obtiendrez un message d'erreur car le mot de passe est maintenant non-null :

```
[root@centos7 ~]# mysql -u root
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
```

Pour vous connecter en tant que l'administrateur de la base il faut maintenant utiliser la commande suivante :

```
[root@centos7 ~]# mysql -u root -p mysql
Enter password: fenestros
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [mysql]>
```

Important - Notez l'utilisation de l'option **-p** qui indique à **MariaDB** que vous souhaitez saisir un mot de passe. Le mot de passe n'est **pas** mysql mais **fenestros**, comme démontre l'exemple ci-dessus. Bien entendu le mot de passe **fenestros** n'apparaît pas réellement en clair. L'utilisation de l'option **mysql** dans la ligne de commande indique simplement que nous souhaitons se connecter à la base **mysql** dès la connexion.

Saisissez la commande suivante pour vérifier la table des utilisateurs :

```
MariaDB [mysql]> SELECT host, user, password FROM user;
+-----------------------+-------+-------------------------------------------+
| host                  | user  | password                                  |
+-----------------------+-------+-------------------------------------------+
| localhost             | root  | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| centos7.fenestros.loc | root  |                                           |
| 127.0.0.1             | root  |                                           |
| ::1                   | root  |                                           |
+-----------------------+-------+-------------------------------------------+
7 rows in set (0.00 sec)

MariaDB [mysql]>
```

Important - Notez que non seulement le mot de passe de root est présent mais qu'il a été crypté. Vous avez aussi la possibilité de sécuriser votre installation de MariaDB en utilisant le script **/usr/bin/mysql_secure_installation**.

LAB #3 - Le Langage SQL

Créez maintenant la base de données **CarnetAdresses** :

```
mysql> CREATE DATABASE `CarnetAdresses` ;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Créez ensuite deux tables **familles** et **enfants** dans la base **CarnetAdresses** :

```
MariaDB [(none)]> use CarnetAdresses;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [CarnetAdresses]>
```

```
MariaDB [CarnetAdresses]> CREATE TABLE familles (CodeFamille BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY, Nom
VARCHAR(40), PrenomPere VARCHAR(40) , PrenomMere VARCHAR(40) , Adresse1 VARCHAR(40) , Adresse2 VARCHAR(40),
CodePostal VARCHAR(5), Ville VARCHAR(40), ProfPere VARCHAR(40), ProfMere VARCHAR(40));
Query OK, 0 rows affected (0.03 sec)

MariaDB [CarnetAdresses]>
```

Pour plus de facilité, copiez simplement la requête et collez-la dans votre terminal :

familles

```
CREATE TABLE familles (CodeFamille BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY, Nom VARCHAR(40), PrenomPere VARCHAR(40) , PrenomMere VARCHAR(40) , Adresse1 VARCHAR(40) , Adresse2 VARCHAR(40), CodePostal VARCHAR(5), Ville VARCHAR(40), ProfPere VARCHAR(40), ProfMere VARCHAR(40));
```

```
MariaDB [CarnetAdresses]> CREATE TABLE Enfants ( CodeFamille BIGINT NOT NULL, Prenom VARCHAR(40) NOT NULL, Sexe VARCHAR(1) NOT NULL, DateNaissance DATE NOT NULL, PRIMARY KEY (CodeFamille));
Query OK, 0 rows affected (0.02 sec)

MariaDB [CarnetAdresses]>
```

Pour plus de facilité, copiez simplement la requête et collez-la dans votre terminal :

enfants

```
CREATE TABLE Enfants ( CodeFamille BIGINT NOT NULL, Prenom VARCHAR(40) NOT NULL, Sexe VARCHAR(1) NOT NULL, DateNaissance DATE NOT NULL, PRIMARY KEY (CodeFamille));
```

Utilisez ensuite la commande SHOW pour visualiser le résultat de chaque instruction, par exemple :

```
MariaDB [CarnetAdresses]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| CarnetAdresses     |
| Nombres            |
| mysql              |
```

```
| performance_schema |
| test               |
+--------------------+
6 rows in set (0.00 sec)

MariaDB [CarnetAdresses]>
```

```
MariaDB [CarnetAdresses]> SHOW TABLES FROM CarnetAdresses;
+--------------------------+
| Tables_in_CarnetAdresses |
+--------------------------+
| Enfants                  |
| familles                 |
+--------------------------+
2 rows in set (0.00 sec)

MariaDB [CarnetAdresses]>
```

```
MariaDB [CarnetAdresses]> SHOW COLUMNS FROM familles FROM CarnetAdresses;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| CodeFamille | bigint(20)  | NO   | PRI | NULL    | auto_increment |
| Nom         | varchar(40) | YES  |     | NULL    |                |
| PrenomPere  | varchar(40) | YES  |     | NULL    |                |
| PrenomMere  | varchar(40) | YES  |     | NULL    |                |
| Adresse1    | varchar(40) | YES  |     | NULL    |                |
| Adresse2    | varchar(40) | YES  |     | NULL    |                |
| CodePostal  | varchar(5)  | YES  |     | NULL    |                |
| Ville       | varchar(40) | YES  |     | NULL    |                |
| ProfPere    | varchar(40) | YES  |     | NULL    |                |
| ProfMere    | varchar(40) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
10 rows in set (0.00 sec)
```

```
MariaDB [CarnetAdresses]>
```

```
MariaDB [CarnetAdresses]> SHOW INDEX FROM familles;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+--
----+------------+---------+--------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed |
Null | Index_type | Comment | Index_comment |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+--
----+------------+---------+--------------+
| familles |          0 | PRIMARY  |            1 | CodeFamille | A         |           0 |     NULL | NULL   |
| BTREE      |         |               |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+--
----+------------+---------+--------------+
1 row in set (0.00 sec)

MariaDB [CarnetAdresses]>
```

```
MariaDB [CarnetAdresses]> SHOW TABLE STATUS FROM CarnetAdresses;
+----------+--------+---------+------------+------+----------------+-------------+-----------------+------------
-+-----------+----------------+---------------------+-------------+-----------+--------------------+----------+--
--------------+---------+
| Name     | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length | Index_length
| Data_free | Auto_increment | Create_time         | Update_time | Check_time | Collation          | Checksum |
Create_options | Comment |
+----------+--------+---------+------------+------+----------------+-------------+-----------------+------------
-+-----------+----------------+---------------------+-------------+-----------+--------------------+----------+--
--------------+---------+
| Enfants  | InnoDB |      10 | Compact    |    0 |              0 |       16384 |               0 |           0
|   9437184 |           NULL | 2017-10-30 15:17:13 | NULL        | NULL       | latin1_swedish_ci |     NULL |
|         |
| familles | InnoDB |      10 | Compact    |    0 |              0 |       16384 |               0 |           0
|   9437184 |              1 | 2017-10-30 15:15:49 | NULL        | NULL       | latin1_swedish_ci |     NULL |
|         |
+----------+--------+---------+------------+------+----------------+-------------+-----------------+------------
```

```
-+-----------+---------------+--------------------+-------------+-----------+------------------+----------+--
-------------+---------+
2 rows in set (0.00 sec)

MariaDB [CarnetAdresses]>
```

Créez les enregistrements dans la table **familles** :

data

```
INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Durant', 'Jacques', 'Jane', '23 rue Dutor','','92200', 'Neuilly', 'Plombier',''); INSERT
INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Tortua', 'Benoît', 'Carole', '7 rue Verget','', '75005', 'Paris', 'Cadre', 'Cadre');
INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Portier', 'Pierre', 'Elisabeth', '5 rue Toulet','', '92200', 'Neuilly', 'Dentiste','');
INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Renault', 'Damien', 'Anne', '2 rue Ragon','', '75007', 'Paris', 'Comptable',
'Enseignante'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal ,
Ville , ProfPere , ProfMere) VALUES ('Darduet', 'Jean','','','','','','',''); INSERT INTO familles (Nom ,
PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Rodier',
'Gérard', 'Aurélie', '6 rue Agien','', '77000', 'Fontainebleau', 'Professeur',''); INSERT INTO familles (Nom
, PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Tarte',
'Alla', 'Crème', '1 allée Durond','', '92200', 'Neuilly', 'Cuisinier', 'Cuisinière'); INSERT INTO familles
(Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES
('Cohen', 'David', 'Sarah', '7 Av d\'Eylau','', '75016', 'Paris', 'PDG', 'Assistante Direction'); INSERT
INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Dupont2', 'Bruno', 'Odile', '12 rue Sébastien','', '75008', 'PARIS', 'Electricien',
'Comptable'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville
, ProfPere , ProfMere) VALUES ('Durand2', 'Jacques', 'Jane', '23 rue Dutor','', '92200', 'Neuilly',
'Plombier',''); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal ,
Ville , ProfPere , ProfMere) VALUES ('Darmon2', 'Bruno', 'Béatrice', '2 rue Nicolo','', '13008',
'Marseille', 'Cadre', 'Peintre'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2
, CodePostal , Ville , ProfPere , ProfMere) VALUES ('Darduet2', 'Jean', '','' ,'' ,'' ,'','',''); INSERT
```

```
INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Tarte2', 'Alla', 'Crème', '1 allée Durond','', '92200', 'Neuilly', 'Cuisinier', 'Cuisinière'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Cohen2', 'David', 'Sarah', '7 Av d\'Eylau','', '75016', 'Paris', 'PDG', 'Assistante Direction');
```

```
MariaDB [CarnetAdresses]> INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Durant', 'Jacques', 'Jane', '23 rue Dutor','','92200', 'Neuilly', 'Plombier',''); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Tortua', 'Benoît', 'Carole', '7 rue Verget','', '75005', 'Paris', 'Cadre', 'Cadre'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Portier', 'Pierre', 'Elisabeth', '5 rue Toulet','', '92200', 'Neuilly', 'Dentiste',''); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Renault', 'Damien', 'Anne', '2 rue Ragon','', '75007', 'Paris', 'Comptable', 'Enseignante'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Darduet', 'Jean','','','','','','',''); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Rodier', 'Gérard', 'Aurélie', '6 rue Agien','', '77000', 'Fontainebleau', 'Professeur',''); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Tarte', 'Alla', 'Crème', '1 allée Durond','', '92200', 'Neuilly', 'Cuisinier', 'Cuisinière'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Cohen', 'David', 'Sarah', '7 Av d\'Eylau','', '75016', 'Paris', 'PDG', 'Assistante Direction'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Dupont2', 'Bruno', 'Odile', '12 rue Sébastien','', '75008', 'PARIS', 'Electricien', 'Comptable'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Durand2', 'Jacques', 'Jane', '23 rue Dutor','', '92200', 'Neuilly', 'Plombier',''); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Darmon2', 'Bruno', 'Béatrice', '2 rue Nicolo','', '13008', 'Marseille', 'Cadre', 'Peintre'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Darduet2', 'Jean', '','' ,'' ,'' ,'','',''); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Tarte2', 'Alla', 'Crème', '1 allée Durond','', '92200', 'Neuilly', 'Cuisinier', 'Cuisinière'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Cohen2', 'David', 'Sarah', '7 Av d\'Eylau','', '75016',
```

```
'Paris', 'PDG', 'Assistante Direction');
Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

MariaDB [CarnetAdresses]>
```

Saisissez maintenant chacune des commandes suivantes et expliquez le résultat obtenu. Les instructions sont numérotées pour faciliter vos prises de notes.

```
1 ALTER TABLE Enfants ADD INDEX (Prenom);
2 ALTER TABLE Enfants DROP INDEX Prenom;
3 ALTER TABLE Enfants ADD TelPort VARCHAR(50) NOT NULL;
4 ALTER TABLE Enfants CHANGE TelPort TelPort VARCHAR( 22 ) NOT NULL;
5 ALTER TABLE Enfants ADD Test VARCHAR( 40 ) NOT NULL;
6 ALTER TABLE Enfants DROP Test;
7 SELECT * FROM familles;
8 SELECT * FROM familles LIMIT 8,12;
```

Important - **LIMIT** - Avec un argument, la valeur spécifie le nombre de lignes à retourner depuis le début du jeu de résultat. Si deux arguments sont donnés, le premier indique le décalage du premier enregistrement à retourner, le second donne le nombre maximum d'enregistrement à retourner. Le décalage du premier enregistrement est 0 (pas 1).

```
9 UPDATE familles SET Adresse1 = '120 rue de Vaugirard', CodePostal = '75015', Ville = 'PARIS' WHERE Nom = 'DARDUET' AND PrenomPere = 'Jean' LIMIT 1;
10 DELETE FROM familles WHERE Nom = 'DURANT' AND PrenomPere = 'Jacques';
11 DELETE FROM familles WHERE Ville = 'Neuilly';
12 SELECT * FROM familles LIMIT 8,12;
13 SELECT nom, PrenomPere FROM familles;
14 SELECT Nom AS 'Nom de famille', PrenomPere AS 'Prenom du Père' FROM familles;
15 INSERT INTO Enfants (Prenom, Sexe, CodeFamille) VALUES ('Georges', 'M', '14');
16 SELECT familles.nom, familles.PrenomPere FROM familles, Enfants;
```

Important - Créez la colonne **familles.nb_enfants**.

```
17 UPDATE familles SET nb_enfants = '2' WHERE CodeFamille = '2';
18 UPDATE familles SET nb_enfants = '1' WHERE CodeFamille = '4';
```

19	UPDATE familles SET nb_enfants = '3' WHERE CodeFamille = '5';
20	UPDATE familles SET nb_enfants = '2' WHERE CodeFamille = '6';
21	UPDATE familles SET nb_enfants = '4' WHERE CodeFamille = '8';
22	UPDATE familles SET nb_enfants = '5' WHERE CodeFamille = '9';
23	UPDATE familles SET nb_enfants = '3' WHERE CodeFamille = '11';
24	UPDATE familles SET nb_enfants = '1' WHERE CodeFamille = '12';
25	UPDATE familles SET nb_enfants = '5' WHERE CodeFamille = '14';
26	SELECT SUM(nb_enfants) FROM familles;
27	SELECT MIN(nb_enfants) AS 'Nb enfants minimum', MAX(nb_enfants) AS 'Nb enfants maximum', AVG(nb_enfants) AS 'Nb enfants moyen' FROM familles;
28	SELECT nom FROM familles WHERE ville = 'Paris';
29	SELECT nom FROM familles WHERE ville = 'Paris' OR ville = 'Neuilly';
30	ALTER TABLE Enfants DROP PRIMARY KEY;
31	INSERT INTO Enfants (Prenom, Sexe, CodeFamille) VALUES ('Alex', 'F', '12');
32	INSERT INTO Enfants (Prenom, Sexe, CodeFamille) VALUES ('Mila', 'F', '2');
33	INSERT INTO Enfants (Prenom, Sexe, CodeFamille) VALUES ('Amandine', 'F', '5');
34	SELECT familles.Nom , Enfants.Prenom FROM familles, Enfants WHERE (Enfants.CodeFamille = familles.CodeFamille);
35	SELECT familles.nom, familles.PrenomPere FROM familles, Enfants;
36	SELECT ville, min(nb_enfants) as 'Nb enfants minimum', max(nb_enfants) as ' Nb enfants maximum', avg(nb_enfants) as 'Nb enfants moyen' from familles GROUP BY Ville;
37	SELECT * FROM familles ORDER BY nom;
38	SELECT * FROM familles GROUP BY ville, Nom;
39	UPDATE familles SET Adresse2 = '-';
40	UPDATE familles SET Adresse2 = '___',nb_enfants=2;
41	UPDATE familles SET nb_enfants=4 WHERE Ville = 'Paris';
42	UPDATE familles SET nb_enfants=7 LIMIT 4;
43	UPDATE familles SET nb_enfants=8 ORDER BY Nom LIMIT 3;
44	DELETE FROM familles WHERE Nom = 'Dupont';
45	CREATE TABLE loisirs(Nom VARCHAR(30) NOT NULL);
46	DROP TABLE loisirs;
47	INSERT INTO familles (Nom, PrenomPere) VALUES ('Alouet','Jean'),('Rahtmi','Patrick');

```
48 INSERT INTO familles (PrenomPere) SELECT Enfants.prenom FROM Enfants;
49 ALTER TABLE familles ADD Sports VARCHAR(50) NOT NULL;
50 ALTER TABLE familles ADD INDEX (Sports);
51 ALTER TABLE familles ADD INDEX (PrenomPere), ADD EmailPere VARCHAR( 50 ) NOT NULL;
52 ALTER TABLE familles CHANGE PrenomPere Prenom_Pere VARCHAR( 45 ) NOT NULL;
53 ALTER TABLE familles DROP Sports;
54 ALTER TABLE familles DROP INDEX PrenomPere;
55 ALTER TABLE familles DISABLE KEYS;
56 ALTER TABLE Enfants RENAME familles_enfants;
57 ALTER TABLE familles ORDER BY Prenom_Pere;
58 ALTER TABLE familles ADD Commentaire LONGTEXT NOT NULL ;
59 ALTER TABLE familles ADD FULLTEXT (Commentaire);
60 UPDATE familles SET Commentaire = 'La vie est un long fleuve tranquille' WHERE CodeFamille = '11';
61 UPDATE familles SET Commentaire = 'Paris, capitale de la France est traversée par un fleuve' WHERE CodeFamille ='4';
62 UPDATE familles SET Commentaire = 'Le ruisseau se jette dans la rivière qui se jette dans le FLEUVE' WHERE CodeFamille = '6';
63 SELECT * FROM familles WHERE MATCH (Commentaire) AGAINST ('fleuve');
64 SELECT Nom, Commentaire FROM familles WHERE MATCH (Commentaire) AGAINST ('-capitale+fleuve'IN BOOLEAN MODE);
```