

Domaine d'activité 1 : Support et mise à disposition de services informatiques	
Activité 1.1. Gestion du patrimoine informatique	<ul style="list-style-type: none"> <li>• Recensement et identification des ressources numériques</li> <li>• Exploitation des référentiels, normes et standards adoptés par le prestataire informatique</li> <li>• Mise en place et vérification des niveaux d'habilitation associés à un service • Vérification des conditions de la continuité d'un service informatique • <b>Gestion des sauvegardes</b></li> <li>• Vérification du respect des règles d'utilisation des ressources numériques</li> </ul>

## SAUVEGARDE ET RESTAURATION D'UNE BD

### Table des matières

#### I. Présentation

L'objectif de l'activité est de mettre en place un mécanisme de sauvegarde / restauration d'une base de données stockées sous MariaDB. Elle se réalisera en plusieurs étapes. La mise en place du service et de la base de données, puis l'expérimentation des différents types de sauvegardes et de restauration.

#### II. Préparation du serveur

Le serveur de base de données sera hébergé sur une machine Debian **sans interface graphique**. Si nécessaire, le protocole **ssh** devra être utilisé pour se connecter au serveur.

Installez les paquets mariadb-serveur et openssh sur votre machine virtuelle.

#### III. Installation de la base de données

##### **A. Création**

Connectez-vous au service mysql puis créez une base de données nommée fouilles. Placez-vous dans cette base de données pour y exécuter les deux scripts fournis :

- createFouilles.sql qui contient des requêtes de définition des données ;
- insertFouilles.sql qui contient des requêtes de manipulation des données.

## B. Tests

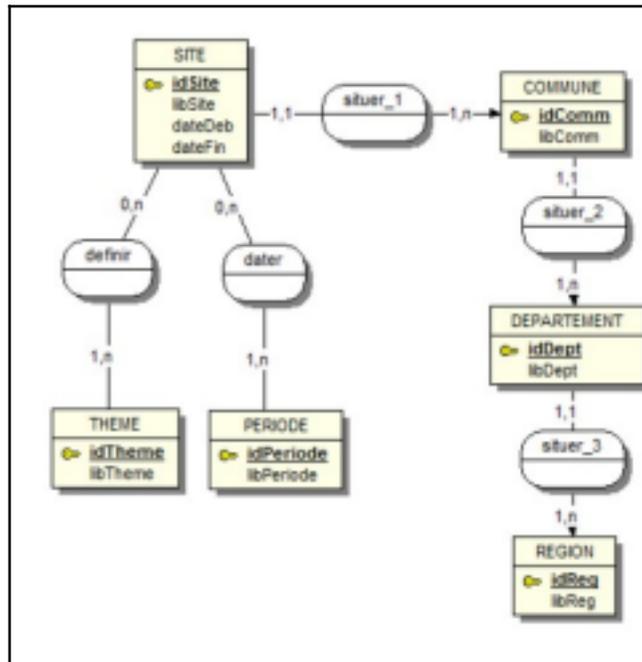
Afin de tester la base de données, écrivez les requêtes retournant :

- la liste des départements classés par ordre alphabétique ;

```
SELECT * FROM departement ORDER BY libDept ASC;
```

- le nom des thèmes et le nombre de sites qu'ils caractérisent.

```
SELECT libTheme, COUNT(*) AS nbSite FROM theme JOIN definir ON theme.idTheme = definir.idTheme join site on definir.idSite = site.idSite group by libTheme;
```

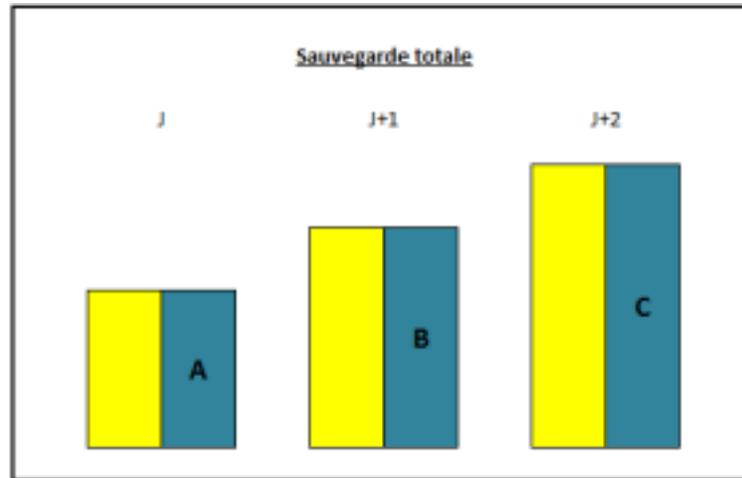


*Schéma entité-association de la base de données fouilles.*

## IV. Rappels sur les différents types de sauvegarde

### A. Sauvegarde totale

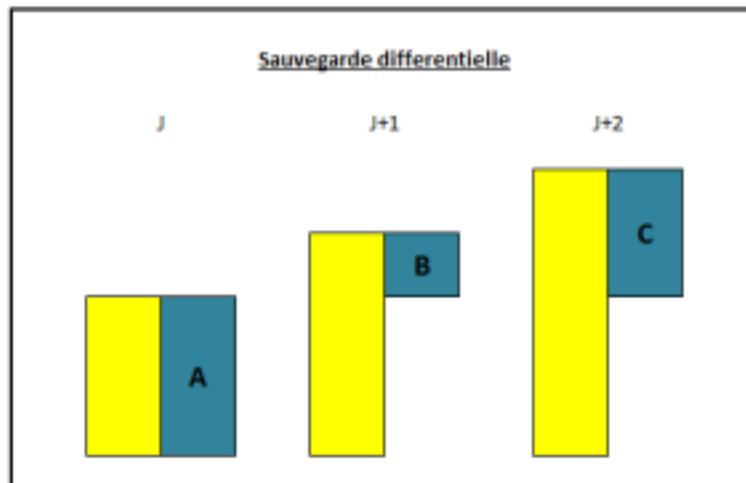
La sauvegarde totale copie l'intégralité de la base de données. Il s'agit de la méthode la plus sûre, mais également de la plus coûteuse (en termes de temps et d'espace).



La restauration d'une base sauvegardée de cette façon se réalise en utilisant la dernière sauvegarde totale (ici C).

### **B. Sauvegarde différentielle**

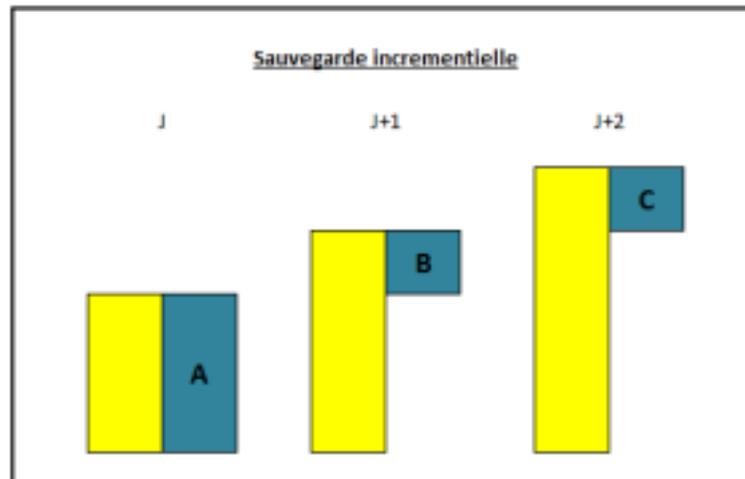
La sauvegarde différentielle se base sur une sauvegarde totale (réalisée le jour J). Chaque sauvegarde différentielle copie la différence entre la sauvegarde totale et l'état de la base actuelle.



La restauration d'une base sauvegardée de cette façon se réalise en utilisant la dernière sauvegarde totale ainsi que la dernière sauvegarde différentielle (ici A + C).

### **C. Sauvegarde incrémentale**

La sauvegarde incrémentale ou incrémentielle se base sur une sauvegarde totale (réalisée le jour J). Chaque sauvegarde incrémentale copie la différence entre la dernière sauvegarde incrémentale et l'état de la base actuelle.



La restauration d'une base sauvegardée de cette façon se réalise en utilisant la dernière sauvegarde totale ainsi que toutes sauvegardes incrémentales réalisées (ici A + B + C).

## **V. Mise en place d'une sauvegarde totale**

Sur le serveur précédemment créé, tapez l'instruction :

```
mysqldump -u root -p fouilles > sauvegardeTotale01.sql
```

Un fichier sauvegardeTotale01.sql a été créé dans le répertoire courant. Ce fichier contient l'intégralité des ordres sql (create et insert) permettant de restaurer la base de données (structure et contenu).

Quel est le résultat retourné par la requête suivante ? Que représente-t-il ?

```
SELECT COUNT(*) FROM dater;
```

A l'aide des requêtes suivantes, supprimez les tables periode et dater de la base de données.

```
DROP TABLE dater;
DROP TABLE periode;
```

Ré-exécutez la requête suivante. Que se passe-t-il ?

```
SELECT COUNT(*) FROM dater;
```

Restaurez la base de données à l'aide de l'instruction

```
mysql -u root -p fouilles < sauvegardeTotale01.sql
```

Vérifiez la présence de la table periode et de son contenu.

## VI. Mise en place d'une sauvegarde incrémentale

### A. Préparation d'une base de données vide

A l'aide des ordres sql suivants, supprimez puis créez à nouveau la base de données fouilles.

```
DROP DATABASE fouilles;  
CREATE DATABASE fouilles;
```

Exécutez le script createFouilles.sql permettant de créer les différentes tables. Pour l'instant, **n'exécutez pas** le script insertFouilles.sql.

```
MariaDB [(none)]> CREATE DATABASE fouilles  
-> ;  
Query OK, 1 row affected (0.000 sec)  
  
MariaDB [(none)]> source /home/lmct65/creationfouilles.sql  
Query OK, 0 rows affected (0.000 sec)  
  
ERROR 1046 (3D000) at line 3 in file: '/home/lmct65/creationfouilles.sql': No database selected  
ERROR 1046 (3D000) at line 8 in file: '/home/lmct65/creationfouilles.sql': No database selected  
ERROR 1046 (3D000) at line 15 in file: '/home/lmct65/creationfouilles.sql': No database selected  
ERROR 1046 (3D000) at line 23 in file: '/home/lmct65/creationfouilles.sql': No database selected  
ERROR 1046 (3D000) at line 27 in file: '/home/lmct65/creationfouilles.sql': No database selected  
ERROR 1046 (3D000) at line 31 in file: '/home/lmct65/creationfouilles.sql': No database selected  
ERROR 1046 (3D000) at line 39 in file: '/home/lmct65/creationfouilles.sql': No database selected  
ERROR 1046 (3D000) at line 46 in file: '/home/lmct65/creationfouilles.sql': No database selected  
MariaDB [(none)]> USE fouilles  
Database changed  
MariaDB [fouilles]> source /home/lmct65/creationfouilles.sql  
Query OK, 0 rows affected (0.000 sec)  
  
Query OK, 0 rows affected, 1 warning (0.020 sec)  
Query OK, 0 rows affected, 1 warning (0.007 sec)  
Query OK, 0 rows affected, 1 warning (0.006 sec)  
Query OK, 0 rows affected (0.005 sec)  
Query OK, 0 rows affected (0.005 sec)  
Query OK, 0 rows affected (0.005 sec)  
Query OK, 0 rows affected, 1 warning (0.007 sec)  
Query OK, 0 rows affected, 1 warning (0.008 sec)  
MariaDB [fouilles]> _
```

La sauvegarde incrémentale peut être mise en place sur MySQL en utilisant les fichiers logs binaires. Ces fichiers contiennent toutes les requêtes qui modifient les données (insert, update et delete).

*L'utilisation du log binaire ralentit le serveur d'environ 1%. Cependant les avantages du log binaire durant les opérations de restauration et pour la réplication sont généralement plus intéressants. (source : dev.mysql.com).*

Chaque arrêt/redémarrage du serveur générera un nouveau fichier de logs binaires qui jouera le rôle de nouvelle sauvegarde incrémentale.

## **B. Activation des logs binaires et création de la première sauvegarde**

Arrêtez le service mysql.

En fonction du serveur installé (MariaDB ou MySQL), éditez son fichier de configuration qui contient ses options de démarrage :

- `mysqld.cnf` contenu dans le répertoire `/etc/mysql/mysql.conf.d/` pour MySQL ;
- `50-server.cnf` contenu dans `/etc/mysql/mariadb.conf.d/` pour MariaBD.

Dans la section `[mysqld]` du fichier, ajoutez la ligne suivante qui permettra d'enregistrer les logs binaires dans le répertoire `/var/log/mysql` :

```
log_bin = /var/log/mysql/logsBinaires.index
```

```
GNU nano 5.4 50-server.cnf *
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# this is read by the standalone daemon and embedded servers
[server]
#
# this is only for the mysqld standalone daemon
[mysqld]
log_bin = /var/log/mysql/logsBinaire.index_
#
# * Basic Settings
#
user                = mysql
pid-file            = /run/mysqld/mysqld.pid
basedir             = /usr
datadir             = /var/lib/mysql
tmpdir              = /tmp
lc-messages-dir    = /usr/share/mysql
lc-messages         = en_US
skip-external-locking
#
# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address        = 127.0.0.1
#
# * Fine Tuning
^G Aide      ^O Écrire   ^W Chercher  ^K Couper   ^T Exécuter  ^C Empl
^X Quitter   ^R Lire fich. ^N Remplacer ^U Coller   ^J Justifier ^_ Aller
```

Démarrez le service mysql.

Exécutez le script insertFouilles.sql permettant d'insérer tous les tuples de la base de données puis redémarrez le service mysql.

### C. Fusion de périodes

Le paléolithique, le mésolithique et le néolithique sont trois périodes précédant la protohistoire. On souhaite désormais qu'elles soient regroupées en une seule et même période : la préhistoire. Les requêtes permettant d'effectuer cette fusion vous sont fournies dans le script fusionPeriodes.sql.

Exécutez le script fusionPeriodes.sql et vérifiez son résultat.

Redémarrez à nouveau le service mysql.



## chemin vers les logs à restaurer serveur à restaurer

```
root@debian:/etc# mysqlbinlog --no-defaults /var/log/mysql/logsBinaire.000001 | mysql -u root -p
Enter password:
```

Il est possible d'utiliser plusieurs fichiers de log dans la même commande en les mettant les uns à la suite des autres.

Il est possible de voir la liste des logs binaires avec la commande suivante :

```
show binary logs;
```

```
MariaDB [(none)]> show binary logs;
+-----+-----+
| Log_name          | File_size |
+-----+-----+
| logsBinaire.000001 |    597417 |
| logsBinaire.000002 |      344  |
+-----+-----+
2 rows in set (0.000 sec)
```

Il est possible de purger les logs binaires en utilisant la commande suivante :

```
reset master;
```

(ATTENTION cela purge les logs binaires, c'est-à-dire qu'ils seront définitivement perdus).

```
MariaDB [(none)]> reset master;
Query OK, 0 rows affected (0.026 sec)
```

## VII. Automatisation des sauvegardes

Cron<sup>1</sup> est un programme informatique utilisé pour automatiser des tâches. Il existe un fichier appelé crontab (situé dans le répertoire /etc) qui contient l'ensemble des actions à exécuter automatiquement et leur périodicité.

Pour éditer le fichier crontab, saisissez la commande

```
crontab -e
```

puis faites défiler le fichier afin de vous positionner sur la dernière ligne. Une fois sur celle-ci, vous pouvez saisir une nouvelle action à automatiser.

Chaque action à automatiser se décrit sur une ligne et se décompose en deux parties :

- la périodicité
- l'action

Pour aller plus loin : <https://doc.ubuntu-fr.org/cron>

## A. La périodicité

Elle se définit à travers cinq champs (minute - heure - jour du mois - mois - jour de la semaine) valorisés ou positionnés à \* lorsqu'ils n'ont pas de valeur.

Exemples :

Périodicité	Signification
5 8 * * 3	tous les mercredis à 8h05
19 8-16/1 * 1,2 1-5	du lundi au vendredi, aux mois de janvier et février, à la 19ème minute de toutes les heures comprises entre 8h et 16h

## B. L'action

Il s'agit de la commande à exécuter.

Automatisez la sauvegarde totale automatique.

```
# m h dom mon dow command
56 10 * * * sh /home/lmct65/scriptBdd.sh
```

```
GNU nano 5.4 save-bdd.sh
backupDate = (date + %Y%m%d-%H%M%S)

dbUser = "root"
dbPass = "P@ssw0rd"
dbName = "fouilles"

backupDir = "/home/lmct65"
backupFile = "$backupDir/$dbName-$bakcupDate.sql"

mysqldump -u $dbUser -p$dbPass $dbName > $backupFile
```