Cours de Programmation Déclarative et Bases de Données

SGBDR et MySQL

Nicolas Jouandeau

n@up8.edu

2022

base de données = ensemble de données

- SGBD = Système de Gestion de Bases de Données
- SGBDR = SGBD Relationnelles
  - les données sont liées par des relations (`1:1`, `1:n`, `n:n`)
  - la relation `1:1`
    - 1 chose est associée à 1 truc
    - exemple : 1 porte avec 1 poignée
  - la relation `1:n`
    - 1 chose est associée à n trucs
    - exemple : 1 maison avec n pièces
  - la relation `n:n`
    - 1 chose est associée à n trucs
    - 1 truc est associé à n choses
    - exemple : 1 armoire pour n personnes et 1 personne pour n armoires
- Il existe d'autres types de bases de données (hiérarchiques, objets, distribuées, ...); nous ne verrons içi que les SBGD relationnelles.

généralités sur les SGBD

- une opération est appelée une transaction
- pour garantir l'intégrité des données, un SGBD assure les propriétés ACID
  - atomicité : une transaction est non sécable, complète ou nulle
  - cohérence : une transaction conduit à un état stable et cohérent (tous les évènements sont traités)
  - isolation : une transaction est traitée indépendamment des autres transactions
  - durabilité : le résultat d'une transaction est permanent

SGBDR en pratique

- SGDB = des tables
- SGBDR = des tables et des relations
  - les relations lient les tables en utilisant des clés
  - l'existence de relations entre deux tables n'est pas obligatoire
  - une table = des colonnes et des enregistrements
  - un enregistrement est unique
  - un enregistrement est identifié par une clé primaire
  - une relation entre deux enregistrements
    - un enregistrement `A` avec une clé primaire `a`
    - un enregistrement `B` avec un champ contenant `a`
    - pour `B`, `a` est une clé étrangère

une clé primaire

- doit être unique (pour identifier de manière unique un enregistrement)
- est composée d'un ou plusieurs champs (i.e. une ou plusieurs colonnes)
- est non nulle

exemple

- pour une personne, on a NOM+PRENOM+TEL
- NOM+PRENOM sont stockés dans une table A
- TEL est stocké dans une autre table B
- NOM+PRENOM sont associés à une clé primaire `clé1` dans A
- TEL est associé à une clé primaire `clé2` dans B
- on veut retrouver le tel d'une personne d'après son nom et pas l'inverse
  - on note `clé2` dans A
  - on ne note pas `clé1` dans B

historique

- le langage SQL apparait dans les années 1975
- *Relational Software, Inc.* (devenu *Oracle Corporation*) propose une première version en 1979
- l'institut américain ANSI le normalise en 1986

SQL

- *Structured Query Language*
- langage déclaratif : description du résultat sans le moyen d'y arriver
- les instructions permettent de contrôler les requêtes
- les requêtes recupèrent des données
- les requêtes sont composées de clauses
  - des mots clés définis par le langage
  - des identificateurs nomment les tables et les colonnes
  - des expressions produisent des scalaires ou des tableaux
  - des prédicats conditionnent les résultats
  - les requêtes se terminent par ";"

conception d'une base de données

- le langage SQL permet
  - de définir les tables et les relations d'une base de données
  - de lire/écrire dans cette base de données
- penser l'organisation de la base de données
  - utiliser l'espace disque en minimisant les redondances
  - assurer la rapidité de l'accès aux données (en accordant les structures aux utilisations prévues)
- principales étapes de conception[a]
  - analyse des besoins
  - organisation des données en tables
  - défintion des relations entre ces tables
  - définition des clés
  - vérification de la cohérence de la base de données (appelée normalisation)

[a] ces étapes dépendent de la méthode et du langage utilisés
pour exemple, MERISE et UML sont une méthode et un langage

requête et jointure

- une opération de lecture/écriture est appelée requête
- une requête intègrant des informations provenant de plusieurs tables est appelée jointure
- les jointures sont très utilisées dans les SGBDR
- une requête intégrant des informations provenant récursivement d'une même table est appelée autojointure
  ce type de requête est utilisé quand un champs renvoie à un autre enregistrement de cette table
- sans condition, une jointure réalise un produit cartésien[a] entre les tables impliquées

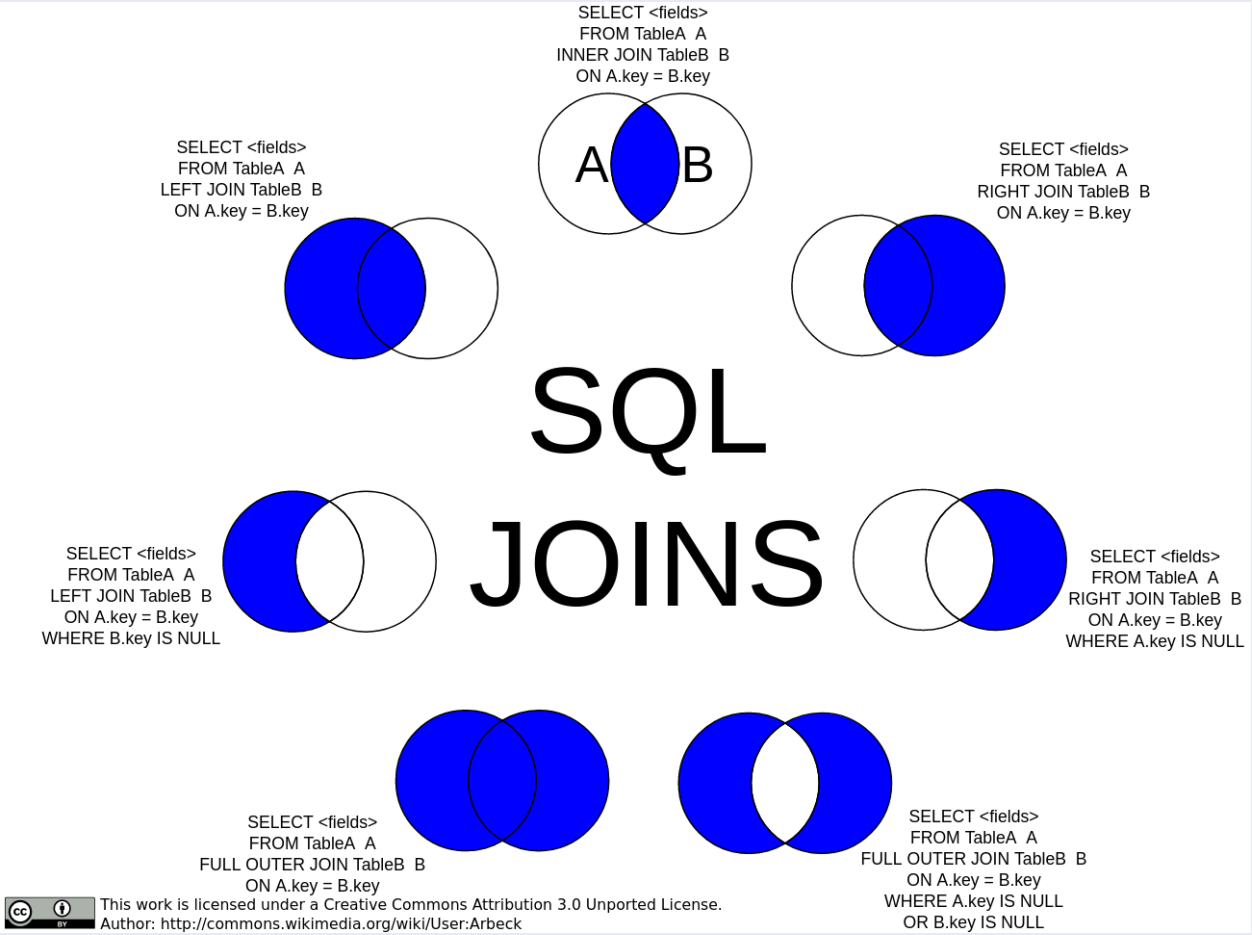
[a] ensemble de tous les couples obtenus par produit des deux ensembles

types de jointure

- pour une table A avec 2 enregistrements `a`, `b`
- pour une table B avec 2 enregistrements `1`, `2`
- la jointure de A et B, retourne 4 enregistrements `a1`, `a2`, `b1`, `b2`
- les types de jointure
  - INNER JOIN : impose que la condition soit vraie dans les tables impliquées
  - CROSS JOIN : demande explicitement le produit cartésien des tables impliquées
  - LEFT JOIN ou LEFT OUTER JOIN : ne vérifie la condition que pour les enregistrements de la première table (i.e. la table spécifiée à gauche en lisant de gauche à droite)
  - RIGHT JOIN ou RIGHT OUTER JOIN : ne vérifie la condition que pour les enregistrements de la deuxième table (i.e. la table spécifiée à droite en lisant de gauche à droite)
  - FULL JOIN ou FULL OUTER JOIN : vérifie la condition dans l'une des tables impliquée
  - SELF JOIN : définie une autojointure dans une même table
  - NATURAL JOIN : définie une jointure automatique sur les colonnes portant le même nom dans plusieurs tables
  - UNION JOIN : réalise de plusieurs requêtes en séquence

types de jointure (suite)

- image prise sur Wikipedia



exemples de jointures pour de deux tables `A` et `B`

- enregistrements communs à `A` et `B`

```
SELECT * FROM A INNER JOIN B ON A.key = B.key
```

- enregistrements de `A` complétés si possible d'infos de `B`

```
SELECT * FROM A LEFT JOIN B ON A.key = B.key
```

- enregistrements de `A` qui ne sont pas dans `B`

```
SELECT * FROM A LEFT JOIN B ON A.key = B.key WHERE B.key IS
NULL
```

- enregistrements dans `A` ou dans `B` sans doublons

```
SELECT * FROM A FULL JOIN B ON A.key = B.key
```

- enregistrements dans `A` sans `B` et dans `B` sans `A`

```
SELECT * FROM A FULL JOIN B ON A.key = B.key WHERE A.key IS
NULL OR B.key IS NULL
```

- mise bout à bout de deux SELECT

```
SELECT * FROM A UNION SELECT * FROM B
```

les principales solutions SQL

- Oracle Database (Oracle)
- MySQL (Oracle)
- Microsoft SQL Server (Microsoft)
- Informix (IBM)
- PostgreSQL (open source)
- SQLite (open source)
- MariaDB (open source)

la solution MySQL

- est une solution classique SQL
- pour les grosses entreprises et/ou les grosses bases de données
- suppose des gros serveurs et un peu d'administration pour les utiliser (gestion de permissions et d'utilisateurs)

solution SGBD relationnelle MySQL

- My pour Prénom de la fille de Michael Widenius (créateur en 1995)
- SQL pour SQL
- MySQL appartient à Sun Microsystems depuis 2008, qui appartient à Oracle depuis 2009
- depuis que MySQL appartient à Sun, Michael Widenius travaille sur MariaDB, un projet open source de SGBD
- `https://www.mysql.com/`

selon la documentation MySQL

- moins de 1000 tables
- moins de 1Go par table
- moins de 20k enregistrements par table
- moins de 2Go pour la taille totale des descriptions des tables (description d'une table = première ligne définissant les types des champs d'un enregistrement)

mise en oeuvre de MySQL

- 1 serveur et 1 client (optionnellement sur le même ordinateur)

MySQL en pratique

- installation sur Ubuntu : `sudo apt-get install mysql-server`
- ou sont les BD : `/var/lib/mysql`
- version

```
$> mysql --version
mysql  Ver 8.0.27-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))
```

- fichiers installés

```
$> dpkg -L mysql-server
/.
/usr
/usr/share
/usr/share/doc
/usr/share/doc/mysql-server
/usr/share/doc/mysql-server/copyright
/usr/share/doc/mysql-server/NEWS.Debian.gz
/usr/share/doc/mysql-server/changelog.Debian.gz
```

fichiers de configuration MySQL

```
$> more /etc/mysql/my.cnf
;;; Il est possible de définir un fichier ~/.my.cnf
;;; Définitions dans les dir /etc/mysql/conf.d/
;;; et /etc/mysql/mysql.conf.d/

$> ls /etc/mysql/conf.d/
mysql.cnf  mysqldump.cnf
$> ls /etc/mysql/mysql.conf.d/
mysql.cnf  mysqld.cnf

$> more /etc/mysql/conf.d/mysql.cnf
[mysql]

$> more /etc/mysql/conf.d/mysqldump.cnf
[mysqldump]
quick
quote-names
max_allowed_packet = 16M
```

- taille max des résultats d'une requête = 16Mo

utiliser MySQL

- démarrer/arrêter le serveur MySQL avec la commande `systemctl`

```
$> systemctl status mysql
o mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; \
             enabled; vendor preset: enabled)
     Active: active (running) since Wed <DATE>; 17min ago
   Main PID: 4113 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 38071)
     Memory: 333.6M
     CGroup: /system.slice/mysql.service
             -- 4113 /usr/sbin/mysqld

<DATE PC> systemd[1]: Starting MySQL Community Server...
<DATE PC> systemd[1]: Started MySQL Community Server.
```

- nous indique
  - le temps depuis le dernier démarrage
  - le pid
  - le log de démarrage

utiliser MySQL (suite)

- démarrer/relancer/arrêter le serveur MySQL

```
$> sudo systemctl start mysql
$> sudo systemctl restart mysql
$> sudo systemctl stop mysql
```

- par défaut dans les services lancés automatiquement au démarrage
- pour le retirer des services lancés au démarrage

```
$> sudo systemctl disable mysql
Synchronizing state of mysql.service with SysV \
  service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable mysql
Removed /etc/systemd/system/multi-user.target.wants/mysql.service.
```

utiliser MySQL (suite)

- consulter les log

```
$> cat /var/log/mysql/error.log | more
$> tail -f /var/log/mysql/error.log
```

- un arret = deux lignes dans les log

```
<DATE> 0 [System] [MY-013172] [Server] Received SHUTDOWN from user
  <via user signal>.
  Shutting down mysqld (Version: 8.0.22-0ubuntu0.20.04.3).
<DATE> 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete
  (mysqld 8.0.22-0ubuntu0.20.04.3)  (Ubuntu).
```

- un redémarrage = plusieurs lignes dans les log

```
<DATE> 0 [System] [MY-010116] [Server] /usr/sbin/mysqld
  (mysqld 8.0.22-0ubuntu0.20.04.3) starting as process 6130
<DATE> 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
<DATE> 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
<DATE> 0 [System] [MY-011323] [Server] X Plugin ready for connections.
  Bind-address: '127.0.0.1' port: 33060, socket: /var/run/mysqld/mysqlx.sock
<DATE> 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
<DATE> 0 [System] [MY-013602] [Server] Channel mysql_main configured
  to support TLS.
  Encrypted connections are now supported for this channel.
<DATE> 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections.
  Version: '8.0.22-0ubuntu0.20.04.3'  socket: '/var/run/mysqld/mysqld.sock'
  port: 3306  (Ubuntu).
```

utiliser MySQL (suite)

- vérifier les ports sur lesquels le serveur écoute

```
$> sudo netstat -tlpn | grep mysql
tcp    0   0 127.0.0.1:33060     0.0.0.0:*       LISTEN      6130/mysqld
tcp    0   0 127.0.0.1:3306      0.0.0.0:*       LISTEN      6130/mysqld
```

- l'utilisateur `root-mysql`
  - possède tous les droits sur la base de données
  - doit définir les tables, les utilisateurs et les droits des utilisateurs
- il est impératif d'avoir un mot de passe pour le login root sur la base de données
- le script `mysql_secure_installation` permet
  - de définir ce mot de passe
  - d'initialiser la configuration de la base de données

commandes MySQL

```
mysql
mysqladmin
mysqlanalyze
mysqlbinlog
mysqlcheck
mysql_config_editor
mysqld
mysqld_multi
mysqld_safe
mysqldump
mysqldumpslow
mysqlimport
mysql_migrate_keyring
mysqloptimize
mysqlpump
mysqlrepair
mysqlreport
mysql_secure_installation
mysqlshow
mysqlslap
mysql_ssl_rsa_setup
mysql_tzinfo_to_sql
mysql_upgrade
```

le script `mysql_secure_installation`

```
$> sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No:
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.
```

le script `mysql_secure_installation` (suite)

```
Remove anonymous users? (Press y|Y for Yes, any other key for No) :

 ... skipping.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) :

 ... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) :

 ... skipping.
All done!
```

ajouter un utilisateur MySQL

- commencer par utiliser MySQL en tant que root

```
$> sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.22-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

- ajouter un utillisateur `n` avec le mot de passe `abc&122!`
(sans accès distant, limitation au nom d'ordi 'localhost')

```
mysql> CREATE USER n@'localhost' IDENTIFIED BY 'abc&122!';
 Query OK, 0 rows affected (0.02 sec)
```

- retirer l'utillisateur `n`

```
mysql> DROP USER n@'localhost';
 Query OK, 0 rows affected (0.04 sec)
```

gestion des bases MySQL

- visualiser les bases de données

```
mysql> SHOW DATABASES;
 +--------------------+
 | Database           |
 +--------------------+
 | information_schema |
 | mysql              |
 | performance_schema |
 | sys                |
 +--------------------+
 4 rows in set (0.00 sec)
```

- créer une base de données nommée Example

```
mysql> CREATE DATABASE Example;
 Query OK, 1 row affected (0.05 sec)

mysql> SHOW DATABASES;
 +--------------------+
 | Database           |
 +--------------------+
 | Example            |
 | information_schema |
 | mysql              |
 | performance_schema |
 | sys                |
 +--------------------+
 5 rows in set (0.00 sec)
```

gestion des tables dans un base de données MySQL

- se placer dans la base de données nommée `Example`
  (les opérations sont implicitement dans cette base de données)

```
mysql> USE Example;
 Database changed
```

- créer une table `Person` avec 3 champs
  (une valeur entière de 10 digits, 2 string de 10 caractères)

```
mysql> CREATE TABLE Person (Id int(10) NOT NULL, \
  Name varchar(10), Last_name varchar(10));
 Query OK, 0 rows affected, 1 warning (0.20 sec)
```

- ajouter un enregistrement dans la table `Person` (une valeur entière de 10 digits, 2 string de 10 caractères)

```
mysql> INSERT INTO Person (Id, Name, Last_name) VALUES (1,'Nicolas','J');
 Query OK, 1 row affected (0.04 sec)
```

donner l'accès à un utilisateur

- les opérations concernées (CRUD - Create Read Update Delete)
  - création de nouvel enregistrement dans la table
  - lecture des enregistrements de la table
  - MAJ d'un enregistrement de la table
  - suppression d'un enregistrement de la table

```
mysql> GRANT ALL PRIVILEGES ON Person TO n@'localhost';
 Query OK, 0 rows affected (0.04 sec)
```

- sortir de l'interprète mysql pour arrêter d'être root

```
mysql> ^DBye
$>
```

- sans avoir fait tout cela :-)

```
$> mysql
ERROR 1045 (28000): Access denied for user 'n'@'localhost' (using password: NO)
```

utiliser une base de données MySQL

- se connecter en tant que `n`

```
$> mysql -u n -p
 Enter password:
 Welcome to the MySQL monitor.  Commands end with ; or \g.
 Your MySQL connection id is 24
 Server version: 8.0.22-0ubuntu0.20.04.3 (Ubuntu)

 Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

 Oracle is a registered trademark of Oracle Corporation and/or its
 affiliates. Other names may be trademarks of their respective
 owners.

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

- utiliser la base de données `Example`

```
mysql> USE Example;
 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
```

utiliser une base de données MySQL (suite)

- afficher le contenu de la base de données (i.e. les tables)

```
mysql> SHOW TABLES;
 +------------------+
 | Tables_in_Example |
 +------------------+
 | Person           |
 +------------------+
 1 row in set (0.00 sec)
```

- afficher le contenu d'une table

```
mysql> SELECT * FROM Person;
 +----+---------+-----------+
 | Id | name    | last_name |
 +----+---------+-----------+
 |  1 | Nicolas | J         |
 +----+---------+-----------+
 1 row in set (0.00 sec)
```

utiliser une base de données MySQL (suite)

- afficher le contenu d'une table avec des conditions

```
mysql> SELECT * FROM Person WHERE Id=1;
 +----+---------+-----------+
 | Id | name    | last_name |
 +----+---------+-----------+
 |  1 | Nicolas | J         |
 +----+---------+-----------+
 1 row in set (0.00 sec)

mysql> SELECT * FROM Person WHERE Id=0;
 Empty set (0.00 sec)
```

- MAJ des enregistrements

```
mysql> UPDATE Person SET last_name = 'Jo' WHERE Id=1;
 Query OK, 1 row affected (0.04 sec)
 Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Person;
 +----+---------+-----------+
 | Id | name    | last_name |
 +----+---------+-----------+
 |  1 | Nicolas | Jo        |
 +----+---------+-----------+
 1 row in set (0.00 sec)
```

utiliser une base de données MySQL (fin) et syntaxe standard

- supprimer des enregistrements

```
mysql> DELETE from Person WHERE Id=1;
 Query OK, 1 row affected (0.04 sec)

mysql> SELECT * FROM Person;
 Empty set (0.00 sec)
```

- sortir de l'interpréteur MySQL avec ctrl-D ou avec la commande `quit`

```
mysql> quit
Bye
```

- même si relativement standard, la syntaxe des requêtes peut légèrement changer
  pour exemple, des SELECT
  - PostgreSQL: `"SELECT * FROM A where n > $1" 10;`
  - MySQL, ODBC: `"SELECT * FROM A where n > ?" 10;`
  - SQLite: supporte les deux syntaxes ci-dessus

avertissement de sécurité concernant MySQL 8.0

- SSL/TLS version TLSv1 est deprecated pour des raisons de sécurité
- SSL/TLS versions 1.2 et 1.3 conseillées
- possibilité d'interception des données en transit
- services garantis par TLS (Transport Layer Security)
  - confidentialité: protection contre un attaquant qui pourrait lire le contenu du trafic
  - intégrité: protection contre la modification du trafic par un pirate
  - prévention du *replay*: protection contre un attaquant qui rejouerait des requêtes contre le serveur
  - authentification: permet au client de vérifier qu'il est connecté au vrai serveur (verifier l'identité du client implique d'utiliser des certificats client)
- sans importance pour une utilisation locale

avertissement de sécurité concernant MySQL 8.0 (suite)

- visualiser les options SSL/TLS actives

```
mysql> SHOW VARIABLES LIKE '%ssl%';
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_openssl  | DISABLED |
| have_ssl      | DISABLED |
| ssl_ca        |          |
| ssl_capath    |          |
| ssl_cert      |          |
| ssl_cipher    |          |
| ssl_crl       |          |
| ssl_crlpath   |          |
| ssl_key       |          |
+---------------+----------+
9 rows in set (0.01 sec)
```