Exercices d'informatique - Série n° 4

Cours 4INOC01 Série distribuée le 11.1.2017

1. Création d'une base de données.

Pour stocker des données, il est souvent plus commode d'utiliser une base de données à la place d'un fichier brut. Voici quelques avantages d'une base de données:

- elle permet d'accéder aux données plus rapidement qu'avec des fichiers bruts,
- elle possède des outils intégrés permettant de rechercher des données satisfaisant des critères choisis par l'utilisateur,
- elle possède des mécanismes permettant de gérer les accès simultanés aux données,
- elle possède des systèmes de privilèges.

Il existe de nombreuses bases de données. Les exercices qui suivent sont consacrés à l'étude de la base de donnée appelée *MySQL*.

(1) Dans un terminal, taper la commande `mysql -u root -p` puis le mot de passe pour accéder à *MySQL*. La commande `show databases;` permet d'obtenir la liste des bases de données *MySQL*.

(2) Créer une base de données appelée `exercices` en tapant la commande `create database exercices;` Vérifier le résultat avec la commande `show databases;`

(3) Créer un utilisateur *eleves* avec la commande `grant insert, select on exercices.* to eleves@localhost identified by 'secret';` Il pourra ainsi accéder librement à la base de données `exercices` avec le mot de passe *secret* pour la consulter (commande `select`) ou pour ajouter des données (commande `insert`).

(4) Taper la commande `exit` pour quitter *MySQL*. Taper ensuite la commande `mysql -u eleves -p` puis le mot de passe *secret*.

(5) Sélectionner la base de données `exercices` avec la commande `use exercices;` La commande `show tables;` permet d'obtenir la liste des tables de la base de données `exercices`. Quelle réponse obtient-on et pourquoi ?

Il est possible d'exécuter toutes les commandes de cet exercice d'un coup en tapant la commande `source fichier.sql` où `fichier.sql` est le fichier (avec le chemin complet) suivant:

```
1 show databases;
2 create database exercices;
3 grant insert, select on exercices.* to eleves@localhost identified by 'secret';
4 use exercices;
5 show tables;
```

2. Création d'une table.

(1) Se connecter à *MySQL* en mode *root*. Dans la base de données `exercices` créer une table nommée `auteurs` en exécutant les commandes suivantes:

```
1 create table auteurs
2 (
3     auteurID int unsigned not null auto_increment primary key,
4     jour timestamp not null,
5     nom varchar(100) character set utf8
6 );
```

Une table est un tableau auquel on ajoute, on modifie ou on enlève une ligne. Chaque ligne du code entre les parenthèses commence par le nom de la colonne et continue par le type de données contenues dans la colonne. Que donne la commande show tables; ?

(2) Que donne la commande **describe** auteurs; Interpréter chaque donnée (la commande help peut être utile ainsi que le site internet `http://dev.mysql.com/doc/`).

3. Effacer et modifier la structure d'une table.

Dans un terminal, exécuter la commande mysql –u root –p puis taper le mot de passe.

(1) Déterminer l'effet de la commande **drop** database exercices;
(2) Déterminer l'effet de la commande **drop table** auteurs;
(3) Déterminer l'effet de la commande **alter table** auteurs **add** prenom **varchar**(50); (utiliser la commande **describe** auteurs;)

4. Ajouter des lignes à une table.

Dans un terminal, taper la commande mysql –u eleves –p puis le mot de passe *secret*.

(1) Déterminer l'effet de la commande **insert into** auteurs (nom)**values** ("Boulgakov");
(2) Déterminer l'effet de la commande **select** * **from** auteurs;
(3) Déterminer l'effet de la commande **update** auteurs **set** nom="Mikhail Boulgakov"**where** auteurID=1;
(4) Exécuter les commandes **insert into** auteurs (nom)**values** ("Albert Cohen"); et **insert into** auteurs (nom) **values** ("Albert Einstein"); puis déterminer l'effet de la commande **select** auteurID **from** auteurs **where** nom **like**("Alb%");

5. Clés étrangères.

Dans un terminal, taper la commande mysql –u root –p puis le mot de passe.

(1) Ajouter une table à la base exercices en exécutant les commandes:

```
1  create table livres
2  (
3      livreID int unsigned not null auto_increment primary key,
4      jour timestamp not null,
5      titre varchar(100) character set utf8,
6      auteur int unsigned not null
7  );
```

Que se passe-t-il lorsqu'on exécute la commande **insert into** livres (titre,auteur) **values** ("Belle du Seigneur",4); ?

(2) Effacer la ligne en exécutant la commande **delete from** livres **where** livreID=1; puis exécuter la commande **alter table** livres **add foreign key** (auteur)references auteurs(auteurID); et exécuter à nouveau la commande **insert into** livres (titre,auteur) **values** ("Belle du Seigneur",4);. Que se passe-t-il ? Interpréter le message renvoyé par *MySQL* et déterminer le rôle de la commande **alter table** livres **add foreign key** (auteur)references auteurs(auteurID);

6. La mémoire du robot.

Créer dans la base de données exercices une table appelée reponses contenant deux colonnes appelées respectivement question et reponse pouvant contenir chacune des chaînes de maximum 100 caractères encodés en UTF-8. Donner des droits **select** et **insert** sur cette table à l'utilisateur eleves identifié avec le mot de passe `secret`.

7. Accès à une base de données MySQL.

Après avoir créé une base de données mysql appelée exercices contenant une table auteurs avec des droits pour l'utilisateur eleves avec le mot de passe secret (voir les exercices concernant MySQL), déterminer l'action de chaque ligne du script *python* donné ci-dessous.
NB Pour exécuter le script, il faut éventuellement installer le module _mysql en exécutant la commande linux `sudo apt-get install python-mysqldb`.

```
1  import _mysql
2  try:
3      db =_mysql.connect('localhost','eleves','secret','exercices')#Modifier eleves ou secret ou
           exercices
4      db.query("select * from auteurs;")
5      r=db.use_result()
6      sortie=r.fetch_row(maxrows=0,how=0)#remplacer maxrows=0 par maxrows=2 et how=0 par how=1 ou
           how=2
7      print sortie
8      db.query("select count(*) from auteurs;")
9      r=db.use_result()
10     resultat=r.fetch_row()
11     print resultat
12     n=resultat[0][0]
13     print type(n)
14     print int(n)
15     db.close()
16 except _mysql.Error, e:
17     print "Erreur: " +str(e[0])+" message SQL: "+e[1]
```

8. Accès à une base de données MySQL.

Déterminer l'effet de chaque ligne du code suivant:

```
1  import _mysql
2  try:
3      db =_mysql.connect('localhost','eleves','secret','exercices')
4      db.query("insert into auteurs (nom) value ('Milan Kundera');")
5      db.close()
6  except _mysql.Error, e:
7      print "Erreur: " +str(e[0])+" message SQL: "+e[1]
```

9. Création d'une base.

En mode root exécuter les commandes MySQL

```
1  grant drop on exercices.* to eleves@localhost identified by 'secret';
2  grant delete on exercices.* to eleves@localhost identified by 'secret';
```

pour ajouter des droits à l'utilisateur eleves.

Créer un script *Python* qui commence par vider les tables livres et auteurs avec les commandes MySQL truncate table livres; pour la table livres et la commande delete from auteurs where 1>0;alter table auteurs auto_increment=1; pour la table auteurs. Ensuite, le script ajoute 2028 auteurs à la table auteurs; leurs noms sont toutes les combinaisons de trois lettres commençant par A, B ou C. Après exécution du script, voici quelques lignes de la requête select * from auteurs;:

```
mysql> select * from auteurs where auteurID<10;
+----------+---------------------+------+
| auteurID | jour                | nom  |
+----------+---------------------+------+
|        1 | 2014-02-03 17:21:04 | Aaa  |
|        2 | 2014-02-03 17:21:04 | Aab  |
|        3 | 2014-02-03 17:21:04 | Aac  |
|        4 | 2014-02-03 17:21:04 | Aad  |
|        5 | 2014-02-03 17:21:04 | Aae  |
|        6 | 2014-02-03 17:21:04 | Aaf  |
|        7 | 2014-02-03 17:21:04 | Aag  |
|        8 | 2014-02-03 17:21:04 | Aah  |
|        9 | 2014-02-03 17:21:04 | Aai  |
+----------+---------------------+------+
9 rows in set (0.00 sec)
mysql> select * from auteurs where auteurID>2010;
+----------+---------------------+------+
| auteurID | jour                | nom  |
+----------+---------------------+------+
|     2011 | 2014-02-03 17:22:03 | Czi  |
|     2012 | 2014-02-03 17:22:03 | Czj  |
|     2013 | 2014-02-03 17:22:03 | Czk  |
|     2014 | 2014-02-03 17:22:03 | Czl  |
|     2015 | 2014-02-03 17:22:03 | Czm  |
|     2016 | 2014-02-03 17:22:03 | Czn  |
|     2017 | 2014-02-03 17:22:03 | Czo  |
|     2018 | 2014-02-03 17:22:03 | Czp  |
|     2019 | 2014-02-03 17:22:03 | Czq  |
|     2020 | 2014-02-03 17:22:03 | Czr  |
|     2021 | 2014-02-03 17:22:03 | Czs  |
|     2022 | 2014-02-03 17:22:04 | Czt  |
|     2023 | 2014-02-03 17:22:04 | Czu  |
|     2024 | 2014-02-03 17:22:04 | Czv  |
|     2025 | 2014-02-03 17:22:04 | Czw  |
|     2026 | 2014-02-03 17:22:04 | Czx  |
|     2027 | 2014-02-03 17:22:04 | Czy  |
|     2028 | 2014-02-03 17:22:04 | Czz  |
+----------+---------------------+------+
18 rows in set (0.00 sec)
```

```
mysql> select * from auteurs where
auteurID>656 and auteurID<686;
+----------+---------------------+------+
| auteurID | jour                | nom  |
+----------+---------------------+------+
|      657 | 2014-02-03 17:21:23 | Azg  |
|      658 | 2014-02-03 17:21:23 | Azh  |
|      659 | 2014-02-03 17:21:23 | Azi  |
|      660 | 2014-02-03 17:21:23 | Azj  |
|      661 | 2014-02-03 17:21:23 | Azk  |
|      662 | 2014-02-03 17:21:23 | Azl  |
|      663 | 2014-02-03 17:21:23 | Azm  |
|      664 | 2014-02-03 17:21:23 | Azn  |
|      665 | 2014-02-03 17:21:23 | Azo  |
|      666 | 2014-02-03 17:21:23 | Azp  |
|      667 | 2014-02-03 17:21:23 | Azq  |
|      668 | 2014-02-03 17:21:23 | Azr  |
|      669 | 2014-02-03 17:21:23 | Azs  |
|      670 | 2014-02-03 17:21:23 | Azt  |
|      671 | 2014-02-03 17:21:23 | Azu  |
|      672 | 2014-02-03 17:21:23 | Azv  |
|      673 | 2014-02-03 17:21:23 | Azw  |
|      674 | 2014-02-03 17:21:23 | Azx  |
|      675 | 2014-02-03 17:21:23 | Azy  |
|      676 | 2014-02-03 17:21:23 | Azz  |
|      677 | 2014-02-03 17:21:23 | Baa  |
|      678 | 2014-02-03 17:21:23 | Bab  |
|      679 | 2014-02-03 17:21:23 | Bac  |
|      680 | 2014-02-03 17:21:23 | Bad  |
|      681 | 2014-02-03 17:21:23 | Bae  |
|      682 | 2014-02-03 17:21:23 | Baf  |
|      683 | 2014-02-03 17:21:24 | Bag  |
|      684 | 2014-02-03 17:21:24 | Bah  |
|      685 | 2014-02-03 17:21:24 | Bai  |
+----------+---------------------+------+
29 rows in set (0.00 sec)
```

10. Python-MySQL.

Rédiger un script qui ajoute 3000 livres à la table livres. Les titres sont composés d'uen lettre X, Y ou Z et d'un nombre entre 1 et 1000. L'auteur du livre est choisi au hasard parmi les auteurs

de la table auteurs. Après exécution du script, voici quelques lignes de la requête **select** * **from** livres;:

```
mysql> select * from livres where livreID<20;
+---------+---------------------+-------+--------+
| livreID | jour                | titre | auteur |
+---------+---------------------+-------+--------+
|       1 | 2014-02-03 17:32:32 | X1    |   1810 |
|       2 | 2014-02-03 17:32:32 | X2    |    973 |
|       3 | 2014-02-03 17:32:32 | X3    |   1174 |
|       4 | 2014-02-03 17:32:32 | X4    |    507 |
|       5 | 2014-02-03 17:32:32 | X5    |    759 |
|       6 | 2014-02-03 17:32:32 | X6    |    331 |
|       7 | 2014-02-03 17:32:32 | X7    |    675 |
|       8 | 2014-02-03 17:32:32 | X8    |     78 |
|       9 | 2014-02-03 17:32:32 | X9    |    447 |
|      10 | 2014-02-03 17:32:32 | X10   |   1753 |
|      11 | 2014-02-03 17:32:32 | X11   |   1877 |
|      12 | 2014-02-03 17:32:32 | X12   |    757 |
|      13 | 2014-02-03 17:32:32 | X13   |   1637 |
|      14 | 2014-02-03 17:32:32 | X14   |    190 |
|      15 | 2014-02-03 17:32:32 | X15   |   1369 |
|      16 | 2014-02-03 17:32:32 | X16   |   1039 |
|      17 | 2014-02-03 17:32:32 | X17   |    936 |
|      18 | 2014-02-03 17:32:32 | X18   |   1513 |
|      19 | 2014-02-03 17:32:32 | X19   |   1741 |
+---------+---------------------+-------+--------+
19 rows in set (0.00 sec)
```

```
mysql> select * from livres where livreID>990
and livreID<1010;
+---------+---------------------+-------+--------+
| livreID | jour                | titre | auteur |
+---------+---------------------+-------+--------+
|     991 | 2014-02-03 17:33:00 | X991  |   1311 |
|     992 | 2014-02-03 17:33:00 | X992  |    401 |
|     993 | 2014-02-03 17:33:00 | X993  |   1262 |
|     994 | 2014-02-03 17:33:00 | X994  |    375 |
|     995 | 2014-02-03 17:33:00 | X995  |   1372 |
|     996 | 2014-02-03 17:33:00 | X996  |    526 |
|     997 | 2014-02-03 17:33:00 | X997  |   1917 |
|     998 | 2014-02-03 17:33:00 | X998  |    249 |
|     999 | 2014-02-03 17:33:01 | X999  |    322 |
|    1000 | 2014-02-03 17:33:01 | X1000 |    408 |
|    1001 | 2014-02-03 17:33:01 | Y1    |   1035 |
|    1002 | 2014-02-03 17:33:01 | Y2    |    997 |
|    1003 | 2014-02-03 17:33:01 | Y3    |   1453 |
|    1004 | 2014-02-03 17:33:01 | Y4    |   1235 |
|    1005 | 2014-02-03 17:33:01 | Y5    |   2014 |
|    1006 | 2014-02-03 17:33:01 | Y6    |   1143 |
|    1007 | 2014-02-03 17:33:01 | Y7    |    643 |
|    1008 | 2014-02-03 17:33:01 | Y8    |    508 |
|    1009 | 2014-02-03 17:33:01 | Y9    |    885 |
+---------+---------------------+-------+--------+
19 rows in set (0.00 sec)
```

```
mysql> select * from livres where livreID>2990;
+---------+---------------------+-------+--------+
| livreID | jour                | titre | auteur |
+---------+---------------------+-------+--------+
|    2991 | 2014-02-03 17:34:01 | Z991  |     38 |
|    2992 | 2014-02-03 17:34:01 | Z992  |    559 |
|    2993 | 2014-02-03 17:34:01 | Z993  |    603 |
|    2994 | 2014-02-03 17:34:01 | Z994  |    809 |
|    2995 | 2014-02-03 17:34:01 | Z995  |   2012 |
|    2996 | 2014-02-03 17:34:01 | Z996  |     87 |
|    2997 | 2014-02-03 17:34:01 | Z997  |     97 |
|    2998 | 2014-02-03 17:34:01 | Z998  |   1448 |
|    2999 | 2014-02-03 17:34:01 | Z999  |    898 |
|    3000 | 2014-02-03 17:34:01 | Z1000 |   1916 |
+---------+---------------------+-------+--------+
10 rows in set (0.00 sec)
```

11. Un robot.

Rédiger deux scripts *Python*. Le premier est un serveur. C'est un robot qui s'appelle Bernard. Plus on lui parle plus il a de la répartie ! Le second est un script qui permet de communiquer à distance avec Bernard ! La mémoire de Bernard est la table reponses. Voici un exemple du fonctionnement de Bernard:

Côté interlocuteur

```
Bonjour, je m'appelle Bernard !
--------------------
Tapez un message: Salut !
Que dois-je repondre ? Bonjour !

Tapez un message:    SALUT !?
Bonjour !
Tapez un message: cOMMENT VAS-TU ?
Que dois-je repondre ? Bien !
Tapez un message: comment
Bien !

Tapez un message: Comment vas-tu Paulette ?
Que dois-je repondre ? Je m'appelle Bernard !
Tapez un message: CommENT
Bien !

Tapez un message: comment
Je m'appelle Bernard !
Tapez un message: comment
Bien !

Tapez un message: stop
```

Table reponses

```
select * from reponses;
+----------+-----------+
| question | reponse   |
+----------+-----------+
| salut    | Bonjour ! |
+----------+-----------+
1 row in set (0.00 sec)

select * from reponses;
+-----------------+-----------+
| question        | reponse   |
+-----------------+-----------+
| salut           | Bonjour ! |
| comment vas-tu  | Bien !    |
+-----------------+-----------+
2 rows in set (0.00 sec)

select * from reponses;
+--------------------------+------------------------+
| question                 | reponse                |
+--------------------------+------------------------+
| salut                    | Bonjour !              |
| comment vas-tu           | Bien !                 |
| comment vas-tu paulette  | Je m'appelle Bernard ! |
+--------------------------+------------------------+
3 rows in set (0.00 sec)
```

Côté Bernard

```
Nouveau client: Thread-1 127.0.0.1 59496

Thread-1 est sorti !
```

Bernard enregistre les réponses suggérées après avoir enlevé les espaces inutiles, la ponctuation et transformé les lettres en minuscules. Avant de répondre, il commence par tester si la phrase est dans la colonne question de sa mémoire. Si oui, il donne la réponse de la colonne reponse. Si non, il teste si une question de sa mémoire "ressemble" à la phrase. Si oui, il choisit au hasard une réponse parmi les réponses correspondantes qui se trouvent dans la table. Si non, il demande à l'interlocuteur de proposer une réponse. Bernard peut avoir des humeurs !