

# TP2 Base de Données

GOURGAUD Kévin / PUP Hugo

L2 Groupe 2

05/11/2019

## Partie I

- a) SOURCE permet d'exécuter un fichier script tandis que mysqldump permet de sauvegarder une base de donnée existante dans un script .sql.
- b)

```
LOCK TABLES `athlete` WRITE;
/*!40000 ALTER TABLE `athlete` DISABLE KEYS */;
load data local infile '~/Bureau/L2/BDD/athlete.txt'
into table athlete
fields terminated by ','
lines terminated by '\r\n' starting by 'AT';
/*!40000 ALTER TABLE `athlete` ENABLE KEYS */;
UNLOCK TABLES;
```

avec cette commande, nous chargeons le fichier texte athlete.txt dans la table athlete. Pour insérer de manière correcte et logique, nous renseignons les délimitations de chaque données. Dans le fichier fourni, les champs sont séparés par ',' et les tuples sont séparés par '\r\n' (retour à la ligne en langage machine). Pour correspondre au type des différentes colonnes, nous devons dire que les tuples commencent par 'AT' car un entier est demandé pour le numéro d'athlète.

Nous obtenons ceci :

numathlete	nom	numnation
6	"John Smith"	3
7	"Moussad Marah"	1
8	"David Douillet"	2
9	"Thierry Lhermite"	2
10	"Alan Lewis"	3
20	"Laura Manaudou"	2
30	"Marie Pierce"	2
40	"Ali Zanou"	5
50	"Lea Cassoli"	4

- c) mysqlimport permet de déplacer une base de donnée SQL d'un serveur à un autre

## Partie II

- a) Premier exemple :

Nous avons créer une vue "athlete\_vue" qui donne les athlètes ayant pour numéro de

nation 2 ou 4.

```
DROP VIEW IF EXISTS `athlete_vue`;  
CREATE VIEW athlete_vue AS SELECT * FROM athlete WHERE numnation=4 OR numnation=2;  
SELECT * FROM athlete_vue;
```

Résultat obtenu ci-dessous :

numathlete	nom	numnation
8	"David Douillet"	2
9	"Thierry Lhermite"	2
20	"Laura Manaudou"	2
30	"Marie Pierce"	2
50	"Lea Cassoli"	4

Nous avons voulu enlever les athlètes de la nation numéro 4 dans la vue précédemment créée avec la commande DELETE.

```
DELETE FROM athlete_vue WHERE numnation=4;  
SELECT * FROM athlete_vue;
```

Nous avons obtenu le résultat suivant :

numathlete	nom	numnation
8	"David Douillet"	2
9	"Thierry Lhermite"	2
20	"Laura Manaudou"	2
30	"Marie Pierce"	2

### Second exemple :

Nous avons créé une vue "epreuve\_vue" avec les épreuves féminines.

```
DROP VIEW IF EXISTS `epreuve_vue`;  
CREATE VIEW epreuve_vue AS SELECT * FROM epreuve WHERE genre='F';  
SELECT * FROM epreuve_vue;
```

Voilà le résultat :

numepreuve	libelle	genre	typeclass	numsport
2	+52kg	F	E	1

b) SET AUTOCOMMIT permet d'activer ou de désactiver les paramètres du fichier.

```
SET AUTOCOMMIT = 1;
```

Dans cet exemple, les paramètres de notre fichier sont activés (on met à 0 pour désactiver les paramètres).

On écrit une TRANSACTION qui permet de mettre à jour la table "jeux" en mettant un COMMIT à la fin pour que la transaction passe à travers le set autocommit = 0 (désactivé) vu précédemment.

```
START TRANSACTION;  
UPDATE jeux SET annee=2019 WHERE pays="Japon";  
SELECT * FROM jeux;  
COMMIT;
```

On obtient ceci :

numjeux	pays	ville	type	annee
1	France	Lyon	E	2016
2	Japon	Tokyo	E	2019
3	Australie	Sidney	E	2004
4	Japon	Nagano	H	2019
5	France	Alberville	H	1990

Avec la commande ROLLBACK, la modification n'a pas lieu.

```
START TRANSACTION;  
UPDATE jeux SET annee=2019 WHERE pays="Japon";  
SELECT * FROM jeux;  
ROLLBACK;
```

numjeux	pays	ville	type	annee
1	France	Lyon	E	2016
2	Japon	Tokyo	E	1998
3	Australie	Sidney	E	2004
4	Japon	Nagano	H	1998
5	France	Alberville	H	1990

On voit bien que la transaction n'a pas été effectué.

- c) Ouvrir deux terminaux simultanément ne fonctionne pas. On ne peut pas vérifier si un autre utilisateur a accès aux tables.



- d) Nous avons d'abord chargés les fichiers Big athlete.txt et Big nation.txt et ensuite nous avons créé plusieurs requêtes imbriquées.

```
load data infile '~/Bureau/L2/BDD/Big athlete.txt'
into table athlete
fields terminated by ' '
line terminated by '\r\n'
ignore 1 line;

load data infile '~/Bureau/L2/BDD/Big nation.txt'
into table nation
fields terminated by ' '
line terminated by '\r\n'
ignore 2 lines;

ALTER TABLE nation
ADD INDEX nom;

(SELECT COUNT (nom,medaille)
AS na
FROM resultat.athlete
WHERE resultat.numathlete = athlete.numathlete AND
numjeux IN (SELECT numjeux FROM JEUX WHERE annee = 1964));

SELECT BENCHMARK(1000000,'SELECT COUNT(numnation) AS big_nat FROM nation WHERE continent LIKE 'Europe');

UPDATE nation
SET medaille= 'Europe'
WHERE medaille='Amérique');

SELECT BENCHMARK(1000000,(SELECT COUNT (numathlete)
AS na
FROM resultat,athlete
WHERE resultat.numathlete = athlete.numathlete AND
numjeux IN (SELECT numjeux FROM jeux WHERE annee = 1964)));

SELECT BENCHMARK(1000000,(SELECT COUNT(medaille) FROM resultat));
SELECT BENCHMARK(1000000,(SELECT COUNT(numathlete) FROM athlete));
SELECT BENCHMARK(1000000,(SELECT COUNT(numnation) FROM nation));
```

Malheureusement, un problème de connexion nous a empêché d'aller plus loin et de finir le TP.

```
No connection. Trying to reconnect...
ERROR 1044 (42000): Access denied for user 'ph01194q'@ '%' to database 'n'
ERROR:
Can't connect to the server

No connection. Trying to reconnect...
ERROR 1044 (42000): Access denied for user 'ph01194q'@ '%' to database 'n'
ERROR:
Can't connect to the server

No connection. Trying to reconnect...
ERROR 1044 (42000): Access denied for user 'ph01194q'@ '%' to database 'n'
ERROR:
Can't connect to the server

No connection. Trying to reconnect...
ERROR 1044 (42000): Access denied for user 'ph01194q'@ '%' to database 'n'
ERROR:
Can't connect to the server
```

- e) CHECK TABLE permet de détecter des erreurs dans une table

OPTIMIZE permet de réduire l'espace de stockage nécessaire de la table en la réorganisant de façon plus claire