

# Administration d'un base de donnée

## Création de Base de données

CREATE DATABASE où comment créer une nouvelle base de données dans MySQL Server.

Pour créer une nouvelle base de données dans MySQL, utilisez l'instruction CREATE DATABASE avec la syntaxe suivante:

```
<source lang='bash'>
```

```
CREATE DATABASE [ IF NOT EXISTS] nom_base[ CHARACTER SET charset][ COLLATE collation]
```

Tout d'abord, vous spécifiez le nom de la base de données après la clause CREATE DATABASE. Le nom de la base de données doit être unique dans l'instance du serveur MySQL. Si vous essayez de créer une base de données avec un nom qui existe déjà, MySQL génère une erreur.

```
mysql> create database testdb;
Query OK, 1 row affected (0,08 sec)

mysql> create database testdb;
ERROR 1007 (HY000): Can't create database 'testdb'; database exists
mysql>
```

Deuxièmement, pour éviter une erreur en cas de création accidentelle d'une base de données existante, vous pouvez spécifier l'option IF NOT EXISTS. Dans ce cas, MySQL ne génère pas d'erreur, mais termine l'instruction CREATE DATABASE.

Troisièmement, vous pouvez spécifier le jeu de caractères et le classement de la nouvelle base de données au moment de la création. Si vous omettez les clauses CHARACTER SET et COLLATE, MySQL utilise le jeu de caractères et le classement par défaut de la nouvelle base de données.

```
mysql> create database testUtf8 CHARACTER SET utf8 COLLATE utf8_unicode_ci;
Query OK, 1 row affected, 2 warnings (0,05 sec)
```

Le jeu de caractères et la collation de la base de données affectent les aspects suivants du fonctionnement du serveur:

- Pour les instructions CREATE TABLE, le jeu de caractères de la base de données et le classement sont utilisés comme valeurs par défaut pour les définitions de table si le jeu de caractères de la table et le classement ne sont pas spécifiés.
- Pour les instructions LOAD DATA qui n'incluent pas de clause CHARACTER SET, le serveur utilise le jeu de caractères indiqué par la variable système character\_set\_database pour interpréter les informations contenues dans le fichier.
- Pour les routines stockées (procédures et fonctions), le jeu de caractères de la base de données et le classement en vigueur au moment de la création de la routine sont utilisés comme jeu de caractères et le classement des paramètres de données de caractères pour lesquels la déclaration ne comprend aucun CHARACTER SET ni aucun attribut COLLATE. Pour remplacer cela, indiquez explicitement CHARACTER SET et COLLATE.

```
mysql> SHOW CHARACTER SET ;
+-----+-----+-----+-----+
| Charset | Description | Default collation | Maxlen |
+-----+-----+-----+-----+
| armSCII8 | ARMSCII-8 Armenian | armSCII8_general_ci | 1 |
| ascii | US ASCII | ascii_general_ci | 1 |
| big5 | Big5 Traditional Chinese | big5_chinese_ci | 2 |
| binary | Binary pseudo charset | binary | 1 |
| cp1250 | Windows Central European | cp1250_general_ci | 1 |
| cp1251 | Windows Cyrillic | cp1251_general_ci | 1 |
| cp1256 | Windows Arabic | cp1256_general_ci | 1 |
| cp1257 | Windows Baltic | cp1257_general_ci | 1 |
| cp850 | DOS West European | cp850_general_ci | 1 |
| cp852 | DOS Central European | cp852_general_ci | 1 |
| cp866 | DOS Russian | cp866_general_ci | 1 |
| cp932 | SJIS for Windows Japanese | cp932_japanese_ci | 2 |
| dec8 | DEC West European | dec8_swedish_ci | 1 |
| eucjpms | UJIS for Windows Japanese | eucjpms_japanese_ci | 3 |
| euckr | EUC-KR Korean | euckr_korean_ci | 2 |
| gb18030 | China National Standard GB18030 | gb18030_chinese_ci | 4 |
| gb2312 | GB2312 Simplified Chinese | gb2312_chinese_ci | 2 |
| gbk | GBK Simplified Chinese | gbk_chinese_ci | 2 |
| geostd8 | GEOSTD8 Georgian | geostd8_general_ci | 1 |
| greek | ISO 8859-7 Greek | greek_general_ci | 1 |
| hebrew | ISO 8859-8 Hebrew | hebrew_general_ci | 1 |
```

```

| hp8 | HP West European | hp8_english_ci | 1 |
| keybcs2 | DOS Kamenicky Czech-Slovak | keybcs2_general_ci | 1 |
| koi8r | KOI8-R Relcom Russian | koi8r_general_ci | 1 |
| koi8u | KOI8-U Ukrainian | koi8u_general_ci | 1 |
| latin1 | cp1252 West European | latin1_swedish_ci | 1 |
| latin2 | ISO 8859-2 Central European | latin2_general_ci | 1 |
| latin5 | ISO 8859-9 Turkish | latin5_turkish_ci | 1 |
| latin7 | ISO 8859-13 Baltic | latin7_general_ci | 1 |
| macce | Mac Central European | macce_general_ci | 1 |
| macroman | Mac West European | macroman_general_ci | 1 |
| sjis | Shift-JIS Japanese | sjis_japanese_ci | 2 |
| swe7 | 7bit Swedish | swe7_swedish_ci | 1 |
| tis620 | TIS620 Thai | tis620_thai_ci | 1 |
| ucs2 | UCS-2 Unicode | ucs2_general_ci | 2 |
| ujis | EUC-JP Japanese | ujis_japanese_ci | 3 |
| utf16 | UTF-16 Unicode | utf16_general_ci | 4 |
| utf16le | UTF-16LE Unicode | utf16le_general_ci | 4 |
| utf32 | UTF-32 Unicode | utf32_general_ci | 4 |
| utf8 | UTF-8 Unicode | utf8_general_ci | 3 |
| utf8mb4 | UTF-8 Unicode | utf8mb4_0900_ai_ci | 4 |
+-----+-----+-----+-----+
41 rows in set (0,00 sec)

```

Il est possible de savoir pour une base de donnée le jeux de données et la collation utilisé:

```

mysql> SELECT @@character_set_database, @@collation_database;
+-----+-----+
| @@character_set_database | @@collation_database |
+-----+-----+
| utf8mb4 | utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0,00 sec)

```

```

show databases;
+-----+
| Database |
+-----+
| information_schema |

```

```
| mysql |
| performance_schema |
| sys |
| testUtf8 |
| testdb |
+-----+
6 rows in set (0,00 sec)
```

Enfin, pour accéder à la base de données nouvellement créée, vous utilisez la commande USE database comme suit:

```
mysql> USE testdb;
```

# Utilisation d'une base de données

Si USE n'a jamais été utilisé, la requete suivante renverras NULL

```
select database();
+-----+
| database() |
+-----+
| NULL |
+-----+
1 row in set (0,00 sec)
```

Après l'utilisation de USE

```
mysql> use testdb;
Database changed
mysql> select database();
+-----+
| database() |
+-----+
```

```
| testdb |
+-----+
1 row in set (0,00 sec)
```

# Storage Engine

```
show engines;
+-----+-----+-----+-----+-----+-----+
| Engine | Support | Comment | Transactions | XA | Savepoints |
+-----+-----+-----+-----+-----+-----+
| FEDERATED | NO | Federated MySQL storage engine | NULL | NULL | NULL |
| InnoDB | DEFAULT | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |
| PERFORMANCE_SCHEMA | YES | Performance Schema | NO | NO | NO |
| MyISAM | YES | MyISAM storage engine | NO | NO | NO |
| MRG_MYISAM | YES | Collection of identical MyISAM tables | NO | NO | NO |
| BLACKHOLE | YES | /dev/null storage engine (anything you write to it disappears) | NO | NO | NO |
| MEMORY | YES | Hash based, stored in memory, useful for temporary tables | NO | NO | NO |
| CSV | YES | CSV storage engine | NO | NO | NO |
| ARCHIVE | YES | Archive storage engine | NO | NO | NO |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0,00 sec)
```

Lors de la création de table, il est possible d'utiliser un des storage engine (ici CSV)

```
CREATE TABLE t2 (i INT NOT NULL) ENGINE = CSV;
Query OK, 0 rows affected (0,08 sec)
```

```
mysql> insert into t2 values(2);  
Query OK, 1 row affected (0,03 sec)
```

Dans le datadir de mysql, on va trouver un repertoire correspondant à la base de données ainsi que le fichier de metadata et de données de la table en CSV:

```
pilou@lubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/data/testdb$ ls  
t2_336.sdi t2.CSM t2.CSV  
pilou@lubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/data/testdb$ more t2.CSV  
2
```

En plus de stocker les métadonnées relatives aux objets de base de données dans le dictionnaire de données, MySQL 8 les stocke sous forme sérialisée. Ces données sont appelées informations de dictionnaire sérialisées (SDI). InnoDB stocke les données SDI dans ses fichiers de tablespace. D'autres moteurs de stockage stockent les données SDI dans des fichiers .sdi créés dans le répertoire du schéma. Les données SDI sont générées dans un format JSON compact.

Il est possible de positionner une table vers une autre storage engine:

```
alter table t2 engine INNODB;  
Query OK, 1 row affected (0,17 sec)  
Records: 1 Duplicates: 0 Warnings: 0
```

En général, les storage engines se doivent de vérifier la transformation des tables:

```
mysql> create table t3 (i INT) ;  
Query OK, 0 rows affected (0,10 sec)  
  
mysql> alter table t3 engine CSV;  
ERROR 1178 (42000): The storage engine for the table doesn't support nullable columns
```

## Renommer une table

Il est possible de renommer une table via la commande RENAME : RENAME ancien\_nom TO nouveau\_nom

```
rename table t3 to t4;  
Query OK, 0 rows affected (0,14 sec)
```

Mais le renommage doit être fait en faisant attention au dépendance

```
create table idtable ( id INT auto_increment, PRIMARY KEY (id));  
Query OK, 0 rows affected (0,11 sec)  
  
mysql> insert into idtable values(0);  
Query OK, 1 row affected (0,09 sec)  
  
mysql> create view testview as select * from idtable;  
Query OK, 0 rows affected (0,09 sec)  
  
mysql> rename table idtable to myidtables;  
Query OK, 0 rows affected (0,11 sec)  
  
mysql> select * from testview;  
ERROR 1356 (HY000): View 'testdb.testview' references invalid table(s) or column(s) or  
function(s) or definer/invoker of view lack rights to use them
```

---

Revision #1

Created 24 November 2019 13:43:04 by Admin

Updated 26 November 2019 19:23:21 by Admin