

Travaux pratiques : moteurs de stockage et tables système MySQL

Temps estimé nécessaire : 25 minutes

Dans ce laboratoire, vous utiliserez l'interface de ligne de commande MySQL (CLI) pour exécuter diverses fonctions liées à la sélection et à la compréhension de certains des moteurs de stockage alternatifs disponibles dans MySQL. Vous continuerez ensuite à explorer les tables système de MySQL qui contiennent des métadonnées sur les objets du serveur.

Objectifs

Après avoir terminé ce laboratoire, vous serez en mesure d'utiliser l'interface de ligne de commande MySQL pour :

- Créez des tables à l'aide de moteurs de stockage alternatifs.
- Interrogez les tables système MySQL pour récupérer des métadonnées sur les objets de la base de données.

Logiciel et base de données utilisés dans ce laboratoire

Logiciel utilisé dans ce laboratoire

Dans ce laboratoire, vous utiliserez [MySQL](#). MySQL est un système de gestion de base de données relationnelle (SGBDR) conçu pour stocker, manipuler et récupérer efficacement des données.



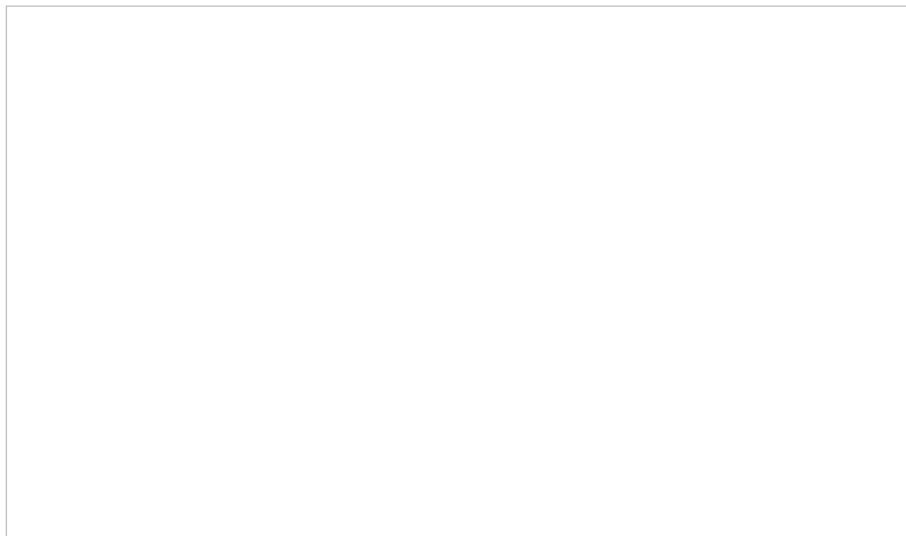
Pour réaliser ce laboratoire, vous utiliserez le service de base de données relationnelle MySQL disponible dans le cadre de l'environnement de développement intégré IBM Skills Network Labs (SN Labs) Cloud. SN Labs est un environnement de laboratoire virtuel utilisé dans ce cours.

Base de données utilisée dans ce laboratoire

La base de données mondiale utilisée dans ce laboratoire provient de la source suivante : <https://dev.mysql.com/doc/world-setup/en/> sous [licence CC BY 4.0](#) avec [Copyright 2021 - Statistics Finland](#).

Vous utiliserez une version modifiée de la base de données pour le laboratoire. Pour suivre avec succès les instructions du laboratoire, veuillez utiliser la base de données fournie avec le laboratoire, plutôt que la base de données de la source d'origine.

Le diagramme ERD suivant montre le schéma de la base de données mondiale :



La première ligne est le nom de la table, la deuxième est la clé primaire et les éléments restants sont des attributs supplémentaires.

Exercice 1 : Créez votre base de données

1. Accédez à **Skills Network Toolbox** en cliquant sur l'icône ci-dessous à partir de l'IDE Cloud lancé côte à côte.
2. Dans le menu déroulant **Bases de données**, cliquez sur **MySQL** pour ouvrir l'onglet Session de service MySQL.
3. Cliquez sur le bouton **Créer** et attendez que la session de service MySQL soit lancée.

Le serveur MySQL va mettre quelques instants à démarrer. Une fois prêt, vous verrez l'étiquette verte « Actif » en haut de la fenêtre.

Faites défiler vers le bas et copiez le mot de passe.

- **REMARQUE** : chaque fois que vous devez saisir votre mot de passe de session de service MySQL à partir de l'onglet Session de service MySQL à n'importe quelle étape du laboratoire, copiez le mot de passe en cliquant sur le petit bouton Copier à droite du bloc de mot de passe. Collez le mot de passe dans le terminal en utilisant **Ctrl + V** (Mac : **⌘ + V**) et appuyez sur **Entrée** sur le clavier. Pour des raisons de sécurité, vous ne verrez pas le mot de passe tel qu'il est saisi sur le terminal.
4. Cliquez sur le bouton **Nouveau terminal** dans l'onglet Session de service MySQL. Vous devez maintenant récupérer deux fichiers de script MySQL dans le stockage de session utilisateur Cloud IDE. Copiez la commande ci-dessous en cliquant sur le petit bouton Copier en bas à droite du bloc de code. Collez-le ensuite dans le terminal à l'invite de ligne de commande en utilisant **Ctrl + V** (Mac : **⌘ + V**) et **Entrée** sur le clavier. Faites cela pour chacune des commandes ci-dessous une à la fois.

- [script_mysql_monde.sql](#)

```
wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0231EN-SkillsNetwork/datasets/World/world_m
```

5. Lancez une session d'invite de commande MySQL en cliquant sur le bouton **MySQL CLI** dans l'onglet de session de service MySQL.

6. Créez un nouveau **monde** de base de données en utilisant la commande ci-dessous dans le terminal :

```
CREATE DATABASE world;
```

7. Pour utiliser la base de données mondiale nouvellement créée, utilisez la commande ci-dessous dans le terminal :

```
USE world;
```

8. Exécutez le script mysql mondial ([world_mysql.sql](#)) pour terminer le processus de création de la base de données mondiale en utilisant la commande ci-dessous dans le terminal :

```
SOURCE world_mysql_script.sql;
```

9. Pour lister tous les noms de tables de la base de données mondiale, utilisez la commande ci-dessous dans le terminal :

```
SHOW TABLES;
```

Exercice 2 : Gérer les moteurs de stockage MySQL

Dans MySQL, *les moteurs de stockage* sont des composants qui gèrent les opérations SQL pour différents types de tables. Le moteur de stockage par défaut et le plus polyvalent de MySQL est **InnoDB**. Lorsque vous créez une nouvelle table dans MySQL à l'aide de la `CREATE TABLE` commande de l'interface de ligne de commande, une table InnoDB est créée par défaut. C'est le moteur le plus utile et il est recommandé pour la plupart des applications générales, à l'exception de quelques cas spécialisés.

Comme détaillé dans la [documentation MySQL](#), MySQL est construit avec une architecture de moteur de stockage enfichable qui permet aux moteurs de stockage d'être facilement chargés et déchargés depuis un serveur MySQL en cours d'exécution.

1. Pour voir la liste des moteurs de stockage pris en charge sur votre serveur MySQL, entrez la commande suivante dans l'interface de ligne de commande MySQL :

```
SHOW ENGINES;
```

Comme vous pouvez le voir, il y a plusieurs colonnes. La première colonne donne le nom du moteur de stockage et la colonne suivante nous indique si ce moteur est pris en charge sur votre serveur MySQL. Comme vous pouvez le voir, le moteur InnoDB est répertorié comme moteur **PAR DÉFAUT**.

Le moteur de stockage CSV

À partir de la `SHOW ENGINES;` commande que vous avez exécutée précédemment, vous pouvez voir que le moteur **CSV** est pris en charge sur le serveur MySQL en cours d'exécution. Les fichiers CSV, abréviation de *Comma Separated Values*, sont des fichiers texte délimités qui utilisent une virgule pour séparer les valeurs.

Allons-y et essayons de créer une table dans notre base de données à l'aide du moteur de stockage CSV.

2. Pour créer une nouvelle table avec un moteur de stockage autre que la base de données InnoDB par défaut, nous spécifions le moteur de stockage que nous souhaitons utiliser dans la `CREATE TABLE` commande. Créons une nouvelle table appelée « `test_csv` » à l'aide du moteur CSV en saisissant la commande suivante dans la CLI :

```
CREATE TABLE csv_test (i INT NOT NULL, c CHAR(10) NOT NULL) ENGINE = CSV;
```

3. Confirmons que la table a été créée avec succès avec la commande suivante :

```
SHOW TABLES;
```

4. ajoutons quelques exemples de données dans notre tableau. Nous ajouterons trois entrées avec la commande suivante :

```
INSERT INTO csv_test VALUES(1,'data one'),(2,'data two'),(2,'data three');
```

5. Jetez un œil aux nouvelles valeurs que vous avez saisies avec la CLI :

```
SELECT * FROM csv_test;
```

Comme vous pouvez le constater, les moteurs de stockage CSV fonctionnent de la même manière que les moteurs InnoDB par défaut. Toutefois, il existe quelques limitations, notamment la non-prise en charge de l'indexation ou du partitionnement.

Comme vous l'avez vu par vous-même à partir de la sortie de la `SHOW ENGINES;` commande, il existe de nombreux moteurs de stockage différents. Nous vous encourageons à les explorer et à les expérimenter vous-même ! Vous pouvez en savoir plus sur leurs cas d'utilisation spécifiques et leurs limites dans la [documentation MySQL](#).

Exercice 3 : Parcourir les tables système MySQL

Le serveur MySQL contient une base de données appelée `mysql`. Il s'agit de la base de données système qui contient les informations *nécessaires* au fonctionnement du serveur, telles que les métadonnées sur toutes les autres tables de la base de données. Cette base de données est l'un des cas particuliers où le moteur de stockage InnoDB par défaut n'est *pas* utilisé. Au lieu de cela, les tables de la `mysql` base de données utilisent le moteur de stockage MyISAM. En général, nous interrogeons principalement les tables système et les modifions rarement directement.

Les tables de la `mysql` base de données se répartissent en plusieurs catégories, dont certaines incluent :

- Tableaux du système de subventions
- Tables du système d'information sur les objets
- Tables du système de journalisation
- Tableaux du système d'aide côté serveur

Pour votre référence, une liste exhaustive des catégories peut être trouvée dans [la section 5.7](#) de la documentation MySQL.

Tableau des catégories du système de subventions

Examinons de plus près la catégorie **Grant System Table**. Elle contient des informations sur les comptes utilisateurs et les privilèges qui leur sont accordés.

1. Tout d'abord, voyons toutes les bases de données sur le serveur MySQL en entrant la commande suivante dans la CLI :

```
SHOW DATABASES;
```

2. Connectez-vous maintenant aux données **mysql** en saisissant :

```
USE mysql;
```

3. Jetez un œil à toutes les tables de la base de données en saisissant les éléments suivants dans la CLI :

```
SHOW TABLES;
```

4. La table **des utilisateurs** contient les comptes d'utilisateurs, les privilèges globaux et d'autres colonnes sans privilèges. Cette table comporte de nombreuses colonnes et est un peu difficile à consulter. Examinons donc uniquement la première colonne qui répertorie les noms des utilisateurs de la base de données. Entrez les informations suivantes dans la CLI :

```
SELECT User from user;
```

5. Ajoutons un nouvel utilisateur à la base de données et voyons si le changement se reflète dans la table des utilisateurs. Nous approfondirons l'ajout d'utilisateurs à une base de données plus tard dans le cours dans le *laboratoire pratique : Gestion des utilisateurs MySQL, contrôle d'accès et chiffrement*, mais pour l'instant, nous allons simplement exécuter une commande simple dans la CLI qui créera un nouvel utilisateur nommé « test_user » :

```
CREATE USER test_user;
```

6. **Essayez par vous-même** : vérifiez que la table *des utilisateurs* dans la base de données *MySQL* a été automatiquement mise à jour pour refléter l'ajout du nouvel utilisateur que vous avez créé. *Entrez la commande appropriée dans la CLI pour afficher la colonne Utilisateur dans la table des utilisateurs.*

▼ **Astuce** (Cliquez ici)

Jetez un œil à la commande que vous avez entrée à l'étape 4.

▼ **Solution** (Cliquez ici)

```
SELECT User from user;
```

Comme vous pouvez le constater, « test_user » a été ajouté automatiquement à cette table, vous n'avez pas eu besoin d'utiliser de `INSERT` commandes. C'est généralement le cas pour les tables système : vous n'aurez que rarement besoin de les modifier directement. Au lieu de cela, elles sont généralement interrogées.

Interroger les INFORMATION_SCHEMA tables de la base de données

La `INFORMATION_SCHEMA` est une base de données présente dans chaque serveur MySQL. Elle contient des métadonnées sur le serveur MySQL telles que le nom d'une base de données ou d'une table, le type de données d'une colonne ou les privilèges d'accès. Notez que cette base de données contient des tables *en lecture seule*, vous ne pouvez donc pas utiliser directement les commandes `INSERT`, `UPDATE`, ou `DELETE` sur celles-ci. Continuons et connectons-nous à la base de données.

1. **ESSAYEZ-LE VOUS-MÊME** : à l'aide de la CLI MySQL, affichez toutes les bases de données sur le serveur.

▼ **Astuce** (Cliquez ici)

Vous avez déjà fait cela à l'étape 1 de la section précédente concernant la catégorie Tableau du système de subventions. N'hésitez pas à vous y référer pour obtenir des conseils.

▼ **Solution** (Cliquez ici)

```
SHOW DATABASES;
```

Comme vous pouvez le voir, la `information_schemabase` de données est là par défaut.

2. **Essayez-le vous-même** : saisissez la commande appropriée dans la CLI pour vous connecter à la `information_schemabase` de données.

▼ **Astuce** (Cliquez ici)

Rappelez-vous la commande que vous avez entrée dans la section précédente pour vous connecter à la base de données « `mysql` ». Comment peut-elle être modifiée pour se connecter à la base de données « `information_schema` » ?

▼ **Solution** (Cliquez ici)

```
USE information_schema;
```

3. Dans la `information_schemabase` de données, il existe une table appelée `COLUMNS` qui contient des métadonnées sur les colonnes de toutes les tables et vues du serveur. L'une des colonnes de cette table contient les noms de toutes les autres colonnes de chaque table. Examinons les noms des colonnes de la `countrytable` dans la `world` base de données en saisissant la commande suivante dans la CLI :

```
SELECT COLUMN_NAME FROM COLUMNS WHERE TABLE_NAME = 'country';
```

4. Un autre point intéressant de la `information_schemabase` de données est la `TABLE` qui contient les métadonnées sur toutes les tables du serveur. L'une des colonnes de cette table contient des informations sur le type de moteur de stockage d'une table. Pour revenir à notre discussion précédente sur les moteurs de stockage, exécutez la commande suivante dans la CLI pour afficher le type de moteur de stockage pour les tables « `country` », « `city` », « `countrylanguage` » et enfin « `csv_test` » que vous avez créées :

```
SELECT table_name, engine FROM INFORMATION_SCHEMA.TABLES
WHERE table_name = 'country' OR table_name = 'city'
OR table_name = 'countrylanguage' OR table_name = 'csv_test';
```

Comme prévu, les trois premières tables mentionnées utilisent le moteur de stockage InnoDB par défaut, tandis que la table « `csv_test` » utilise le moteur de stockage CSV.

5. Enfin, la `TABLE` de la `information_schemabase` de données contient des informations sur la taille d'une table donnée en octets. Ces informations sont stockées dans deux colonnes : **`data_length`** et **`index_length`** qui stockent respectivement la taille des données de la table et la taille du fichier d'index de cette table. Par conséquent, la taille totale de la table est la somme des valeurs de ces deux colonnes. Cette valeur serait donnée en octets, cependant, si vous souhaitez utiliser une unité plus pratique, la somme peut être convertie en Ko en la divisant par 1024. Vous pouvez trouver la taille des tables (en Ko) que vous avez interrogées à l'étape précédente avec la commande suivante dans la CLI :

```
SELECT table_name, (data_length + index_length)/1024 FROM INFORMATION_SCHEMA.TABLES
WHERE table_name = 'country' OR table_name = 'city'
OR table_name = 'countrylanguage' OR table_name = 'csv_test';
```

Exercice 4 : Essayez-le vous-même !

Dans cet exercice pratique, vous allez mettre en pratique ce que vous avez appris sur les moteurs de stockage dans MySQL pour commencer par créer une nouvelle table à l'aide du moteur de stockage MyISAM non par défaut. Vous appliquerez ensuite ce que vous avez appris sur les tables système MySQL pour récupérer les métadonnées de votre table nouvellement créée.

1. **Essayez-le vous-même** : Tout d'abord, connectez-vous à la `worldbase` de données à l'aide de la CLI :

▼ **Astuce** (Cliquez ici)

La commande pour se connecter à une base de données prend la forme générale suivante :

```
USE <database name>;
```

▼ **Solution** (Cliquez ici)

```
USE world;
```

2. **Essayez-le vous-même** : créez une nouvelle table appelée `MyISAM_test` qui utilise le `MYISAM` moteur de stockage.

▼ **Astuce** (Cliquez ici)

Rappelez-vous que vous avez fait quelque chose de très similaire dans l'exercice 2 lorsque vous avez créé une nouvelle table à l'aide du moteur de stockage `csv`. Que pouvez-vous modifier dans la commande que vous avez utilisée pour accomplir cette tâche ?

Pour vous épargner le défilement, voici la commande que vous avez utilisée dans l'exercice 2 pour créer `csv_test`:

```
CREATE TABLE csv_test (i INT NOT NULL, c CHAR(10) NOT NULL) ENGINE = CSV;
```

▼ **Solution** (Cliquez ici)

```
CREATE TABLE MyISAM_test (i INT NOT NULL, c CHAR(10) NOT NULL) ENGINE = MYISAM;
```

Cela créera une table vide appelée `MyISAM_test` où la première colonne, nommée `i` contiendra éventuellement un entier, et la deuxième colonne nommée `c` contiendra jusqu'à 10 caractères.

3. **Essayez par vous-même** : vous souhaitez ensuite interroger une table dans la `information_schemabase` de données, mais avant cela, vous devrez d'abord vous connecter à la base de données. Utilisez la CLI pour vous connecter à la `information_schemabase` de données.

▼ **Astuce** (Cliquez ici)

Rappelez-vous ce que vous avez fait à l'étape 1 de cet exercice pratique. Cette étape est très similaire.

▼ **Solution** (Cliquez ici)

```
USE information_schema;
```

4. **Essayez-le vous-même** : à l'aide de la CLI, interrogez la `TABLE` dans la `information_schemabase` de données pour afficher les colonnes `table_name` et `engine` de toutes les tables qui contiennent `table_schema = 'world'`. Vérifiez que la table que vous avez créée dans cet exercice existe et qu'elle dispose du moteur de stockage approprié.

▼ **Astuce** (Cliquez ici)

Pour interroger des colonnes spécifiques d'une table en fonction d'une condition, utilisez une commande du type :

```
SELECT <column_1>, <column_2>, ... FROM <table name> WHERE <condition>;
```

Astuce bonus : dans ce cas, le nom de la table doit être `INFORMATION_SCHEMA.TABLES`.

▼ **Solution** (Cliquez ici)

```
SELECT table_name, engine FROM INFORMATION_SCHEMA.TABLES WHERE table_schema = 'world';
```

Comme vous pouvez le voir, la `MyISAM_test` table est là et elle utilise le moteur de stockage `MyISAM` comme prévu.

Conclusion

Félicitations ! Vous avez terminé ce laboratoire pratique et comprenez désormais les bases des moteurs de stockage MySQL et disposez des compétences nécessaires pour créer de nouvelles tables avec des moteurs de stockage alternatifs pour des cas d'utilisation spécialisés. De plus, vous avez acquis une certaine familiarité avec les tables système de MySQL et la manière de les interroger pour récupérer des métadonnées sur d'autres objets de votre base de données.

Auteur

- [David Pasternak](#)

Autres contributeurs

- Sandip Saha Joy

© IBM Corporation 2023. Tous droits réservés.