

# Hands-on Lab : MySQL User Management, Access Control, and Encryption

**Estimated time needed:** 25 minutes

In this lab, first you will learn how to manage MySQL user accounts and roles using phpMyAdmin graphical user interface (GUI) tool. Then you will learn how to control access to MySQL databases and their objects. Finally you will learn how to secure your data adding extra layer of security using data encryption.

## Objectives

After completing this lab, you will be able to use the phpMyAdmin to:

- Manage MySQL user accounts and roles
- Control access to MySQL databases and their objects
- Add last line of defense to secure data using encryption

## Software Used in this Lab

In this lab, you will use [MySQL](#). MySQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve data.



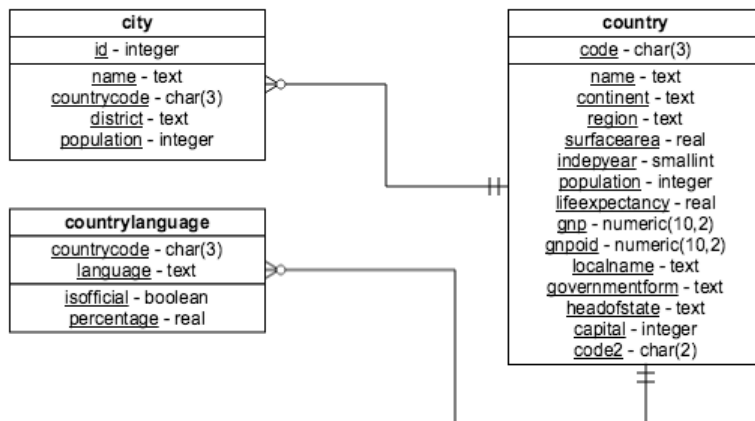
To complete this lab you will utilize the MySQL relational database service available as part of the IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

## Database Used in this Lab

The World database used in this lab comes from the following source: <https://dev.mysql.com/doc/world-setup/en/> under [CC BY 4.0 License](#) with [Copyright 2021 - Statistics Finland](#).

You will use a modified version of the database for the lab, so to follow the lab instructions successfully please use the database provided with the lab, rather than the database from the original source.

The following ERD diagram shows the schema of the World database:



The first row is the table name, the second is the primary key, and the remaining items are any additional attributes.

## Exercise 1: Manage MySQL user accounts and roles

In this example exercise, you will go through an example on how to manage MySQL user accounts and roles using phpMyAdmin.

User management is the process of controlling which users are allowed to connect to the MySQL server and what permissions they have on each database. phpMyAdmin does not handle user management, rather it passes the username and password on to MySQL, which then determines whether a user is permitted to perform a particular action. Within phpMyAdmin, administrators have full control over creating users, viewing and editing privileges for existing users, and removing users.

1. Go to **Skills Network Toolbox** by clicking the following icon from the side by side launched Cloud IDE.
2. From the **Databases** drop-down menu, click **MySQL** to open the MySQL service session tab.
3. Click the **Start** button and wait until MySQL service session gets launched.

The screenshot shows the Skills Network interface for MySQL. On the left sidebar, the 'Databases' icon is highlighted with a red box and a red circle containing the number 1. The main panel shows the 'MySQL INACTIVE' status, with the 'MySQL' text and 'INACTIVE' label highlighted by a red box and a red circle containing the number 2. Below this, the 'Start' button is highlighted by a red box and a red circle containing the number 3. The interface also displays version information (v8.0.22, v5.0.4, v14.14) and a 'Summary' tab.

SKILLS NETWORK... MySQL x

▼ DATABASES

MySQL INACTIVE

PostgreSQL INACTIVE

Cassandra INACTIVE

MongoDB INACTIVE

> BIG DATA

> CLOUD

> OTHER

MySQL INACTIVE

v8.0.22 | v5.0.4 | v14.14

Connect to MySQL and phpMyAdmin directly in your Skills

Start

Summary Connection Information Details

Get started with MySQL in a faster, easier way. To launch y

The MySQL server will take a few moments to start. Once it is ready, you will see the green "Active" label near the top of the window.

SKILLS NETWORK... MySQL x

▼ DATABASES

MySQL **ACTIVE**

PostgreSQL **INACTIVE**

Cassandra **INACTIVE**

MongoDB **INACTIVE**

> BIG DATA

> CLOUD

> OTHER

# MySQL

**ACTIVE**

v8.0.22 | v5.0.4 | v14.14

Connect to MySQL and phpMyAdmin directly in your Skills

**Stop**

Summary Connection Information Details

Your database and phpMyAdmin server are now ready to u  
login credentials. For more details on how to navigate MyS  
section.

**Username:** [Redacted] [Copy]

**Password:** [Redacted] [Copy] ←

You can manage MySQL via:

**phpMyAdmin** [External Link]

Or to interact with the database in the terminal, select one







**MySQL CLI** **New Terminal**

**NOTE:** Whenever you are required to enter your MySQL service session password from the MySQL service session tab at any step of the lab, copy the password by clicking on the small copy button on the right of the password block. Paste the password into the terminal using **Ctrl + V** (Mac: **⌘ + V**), and press **Enter** on the keyboard. For security reasons, you will not see the password as it is entered on the terminal.

3. Click **phpMyAdmin** button from the mysql service session tab. You will see the phpMyAdmin GUI tool.

← → ↻ 🏠 🔒 sandipsahajo-8080.theiadocker-27.proxy.cognitiveclass.ai

# phpMyAdmin



Recent Favorites

New

+ information\_schema

+ mysql

+ performance\_schema

+ sakila

+ sys

← Server: mysql:3306

Databases SQL Status User a

## General settings

☰ Server connection collation: ⓘ utf8mb4\_unicode\_

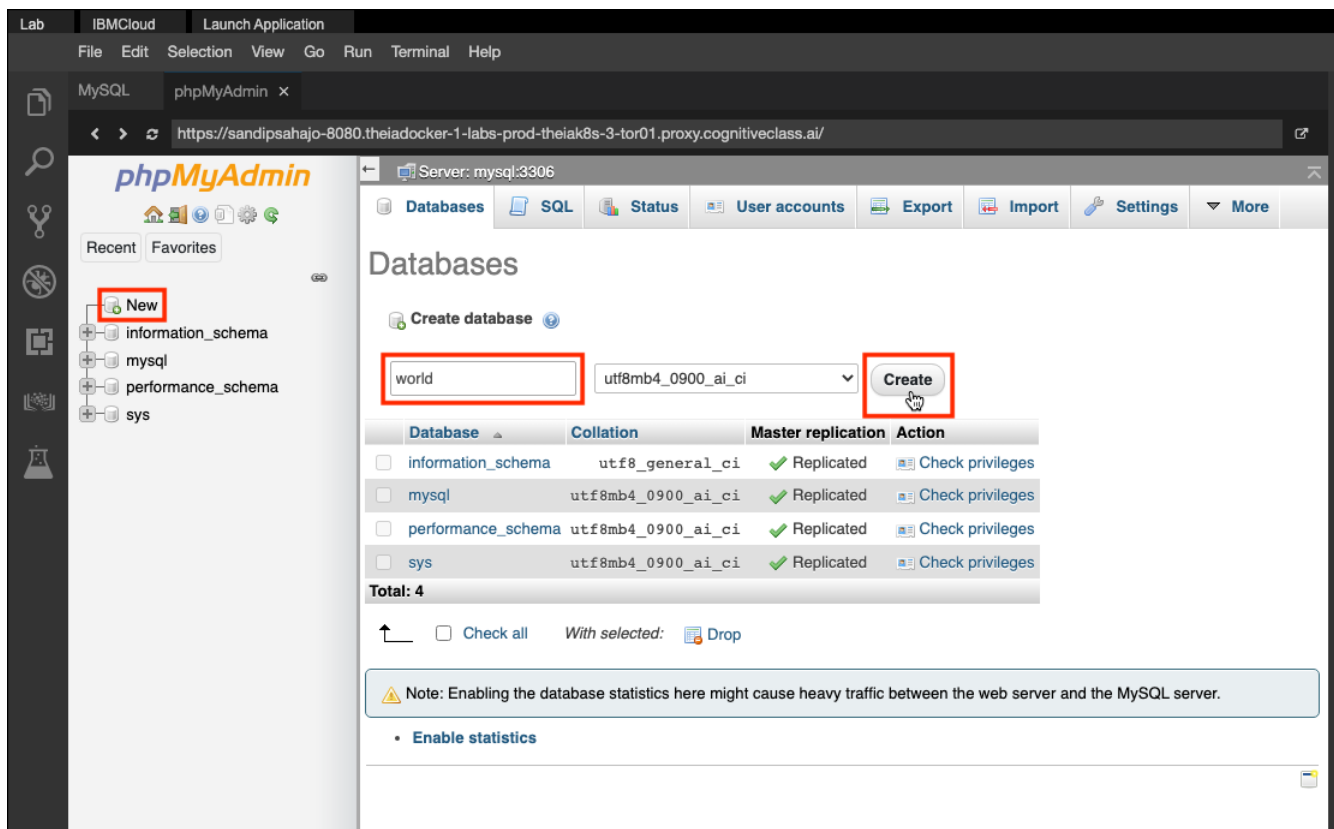
🔑 More settings

## Appearance settings

🗣️ Language ⓘ English

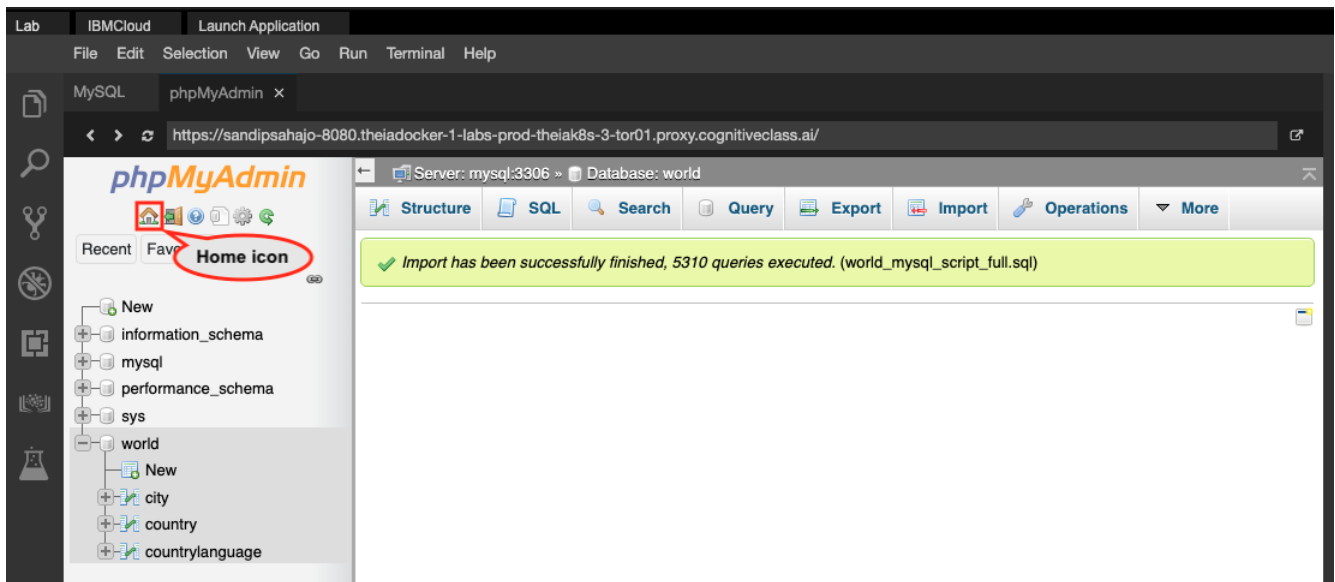
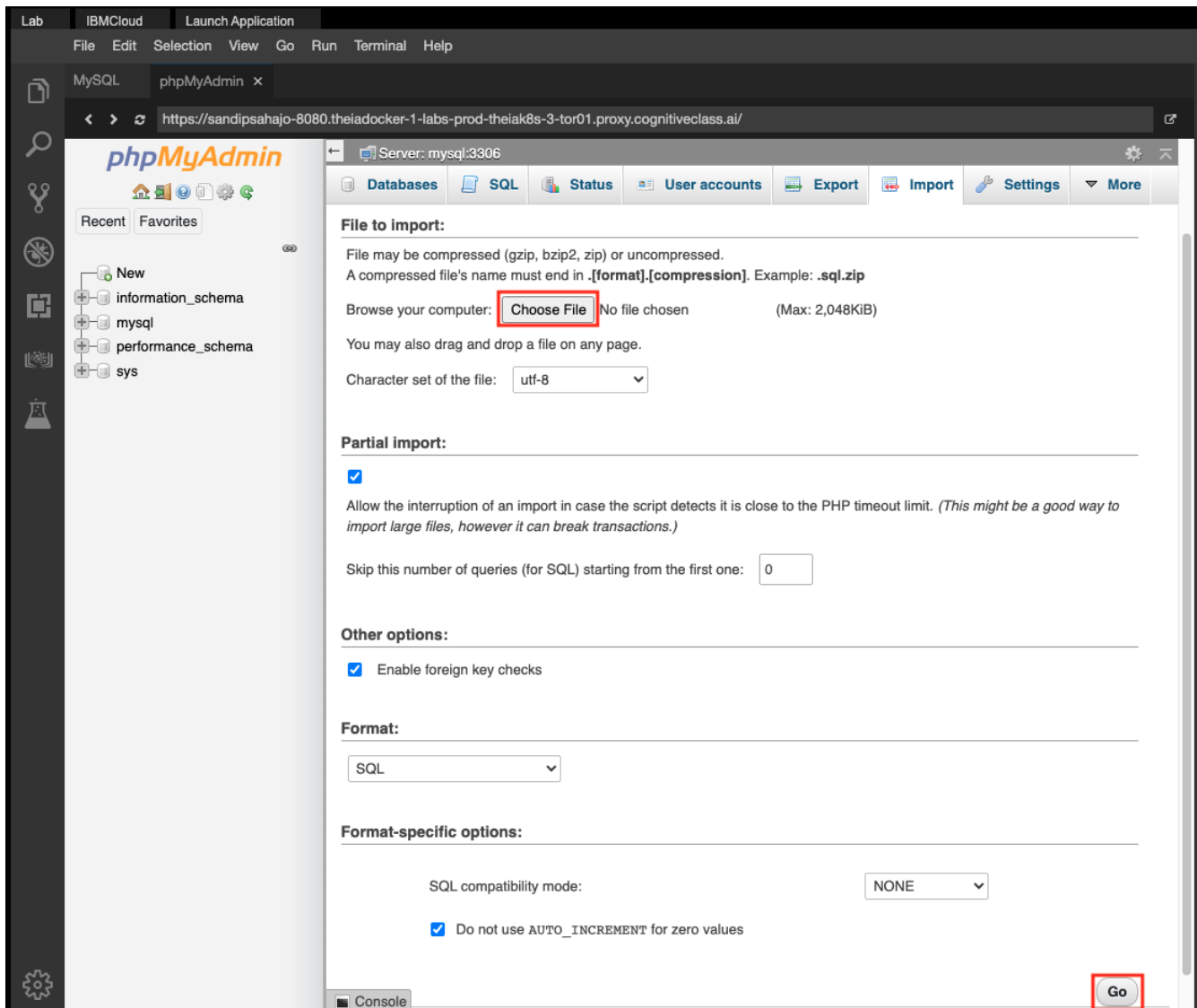
🎨 Theme: pmahomme ▼

4. In the tree-view, click **New** to create a new empty database. Then enter **world** as the name of the database and click **Create**.



5. Go to the **Import** tab. Upload the following sql script file using the **Choose File** button (first download the following sql script file to your local computer storage). Then click **Go** button at the bottom. You will be notified when the import successfully gets finished. Click the **Home** icon.

- [world\\_mysql\\_script\\_full.sql](#)



6. Now you will create a user account with custom role “db\_owner”. Usually a user with db\_owner role has all global privileges and access to all existing databases. Go to the **User accounts** tab and click **Add user account**.

Lab | IBMCloud | Launch Application

File Edit Selection View Go Run Terminal Help

MySQL phpMyAdmin x

https://sandipsahajo-8080.theiadocker-1-labs-prod-theiak8s-3-tor01.proxy.cognitiveclass.ai/

Server: mysql:3306

Databases SQL Status **User accounts** Export Import Settings More

### User accounts overview

User name	Host name	Password	Global privileges	Grant	Action
<input type="checkbox"/> mysql.infoschema	localhost	Yes	SELECT	No	<a href="#">Edit privileges</a> <a href="#">Export</a>
<input type="checkbox"/> mysql.session	localhost	Yes	SHUTDOWN, SUPER	No	<a href="#">Edit privileges</a> <a href="#">Export</a>
<input type="checkbox"/> mysql.sys	localhost	Yes	USAGE	No	<a href="#">Edit privileges</a> <a href="#">Export</a>
<input type="checkbox"/> root	%	Yes	ALL PRIVILEGES	Yes	<a href="#">Edit privileges</a> <a href="#">Export</a>
<input type="checkbox"/> root	localhost	Yes	ALL PRIVILEGES	Yes	<a href="#">Edit privileges</a> <a href="#">Export</a>

↑ ☐ Check all With selected: [Export](#)

**New**

[Add user account](#)

[Remove selected user accounts](#)

(Revoke all active privileges from the users and delete them afterwards.)

☐ Drop the databases that have the same names as the users.

[Go](#)

**Note:** phpMyAdmin gets the users' privileges directly from MySQL's privilege tables. The content of these tables may differ from the privileges the server uses, if they have been changed manually. In this case, you should [reload the privileges](#) before you continue.

7. Fill the **Login Information** as shown in following image (enter your own password). Under **Global privileges**, click **Check all**. Scroll down and click **Go**.

Lab

IBMCLOUDLaunch Application

FileEditSelectionViewGoRunTerminalHelp

MySQLphpMyAdmin x

https://sandipsahajo-8080.theiadocker-1-labs-prod-theiak8s-3-tor01.proxy.cognitiveclass.ai/

phpMyAdmin

RecentFavorites

New

information\_schema

mysql

performance\_schema

sys

world

New

city

country

countrylanguage

Server: mysql:3306

DatabasesSQLStatusUser accountsExportImportSettingsMore

Add user account

Login Information

User name:Use text field:db\_owner

Host name:Locallocalhost

Password:Use text field:Strength:Extremely weak

Re-type:

Authentication PluginCaching sha2 authentication

Generate password:Generate

Database for user account

☐ Create database with same name and grant all privileges.

☐ Grant all privileges on wildcard name (username\\_%).

Global privileges

☐ Check all

☐ Data

☐ SELECT

☐ Structure

☐ CREATE

☐ ALTER

☐ Administration

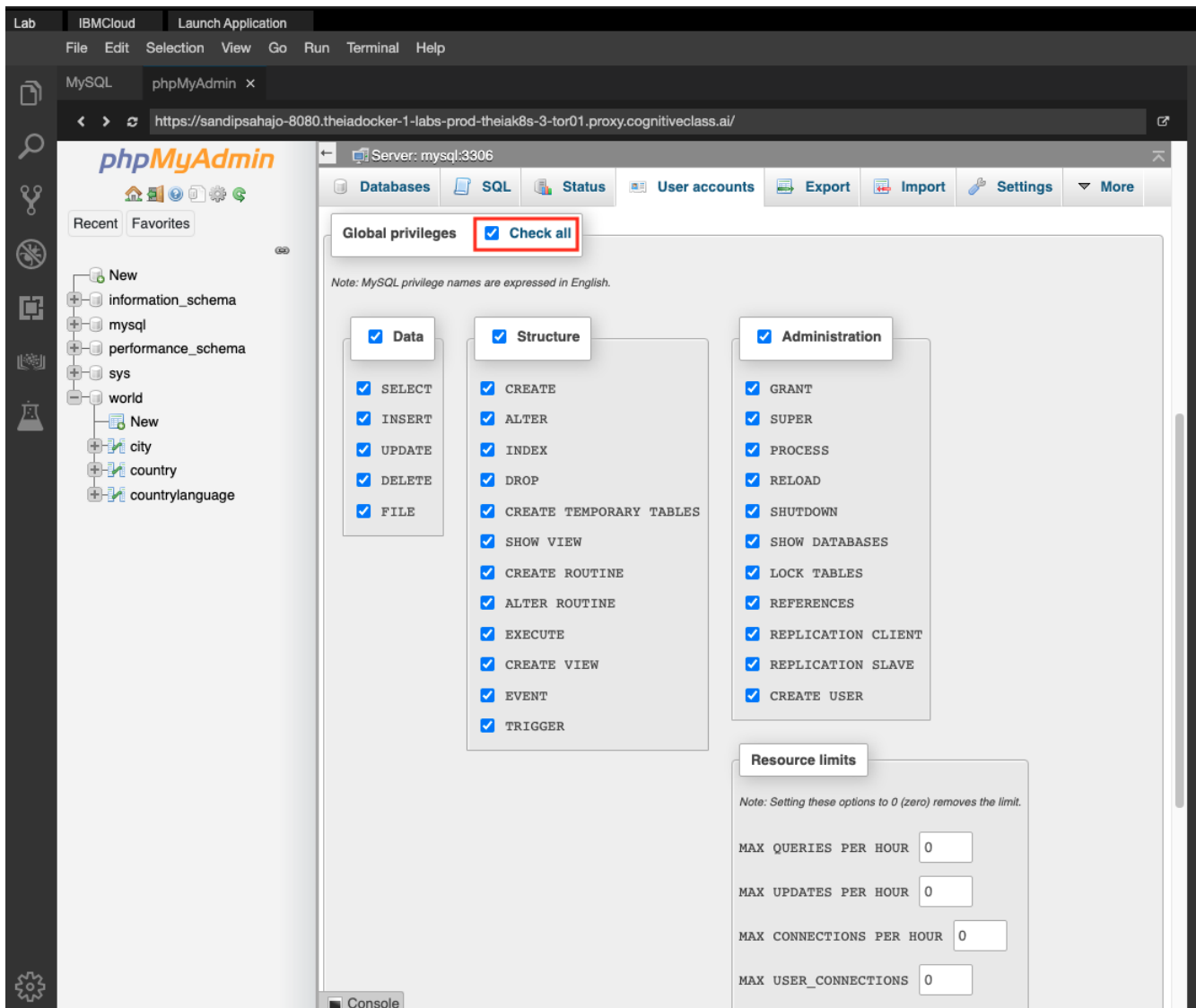
☐ GRANT

☐ SUPER

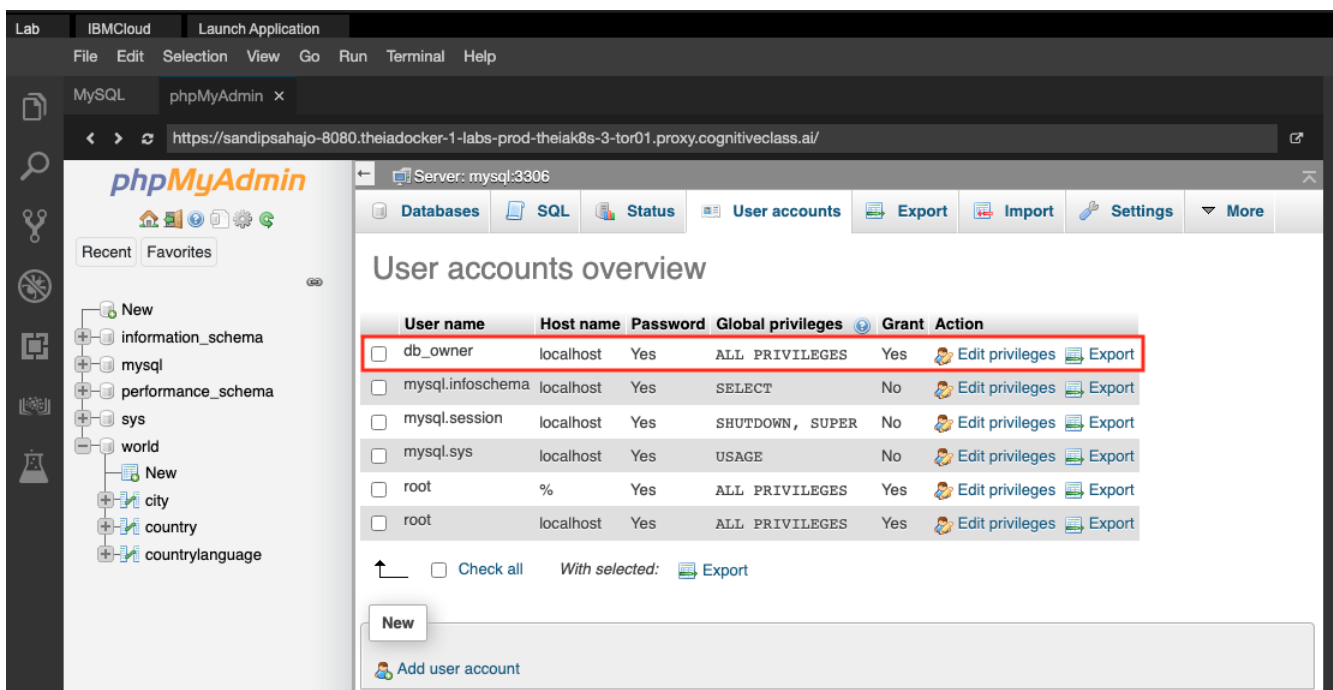
Note: MySQL privilege names are expressed in English.

Console





8. You have successfully created a user account with appropriate privileges.

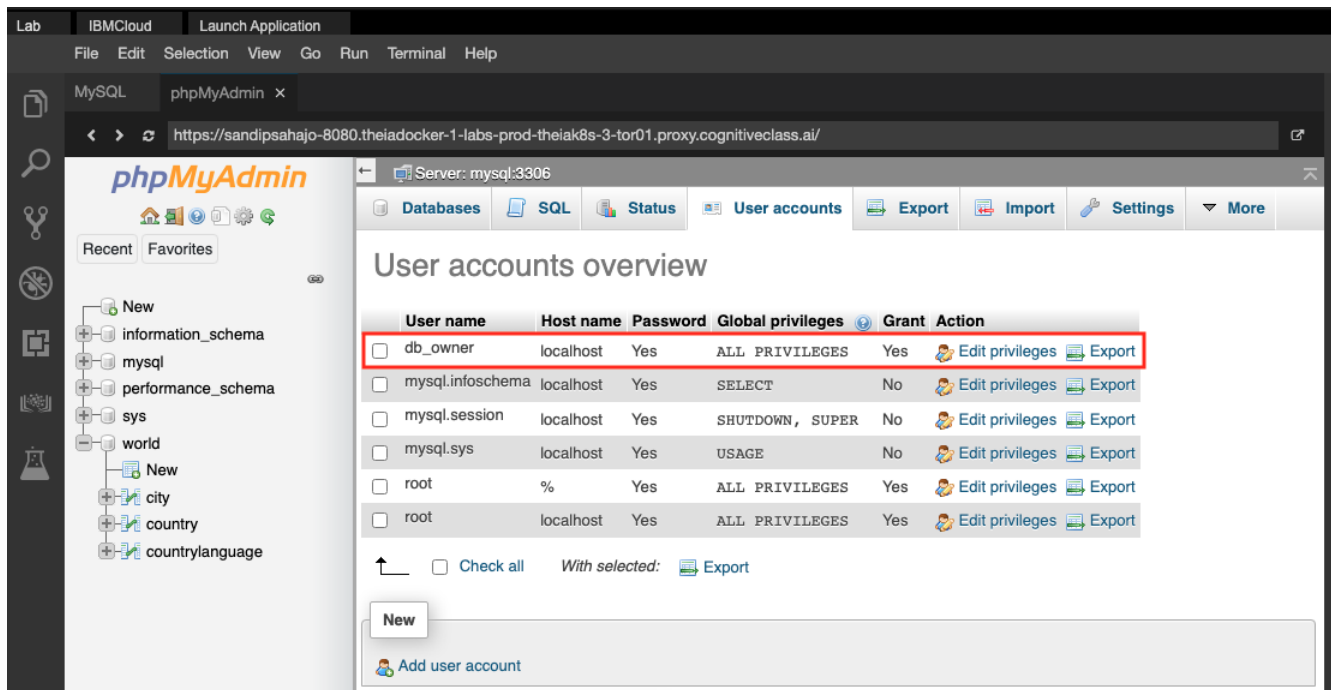


## Exercise 2: Control access to MySQL databases and their objects

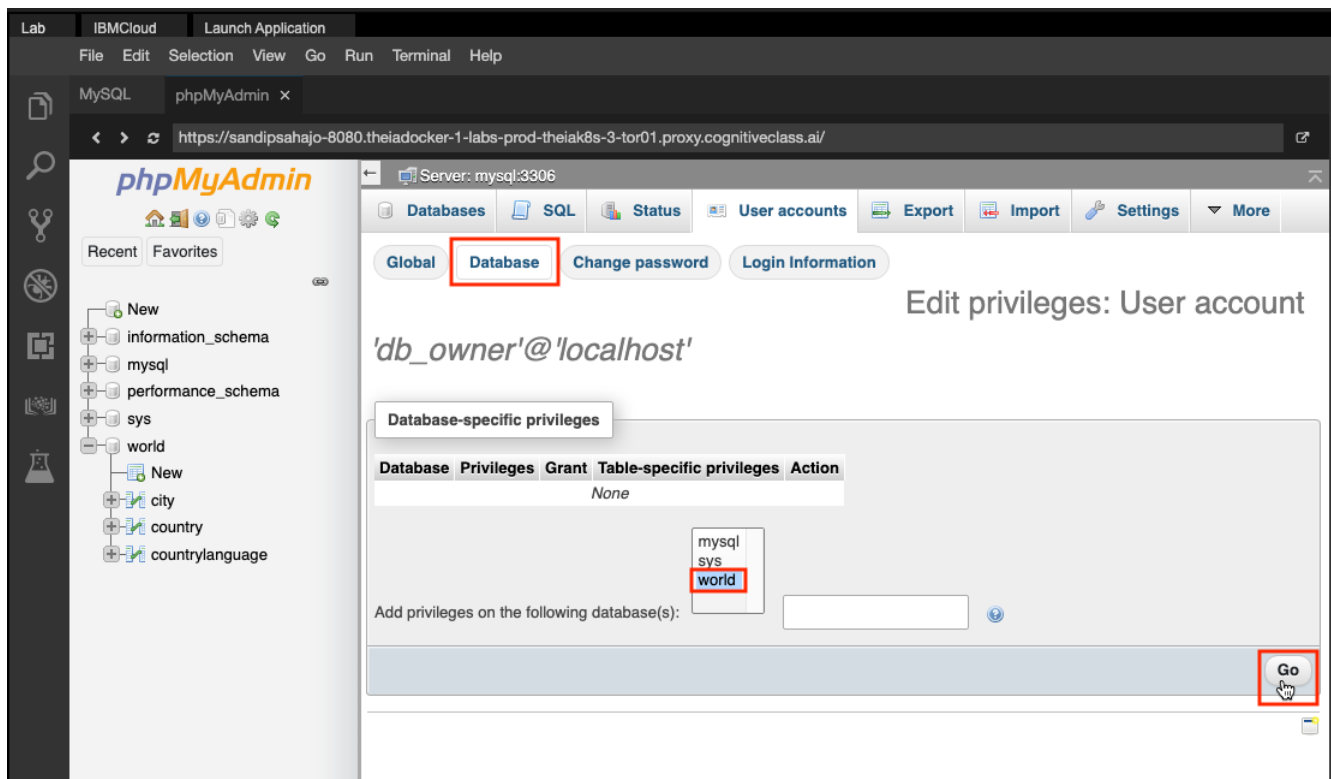
In this example exercise, you will learn how to control access to MySQL databases and their objects.

Making an exception to the user definition of `db_owner` role you created earlier, you will modify privileges of this user so that this user won't be able to update other columns except a specific column of a specific table of a specific database. You will restrict `db_owner` from updating all the other columns except the column **Population** of the table **city** of the database **world**.

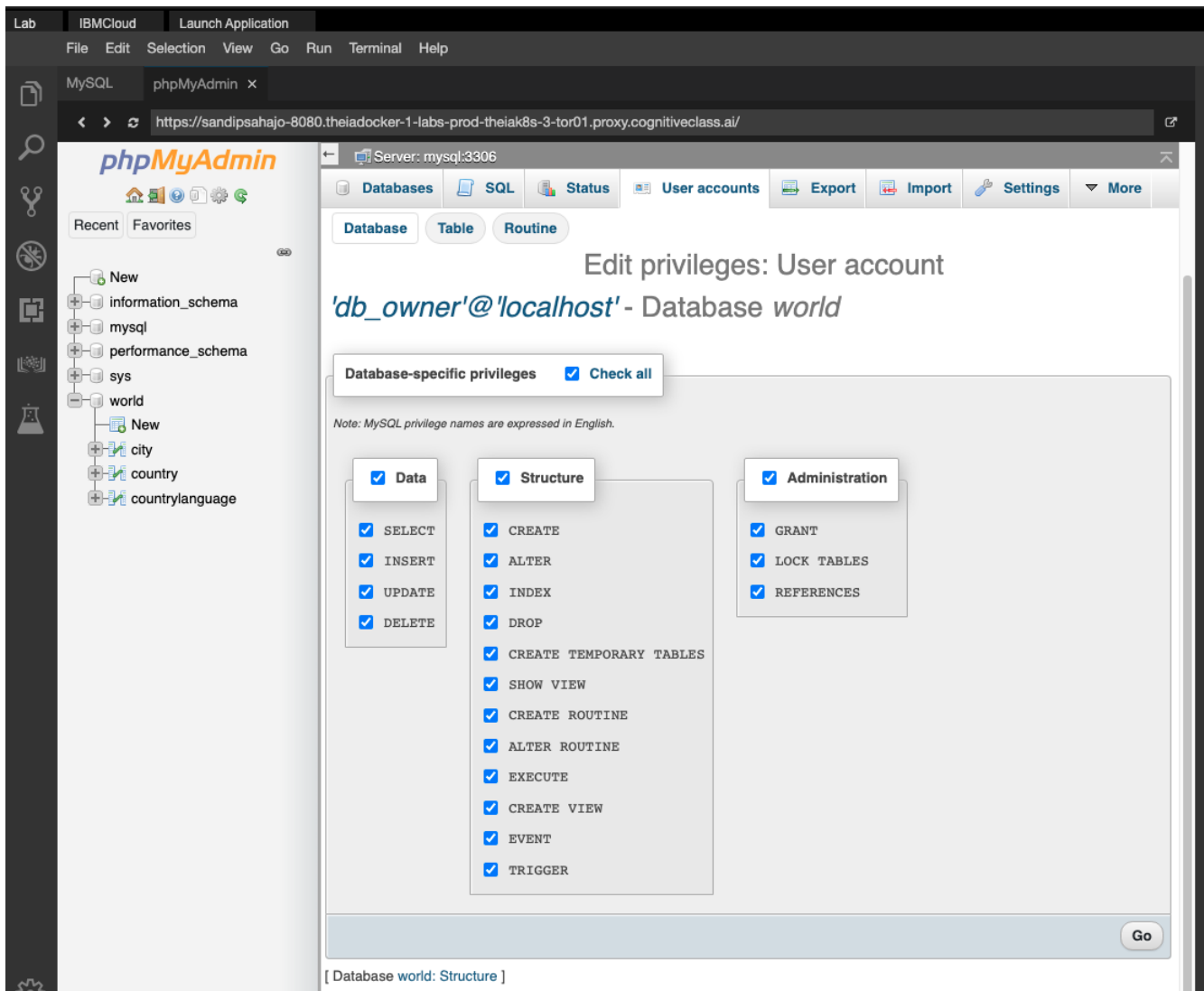
1. Go to **Home > User accounts** tab. Click **Edit privileges** option of `db_owner` user name.



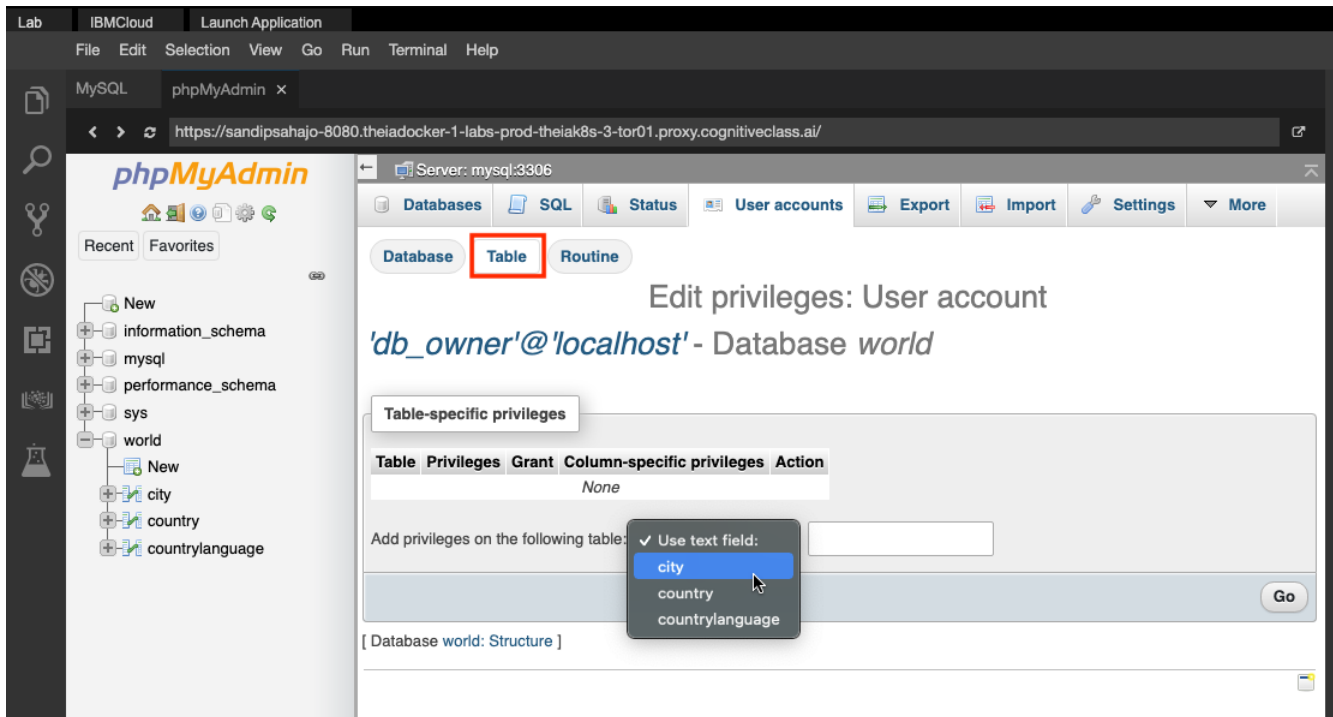
2. Under **Database** sub-tab, select **world** database and click **Go**.



3. Under **Database-specific privileges**, select **Check all** and click **Go** at the bottom.



4. Switch to **Table** sub-tab. Select the table **city** from the drop-down menu and click **Go**.



5. Under **Table-specific privileges**, configure all the SQL commands and their custom access to the columns of the table **city** as shown below. Then click **Go**. Such table-specific privilege configuration will restrict db\_owner from updating all the other columns except the column **Population** of the table **city** of the database **world**.

Table

Edit privileges: User account 'db\_owner'@'localhost' - Database world - Table city

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT

INSERT

UPDATE

REFERENCES

☒ DELETE

ID  
Name  
CountryCode  
District  
Population

ID  
Name  
CountryCode  
District  
Population

ID  
Name  
CountryCode  
District  
Population

ID  
Name  
CountryCode  
District  
Population

☐ None

☐ None

☐ None

☐ None

☒ CREATE

☒ DROP

☒ GRANT

☒ INDEX

☒ ALTER

☒ CREATE\_VIEW

☒ SHOW\_VIEW

☒ TRIGGER

Go

[ Database world: Structure ] [ Table city: Browse ]

### Exercise 3: Secure data using encryption

In this example exercise, you will learn how to secure your data adding extra layer of security using data encryption. There may be certain parts of your database containing sensitive information which *should not* be stored in plain text. This is where encryption comes in.

You will implement encryption and decryption of a column in the world database using the official AES (Advanced Encryption Standard) algorithm. AES is a symmetric encryption where the same key is used to encrypt and decrypt the data. The AES standard permits various key lengths. By default, key length of 128-bits is used. Key lengths of 196 or 256 bits can be used. The key length is a trade off between performance and security. Let's get started.

1. Click the **MySQL CLI** button from the mysql service session tab.

Lab | IBMCloud | Launch Application

File | Edit | Selection | View | Go | Run | Terminal | Help

MySQL x | phpMyAdmin

# MySQL

ACTIVE

v8.0.22 | v5.0.4 | v14.14

Connect to MySQL and phpMyAdmin directly in your Skills Network Labs environment.

Stop

Summary | Connection Information | Details

Your database and phpMyAdmin server are now ready to use and available with the following details on how to navigate MySQL, please check out the Details section.

**Username:** davidpaster2

**Password:** MTg4OTctZGF2aWRw

You can manage MySQL via:

phpMyAdmin

Or to interact with the database in the terminal, select one of these options:

MySQL CLI | New Terminal

2. First, you will need to hash your passphrase (consider your passphrase is **My secret passphrase**) with a specific hash length (consider your hash length is **512**) using a hash function (here you will use hash function from **SHA-2** family). It is good practice to hash the passphrase you use, since storing the passphrase in plaintext is a significant security vulnerability. Use the following command in the terminal to use the SHA2 algorithm to hash your passphrase and assign it to the variable `key_str`:

```
1. 1
1. SET @key_str = SHA2('My secret passphrase', 512);
```

Copied!

3. Now, let's take a look at the `world` database. First, you will want to connect to the database by entering the following command in the CLI:

```
1. 1
1. USE world;
```

Copied!

4. Next, let's take a quick look at the `countrylanguage` table in our database with the following command:

```
1. 1
1. SELECT * FROM countrylanguage LIMIT 5;
```

Copied!

```
theia@theiadocker-davidpaster2: /home/project ×
Database changed
mysql> SELECT * FROM countrylanguage LIMIT 5;
+-----+-----+-----+-----+
| CountryCode | Language | IsOfficial | Percentage |
+-----+-----+-----+-----+
| ABW         | Dutch   | T          | 5.3        |
| ABW         | English | F          | 9.5        |
| ABW         | Papiamentu | F        | 76.7       |
| ABW         | Spanish | F          | 7.4        |
| AFG         | Balochi | F          | 0.9        |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> █
```

For *demonstration purposes*, **suppose** that the last column in the table, labeled *Percentage* contains sensitive data, such as a citizen's passport number. Storing such sensitive data in plain text is an enormous security concern, so let's go ahead and encrypt that column.

5. To encrypt the *Percentage* column, we will first want to convert the data in the column into binary byte strings of length 255 by entering the following command into the CLI:

```
1. 1
1. ALTER TABLE countrylanguage MODIFY COLUMN Percentage varbinary(255);
```

Copied!

6. Now to actually encrypt the *Percentage* column, we use the AES encryption standard and our hashed passphrase to execute the following command:

```
1. 1
1. UPDATE countrylanguage SET Percentage = AES_ENCRYPT(Percentage, @key_str);
```

Copied!

7. Let's go ahead and see if the column was successfully encrypted by taking another look at the *countrylanguage* table. We again run the same command as in step 4:

```
1. 1
1. SELECT * FROM countrylanguage LIMIT 5;
```

Copied!

theia@theiadocker-davidpaster2: /home/project ×

```
mysql> SELECT * FROM countrylanguage LIMIT 5;
```

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	d0L0D00e0!0h~
ABW	English	F	0rG0000000(0R0KK
ABW	Papiamentu	F	100Ã0;00 ðz0m
ABW	Spanish	F	K0000A0 o00
AFG	Balochi	F	00'0hB0zI0Hr

5 rows in set (0.00 sec)

```
mysql> █
```



As you can see, the data on the *Percentage* column is encrypted and completely illegible.

8. The supposedly sensitive data is now encrypted and secured from prying eyes. However, we should still have a way to access the encrypted data when needed. To do this, we use the `AES_DECRYPT` command, and since AES is symmetric, we use the same key for both encryption and decryption. In our case, recall that the key was a passphrase which was hashed and stored in the variable `key_str`. Suppose we need to access the sensitive data in that column. We can do so by entering the following command in the CLI:

```
1. 1
1. SELECT cast(AES_DECRYPT(Percentage, @key_str) as char(255)) FROM countrylanguage;
```

Copied!

theia@theiadocker-davidpaster2: /home/project ×

5 rows in set (0.01 sec)

```
mysql> SELECT cast(AES_DECRYPT(Percentage, @key_str) as char(255)) FROM
```

cast(AES_DECRYPT(Percentage, @key_str) as char(255))
5.3
9.5
76.7
7.4
0.9

5 rows in set (0.00 sec)



## Practice Exercise: Control access to MySQL databases and their objects

In this practice exercise, you will get to put what you learned to use and modify privileges for a user.

Scenario: You will modify privileges of the user `db_owner` you created in example exercise A such a way that this user won't be able to insert, update and delete any column of a specific table `country` of the `world` database.

▼ Hint (Click Here)

Follow example exercise 2. If necessary create the user account following example exercise 1.

▼ Solution (Click Here)

1. Go to **Home** > **User accounts** tab. Click **Edit privileges** option of `db_owner` user name.
2. Under **Database** sub-tab, select `world` database and click **Go**.

- Under **Database-specific privileges**, select **Check all** and click **Go** at the bottom.
- Switch to **Table** sub-tab. Select the table **country** from the drop-down menu and click **Go**.
- Under **Table-specific privileges**, configure all the SQL commands and their custom access to the columns of the table **city** as shown in the following image. Then click **Go**. Such table-specific privilege configuration will restrict db\_owner from inserting, updating and deleting any column of a specific table **country** of the **world** database.

Table

Edit privileges: User account 'db\_owner'@'localhost' - Database world - Table country

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	<input type="checkbox"/> DELETE
<div>Code</div> <div>Name</div> <div>Continent</div> <div>Region</div> <div>SurfaceArea</div> <div>IndepYear</div> <div>Population</div> <div>LifeExpectancy</div>	<div>Code</div> <div>Name</div> <div>Continent</div> <div>Region</div> <div>SurfaceArea</div> <div>IndepYear</div> <div>Population</div> <div>LifeExpectancy</div>	<div>Code</div> <div>Name</div> <div>Continent</div> <div>Region</div> <div>SurfaceArea</div> <div>IndepYear</div> <div>Population</div> <div>LifeExpectancy</div>	<div>Code</div> <div>Name</div> <div>Continent</div> <div>Region</div> <div>SurfaceArea</div> <div>IndepYear</div> <div>Population</div> <div>LifeExpectancy</div>	<div><input checked="" type="checkbox"/> CREATE</div> <div><input checked="" type="checkbox"/> DROP</div> <div><input checked="" type="checkbox"/> GRANT</div> <div><input checked="" type="checkbox"/> INDEX</div> <div><input checked="" type="checkbox"/> ALTER</div> <div><input checked="" type="checkbox"/> CREATE_VIEW</div> <div><input checked="" type="checkbox"/> SHOW_VIEW</div> <div><input checked="" type="checkbox"/> TRIGGER</div>
Or <input type="checkbox"/> None	Or <input checked="" type="checkbox"/> None	Or <input checked="" type="checkbox"/> None	Or <input type="checkbox"/> None	

Go

[ Database world: Structure ] [ Table country: Browse ]

Congratulations! You have completed this lab, and you are ready for the next topic.

Author(s)

- [Sandip Saha Joy](#)

Other Contributor(s)

- [David Pasternak](#)

Changelog

Date	Version	Changed by	Change Description
2021-07-13	1.0	Sandip Saha Joy	Created initial version
2021-10-05	1.1	David Pasternak	Updated instructions
2021-10-13	1.2	Steve Hord	Copy edits
2021-07-13	1.3	Jaskomal Natt	Updated copyright date

© IBM Corporation 2023. All rights reserved.