Lab: Create Tables and Load Data in MySQL using phpMyAdmin

In this lab, you will learn how to create tables and load data in the MySQL database service using the phpMyAdmin graphical user interface (GUI) tool.

Objectives

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

- Create a database.
- Create tables.
- Load data into tables manually using the phpMyAdmin GUI.
- Load data into tables using a text/script file.

Software Used in this Lab

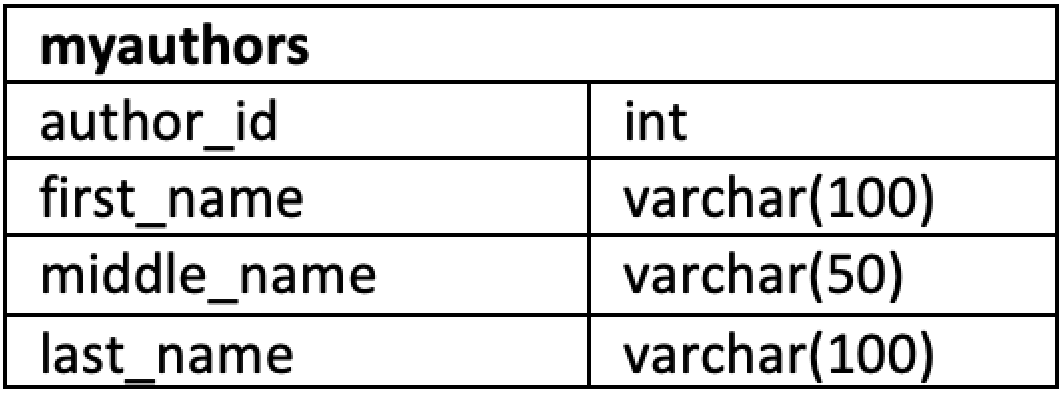
In this lab, you will use MySQL. MySQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve data.

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

Database Used in this Lab

Books database has been used in this lab.

The following diagram shows the structure of the **myauthors** table from the Books database:



myauthors	
author_id	int
first_name	varchar(100)
middle_name	varchar(50)
last_name	varchar(100)

In the table, **author_id** is an integer, **first_name** is a string that stores a maximum of 100 characters, **middle_name** is a string that stores a maximum of 50 characters, and **last_name** is a string that stores a maximum of 100 characters.

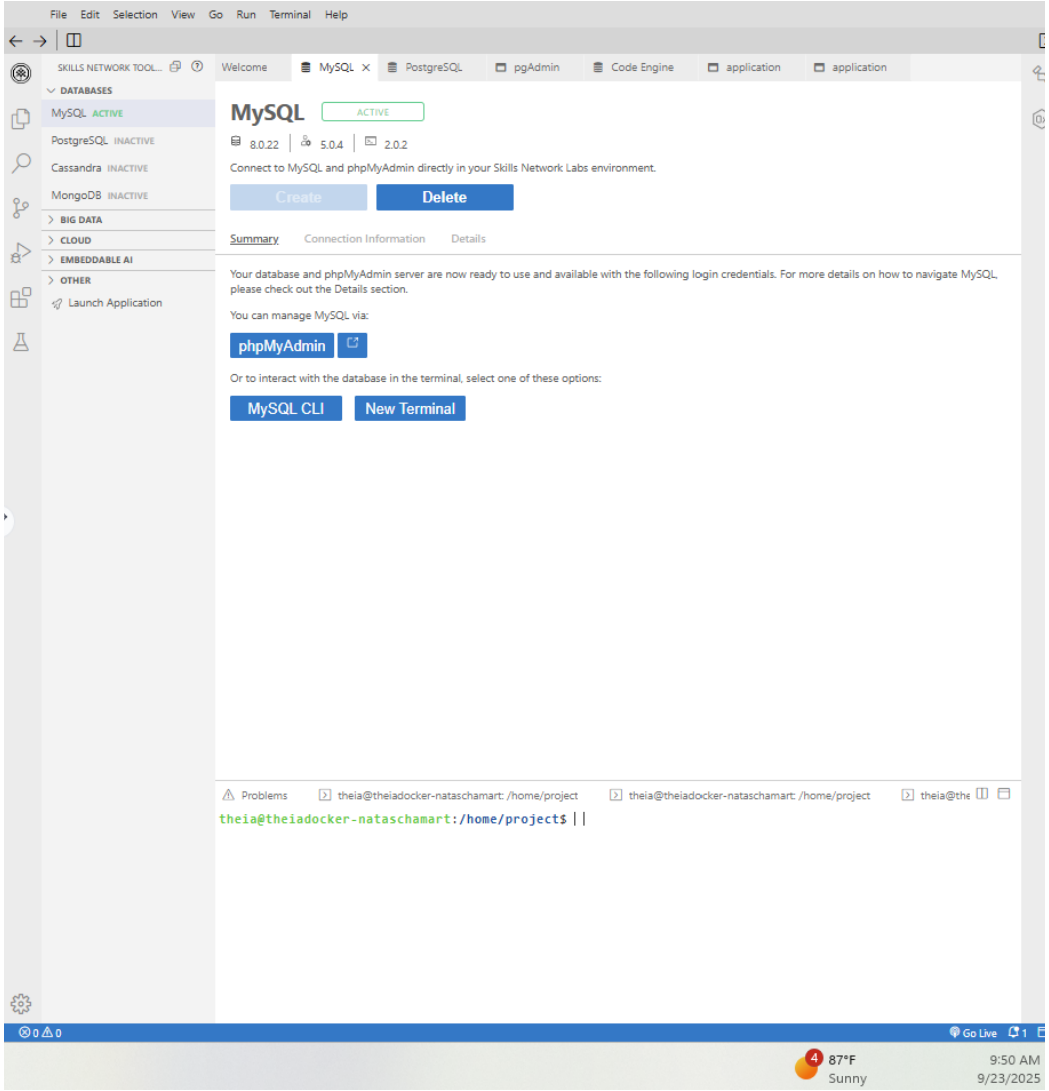
Task A: Create a database

Start the MySQL service session using the Start MySQL in IDE button directive.

Open MySQL Page in IDE

If the icon doesn't start the MySQL database, follow the steps below.

1. Click the Skills Network extension button on the left side of the window.
2. Open the DATABASES menu and click MySQL.
3. Click Create. MySQL may take a few moments to start



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

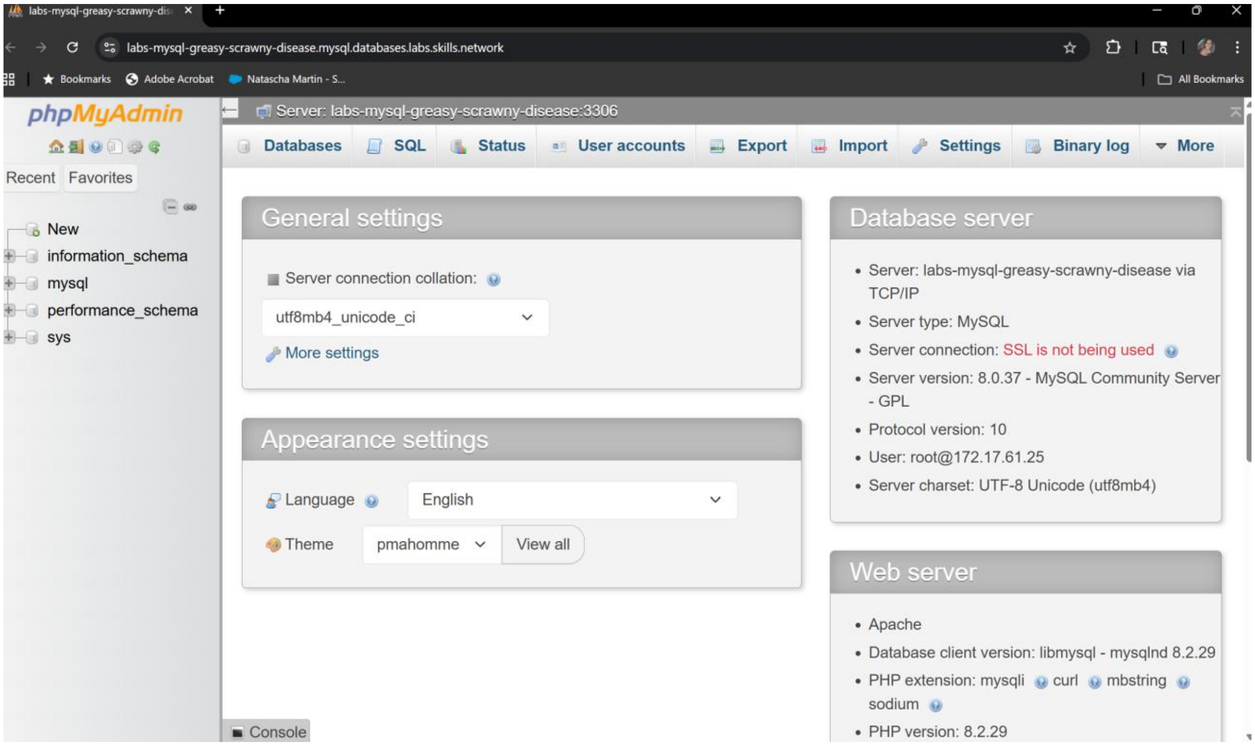
Your database and phpMyAdmin server are now ready to use and available with the following login credentials. For more details on how to navigate MySQL, please check out the Details section.

You can manage MySQL via:

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

```
theia@theiadocker-nataschamart:/home/project$ |
```

Open the phpMyAdmin tool in a new tab in your browser.



General settings

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language: English

Theme: pmahomme

Database server

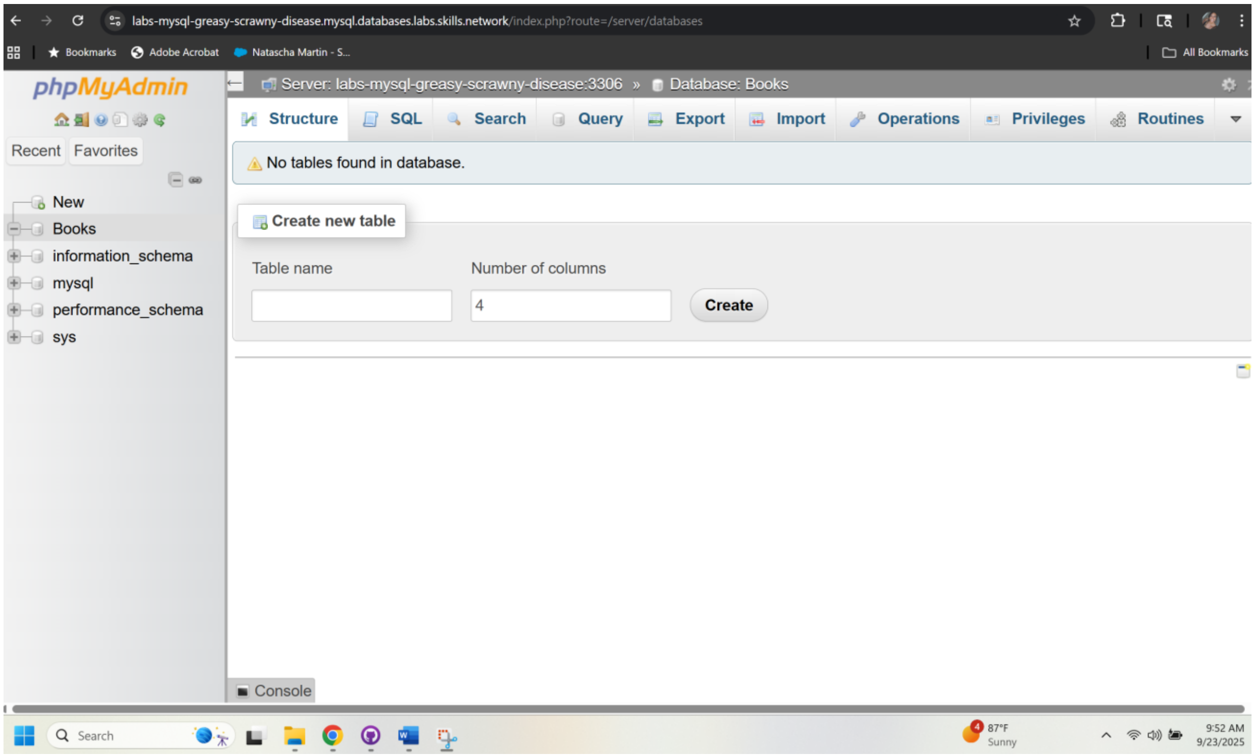
- Server: labs-mysql-greasy-scrawny-disease via TCP/IP
- Server type: MySQL
- Server connection: SSL is not being used
- Server version: 8.0.37 - MySQL Community Server - GPL
- Protocol version: 10
- User: root@172.17.61.25
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache
- Database client version: libmysql - mysqlnd 8.2.29
- PHP extension: mysqli curl mbstring sodium
- PHP version: 8.2.29

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

The encoding will be left as **utf8mb4_0900_ai_ci**. UTF-8 is the most commonly used character encoding for content or data.



No tables found in database.

Create new table

Table name

Number of columns 4

Task B: Create tables

1. In the Create table interface for the empty database **Books**, enter **myauthors** as the table name and **4** for the Number of columns. This is the first step to creating the table **myauthors** that was shown earlier in this lab.

Then click **Go**.

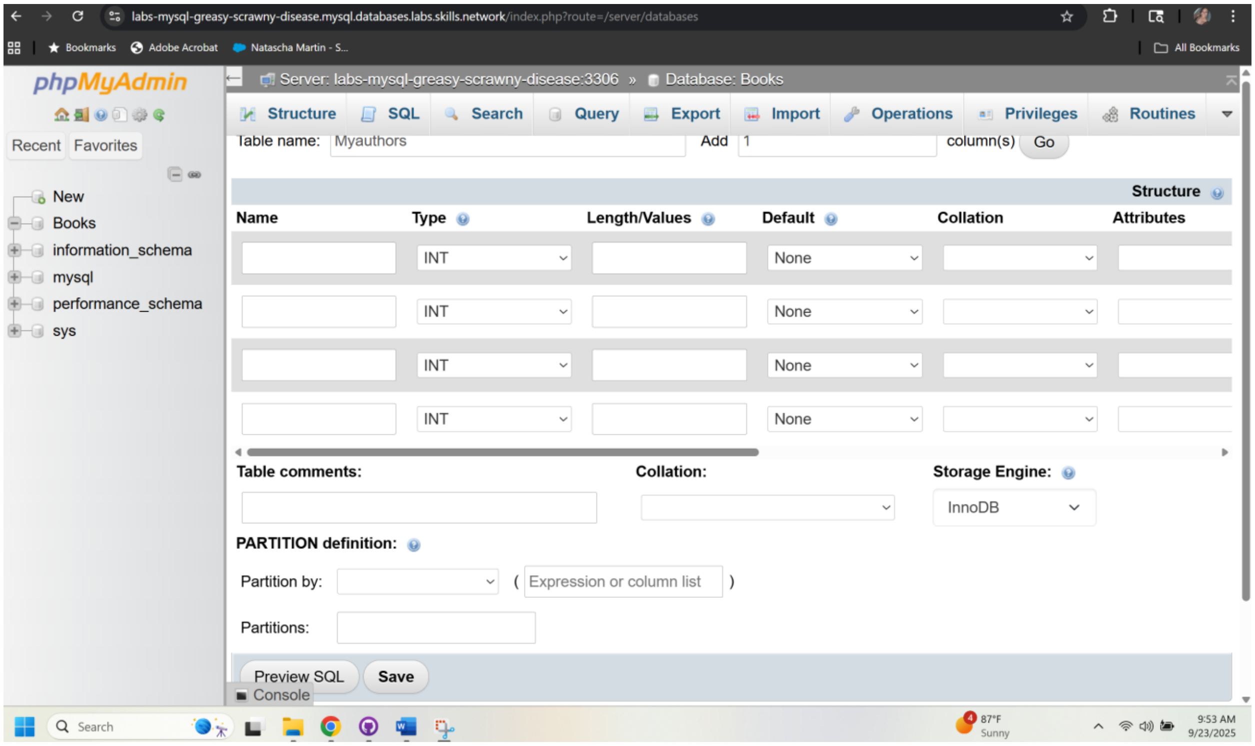


Table name: Myauthors Add 1 column(s)

Structure

Name	Type	Length/Values	Default	Collation	Attributes
	INT		None		
	INT		None		
	INT		None		
	INT		None		

Table comments:

Collation:

Storage Engine: InnoDB

PARTITION definition:

Partition by: (Expression or column list)

Partitions:

Enter the table definition for the **myauthors** table as shown in the image below with highlighted boxes. Then click **Save**.

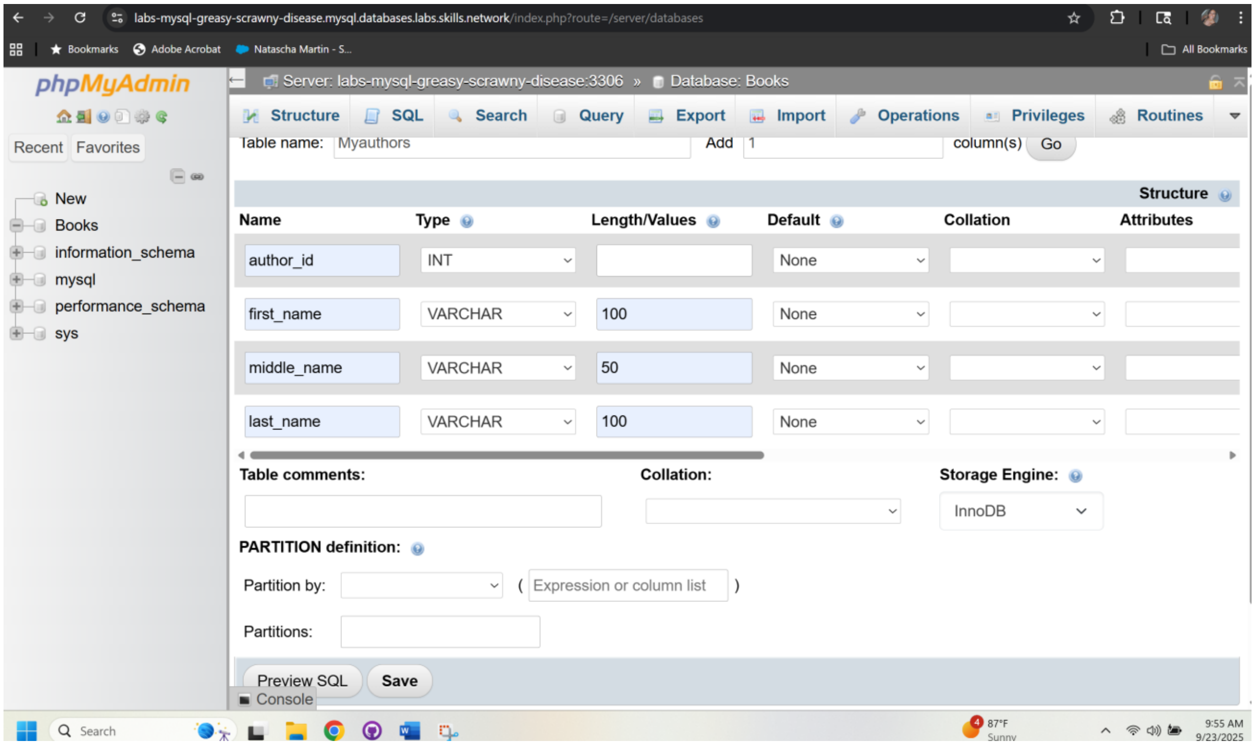


Table name: Myauthors Add 1 column(s)

Structure

Name	Type	Length/Values	Default	Collation	Attributes
author_id	INT		None		
first_name	VARCHAR	100	None		
middle_name	VARCHAR	50	None		
last_name	VARCHAR	100	None		

Table comments:

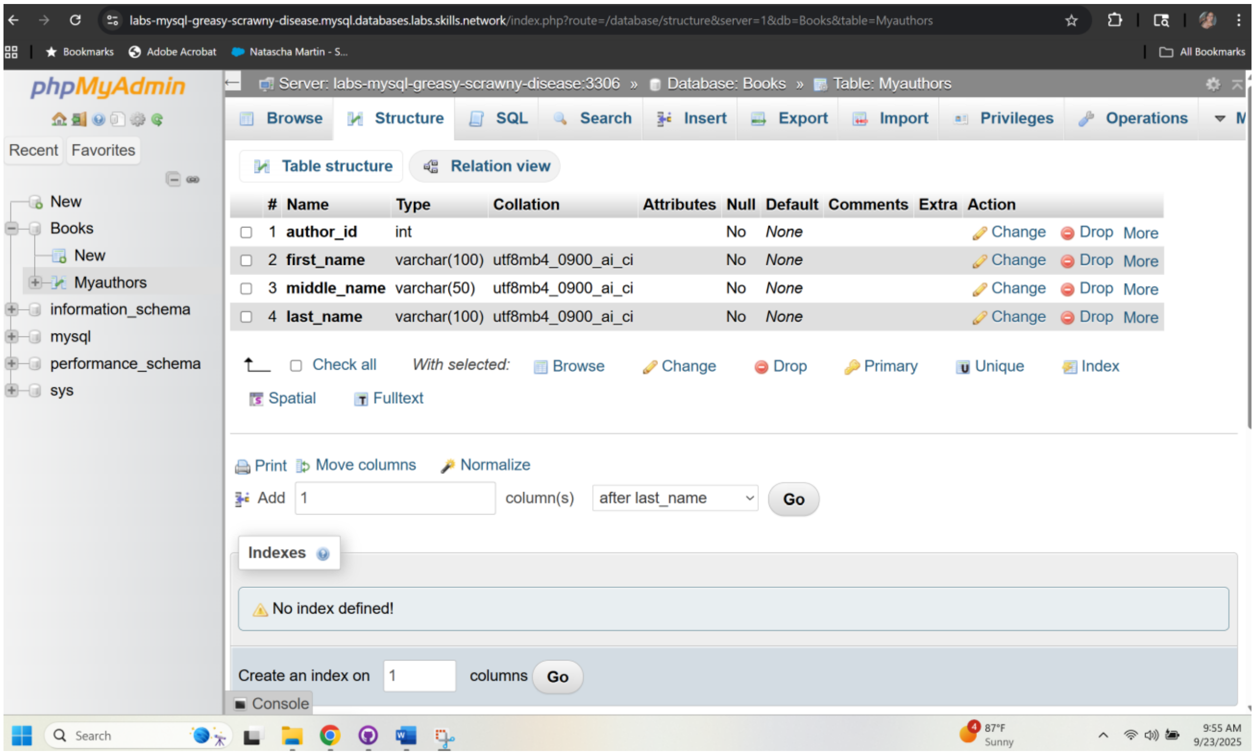
Collation:

Storage Engine: InnoDB

PARTITION definition:

Partition by: (Expression or column list)

Partitions:



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**author_id**	int			No	*None*			Change Drop More
2	**first_name**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*			Change Drop More
3	**middle_name**	varchar(50)	utf8mb4_0900_ai_ci		No	*None*			Change Drop More
4	**last_name**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*			Change Drop More

Add 1 column(s) after last_name

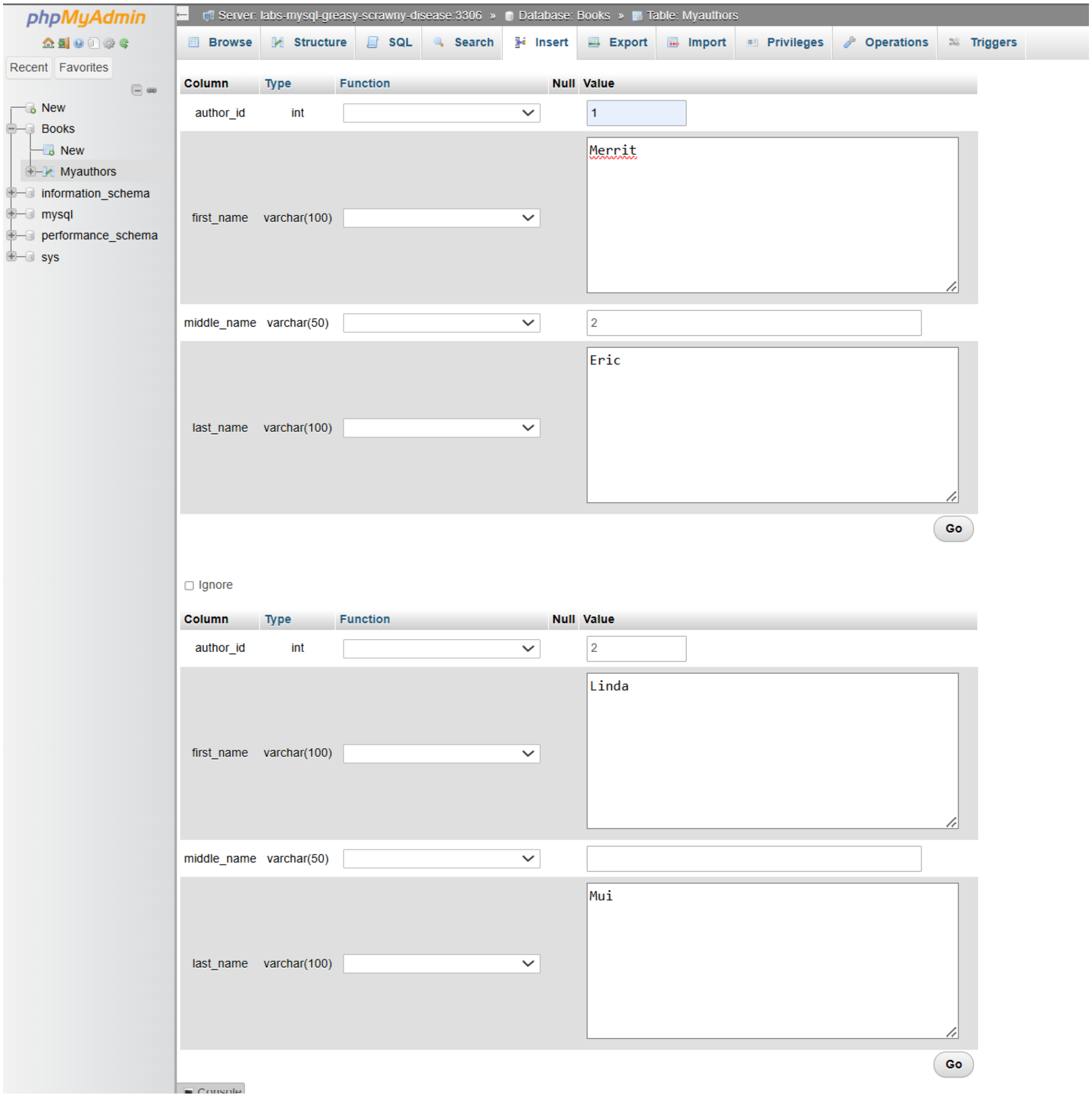
No index defined!

Create an index on 1 columns

Task C: Load data into tables manually using the phpMyAdmin GUI

1. Sometimes, you may want to load a few data rows of data, but you may not have a SQL script on hand to do that. In this case, you can manually load the data into phpMyAdmin. Since this is a manual process, it is better for inserting a small amount of data rather than a large amount.

To load data manually, go to the **Insert** tab for the **myauthors** table. Enter data for 2 rows of the **myauthors** table as shown in the image below with highlighted boxes. Then click **Go** at the bottom.

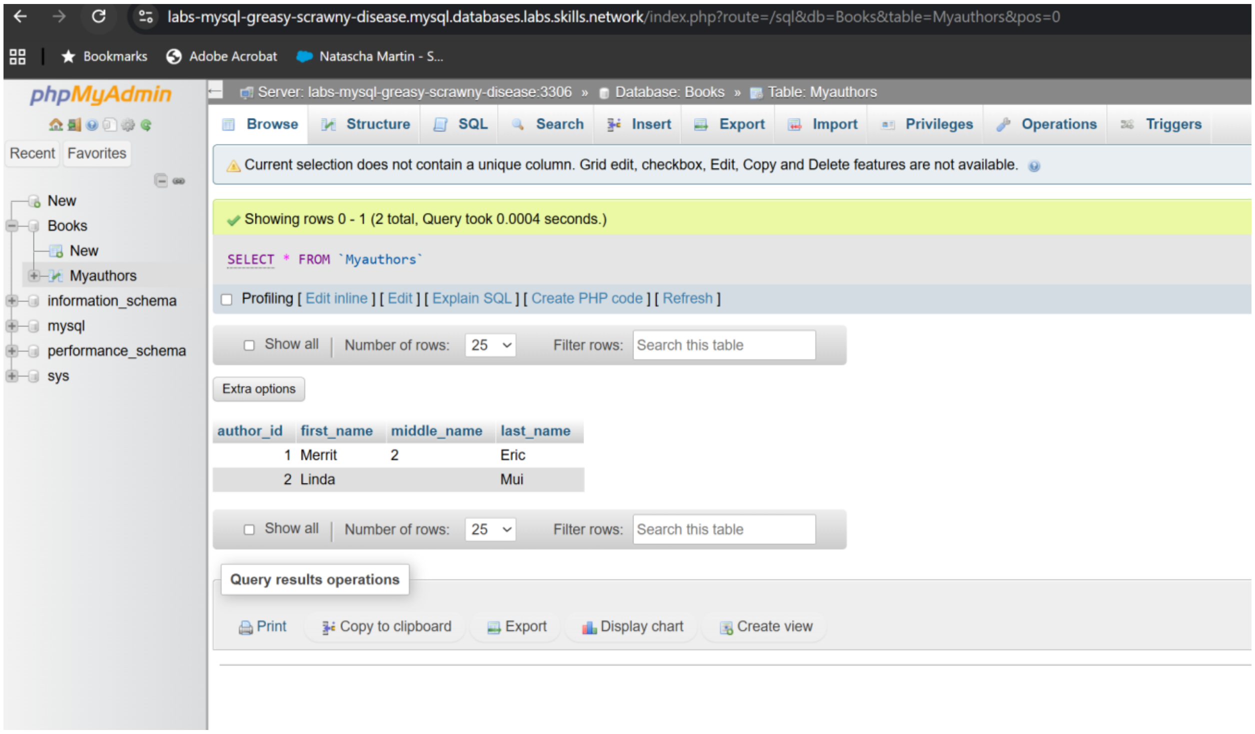


Server: labs-mysql-greasy-scrawny-disease:3306 » Database: Books » Table: Myauthors

Column	Type	Function	Null	Value
author_id	int			1
first_name	varchar(100)			Merrit
middle_name	varchar(50)			2
last_name	varchar(100)			Eric

☐ Ignore

Column	Type	Function	Null	Value
author_id	int			2
first_name	varchar(100)			Linda
middle_name	varchar(50)			
last_name	varchar(100)			Mui



labs-mysql-greasy-scrawny-disease.mysql.databases.labs.skills.network/index.php?route=/sql&db=Books&table=Myauthors&pos=0

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
SELECT * FROM `Myauthors`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

author_id	first_name	middle_name	last_name
1	Merrit	2	Eric
2	Linda		Mui

□ Show all | Number of rows: 25 Filter rows: Search this table

Task D: Load data into tables using a text/script file

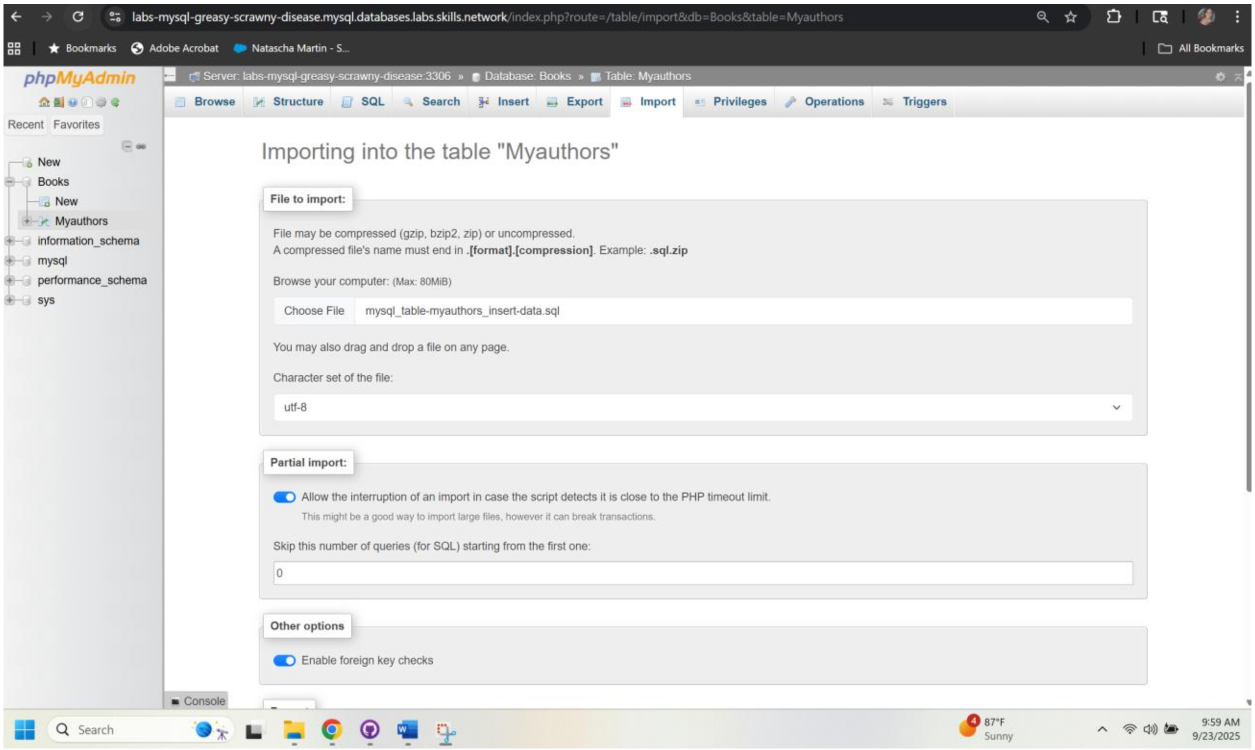
1. Now you will use a SQL script to import the remainder of the **myauthors** table data. A SQL script file contains commands and statements that perform operations on your database, and can be useful when importing a large amount of data.

Download the SQL script below to your local computer:

- mysql_table-myauthors_insert-data.sql

2. Go to **Import** tab for the **myauthors** table. Click **Choose File** and load the **mysql_table-myauthors_insert-data.sql** file from your local computer storage. The rest of the settings can be left as they are because you are importing a SQL script that is encoded with UTF-8.

Then click **Go**. Notification of import success will appear.



Importing into the table "Myauthors"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: (Max: 80MiB)

Choose File mysql_table-myauthors_insert-data.sql

You may also drag and drop a file on any page.

Character set of the file: utf-8

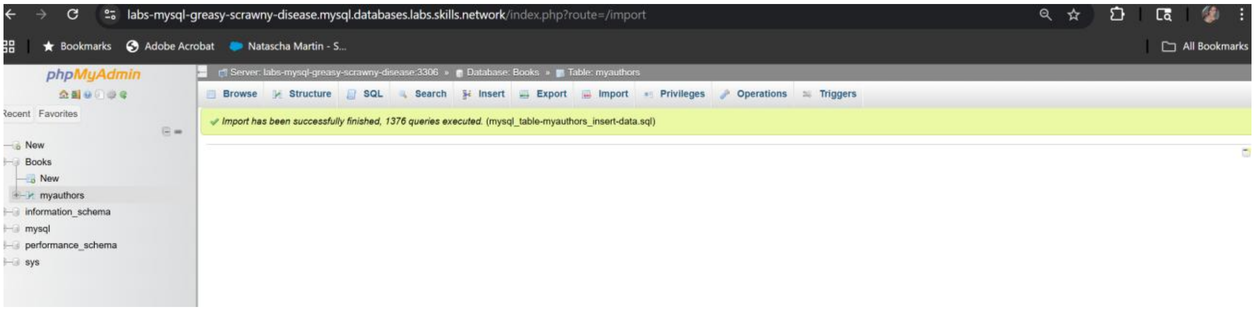
Partial import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit.
This might be a good way to import large files, however it can break transactions.

Skip this number of queries (for SQL) starting from the first one: 0

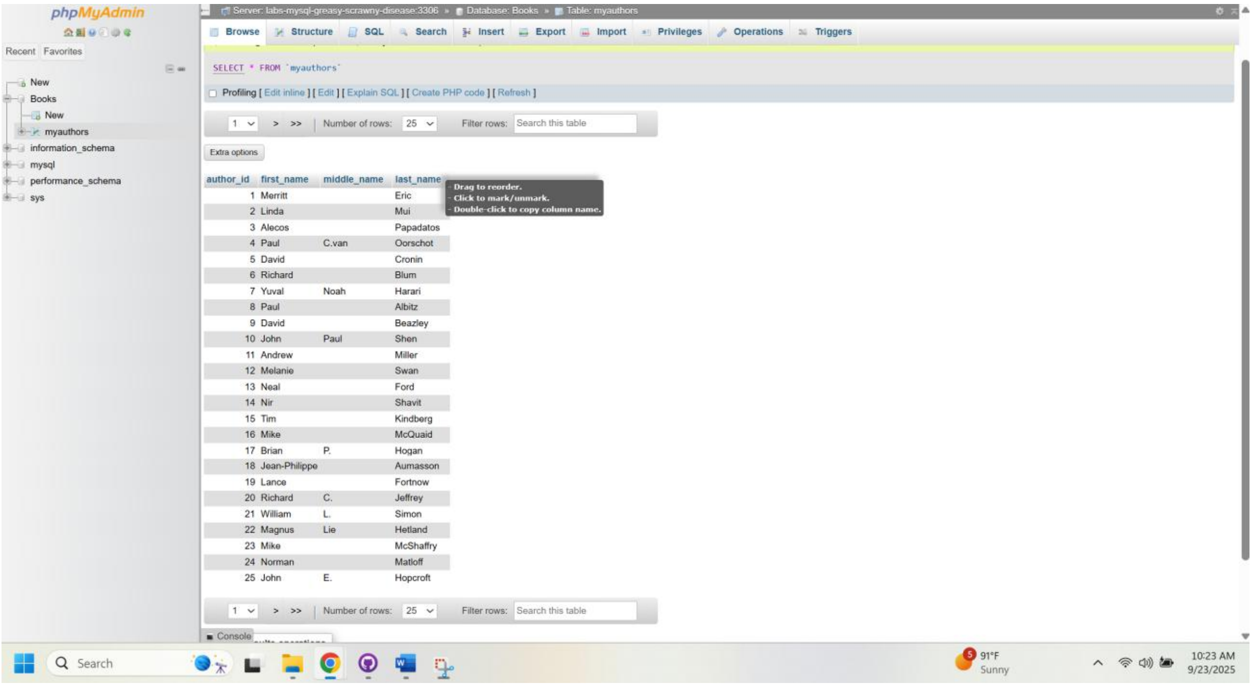
Other options

☑ Enable foreign key checks



✓ *Import has been successfully finished, 1376 queries executed.* (mysql_table-myauthors_insert-data.sql)

Go to the **Browse** tab for the **myauthors** table again to check the newly inserted rows appear along with previously inserted 2 rows.



```
SELECT * FROM `myauthors`
```

author_id	first_name	middle_name	last_name
1	Merritt		Eric
2	Linda		Mui
3	Alecos		Papadatos
4	Paul	C.van	Oorschot
5	David		Cronin
6	Richard		Blum
7	Yuval	Noah	Harari
8	Paul		Albitz
9	David		Beazley
10	John	Paul	Shen
11	Andrew		Miller
12	Melanie		Swan
13	Neal		Ford
14	Nir		Shavit
15	Tim		Kindberg
16	Mike		McQuaid
17	Brian	P.	Hogan
18	Jean-Philippe		Aumasson
19	Lance		Fortnow
20	Richard	C.	Jeffrey
21	William	L.	Simon
22	Magnus	Lie	Hetland
23	Mike		McShaffry
24	Norman		Matloff
25	John	E.	Hopcroft

Lab Summary: Create Tables and Load Data in MySQL using phpMyAdmin

Objective

The purpose of this lab was to learn how to use phpMyAdmin with MySQL to:

- Create a database
- Create and define tables
- Load data manually into tables
- Load data using a script file

This lab emphasized both GUI-based operations and importing SQL scripts for efficiency.

Steps Completed

1. Database Creation

- A new database named **Books** was created in phpMyAdmin.
- UTF-8 encoding (utf8mb4_0900_ai_ci) was used as default.

2. Table Creation

- Within the *Books* database, a table named **Myauthors** was created.
- Columns were defined as follows:

    - author_id (INT)
    - first_name (VARCHAR 100)
    - middle_name (VARCHAR 50)
    - last_name (VARCHAR 100)

3. Manual Data Insertion

- Using the **Insert** tab in phpMyAdmin, two rows of author data were entered manually.
- Verification was done through the **Browse** tab to confirm successful insertion.

4. Script-Based Data Load

- The SQL script file mysql_table-myauthors_insert-data.sql was uploaded via the **Import** tab.
- This file inserted the remainder of the rows into the **Myauthors** table.
- Initial errors occurred due to table name mismatches (Myauthors vs myauthors). The issue was resolved by ensuring consistent naming.
- Once corrected, the import completed successfully.

Key Learnings

- phpMyAdmin provides two main ways to load data: manual entry and bulk import through SQL scripts.
- Case sensitivity in table names can cause import errors, especially if the script and database objects don't match exactly.
- Domain, entity, and referential integrity all depend on correct schema setup and careful consistency across naming conventions.

Completion Status

Lab completed successfully.

- All tasks (database creation, table creation, manual insert, and script-based insert) were executed and verified.
- Issues encountered were related to naming mismatches, which were resolved without restarting the entire lab.