



# Hands-on Lab : Keys and Constraints in MySQL

**Estimated time needed:** 20 minutes

In this lab, you will learn how to add keys to create relationships between the tables and how to use constraints to enforce rules on the data entry in the MySQL database service using the phpMyAdmin graphical user interface (GUI) tool.

## 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.

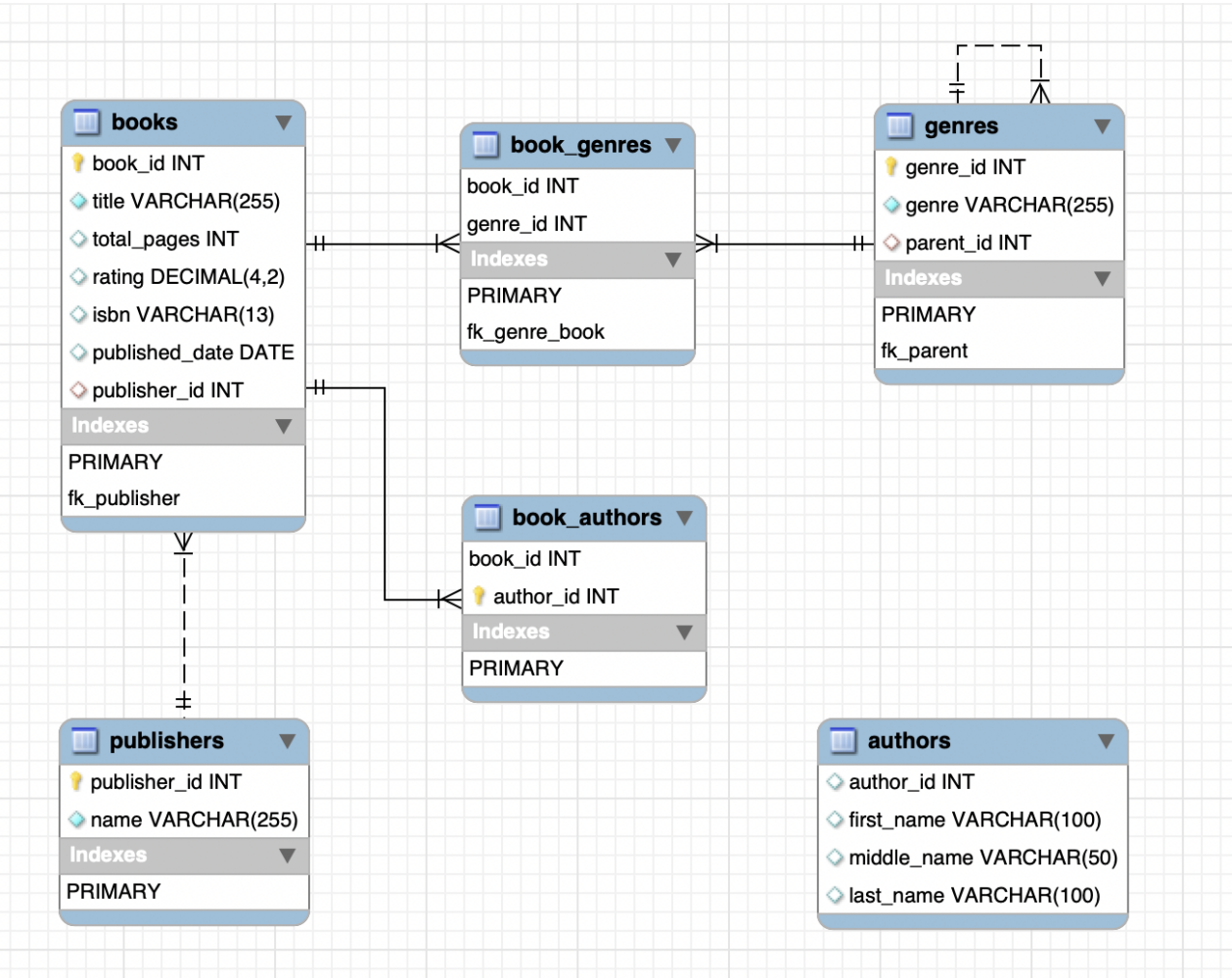


To complete this lab you will utilize MySQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

## Database Used in this Lab

The eBooks database has been used in this lab.

The following entity relationship diagram (ERD) shows the current status of the schema of the eBooks database used in this lab:



## Objectives

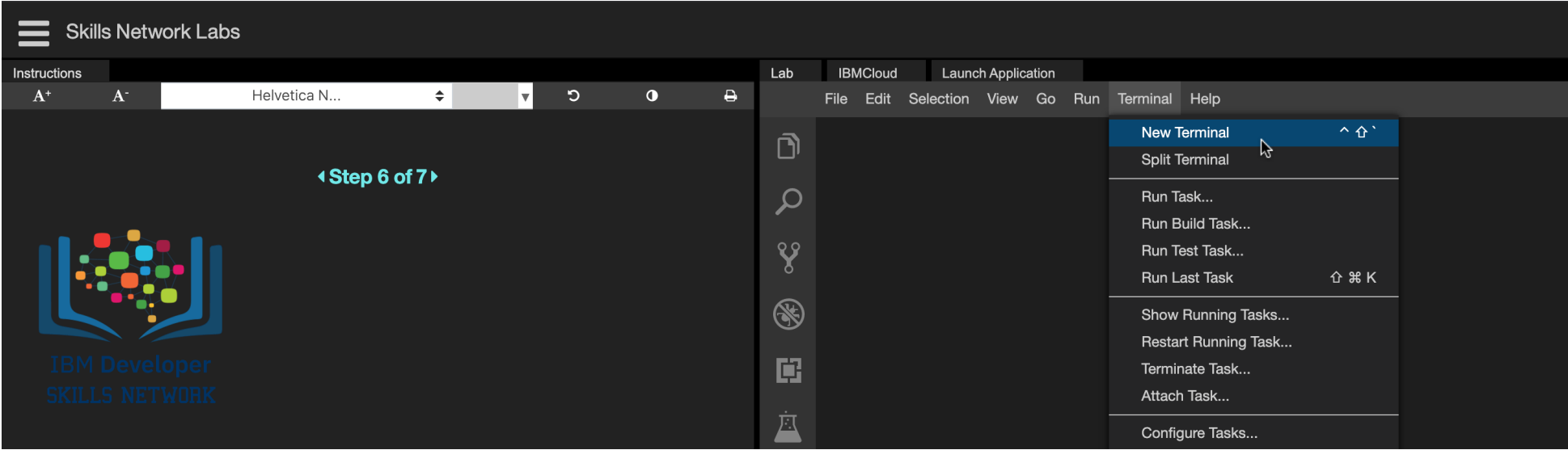
After completing this lab, you will be able to use MySQL phpMyAdmin to:

- Create primary and foreign keys.
- Add constraints to data columns.

# Exercise

In this exercise, you will learn how to add keys to create relationships between the tables and how to use constraints to enforce rules on the data entry in the MySQL database service using the phpMyAdmin graphical user interface (GUI) tool.

1. To start, download the the **eBooks** mysql dump file (containing the eBooks database table definitions and data) below to your local computer storage. A dump file is a text file that contains the data from a database in the form of SQL statements. This file can then be imported in phpMyAdmin.
  - [eBooks\\_mysql\\_dump.sql](#)
2. Go to **Terminal > New Terminal** to open a terminal from the side by side launched Cloud IDE.



3. Start the MySQL service session in the Cloud IDE using the command below in the terminal. Find your MySQL service session password from the highlighted location of the terminal shown in the image below. Note down your MySQL service session password because you may need to use it later in the lab.

```
start_mysql

theia@theiadocker-sandipsahajo:/home/project$ start_mysql
Starting your MySQL database....
This process can take up to a minute.

MySQL database started, waiting for all services to be ready....

Your MySQL database is now ready to use and available with username: root password: MTY5MTUtc2FuZG1w

You can access your MySQL database via:
• The browser at: https://sandipsahajo-8080.theiadocker-27.proxy.cognitiveclass.ai
• CommandLine: mysql --host=127.0.0.1 --port=3306 --user=root --password=MTY5MTUtc2FuZG1w
theia@theiadocker-sandipsahajo:/home/project$
```

4. Copy your phpMyAdmin weblink from the highlighted location of the terminal shown in the image below and paste it to a new tab of your web browser.

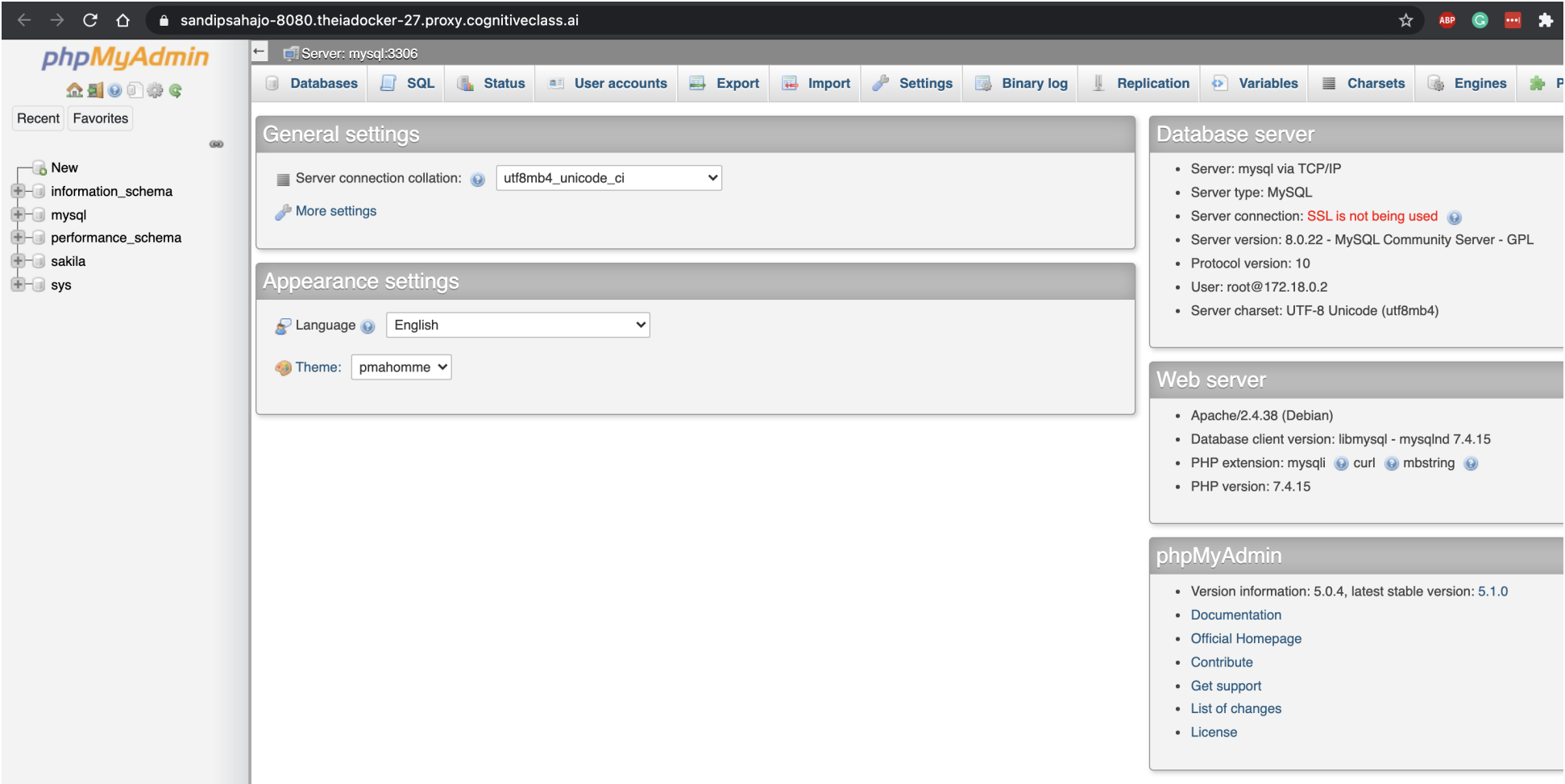
```
theia@theiadocker-sandipsahajo:/home/project$ start_mysql
Starting your MySQL database....
This process can take up to a minute.

MySQL database started, waiting for all services to be ready....

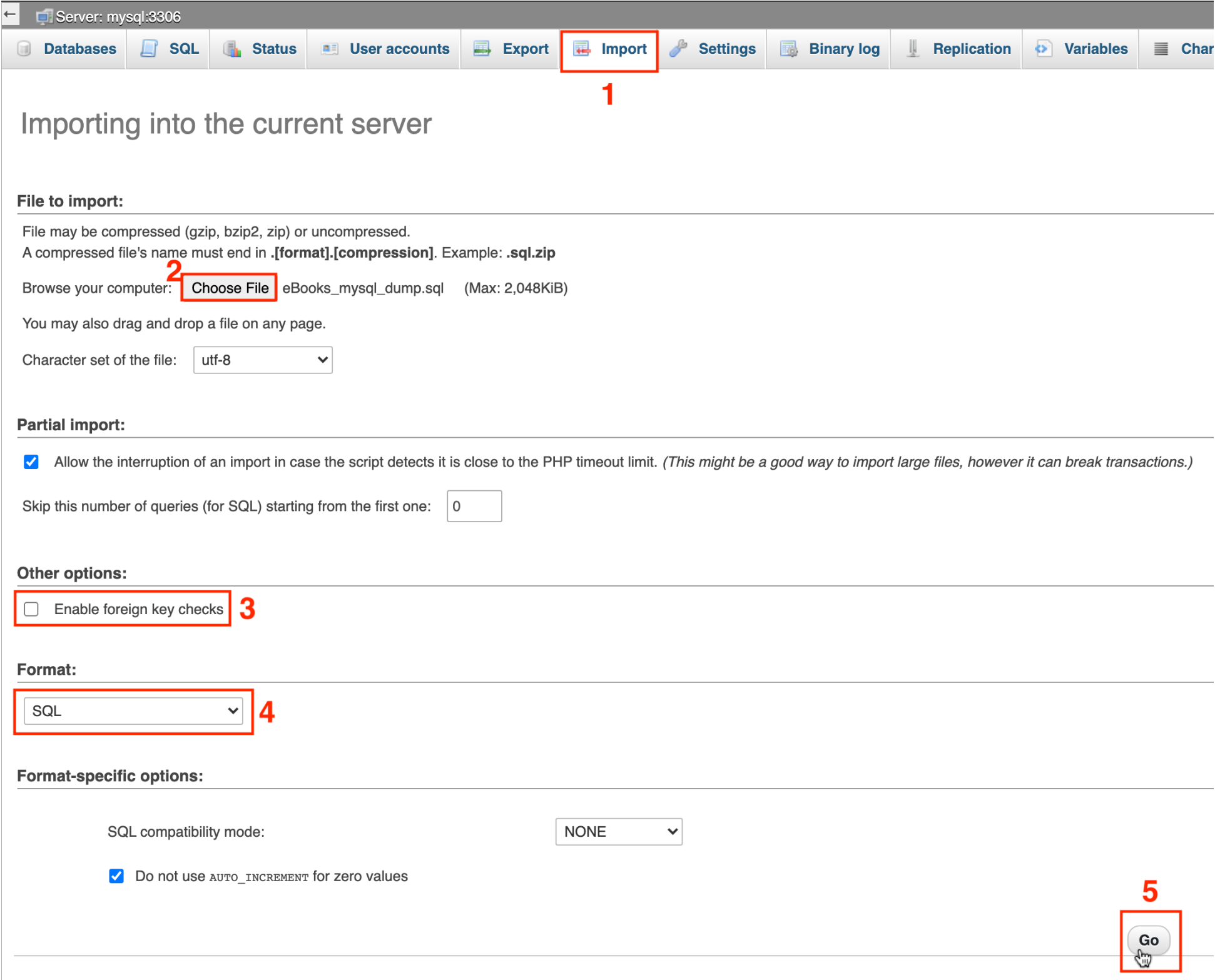
Your MySQL database is now ready to use and available with username: root password: MTY5MTUtc2FuZG1w

You can access your MySQL database via:
• The browser at: https://sandipsahajo-8080.theiadocker-27.proxy.cognitiveclass.ai
• CommandLine: mysql --host=127.0.0.1 --port=3306 --user=root --password=MTY5MTUtc2FuZG1w
theia@theiadocker-sandipsahajo:/home/project$
```

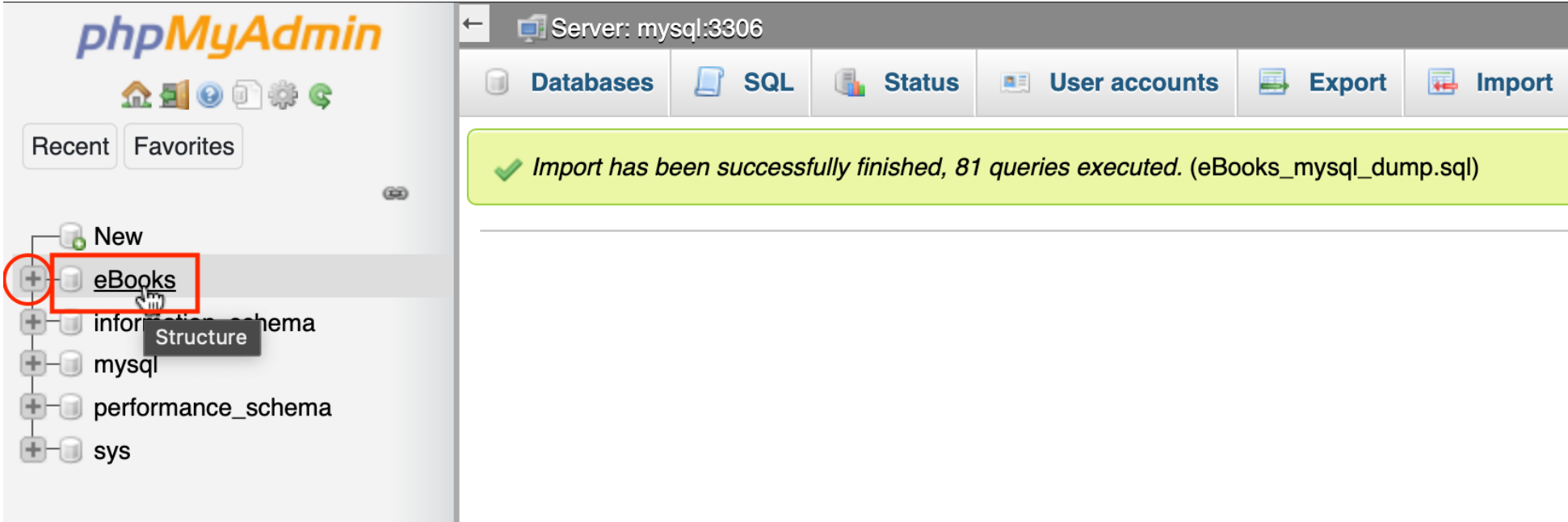
5. You will see the phpMyAdmin GUI tool.



6. Go to the **Import** tab. Click **Choose File** and load the **eBooks\_mysql\_dump.sql** file you downloaded earlier to your local computer storage. Next, uncheck **Enable foreign key checks** and select SQL as **Format**. Then click **Go**.



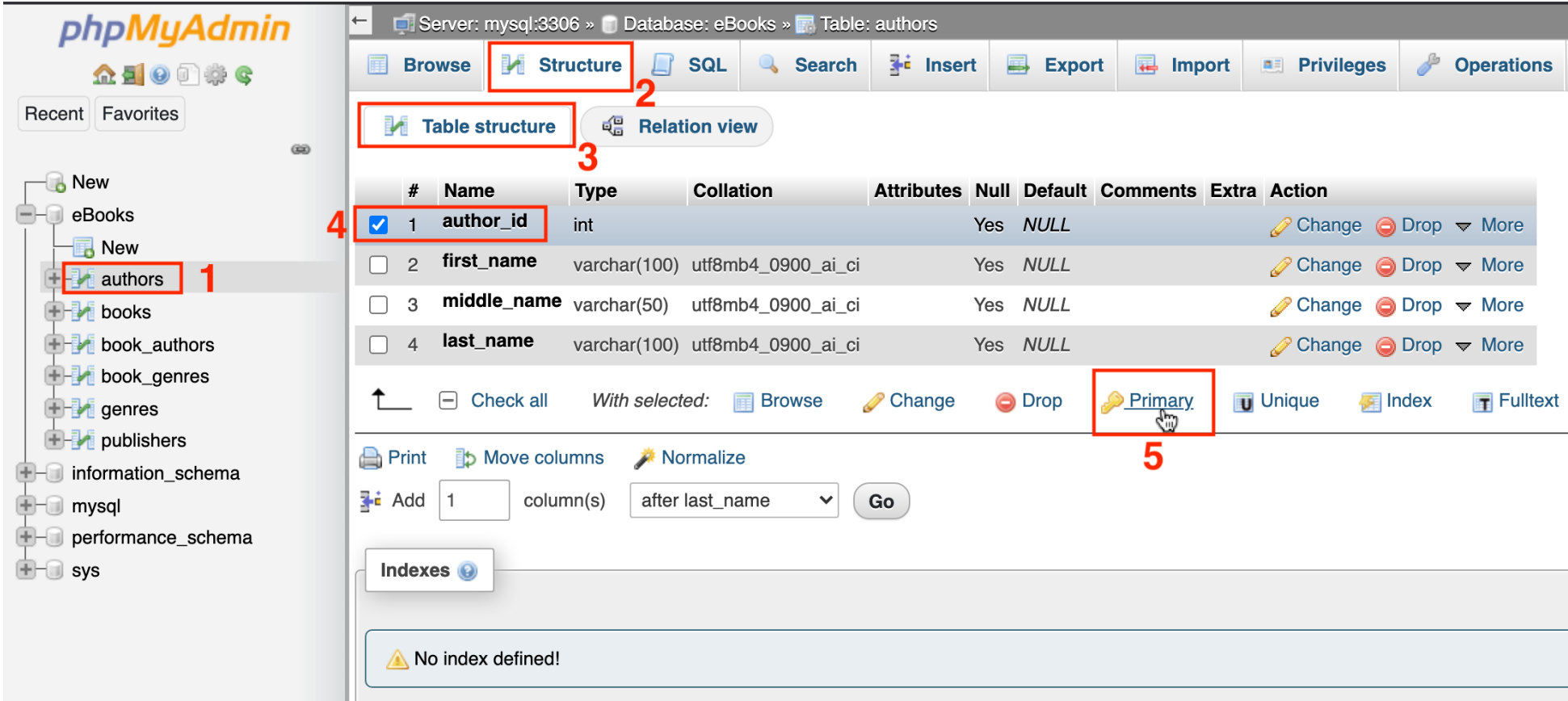
7. You will be notified that the import has successfully finished. Click the database **eBooks** as shown below to expand (if necessary click the **+** icon beside **eBooks**). You will see the list of tables from the eBooks database.



8. **Primary Keys:** A primary key is a column or group of columns that uniquely identify every row in a table. They constrain the table by only accepting unique, non-NULL values for that column. Non-NULL data is data where the values exist.

Creating a primary key on a table automatically creates an index on the key. You will create a primary key for the **author** table to uniquely identify every row in the table. You will set the **author\_id** column of the **author** table as a primary key.

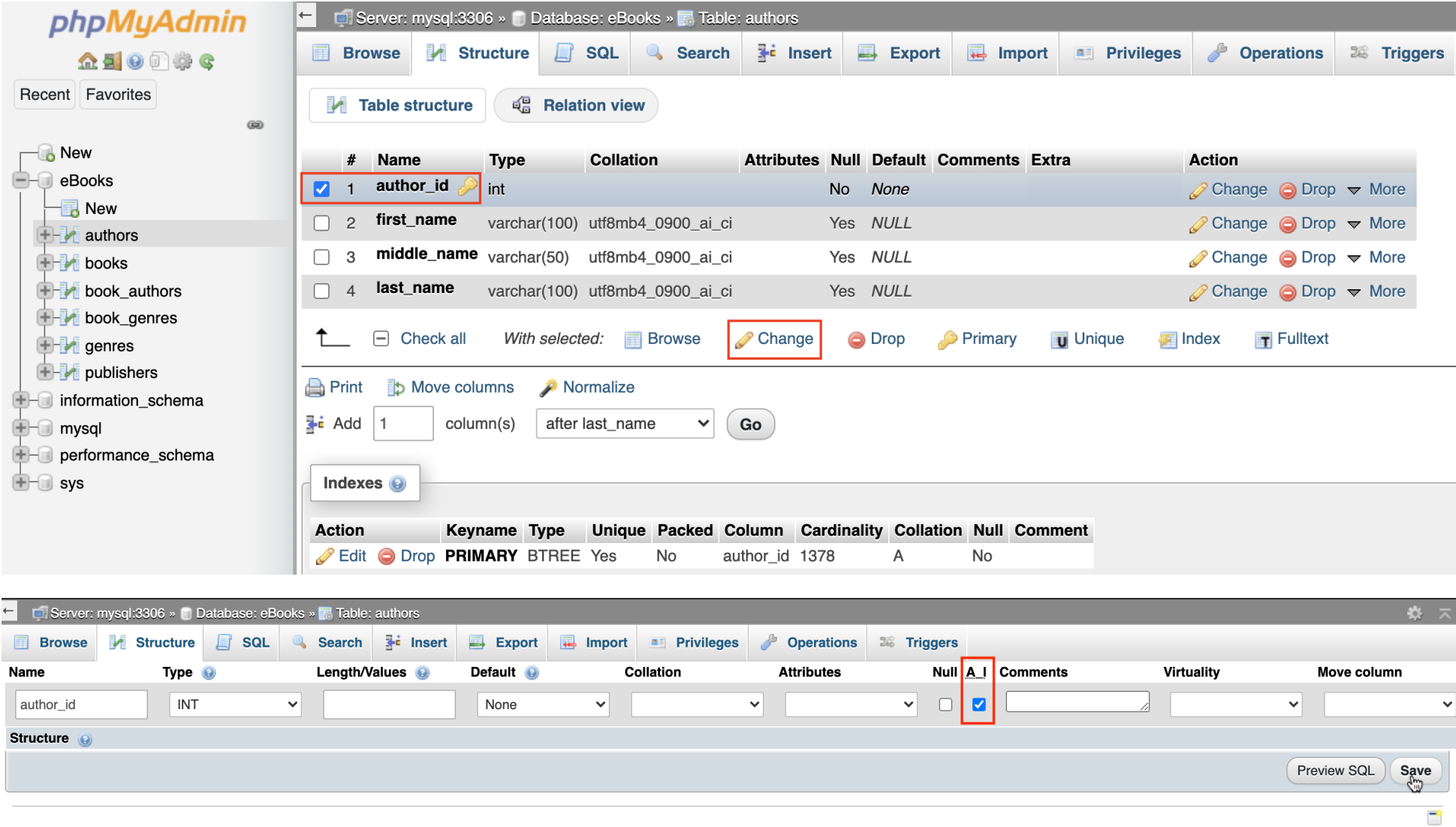
- In the tree-view, click the **authors** table. Switch to the **Structure** tab and make sure you are inside **Table structure** subtab.
- Check the **author\_id** column.
- Click the **Primary** option.



9. **Auto-increment:** Auto-increment enables a unique number to be generated automatically when a new record is inserted into a table. Often this is used for the primary key field to be created automatically every time a new record is inserted. You will set the auto-increment feature for the primary key of the **author** table.

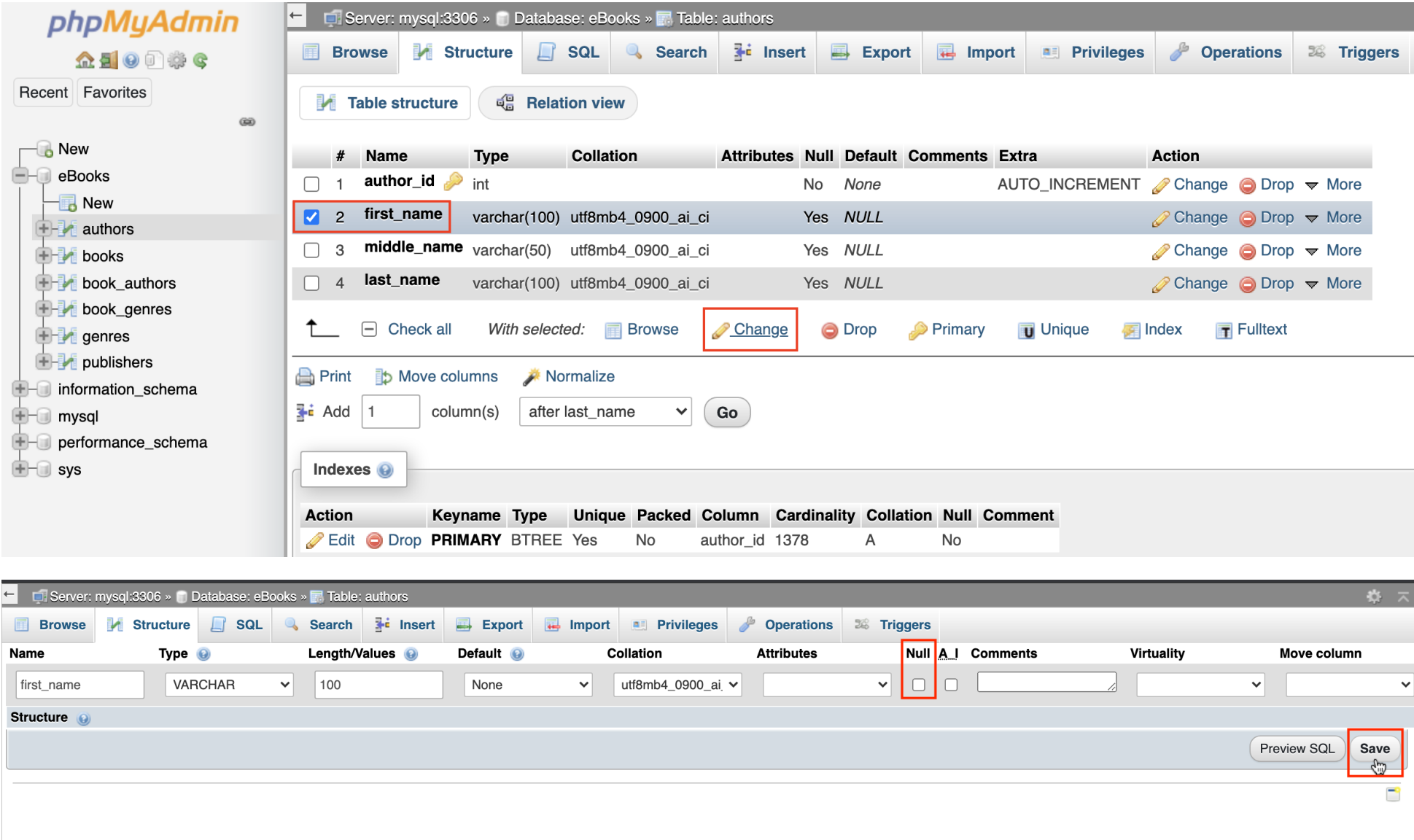
- In the tree-view, click the **authors** table. Switch to the **Structure** tab and make sure you are inside the **Table structure** subtab.
- Check the **author\_id** column.
- Click the **Change** option.
- Check **A\_I** option (A\_I = Auto\_Increment).
- Click **Save**.





10. **Null constraints:** By default, a column can hold NULL values. If you do not want a column to have a NULL value, you can restrict the column from having a NULL value. You will restrict the **first\_name** column of the **authors** table from having a NULL value.

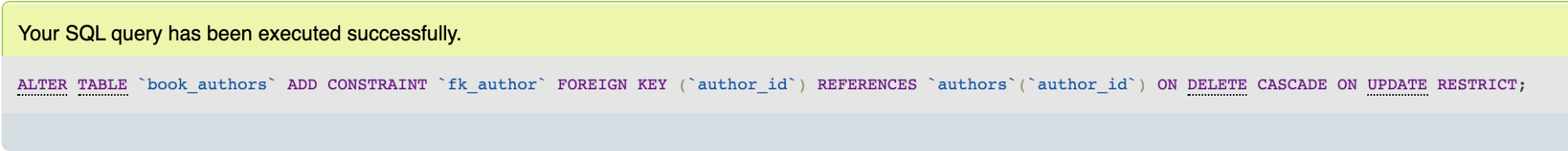
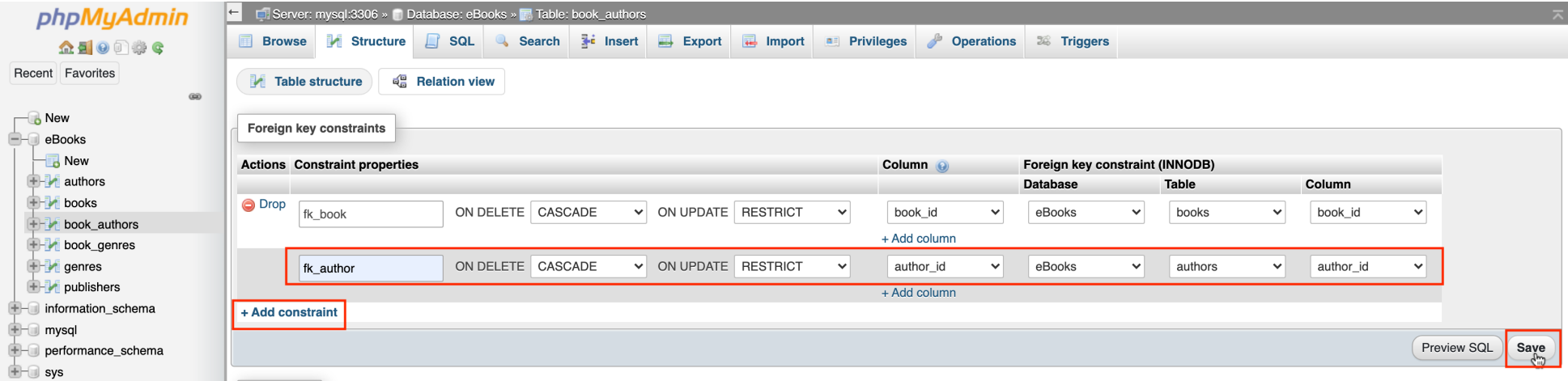
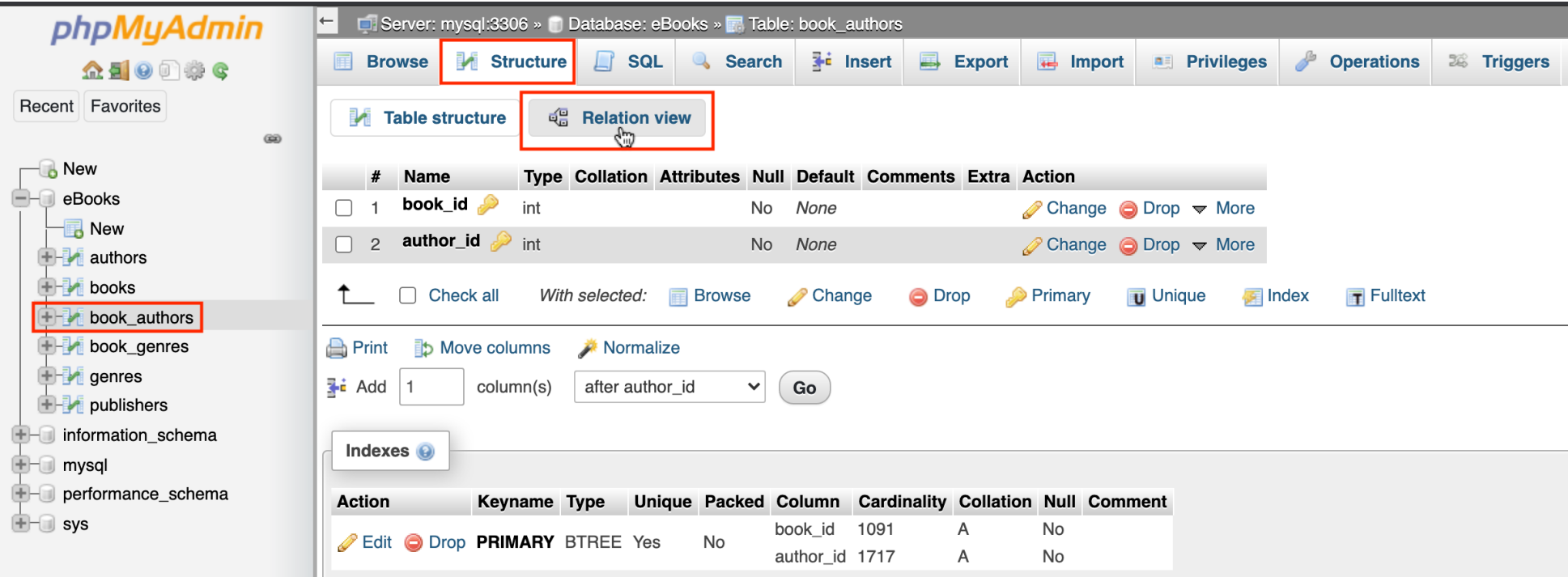
- In the tree-view, click the **authors** table. Switch to the **Structure** tab and make sure you are inside the **Table structure** subtab.
- Check the **first\_name** column.
- Click the **Change** option.
- Uncheck the **Null** option.
- Click **Save**.



11. **Foreign keys:** A foreign key is a column that establishes a relationship between two tables. It acts as a cross-reference between two tables as it points to the primary key of one table (the parent table), constraining the data in another table (the child table) by the data in the parent table. You will create a foreign key for the **book\_authors** table by setting its **author\_id** column as a foreign key, to

establish a relationship between the **book\_authors** and **authors** tables.

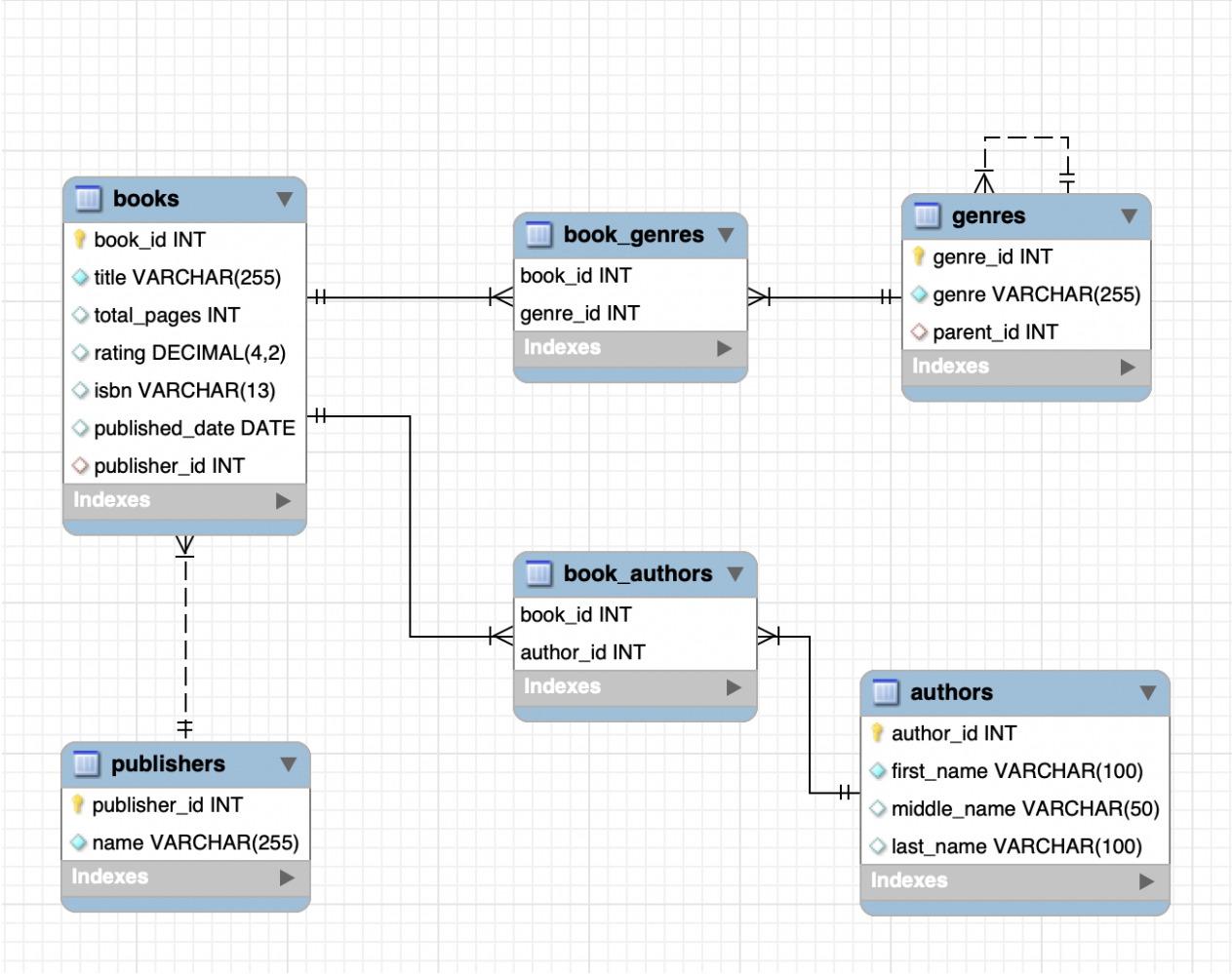
- In the tree-view, click the **book\_authors** table. Switch to the **Structure** tab and make sure you are inside the **Relation view** subtab.
- If necessary, click **Add constraint** to create a new foreign key constraint placeholder.
- Fill the placeholders as shown in a image below.
- Click **Save**.



**CASCADE** means that when rows are deleted or updated in the parent table, the corresponding rows in the child table will also be deleted or updated.

**RESTRICT** means that rows cannot be deleted or updated in the parent table if there are corresponding rows in the child table.

12. After creating/adding all the above necessary primary keys, foreign keys, and constraints, the schema of the complete eBooks database will look like the following ERD diagram:
- **Note:** You don't need to generate any ERD diagram like below for this lab. By comparing the earlier eBooks schema ERD (shown in the section "Database Used in this Lab") and this complete eBooks schema ERD, just try to understand how all the operations you did above made the eBooks database complete.



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

## Author(s)

- [Sandip Saha Joy](#)

## Other Contributor(s)

- Kathy An

## Changelog

Date	Version	Changed by	Change Description
2021-03-22	1.0	Sandip Saha Joy	Created initial version
2021-10-18	1.1	Kathy An	Updated lab instructions

© IBM Corporation 2021. All rights reserved.