SQL Lab

Estimated Time Needed: 60 min

In this challenge, you will create a database, import data from three sources to populate tables, and perform database operations 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 and populate a database and tables.
- Execute Structured Query Language (SQL) commands to perform basic database operations.
- Retrieve data from tables using SELECT statements.
- Filter the data output using WHERE statements.
- Aggregate data get ordered results using functions like SUM, MIN, MAX, ORDER BY.
- Use window functions to get specified output.
- Retrieve data from two or more tables using SQL JOINS.

Software used in this lab

You will use MySQL to complete this lab. MySQL is a free, open-source relational database system that offers a command line interface (MySQL) and a third-party web interface (phpMyAdmin) to efficiently store, manipulate, and retrieve data.

MySQL is a service available on Skills Network Labs (SN Labs) Cloud IDE, a virtual lab environment used in this course. SN Labs Cloud IDE is great way to do projects without downloading, installing, configuring, and integrating software on your own computer.

Two Components of the SN Labs Cloud IDE:

- The instructions that you will follow to complete this lab are displayed on the left side of the screen.
- The area on the right side of the screen is where you will use the menus, terminals, and tools to complete the lab exercises.

Dataset used in This Lab

The datasets used in this lab are three SQL files called **Salary Data**, **Sales Data** and **Employee Data**. To complete the exercises in this lab, you will be instructed to save and upload the datasets to your local machine and use a locally installed database or the tool provided in the course.

Import the given SQL input files into a database.

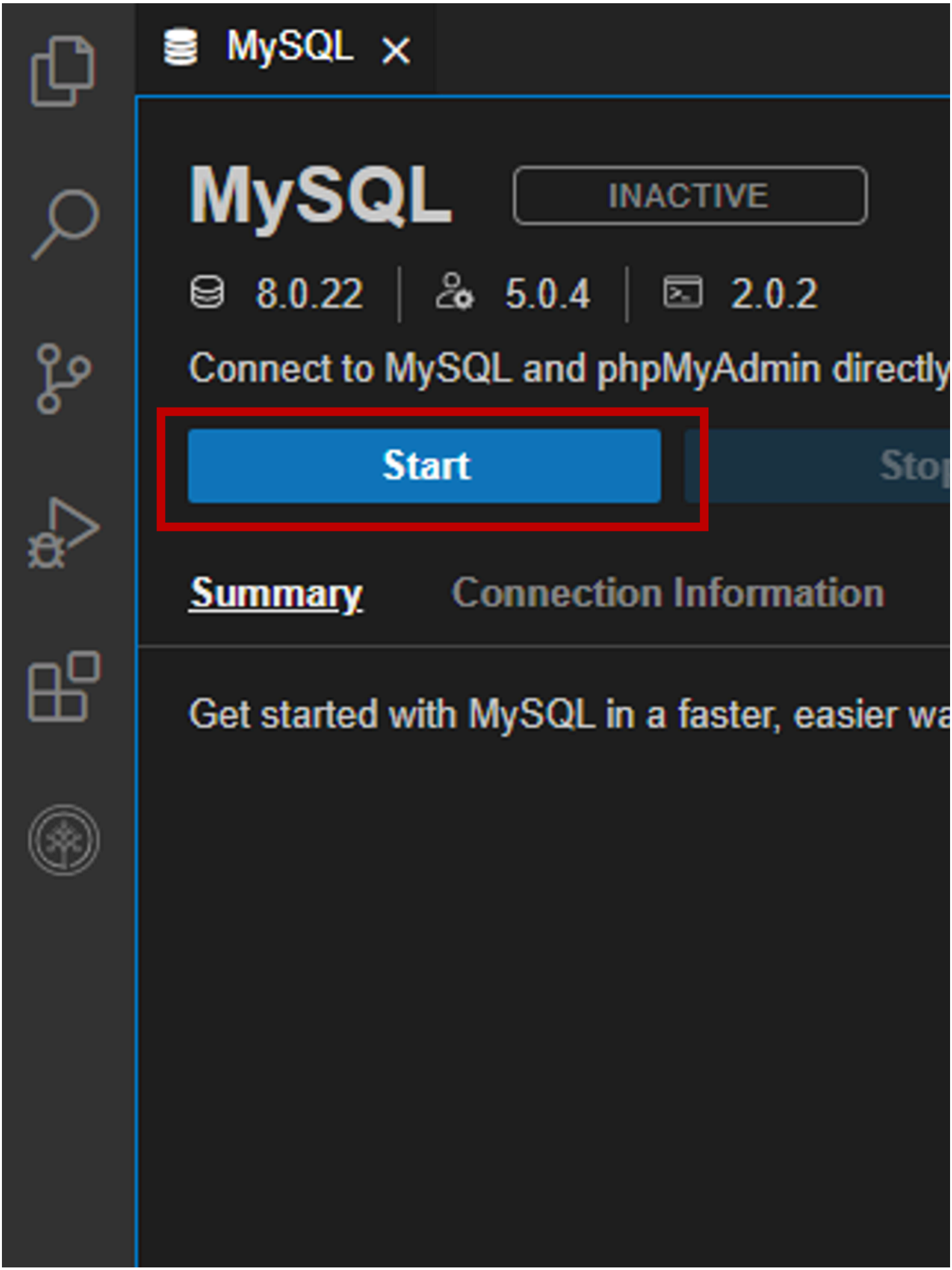
Prework - Create and populate database

TASK A: Create a Database

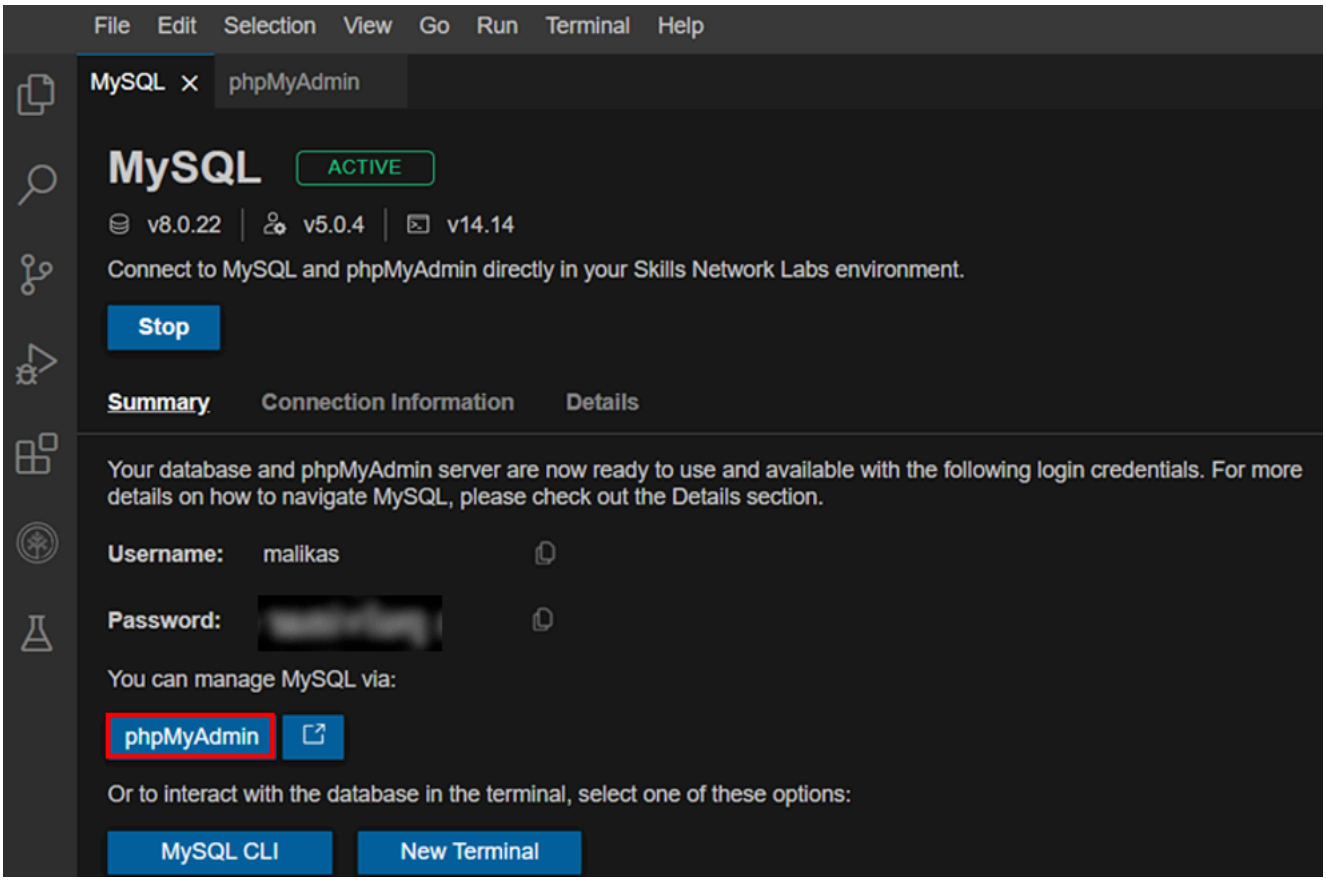
1. Start the MySQL service session using the `Open MySQL Page in IDE` button.

Open MySQL Page in IDE

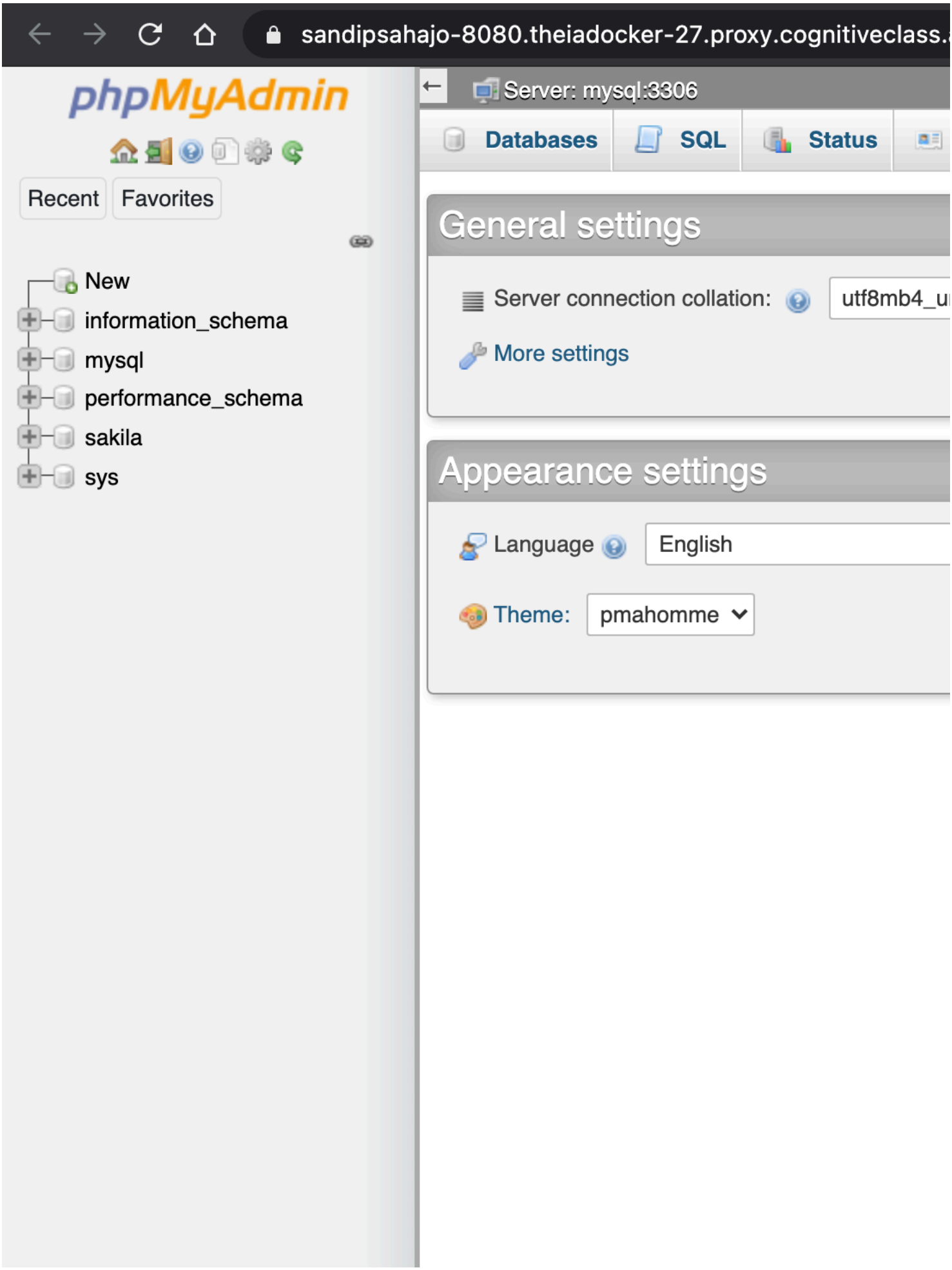
To start the MySQL, click `Start`.


MySQL
MySQL
INACTIVE
8.0.22
5.0.4
2.0.2
Connect to MySQL and phpMyAdmin directly
Start
Summary
Connection Information
Get started with MySQL in a faster, easier wa

2. Once MySQL has started, click on `phpMyAdmin` button to open phpMyAdmin in the same window.

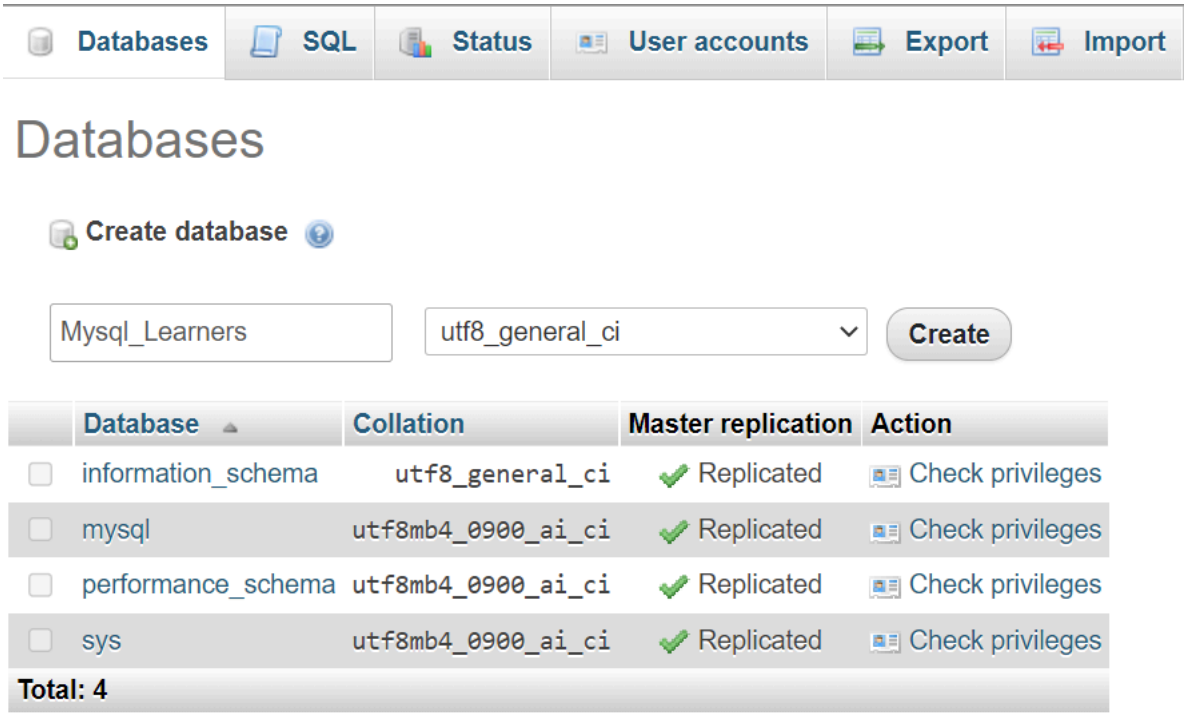


3. You will see the **phpMyAdmin GUI** tool.


sandipsahajo-8080.theiadocker-27.proxy.cognitiveclass.
phpMyAdmin
Recent
Favorites
New
information_schema
mysql
performance_schema
sakila
sys
Server: mysql:3306
Databases
SQL
Status
General settings
Server connection collation:
utf8mb4_u
More settings
Appearance settings
Language
English
Theme:
pmahomme

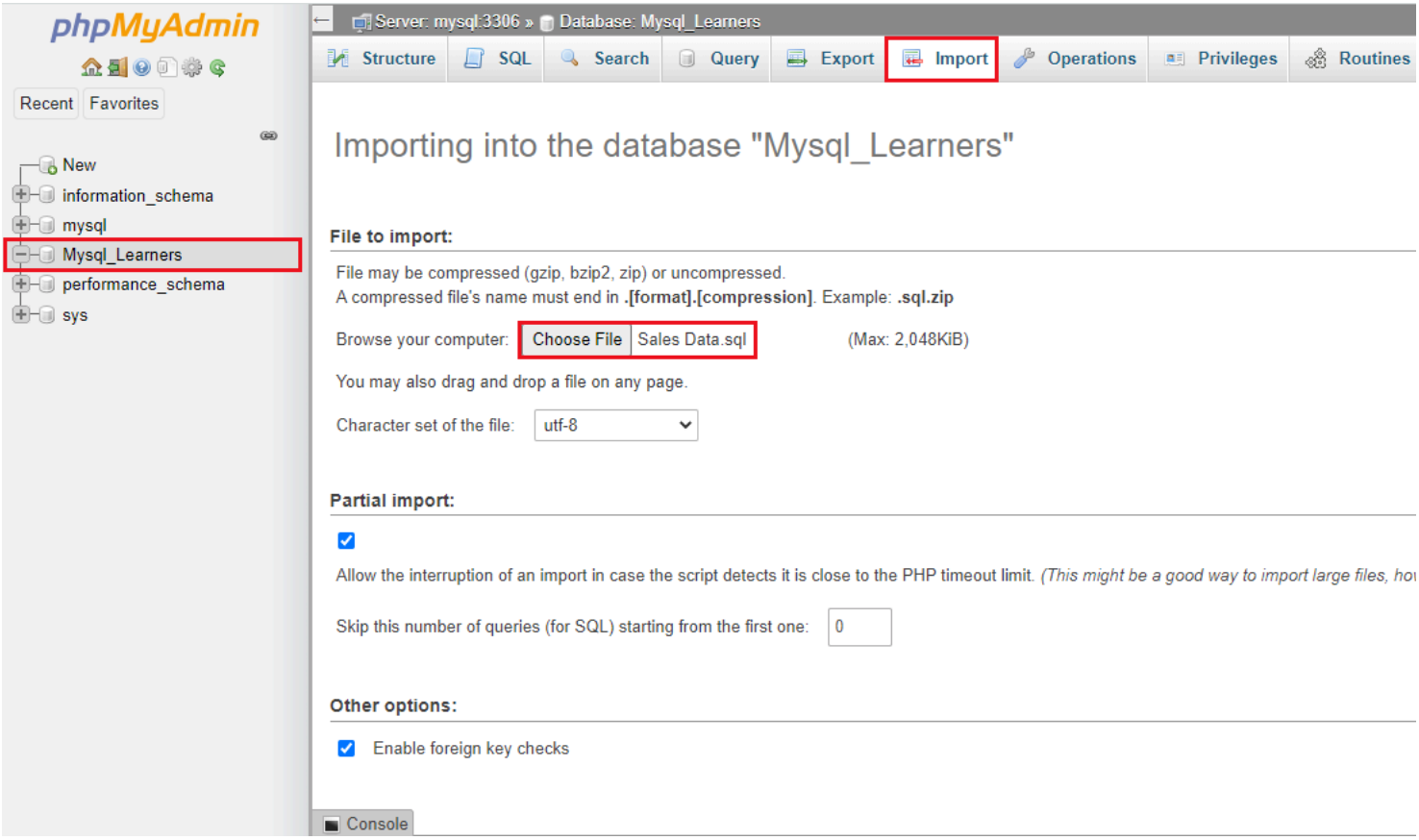
4. In the tree-view, click `New` to create a new empty database. Then enter `Mysql_Learners` or any other name you desire, as the name of the database and select utf8_general_ci and click `Create`.

   UTF-8 is the most commonly used character encoding for content or data.



TASK B: Create and load tables using sql files.

1. Download the 3 sql files below to your local computer:
   - Employee Data.sql
   - Salary Data.sql
   - Sales Data.sql
2. To load each sql file do the following steps.
   - Select your database, in the case of the example shown if it is **Mysql_Learners** and click on **Import** tab.
   - Click on `Choose File`. Browse for the file and upload it .
   - Later scroll down and click the `Go` button.



- Once the scripts are loaded, you will get a message that, it is imported successfully.

Import has been successfully finished, 3 queries executed. (Sales Data.sql)

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0366 seconds.)

```
CREATE TABLE `Sales` ( `SalesID` varchar(10) DEFAULT NULL, `EmpID` v
varchar(10) DEFAULT NULL, `Units_Sold` double DEFAULT NULL, `Sale_Pr
NULL, `Profit` double DEFAULT NULL, `Date` varchar(10) DEFAULT NULL
```

24 rows inserted. (Query took 0.0068 seconds.)

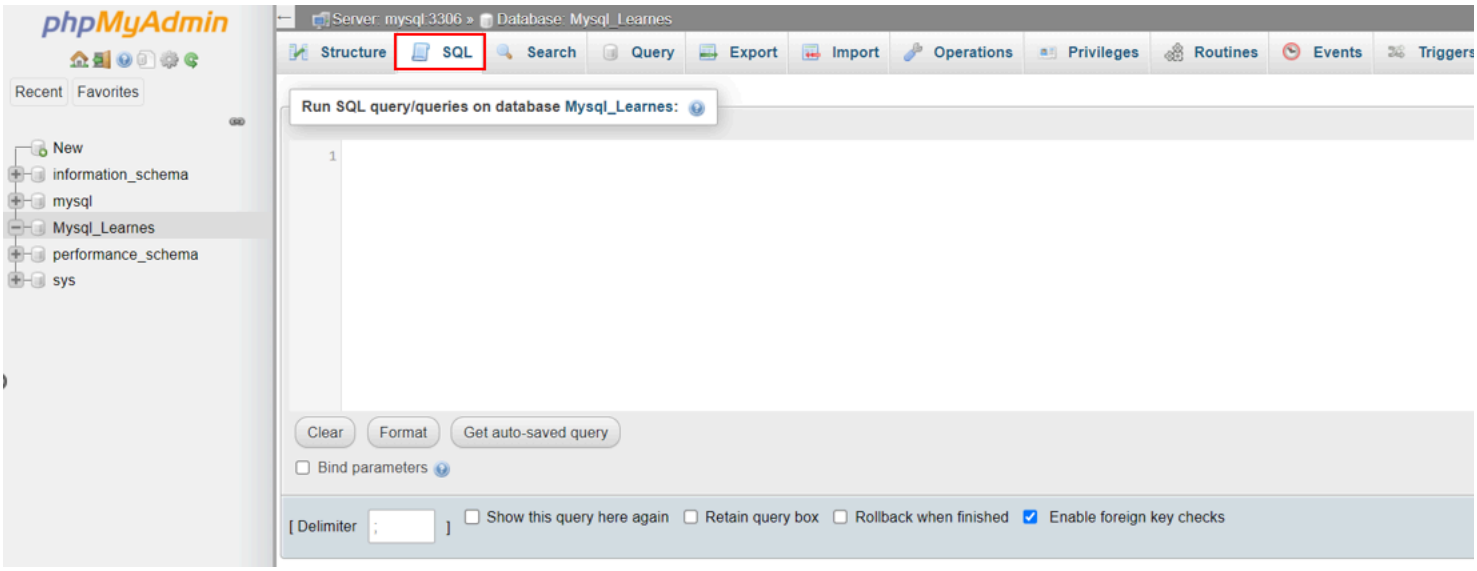
```
INSERT INTO `Sales` (`SalesID`, `EmpID`, `Segment`, `Product`, `Unit
'E04732', 'Government', 'Product2', 252, 20, 5040, 2920, 2120, '04/0
571, '07/24/2021'), ('S2530', 'E03496', 'Midmarket', 'Product2', 211
'Product1', 2133, 7, 14931, 10730, 4201, '09/29/2022'), ('S2512', 'E
('S2513', 'E04732', 'Channel Partners', 'Product1', 1001, 30, 30030,
'Product1', 2513, 12, 30156, 7554, 22602, '06/21/2022'), ('S2514', '
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
COMMIT
```

You can import the other sql files in the same way.

3. To run the SQL queries you need to copy the given codes and paste it to the text area of the SQL page and click on Go.



Data Engineering

In this section you will perform data cleansing (removing duplicates) and data transformation (change column name) operations on the data.

1. Identify the duplicate entry for employees in the employee table using GROUP BY and HAVING statements.

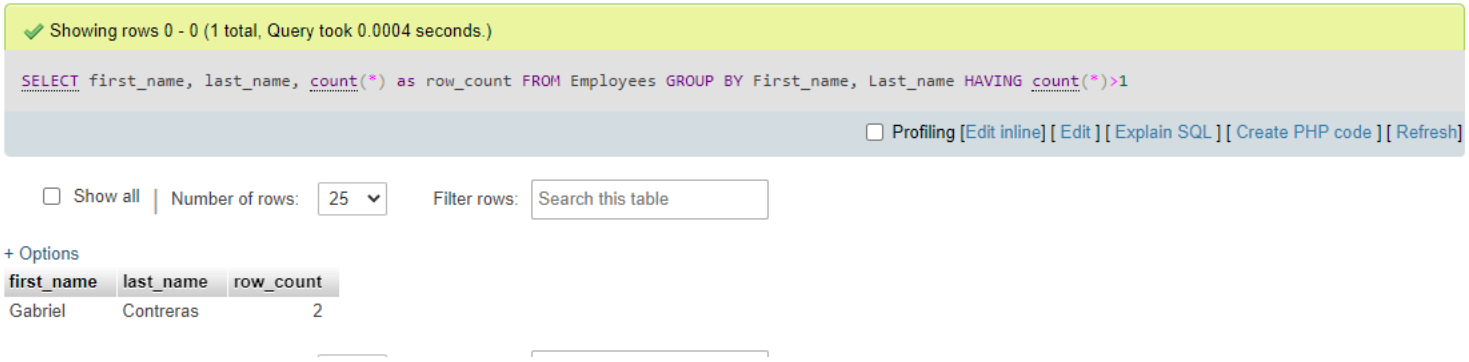
▼ Solution syntax

```
1
2
3
4
```

```
SELECT first_name, last_name, count(*) as row_count
    FROM Employees
    GROUP BY First_name, Last_name
    HAVING count(*)>1;
```

Copied!

▼ Output



2. Select the duplicate entry for employees and delete the row with the higher EMPID.

▼ Solution syntax

```
1
2
3
4
5
6
```

```
    CREATE TABLE my_cte(Emp_ID VARCHAR(100), First_Name VARCHAR(100),
    Last_Name VARCHAR(100), Row_Num INT) as
    (SELECT Emp_ID, First_Name,Last_Name,ROW_NUMBER() Over (PARTITION BY
    First_Name,Last_Name ORDER BY Emp_ID )as Row_Num FROM Employees);

    SELECT * FROM my_cte WHERE Row_Num > 1;
```

Copied!

Once the duplicate entry is displayed on screen, delete the row using the EMP_ID

```
1
```

```
    delete from Employees where EMP_ID = "E04713";
```

Copied!

▼ Output

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0383 seconds.)

```
CREATE TABLE my_cte(Emp_ID VARCHAR(100), First_Name VARCHAR(100), Last_Name VARCHAR(100), Row_Num INT) as (SELECT Emp_ID,
First_Name,Last_Name,ROW_NUMBER() Over (PARTITION BY First_Name,Last_Name ORDER BY Emp_ID )as Row_Num FROM Employees)
```

[Edit inline] [Edit] [Create PHP code]

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

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT * FROM my_cte WHERE Row_Num > 1
```

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

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

+ Options

Emp_ID	First_Name	Last_Name	Row_Num
E04713	Gabriel	Contreras	2

3. Data Transformation – Change the column Salary in the Salary table to “Annual_Income”

▼ Solution syntax

```
1
```

```
Alter Table Salary RENAME COLUMN Salary TO Annual_Income;
```

Copied!

▼ Output

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0142 seconds.)

```
Alter Table Salary RENAME COLUMN Salary TO Annual_Income
```

[Edit inline] [Edit] [Create PHP code

SQL JOINS, Aggregations

Use SQL JOINS, Aggregations where needed, to derive metrics from the database tables.

1. Using the tables given, find out the Total number of men and women employees in the company who are aged below 50 yrs.

▼ Solution syntax

```
1
```

```
SELECT Gender, count(*) as "Total Employees" From Employees Where age <= 50 Group by Gender;
```

Copied!

▼ Output

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

```
SELECT Gender, count(*) as "Total Employees" From Employees Where age <= 50 Group by Gender
```

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

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

+ Options

Gender	Total Employees
Female	9
Male	13

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

2. Using the tables, find the employees whose salary is greater than $150000.

   Note – Rename column Annual_Income back to Salary in the table Salary

▼ Solution syntax

```
1
2
3
4
```

```
SELECT first_name, last_name, gender, country, salary
From Employees e inner join Salary s
On e.Emp_ID = s.Emp_ID
Where salary>150000
```

Copied!

▼ Output

Showing rows 0 - 11 (12 total, Query took 0.0008 seconds.)

```
#SELECT Gender, count(*) as "Total Employees" From Employees Where age <= 50 Group by Gender; SELECT first_name, last_name, gender, country, salary
From Employees e inner join Salary s On e.Emp_ID = s.Emp_ID Where salary>150000
```

[Edit inline] [Edit] [Create PHP code]

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

+ Options

first_name	last_name	gender	country	salary
Luna	Jackson	Female	United Sta	163099
Joshua	Juarez	Male	Brazil	172787
Sadie	Coleman	Female	United Sta	157333
Leilani	Anthony	Female	United Sta	249270
Rylee	Yu	Female	United Sta	175837
Piper	Watson	Female	United Sta	154828
Zoey	Benjamin	Female	United Sta	151703
Jackson	Martin	Male	United Sta	256420
John	Moore	Male	United Sta	199808
Christian	Sanders	Male	United Sta	236946
Thomas	Padilla	Male	Brazil	206624
Lincoln	Reed	Male	United Sta	180664

SQL GROUP BY, HAVING

Use SQL GROUP BY and HAVING statements to get some count metrics from database tables.

1. Display products grouped by segments with total Sales greater than $100,000.

▼ Solution syntax

```
1
2
3
```

```
SELECT product, segment, sum(sales) as TotalSales  From Sales
Group by segment, product
Having sum(sales)>100000
```

Copied!

▼ Output

Showing rows 0 - 7 (8 total, Query took 0.0003 seconds.)

```
#SELECT Gender, count(*) as "Total Employees" From Employees Where age <= 50 Group by Gender; SELECT product, segment, sum(sales) as TotalSales From Sales Group by segment, product Having sum(sales)>100000
```

[Edit inline] [Edit] [Create PHP code]

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

+ Options

product	segment	TotalSales
Product2	Government	135060
Product4	Midmarket	467785
Product2	Midmarket	587901
Product1	Channel Pa	1527643
Product1	Midmarket	1503631
Product3	Government	650194
Product3	Channel Pa	1312280
Product4	Government	551584

Formatted Output

Show output result ordered in a certain way (Use window functions row_number() or rank() and order by statements).

1. Show an output table of Sales generated by employees ordered highest to lowest.

▼ Solution syntax

```
1. 1
2. 2
3. 3
```

```
1. SELECT first_name, last_name, country, segment, product, Sales, ROW_NUMBER() OVER(ORDER BY Sales desc) RowNumber
2. From Employees e inner join Sales s1
3. On e.EMP_ID = s1.EMPID
```

Copied!

▼ Output

Showing rows 0 - 23 (24 total, Query took 0.0008 seconds.)

```
#SELECT Gender, count(*) as "Total Employees" From Employees Where age <= 50 Group by Gender; SELECT first_name, last_name, country, segment, product, Sales, ROW_NUMBER() OVER(ORDER BY Sales desc) RowNumber From Employees e inner join Sales s1 On e.EMP_ID = s1.EMPID
```

[Edit inline] [Edit] [Create PHP code]

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

+ Options

first_name	last_name	country	segment	product	Sales	RowNumber
Adrian	Ruiz	Brazil	Midmarket	Product1	1133595	1
Zoey	Benjamin	United Sta	Channel Pa	Product3	1010940	2
Luna	Jackson	United Sta	Channel Pa	Product1	983430	3
Ethan	Simmons	United Sta	Midmarket	Product2	579250	4
Adrian	Ruiz	Brazil	Government	Product4	551584	5
Daniel	Perry	United Sta	Government	Product3	539116	6
Daniel	Perry	United Sta	Channel Pa	Product1	469096	7
Luna	Jackson	United Sta	Midmarket	Product4	461724	8
Zoey	Benjamin	United Sta	Channel Pa	Product3	301340	9
Luna	Jackson	United Sta	Midmarket	Product1	292636	10
Zoey	Benjamin	United Sta	Government	Product2	95340	11
Daniel	Perry	United Sta	Government	Product1	72384	12

2. Show an output table of Sales(ordered highest to lowest) generated by employees in different segments and rank them for each employee.

▼ Solution syntax

```
1. 1
2. 2
3. 3
4. 4
```

```
1.  SELECT First_Name, last_name, product, segment, Sales, RANK() OVER(PARTITION BY last_name ORDER BY Sales desc) sales_rank
```

```
2. From Employees e inner join Sales s1
3. On e.EMP_ID = s1.EMPID
4. ORDER BY last_name, sales_rank
```

Copied!

▼ Output

Showing rows 0 - 23 (24 total, Query took 0.0008 seconds.)

```
#SELECT Gender, count(*) as "Total Employees" From Employees Where age <= 50 Group by Gender; SELECT First_Name, last_name, product, segment, Sales, RANK() OVER(PARTITION BY last_name ORDER BY Sales desc) sales_rank From Employees e inner join Sales s1 On e.EMP_ID = s1.EMPID ORDER BY last_name, sales_rank
```

[Edit inline] [Edit] [Create PHP code]

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

+ Options

First_Name	last_name ▲ 1	product	segment	Sales	sales_rank ▲ 2
Zoey	Benjamin	Product3	Channel Pa	1010940	1
Zoey	Benjamin	Product3	Channel Pa	301340	2
Zoey	Benjamin	Product2	Government	95340	3
Zoey	Benjamin	Product1	Midmarket	53400	4
Zoey	Benjamin	Product1	Channel Pa	30030	5
Zoey	Benjamin	Product4	Midmarket	6061	6
Zoey	Benjamin	Product2	Government	5040	7
Luna	Jackson	Product1	Channel Pa	983430	1
Luna	Jackson	Product4	Midmarket	461724	2
Luna	Jackson	Product1	Midmarket	292636	3
Luna	Jackson	Product1	Midmarket	24000	4
Luna	Jackson	Product1	Channel Pa	14931	5

Console

Explain the following:

▼ When you would use COALESCE function
The COALESCE function takes a set of inputs and returns the first non-null value.
▼ What is the difference between Union and Union
UNION and UNION ALL are SQL operators used to concatenate 2 or more result sets. UNION: only keeps unique records, UNION ALL: keeps all records, including duplicates
▼ What is the difference between clustered and non-clustered indexes
Clustered indexes can be read rapidly rather than non-clustered indexes. Clustered index requires less memory for operations. The size of clustered index is large. Size of non-clustered index is comparatively smaller but requires more memory for operations.

Clustered index store pointers to block not data. Non-Clustered index store both value and a pointer to actual row that holds data. In non-clustered index, index contains the pointer to data.

Author

Roopa Raghavan

Changelog

Date	Version	Changed by	Change Description
13-Dec-2022	1.0	Lavanya	Initial version created
05-Jun-2024	1.1	Anita Verma	Updated instruction and screenshsot