

LAB 11 CREATE A SIMPLE DATABASE WITH TWO TABLES

OBJECTIVES:

- Create a database with two tables
- Add user access to the created database
- Use a Web Stack with a GUI database administration tool
- Use the command line to connect to a database

BASIC CONCEPTS FOR CREATING DATABASES

In this lab, we will work with the three language subsets of SQL:

- Data Definition Language (DDL),
- Data Control Language (DCL), and
- Data Manipulation Language (DML).

We will be using commands from the DDL and DCL subsets. With the DDL we will create a simple database with two tables. We will control access to our database using the DCL. The tool that we will use is called [phpMyAdmin](#), part of the XAMPP install package.

[phpMyAdmin](#) is a DDL, DCL, and DML GUI interface used to create and manage MySQL/Maria databases. Another popular GUI interface for creating and managing MySQL/Maria database is [MySQL Workbench](#).

Remember behind the GUIs are the actual SQL statements. These statements are the same statements we would use in our code when creating applications that work with databases or working at the command line without a GUI interface. We will work with the [mydatabase](#) database using the Data Manipulation Language (DML) at the command line and in our code in later labs.

This lab creates a database named [mydatabase](#) with two tables ([personnel](#) & [timesheet](#)) and adds a user and password to access the [mydatabase](#) database. This database is used in the upcoming labs, so follow the instructions closely to make sure your database is setup correctly.

LAB TASK CHECKLIST

Complete the following tasks for this lab:

1. Review the Lesson (Assignment, Reading and Additional materials)
2. Create a Database called **mydatabase**
3. Add the **cti110** user with a password of **wtcc** for **mydatabase**
4. Create the following tables (see Database Requirements below):
 - a. **personnel**

- i. empID, firstName, lastName, jobTitle, hourlyWage
 - ii. populate the table and add data to columns
 - b. **timesheet**
 - i. empID, hoursWorked
 - ii. populate the table and add data to columns
5. Submit Assignment files:
 - a. *lastname_mydatabase.sql*
 - b. *lastname_designer*
 - c. *lastname_login*

where *lastname* is your last name.

DATABASE REQUIREMENTS:

Database Name: mydatabase

Tables:

- personnel
- timesheet

User Account:

- Username: *cti110*
- Password: *wtcc*

Table Structure:

personnel:

Column	Data Type	Null	Links to
empID	int (11)	No	
firstName	varchar (64)	No	
lastName	varchar (64)	No	
jobTitle	varchar (64)	No	
hourlyWage	float	No	

timesheet:

Column	Data Type	Null	Links to
empID	int (11)	No	->personnel. EmpID
hoursWorked	int (11)	No	

Table Data:

personnel records (empID, firstName, lastName, jobTitle, hourlyWage)

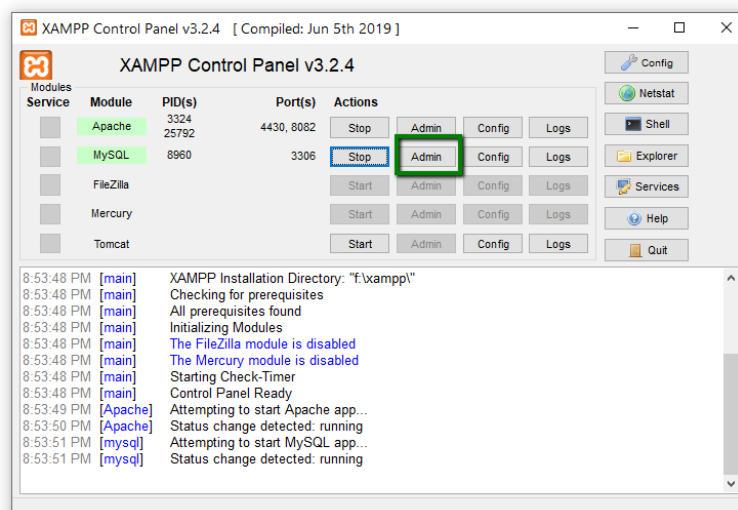
- 12345, Chris, Smith, Sales, 12.55
- 12347, Mary, Peters, Sales, 12.55
- 12348, Mike, Jones, Manager, 24.15
- 12353, Anne, Humphries, Accountant, 25.45
- 12356, Ann, Jones, Sales, 13.75
- 12357, John, Jackson, Reception, 8.75
- 12358, John, King, Cleaner, 7.75
- 12360, Ken, Stewart, Accountant, 28.55
- 12361, Joan, Smith, Cleaner, 8.25
- 12363, Jesse, Andrews, Sales, 10.75

timesheet records (emplID, hoursWorked)

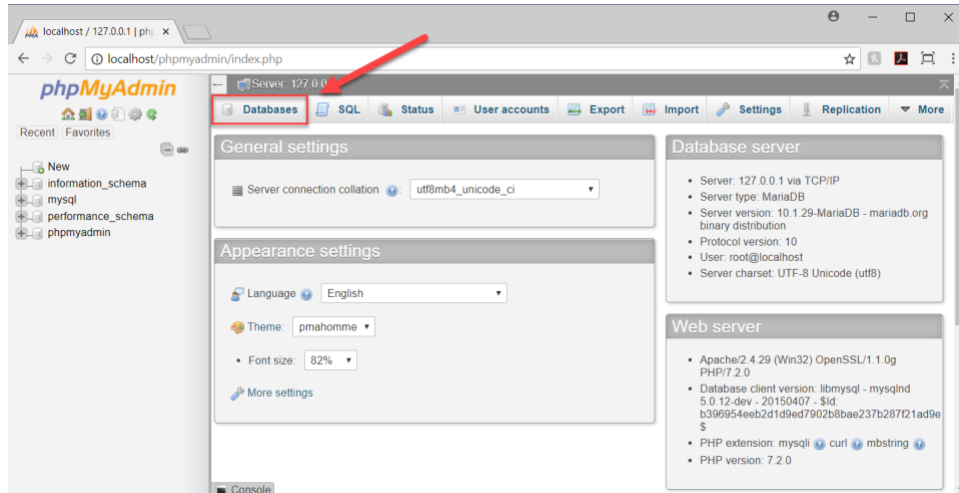
- 12345, 30
- 12347, 35
- 12348, 40
- 12353, 35
- 12356, 20
- 12357, 40
- 12358, 32
- 12360, 20
- 12361, 32
- 12363, 35

Lab Instructions for creation and setup of database

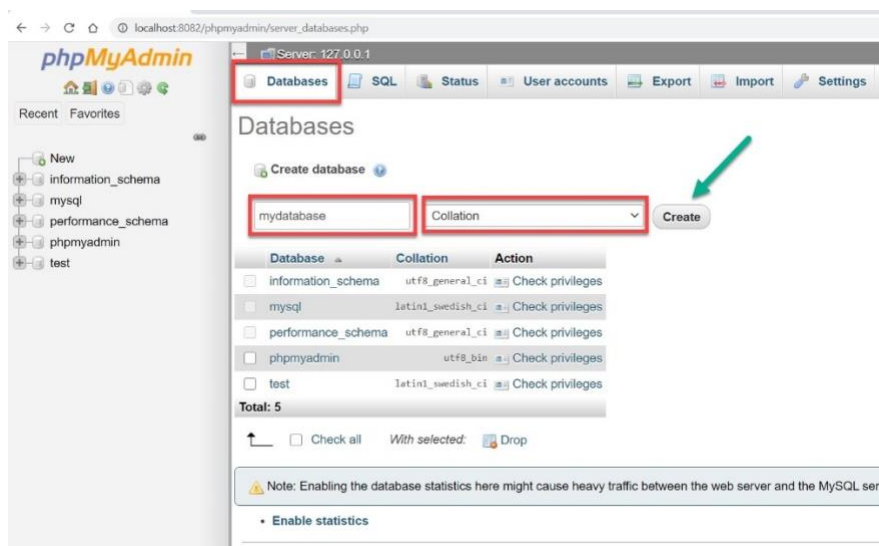
1. Select **Admin** to open Admin Console



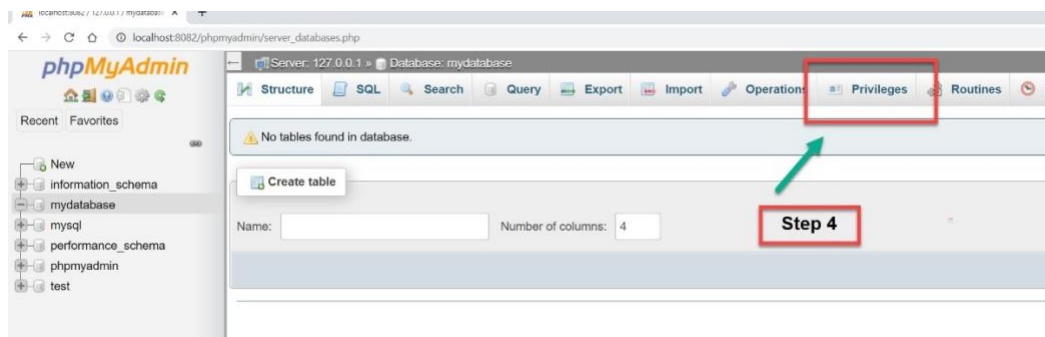
2. Once the Admin Console opens (in your browser), select the **Databases** tab.



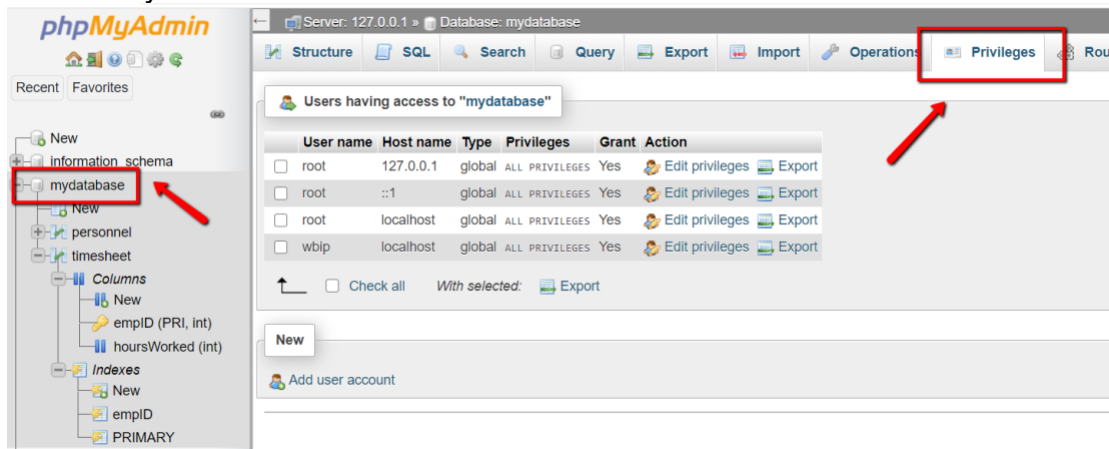
3. Create the **mydatabase** database (fill in the required values and click **Create**)



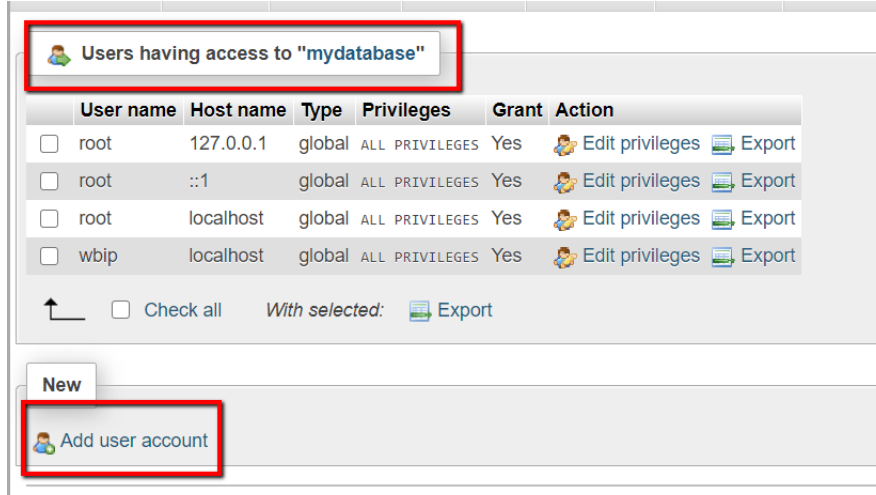
4. Create a new user for the Database **mydatabase**



5. Make sure mydatabase is selected



6. Select Add user account



7. Fill in the required fields

Databases SQL Status User accounts Export Import Settings

Add user account

Login Information

User name: Use text field:

Host name: Local

Password: Use text field: Strength: Very weak

Re-type:

Authentication Plugin: Native MySQL authentication

Generate password:

Database for user account

☐ Create database with same name and grant all privileges.

☐ Grant all privileges on wildcard name (username_%).

☒ Grant all privileges on database mydatabase.

8. Scroll down the screen, set the privileges (**Check All**), then **Go**.

Global privileges ☒ Check all

Note: MySQL privilege names are expressed in English.

Data

- ☒ SELECT
- ☒ INSERT
- ☒ UPDATE
- ☒ DELETE
- ☒ FILE

Structure

- ☒ CREATE
- ☒ ALTER
- ☒ INDEX
- ☒ DROP
- ☒ CREATE TEMPORARY TABLES
- ☒ SHOW VIEW
- ☒ CREATE ROUTINE
- ☒ ALTER ROUTINE
- ☒ EXECUTE
- ☒ CREATE VIEW
- ☒ EVENT
- ☒ TRIGGER

Administration

- ☒ GRANT
- ☒ SUPER
- ☒ PROCESS
- ☒ RELOAD
- ☒ SHUTDOWN
- ☒ SHOW DATABASES
- ☒ LOCK TABLES
- ☒ REFERENCES
- ☒ REPLICATION CLIENT
- ☒ REPLICATION SLAVE
- ☒ CREATE USER

Resource limits

Note: Setting these options to 0 (zero) removes the limit.

MAX QUERIES PER HOUR: 0

MAX UPDATES PER HOUR: 0

MAX CONNECTIONS PER HOUR: 0

MAX USER CONNECTIONS: 0

SSL

☒ REQUIRE NONE

☐ REQUIRE SSL

☐ REQUIRE X509

☐ SPECIFIED

REQUIRE CIPHER:

REQUIRE ISSUER:

REQUIRE SUBJECT:

Go

9. The resulting screen should include a confirmation message indicating the successful addition of your user. Select the User accounts tab to return to the User accounts overview page

You have added a new user

CREATE USER 'cti110'@'localhost' IDENTIFIED VIA mysql_native_password USING '****'; GRANT ALL PRIVILEGES ON *.* TO 'cti110'@'localhost' REQUIRE NONE WITH GRANT OPTION MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0; GRANT ALL PRIVILEGES ON 'mydatabase'.* TO 'cti110'@'localhost';

User accounts

Edit privileges: User account 'cti110'@'localhost' - Database mydatabase

Database-specific privileges ☒ Check all

Note: MySQL privilege names are expressed in English.

Data

Structure

Administration

10. The new user ('cti110') has now been added.

Privileges

Users having access to "mydatabase"

User name	Host name	Type	Privileges	Grant	Action
<input checked="" type="checkbox"/> cti110	localhost	global	ALL PRIVILEGES	Yes	Edit privileges Export
<input type="checkbox"/> root	127.0.0.1	global	database-specific ALL PRIVILEGES	No	Edit privileges Export
<input type="checkbox"/> root	:::1	global	ALL PRIVILEGES	Yes	Edit privileges Export
<input type="checkbox"/> root	localhost	global	ALL PRIVILEGES	Yes	Edit privileges Export
<input type="checkbox"/> wbp	localhost	global	ALL PRIVILEGES	Yes	Edit privileges Export

☐ Check all With selected: Export

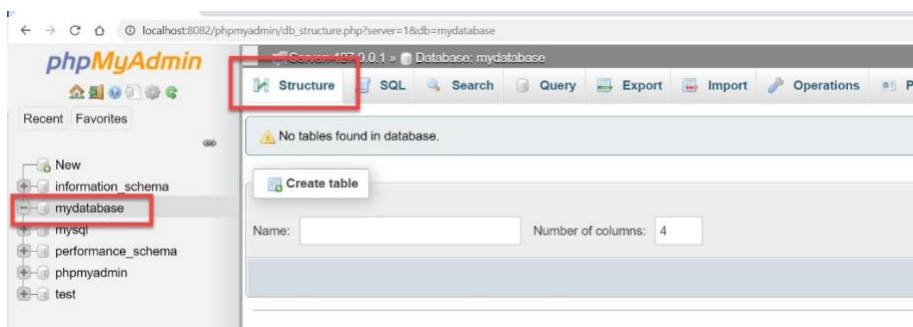
New

Add user account

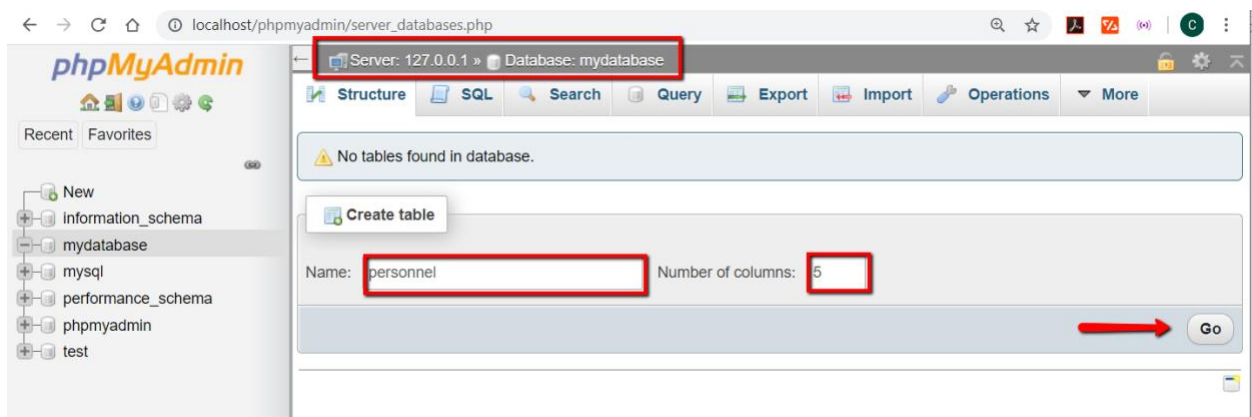
NOTE: If you were creating a user at the “command prompt” or in your code, the Data Access Language (DAL) SQL statement would be similar to this:

```
CREATE USER 'wbip'@'localhost' IDENTIFIED VIA mysql_native_password USING '***';GRANT ALL PRIVILEGES ON *.* TO 'wbip'@'localhost' REQUIRE NONE WITH GRANT OPTION MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;GRANT ALL PRIVILEGES ON `test`.* TO 'wbip'@'localhost';
```

11. Select the mydatabase. You should now be in the Structure Tab.



12. Create the “personnel” table. The “personnel” table contains 5 fields/columns. Enter the table **Name**, **Number of columns** and click **Go**.



13. Define the information required for this new table using the information provided in the Table Structure section. Enter the **Name**, **Type**, and **Length/Values** for each column.

Table name: Add column(s)

Name	Type	Length/Values	Default
<input type="text" value="empId"/> <small>Pick from Central Columns</small>	<input type="text" value="INT"/>	<input type="text" value="11"/>	<input type="text" value="None"/>
<input type="text" value="firstName"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="64"/>	<input type="text" value="None"/>
<input type="text" value="lastName"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="64"/>	<input type="text" value="None"/>
<input type="text" value="jobTitle"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="64"/>	<input type="text" value="None"/>
<input type="text" value="hourlyWage"/> <small>Pick from Central Columns</small>	<input type="text" value="FLOAT"/>	<input type="text" value=""/>	<input type="text" value="None"/>

14. Before leaving this screen, click on the **Index** field associated with **empID** to define it as the table's **PRIMARY** key. A pop-up window will open. Click **Go** and **Save** to complete the process.

Server: 127.0.0.1 » Database: mydatabase » Table: personnel

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	Comments	Virtuality
<input type="text" value="empId"/> <small>Pick from Central Columns</small>	<input type="text" value="INT"/>	<input type="text" value="11"/>	<input type="text" value="None"/>			<input type="checkbox"/>	<input type="text" value="PRIMARY"/>		
<input type="text" value="firstName"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="64"/>				<input type="checkbox"/>			
<input type="text" value="lastName"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="64"/>				<input type="checkbox"/>			
<input type="text" value="jobTitle"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="64"/>				<input type="checkbox"/>			
<input type="text" value="hourlyWage"/> <small>Pick from Central Columns</small>	<input type="text" value="FLOAT"/>					<input type="checkbox"/>			

Add index

Index name:

Index choice:

+ Advanced Options

Column	Size
<input type="text" value="empId [int]"/>	

Table comments:

PARTITION definition:

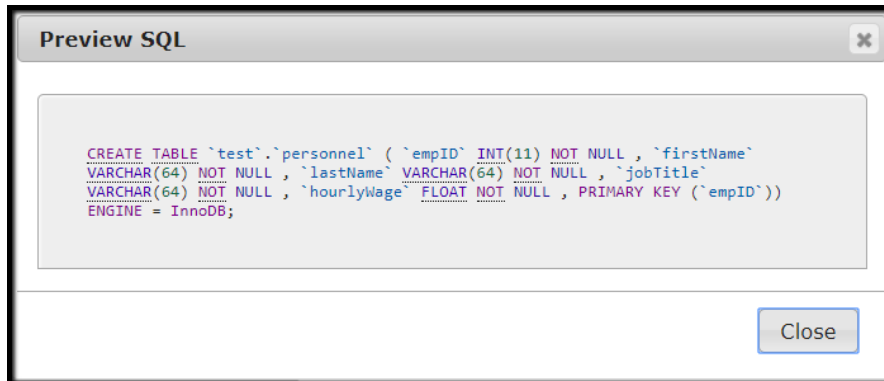
Partitions:

Step 1: Click on the **Index** field for **empId**.

Step 2: Click **Go** in the **Add index** dialog.

Step 3: Click **Save** at the bottom right.

NOTE: If you were creating this table at the “command prompt” or in your code, the Data Definition Language (DDL) SQL statement would be similar to this:



15. The resulting screen shows the action was successful [Server: 127.0.0.1 >> Database: mydatabase >> Table: personnel].

Server: 127.0.0.1 » Database: mydatabase » Table: personnel

Browse Structure SQL Search Insert Export Import Privileges Operations

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	empId	int(11)			No	None			Change Drop More
2	firstName	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More
3	lastName	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More
4	jobTitle	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More
5	hourlyWage	float			No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after hourlyWage Go

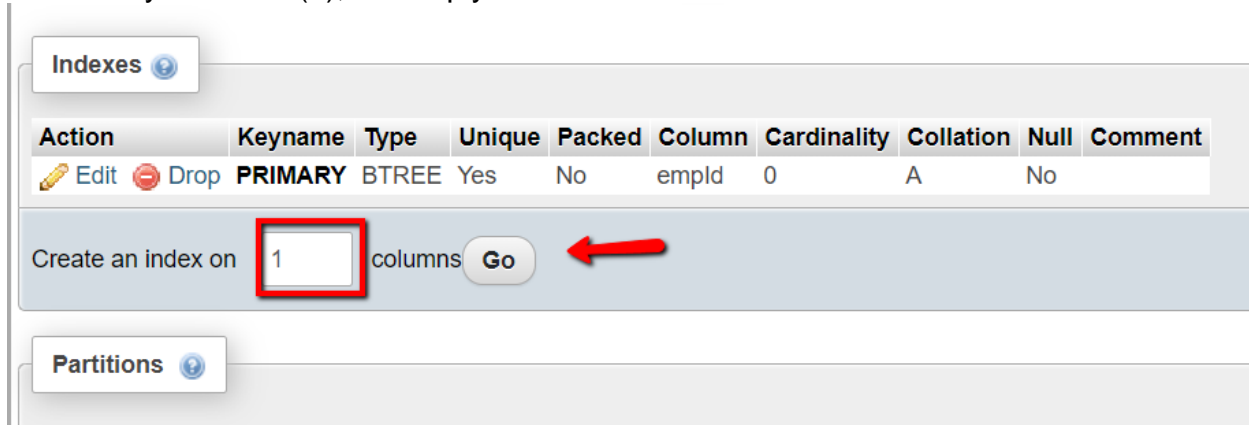
Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	empId	0	A	No	


Create an index on 1 columns Go

Partitions

16. Make **empID** a UNIQUE Key. In the Indexes section, the correct column number is already selected (1), so simply click on **Go**.



Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	empID	0	A	No	

Create an index on 1 columns Go 

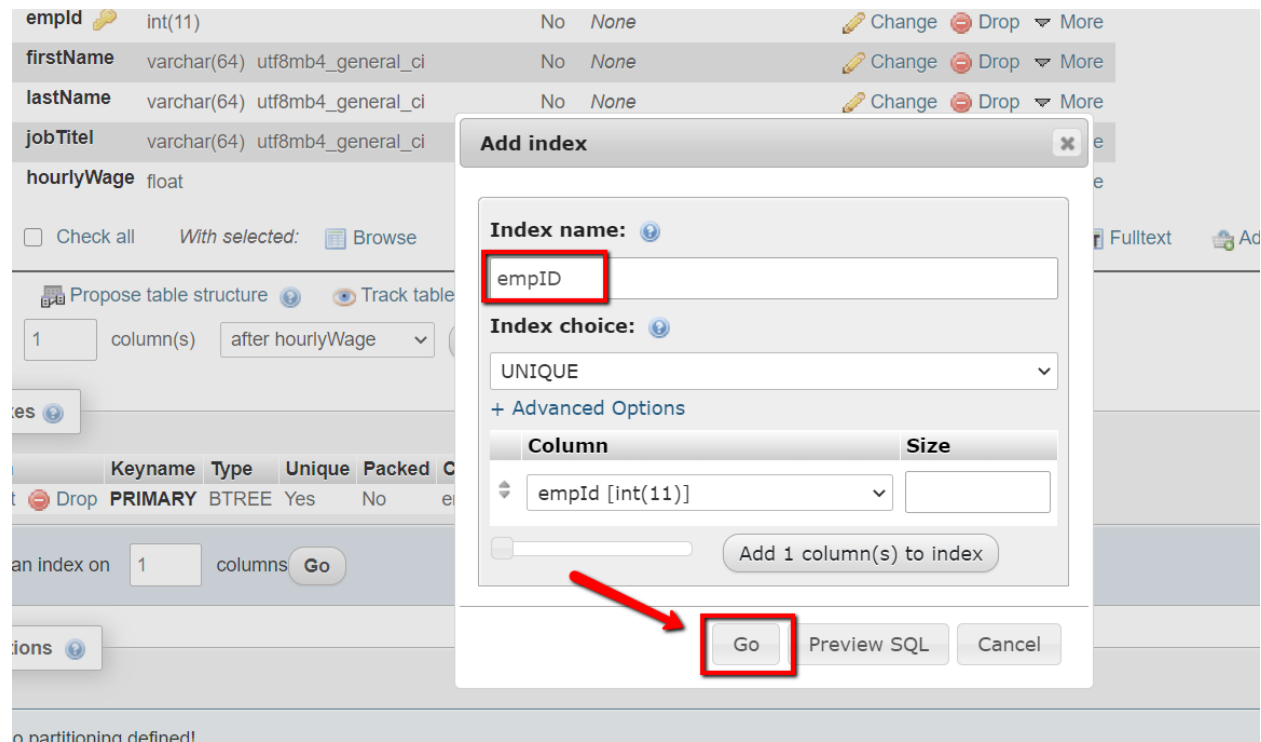
NOTE: If you were modifying this table at the “command prompt” or in your code the Data Definition Language (DDL) SQL statement would be similar to this:



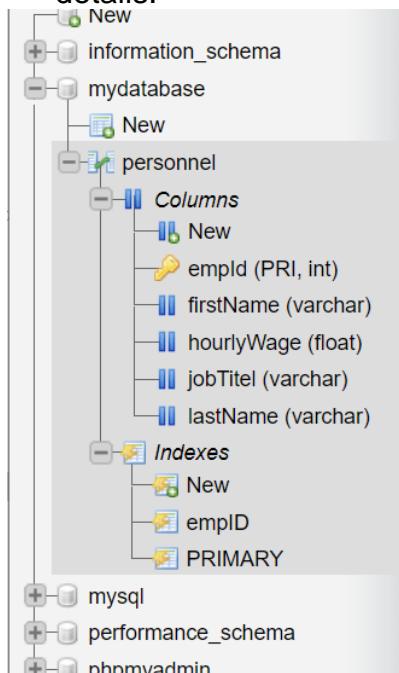
```
ALTER TABLE `test`.`personnel` ADD UNIQUE `empID` (`empID`);
```

Close

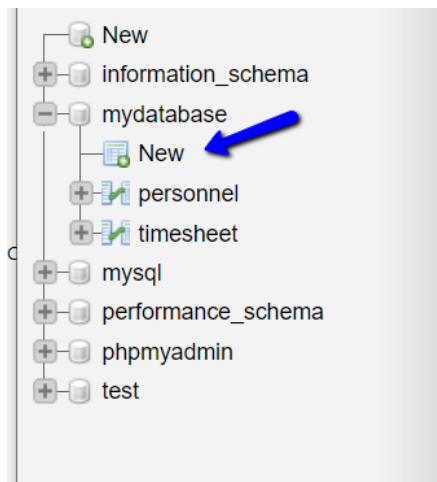
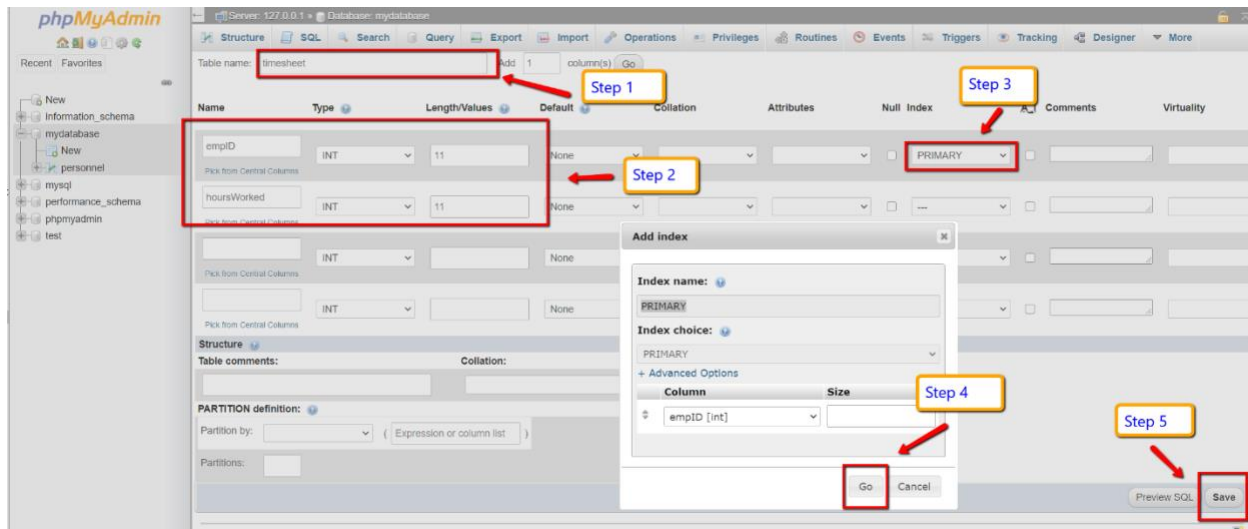
17. Enter the index name (**empID**) info and click **Go**.



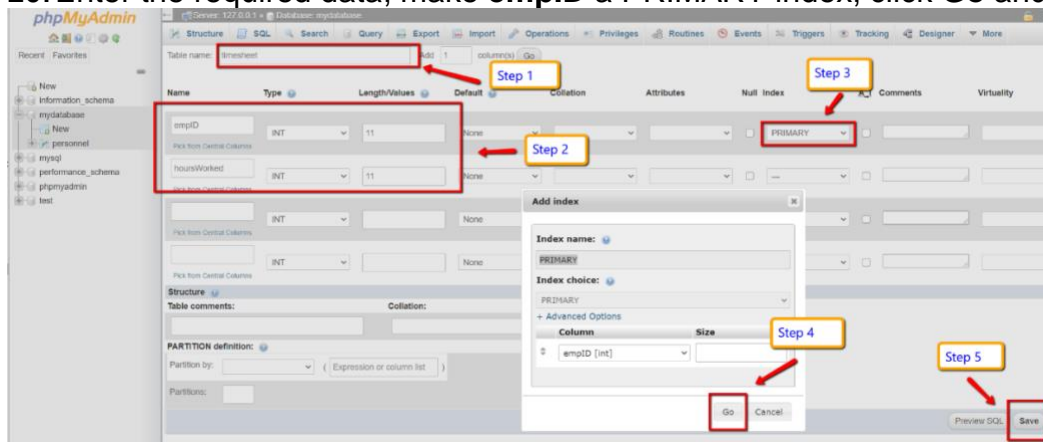
18. Review the details for the **mydatabase** database, and specifically the **personnel** table, in the left-hand navigation pane. Expand all the “+” signs to see the underlying details.



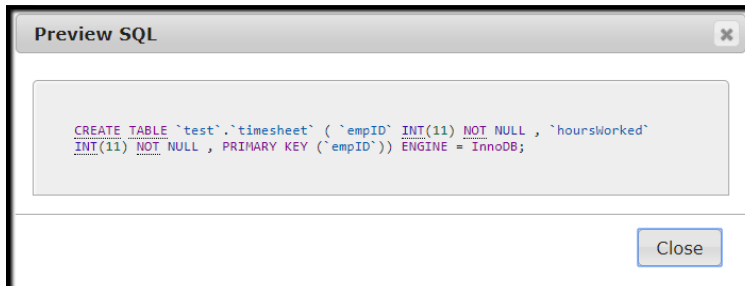
19. Now, create the **timesheet** table by repeating the same steps used to create the **personnel** table. Reference the correct Table Structure information for the column and index details.



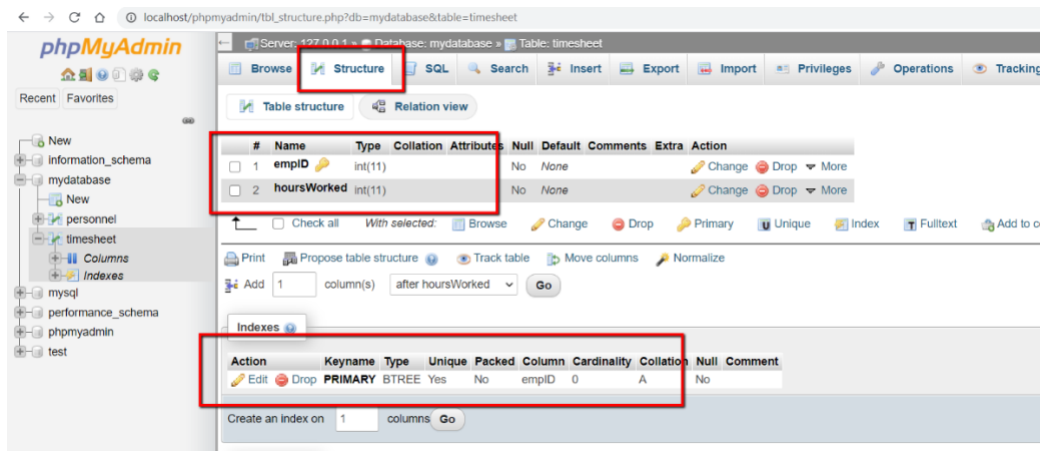
20. Enter the required data, make **empID** a PRIMARY index, click **Go** and **Save**.



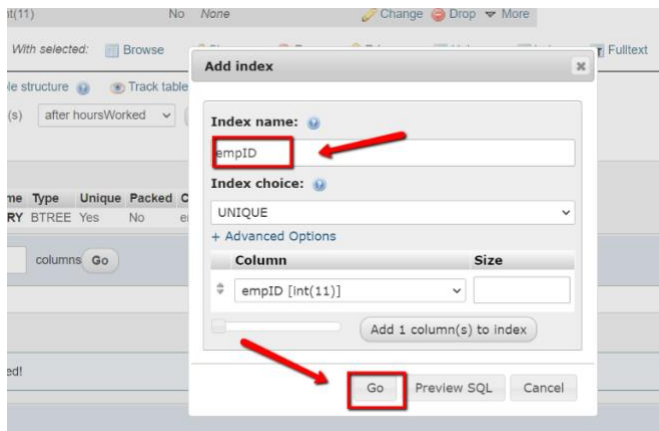
NOTE: If you were creating this table at the “command prompt” or in your code the Data Definition Language (DDL) SQL statement would be like this:



21. The resulting screen shows the action was successful [Server: 127.0.0.1 >> Database: mydatabase >> Table: timesheet].



22. Make **empID** a UNIQUE Key. In the Indexes section, the correct column number is already selected (1), so simply click on **Go**.

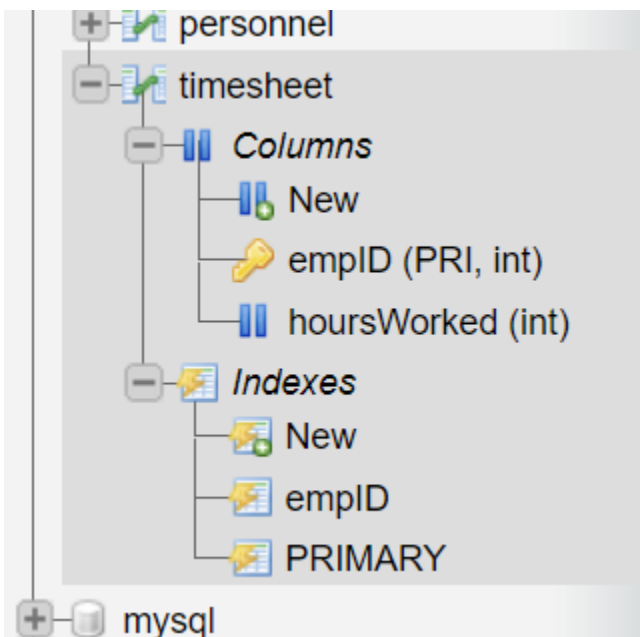


23. Enter the index name (**empID**) info and click **Go**.

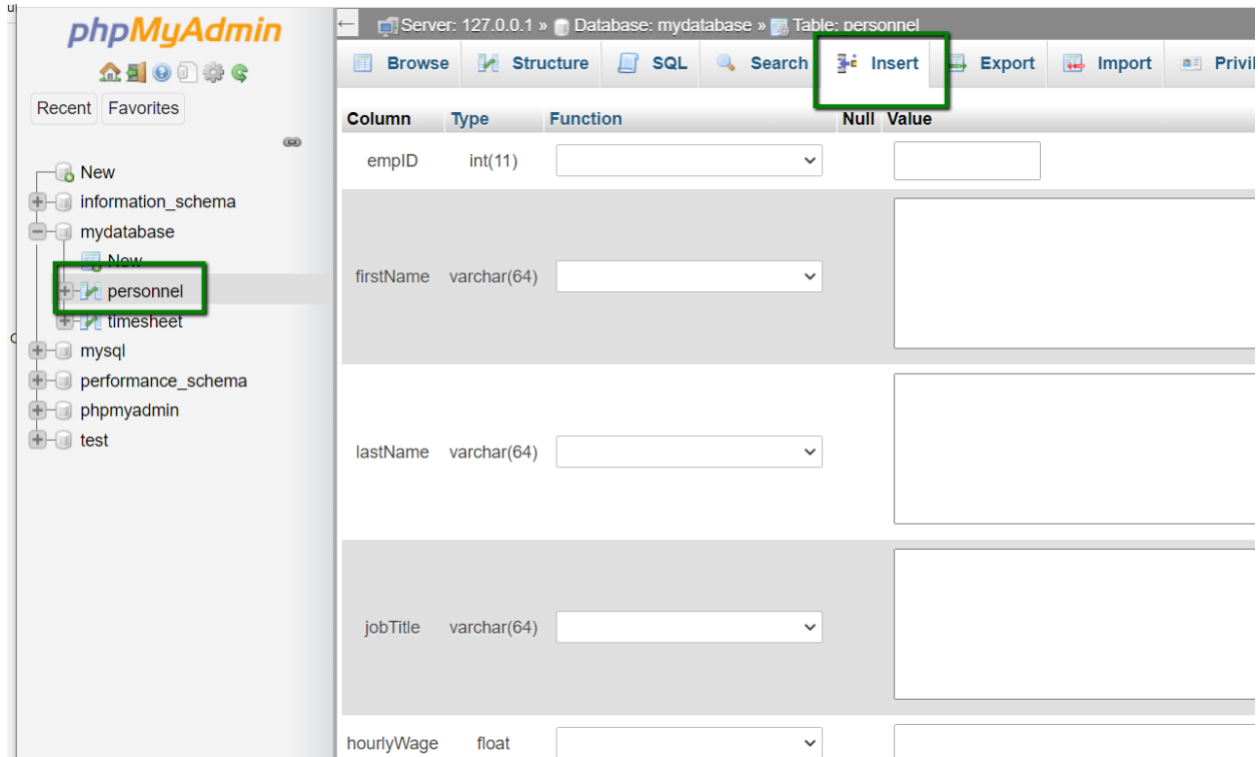
NOTE: If you were modifying this table at the “command prompt” or in your code the Data Definition Language (DDL) SQL statement would be similar to this:



24. Review the details for the **mydatabase** database, and specifically the **timesheet** table, in the left-hand navigation pane. Expand all the “+” signs to see the underlying details.



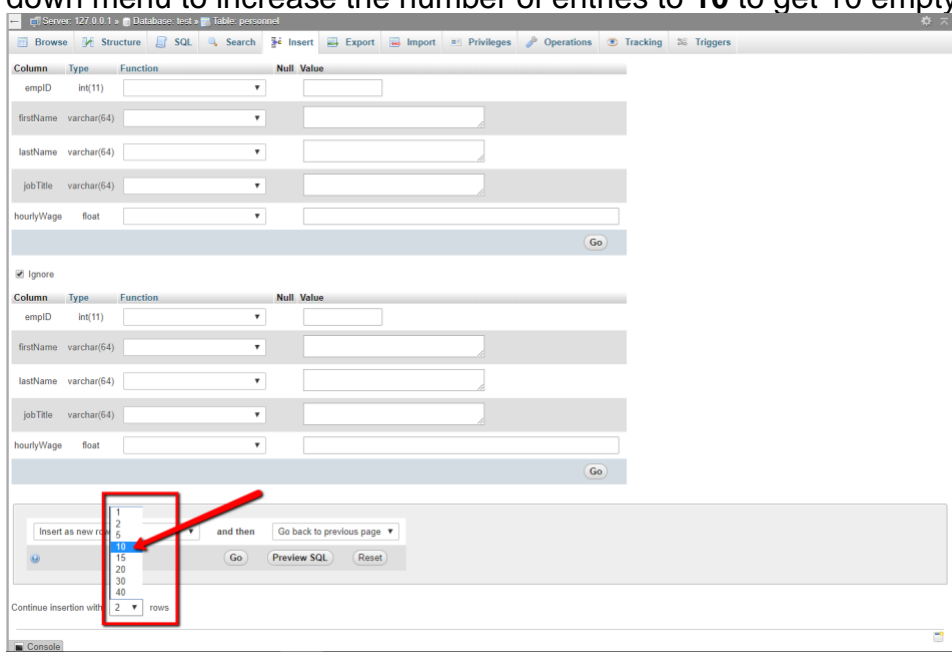
25. Now it's time to add the data. Starting with the **personnel** table ...



The screenshot shows the phpMyAdmin interface. On the left, the database structure is listed, with 'personnel' highlighted under the 'mydatabase' schema. On the right, the 'Table: personnel' structure is displayed. The 'Insert' button in the top navigation bar is highlighted with a green box. The table structure shows columns: empID (int(11)), firstName (varchar(64)), lastName (varchar(64)), jobTitle (varchar(64)), and hourlyWage (float).

Column	Type	Function	Null	Value
empID	int(11)			
firstName	varchar(64)			
lastName	varchar(64)			
jobTitle	varchar(64)			
hourlyWage	float			

26. When the data entry page opens, use the **Continue insertion with # rows** drop-down menu to increase the number of entries to **10** to get 10 empty slots for data.



The screenshot shows the 'Insert' page for the 'personnel' table. The 'Continue insertion with # rows' dropdown menu is highlighted with a red box, and the value '10' is selected. A red arrow points to the '10' option. The page also shows the 'Ignore' checkbox checked and the 'Go' button.

Column	Type	Function	Null	Value
empID	int(11)			
firstName	varchar(64)			
lastName	varchar(64)			
jobTitle	varchar(64)			
hourlyWage	float			

Insert as new row and then Go back to previous page

Continue insertion with 10 rows

27. Enter the personnel data provided in the Table Data section. When all 10 entries are complete, click **Go** to save the data.

Server: 127.0.0.1 » Database: test » Table: personnel

Column Type Function Null Value

empID	int(11)			12345
firstName	varchar(64)			Chris
lastName	varchar(64)			Smith
jobTitle	varchar(64)			Sales
hourlyWage	float			12.55

Go

☐ Ignore

Column Type Function Null Value

empID	int(11)			12347
firstName	varchar(64)			Mary
lastName	varchar(64)			Peters
jobTitle	varchar(64)			Sales
hourlyWage	float			12.55

Go

... more data entries ...

Server: 127.0.0.1 » Database: test » Table: personnel

jobTitle varchar(64) Cleaner

hourlyWage float 8.25

Go

☐ Ignore

Column Type Function Null Value

empID	int(11)			12363
firstName	varchar(64)			Jesse
lastName	varchar(64)			Andrews
jobTitle	varchar(64)			Sales
hourlyWage	float			10.75

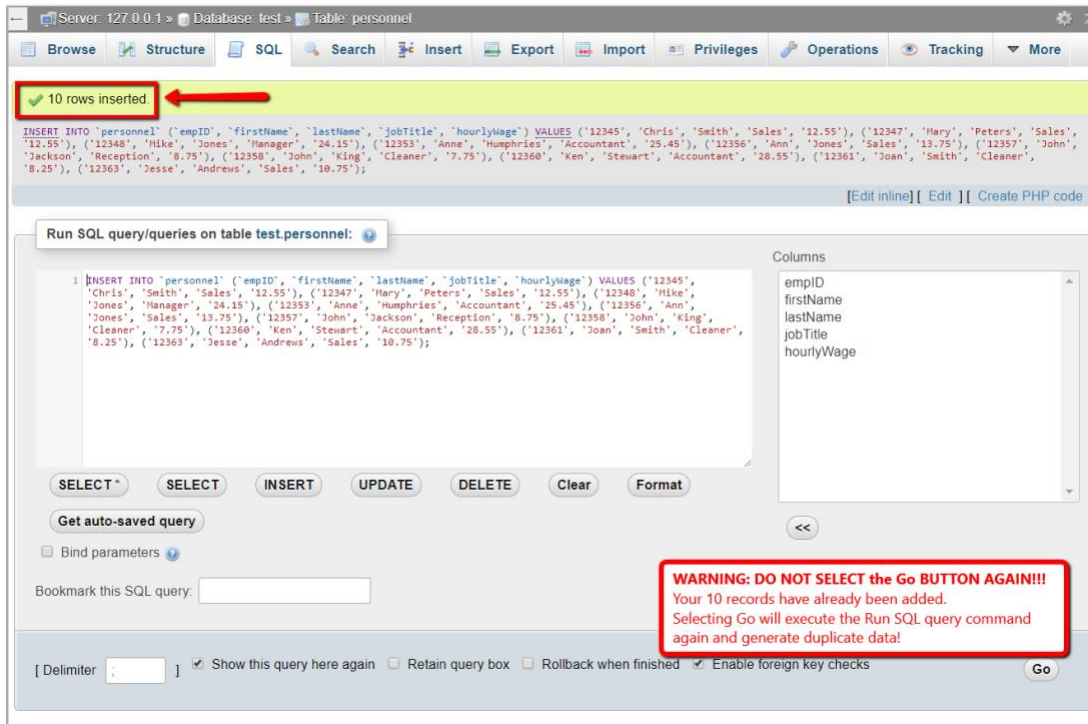
Go

Insert as new row and then Go back to previous page

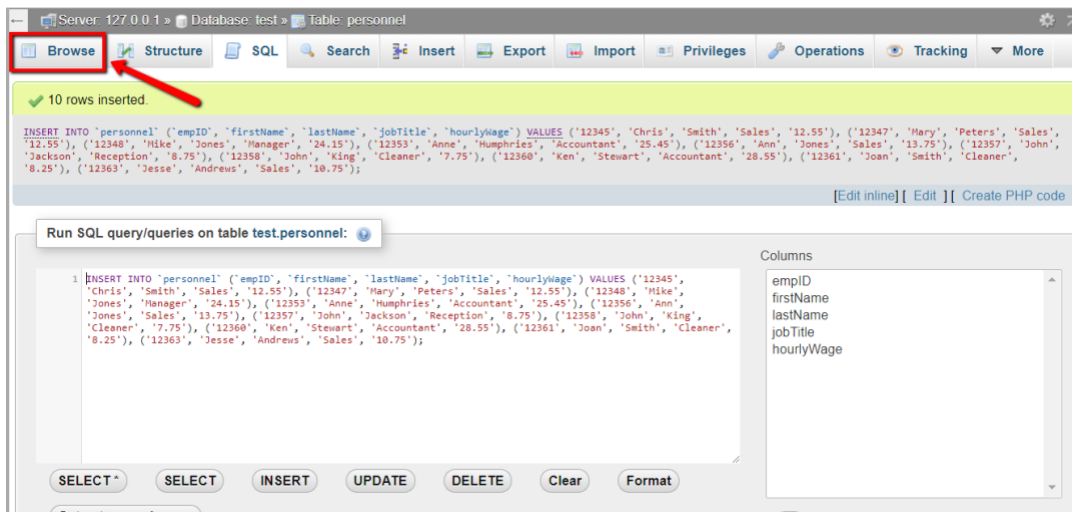
Go Preview SQL Reset

Continue insertion with 10 rows

28. The resulting screen should include a confirmation message indicating the successful addition of 10 rows of data. **DO NOT PRESS GO HERE!**

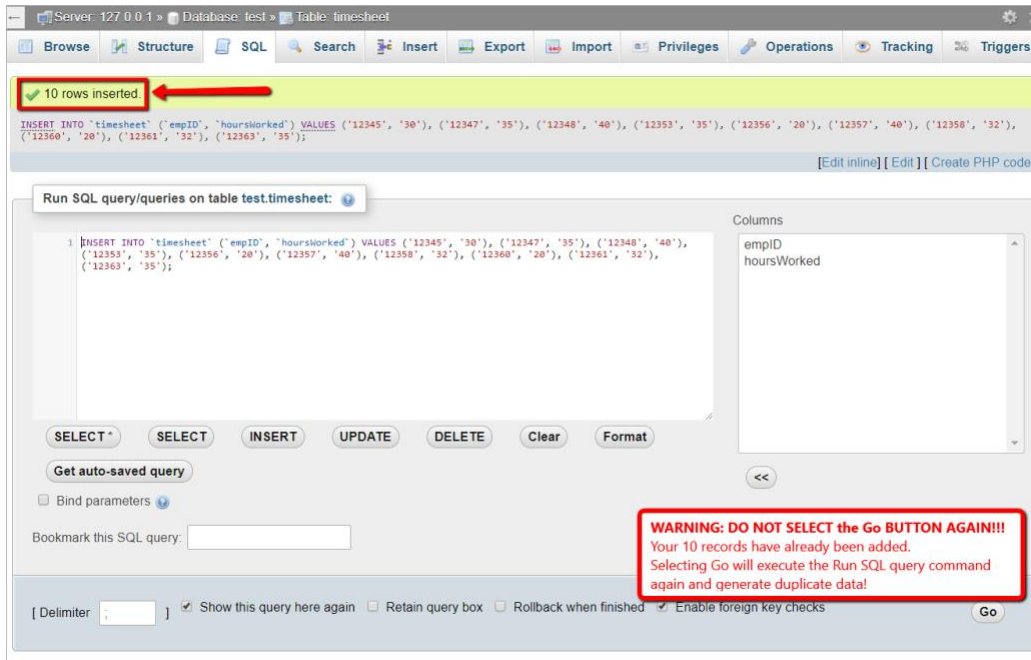


29. Select the Browse tab to see/review the **personnel** data you entered.

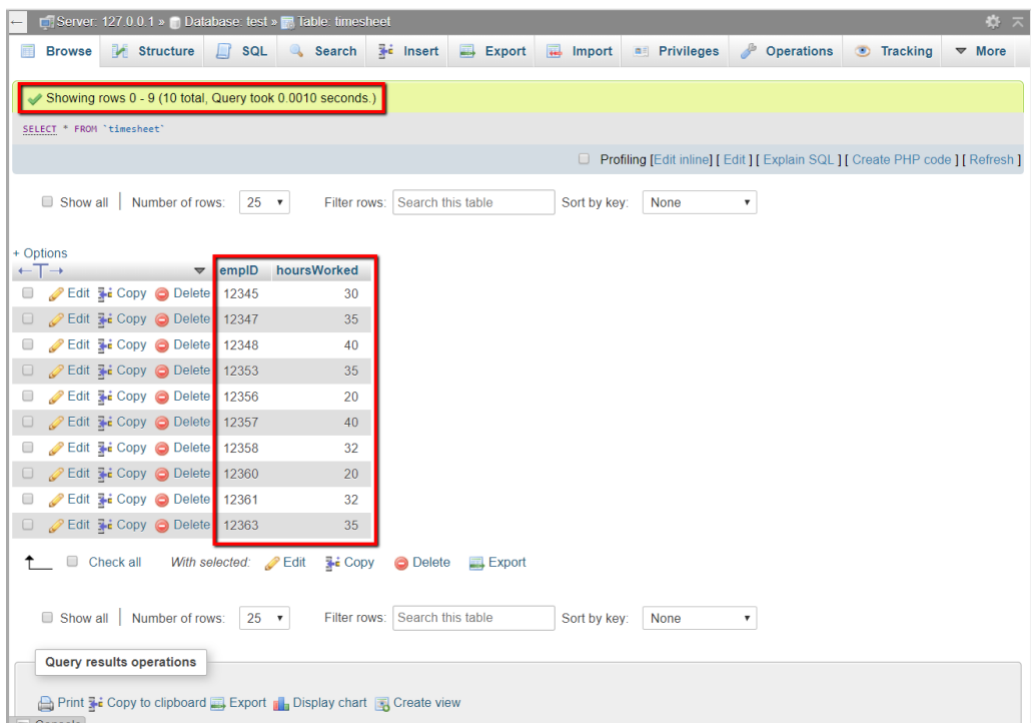


30. Now, add the timesheet data following the same procedure used for personnel. Start by selecting **timesheet** and allow for 10 rows of data.

31. The resulting screen should include a confirmation message indicating the successful addition of 10 rows of data. **REMEMBER: DO NOT PRESS GO HERE!**



32. Select the Browse tab to see/review the **timesheet** data you entered. The data information should look like this:

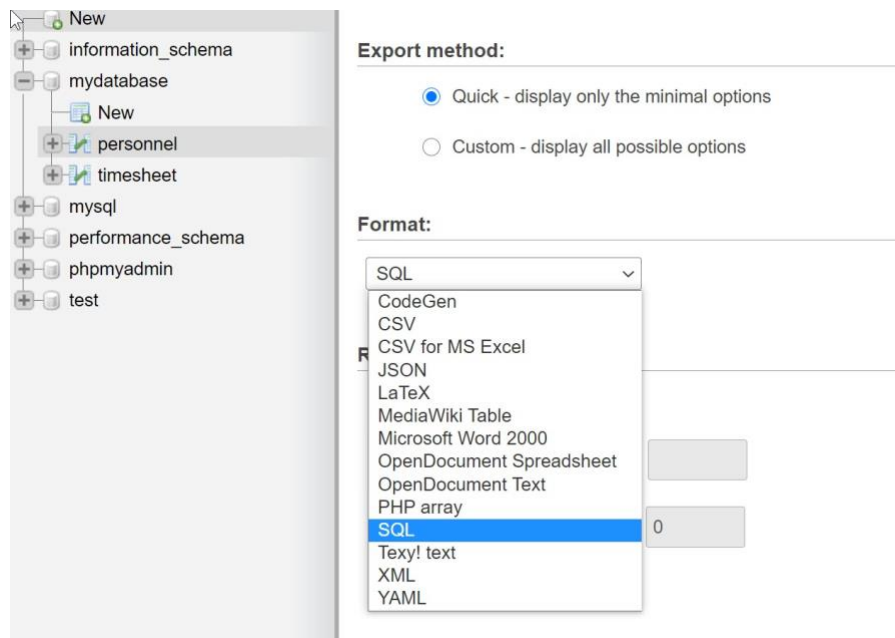


You have now successfully created the database **mydatabase**. This database will be used in future labs, so it is important to create the database, tables and indexes correctly!

SUBMISSIONS REQUIRED:

Creating files for submission:

1. **lastname_12_mydatabase.respective extension (docx or sql)**
 - a. Select the **mydatabase** database in the left navigation area.
 - b. Select the **Export** tab.
 - c. Change the **Format** to SQL (can be used as backup later) or Word and select **Go**.

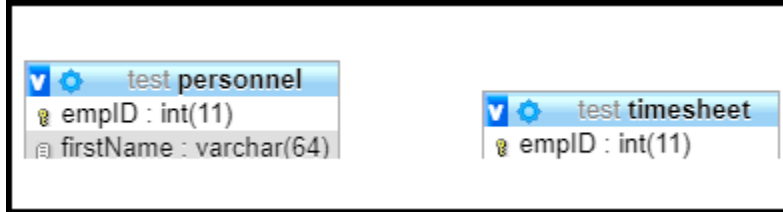


-
-
-
- d. Upload the file in Blackboard.

2. [lastname_12_designer.pdf](#)

- Select the **mydatabase** database in the left navigation area.
- Select the **Designer** tab.
- Take a snapshot and paste it into a Word document.

Note: It should look something like the following:



- Save the file as [lastname_11_designer.pdf](#)
- Upload the file in Blackboard.

3. [lastname_login.docx](#)

- Launch your command line interface.
You may press the Windows logo key plus the 'r' key (i.e. hold down Windows logo key + 'r' key) and type 'cmd'.
- Your Windows command link window pops up.
- Assume your thumb drive is on g: drive, and XAMPP has been installed in folder CTI110.
 - Change your drive from C: to g: drive
 - Change directory by typing **cd cti110\xampp\mysql\bin** as shown in the following:

```
Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\talon>g:

G:\>cd cti110\xampp\mysql\bin

G:\CTI110\xampp\mysql\bin>
```

You can now access MySQL by typing: **mysql -u cti110 -p**

When prompted for a password, enter the password mentioned above (**wtcc**).

- Take a snapshot and paste it into a Word document.
- Note:** It should look something like the following:

```
Command Prompt - mysql -u cti110 -p
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\talon>f:

F:\>cd xampp\mysql\bin

F:\xampp\mysql\bin>mysql -u cti110 -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.11-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

- e. Save the file as [lastname_login.docx](#)
- f. Upload the file in Blackboard

GRADING RUBRIC FOR ASSIGNMENT

Database (mydatabase), Table (personnel), Table (timesheet) setup correctly
Data for personnel and timesheet is setup correctly
Data Type for all Fields set up correctly
Primary and Unique keys set up correctly
User Account set up correctly