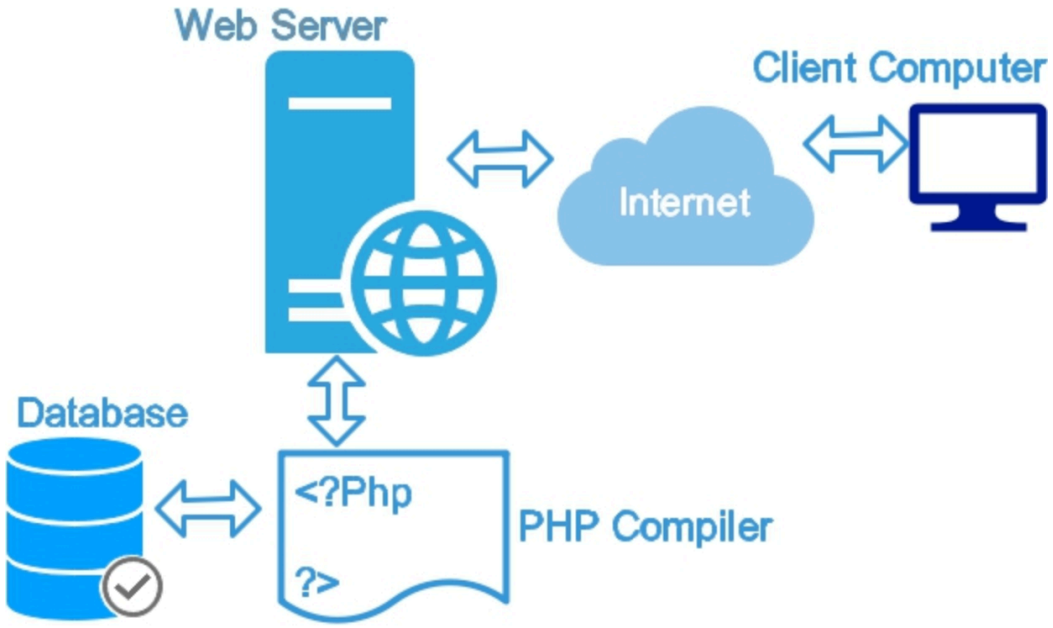
PHP DB's MySQL and SQL - Written by Anthony Adams

When it comes to HTML and PHP, the real fun and beauty about it, is being able to store data in a database and call it when it's needed. This is the basis of why we use it and how all dynamic websites work. This is how all CMS systems like Wordpress, Joomla, PHPBB and so on work. This is how all social media platforms work.

We can scale it down and start small with the basics.



When we work with a database, there is a flow of data. The data can start from the client or the DB and flow through to each other. Let's imagine the case where the data starts at the client side.

-The user interacts with a form and fills out some data in it.

-When the user presses the send or submit button that data gets encrypted in most cases and sent across the internet via the HTTPS protocol (Port 443).

-It arrives at that port on the server and there will be some PHP file (For example "handler.php").

-This file will have some PHP code that will instruct it to connect to some database using a server name, username and password.

- The once that connection is made, the same PHP file will dictate the data flow and tell the form data where to save itself (what table and what columns).

- The data will save in the DB and can now be retrieved. So if the user ever needs to get that data again it can be returned from the DB back up the chain of data flow back to the client.

Please note that this example assumes that the SQL database already exists if not, one will have to be made.

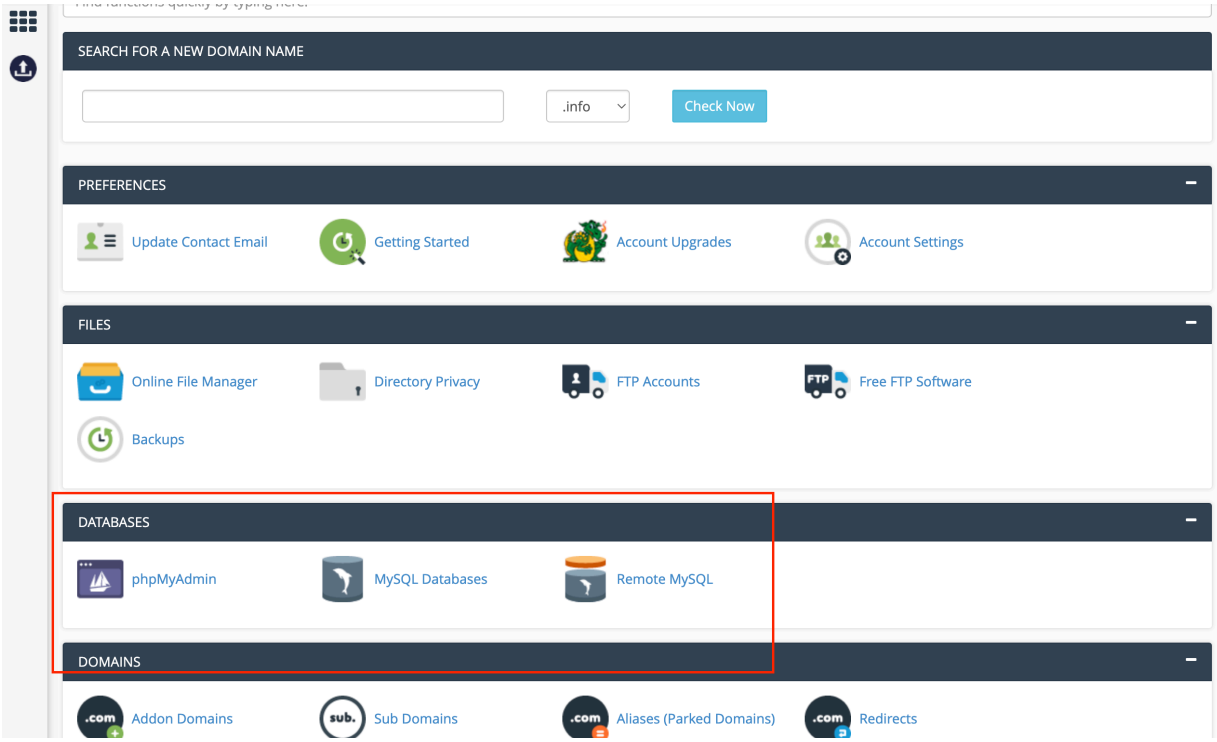
Creating a MySQL database

As previously stated. We must create a DB and the table columns where the data will be saved. To do this (usually) web hosts will use a tool called PHPMyAdmin. This tool is industry standard and even if you don't plan on creating tables from scratch on the regular, you should still learn your way around the interface because if you plan on working with CMS systems, you will need to look in here form time to time.

Also, another tool we will need to get used to is cPanel. This is the back end admin panel of

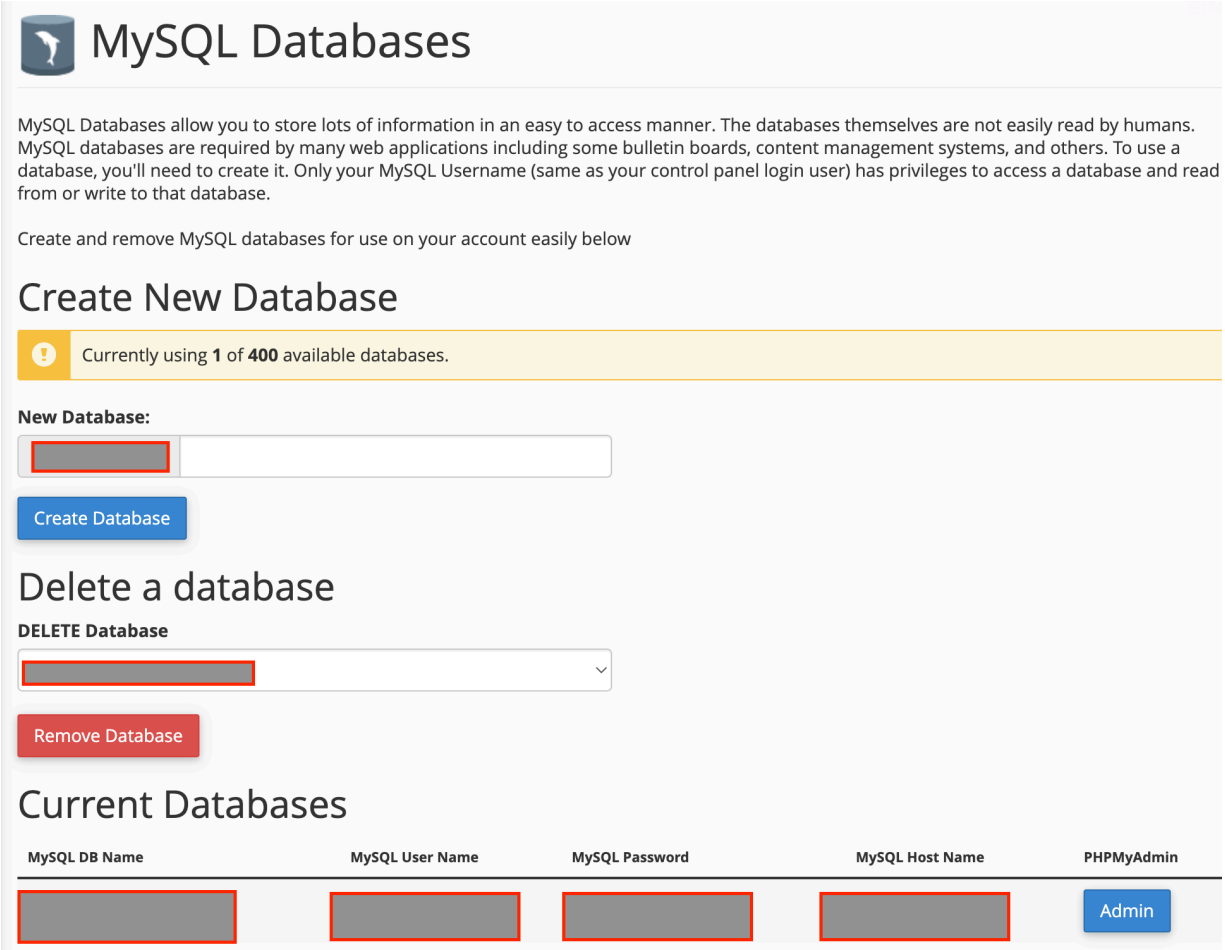
your hosting account. Within cPanel, usually this is where you will create the actual database in most cases. cPanel is also industry standard. Most web hosts run it or a skinned version of it. Even if you run into a host that uses something else. If you know cPanel, you should be good because most web hosting admin panels are similar.

In cPanel you will see something like this:



Where you see “DATABASES” is where we need to look. In most cases, to create the actual database, you need to go to the MySQL

Databases link.

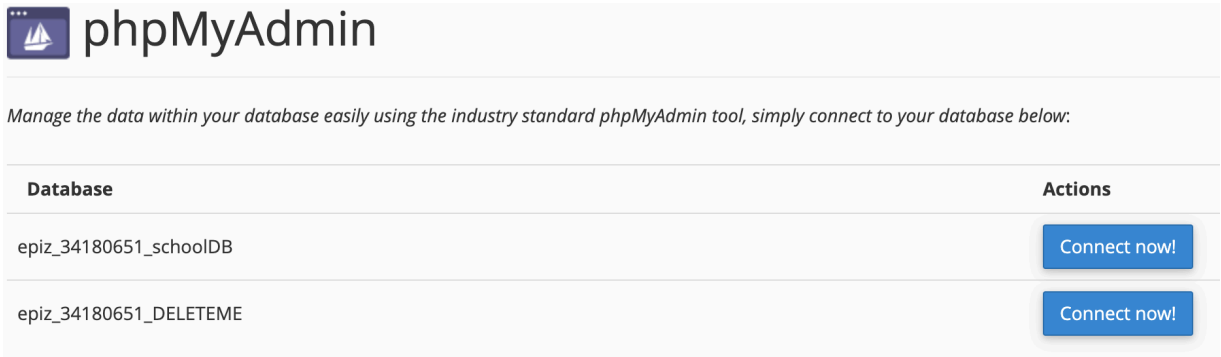


Here you will see a page like this, you can manage your databases, create new

databases, manage database users and their permissions. Different hosts might work slightly different. Once you create the DB you may have to use cPanel to create or assign a user to the DB and set its permissions. If not, you can do the same in PHPMy Admin.

If you go back to the cPanel main page and click the PHPMy Admin button.

You may see something like this, it will show you all of you databases (if any exist) click the one you want to interact with and it will bring you to PHPMy Admin.

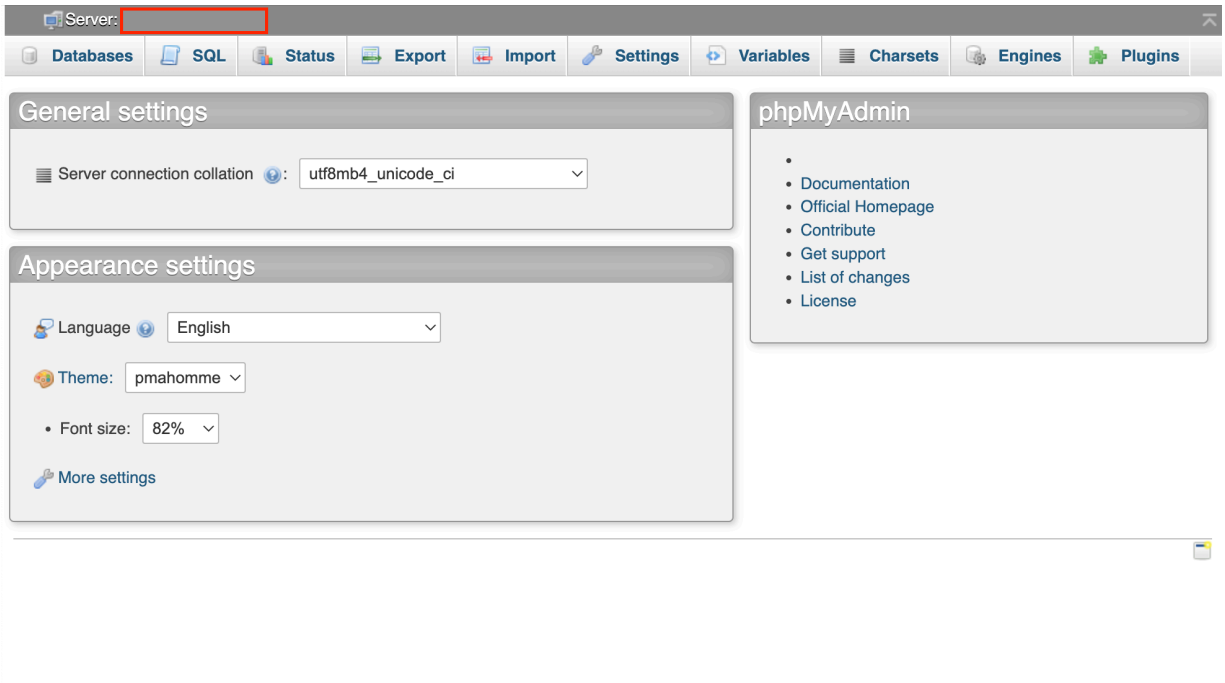


Create a table and we can move on to the next step.

PHPMyAdmin

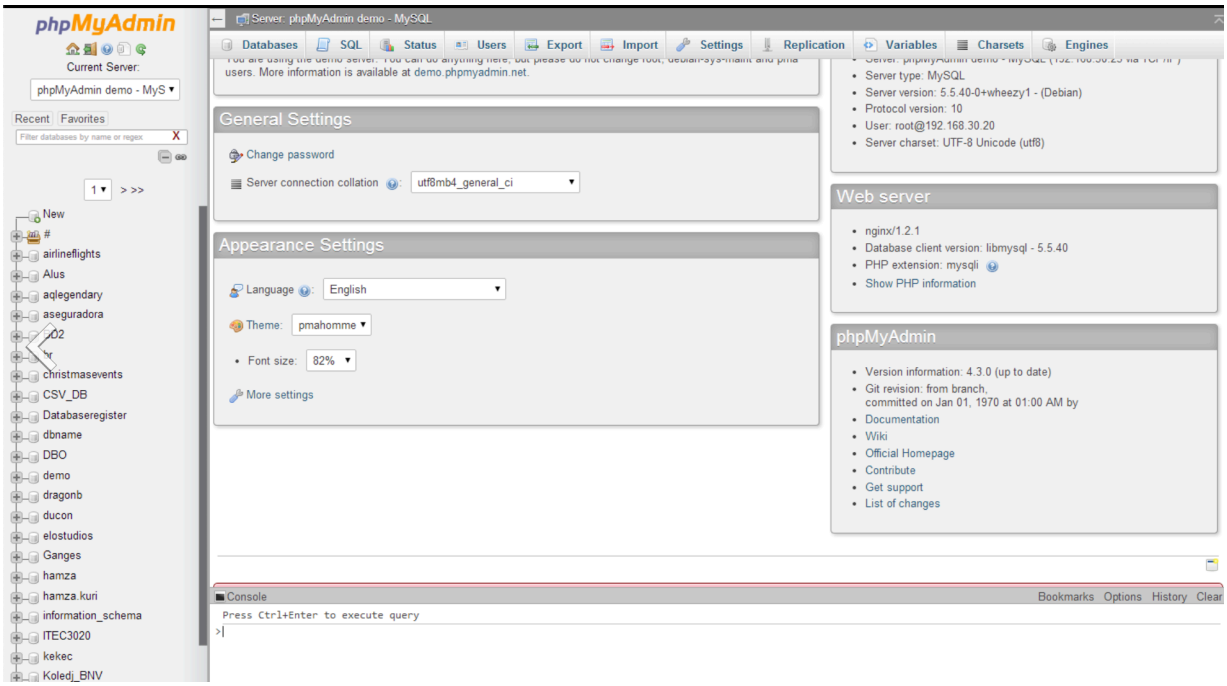
Depending on the server configuration, you can enter PHPMy Admin and you will see the interface. Again the interface may vary depending on server config.

You will see something like this:



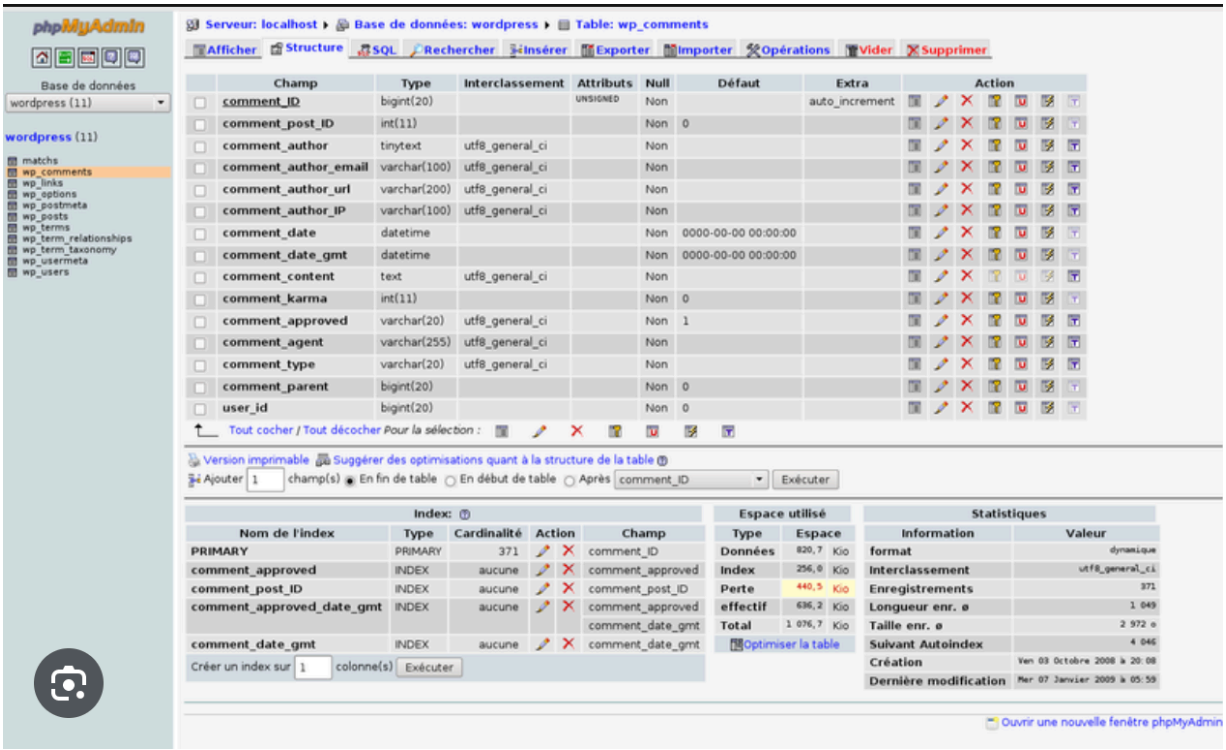
In here you can interact with the database. We can use the UI to accomplish many tasks including tasks that would require us to write SQL code. Whether you want to code manually or use the UI, it's really up to you. As long as you get the job done!

Note that most hosts will have a left hand navigation column where you can navigate through all your databases as well as more UI elements that show you more information like server info and DB info. See below:



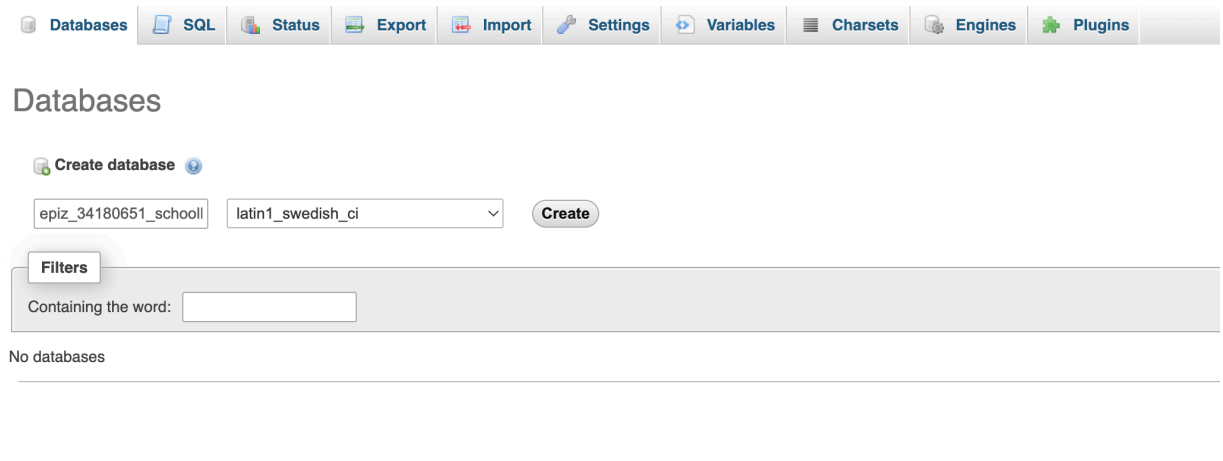
The infinity free version is made to be very basic compared to most hosts so if you are using that just keep that in mind!

Another thing to note is that you may run into old skinned versions of PHP My Admin. It's the same just the UI looks more like this below:



Now let's take a look at it more. At the top of the UI there are tabs, let's check some of them out:

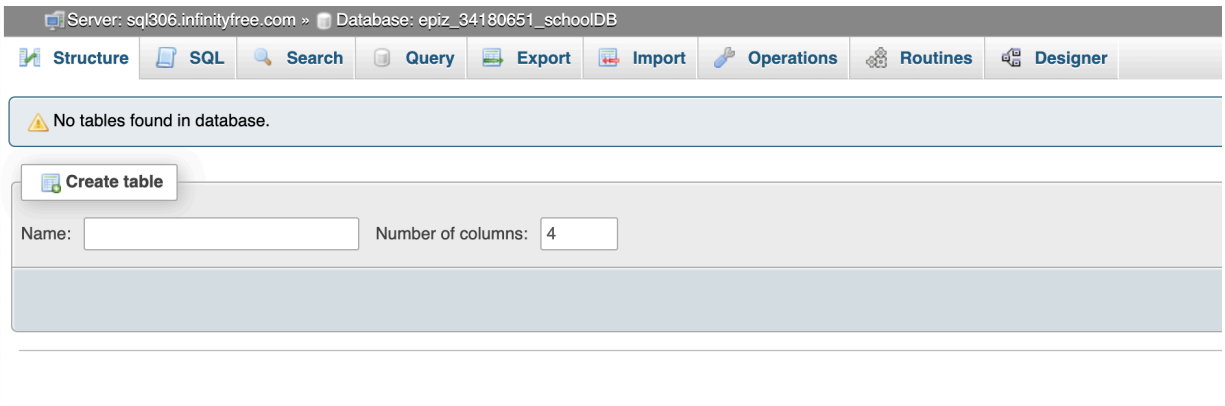
Databases:



Here you *should* see all your databases and be able to create new ones. But this step was already done in cPanel. So We can skip it.

Structure:

Note that on infinity free hosting you have to have a DB selected and enter PHPMyAdmin by clicking on the DB, if not this link will not show up.



In here you can make new tables for the DB.

Just remember it goes:

DB > Table(s) > Table Columns > Table rows (the actual data saved row by row).

So with this understood, we already made the DB. Now we need a table and we can create one here.

Let’s imagine we want to make a table for personnel. Let’s imagine we want to store:

ID

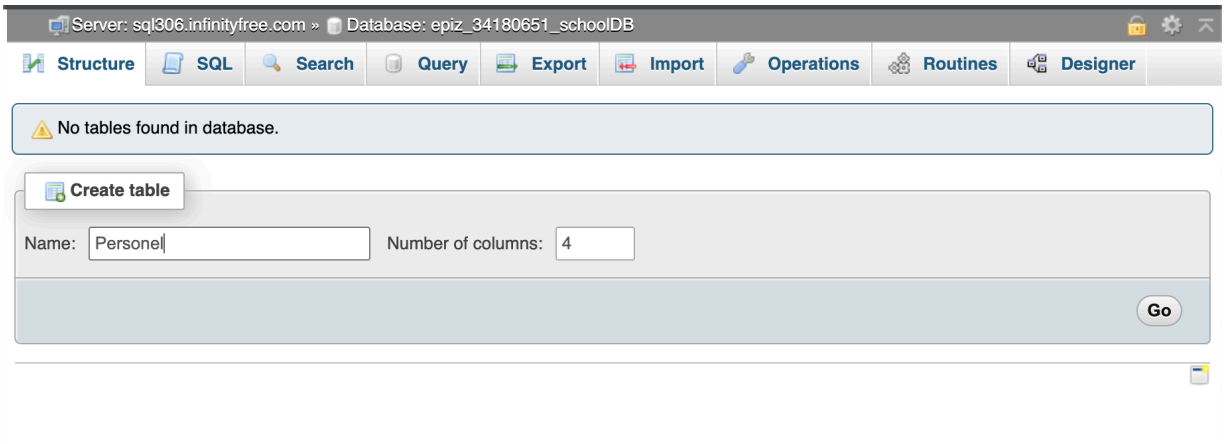
First Name

Last Name

Position

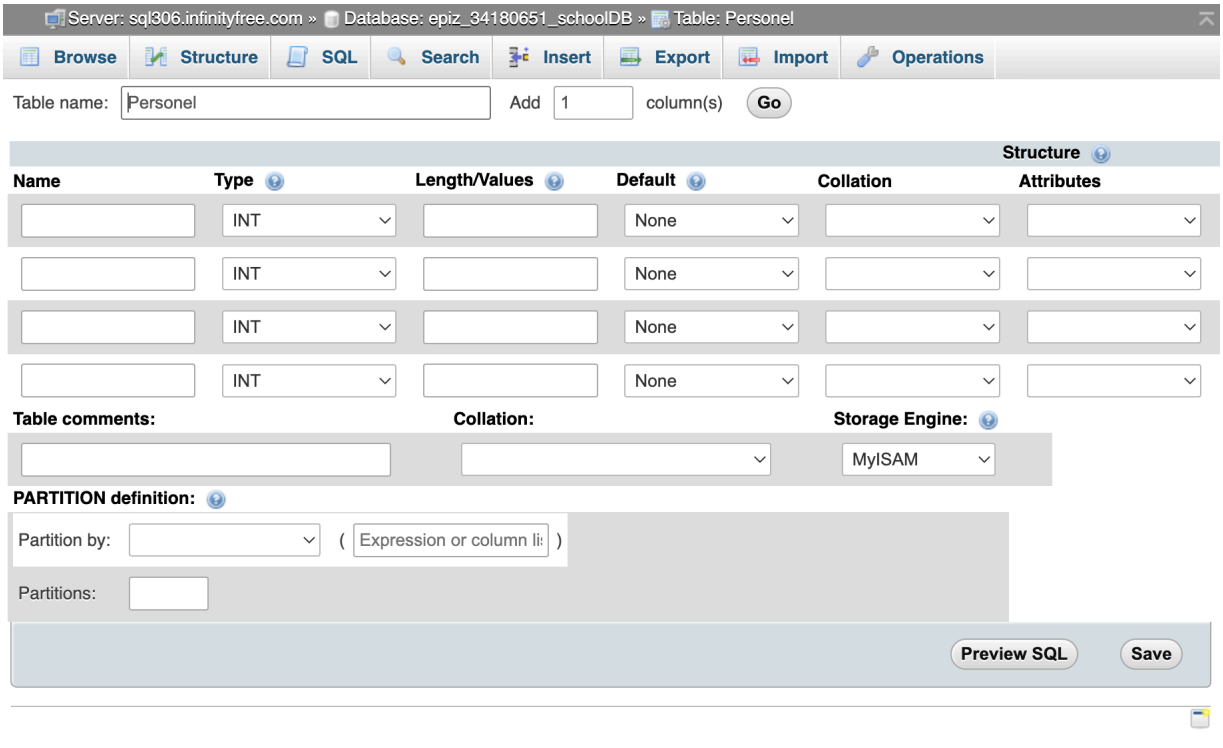
Phone Number

We can name the table Personnel or something to that effect and we would need 5 columns.



Enter that and hit “Go”.

Next you will see something like this:



We need to give the columns more information

about the data we will be storing in them:

Name of column

The data type

The length

Default value

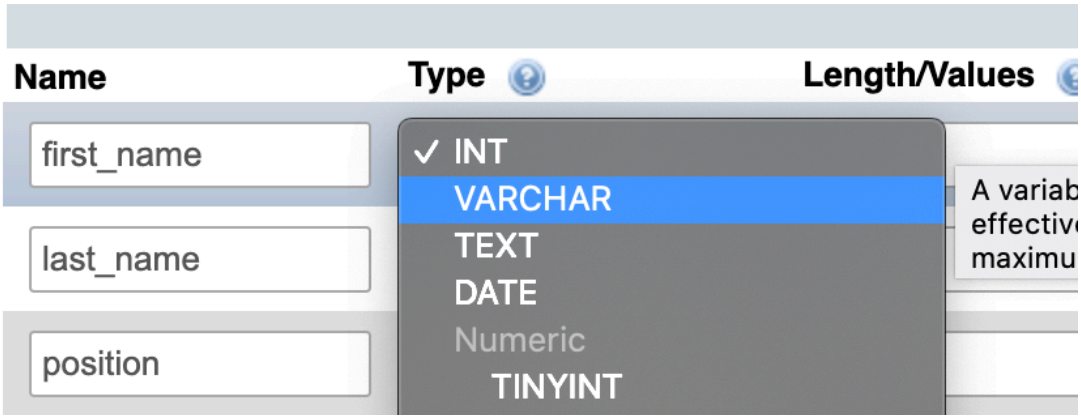
and so on.

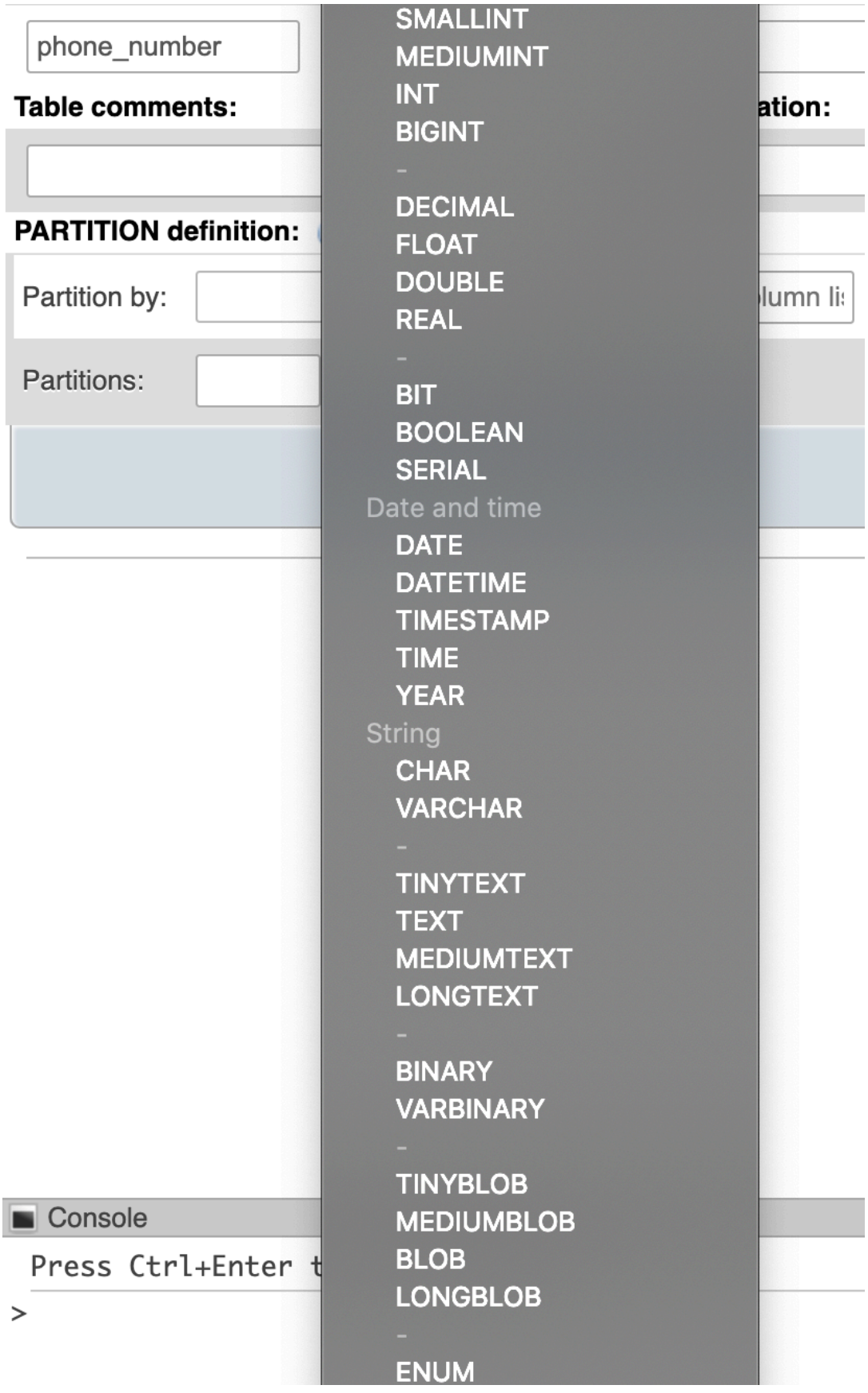
For the Names:

Use things like: “first_name” don’t use things like “First Name”.

For the Type:

There are many. But you can use these to be specific about the data being stored if needed.



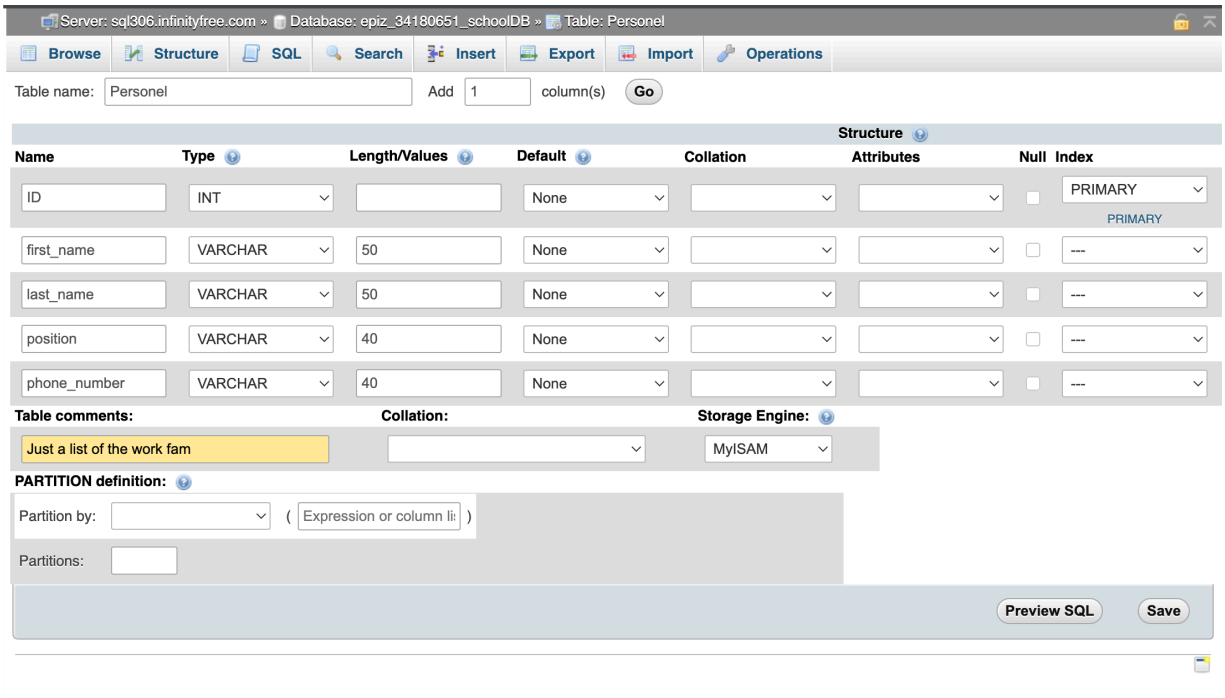

phone_number
Table comments:
ation:
PARTITION definition:
Partition by:
lumn li
Partitions:
SMALLINT
MEDIUMINT
INT
BIGINT
-
DECIMAL
FLOAT
DOUBLE
REAL
-
BIT
BOOLEAN
SERIAL
Date and time
DATE
DATETIME
TIMESTAMP
TIME
YEAR
String
CHAR
VARCHAR
-
TINYTEXT
TEXT
MEDIUMTEXT
LONGTEXT
-
BINARY
VARBINARY
-
TINYBLOB
MEDIUMBLOB
BLOB
LONGBLOB
-
ENUM
Console
Press Ctrl+Enter t
>

Use this for your reference:

https://www.w3schools.com/sql/sql_datatypes.asp

For Length:

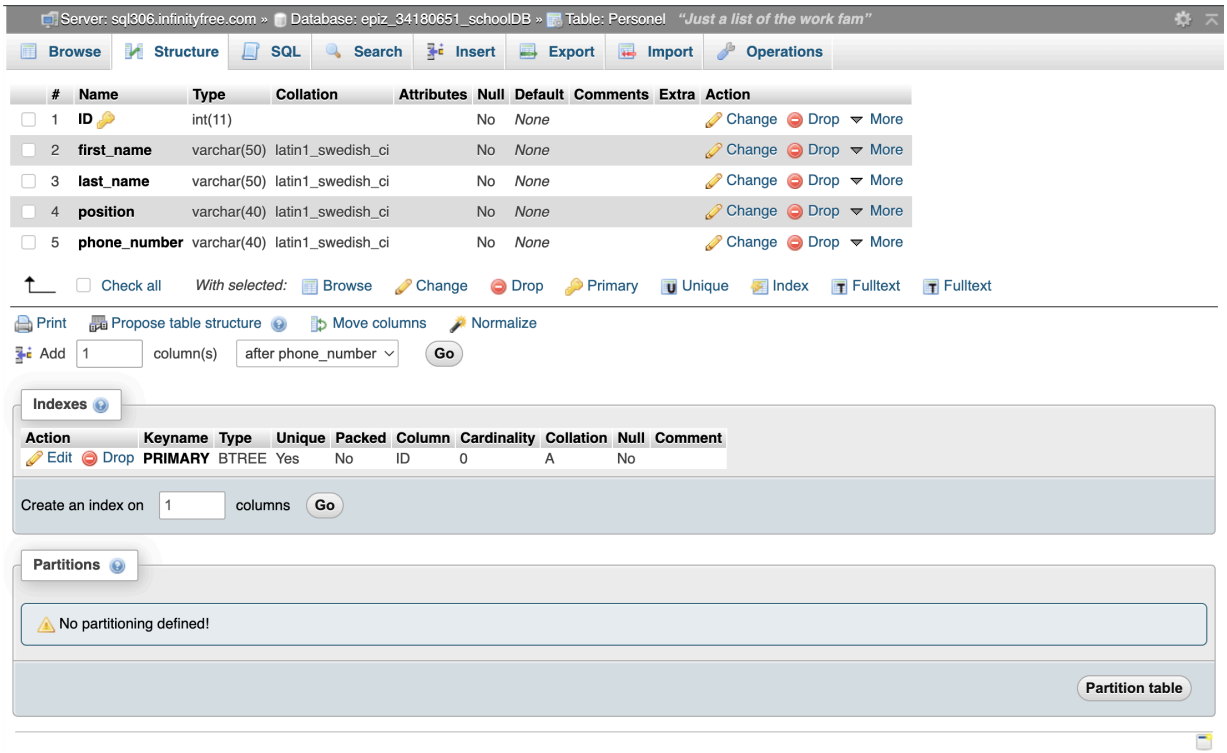
If you want to limit the users input length you can do so here. So for example a VARCHAR(20), would limit the user to some characters that vary (VAR CHAR) up to 20 characters.



From here we can preview the SQL query:

```
CREATE TABLE `epiz_34180651_schoolDB`.`Personel` ( `ID` INT NOT NULL , `f
irst_name` VARCHAR(50) NOT NULL , `last_name` VARCHAR(50) NOT NULL , `pos
ition` VARCHAR(40) NOT NULL , `phone_number` VARCHAR(40) NOT NULL , PRIMA
RY KEY (`ID`)) ENGINE = MyISAM COMMENT = 'Just a list of the work fam';
```

Or just click save. Let’s save it. We will see this:



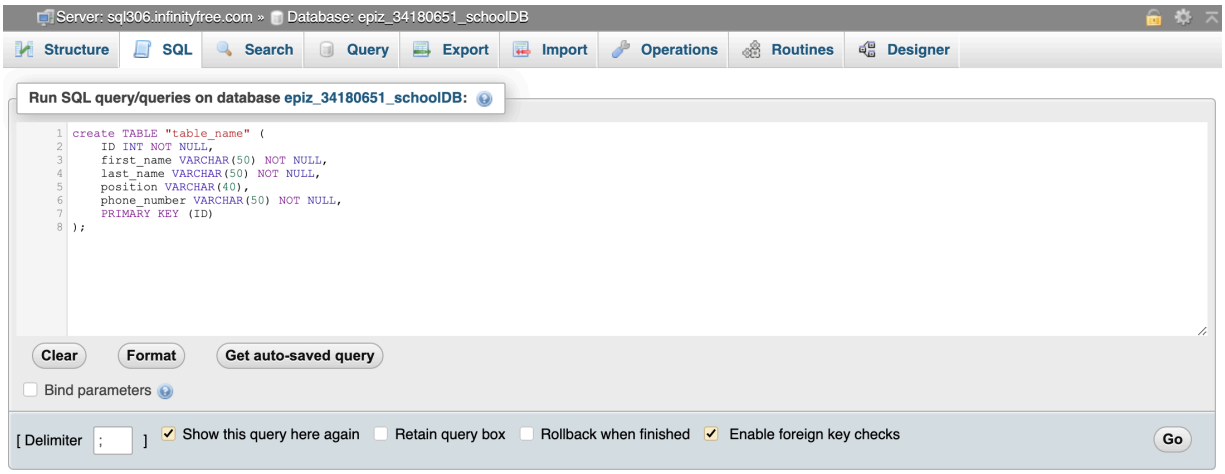
Here we can see the structure of the table. It's ready to go and have some stuff inserted into it. We will look at a method in the **Insert** section.

SQL:

In here you can actually write SQL queries. Before you get worried about having to learn yet another language. For the most part you don't need to learn SQL programming unless you plan on working as a dedicated SQL/ database dev or working very heavily on the

back end of any platform. Even if you want to be a full stack dev you may not always be working so heavy on the DB side of things. You can learn a little bit and be able to get by and of not, you can just search up whatever you need to get done!

But basically what we used the interface for above, we could of coded this in here:

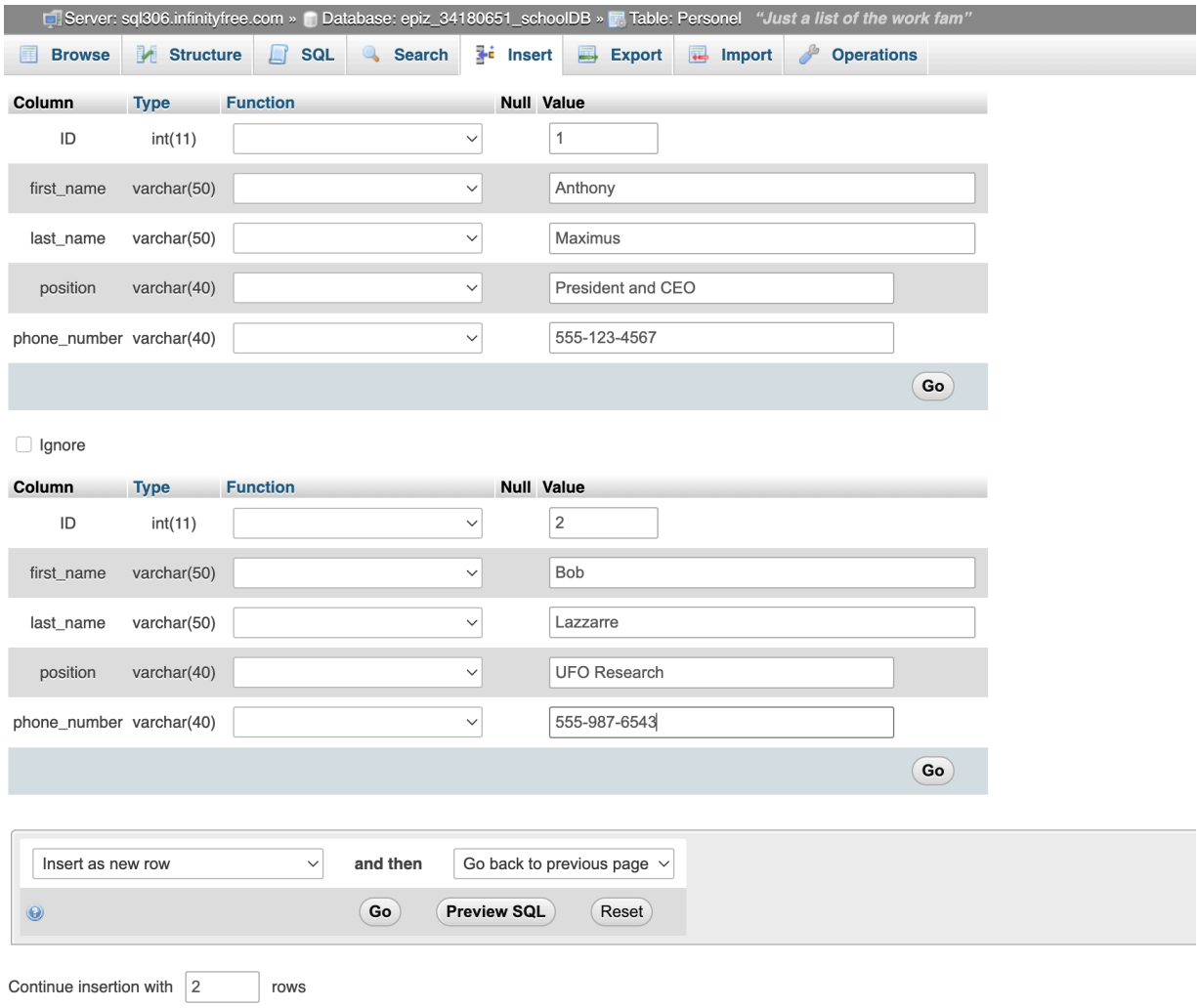



```
create TABLE "table_name" (
    ID INT NOT NULL,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    position VARCHAR(40),
```

```
    phone_number VARCHAR(50) NOT NULL,
    PRIMARY KEY (ID)
);
```

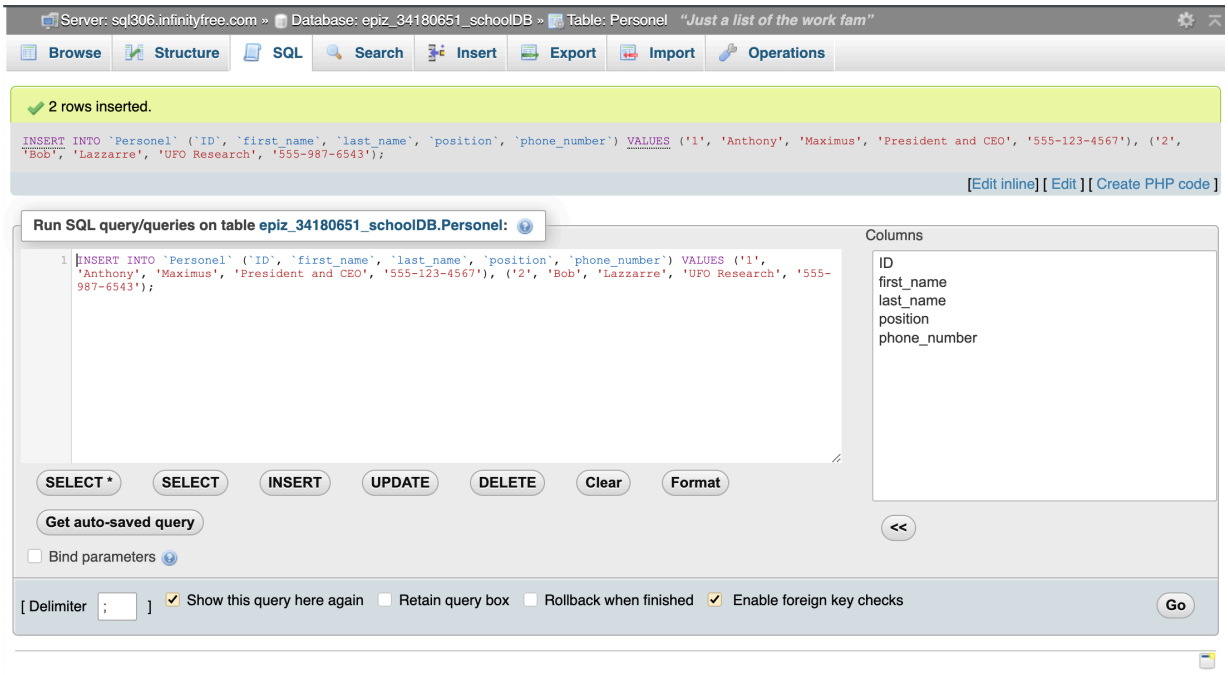
Insert

The insert tab is one way we can insert data into the DB manually (its not the dynamic way but still). Here we can insert into the fields like this:

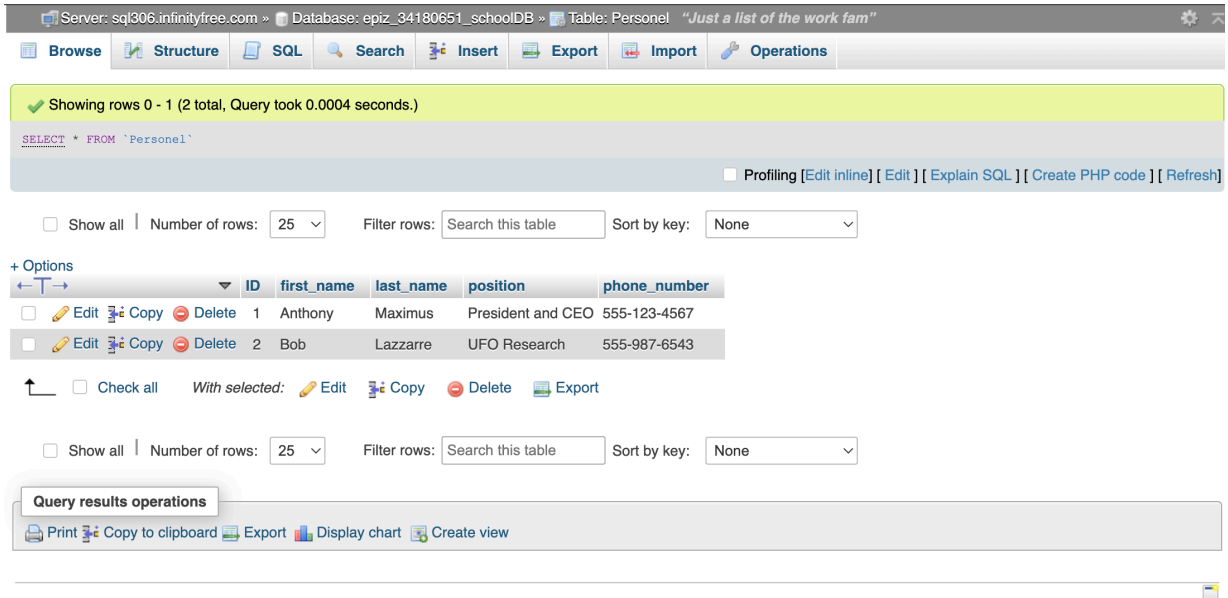


In the above case the UI allows us to enter 2 entries we can only enter 1 if thats all we need.

Hit “go” at the bottom and you will see this:



It shows us the SQL code generated. After that we can see the data in the “Structure” tab.



Use PHP to connect to DB and get data

We can use some PHP code to connect to the DB:

```
<?php

    //Make variables that store the
login cred
    $host='hostnameaddress;
              //mysql host name
    $user='username1234;
              //mysql username

$pass='makesuretouseaniceandstrongp
assword;   //mysql password
    $db='the_name_of_the_DB;
              //mysql database

    //use the vars to login
    $con=mysqli_connect($host,
$user,$pass,$db);
    if($con) {
    echo "Connection successful";
    }
    else {
    echo "Connection error";
```

```
    }

?>
```

Note that the above will not work locally, upload it to the server and use it from there.

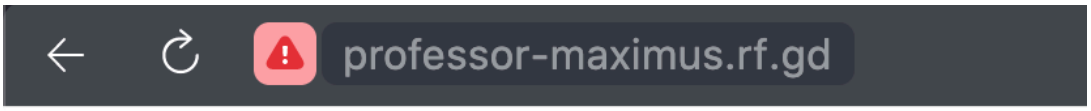
The checker works. Let's make another file and in it retrieve the data.

```
//make some vars to store the login
cred
$host = "your_host";
$user = "your_username";
$pass = "your_password";
$db = "your_database";

// Connect to the database
$con = mysqli_connect($host, $user,
$pass, $db);
if (mysqli_connect_errno()) {
  echo "Failed to connect to MySQL:
" . mysqli_connect_error();
  exit();
}
```

```php
// Fetch data from the "Personel" table. Yeah yeah I made a typo
$query = "SELECT ID, first_name, last_name, position, phone_number FROM Personel"//lulz;
$result = mysqli_query($con, $query);

// Check if there are any rows returned
if (mysqli_num_rows($result) > 0) {
  echo "<ul>";

  // Loop through each row and display the data as list items
  while ($row = mysqli_fetch_assoc($result)) {
    echo "<li>ID: " . $row['ID'] . "</li>";
    echo "<li>First Name: " . $row['first_name'] . "</li>";
    echo "<li>Last Name: " . $row['last_name'] . "</li>";
    echo "<li>Position: " . $row['position'] . "</li>";
```

```
    echo "<li>Phone Number: " .
$row['phone_number'] . "</li>";
    echo "<br>";
  }

  echo "</ul>";
} else {
  echo "No data found in the
'Personel' table.";
}

// Close the database connection
mysqli_close($con);
?>
```

Output:



- ID: 1
- First Name: Anthony
- Last Name: Maximus
- Position: President and CEO
- Phone Number: 555-123-4567

- ID: 2
- First Name: Bob
- Last Name: Lazzarre
- Position: UFO Research
- Phone Number: 555-987-6543