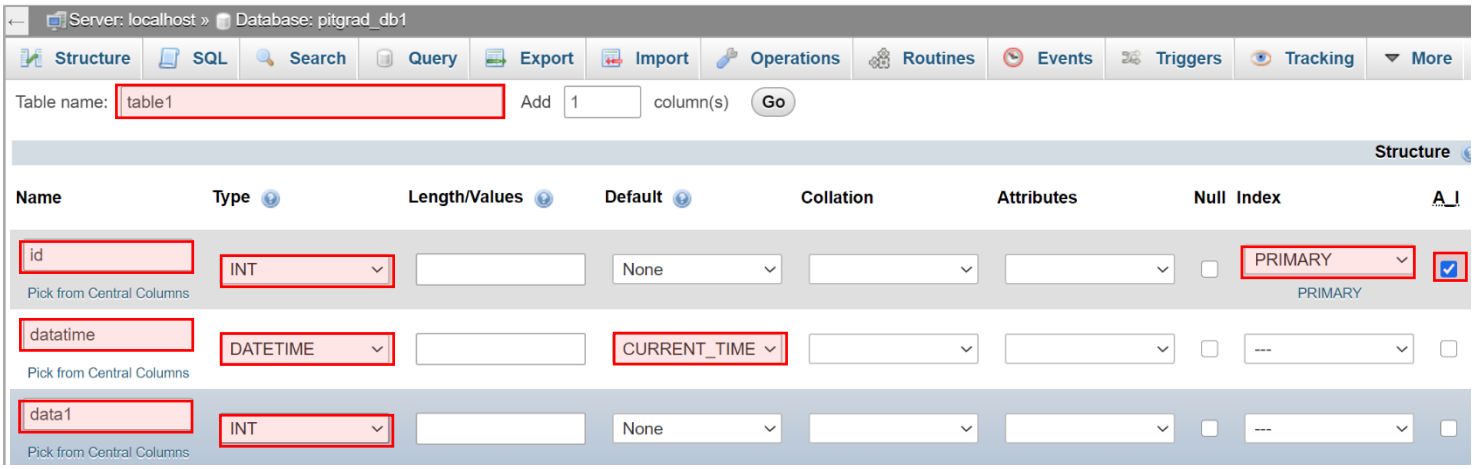
a10. MySQL direct query

The exercise consists of two parts:
- the MySQL database table created on your hosting server
- the Arduino application sending data (numbers) to that database.

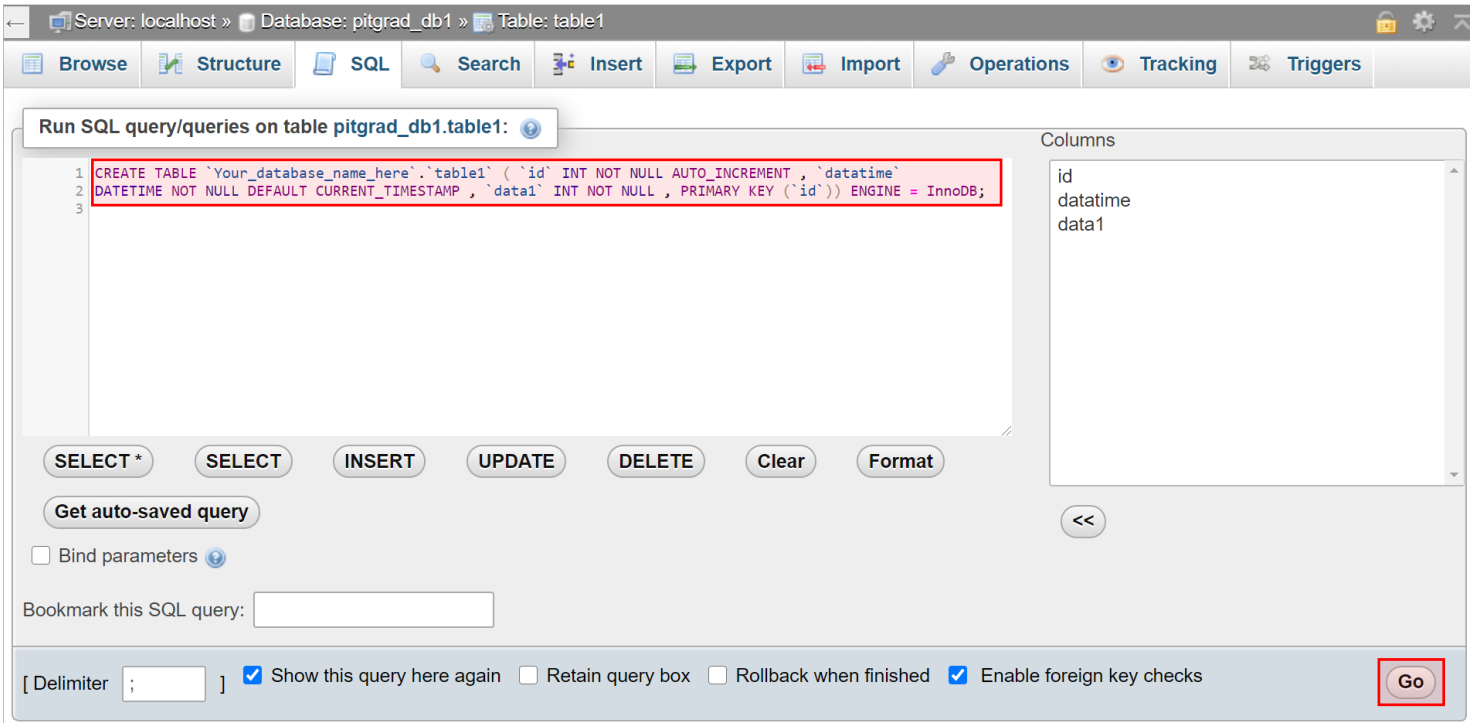
HOSTING SERVER PART

1. First you need to create a database on your hosting.
    a. You need to use the name of this database in your query in the Arduino code.
2. Create a table with the name ‘table1’ in this database. It should have the structure presented below.



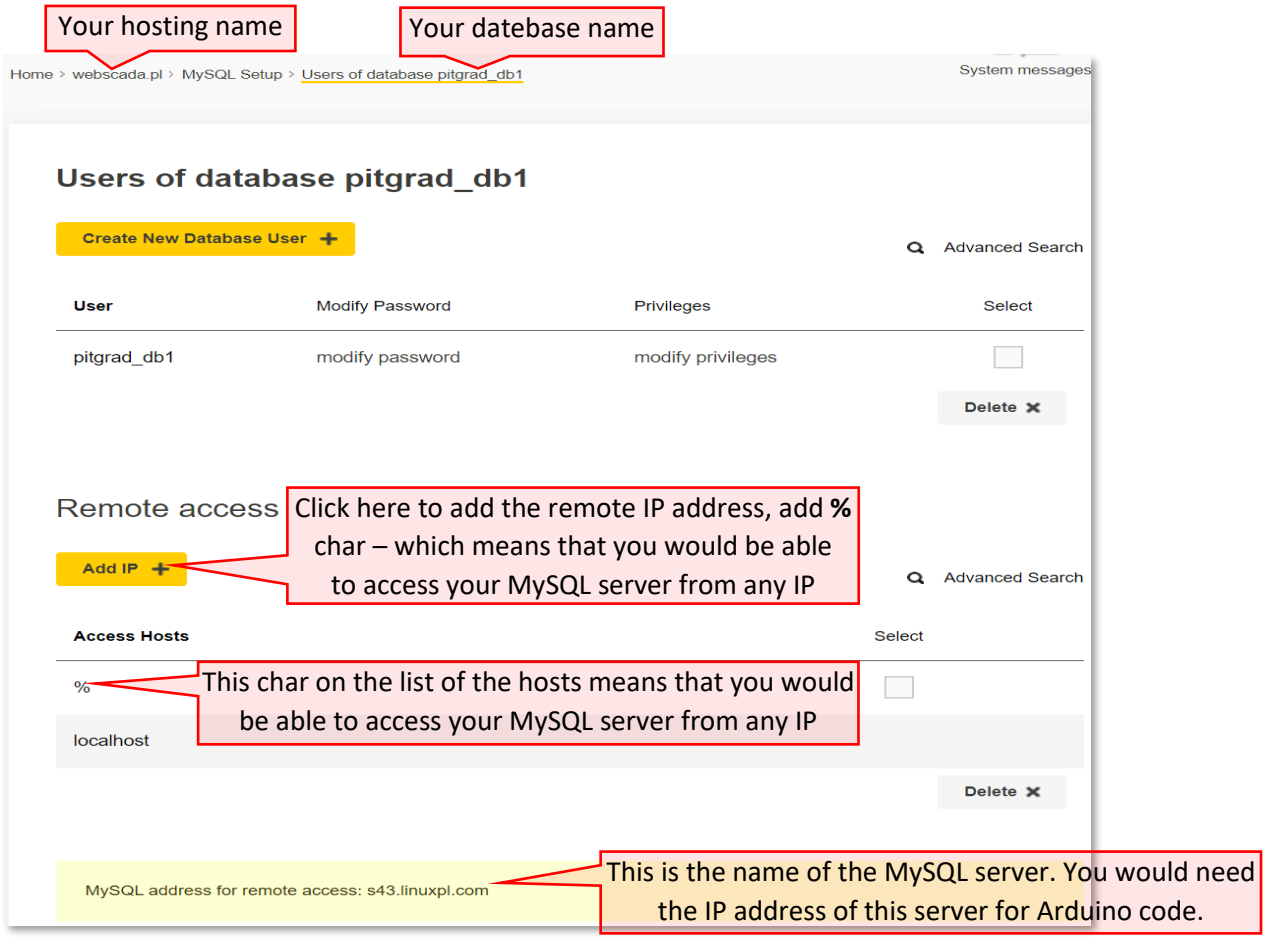
Alternatively you can create this table putting SQL code to the SQL query field and executing it

```
CREATE TABLE `Your_database_name_here`.`table1` ( `id` INT NOT NULL AUTO_INCREMENT , `datatime`
DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP , `data1` INT NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB;
```

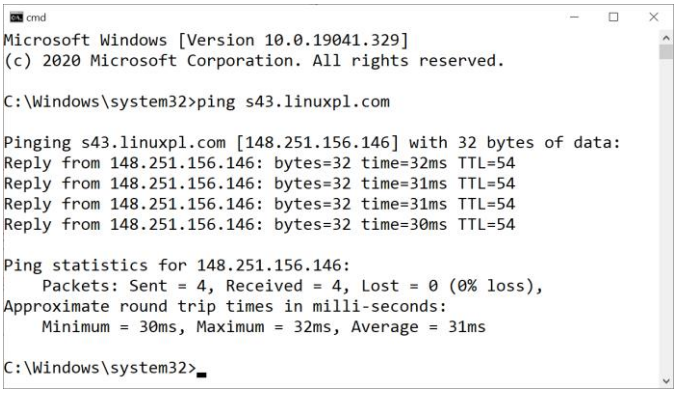


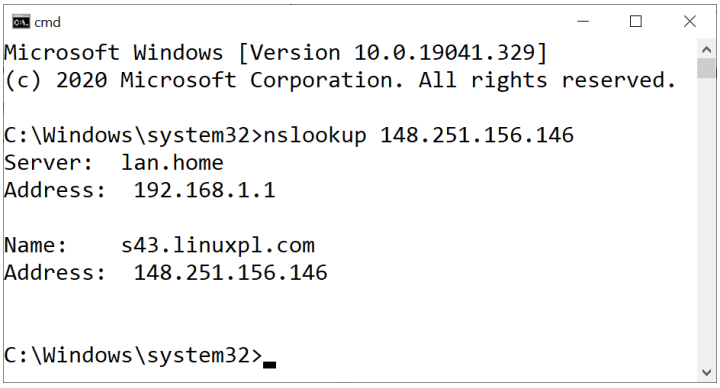
Most users access their MySQL database tables from the PHP scripts located in the same hosting. This is local access and the name of the server is then localhost. In most cases it is default configuration.
In most hosting services it is not possible to access the database using 3306 port without additional configuration.
It is organized in this way because of security reasons.
Your Arduino controller has no local access to the hosting server so you need to allow to access your database using 3306 port from any IP address, because you should realize that your Arduino would get the access to Internet using IP address of the router that provides the Internet access for the LAN network that you are using in laboratory.

There is the printscreen presenting how to do it in some hostings.



To get the IP address of your MySQL server, you need to ping this host, using its domain name.
Just use *ping domain name* command in the Command window, as it is presented below on the left.
The ping command returns you the IP address of the host specified using domain name.
If you need to get a domain name based on IP address you should use nslookup command presented below on the right.

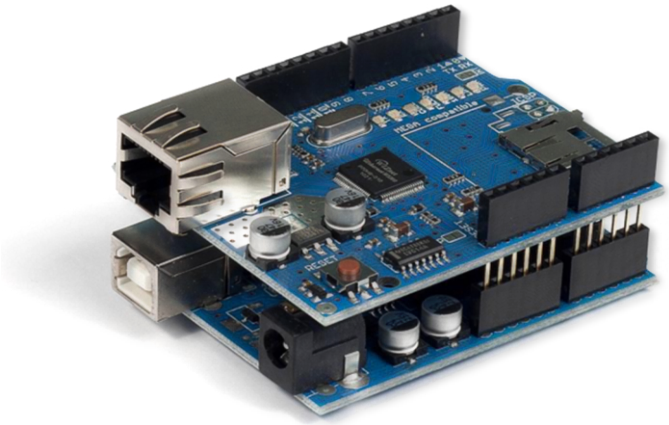


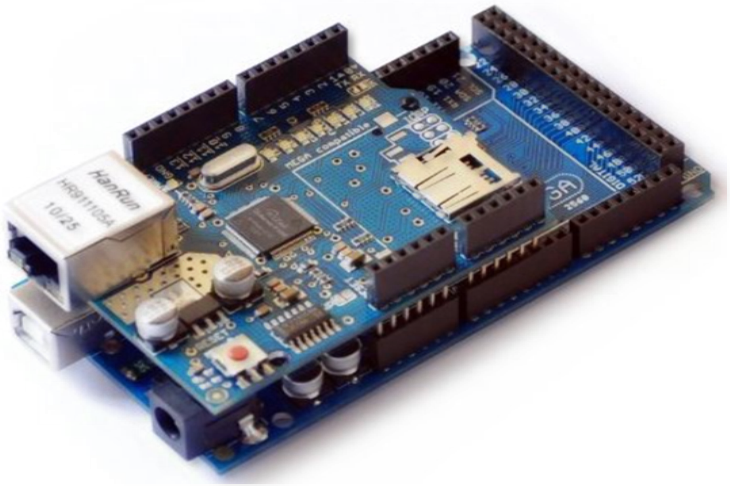


When structure of your hosting management looks different – just search for MySQL Setup and Remote access parameters in your hosting configuration. If your hosting configuration is complicated and you do not know where to find it - please google that information specifying the following query in Google inputline:
your_hosting_provider_name MySQL remote access

ARDUINO PART

To run this application you need to have Arduino Ethernet model or you need to connect Ethernet Shield to any Arduino you have, for instance Arduino UNO or Arduino MEGA:





Your Arduino Ethernet Shield should be connected to Internet using Ethernet cable.
When it is done correctly – the LED on RJ45 socket should blink - activity – usually yellow, link-up – usually green.

The application for Arduino presented below requires additional libraries: MySQL_Connection.h, MySQL_Cursor and Ethernet.h. The Ethernet.h library is usually installed on Arduino software, but the other two libraries should be usually installed manually. You would know that manual installation is required when there is a compilation error showing that the library is not found. Is such situation please go to Tools -> Manage Libraries… and put to inputline of this window the name of the library that was not found on your Arduino software, then you should agree to install this library.

The application presented below allows you to send the constant value '1234' to the column data1 in your table1 in your database. It makes it sending to your MySQL server the following SQL query:

```
INSERT INTO Your_database_name.table1 (data1) VALUES ('1234')"
```

1. Replace the parameters marked with yellow colour with the parameters of your hosting and database.
2. Compile it and send it to your Arduino.
3. When there is no error open Tools -> Serial monitor, to observe if your Arduino is connecting with your database server.
4. If it is so, go to your hosting and open phpMyAdmin to observe if new data appeared in the table1.
5. To observe new records – please refresh your browser with phpMyAdmin displaying table1 contents.

App1 - INSERT to the database of the constant value '1234' every 3 seconds

```
#include <Ethernet.h>
#include <MySQL_Connection.h>  // library for MySQL connection
#include <MySQL_Cursor.h>      // library for SQL code execution
byte myMAC[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0x77};  // Arduino MAC address - every student should have a different one
IPAddress mySQLServer(123,123,123,123);   // replace 123,123,123,123 with your MySQL server IP address

EthernetClient client;
MySQL_Connection conn((Client *)&client);

void setup() {
 Serial.begin(9600);
 Serial.println("Connecting...");
 Ethernet.begin(myMAC);
 do { delay(1000); }
  while (!conn.connect(mySQLServer, 3306, "Your_database_name", "Your_database_password"));
}

void loop() {
 delay(3000);
 MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);              // Initiate the query class instance
 cur_mem->execute("INSERT INTO Your_database_name.table1 (data1) VALUES ('1234')");             // Execute the query
 delete cur_mem;                       // Deleting the cursor frees up memory used
}
```

The second application allows you to put any data to your database. That is why this application is a little bit more complicated, because you do not send to your MySQL server just the constant string as the SQL query, but you need to prepare that string and insert into this string a variable which would represent the value that you want to place into the column data1 in the table1.

a) Declare the empty SQL query string:

```
char mySQL1[] = "                                                    ";
```

b) Declare a variable, that would be used to insert increasing numbers to the SQL query string mySQL1:

```
int myCounter = 0;
```

c) Increase the value of myCounter variable every 3 seconds:

```
myCounter++;
```

d) Put myCounter variable into the SQL query string mySQL1:

```
sprintf(mySQL1,"INSERT INTO Your_database_name.table1 (data1) VALUES ('%u')",myCounter);
```

e) Execute that SQL query using SQL query string mySQL1:

```
cur_mem->execute(mySQL1);
```

App2 - INSERT to the database of the following numbers 1, 2, 3, …. every 3 seconds

```
#include <Ethernet.h>
#include <MySQL_Connection.h>  // library for MySQL connection
#include <MySQL_Cursor.h>      // library for SQL code execution
byte myMAC[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0x77};  // Arduino MAC address - every student should have a different one
IPAddress mySQLServer(123,123,123,123);   // replace 123,123,123,123 with your MySQL server IP address
char mySQL1[] = "                                                    ";   // a string for the SQL query
int myCounter = 0;    // the variable used to count numbers to be sent to your database

EthernetClient client;
MySQL_Connection conn((Client *)&client);

void setup() {
 Serial.begin(9600);
 Serial.println("Connecting...");
 Ethernet.begin(myMAC);
 do { delay(1000); }
  while (!conn.connect(mySQLServer, 3306, "Your_database_name", "Your_database_password"));
}

void loop() {
 delay(3000);
 MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);              // Initiate the query class instance
 myCounter++;
 sprintf(mySQL1,"INSERT INTO Your_database_name.table1 (data1) VALUES ('%u')",myCounter);  // placing a variable to the SQL query
 Serial.println(mySQL1);
 cur_mem->execute(mySQL1);              // Execute the query
 delete cur_mem;                        // Deleting the cursor frees up memory used
}
```

Go to your hosting and open phpMyAdmin to observe if new data appeared in the table1.
There should be every 3 seconds new record with the following numbers 1, 2, 3, and so on.
To observe new records – please refresh your browser with phpMyAdmin displaying table1 contents.

App3. Replace the value of myCounter with any measured value specified by your teacher for instance:

a) Light level,
b) Noice level,
c) Temperature
d) Voltage.

Create a simple PHP application presenting the data from your table1 in the form of table.

```php
<body>
    <?php
    $dbhost="localhost"; $dbuser="yourdbusr"; $dbpassword="yourpass"; $dbname="yourdbname";
    $connection = mysqli_connect($dbhost, $dbuser, $dbpassword, $dbname);
    if (!$connection) {
        echo " MySQL Connection error." . PHP_EOL;
        echo "Errno: " . mysqli_connect_errno() . PHP_EOL;
        echo "Error: " . mysqli_connect_error() . PHP_EOL;
        exit;
    }
    $result = mysqli_query($connection, "SELECT * FROM table1") or die ("DB error: $dbname");
    print "<TABLE CELLPADDING=5 BORDER=1>";
    print "<TR><TD>id</TD><TD>datatime</TD><TD>data1</TD></TR>\n";
    while ($row = mysqli_fetch_array ($result)) {
        $id = $row[0];
        $datatime = $row[1];
        $data1 = $row[2];
        print "<TR><TD>$id</TD><TD>$datatime</TD><TD>$data1</TD></TR>\n";
    }
    print "</TABLE>";
    mysqli_close($connection);
?>
```