***Создание базы данных для программы на С++ «Обои» в SQL пошагово.***

1. **Скачать (с официального сайта - https://www.apachefriends.org/)**
2. **Установить XAMPP (по умолчанию в корень С:\ )**

**XAMPP**— кроссплатформенная сборка локального веб-сервера, содержащая [Apache](https://ru.wikipedia.org/wiki/Apache_HTTP_Server" \o "Apache HTTP Server), [MariaDB](https://ru.wikipedia.org/wiki/MariaDB" \o "MariaDB), интерпретатор скриптов [PHP](https://ru.wikipedia.org/wiki/PHP), язык программирования [Perl](https://ru.wikipedia.org/wiki/Perl" \o "Perl) и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер. Программа регулярно обновляется, для включения в состав новейших версий [Apache](https://ru.wikipedia.org/wiki/Apache_HTTP_Server" \o "Apache HTTP Server)/[MySQL](https://ru.wikipedia.org/wiki/MySQL" \o "MySQL)/[PHP](https://ru.wikipedia.org/wiki/PHP) и [Perl](https://ru.wikipedia.org/wiki/Perl" \o "Perl). Также XAMPP идёт со множеством других модулей, включая [OpenSSL](https://ru.wikipedia.org/wiki/OpenSSL" \o "OpenSSL) и [phpMyAdmin](https://ru.wikipedia.org/wiki/PhpMyAdmin" \o "PhpMyAdmin). Программа поддерживает создание и управление базами данных [MySQL](https://ru.wikipedia.org/wiki/MySQL" \o "MySQL) и [SQLite](https://ru.wikipedia.org/wiki/SQLite" \o "SQLite).

XAMPP — это [акроним](https://ru.wikipedia.org/wiki/%D0%90%D0%BA%D1%80%D0%BE%D0%BD%D0%B8%D0%BC):

X - (любая из четырёх операционных систем)

A - Apache

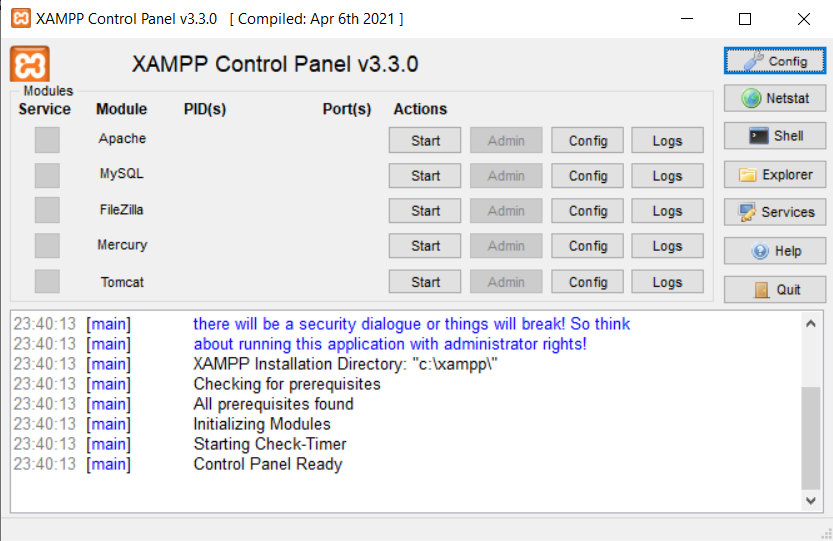
M - MariaDB

P - PHP

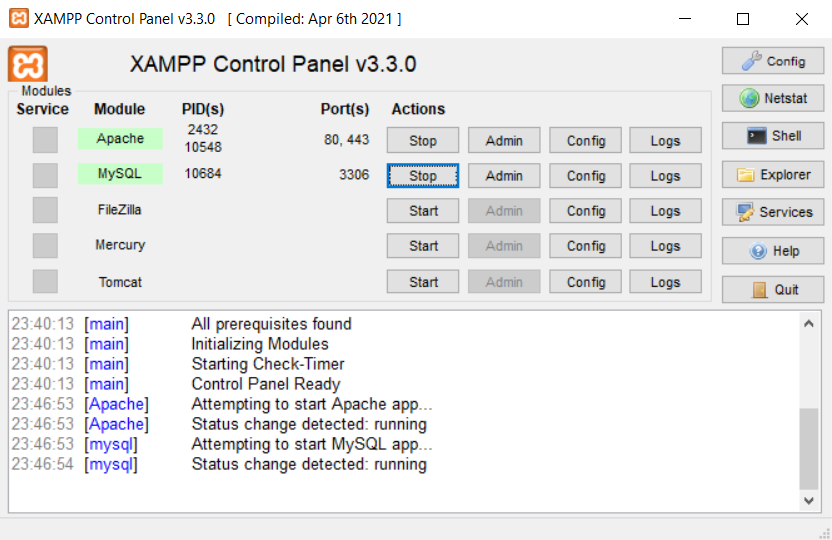
P - Perl.

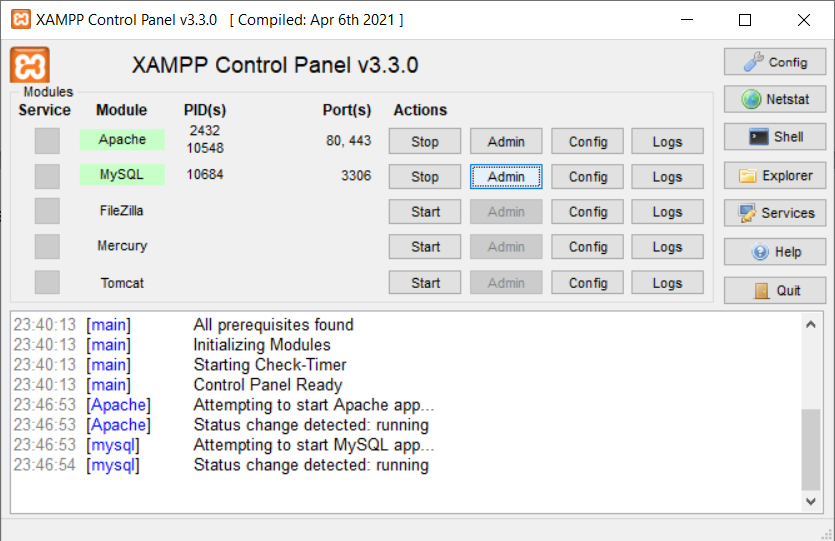


1. **Запустить XAMPP**

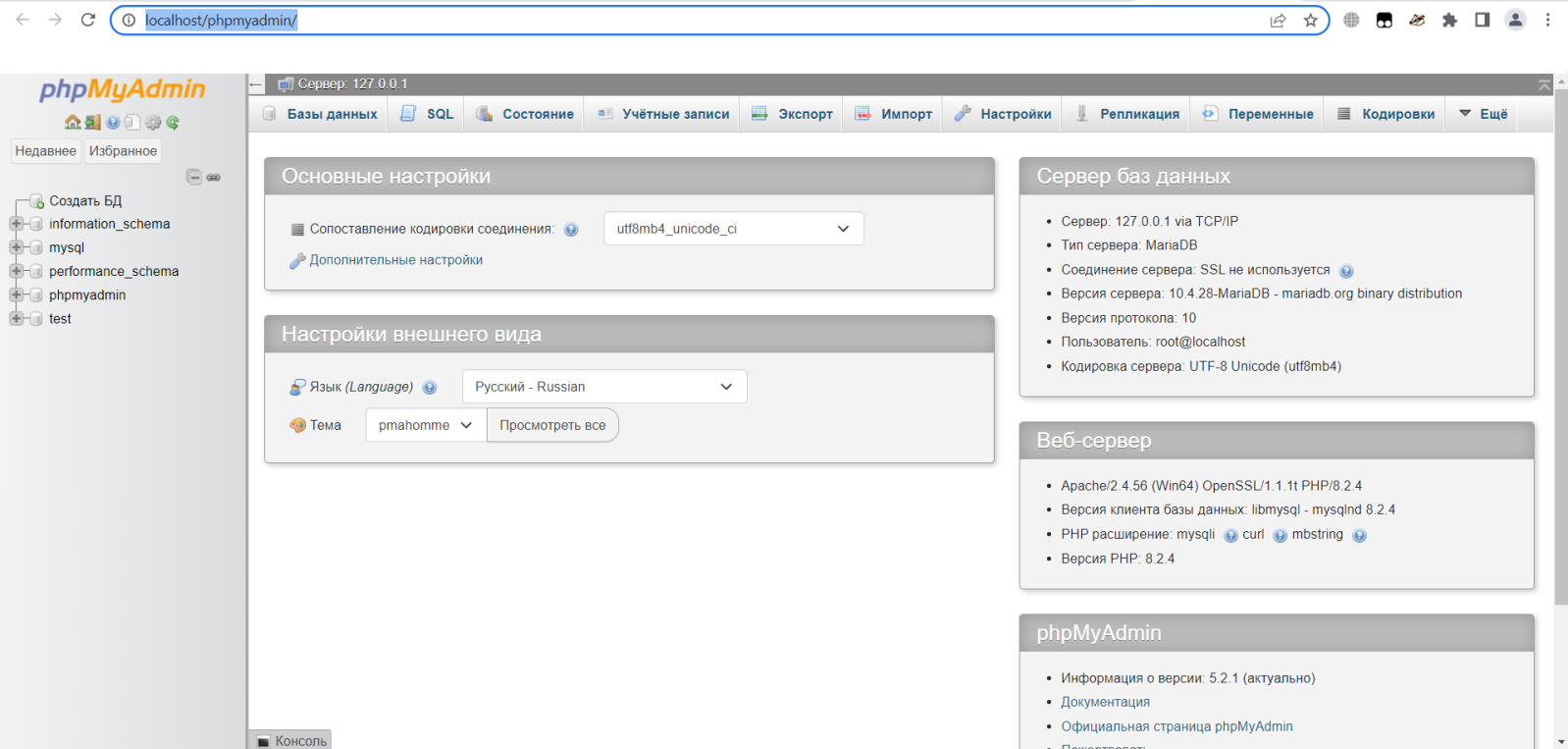


1. **Нажать кнопки Start напротив модулей Apache и MySQL (они подсветятся зеленым цветом)**

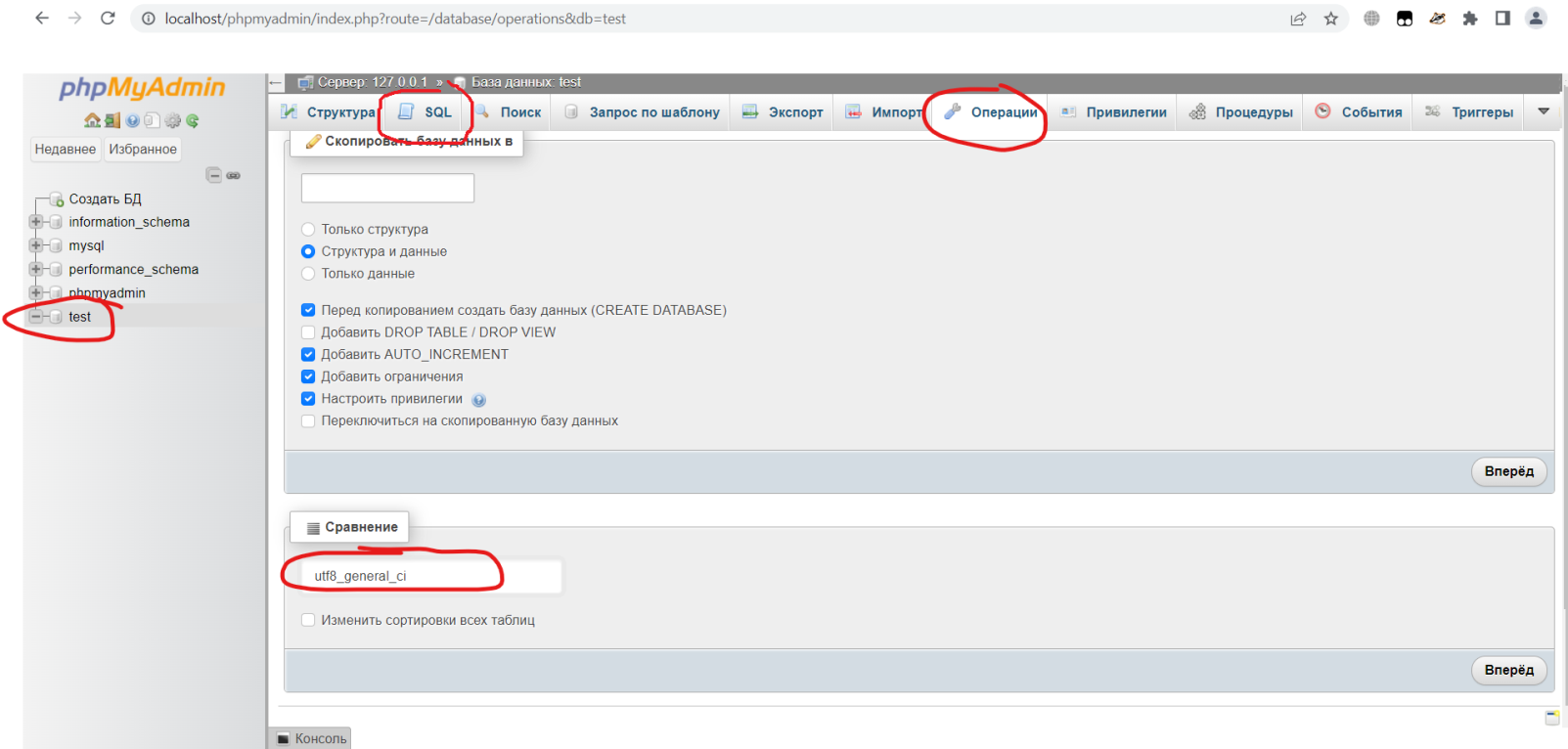
****

1. **Нажать кнопку Admin напротив модуля MySQL, **

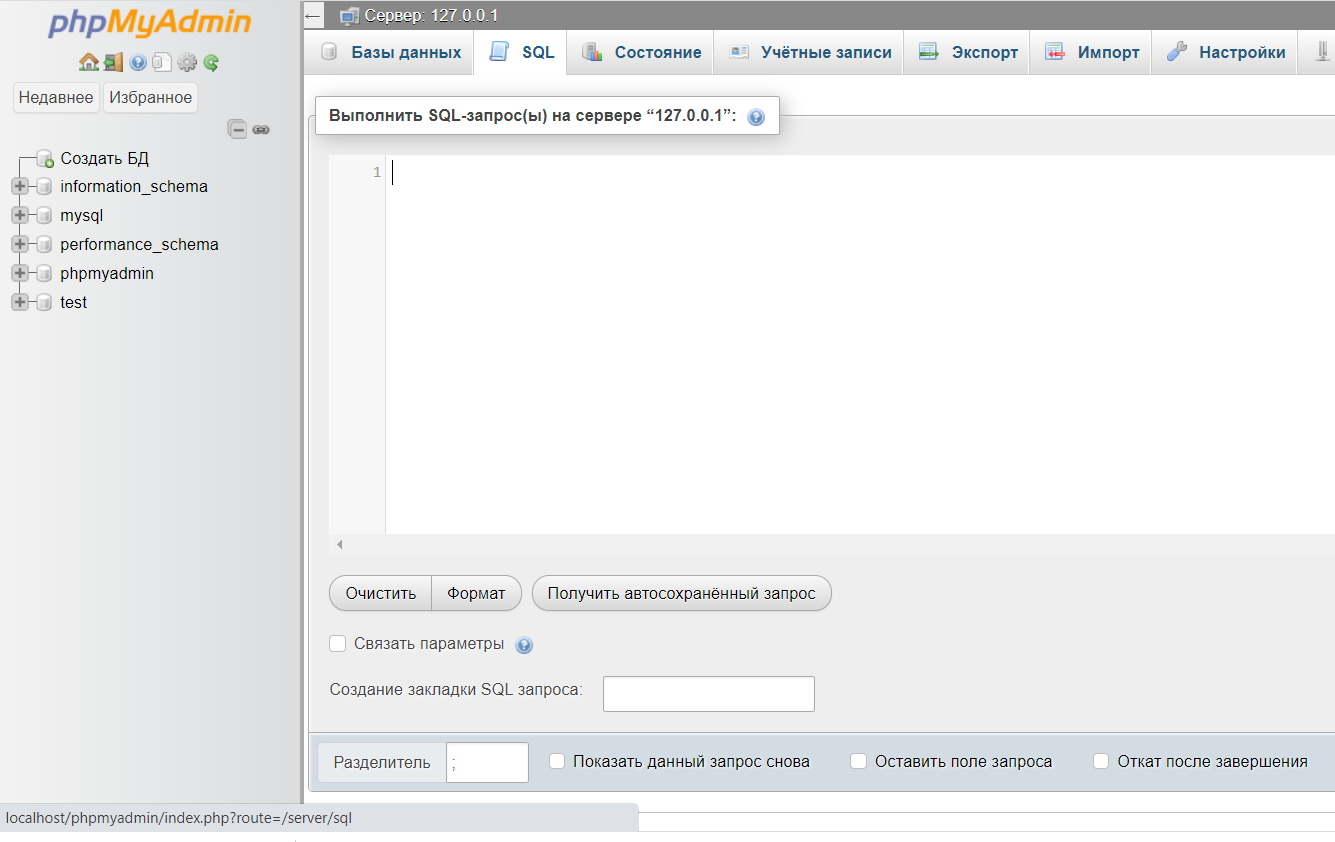
**произойдет переход на модуль phpMyAdmin (**[**http://localhost/phpmyadmin/**](http://localhost/phpmyadmin/)**). Работаем в нём.**

****

1. **Выбрать сопоставление кодировки соединения в закладке «Операции» (для корректного отображения кирилицы) – utf8\_general\_si, нажать «Вперёд», далее нажать на «test», затем на SQL (наша база данных называется «test») и поставить галочку на «Изменить сортировки всех таблиц»**

****

**Откроется следующее пустое поле**

****

1. **Теперь необходимо выполнить запрос(ы) к базе данных «test»**, написав следующее (обратить внимание на кавычки ` ` - это обратные кавычки на кнопке клавиатуры «ё ~»):

CREATE TABLE `catalogWallpaper` # название каталога

( # объявляем переменные

`idwallpaper` int (11) unsigned not null AUTO\_INCREMENT,

`nameMark` varchar(256) not null,

`price` double not null,

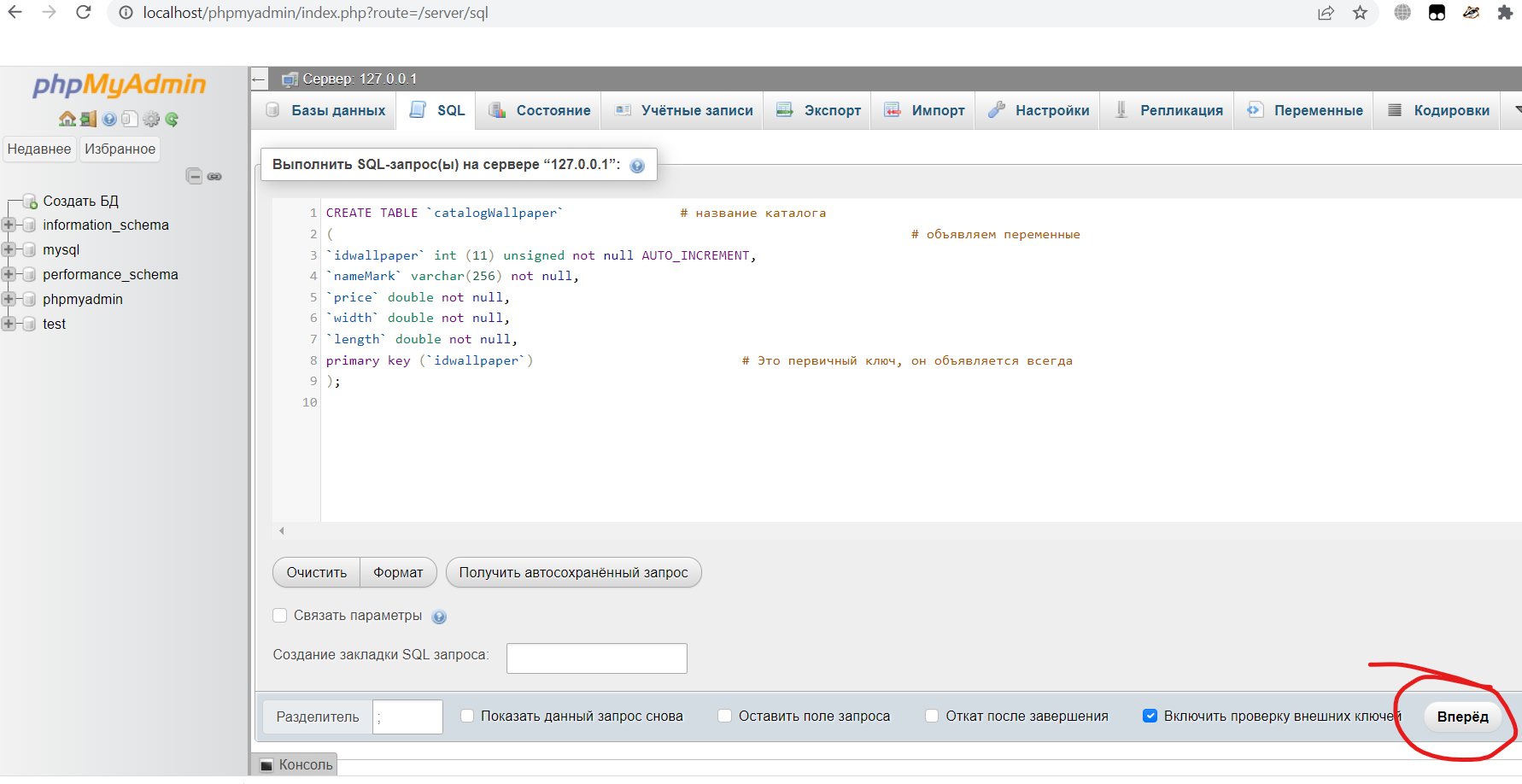
`width` double not null,

`length` double not null,

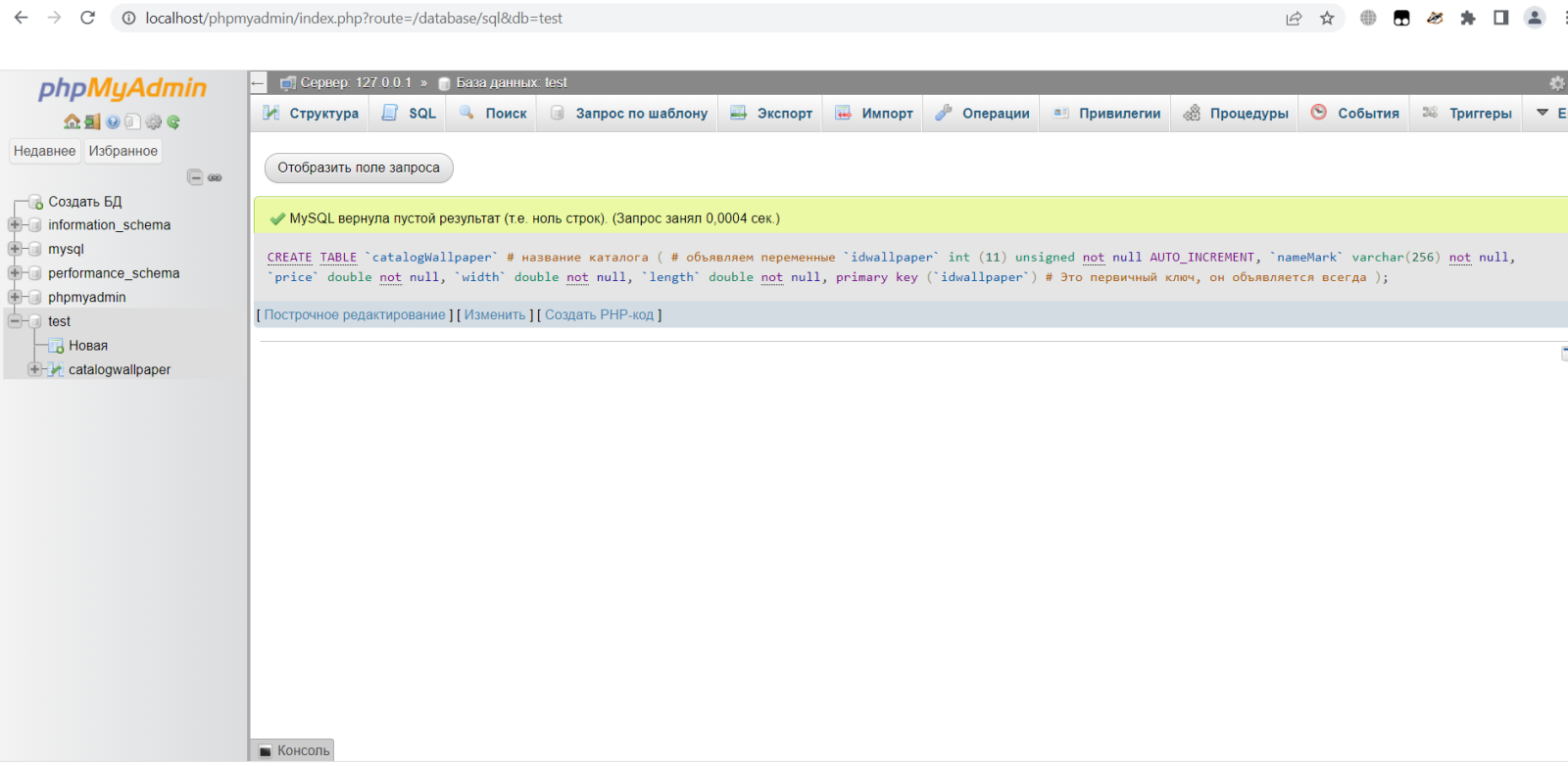
primary key (`idwallpaper`) # Это первичный ключ, он объявляется всегда

);

**Затем нажать «Вперёд»**



**При безошибочном вводе получится следующий результат, появятся каталоги под базой «test»**

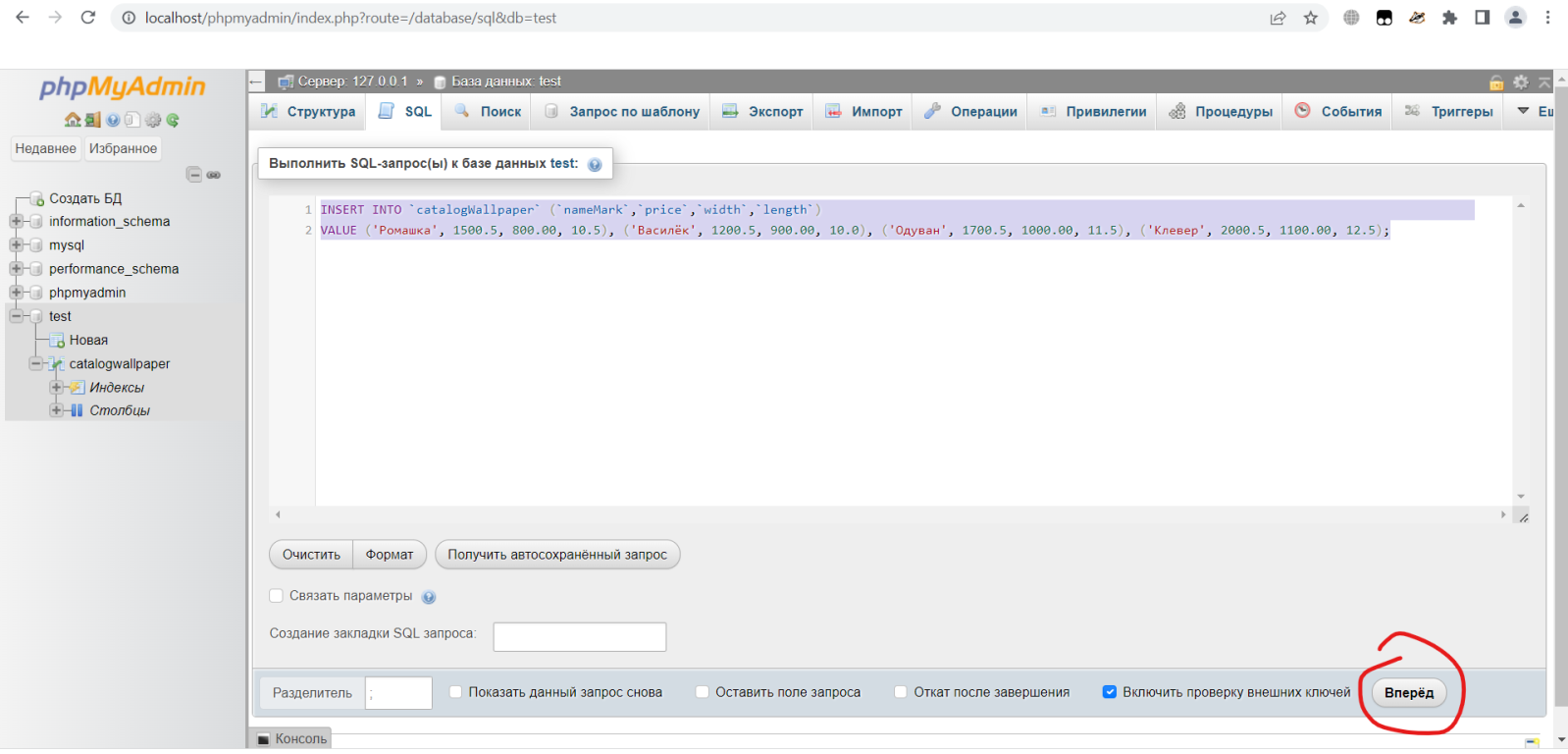


1. **Затем снова нажать на SQL и выполнить следующий запрос** (обратить внимание на кавычки ` ` - это обратные кавычки на кнопке клавиатуры «ё ~»):

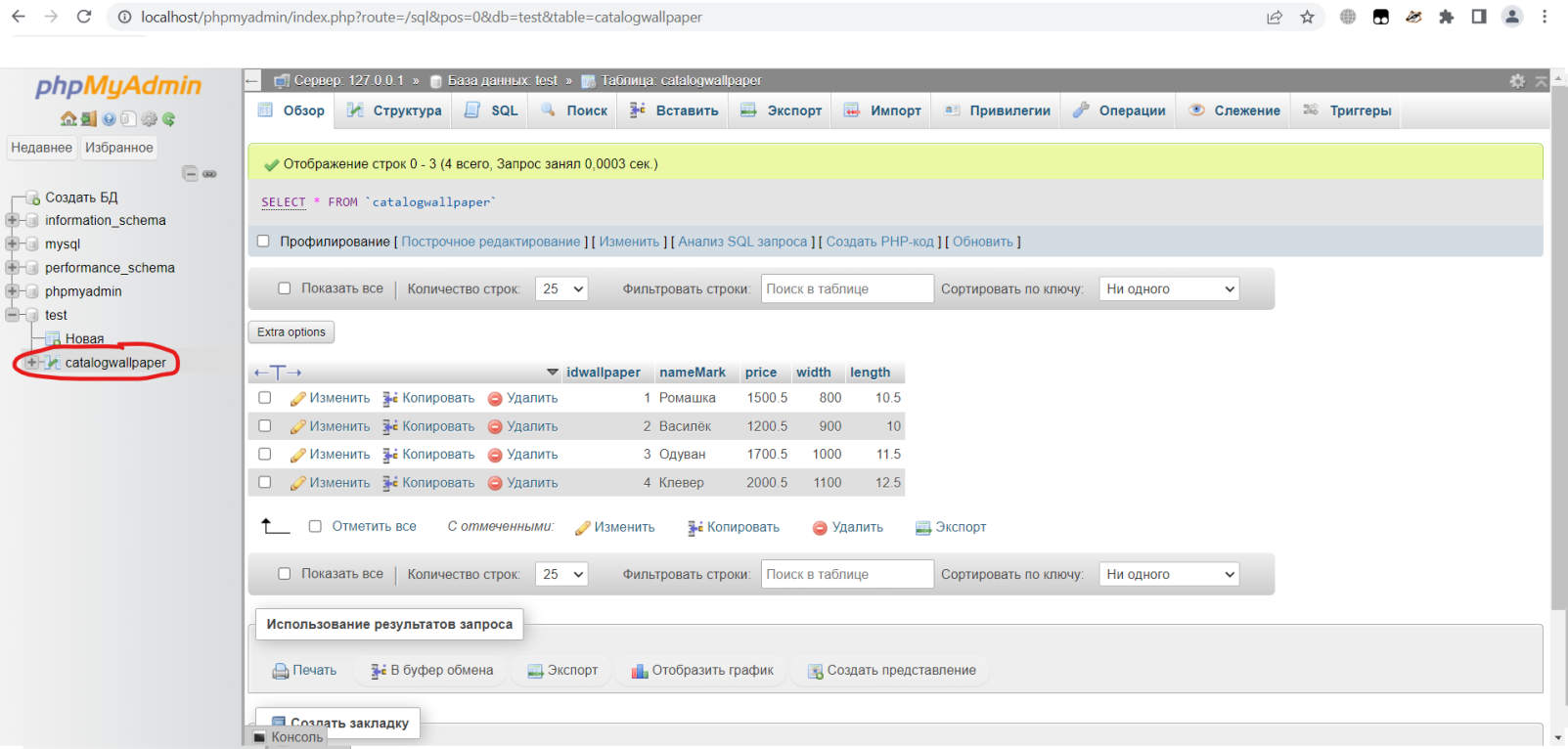
INSERT INTO `catalogWallpaper` (`nameMark`,`price`,`width`,`length`)

VALUE ('Ромашка', 1500.5, 800.00, 10.5), ('Василёк', 1200.5, 900.00, 10.0), ('Одуван', 1700.5, 1000.00, 11.5), ('Клевер', 2000.5, 1100.00, 12.5);

Далее жмём «Вперёд», а затем на каталог ` catalogWallpaper `

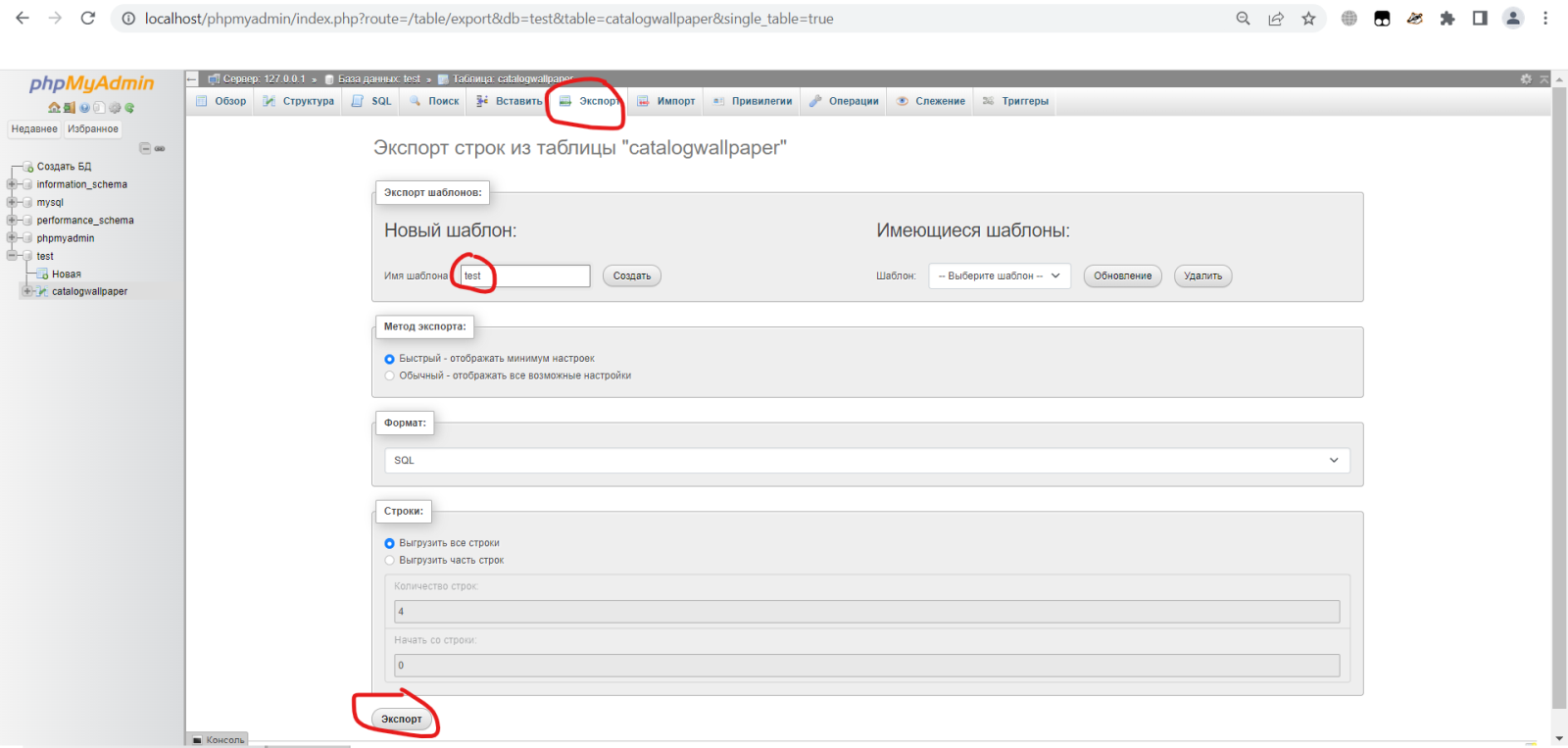


Получим следующий результат:

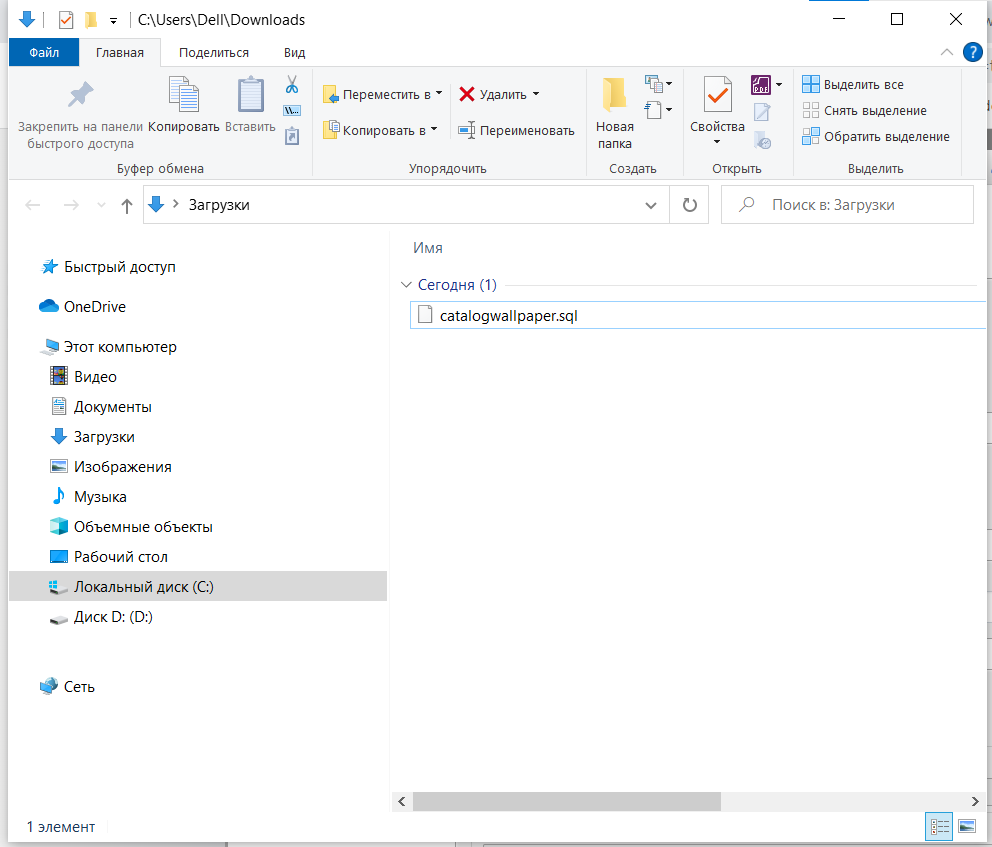


**База данных успешно создана**

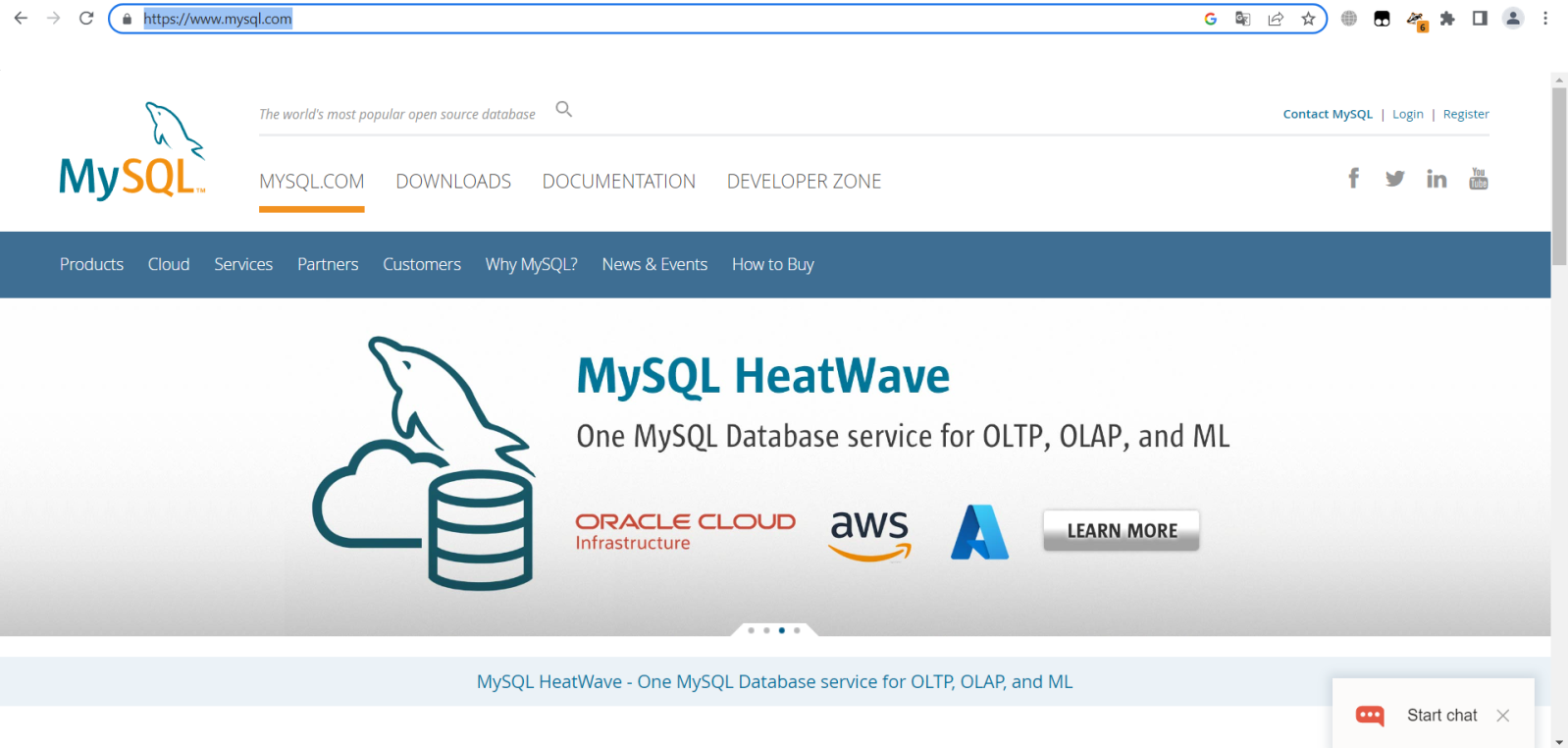
1. **Теперь базу возможно экспортировать и импортировать**
2. **Экспортируем: нажать «Экспорт», написать имя базы данных, нажать «Экспорт»**



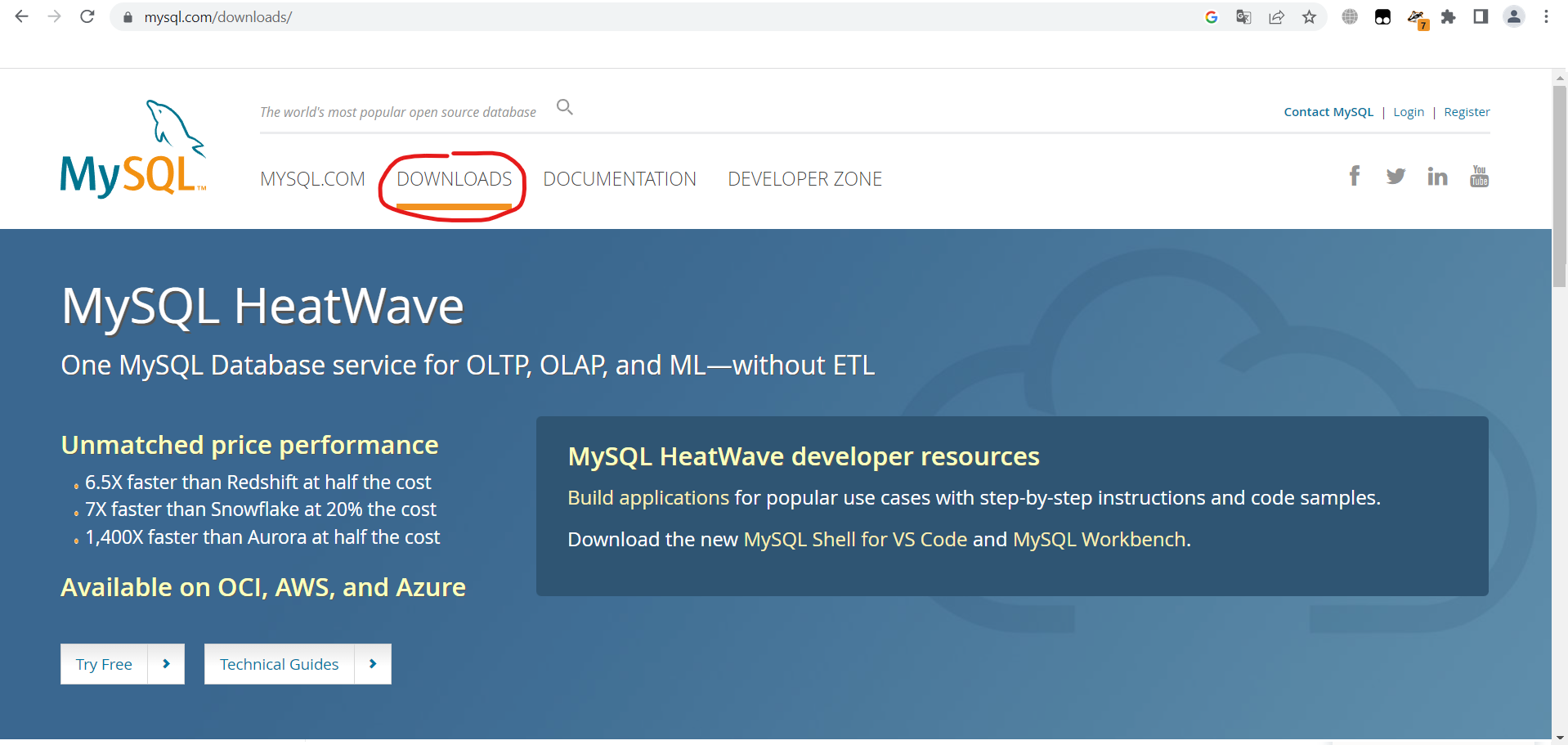
**Скачанный файл отобразится в «Загрузках», его нужно переместить в папку с файлами программы СPP**

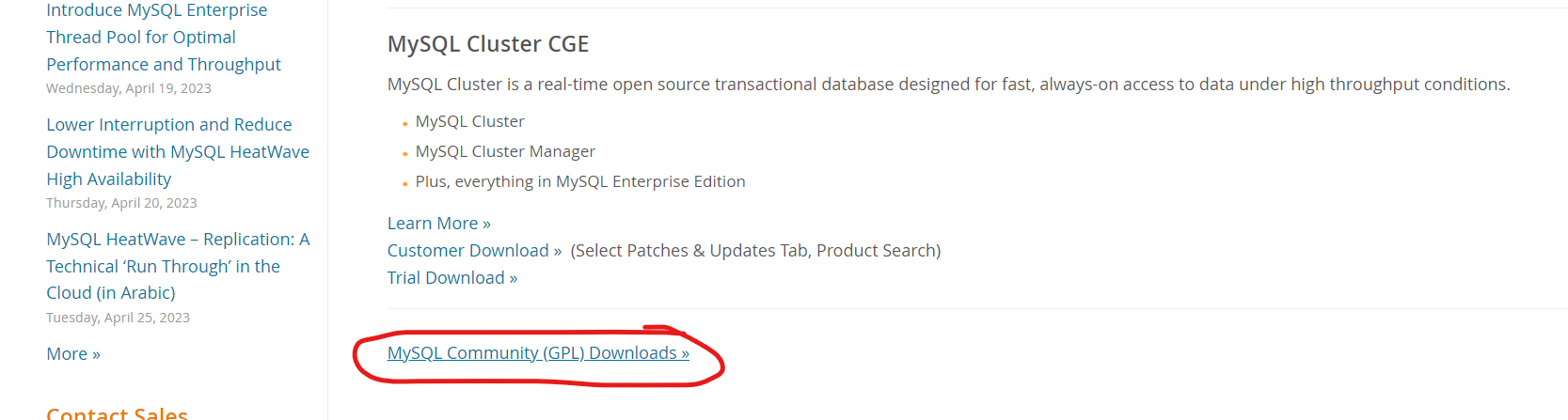
****

1. **Создадим коннектор С++ с базой данных MySQL (идём на** [**https://www.mysql.com/**](https://www.mysql.com/)**)**

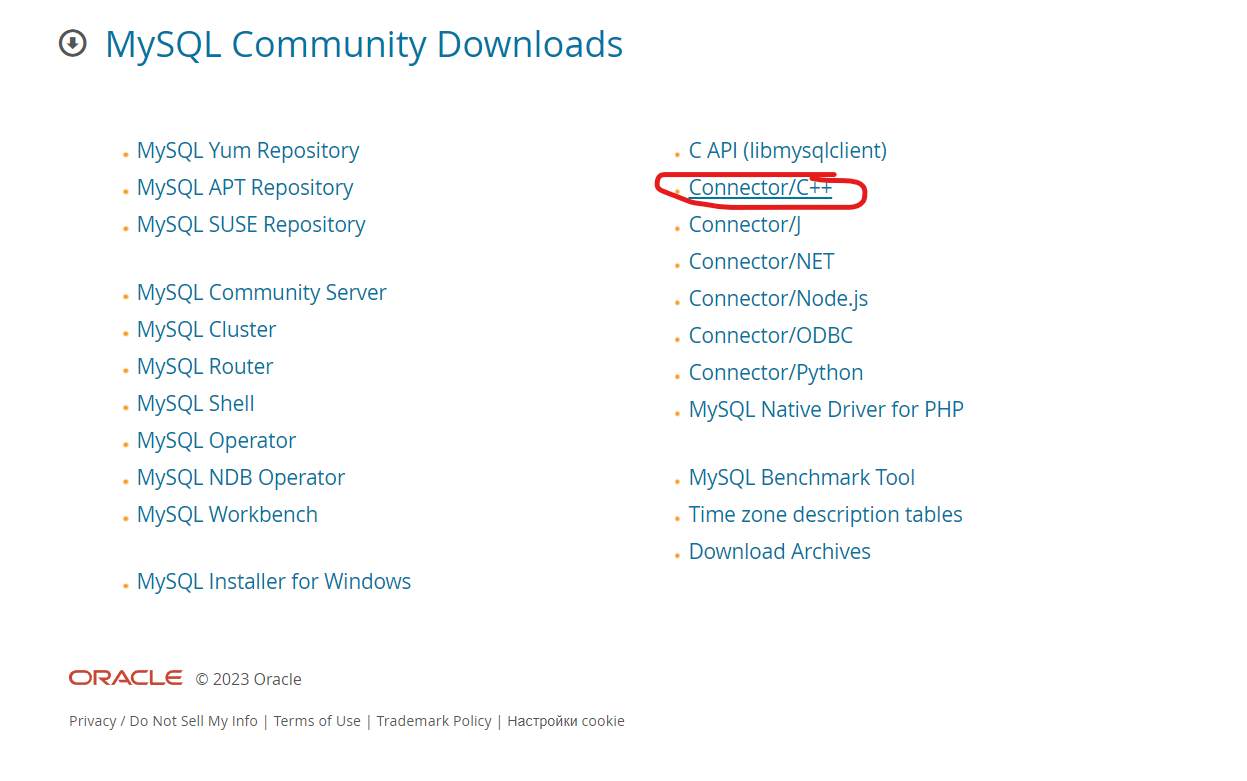
****

1. **Переходим в «Downloads» и выбираем** [MySQL Community (GPL) Downloads »](https://dev.mysql.com/downloads/)

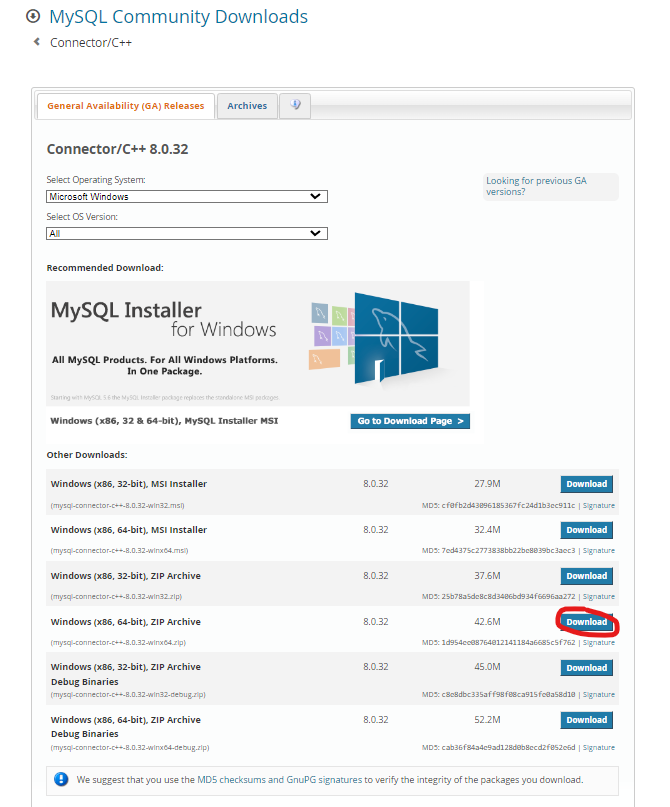


****

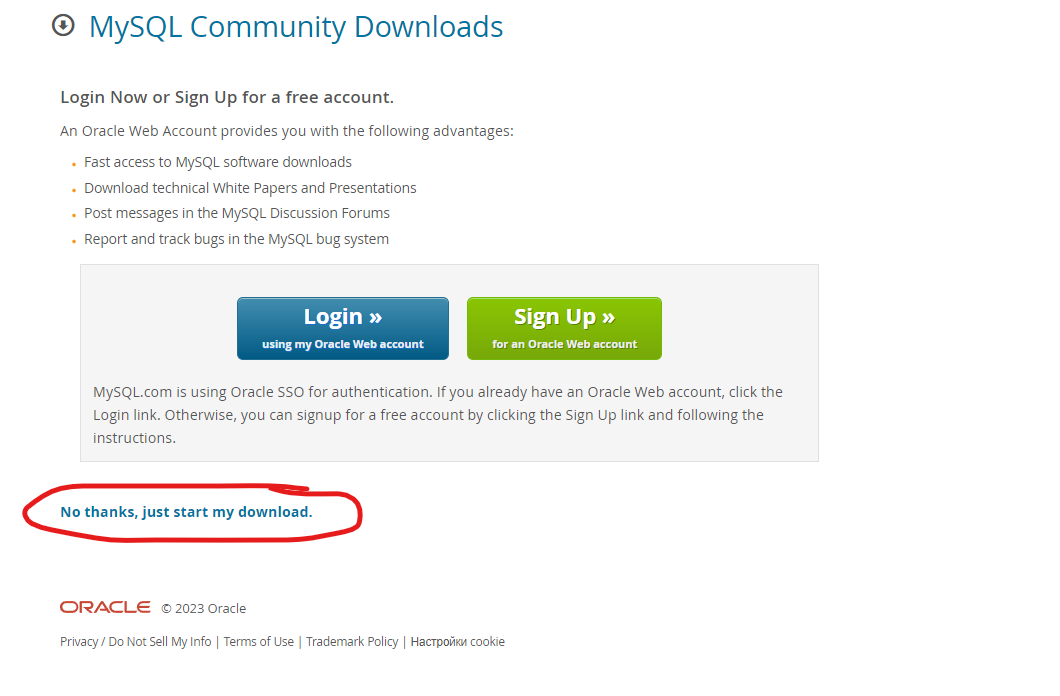
1. **Выбираем «Connector/С++»**

****

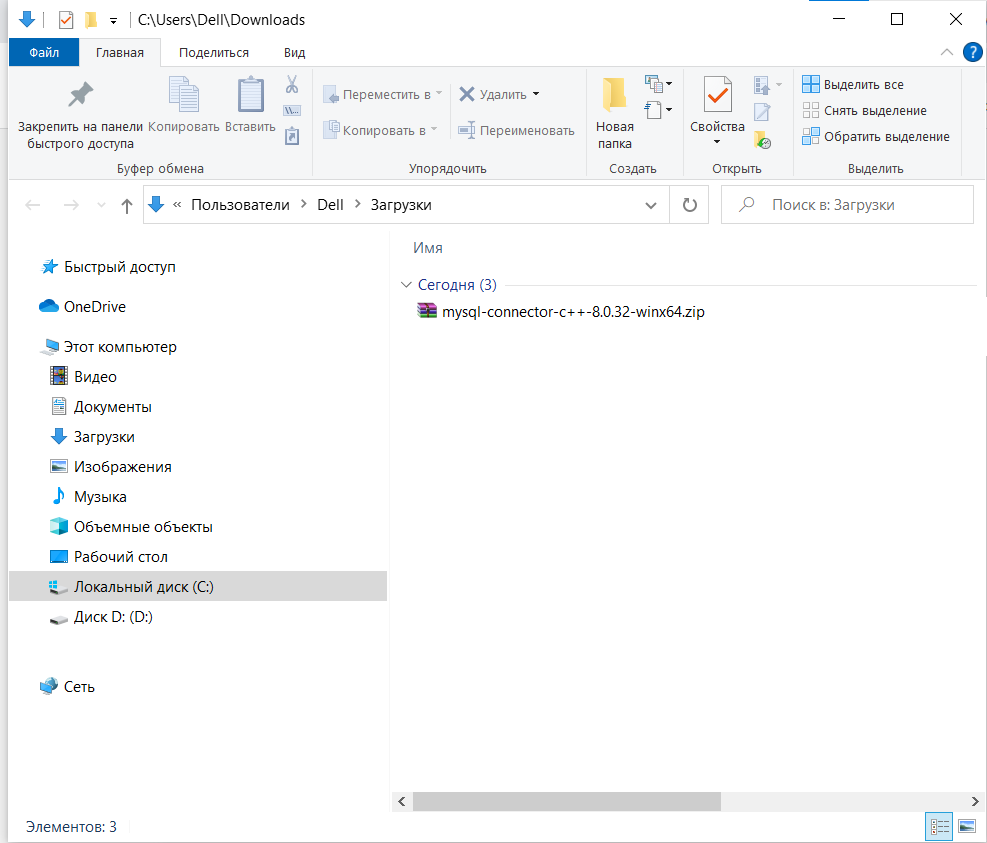
1. **Выбираем подходящую версию**

****

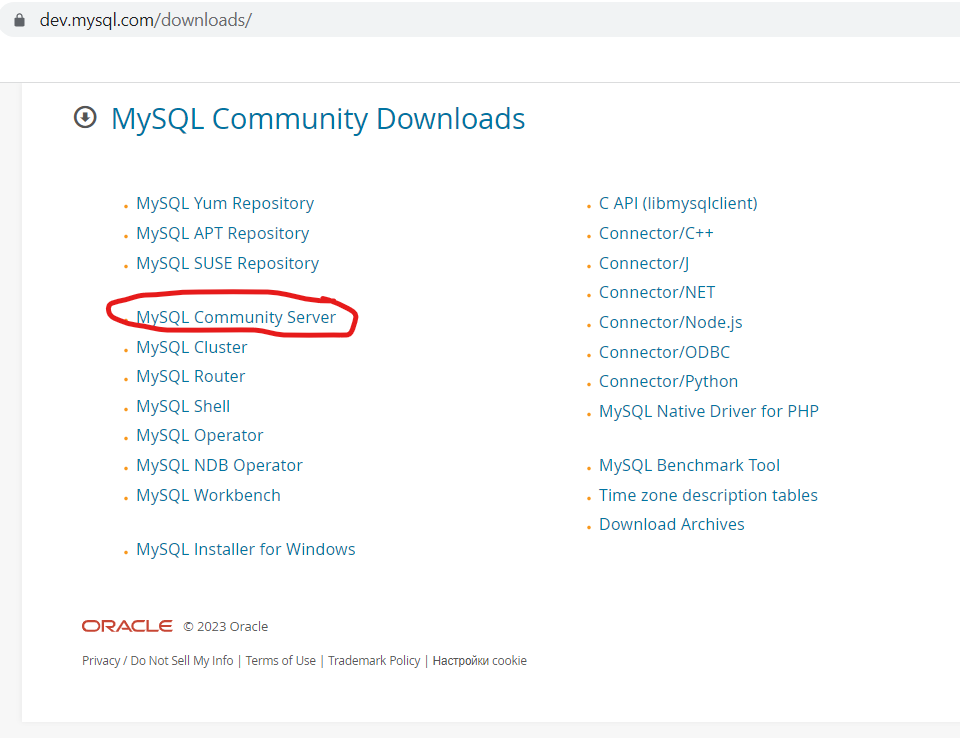
1. **Скачиваем её**

****

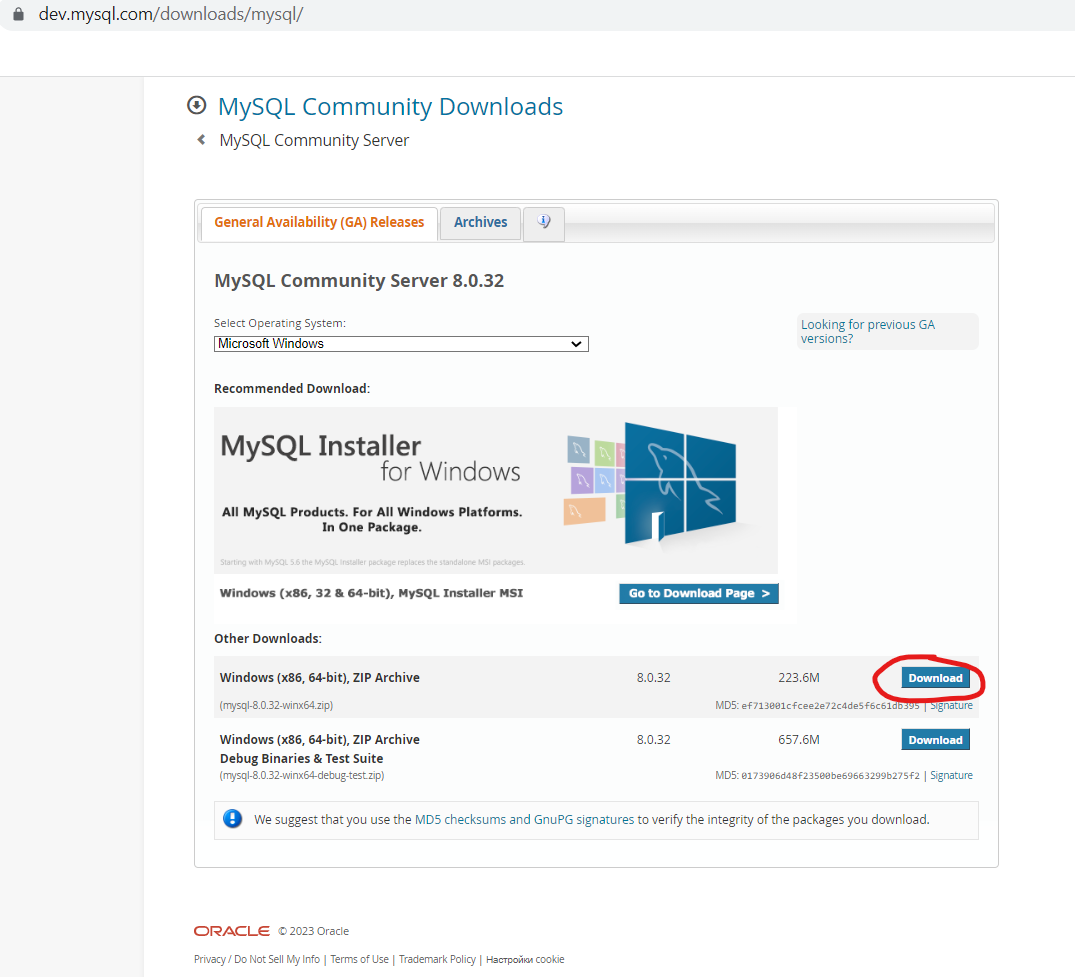
1. **Скачанный архив отобразится в «Загрузках», его также нужно переместить в папку с файлами программы СPP и распаковать там**



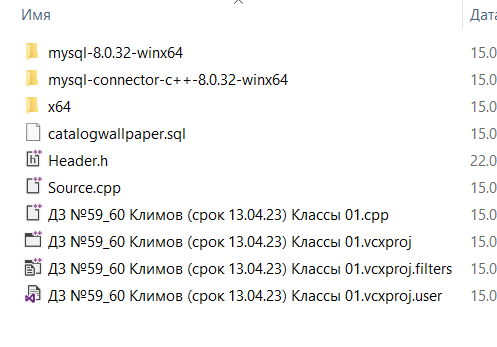
1. **Переходим в «Downloads» и выбираем** [**MySQL Community (GPL) Downloads »**](https://dev.mysql.com/downloads/)
2. **Выбрать MySQL Community Server**



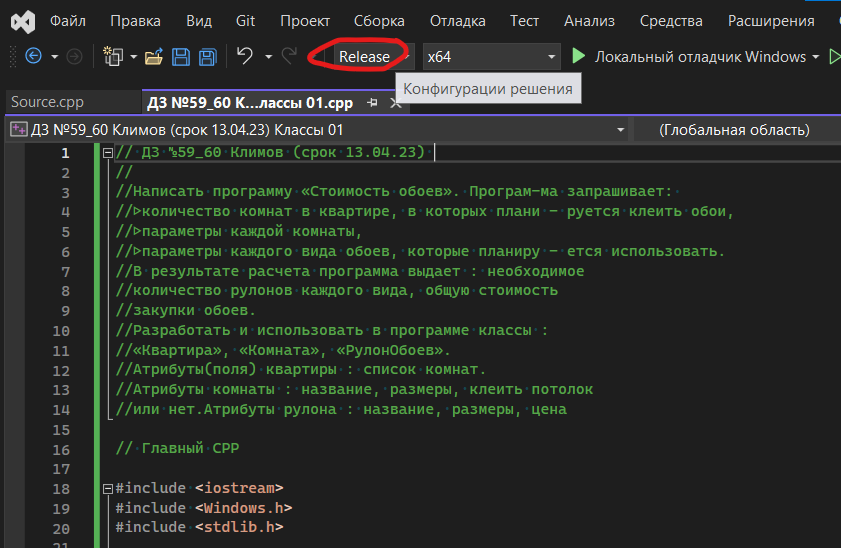
1. **Выбираем подходящую версию дистрибутива и скачиваем её**



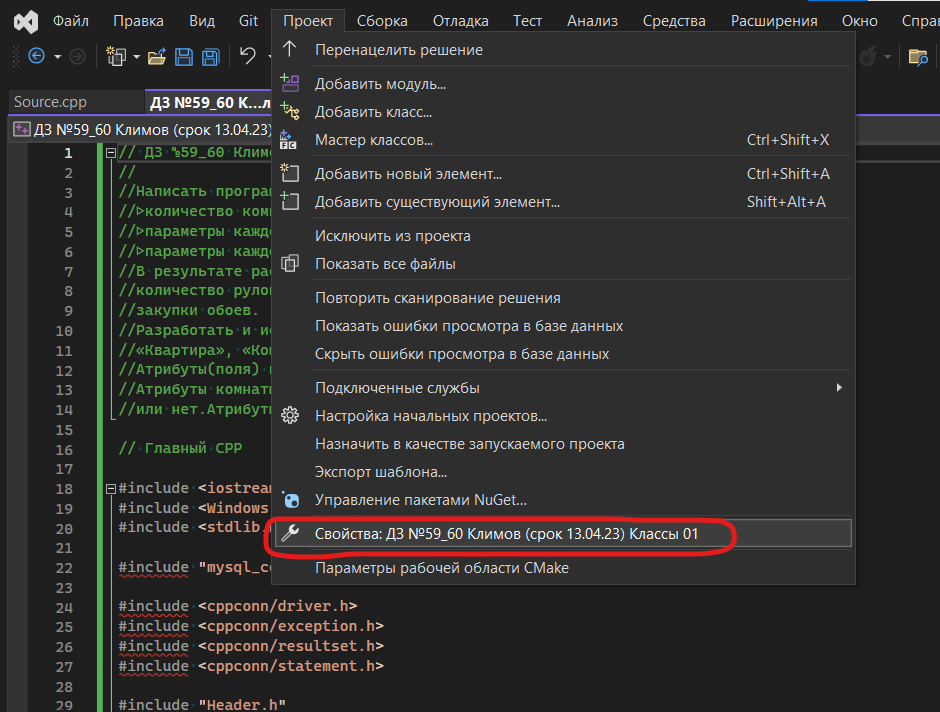
1. **Разархивируем рядом с папкой коннектора в корне нашей программы (где СPP)**

****

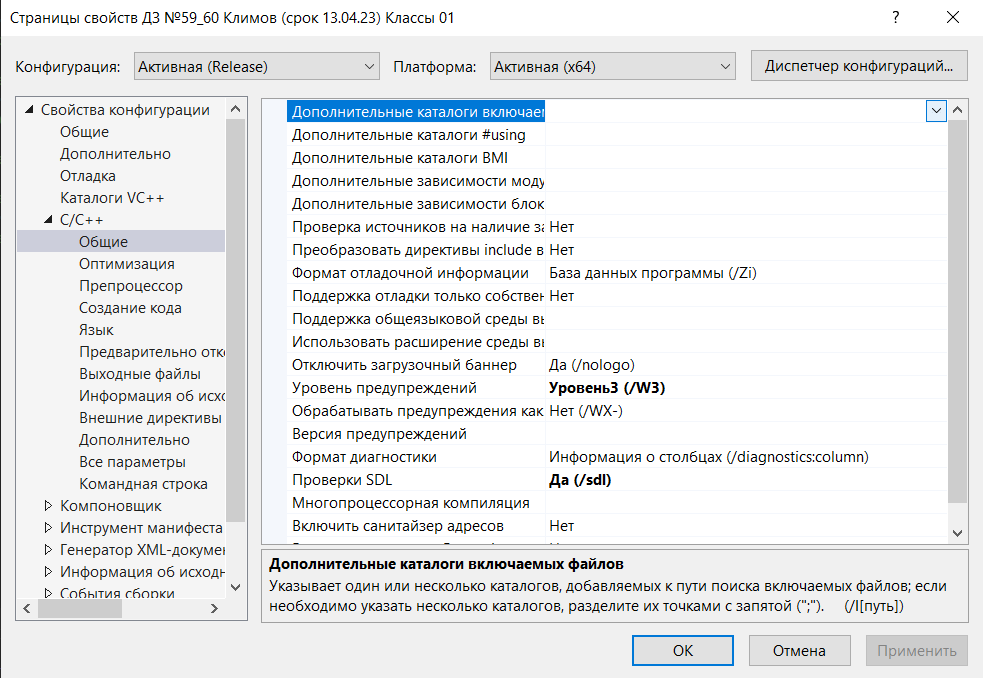
1. копируем все файлы из **D:\...\mysql-connector-c++-8.0.32-winx64\lib64\vs14** (папка с установленным коннектором) в **D:\...\mysql-8.0.32-winx64\lib** (папка с сервером).
2. Перекидывания файлов окончены, теперь настроим сам С++:
3. **Следующим этапом настроим С++ в Visual Studio**
4. **Конфигурацию решения нужно поменять с Debug на Release. Обязательно (!!!). Если выбрать стандартный Debug, то нужно файлы сохранять в другие директории.**
5. **Скопировать файлы libcrypto-1\_1-x64.dll, libssl-1\_1-x64.dll из коннектора** **D:\...\mysql-connector-c++-8.0.32-winx64\lib64 в папку с .exe нашей программы D:\...\x64\Release (у меня выпадала ошибка при компиляции без этого)**

****

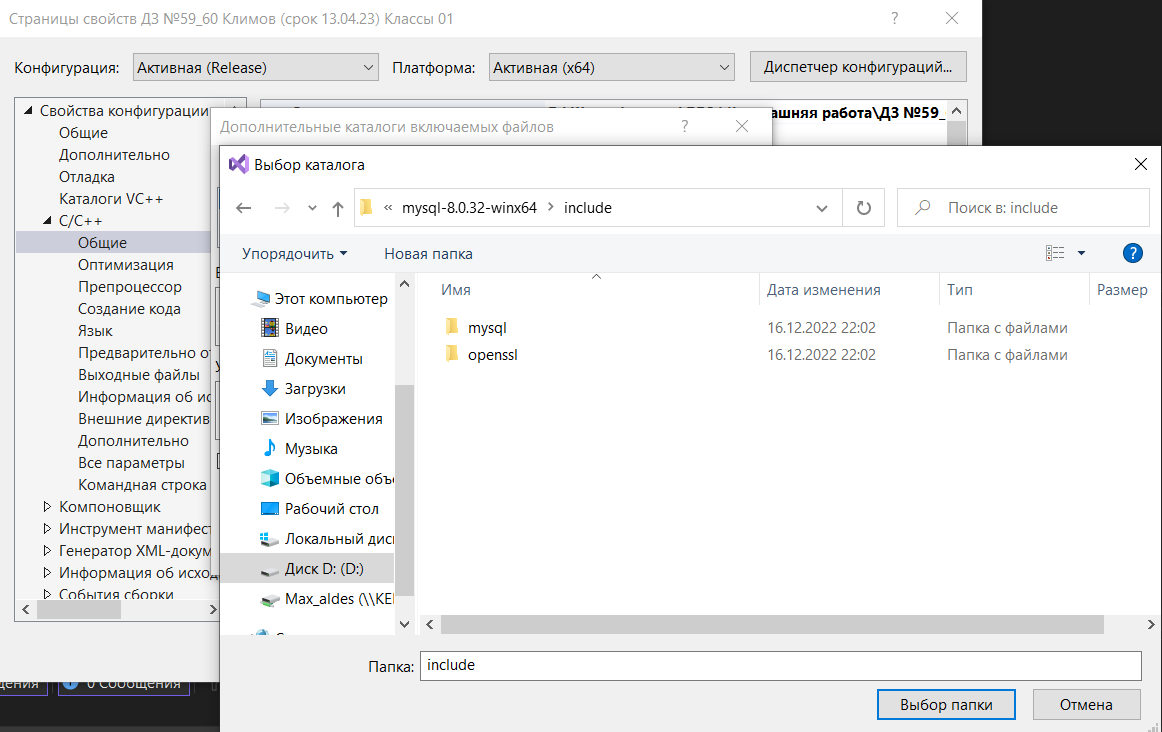
1. **Затем: Проект -> Свойства**

****

1. **С/С++ -> Общие -> Дополнительные каталоги включаемых файлов -> Выпадающее меню - > Изменить**

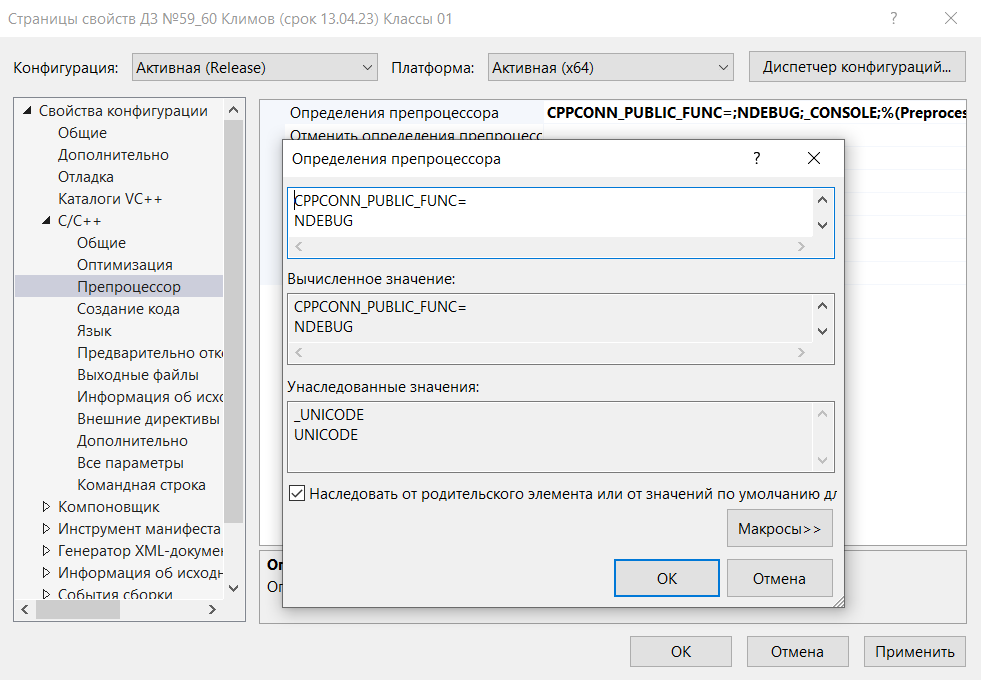
****

1. **Выбираем папку «include» из папки с сервером D:\...\mysql-8.0.32-winx64\include –>выбор папки**

****

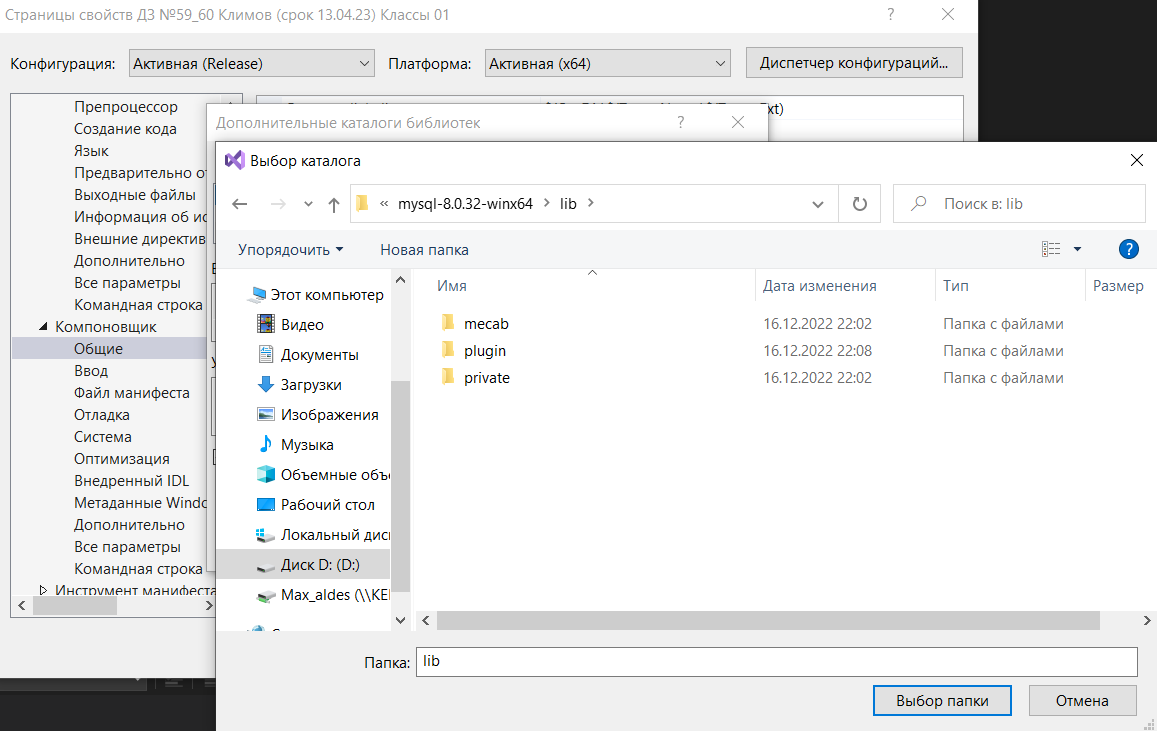
1. **Следующим этапом записываем CPPCONN\_PUBLIC\_FUNC=:**

**С/С++ -> Препроцессор - > Определение препроцессора - > Изменить – Вписать CPPCONN\_PUBLIC\_FUNC= в верхнее поле над всеми записями -> ok**

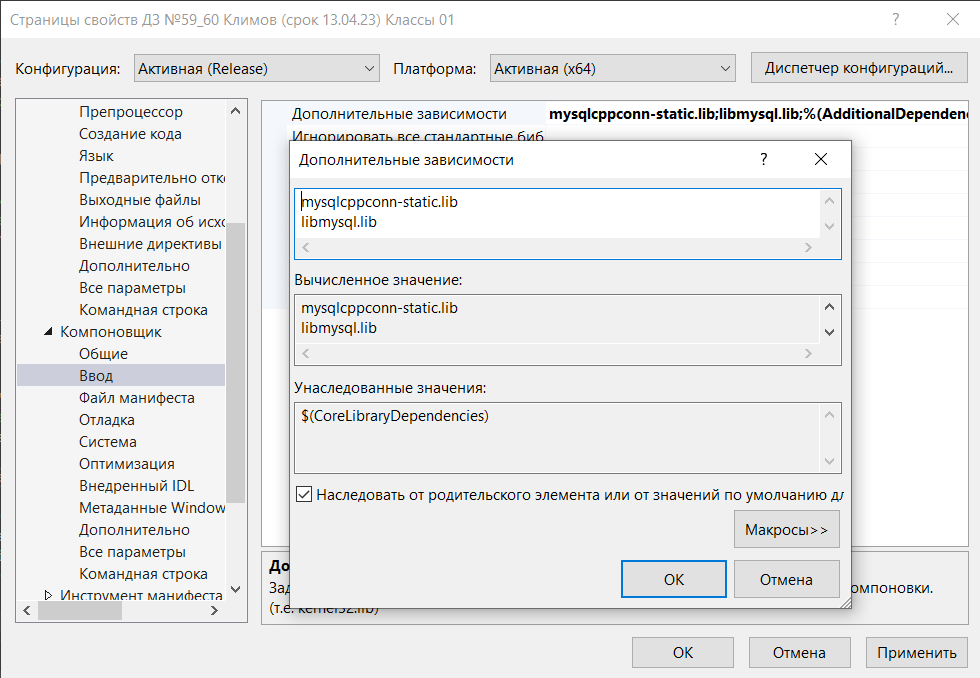
****

1. **Компоновщик-> Общие -> Дополнительные каталоги библиотек -> Выпадающее меню - > Изменить-> выбрать папку lib в папке с сервером**
2. **Выбираем папку «lib»**

**D:\...\mysql-8.0.32-winx64\lib -> выбор папки -> ok. Настройки не закрываем**

****

1. **Следующим этапом выбираем Компоновщик-> Ввод -> Дополнительные зависимости -> Выпадающее меню - > Изменить-> вставить в верхнее поле 2 значения mysqlcppconn-static.lib и libmysql.lib. -> ok -> применить -> ok**

****

1. **Сохраняем свойства проекта Применить->ok**
2. **Далее все, что остается это дописать в проект #include<mysql.h>, и можно работать!**
3. **Теперь нужно вставить в программу примерно вот такой код для вывода нужных значений из базы данных в консоль программы:**

#include <iostream>

#include <Windows.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <mysql.h>

#include "Header.h"

#include "Source.cpp"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

//соединение с базой данный MySQL и вывод оттуда данных

cout<<"Вывод значений базы данных обоев: "<<"\n";

MYSQL\* conn;

MYSQL\_RES\* res;

MYSQL\_ROW row;

int i = 0;

// Получаем дескриптор соединения

conn = mysql\_init(NULL);

if (conn == NULL)

{

// Если дескриптор не получен – выводим сообщение об ошибке

fprintf(stderr, "Error: can'tcreate MySQL-descriptor\n");

//exit(1); //Если используется оконное приложение

}

// Подключаемся к серверу

if (!mysql\_real\_connect(conn, "localhost", "root", "", "test", NULL, NULL, 0))

{

// Если нет возможности установить соединение с сервером

// базы данных выводим сообщение об ошибке

fprintf(stderr, "Error: can'tconnecttodatabase %s\n", mysql\_error(conn));

}

else

{

// Если соединение успешно установлено выводим фразу - "Success!"

fprintf(stdout, "Соединение с базой MySQL успешно!\n");

}

mysql\_set\_character\_set(conn, "utf8");

//Смотрим изменилась ли кодировка на нужную, по умалчанию идёт latin1

cout << "кодировка: " << mysql\_character\_set\_name(conn) << endl;

mysql\_query(conn, "SELECT idwallpaper, nameMark FROM catalogwallpaper"); //Делаем запрос nameMark к таблице по имени catalogwallpaper =)

if (res = mysql\_store\_result(conn)) {

while (row = mysql\_fetch\_row(res)) {

for (i = 0; i < mysql\_num\_fields(res); i++) {

cout << row[i] << "\n"; //Выводим все что есть в базе через цикл

}

}

}

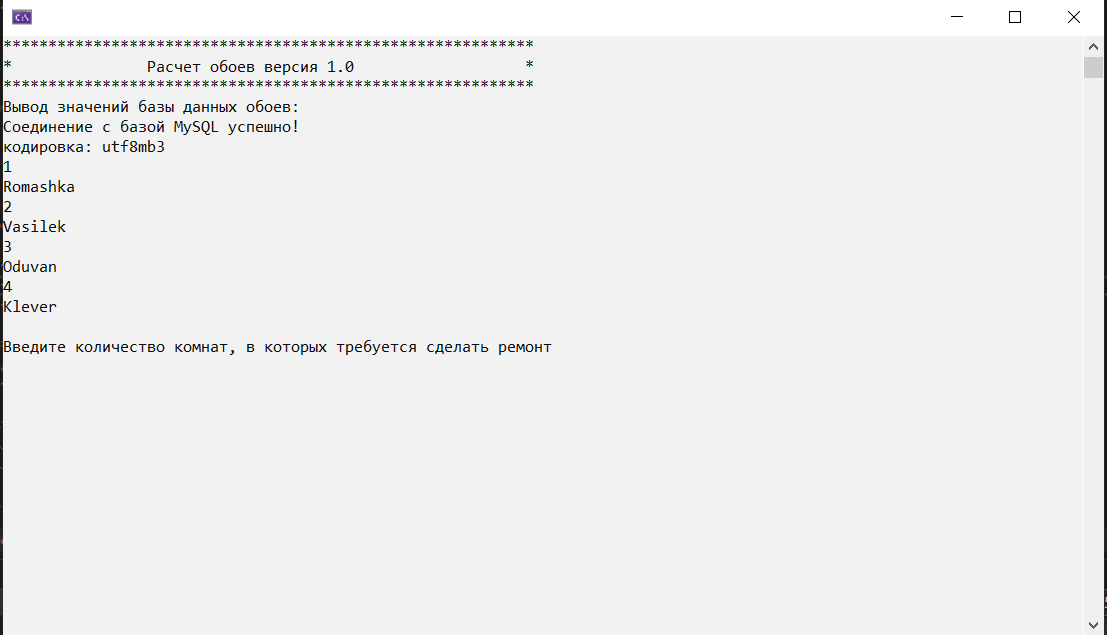
else fprintf(stderr, "%s\n", mysql\_error(conn));

// Закрываем соединение с сервером базы данных

mysql\_close(conn);

**}**

**Результат компиляции**

****

**PS проблемы с отображением кириллицы не устранены, поэтому пришлось исправить значения имен в базе данных MySQL на транслит**