

Самостоятельная работа №9.1. Работа с СУБД MySQL (Базовый уровень)

Установка и настройка MySQL

Для входа в контейнер, созданный в самостоятельной работе №1, выполняется команда, запускающая в контейнере оболочку bash:

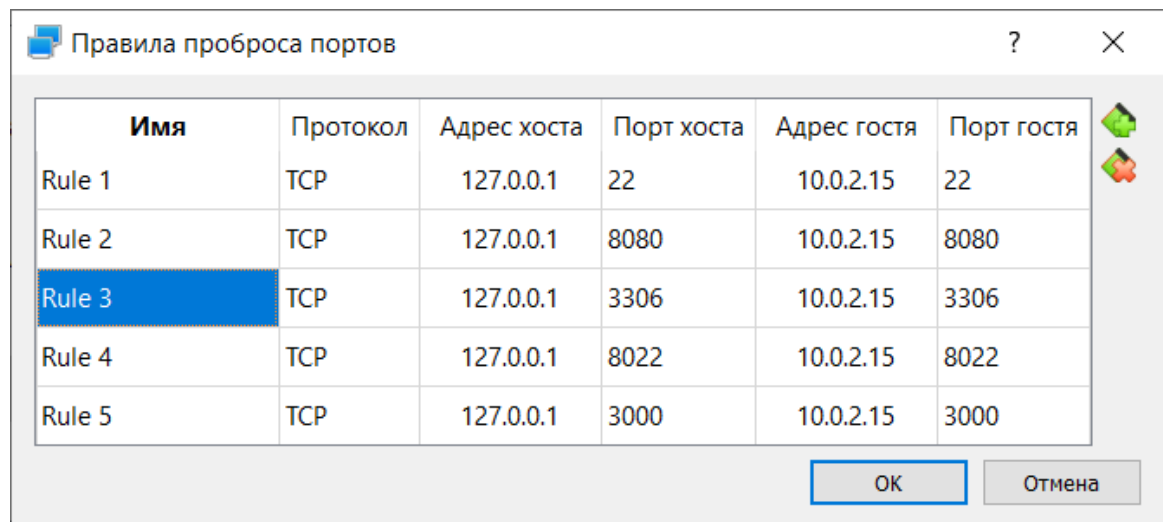
```
docker exec -it php_apache /bin/bash
```

Вход внутрь контейнера выполняется с использованием его имени, а не идентификатора (id). Внутри контейнера производится обновление списка репозитория и обновление установленных пакетов:

```
apt-get update && apt-get upgrade -y
```

```
root@ubuntuserver:~# docker exec -it php_apache /bin/bash
root@3ace2e3bf10a:/# apt-get update && apt-get upgrade -y
```

Для работы с сервисами, запущенными внутри виртуальной машины, в ней был проброшен порт 3306 (порт по умолчанию для MySQL и MariaDB):



Порт 3306 также проброшен из контейнера в виртуальную машину.

Установка клиента БД MySQL и сервера БД MySQL из репозитория Ubuntu

```
apt-get install mysql-client mysql-server -y
```

```
root@3ace2e3bf10a:/var/www/html# apt-get install mysql-client mysql-server -y
```

После успешного завершения установки выполняется проверка того, запущена ли служба MySQL, и, если служба не запущена, производится её запуск:

```
service mysql status
```

```
service mysql start
```

```
root@3ace2e3bf10a:/var/www/html# service mysql status
* MySQL is stopped.
root@3ace2e3bf10a:/var/www/html# service mysql start
* Starting MySQL database server mysqld
No directory, logging in with HOME=/
```

[OK]

Для проверки возможности локального подключения к БД MySQL производится вход в сервер MySQL:

```
mysql -h hostname -u root -p
```

Если в качестве hostname используется стандартный localhost, то флаг **-h** можно не писать, тогда команда будет следующая:

```
mysql -u root -p
```

-u — флаг, означающий имя пользователя (в данном случае, root).

-p — флаг, означающий пароль.

-h — флаг, позволяющий задать хост базы данных (hostname).

Пароль для root еще не был установлен, поэтому на запрос ввода пароля просто нажимается Enter и осуществляется вход в режим работы с БД MySQL через терминал.

Для выхода из данного режима необходимо ввести команду:

```
quit
```

или

```
exit
```

```
root@3ace2e3bf10a:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.37-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> exit
Bye
```

Запуск стандартного скрипта настройки безопасности сервера БД MySQL:

`mysql_secure_installation`

В появляющемся текстовом диалоге выбираются следующие ответы:

- **запрос о запуске VALIDATE PASSWORD PLUGIN:** No;
- **пароль (и повторный ввод пароля):** mysqlpwd;
- **вопрос об удалении анонимных пользователей (Remove anonymous users?):** Yes;
- **вопрос о запрете удаленного доступа для пользователя root (Disallow root login remotely?):** No;
- **вопрос об удалении тестовой базы данных (Remove test database and access to it?):** Yes;
- **вопрос о перезагрузке таблицы привилегий (Reload privilege tables now?):** Yes.

```
root@3ace2e3bf10a:/var/www/html# mysql_secure_installation
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: No
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Yes
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : No
... skipping.
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No)
: Yes
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Yes
Success.

All done!
```

По умолчанию кодировкой в MySQL является latin1. Для того, чтобы сменить кодировку по умолчанию на поддерживающую кириллицу (UTF-8 или Windows-1251), её нужно указать в конфигурационном файле MySQL. Для этого он открывается в текстовом редакторе nano:

```
nano /etc/mysql/my.cnf
```

В конфигурационный файл добавляются следующие строки:

```
[mysqld]
```

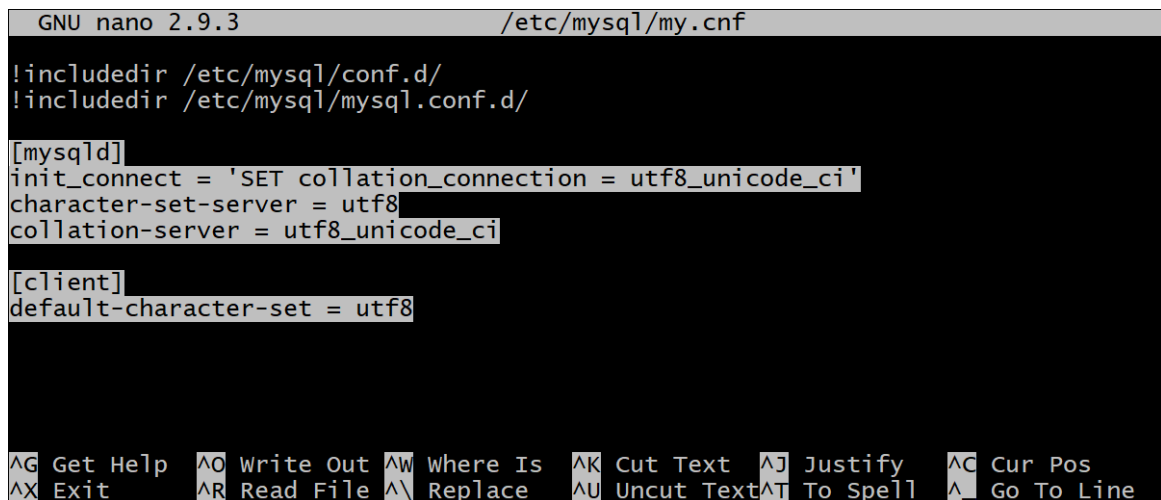
```
init_connect = 'SET collation_connection = utf8_unicode_ci'
```

```
character-set-server = utf8
```

```
collation-server = utf8_unicode_ci
```

```
[client]
```

```
default-character-set = utf8
```



```
GNU nano 2.9.3 /etc/mysql/my.cnf
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/

[mysqld]
init_connect = 'SET collation_connection = utf8_unicode_ci'
character-set-server = utf8
collation-server = utf8_unicode_ci

[client]
default-character-set = utf8

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Изменения в файле конфигурации сохраняются. После выполнения настройки безопасности и установки кодировки, поддерживающей кириллицу, сервер (служба mysql) перезапускается:

```
service mysql restart
```

Вход в режим работы с БД MySQL через терминал:

```
mysql -u root -p
```

Вводится тот пароль, который был установлен во время настройки безопасности (mysqlpwd).

Для верификации настроек кодировки выполняются следующие команды:

```
SHOW VARIABLES LIKE 'character%';
```

```
SHOW VARIABLES LIKE 'collation%';
```

Если настройки были произведены верно, то везде будет отображаться кодировка utf8, а не latin1.

Выход из режима работы с БД MySQL через терминал:

quit

или

exit

```
root@3ace2e3bf10a:/# service mysql restart
* Stopping MySQL database server mysqld [ OK ]
* Starting MySQL database server mysqld
No directory, logging in with HOME=/ [ OK ]

root@3ace2e3bf10a:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.37-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SHOW VARIABLES LIKE 'character%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | utf8 |
| character_set_connection | utf8 |
| character_set_database | utf8 |
| character_set_filesystem | binary |
| character_set_results | utf8 |
| character_set_server | utf8 |
| character_set_system | utf8 |
| character_sets_dir | /usr/share/mysql/charsets/ |
+-----+-----+
8 rows in set (0.00 sec)

mysql> SHOW VARIABLES LIKE 'collation%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| collation_connection | utf8_general_ci |
| collation_database | utf8_unicode_ci |
| collation_server | utf8_unicode_ci |
+-----+-----+
3 rows in set (0.01 sec)

mysql> quit
Bye
```

Выход из Docker контейнера:

exit

Если терминал MySQL в ОС Ubuntu еще не был установлен, то выполняется его установка:

```
apt-get update && apt-get install mysql-client-core-8.0 -y
```

```
root@3ace2e3bf10a:/# exit
exit
root@ubuntuserver:~# apt-get update && apt-get install mysql-client-core-8.0 -y
```

Производится попытка удаленного подключения к серверу БД MySQL через внешний терминал ОС Ubuntu:

```
mysql -h localhost -u root -p
```

Вводится тот пароль, который был установлен во время настройки безопасности (mysqlpwd). После ввода пароля выводится сообщение об ошибке, что невозможно соединиться с локальным сервером MySQL через сокет: «*ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)*».

Для исправления этой ошибки производится вход в Docker контейнер:

```
docker exec -it php_apache /bin/bash
```

```
root@ubuntuserver:~# mysql -h localhost -u root -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)
root@ubuntuserver:~# docker exec -it php_apache /bin/bash
root@3ace2e3bf10a:/# |
```

Вход в режим работы с БД MySQL через терминал:

```
mysql -u root -p
```

Вводится тот пароль, который был установлен во время настройки безопасности (mysqlpwd).

Для успешного дистанционного подключения к серверу БД MySQL извне необходимо создать нового пользователя. Это делается при помощи четырёх основных команд:

1. Создание пользователя myuser (пароль password) с возможностью работы с хостом localhost:

```
CREATE USER 'myuser'@'localhost' IDENTIFIED BY 'password';
```

2. Предоставление пользователю myuser всех привилегий для работы с данным хостом:

```
GRANT ALL PRIVILEGES ON *.* TO 'myuser'@'localhost' WITH GRANT OPTION;
```

3. Создание еще одного пользователя с тем же именем myuser (пароль password), но на этот раз, с возможностью работы с любым хостом:

```
CREATE USER 'myuser'@'%' IDENTIFIED BY 'password';
```

4. Предоставление этому пользователю всех привилегий для работы с любым хостом:

```
GRANT ALL PRIVILEGES ON *.* TO 'myuser'@'%' WITH GRANT OPTION;
```

Выход из режима работы с БД MySQL через терминал:

```
quit
```

или

```
exit
```

```
root@3ace2e3bf10a:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.37-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE USER 'myuser'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'myuser'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE USER 'myuser'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'myuser'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
Bye
```

В текстовом редакторе nano открывается конфигурационный файл MySQL:

```
nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
root@3ace2e3bf10a:/# nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

В конфигурационном файле комментируется (в начало строки добавляется символ #) следующая строка:

```
bind-address          = 127.0.0.1
```

```
GNU nano 2.9.3 /etc/mysql/mysql.conf.d/mysqld.cnf Modified
datadir          = /var/lib/mysql
tmpdir           = /tmp
lc-messages-dir  = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
# bind-address    = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size  = 16M
max_allowed_packet = 16M
thread_stack     = 192K
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Также в настройках конфигурационного файла БД MySQL находится строка «socket». Она содержит путь к файлу. В данном случае это:

```
socket = /var/run/mysqld/mysqld.sock
```

```
GNU nano 2.9.3 /etc/mysql/mysql.conf.d/mysqld.cnf
# This will be passed to all mysql clients
# It has been reported that passwords should be enclosed with ticks/quotes
# especially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket location.
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
[mysqld_safe]
socket = /var/run/mysqld/mysqld.sock
nice = 0
[mysqld]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Сокет — это программа, которая позволяет нам взаимодействовать с сервером БД MySQL через клиент. Т.е. клиент обращается к сокету, а сокет уже направляет входящие команды серверу БД.

После сохранения файла конфигурации устанавливаем файлу сокета полные права доступа:

```
chmod -R 777 /var/run/mysqld/mysqld.sock
```

Перезагрузка службы MySQL:

```
service mysql restart
```

Запуск демона SSH и веб-сервера Apache 2:

```
service ssh start; service apache2 start
```


Теперь в Docker контейнере запущен сервер БД MySQL, к контейнеру можно присоединиться по протоколу SSH, а приветственная страница веб-сервера Apache 2 доступна из браузера по адресу localhost:8080.

Выход из Docker контейнера:

exit

```
root@3ace2e3bf10a:/# chmod -R 777 /var/run/mysqld/mysqld.sock
root@3ace2e3bf10a:/# service mysql restart
* Stopping MySQL database server mysqld          [ OK ]
* Starting MySQL database server mysqld
No directory, logging in with HOME=/
[ OK ]
root@3ace2e3bf10a:/# service ssh start; service apache2 start
* Starting OpenBSD Secure Shell server sshd      [ OK ]
* Starting Apache httpd web server apache2
*
root@3ace2e3bf10a:/# exit
exit
root@ubuntuserver:~#
```

Удаленное подключение к серверу БД MySQL через внешний терминал ОС Ubuntu (в этот раз вместо localhost вводится адрес 127.0.0.1):

mysql -h 127.0.0.1 -u myuser -p

Вводится тот пароль (password), который был установлен во время создания нового пользователя myuser.

После того, как проверка удаленного подключения к MySQL произведена успешно, производится выход из режима работы с БД MySQL через терминал:

quit

или

exit

```
root@ubuntuserver:~# mysql -h 127.0.0.1 -u myuser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.37-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> quit
Bye
```

Соединение с базой данных посредством HeidiSQL

На рабочей машине открывается HeidiSQL (<https://www.heidisql.com/>) — инструмент для управления базами данных MariaDB, MySQL, Microsoft SQL, PostgreSQL server и SQLite. В HeidiSQL создаётся новое подключение для доступа к серверу MySQL:

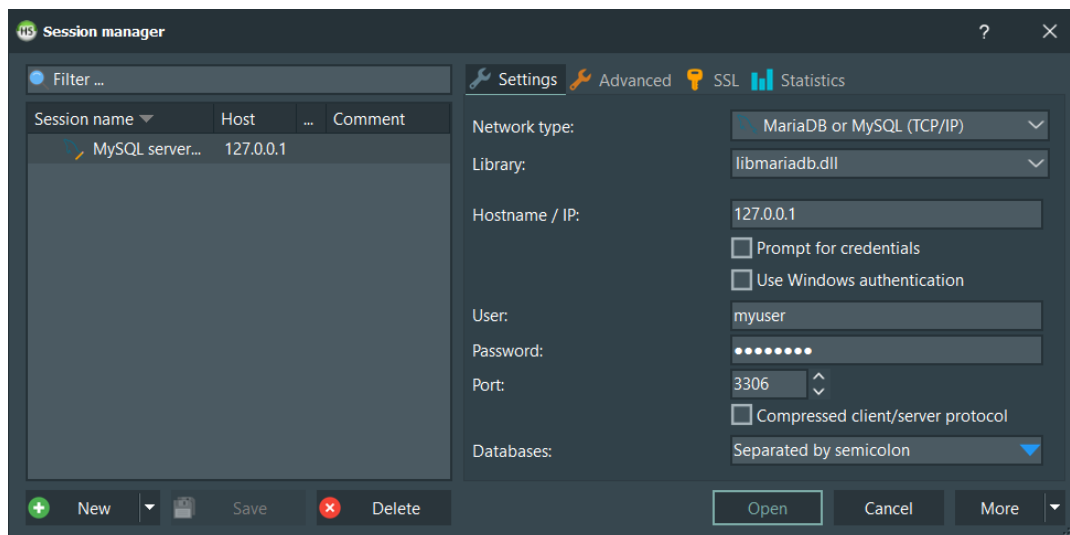
- **network type (тип сети):** MariaDB or MySQL (TCP/IP);
- **hostname / IP (имя хоста/IP):** 127.0.0.1 или localhost;
- **user (пользователь):** myuser;
- **port (порт):** 3306;
- **password (пароль):** password.

Тип сети — база данных MySQL, которая будет использоваться при выполнении запросов и стек протоколов передачи данных TCP/IP.

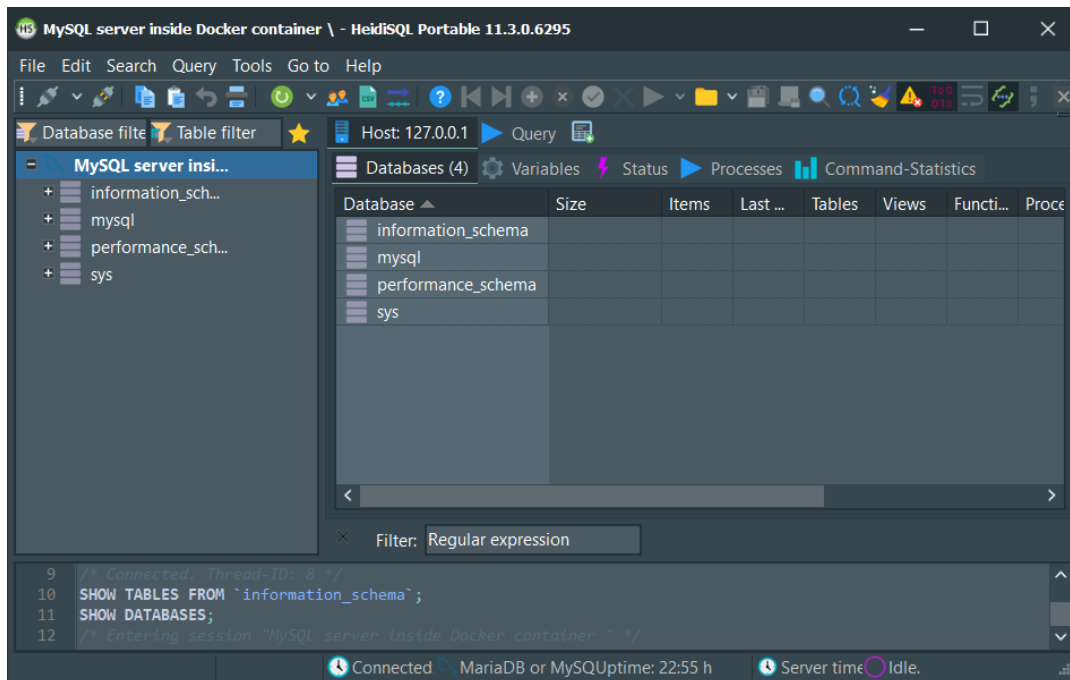
Имя хоста — это имя хоста (например, localhost) или IP-адрес сервера MySQL.

Пользователь и пароль указывают учётные данные для доступа к серверу MySQL.

Порт 3306 — это порт по умолчанию для MySQL и MariaDB. Он остаётся без изменений, поскольку из контейнера в виртуальную машину и из виртуальной машины проброшен порт 3306.

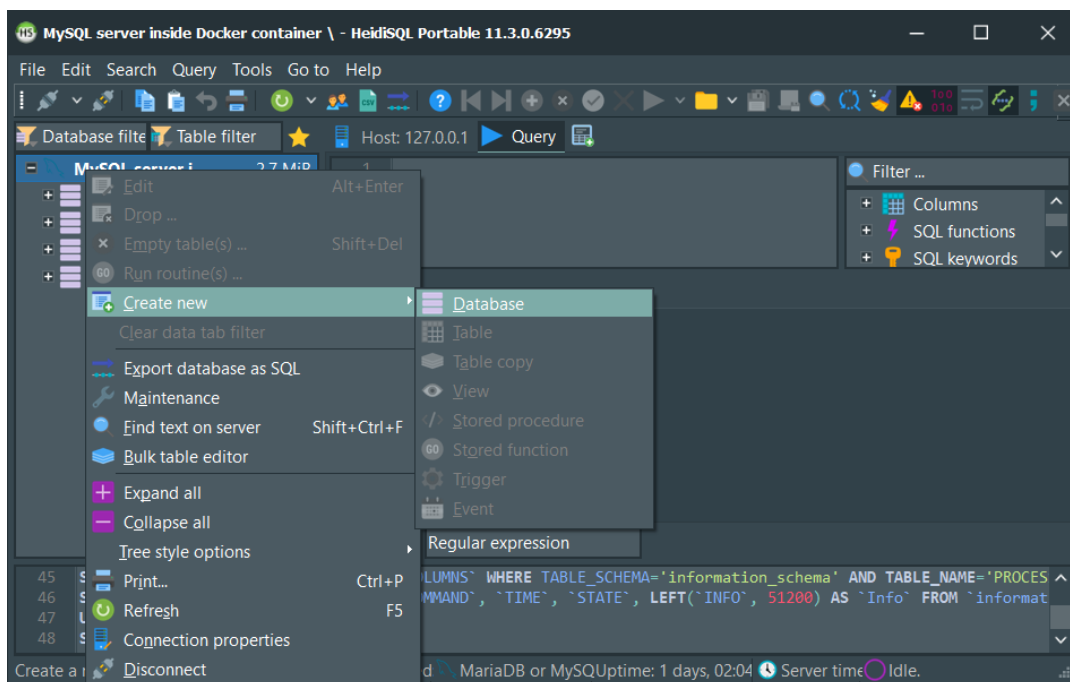


После нажатия на кнопку «Open» («Открыть») осуществляется соединение с базой данных:



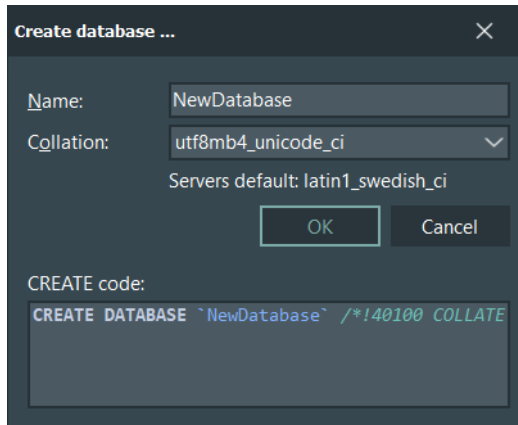
Создание базы данных

В левой части главного окна программы выбирается установленное соединение и нажимается ПКМ (правая кнопка мыши), в появившемся контекстном меню выбирается «Create new» -> «Database» («Создать» -> «База данных»):



В поля ввода открывшегося диалогового окна «Create database» («Создать базу данных») вносятся следующие данные:

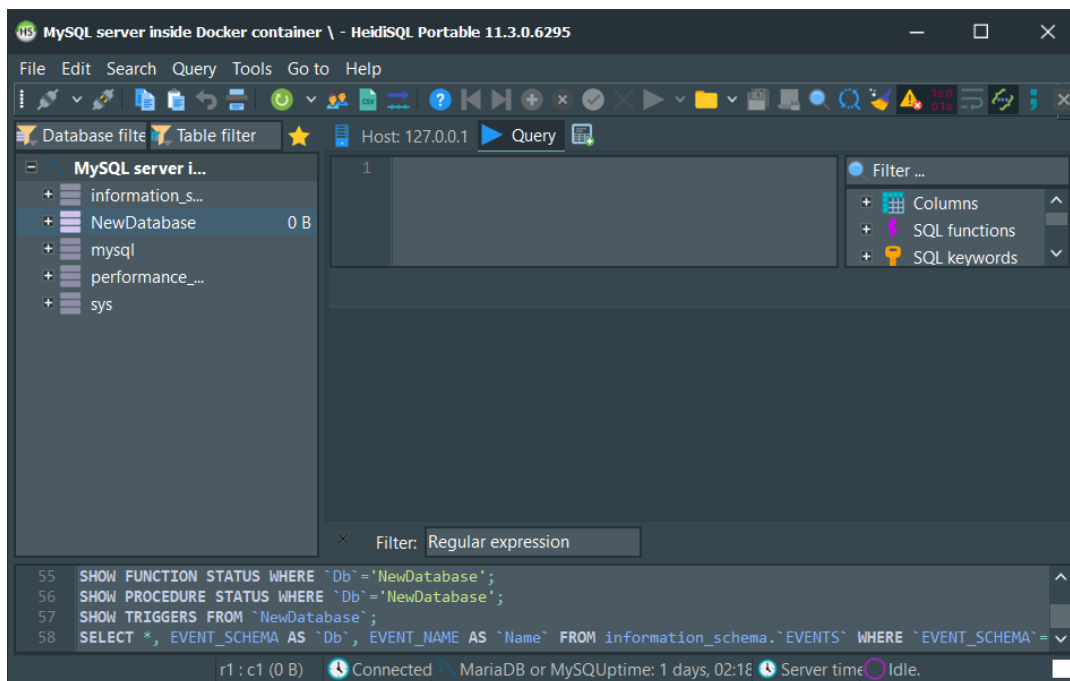
- **Name (Имя):** NewDatabase.
- **Collation (Сопоставление):** utf8mb4_unicode_ci.



Выбранные параметры формируют следующие команды SQL, создающие БД NewDatabase:

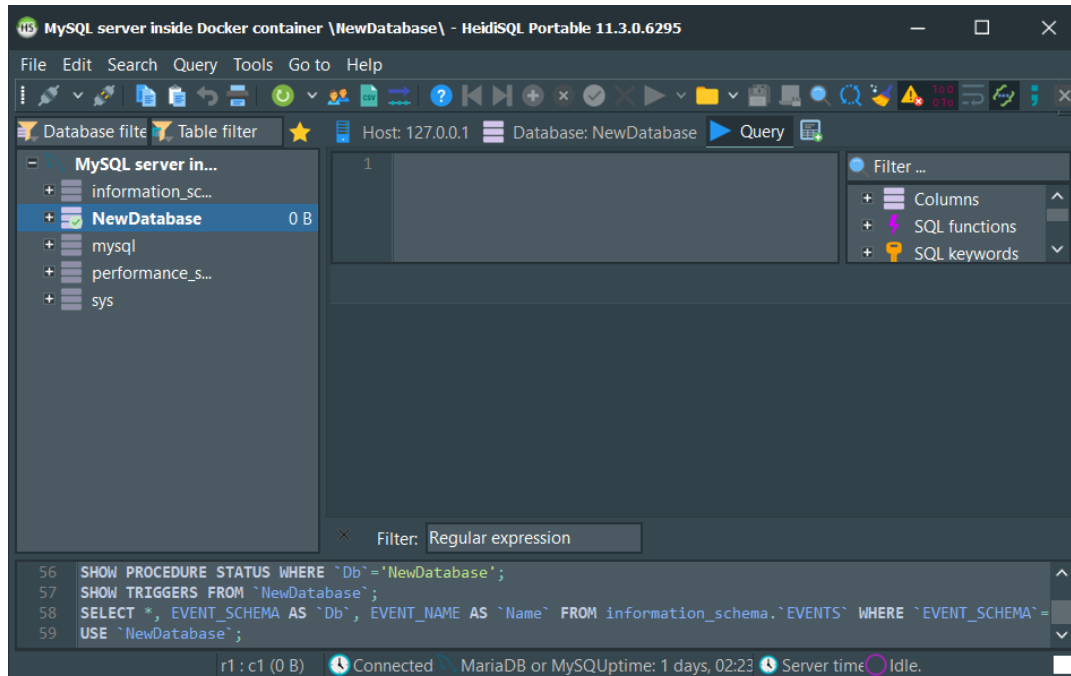
```
CREATE DATABASE IF NOT EXISTS `NewDatabase` /*!40100  
COLLATE 'utf8mb4_unicode_ci' */
```

После нажатия кнопки «OK», в левой части главного окна программы появляется новая БД NewDatabase:

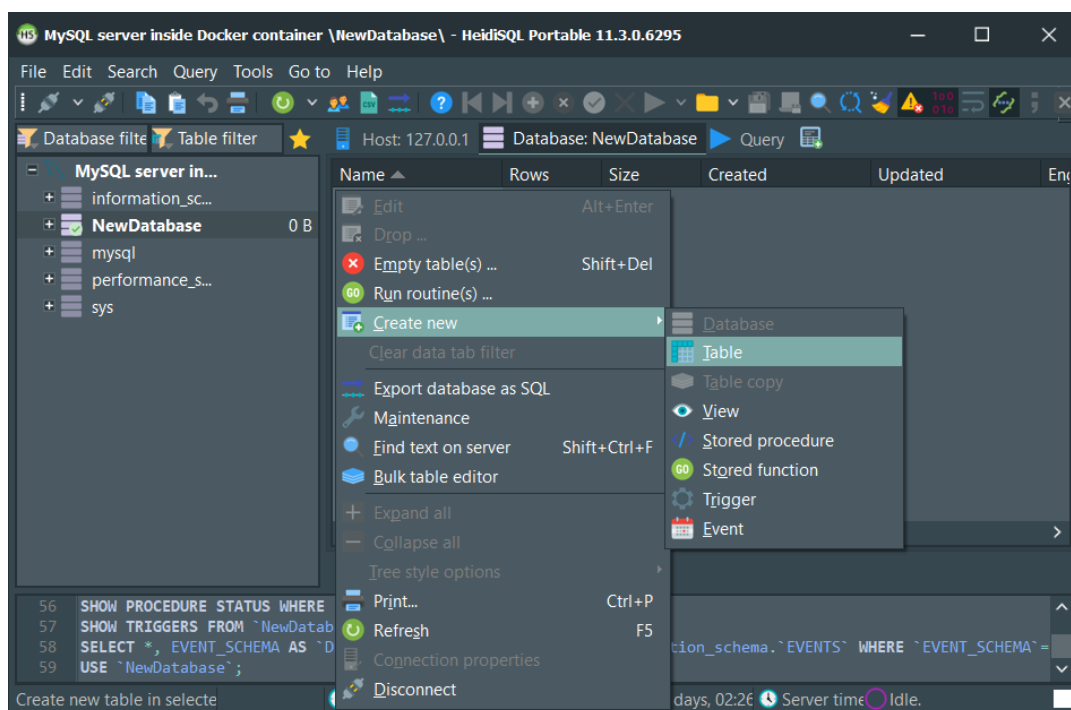


Создание новой таблицы

После нажатия ЛКМ (левой кнопки мыши) по названию БД NewDatabase, в правой части экрана откроется новая вкладка «Database: NewDatabase» («База данных: NewDatabase»):



Осуществляется переход в эту вкладку, на пустом месте в этой вкладке нажимается ПКМ (правая кнопка мыши), в появившемся контекстном меню выбирается «Create new» -> «Table» («Создать» -> «Таблица»).



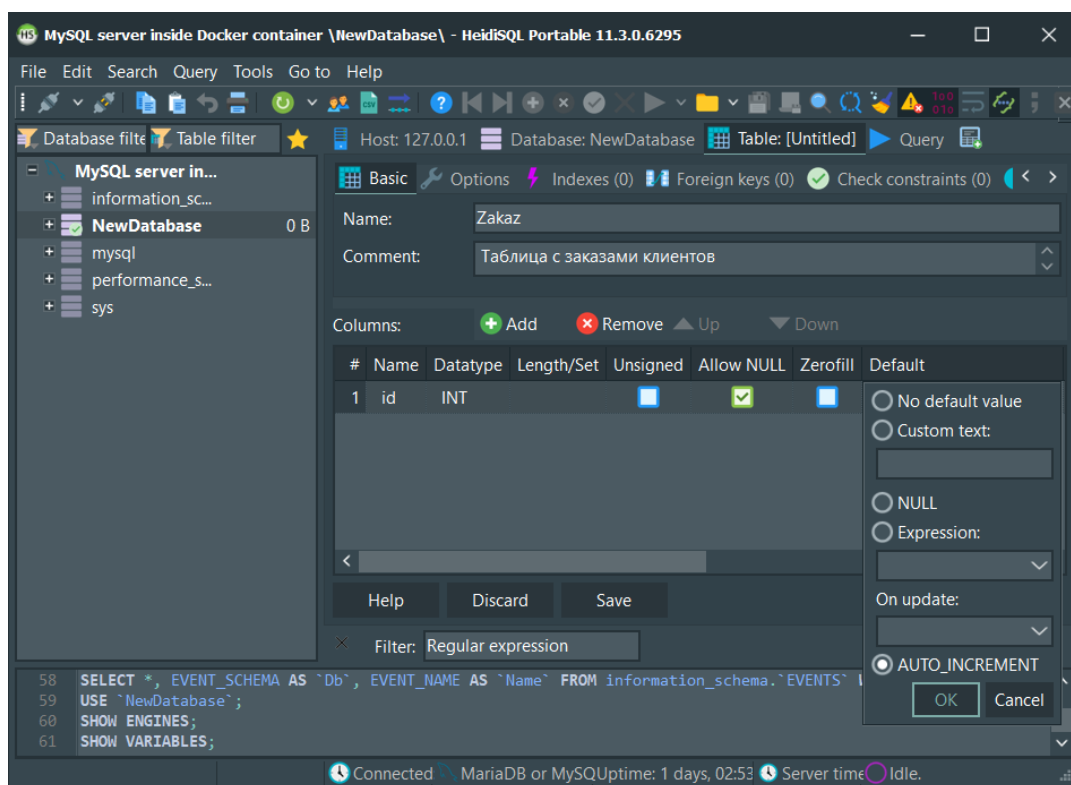
В появившейся вкладке «Table» («Таблица») заполняются следующие поля ввода:

Name (Имя): Zakaz.

Comment (Комментарий): Таблица с заказами клиентов.

Столбец — это поле, предназначенное для хранения определенной информации о каждой записи в таблице. Для того, чтобы создать автоинкрементный столбец (column) в этой таблице, нажимается кнопка «Add» («Добавить»), затем указываются следующие параметры столбца:

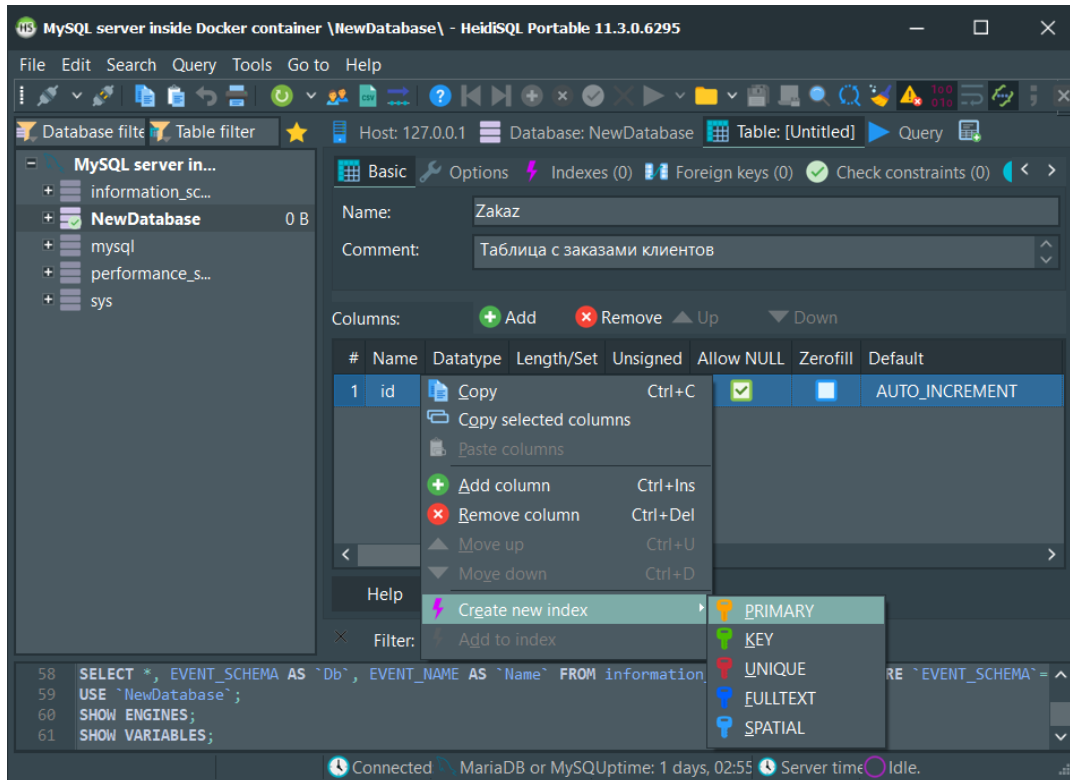
- **Name (Имя):** id.
- **Datatype (Тип данных):** INT.
- **Default (По умолчанию):** AUTO_INCREMENT.



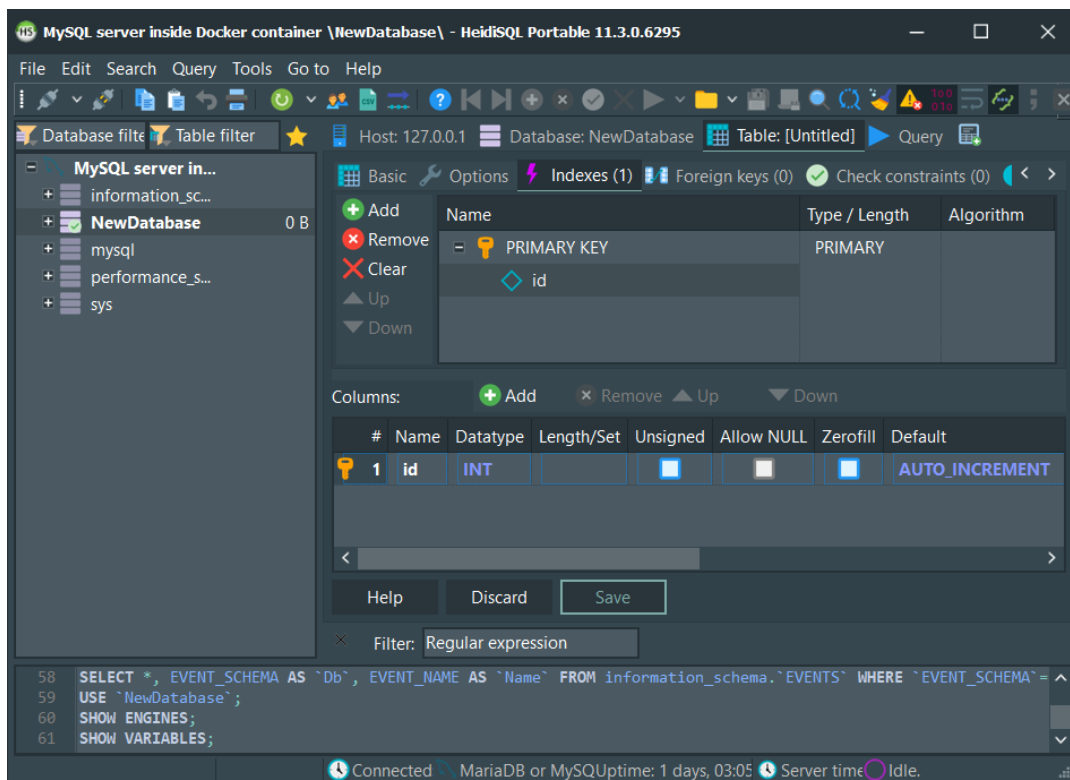
Первичный ключ (primary key) представляет собой один из примеров уникальных индексов и применяется для однозначной идентификации записей таблицы. Никакие из двух записей таблицы не могут иметь одинаковых значений первичного ключа. В реляционных базах данных практически всегда разные таблицы логически связаны друг с другом. Именно первичные ключи используются для однозначной организации такой связи.

Атрибут AUTO_INCREMENT позволяет автоматически генерировать уникальный ключ, если его тип является целочисленным. При вставке записи в базу данных значение ключа выставляется равным нулю, MySQL автоматически вычисляет максимальный номер первичного ключа, увеличивает его на единицу и присваивает это значение первичному ключу новой записи.

Для того, чтобы сделать столбец id первичным ключом, на его названии нажимается ПКМ и выбирается «Create new index» («Создать новый индекс») -> «PRIMARY»:



После успешного создания уникального индекса рядом с названием столбца отобразится иконка ключа, а во вкладке «Indexes» («Индексы») появится его название:



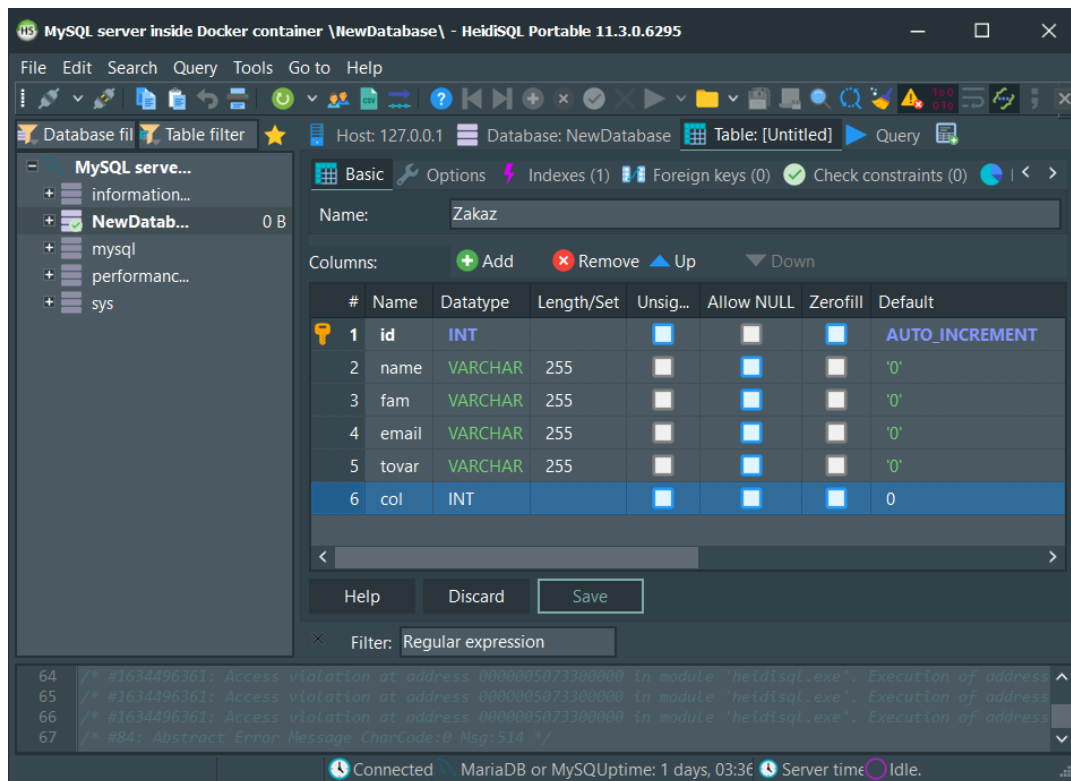
Добавляются ещё четыре столбца с типом данных **VARCHAR** (строковый):

- **name** — имя заказчика;
- **fam** — фамилия заказчика;
- **email** — e-mail заказчика;
- **товар** — название товара.

Последним добавляется столбец с типом данных **INT** (числовой):

- **col** — количество штук.

Для сохранения внесенных в таблицу изменений нажимается кнопка «Save» («Сохранить»):



В результате формируются следующие команды SQL, создающие таблицу Zakaz:

```
USE `NewDatabase`;
```

```
CREATE TABLE IF NOT EXISTS `Zakaz` (
```

```
  `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL  
  DEFAULT '0',
```

```
  `fam` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL  
  DEFAULT '0',
```

```
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL  
  DEFAULT '0',
```



```

`tovar` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL
DEFAULT '0',

`col` int(11) NOT NULL DEFAULT '0',

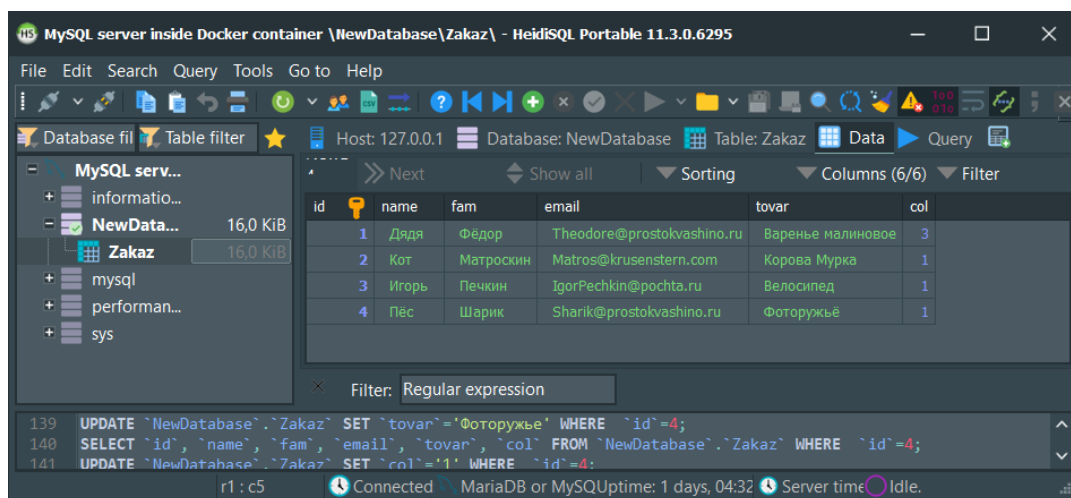
PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci COMMENT='Таблица с заказами
клиентов';

```

Добавление записей в таблицу БД

Открывается вкладка «Data» («Данные») и в верхнем меню нажимается кнопка «Insert row into table» («Вставьте строку в таблицу»), представляющая из себя белый знак «+» на фоне зелёного круга, затем в таблицу вносятся данные новой записи. Всего в таблицу Zakaz вносятся четыре записи:



В результате формируются следующие команды SQL, вставляющие четыре новые записи в таблицу Zakaz:

```

INSERT INTO `Zakaz` (`id`, `name`, `fam`, `email`, `tovar`,
`col`) VALUES

```

```

    (1, 'Дядя', 'Фёдор', 'Theodore@prostokvashino.ru',
'Варенье малиновое', 3),

```

```

    (2, 'Кот', 'Матроскин', 'Matros@krusenstern.com',
'Корова Мурка', 1),

```

```

    (3, 'Игорь', 'Печкин', 'IgorPechkin@pochta.ru',
'Велосипед', 1),

```

```

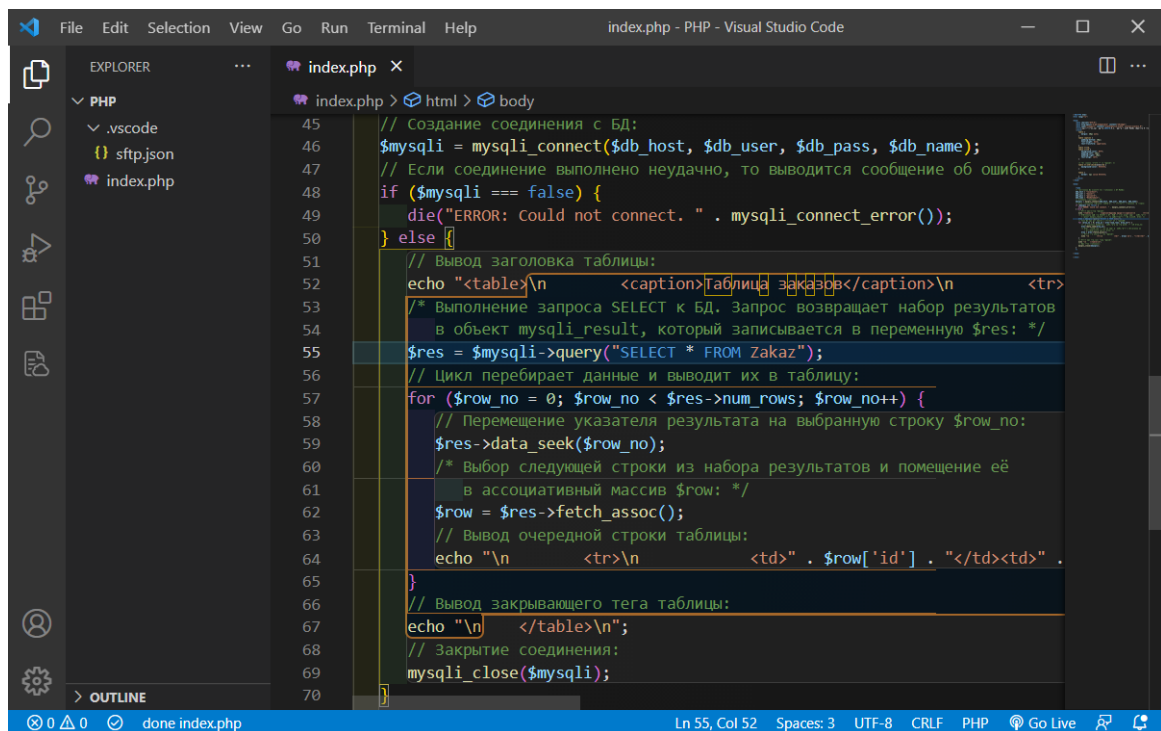
    (4, 'Пёс', 'Шарик', 'Sharik@prostokvashino.ru',
'Фоторужье', 1);

```

Создание программы на языке PHP

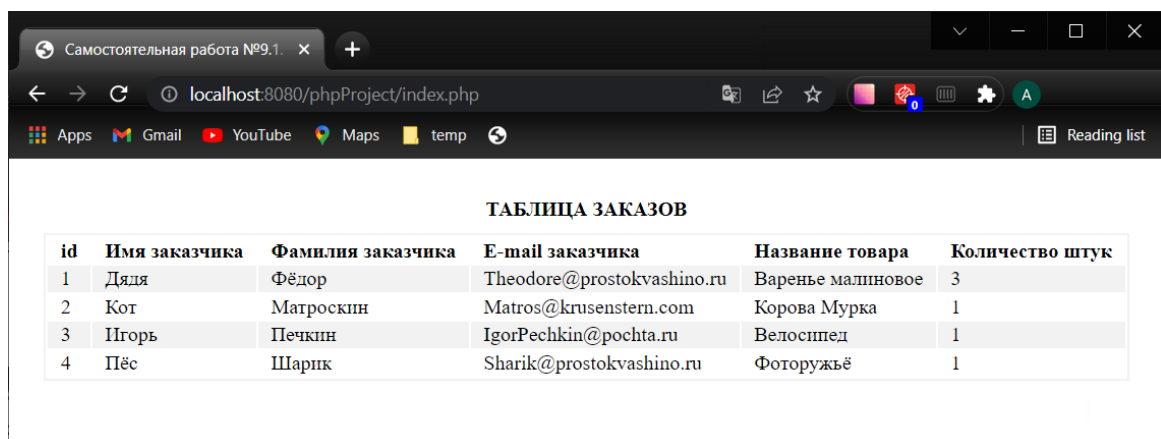
В самостоятельной работа №1 была произведена настройка расширения SFTP для редактора кода Visual Studio Code. Это расширение позволяет подключаться по SSH к Docker контейнеру внутри Ubuntu Server.

В редакторе Visual Studio Code в директории PHP создается файл index.php, в который добавляется HTML страница с программой на языке PHP. Программа соединяется с БД MySQL и выводит всё содержимое таблицы Zakaz на страницу. Файл index.php сохраняется и автоматически загружается на сервер (в контейнер):



```
45 // Создание соединения с БД:
46 $mysqli = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
47 // Если соединение выполнено неудачно, то выводится сообщение об ошибке:
48 if ($mysqli === false) {
49     die("ERROR: Could not connect. " . mysqli_connect_error());
50 } else {
51     // Вывод заголовка таблицы:
52     echo "<table>\n    <caption>Таблица заказов</caption>\n    <tr>
53     /* Выполнение запроса SELECT к БД. Запрос возвращает набор результатов
54     в объект mysqli_result, который записывается в переменную $res: */
55     $res = $mysqli->query("SELECT * FROM Zakaz");
56     // Цикл перебирает данные и выводит их в таблицу:
57     for ($row_no = 0; $row_no < $res->num_rows; $row_no++) {
58         // Перемещение указателя результата на выбранную строку $row_no:
59         $res->data_seek($row_no);
60         /* Выбор следующей строки из набора результатов и помещение её
61         в ассоциативный массив $row: */
62         $row = $res->fetch_assoc();
63         // Вывод очередной строки таблицы:
64         echo "\n        <tr>\n            <td>" . $row['id'] . "</td><td>" .
65     }
66     // Вывод закрывающего тега таблицы:
67     echo "\n    </table>\n";
68     // Закрытие соединения:
69     mysqli_close($mysqli);
70 }
```

По умолчанию, сервер Apache работает на 80-м порту, но при запуске контейнера порт 80 был проброшен на порт 8080. Страница с загруженным файлом index.php доступна из браузера по адресу <http://localhost:8080/phpProject/index.php>



id	Имя заказчика	Фамилия заказчика	E-mail заказчика	Название товара	Количество штук
1	Дядя	Фёдор	Theodore@prostokvashino.ru	Варенье малиновое	3
2	Кот	Матроскин	Matros@krusenstern.com	Корова Мурка	1
3	Игорь	Печкин	IgorPechkin@pochta.ru	Велосипед	1
4	Пёс	Шарик	Sharik@prostokvashino.ru	Фоторужьё	1

Как видно на снимке экрана, при открытии этой страницы в браузере выводится список всех записей таблицы Zakaz.