

СОДЕРЖАНИЕ

ПЕРЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	5
ВВЕДЕНИЕ	6
1 ПРОБЛЕМАТИКА РАЗРАБОТКИ МУЛЬТИСИНХРОННЫХ УСТРОЙСТВ	7
1.1 Домен синхрочастоты	7
1.2 Основные принципы	8
1.3 Решение проблемы метастабильности	9
1.4 Асинхронная очередь	10
2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ И СТРУКТУРНОЙ СХЕМ УЗЛА	12
2.1 Описание принципа работы	12
2.2 функциональная схема узла	12
3 Создание базы данных	16
3.1 Работа по методическим указаниям	16
3.2 Индивидуальное задание	19
3.2.1 Использование MySQL CLI Client	24
3.2.2 Использование MySQL Workbench	26
4 Создание веб-клиента	28
4.1 Общие требования к приложению	28
4.1.1 Требования к функционалу	28
4.1.2 Требования к интерфейсу	28
4.2 Реализация приложения	29
4.2.1 Используемые технологии	29
4.2.2 Возможные сценария использования веб-приложения	30
4.2.3 Демонстрация работы веб-приложения	30
4.2.3.1 Авторизация	30
4.2.3.2 Отображение записей таблиц	31
4.2.3.3 Добавление записей в в таблицы	31
4.2.3.4 Выполнение произвольных запросов	32
ЗАКЛЮЧЕНИЕ	35
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	36

ПЕРЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

CDC — Clock domain crossing. Пересечение доменов синхрочастоты

FIFO —

FPGA —

MSB — Most significant bit. Старший бит

ВВЕДЕНИЕ

Целью работы является построение цифрового устройства ресинхронизации данных. Данное устройство представляет собой FIFO с восьмиразрядным входом и тридцатидвух разрядным выходом. Данные на выход подаются по мере образования во входном буфере не менее 4 8-разрядных слов. Тактовые сигналы входного и выходного каналов независимы, однако тактовая частота для выхода FIFO составляет как минимум четверть от тактовой частоты входного интерфейса FIFO.

1 ПРОБЛЕМАТИКА РАЗРАБОТКИ МУЛЬТИСИНХРОННЫХ УСТРОЙСТВ

В данном разделе будут рассмотрены проблемы и особенности построение устройств

В этом разделе будут описаны проблемы, возникающие в процессе разработки мультисинхронного проекта, то есть устройства, в котором имеют место пересечения клаковых доменов или доменов синхрочастоты (CDC).

1.1 Домен синхрочастоты

Домен синхрочастоты представляет собой ту часть проекта, которая тактируется одной или несколькими синхрочастотами, причем все эти синхрочастоты должны иметь постоянные сдвиги фазы. Если в какой-либо части проекта имеется синхрочастота или инвертированная синхрочастота, или синхрочастота, полученная из исходной путем деления на 2, то такая часть проекта считается блоковым доменом с одной синхрочастотой. Если же домены имеют синхрочастоты с переменной фазой и соотношениями времени, то такие домены считают доменами с различными синхрочастотами.

На Рисунке 1.1 показано, что проект имеет единственный домен синхрочастоты, потому что синхрочастота `divClk` — есть деленная на два частота генератора синхронизации `Clk`.

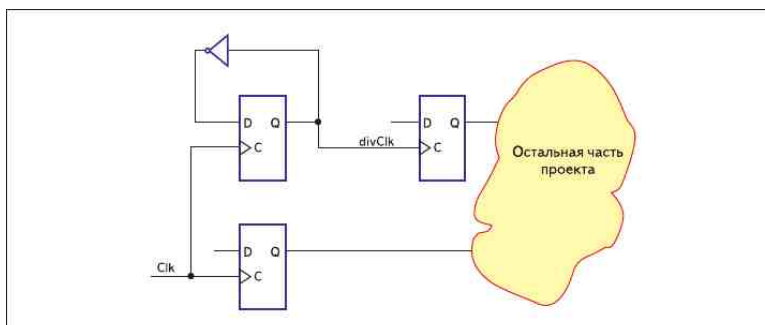


Рисунок 1.1 – Схема с одним доменом синхрочастоты

На Рисунке 1.2 показано несколько синхрочастот от различных источников. Ту часть проекта, которая управляется этими синхрочастотами, называют доменами синхрочастоты, и сигналы, осуществляющие передачу

импульсов между этими асинхронными доменами синхрочастоты, называют путями пересечения домена синхрочастоты. Сигнал DA считают асинхронным сигналом в домене синхрочастоты, так как между генератором синхронизации A (clkA) и генератором синхронизации B (ClkB) не существуют постоянные соотношения фазы и времени.

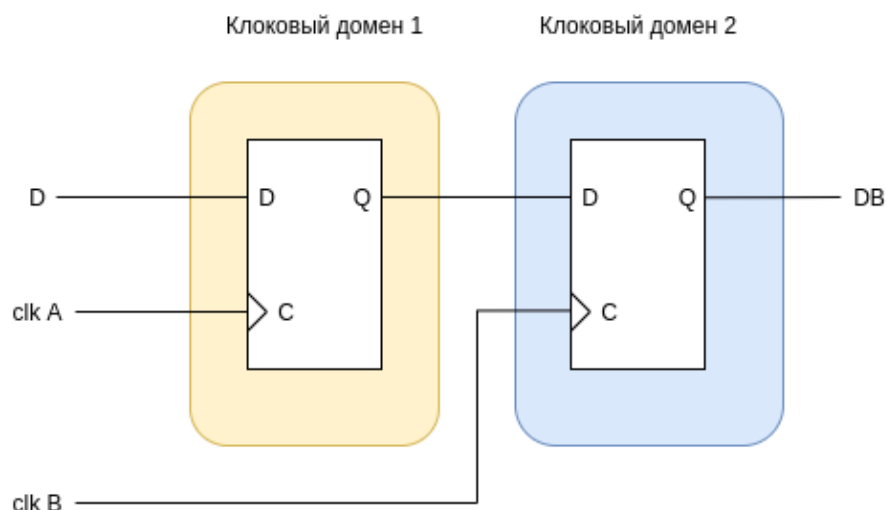


Рисунок 1.2 — Путь домена синхрочастоты

1.2 Основные принципы

При разработке ультисинхрочастотных проектов следует уделять особое внимание стабильности сигнала. Когда сигнал пересекает домен синхрочастоты, то он появляется в новом домене синхрочастоты как асинхронный сигнал и должен быть засинхронизирован.

Синхронизация предотвращает в новом домене синхрочастоты метастабильное состояние первого запоминающего элемента схемы (триггера), и это позволяет в новом домене работать уже со стабильным сигналом.

Метастабильность — это неспособность триггера достигнуть известного состояния в определенный момент времени. Когда триггер входит в метастабильное состояние, то невозможно предсказать ни уровень выходного напряжения элемента, ни период времени, за который этот выход перейдет к правильному уровню напряжения.

В течение этого переходного времени выход триггера будет находиться на некотором промежуточном уровне напряжения или колебаться и может

передать этот недопустимый уровень сигнала со своего выхода к другим триггерам схемы.

1.3 Решение проблемы метастабильности

С целью решения проблемы метастабильности применяются каскады стабилизирующих триггеров, включенных последовательно. Рассмотрим это решение.

Простейший синхронизатор представляет собой два триггера, включенных последовательно без какой-либо комбинационной схемы между ними. Такая схема проекта гарантирует, что первый триггер выходит из своего метастабильного состояния, и его выход переходит в устойчивое состояние перед тем, как второй триггер сохраняет его. Необходимо также разместить эти триггеры, насколько возможно, ближе друг к другу для того, чтобы гарантировать наименьшую расфазировку тактовых сигналов между ними.

Другой тип ячейки синхронизатора представляет собой два близко расположенных триггера без какой-либо комбинационной логики между ними. Для того чтобы синхронизация работала должным образом, сигнал, пересекающий домен синхрочастоты, должен проследовать от триггера в домене синхрочастоты источника сигнала к первому триггеру синхронизатора, не пройдя через комбинационную логику.

Синхронизацию в домене называют привязкой входного сигнала к тактовой частоте домена, т. е. все изменения этого сигнала в домене будут происходить по фронту или срезу тактового сигнала домена, а не родительского тактового сигнала. На Рисунке 1.3) приведена схема традиционного синхронизатора из двух триггеров (сдвиговый регистр) для одиночного сигнала. Схема синхронизации приведена на Рисунке 1.4.

Данный способ борьбы с метастабильностью будет в дальнейшем использован при разработке узла ресинхронизации данных.

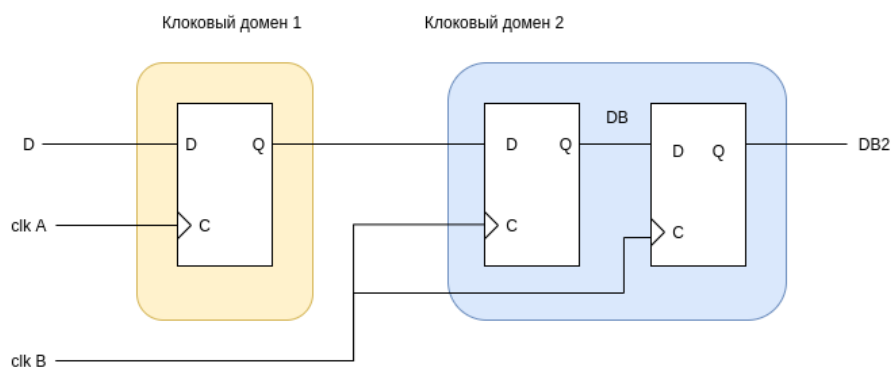


Рисунок 1.3 — Триггеры синхронизации

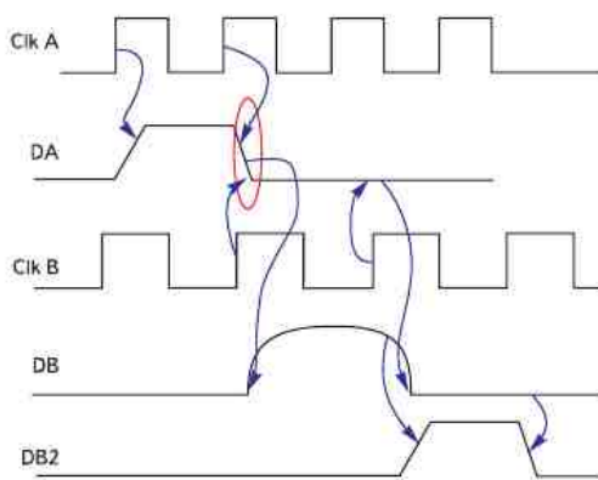


Рисунок 1.4 — Схема синхронизации сигнала

1.4 Асинхронная очередь

Во многих проектах необходимо передавать из одного домена в другой не только несколько одиночных сигналов, но и сигналы типа шин: шины данных, адреса и шины управления.

В этих задачах активно используются буферы типа FIFO и специальные протоколы для процедуры установления связи. Особенно полезной оказывается асинхронная очередь, позволяющая производить запись и чтение по синхросигналам двух клоковых доменов. Данная схема позволяет быстро и удобно передавать данные в реальном времени между двумя разными областями синхронизации.

На Рисунке 1.5 приведена схема использования двухпортовой оперативной памяти, позволяющей осуществлять передачу данных между частями устройства, использующими синхросигналы разной частоты.

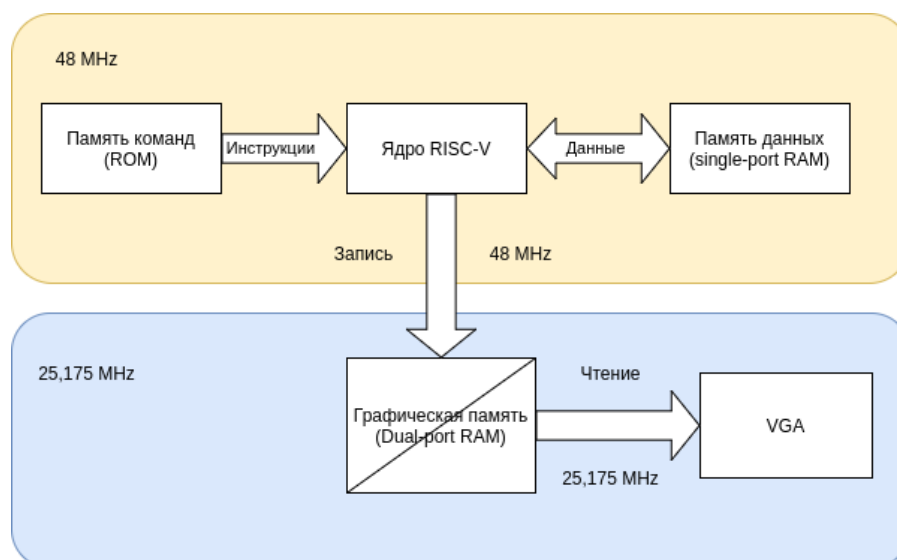


Рисунок 1.5 — Двухпортовая оперативная память

Двухпортовая память с независимыми тактовыми сигналами — аппаратный блок, используемый в современных FPGA и в технологических библиотеках для заказных микросхем. Чтение и запись производится совершенно независимо через два отдельных порта.

Единственным ограничением является одновременное обращение на запись и чтение по одному и тому же адресу памяти — оно может привести к неопределенному результату. На основе такого блока памяти зачастую создается модуль FIFO, который позволяет с одной стороны записывать данные из одного тактового домена, а с другой — забирать в другой тактовый домен. Заодно логика FIFO следит за тем, чтобы не происходило обращения к одной и той же ячейке памяти.

Данный узел работает по следующему принципу.

1. Источник данных загружает данные в асинхронную очередь. Тактирование производится по тактовому сигналу источника данных.
2. Потребитель данных считывает данные из асинхронной очереди в исходном порядке. Тактирование производится по тактовому сигналу потребителя данных.

2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ И СТРУКТУРНОЙ СХЕМ УЗЛА

К разработке предлагается устройство ресинхронизации данных. Определим структурные элементы устройства, принцип и порядок его работы.

2.1 Описание принципа работы

Взаимодействие с устройством производится посредством передачи восьмибитного слова на вход *wdata*. Аппаратный блок осуществляет запись данных во внутреннюю память по сигналам тактового генератора домена источника данных *wclk*, основываясь на состоянии входных сигналов *wrst*, *wen* и служебных сигналов *wr_full*, *wptr*, *rptr*.

Устройство может одновременно хранить шестнадцать восьмибитных слов, используя для их адресации пятиразрядный адрес, в котором старший бит призван реализовать возможность контроля заполнения и опустошения очереди.

Чтение из устройства ресинхронизации производится посредством считывания тридцатидвухразрядного слова, генерируемого на выходе *rdata*. Аппаратный блок осуществляет чтение данных из внутренней памяти по сигналам тактового генератора домена получателя данных, основываясь на состоянии входных сигналов *rrst*, *ren*, *r_empty*, *wptr*, *rptr*, *rdredy*.

Тактовые сигналы входного и выходного каналов независимы, однако тактовая частота для выхода очереди составляет как минимум $\frac{1}{4}$ от тактовой частоты входного интерфейса FIFO.

Устройство

2.2 функциональная схема узла

Предлагается для построения устройства ресинхронизации данных, работающего в режиме очереди с независимыми тактовыми сигналами, использовать данную функциональную схему (см. Рисунок 2.1).

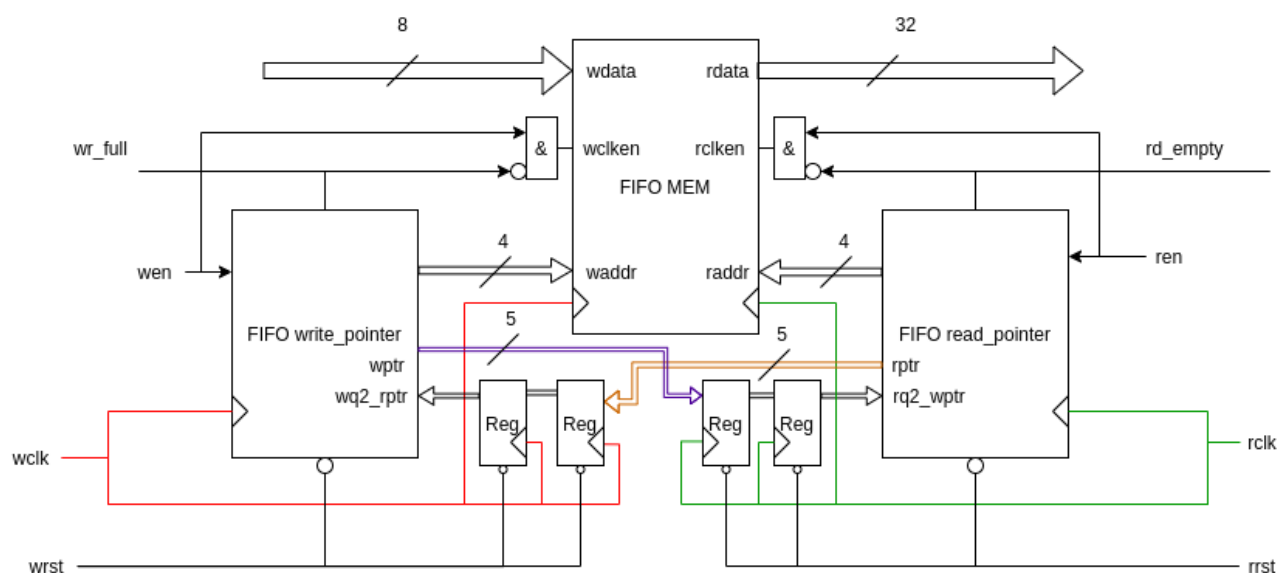


Рисунок 2.1 — Функциональная схема устройства

Приведем описание входов, выходов и служебных сигналов устройства.

Таблица 2.1

Название	Назначение
asize	размер адреса данных с учетом адресации MSB
dsize	размер данных в битах
mem	ячейки памяти очереди
rbin	текущий адрес чтения в бинарном коде
rbinnext	адрес чтения в бинарном коде в следующем такте
rclk	тактирование на чтение данных
rdata	данные на чтение
rden	разрешение на чтение
rdredy	готовность выдать данные
rempty	сигнал пустоты очереди (прочтено все, что записано)
rempty_next	пустота очереди на следующем такте
rgray	текущий адрес чтения в коде Грея
rgraynext	адрес чтения в коде Грея в следующем такте
rq1_wgray	адрес чтения в первом синхро-триггере
rq2_wgray	адрес чтения во втором синхро-триггере
rrstn	сброс указателя чтения
wbin	текущий адрес записи в бинарном коде
wbinnext	адрес записи в бинарном коде в следующем такте
wclk	Тактирование на запись данных
wdata	данные на запись
wfull	сигнал заполнения очереди
wfull_next	заполненность очереди на следующем такте
wgray	текущий адрес записи в коде Грея
wgraynext	адрес записи в коде Грея в следующем такте
wq1_rgray	адрес записи в первом синхро-триггере
wq2_rgray	адрес записи во втором синхро-триггере
wren	разрешение на запись
wrstn	сброс указателя записи

3 Создание базы данных

3.1 Работа по методическим указаниям

Создадим базу данных forum, которая хранит в себе сведения о пользователях форумах и размещенных ими темах.

Помимо суперпользователя root, был создан пользователь denilai, под которым производятся все манипуляции с данными.

Создадим базу данных forum с помощью команды CREATE DATABASE forum;

Создадим таблицу users (см. Рисунок 3.1):

```
mysql> create table users (  
-> id_user int (10) AUTO_INCREMENT,  
-> name varchar (20) NOT NULL,  
-> email varchar (50) NOT NULL,  
-> password varchar (15) NOT NULL,  
-> PRIMARY KEY (id_user));  
Query OK, 0 rows affected (0.16 sec)
```

Рисунок 3.1 — Создание базы данных users

Создадим таблицу topics (см. Рисунок 3.2):

```
mysql> create table topics (  
-> id_topic int (10) AUTO_INCREMENT,  
-> topic_name varchar(100) NOT NULL,  
-> id_author int(10) NOT NULL,  
-> PRIMARY KEY (id_topic),  
-> FOREIGN KEY (id_author) REFERENCES users (id_user));  
Query OK, 0 rows affected (0.13 sec)
```

Рисунок 3.2 — Создание базы данных topics

Создадим таблицу posts (см. Рисунок 3.3):

После создания таблиц, заполним их данными о пользователях форума, о темах и размещенных публикациях. Выполним операцию выборки данных без условия, чтобы увидеть все записи, занесенные в таблицы с помощью команды SELECT * FROM <table-name> (см. Рисунок 3.4).

Выполним запрос SELECT message, topic_name FROM posts p JOIN topics t ON t.id_author = p.id_author; для объединения данных из

```
mysql> create table posts (
  -> id_post int (10) AUTO_INCREMENT,
  -> message text NOT NULL,
  -> id_author int (10) NOT NULL,
  -> id_topic int (10) NOT NULL,
  -> primary key (id_post),
  -> foreign key (id_author) references users (id_user),
  -> foreign key (id_topic) references topics (id_topic));
Query OK, 0 rows affected (0.17 sec)
```

Рисунок 3.3 — Создание базы данных posts

```
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19673
Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

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> use forum;
Database changed
mysql> select * from users;
+----+-----+-----+-----+-----+-----+-----+
| id_user | name  | email          | password | kol | reputation | role      |
+----+-----+-----+-----+-----+-----+-----+
| 1      | sergey | sergey@mail.ru | 111      | 50  | Profi      | moderator |
| 2      | valera | valera@mail.ru | 2222     | 30  | changed    | NULL      |
| 3      | katy   | katy@gmail.ru  | 3333     | 45  | Profi      | NULL      |
| 4      | sveta  | sveta@rambler.ru | 4444     | 20  | NULL       | NULL      |
| 5      | oleg   | oleg@yandex.ru | 5555     | 2   | NULL       | NULL      |
+----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from topics;
+----+-----+-----+
| id_topic | topic_name | id_author |
+----+-----+-----+
| 1        | about fishing | 1        |
| 2        | cycles      | 2        |
| 3        | nightclubs  | 1        |
| 4        | about fishing | 4        |
+----+-----+-----+
4 rows in set (0.01 sec)

mysql> select * from posts;
+----+-----+-----+-----+
| id_post | message      | id_author | id_topic |
+----+-----+-----+-----+
| 1       | hello there  | 1         | 1        |
| 3       | have a nice day | 3         | 1        |
+----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Рисунок 3.4 — Операция выборки из всех таблиц

таблиц topics и posts по ключу id_author и получения полной информации о сообщении и названию темы, в которой оно было размещено (см. Рисунок 3.5).

Выполним запрос выборки данных, явно указав поля отношения. Для этого перечислим имена полей через запятую после зарезервированного слова SELECT (см. Рисунок 3.6).

Выполним более сложные запросы выборки, отсортировав записи в таблице topics по убыванию значения поля topic_name и id_author, а также опишем условие сравнения значения поля id_author в секции WHERE (см. Рисунок 3.7).

```
mysql> select message, topic_name from posts p join topics t on t.id_author=p.id_author
-> ;
+-----+-----+
| message | topic_name |
+-----+-----+
| hello there | about fishing |
| hello there | nightclubs |
+-----+-----+
2 rows in set (0.00 sec)

mysql> _
```

Рисунок 3.5 — Запрос объединения

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select id_user from users;
+-----+
| id_user |
+-----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+-----+
5 rows in set (0.01 sec)

mysql> select name, email from users;
+-----+-----+
| name | email |
+-----+-----+
| sergey | sergey@mail.ru |
| valera | valera@mail.ru |
| katy | katy@gmail.ru |
| sveta | sveta@rambler.ru |
| oleg | oleg@yandex.ru |
+-----+-----+
5 rows in set (0.00 sec)
```

Рисунок 3.6 — Операция выборки с указанием полей

Выполним операции по модификации таблицы — добавим в таблицу users поле country типа varchar (20) со значением по умолчанию "Russia" а также добавим в эту же таблицу поле age int(10) со значением по умолчанию 19.

Выведем все записи из таблицы users, обнаружим, что столбцы были вставлены успешно (см. Рисунок 3.8).

В ходе данной практической работы были рассмотрены операторы DDL и DML диалекта MySQL. С помощью данных операторов была создана база учебная база данных forum, содержащая о пользователях форумах и размещенных ими темах.

```

mysql>
mysql>
mysql>
mysql>
mysql> select * from topics order by topic_name desc;
+-----+-----+-----+
| id_topic | topic_name | id_author |
+-----+-----+-----+
| 3 | nightclubs | 1 |
| 2 | cycles | 2 |
| 1 | about fishing | 1 |
| 4 | about fishing | 4 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from topics order by topic_name desc, id_author desc;
+-----+-----+-----+
| id_topic | topic_name | id_author |
+-----+-----+-----+
| 3 | nightclubs | 1 |
| 2 | cycles | 2 |
| 4 | about fishing | 4 |
| 1 | about fishing | 1 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from topics where topic_name='cycles';
+-----+-----+-----+
| id_topic | topic_name | id_author |
+-----+-----+-----+
| 2 | cycles | 2 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from topics where id_author between 1 and 3;
+-----+-----+-----+
| id_topic | topic_name | id_author |
+-----+-----+-----+
| 1 | about fishing | 1 |
| 3 | nightclubs | 1 |
| 2 | cycles | 2 |
+-----+-----+-----+
3 rows in set (0.00 sec)

```

Рисунок 3.7 — Сложные запросы выборки с сортировкой и условием

```

mysql> alter table users add column country varchar(20) default 'Russia';
Query OK, 0 rows affected (0.42 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from users;
+-----+-----+-----+-----+-----+-----+-----+
| id_user | name | email | password | kol | reputation | role | country |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | sergey | sergey@mail.ru | 111 | 50 | Profi | moderator | Russia |
| 2 | valera | valera@mail.ru | 2222 | 30 | changed | NULL | Russia |
| 3 | katy | katy@mail.ru | 3333 | 45 | Profi | NULL | Russia |
| 4 | sveta | sveta@rambler.ru | 4444 | 20 | NULL | NULL | Russia |
| 5 | oleg | oleg@yandex.ru | 5555 | 2 | NULL | NULL | Russia |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> alter table users add column age int(10) after name default 19;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'default 19' at line 1
mysql> alter table users add column age int(10) default 19 after name;
Query OK, 0 rows affected (0.42 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_user | name | age | email | password | kol | reputation | role | country |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | sergey | 19 | sergey@mail.ru | 111 | 50 | Profi | moderator | Russia |
| 2 | valera | 19 | valera@mail.ru | 2222 | 30 | changed | NULL | Russia |
| 3 | katy | 19 | katy@mail.ru | 3333 | 45 | Profi | NULL | Russia |
| 4 | sveta | 19 | sveta@rambler.ru | 4444 | 20 | NULL | NULL | Russia |
| 5 | oleg | 19 | oleg@yandex.ru | 5555 | 2 | NULL | NULL | Russia |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Рисунок 3.8 — Добавление полей в таблицу users

3.2 Индивидуальное задание

Продолжим работу над созданием модели данных велосипедного предприятия. Создадим базу данных cycle, сущностями которой будут таблицы, описания которых были проработаны в прошлых практических работах.

Описание проектируемых отношений базы данных cycle приведены в таблицах 3.1–3.15.

Таблица 3.1 — Описание таблицы Bar

Имя	Тип
Width	INTEGER NOT NULL
Diameter	INTEGER NOT NULL
BarID	INTEGER NOT NULL AUTO_INCREMENT
Color	VARCHAR(20) NULL DEFAULT 'black'

Таблица 3.2 — Описание таблицы Brake

Имя	Тип
BrakeID	INT NOT NULL AUTO_INCREMENT
ComponentID	INTEGER NULL

Таблица 3.3 — Описание таблицы Component

Имя	Тип
ComponentID	INTEGER NOT NULL AUTO_INCREMENT
Brand	VARCHAR(20) NOT NULL
ManufacturerCountry	VARCHAR(20) NOT NULL
PriceCent	INTEGER NOT NULL
Model	VARCHAR(20) NOT NULL
ComponentTypeID	INTEGER NOT NULL
WeightGramm	INTEGER NOT NULL

Таблица 3.4 — Описание таблицы ComponentType

Имя	Тип
ComponentTypeID	INTEGER NOT NULL AUTO_INCREMENT
Name	VARCHAR(20)

Таблица 3.5 — Описание таблицы CycleType

Имя	Тип
CycleTypeID	INTEGER NOT NULL AUTO_INCREMENT
Name	VARCHAR(20) NOT NULL

Таблица 3.6 — Описание таблицы FctCycleBuild

Имя	Тип
SetupID	INTEGER NOT NULL
Datetime	DATE NOT NULL
Stage	INTEGER NOT NULL
Workshop	varchar(20) NOT NULL DEFAULT 'main'
Master	VARCHAR(20) NOT NULL
Status	VARCHAR(20) NOT NULL
Comment	VARCHAR(20) NOT NULL

Таблица 3.7 — Описание таблицы Fork

Имя	Тип
ForkID	INTEGER NOT NULL AUTO_INCREMENT
Color	VARCHAR(20) NULL DEFAULT 'black'
ComponentID	INTEGER NULL

Таблица 3.8 — Описание таблицы Frame

Имя	Тип
FrameID	INTEGER NOT NULL AUTO_INCREMENT
Color	VARCHAR(20) NULL DEFAULT 'black'

Таблица 3.9 — Описание таблицы FrameInfo

Имя	Тип
FrameID	INTEGER NOT NULL
Size	INTEGER NOT NULL
Stack	INTEGER NOT NULL
Reach	INTEGER NOT NULL

Таблица 3.10 — Описание таблицы Frameset

Имя	Тип
FrameSizeID	INTEGER NOT NULL AUTO_INCREMENT
ForkID	INTEGER NOT NULL
FramesetID	INTEGER NOT NULL
ComponentID	INTEGER NOT NULL

Таблица 3.11 — Описание таблицы FrameSize

Имя	Тип
FrameID	INTEGER NOT NULL
Size	INTEGER NOT NULL
FrameSizeID	INTEGER NOT NULL AUTO_INCREMENT
ComponentID	INTEGER NOT NULL

Таблица 3.12 — Описание таблицы Groupset

Имя	Тип
Ratio	INTEGER NOT NULL
BrakeID	CHAR(18) NOT NULL
GroupsetID	INTEGER NOT NULL AUTO_INCREMENT
Color	VARCHAR(20) NOT NULL
ComponentID	INTEGER NOT NULL

Таблица 3.13 — Описание таблицы Setup

Имя	Тип
GroupsetID	INTEGER NOT NULL
WheelsetID	INTEGER NOT NULL
FramesetID	INTEGER NOT NULL
BarID	INTEGER NOT NULL
CycleTypeID	INTEGER NOT NULL
SetupID	INTEGER NOT NULL AUTO_INCREMENT

Таблица 3.14 — Описание таблицы Wheelset

Имя	Тип
DiametrInch	INTEGER NOT NULL
Width	INTEGER NOT NULL
SpokesCount	INTEGER NULL
WheelsetID	INTEGER NOT NULL AUTO_INCREMENT
Color	VARCHAR(20) NOT NULL
ComponentID	INTEGER NOT NULL

Таблица 3.15 — Описание таблицы Log

Имя	Тип
ID	INTEGER NOT NULL AUTO_INCREMENT
msg	VARCHAR100 NOT NULL
row_id	INTEGER NOT NULL

3.2.1 Использование MySQL CLI Client

Создадим данные таблицы с помощью MySQL CLI Client — клиента, предоставляющего доступ к СУБД через интерфейс командой строки, используя ключевое слово `CREATE TABLE`. После создания таблиц выполним команду `SHOW TABLES`, выбрав базу данных `cycle` (см. Рисунок 3.9).

```
mysql>
mysql> use cycle;
Database changed
mysql> show tables;
+-----+
| Tables_in_cycle |
+-----+
| bar              |
| brake            |
| component         |
| componenttype     |
| cycletype         |
| fctcyclebuild     |
| fork             |
| frame            |
| frameset          |
| framesize         |
| groupset          |
| log               |
| setup            |
| test             |
| wheelset         |
+-----+
15 rows in set (0.02 sec)
```

Рисунок 3.9 — таблицы базы данных `cycle`

Выведем записи из таблицы `Component` (см. Рисунок 3.10).

Выполним более сложные запросы выборки — объединим таблицы с помощью оператора `INNER JOIN` `Component` и `fork` по полю `component_id`, а также воспользуемся функцией `ROW_NUMBER()` в сочетании с оконной функцией `OVER()`, пронумеровав компоненты из таблицы `Fork` одного цвета по возрастанию значения поля `color` (см. Рисунок 3.11).

Создадим триггеры `delete_component` и `add_component` на добавление и удаление записи в таблице `Component`. При срабатывании данного триггера, будет добавляться запись в таблицу `log`, информирующая о совершении манипуляций с данными (см. Листинг 3.1). Приведем объявление данного триггера на диалекте MySQL:

Листинг 3.1 — Триггер

```
mysql> select * from component;
```

ComponentID	Brand	ManufacturerCountry	PriceCent	ComponentTypeID	WeightGramm	Model
47	3T	Italy	9000	2	210	Superleggera
48	Deda	Italy	7000	2	200	Elementi
49	Ritchey	England	25100	2	140	Superlogic
50	FSA	USA	3000	2	220	Omega Compact
51	Fox	USA	40000	6	1740	Suspension 40
52	Fox	USA	34000	6	2040	Suspension 70
53	Columbus	Italia	22000	6	340	Minimal Road Forks
54	Dedacciai	Italia	31400	6	300	RS Carbon Road Fork
55	Tifosi	Italia	26000	6	350	SRS Road Disc Brake Forks
56	Cinelli	Italia	80000	4	1780	Zydeco
57	Ragley	England	65000	4	2000	Trig Gravel
58	Cinelli	Italy	17000	4	1900	King Zydeco Gravel
59	Campagnolo	Italy	180000	8	600	Record 2x12 Speed Road Groupset
60	SRAM	USA	95000	8	500	Red eTap AXS 1x12 Speed Road Groupset
61	Campagnolo	Italy	60000	8	570	Centaur 11 Speed Rim Brake Road Groupset
62	Shimano	Japan	4500	7	200	Ultegra 6800 Brake Caliper
63	Shimano	Japan	4600	7	200	105 R7000 Brake Caliper
64	Easton	England	90000	5	1700	EC90 SL Disc Road Tubular Wheelset
65	Hope	China	15000	5	2000	Pro 4 Boost MTB Rear Hub
66	Novatec	China	9000	5	2000	Model 32
67	Mavic	Italy	15000	5	1400	Ellipse
70	Novatec	China	5000	5	300	One
71	Novatec	China	5000	6	300	One
82	3T	Italy	90000	2	2220	Twins One
83	3T	Italy	90000	2	2220	Twins One
84	3T	Italy	90000	2	2220	Twins One
85	3T	Italy	90000	2	2220	Twins One
86	3T	Italy	90000	2	2220	Twins One
87	3T	Italy	90000	2	2220	Twins One
88	3T	Italy	90000	2	2220	Twins One
89	3T	Italy	90000	2	2220	Twins One
90	3T	Italy	90000	2	2220	Twins One
91	3T	Italy	90000	2	2220	Twins One
92	3T	Italy	90000	2	2220	Twins One

Рисунок 3.10 — Вывод записей из таблицы Components

```
mysql>
mysql> select c.brand, c.model, c.pricecent from component c join fork f on c.componentid=f.componentid where color='black';
```

brand	model	pricecent
Fox	Suspension 40	40000
Fox	Suspension 70	34000
Columbus	Minimal Road Forks	22000

3 rows in set (0.00 sec)

```
mysql> select *, row_number() over(partition by color order by forkid) from fork order by color;
```

ForkID	Color	ComponentID	row_number() over(partition by color order by forkid)
1	black	51	1
2	black	52	2
3	black	53	3
9	Green	51	1
4	Red matte	54	1
8	Steel	55	1
5	Yellow gloss	55	1

7 rows in set (0.00 sec)

Рисунок 3.11 — Сложные запросы выборки. База данных cycle

```

1 DELIMITER $$
2 drop trigger delete_component; $$
3
4 create trigger 'delete_component' after delete on component
5 for each row begin
6 insert into log (msg, row_id) values (concat('delete component
      ',old.Brand,', ', old.Model), old.ComponentID);
7 end; $$
8
9 create trigger 'add_component' after insert on component
10 for each row begin
11 insert into log (msg, row_id) values (concat('insert component
      ',new.Brand,', ', new.Model), new.ComponentID);

```

```
12 end; $$
13 DELIMITER $$
```

3.2.2 Использование MySQL Workbench

Выполним операции по изменению и просмотру данных, занесенных в базу данных `cycle`, с помощью инструмента для визуального проектирования баз данных MySQL Workbench.

Для этого подключимся к локально развернутому на машине MySQL Server, указав порт, имя пользователя и пароль. После этого мы получим доступ к пользовательскому интерфейсу программы, представляющему собой две области — область выполнения запросов и область отображения результатов (см. Рисунок 3.12).

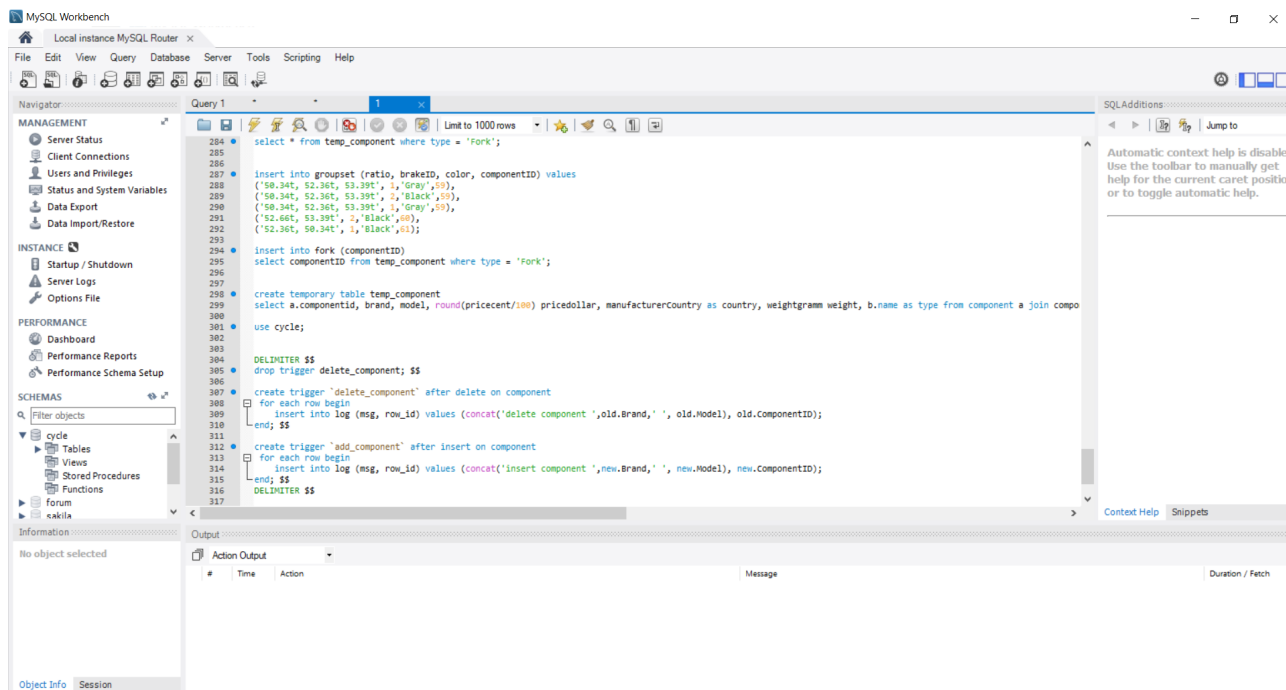


Рисунок 3.12 — Интерфейс MySQL Workbench

Создадим временную таблицу, в котором объединим сведения о компонентах (см. Рисунок 3.13).

Выполним еще один запрос — ограничим вывод только компонентами типа «Fork» (см. Рисунок 3.14).

Данный визуальный инструмент позволяет удобно организовать работу с базой данных, сохранять SQL-скрипты, параметры подключения и настройки.

Query 1

```

294 ('52.36t, 50.34t', 1, 'Black', 61);
295
296 insert into fork (componentID)
297 select componentID from temp_component where type = 'Fork';
298
299
300 create temporary table temp_component
301 select a.componentid, brand, model, round(pricecent/100) pricedollar, manufacturerCountry as country, weightgramm weight, b.name as type
302 |
303 select * from temp_component;
304
305 use cycle;
306

```

Result Grid

componentid	brand	model	pricedollar	country	weight	type
56	Cinelli	Zvdeco	800	Italia	1780	Frameset
57	Raolev	Trio Gravel	650	England	2000	Frameset
58	Cinelli	Kino Zvdeco Gr...	170	Italy	1900	Frameset
64	Easton	EC90 SL Disc R...	900	England	1700	Wheelset
65	Hooe	Pro 4 Boost MT...	150	China	2000	Wheelset
66	Novatec	Model 32	90	China	2000	Wheelset
67	Mavic	Elliose	150	Italy	1400	Wheelset
70	Novatec	One	50	China	300	Wheelset
51	Fox	Suspension 40	400	USA	1740	Fork
52	Fox	Suspension 70	340	USA	2040	Fork
53	Colum...	Minimal Road F...	220	Italia	340	Fork
54	Dedacciai	RS Carbon Roa...	314	Italia	300	Fork
55	Tifosi	SR5 Road Disc ...	260	Italia	350	Fork
71	Novatec	One	50	China	300	Fork
62	Shimano	Ultegra 6800 Br...	45	Jaolan	200	Brake
63	Shimano	105 R7000 Brak...	46	Jaolan	200	Brake
59	Camosa...	Record 2x12 S...	1800	Italy	600	Grouset
60	SRAM	Red eTao AXS ...	950	USA	500	Grouset

temp_component 2

Рисунок 3.13 — Просмотр временной таблицы компонентов

```

271
272 insert into bar (width, diametr, componentID)
273 select 400+round(rand()*(8-2)+3)*10, 31.8, componentID from component c, componenttype t where name='Bar' and t.componenttypeid=c.compor
274
275
276 describe component;
277
278
279 select * from temp_component;
280
281 create temporary table temp_component
282 select a.componentid, brand, model, round(pricecent/100) pricedollar, manufacturerCountry as country, weightgramm weight, b.name as type
283 |
284 use cycle;

```

Result Grid

componentid	brand	model	pricedollar	country	weight	type
51	Fox	Suspension 40	400	USA	1740	Fork
52	Fox	Suspension 70	340	USA	2040	Fork
53	Columbus	Minimal Road Forks	220	Italia	340	Fork
54	Dedacciai	RS Carbon Road Fork	314	Italia	300	Fork
55	Tifosi	SR5 Road Disc Brake Forks	260	Italia	350	Fork
71	Novatec	One	50	China	300	Fork

Рисунок 3.14 — Просмотр временной таблицы компонентов. Компоненты «Fork»

4 Создание веб–клиента

В данном разделе будет рассмотрен процесс реализации веб–приложения, с помощью которого будет производиться взаимодействие конечного пользователя с созданной в рамках предыдущих работ базой данных.

4.1 Общие требования к приложению

Приведем технические требования, предъявляемые к разрабатываемому приложению.

4.1.1 Требования к функционалу

К функционалу приложения предъявляются следующие требования:

- регистрация и авторизация пользователя в системе;
- добавление, изменение, удаление, обновление информации по теме;
- фильтрация списков по соответствующим признакам;
- просмотр информации по запросу.

4.1.2 Требования к интерфейсу

Интерфейс системы должен поддерживать русский язык.

Интерфейс системы должен быть спроектирован с учетом ролевой модели и уровней доступа пользователей.

Интерфейс системы должен обеспечивать наглядное, интуитивно понятное представление структуры размещенной информации, быстрый и логичный переход к соответствующим разделам.

Навигационные элементы интерфейса должны обеспечивать понимание пользователем их смысла и обеспечивать навигацию по всем доступным пользователю разделам и отображать соответствующую информацию.

Интерфейс системы должен позволять решать задачи пользователя наиболее быстрым, простым и удобным из возможных способов.

Дизайн и удобство интерфейса должны быть на уровне ожиданий современного пользователя и восприниматься им как комфортная, удобная и приятная рабочая среда.

4.2 Реализация приложения

На основании функциональных требований, указанных в задании, было реализовано веб приложение, позволяющие взаимодействовать с данными, хранящимися в базе данных `sysle`, созданной в ходе выполнения предыдущих практических работ данного курса.

Интерфейс системы спроектирован с учетом ролевой модели и уровней доступа пользователей и подразумевает регистрацию пользователя в СУБД MySQL.

Интерфейс поддерживает русский язык и позволяет пользователю удобно просматривать сведения, содержащиеся в таблицах выбранной базы данных.

Акцент в приложении сделан на содержании, дизайн интерфейса прост и лаконичен, чтобы облегчить и упростить взаимодействие пользователя с системой.

4.2.1 Используемые технологии

В качестве набора программных технологий был выбран LAMP–стэк, объединяющий в себе следующие компоненты:

- Linux;
- Apache;
- MySQL;
- PHP.

Программно-аппаратная часть сервиса (бэкенд) реализована на языке PHP в парадигме объектно-ориентированного программирования, а также с использованием языка MySQL для работы с базой данных.

Структура гипертекстового документа описана средствами HTML и CSS.

Клиентская сторона пользовательского интерфейса системы (фронтэнд) основана на подходе AJAX (Asynchronous Javascript and XML). В качестве языка программирования был использован Java Script.

Запросы веб-сервиса обрабатываются сервером Apache.

4.2.2 Возможные сценария использования веб-приложения

Реализуемое веб–приложение может быть использовано для просмотра таблиц, добавления данных в таблицы базы данных под управлением СУБД MySQL, а также для написания произвольных DML и DDL запросов на диалекте MySQL.

Веб–приложение объединяет в себе часть возможностей командой строки СУБД MySQL и веб-клиента для просмотра таблиц.

4.2.3 Демонстрация работы веб-приложения

Продемонстрируем общие сценарий использования веб–приложения, созданного в рамках данной практической работы.

4.2.3.1 Авторизация

При первом входе на сайт пользователю необходимо авторизоваться и выбрать пользователя СУБД MySQL, который имеет права на изменение и просмотр данных в выбранной базе данных. Форма авторизации приведена на Рисунке 4.1.

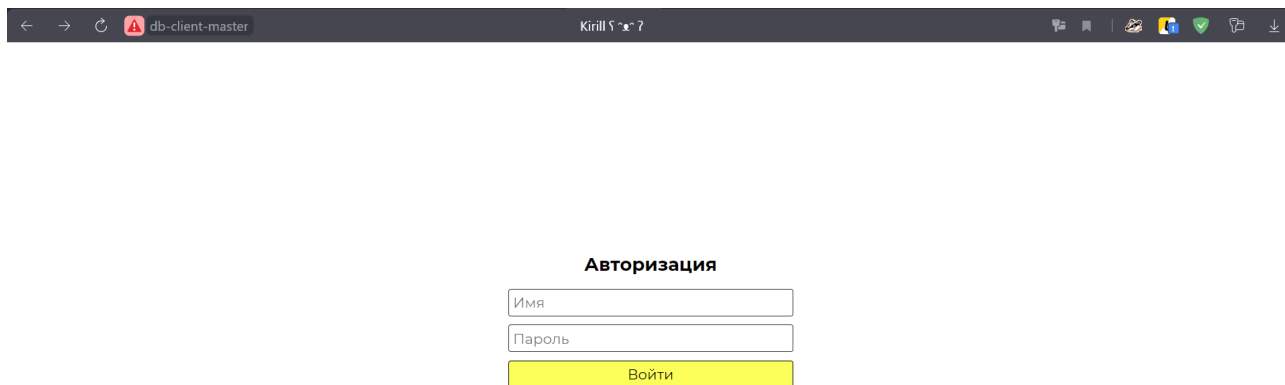


Рисунок 4.1 — Авторизация пользователя

В данном примере будет выбран пользователь `denilai`, который имеет привилегии супер-пользователя в данной базе данных и может вносить изменения в базу данных, просматривать все записи в ней, создавать новые постоянные и временные таблицы, процедуры и макросы.

4.2.3.2 Отображение записей таблиц

После успешной авторизации пользователю будет предложено выбрать таблицу в базе данных, записи в которой будут отображены на экране. Поля таблиц могут содержать кириллические символы — они будут корректно отображаться веб-приложением (см. Рисунки 4.2, 4.3).

4.2.3.3 Добавление записей в в таблицы

Ниже, под выведенными записями таблицы расположены поля для добавления значений в выбранную базу данных. В случае ввода корректных данных, после нажатия кнопки «ОК», поля будут немедленно добавлены в таблицу и отобразятся на экране (см. Рисунок 4.4).

ComponentID	Brand	ManufacturerCountry	PriceCent	ComponentTypeID	WeightGramm	Model
47	3T	Italy	9000	2	210	Superleggera
48	Deda	Italy	7000	2	200	Elementi
49	Ritchey	England	25100	2	140	Superlogic
50	FSA	USA	3000	2	220	Omega Compact
51	Fox	USA	40000	6	1740	Suspension 40
52	Fox	USA	34000	6	2040	Suspension 70
53	Columbus	Italy	22000	6	340	Minimal Road Forks
54	Dedacciai	Italy	31400	6	300	RS Carbon Road Fork
55	Tifosi	Italy	26000	6	350	SR5 Road Disc Brake Forks
56	Cinelli	Italy	80000	4	1780	Zydeco
57	Ragley	England	65000	4	2000	Trig Gravel
58	Cinelli	Italy	17000	4	1900	King Zydeco Gravel
59	Campagnolo	Italy	180000	8	600	Record 2x12 Speed Road Groupset
60	SRAM	USA	95000	8	500	Red eTap AXS 1x12 Speed Road Groupset
61	Campagnolo	Italy	60000	8	570	Centaur 11 Speed Rim Brake Road Groupset
62	Shimano	Japan	4500	7	200	Ultegra 6800 Brake Caliper
63	Shimano	Japan	4600	7	200	105 R7000 Brake Caliper

Рисунок 4.2 — Отображение записей таблицы Component

ComponentID	Brand	ManufacturerCountry	PriceCent	ComponentTypeID	WeightGramm	Model
47	3T	Italy	9000	2	210	Superleggera
48	Deda	Italy	7000	2	200	Elementi
49	Ritchey	England	25100	2	140	Superlogic
50	FSA	USA	3000	2	220	Omega Compact
51	Fox	USA	40000	6	1740	Suspension 40
52	Fox	USA	34000	6	2040	Suspension 70
53	Columbus	Italy	22000	6	340	Minimal Road Forks
54	Dedacciai	Italy	31400	6	300	RS Carbon Road Fork
55	Tifosi	Italy	26000	6	350	SR5 Road Disc Brake Forks
56	Cinelli	Italy	80000	4	1780	Zydeco
57	Ragley	England	65000	4	2000	Trig Gravel
58	Cinelli	Italy	17000	4	1900	King Zydeco Gravel
59	Campagnolo	Italy	180000	8	600	Record 2x12 Speed Road Groupset
60	SRAM	USA	95000	8	500	Red eTap AXS 1x12 Speed Road Groupset
61	Campagnolo	Italy	60000	8	570	Centaur 11 Speed Rim Brake Road Groupset
62	Shimano	Japan	4500	7	200	Ultegra 6800 Brake Caliper
63	Shimano	Japan	4600	7	200	105 R7000 Brake Caliper

Рисунок 4.3 — Отображение записей таблицы Log

4.2.3.4 Выполнение произвольных запросов

В самом низу веб-страницы расположено текстовое поле для выполнения SQL-запросов. Запросы могут относиться не только к выбранной таблице или базе данных, а к любой таблице, права на взаимодействие с которой имеет выбранный пользователь.

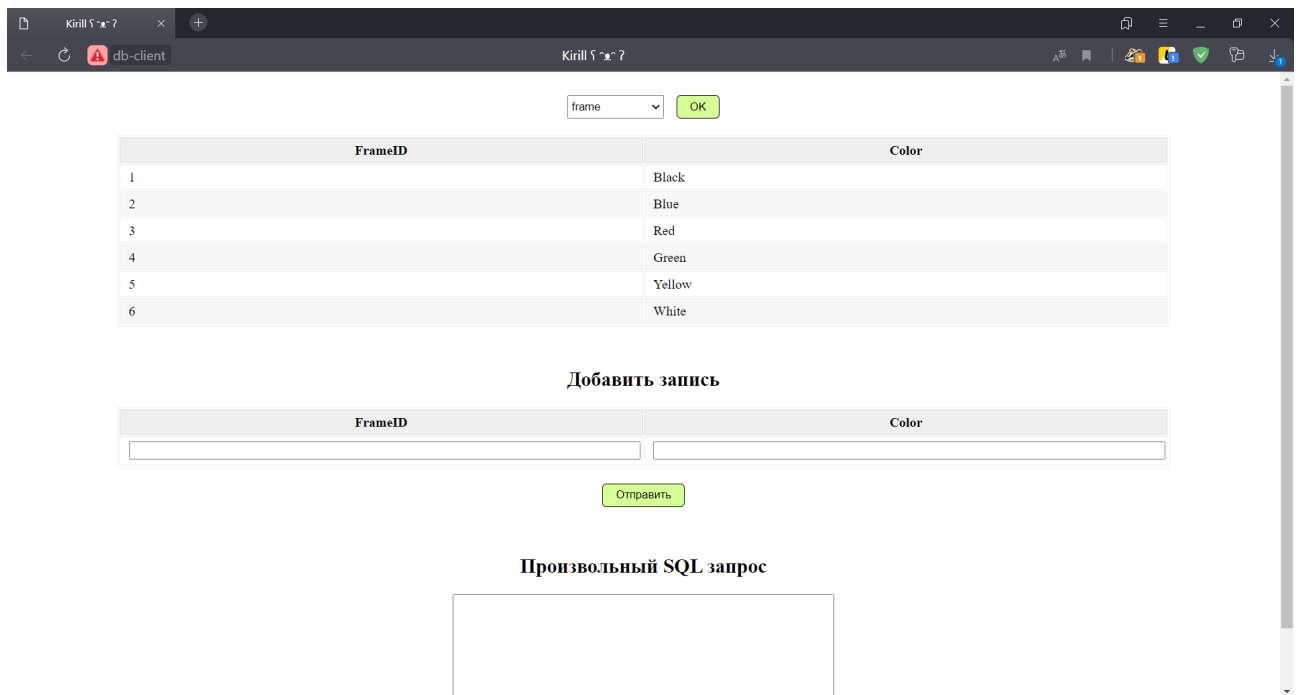


Рисунок 4.4 — Добавление записей в таблицу Frame

Текст, введенный пользователем, передается в виде строки на исполнение MySQL серверу. В случае ввода корректного запроса, он будет выполнен, о чем будет сообщено пользователю.

С помощью данного поля также можно производить операции по удалению и вставке записей в таблицы, отображаемые на экране (см. Рисунок 4.5).

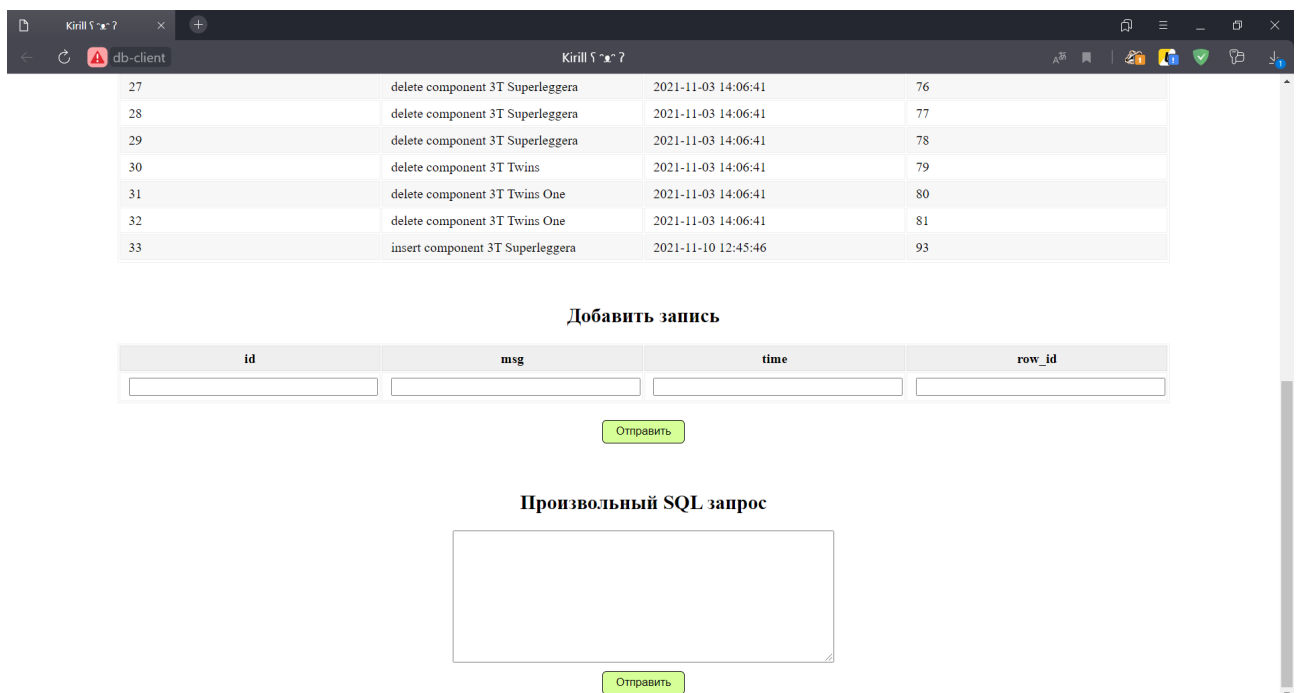


Рисунок 4.5 — Выполнение произвольных запросов

С помощью данного веб–клиента пользователь может просматривать записи, хранящиеся в базе данных, отмечать продвижение деталей по производственным цехам, отражать текущий статус сборки велосипеда, проверять наличие тех или иных компонентов на складе, получать полную информацию о них. Интерфейс отвечает требованиям, выдвинутым в задании: поддерживает русский язык, акцент в приложении сделан на содержании, дизайн интерфейса прост и лаконичен, чтобы упростить взаимодействие пользователя с системой.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данных практических работ были получены знания по написанию запросов на диалекте MySQL, отработаны навыки работы в средах ER Assistant и ERwin Data Modeler, навыки работы с интерфейсом командной строки MySQL CLI Client, а также с инструментом для визуального проектирования баз данных MySQL Workbench.

Полученные знания были применены на практике для построения проектов баз данных, физические и логические модели баз данных. В последствии данные базы данных были реализованы в СУБД MySQL. Также было создано пользовательское веб – приложение, представляющее удобный интерфейс для взаимодействия с информационной системой и базой данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Дюбуа, Поль*. MySQL: Полн. и исчерпывающее руководство по применению и администрированию баз данных MySQL 4, а также программированию приложений / Поль Дюбуа. — Издательский дом Вильямс, 2004.
2. *Балдин, Евгений Михайлович*. Компьютерная типография LaTeX / Евгений Михайлович Балдин. — Evgeny Baldin, 2008.
3. *Converse, Tim*. PHP5 and MySQL bible / Tim Converse, Joyce Park, Clark Morgan. — John Wiley & Sons, 2004. — Vol. 147.
4. *Юртанова, Екатерина Михайловна*. Разработка Web-приложений с использованием языка PHP / Екатерина Михайловна Юртанова // *Учебный эксперимент в образовании*. — 2011. — С. 47–50.