



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий
Кафедра промышленной информатики (ПИ)

ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине
«Разработка баз данных»

Выполнил: студент группы ИВБО-02-19

К. Ю. Денисов

Принял: инженер, ассистент

Д. С. Киселев

Работа выполнена «____» _____ 202__

«Зачтено» «____» _____ 202__

Москва 2021

Содержание

1	Построение модели данных	3
1.1	Работа по методическим указаниям	3
1.2	Индивидуальный вариант	5
2	Создание логической и физической модели данных	7
2.1	Работа по методическим указаниям	7
2.2	Индивидуальный вариант	9
3	Создание базы данных	11
3.1	Работа по методическим указаниям	11
3.2	Индивидуальный вариант	14
4	Создание веб-клиента	19
4.1	Общие требования к приложению	19
4.2	Реализация приложения	20

1 Построение модели данных

1.1 Работа по методическим указаниям

Описание предметной области

Компания занимается производством и продажей небольших статуэток, раскрашиваемых вручную. Компания имеет несколько производственных направлений. Миниатюры изготавливаются из гипса, олова или алюминия.

Компания распространяет свои товары по трем каналам. Компания содержит пять собственных розничных магазинов. Помимо этого, компания владеет сайтом, на котором ведется online-торговля, и осуществляет оптовые поставки сторонним дистрибьюторам. Для анализа статистики, системой автоматизации производства, нужен интерактивный аналитический инструмент. Поэтому необходимо спроектировать и построить модель данных, которая станет хранилищем информации по производству.

В ходе производства изделий система автоматизации производства управляет всеми станками компании. Каждый станок реализует полный цикл производства изделий, включая:

- заполнение формы сырьем (гипсом, оловом или алюминием);
- затвердевание материала;
- удаление изделия из формы после затвердевания;
- при необходимости автоматизированная раскраска изделий (оловянные фигурки не раскрашиваются);
- сушку после покраски (при необходимости).

Покраска и сушка могут производиться за несколько этапов в зависимости от сложности изделия. По мере готовности изделия проходят проверку, выполняемую оператором станка.

Оператор станка регистрируется в системе. В ходе этого процесса оператор сообщает системе автоматизации производства тип производимых изделий и объем загруженного в машину сырья. Оператор также делает в системе запись при отбраковке изделий.

В ходе интервью необходимые для эффективного анализа статистики:

- число принятых изделий по объему сырья, видам изделий, машинам и
- время формовки и затвердевания по видам изделий, машинам и дням;

1.2 Индивидуальный вариант

Описание предметной области

В качестве индивидуального задания была выбрана реализация модели информационной системы по хранению и анализу данных предприятия, занимающегося сборкой и поставкой спортивных велосипедов для конечных потребителей по индивидуальному заказу.

Предприятие располагает широким выбором компонентов и комплектующих для сборки велосипедов следующих типов:

- Дорожный;
- Горный;
- Кросс-кантри;
- Эндуро;
- Прогулочный.

Предприятие может предложить сконфигурировать велосипед, отдельно выбрав каждый из предложенных компонентов:

- Рама;
- Вилка;
- Руль;
- Трансмиссия;
- Колеса;
- Тормозная система.

Контроль над выполнением работ по сборке велосипеда проводится в виде учета всех операций по сборке, настройке и тестированию, проводимых на территории предприятия ее сотрудниками. При этом каждая запись содержит следующую информацию о проведенных работах:

1. Внутренний номер изделия;
2. Время;
3. Этап работ;
4. Название цеха;
5. Имя мастера;
6. Статус;
7. Примечание.

2 Создание логической и физической модели данных

2.1 Работа по методическим указаниям

Теоретический материал

На этапе инфологического проектирования базы данных должна быть построена модель предметной области, не привязанная к конкретной СУБД, понятная не только разработчикам информационной системы, но и экономистам, менеджерам и другим специалистам. В то же время модель предметной области должна максимально точно отражать семантику предметной области и позволять легко перейти к модели данных конкретной СУБД.

Логический уровень — это уровень, соответствующий инфологическому этапу проектирования и не привязанный к конкретной СУБД. Модели логического уровня оперируют с понятиями сущностей, атрибутов и связей, которые на этом уровне именуется на естественном языке (в нашем случае – на русском) так, как они называются в реальном мире.

Физический уровень — это отображение логической модели на модель данных конкретной СУБД. Одной логической модели может соответствовать несколько физических моделей. Причем, Erwin (как и другие CASE-системы проектирования баз данных) позволяет автоматизировать отображение логической модели на физическую.

В ходе данной практической работы нами была описана и построена логическая и физическая модель данных средствами ПО **ERwin Data Modeler**.

Порядок построения модели данных в среде **ERwin** рассмотрим на примере автоматизированной информационной системы "Реализация средств вычислительной техники предназначенной для учета продаж настольных компьютеров по заказам клиентов.

Создание модели данных начинается с разработки логической модели, которая должна представлять состав сущностей предметной области с перечнем атрибутов и отношений между ними.

Результат разработки логической модели данных системы «Реализация средств вычислительной техники», предназначенной для учета продаж на-

стольных компьютеров по заказам клиентов приведен на Рисунке 3.

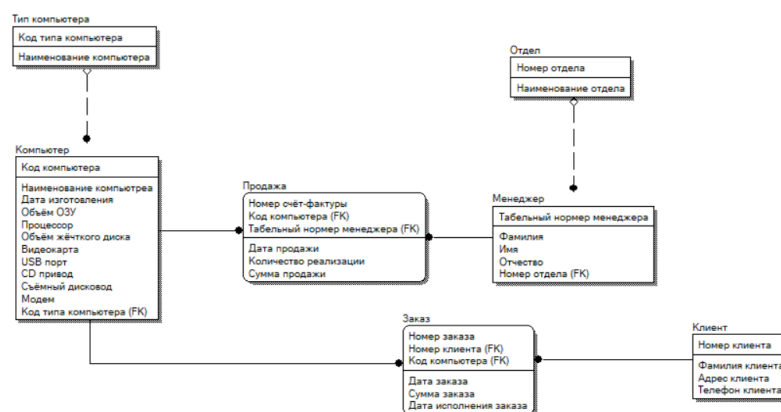


Рисунок 3 — Логическая модель данных системы «Реализация средств вычислительной техники»

Для построения физической модели данных системы, следует определиться с СУБД, в которой будет реализована модель. При построении физической модели данных следует учитывать формальную теория представления и обработки данных в конкретной системе управления базами данных (СУБД).

В данной практической работе в качестве СУБД выбрана MySQL.

Приступим к построению физической модели данных системы «Реализация средств вычислительной техники». Результат работы можно видеть на рисунке 4.

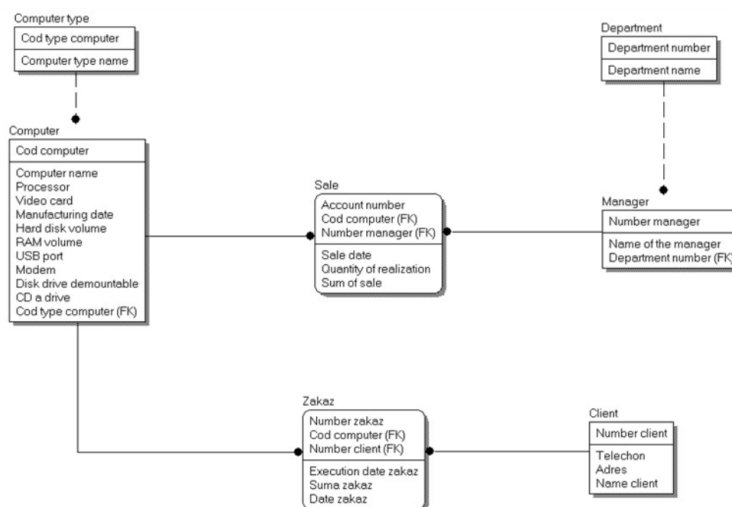


Рисунок 4 — Физическая модель данных системы «Реализация средств вычислительной техники»

2.2 Индивидуальный вариант

Приступим к построению логической модели данных системы «Велосипедное предприятие». В соответствии с моделью, реализованной в ходе первой практической работы, добавим в рабочую область следующие сущности:

- Component;
- FrameInfo;
- Frame;
- Frameset;
- FrameSize;
- Fork;
- ComponentType;
- Wheelset;
- Groupset;
- Brake;
- FctCycleBuild;
- CycleType;
- Bar;
- Setup.

Добавим связи между сущностями в соответствии с ранее построенной моделью. Логическая модель системы «Велосипедное предприятие» приведена на рисунке 5.

После уточнения типов данных, выбранных в соответствии с предметной областью и спецификой СУБД MySQL. Физическая модель системы «Велосипедное предприятие» приведена на рисунке 6.

После реализации физической и логической модели можно приступать к реализации модели данной системы в СУБД MySQL.

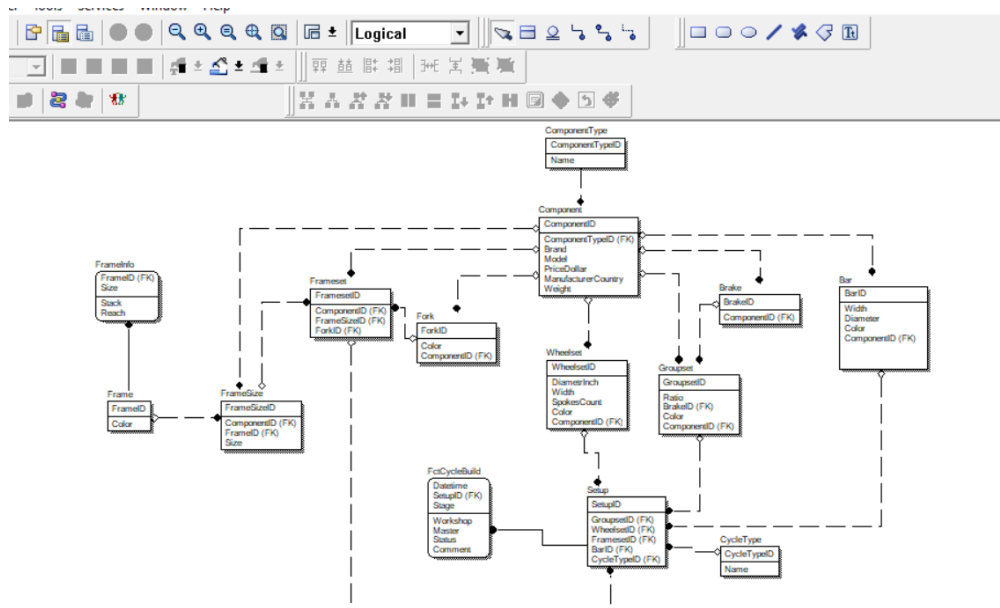


Рисунок 5 — Логическая модель данных системы «Велосипедное предприятие»

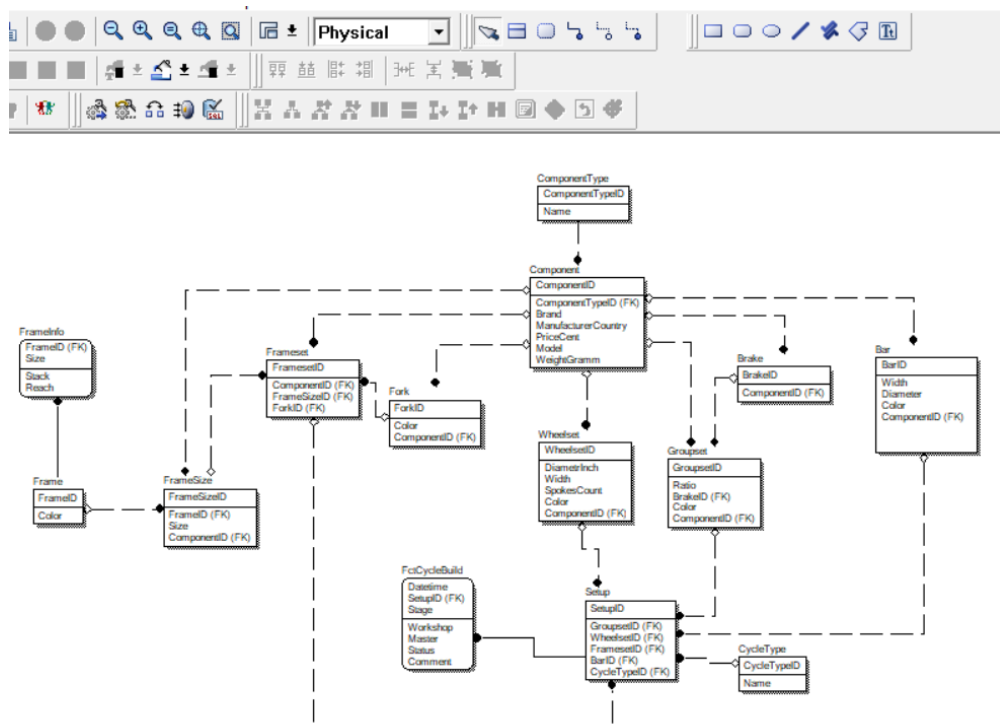


Рисунок 6 — Физическая модель данных системы «Велосипедное предприятие»

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 7):

```
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 7

Создадим таблицу `topics` (см. рисунок 8):

```
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 8

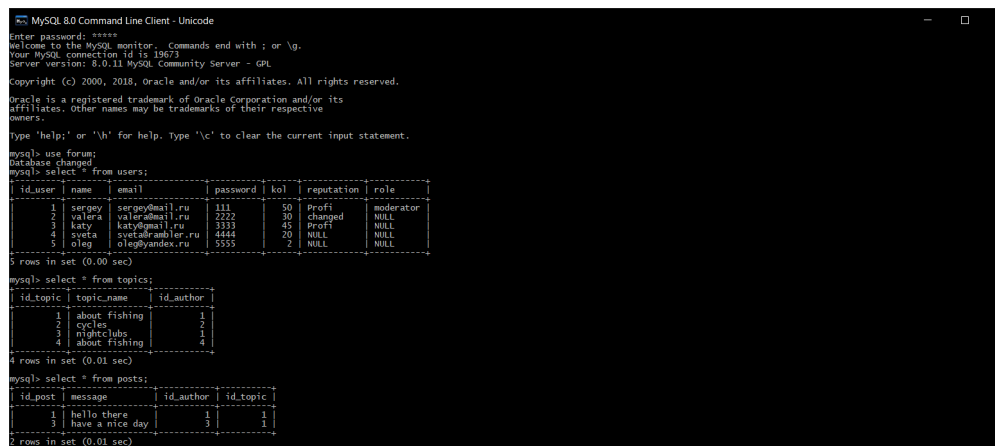
Создадим таблицу `posts` (см. рисунок 9):

```
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 9

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 10).



```
MySQL 8.0 Command Line Client - Unicode
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 | katya | katya@mail.ru | 3333 | 45 | Profi | NULL |
| 4 | sveta | svetadrambler.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)
```

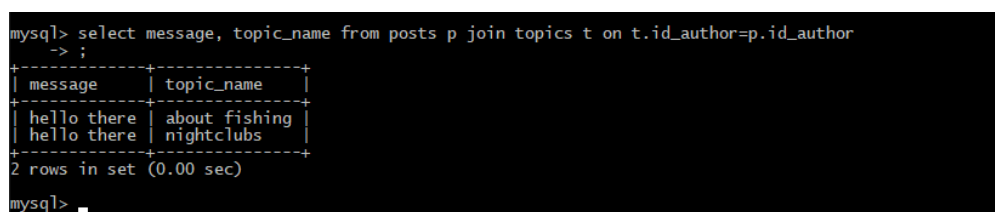
Рисунок 10 — Операция выборки из всех таблиц

Выполним запрос `SELECT message, topic_name FROM posts p JOIN topics t ON t.id_author = p.id_author;` для объединения данных из таблиц `topics` и `posts` по ключу `id_author` и получения полной информации о сообщении и названию темы, в которой оно было размещено (см. рисунок 11).

Выполним запрос выборки данных, явно указав поля отношения. Для этого перечислим имена полей через запятую после зарезервированного слова `SELECT` (см. рисунок 12).

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 13).

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.



```
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 11 — Запрос объединения

```
MySQL 8.0 Command Line Client
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@rambler.ru |
| sveta | sveta@rambler.ru |
| oleg | oleg@yandex.ru |
+-----+-----+
5 rows in set (0.00 sec)
```

Рисунок 12 — Операция выборки с указанием полей

```
MySQL 8.0 Command Line Client
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 13 — Сложные запросы выборки с сортировкой и условием

Выведем все записи из таблицы `users`, обнаружим, что столбцы были вставлены успешно (см. рисунок 14).

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> 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@mail.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 (21000): 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@mail.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 14 — Добавление полей в таблицу users

3.2 Индивидуальный вариант

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

Описание проектируемых отношений базы данных `cycle` приведены в таблицах 1-15.

Таблица 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 2 — Описание таблицы Brake

Имя	Тип
BrakeID	int NOT NULL auto_increment
ComponentID	INTEGER NULL

Таблица 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 4 — Описание таблицы ComponentType

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

Таблица 5 — Описание таблицы CycleType

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

Таблица 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 7 — Описание таблицы Fork

Имя	Тип
ForkID	INTEGER NOT NULL auto_increment
Color	VARCHAR(20) NULL default 'black'
ComponentID	INTEGER NULL

Таблица 8 — Описание таблицы Frame

Имя	Тип
FrameID	INTEGER NOT NULL auto_increment
Color	VARCHAR(20) NULL default 'black'

Таблица 9 — Описание таблицы FrameInfo

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

Таблица 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 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 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 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 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 15 — Описание таблицы Log

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

Создадим данные таблицы в СУБД MySQL, используя ключевое слово `CREATE TABLE`. После создания таблиц выполним команду `SHOW TABLES`, выбрав базу данных `cycle` (см. рисунок 15).

Выведем записи из таблицы `Component` (см. рисунок 16).

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 17).

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.

```

MySQL 8.0 Command Line Client

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 15 — Таблицы базы данных cycle

```

MySQL 8.0 Command Line Client

mysql> select * from component;
+-----+-----+-----+-----+-----+-----+
| ComponentID | Brand | ManufacturerCountry | PriceCent | ComponentTypeID | WeightGram | 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 | Dejarial | Italia | 31400 | 6 | 300 | RS Carbon Road Fork |
| 55 | Tifosi | Italia | 26000 | 6 | 350 | SR5 Road Disc Brake Forks |
| 56 | Cinelli | Italia | 80000 | 4 | 1700 | Zydco |
| 57 | Ragley | England | 65000 | 4 | 2000 | Trig Gravel |
| 58 | Cinelli | Italy | 17000 | 4 | 1000 | King Zydco 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 R7800 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 | Havic | 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 16 — 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 |
| 4 | 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 17 — Сложные запросы выборки. База данных cycle

Приведем объявление данного триггера на диалекте MySQL:

```
DELIMITER $$
2 drop trigger delete_component; $$

4 create trigger 'delete_component' after delete on component
  for each row begin
6 insert into log (msg, row_id) values (concat('delete component ',old.Brand, ' ',
  , old.Model), old.ComponentID);
end; $$

8
create trigger 'add_component' after insert on component
10 for each row begin
insert into log (msg, row_id) values (concat('insert component ',new.Brand, ' ',
  , new.Model), new.ComponentID);
12 end; $$
DELIMITER $$
```

4 Создание веб-клиента

4.1 Общие требования к приложению

Требования к функционалу

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

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 `cycle`, 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.

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.

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 (Data Manipulation Language) и DDL (Data Definition Language) запросов на диалекте MySQL.

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

Демонстрация работы веб-приложения

Продemonстрируем сценарий использования созданного в рамках данной практической работы веб-приложения.

Авторизация При первом входе на сайт пользователю необходимо авторизоваться — выбрать пользователя СУБД MySQL, который имеет права на изменение и просмотр данных в выбранной базе данных. В данном примере будет выбран пользователь `denilai`, который имеет привилегии супер-пользователя в данной базе данных — может вносить изменения в базу данных, просматривать все записи в ней, создавать новые постоянные и временные таблицы, процедуры и макросы.

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 18, 19).

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

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	SR5 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

Рисунок 18 — 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	Italia	22000	6	340	Minimal Road Forks
54	Dedacciai	Italia	31400	6	300	RS Carbon Road Fork
55	Tifosi	Italia	26000	6	350	SR5 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

Рисунок 19 — Отображение записей таблицы Log

будут немедленно добавлены в таблицу и отобразятся на экране (см. рисунок 20).

FrameID	Color
1	Black
2	Blue
3	Red
4	Green
5	Yellow
6	White

Добавить запись

FrameID	Color
<input type="text"/>	<input type="text"/>

Отправить

Произвольный SQL запрос

Рисунок 20 — Добавление записей в таблицу Frame

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.

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 21).

The screenshot displays a web application window titled 'db-client'. It features a table with the following data:

id	msg	time	row_id
27	delete component 3T Superleggera	2021-11-03 14:06:41	76
28	delete component 3T Superleggera	2021-11-03 14:06:41	77
29	delete component 3T Superleggera	2021-11-03 14:06:41	78
30	delete component 3T Twins	2021-11-03 14:06:41	79
31	delete component 3T Twins One	2021-11-03 14:06:41	80
32	delete component 3T Twins One	2021-11-03 14:06:41	81
33	insert component 3T Superleggera	2021-11-10 12:45:46	93

Below the table is a section titled 'Добавить запись' (Add record) with a form containing four input fields labeled 'id', 'msg', 'time', and 'row_id'. A green 'Отправить' (Send) button is positioned below the form.

Below the form is a section titled 'Произвольный SQL запрос' (Arbitrary SQL query) with a large text area for input. A green 'Отправить' (Send) button is positioned below the text area.

Рисунок 21 — Выполнение произвольных запросов