ГОСУДАРСТВЕННОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ЯРОСЛАВСКОЙ ОБЛАСТИ

РОСТОВСКИЙ ПЕДАГОГИЧЕСКИЙ КОЛЛЕДЖ

КУРСОВОЙ ПРОЕКТ

**Создание конструктора запросов MySQL**

Выполнил: Горлов Евгений Сергеевич

Специальность 09.02.05 Прикладная информатика (по отраслям)

Курс III группа 35

Научный руководитель: преподаватель

Сабуров Павел Андреевич

Форма обучения: очная

РОСТОВ

2018

Государственное профессиональное образовательное учреждение   
Ярославской области

Ростовский педагогический колледж

|  |  |  |
| --- | --- | --- |
| Рассмотрено на заседании цикловой комиссии  Протокол № \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  от «\_\_\_» \_\_\_\_\_\_\_\_\_201\_г.  Председатель комиссии  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.О.Фамилия |  | УТВЕРЖДАЮ  Зам. директора по методической работе \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.О.Фамилия  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 201\_г. |

**ЗАДАНИЕ**

**на курсовое проектирование по профессиональному модулю (модулям)**

**ПМ.02 Разработка, внедрение и адаптация программного обеспечения отраслевой направленности.**

**09.02.05 Прикладная информатика (по отраслям)**

*(код и наименование специальности)*

Студенту \_\_\_\_\_\_3\_\_\_\_\_\_ курса \_\_\_35\_\_\_ группы

Ф.И.О студента\_\_\_\_\_\_Горлову Евгению Сергеевичу\_\_\_\_\_\_\_

Тема и исходные данные к проекту: Создание конструктора запросов MySQL

Вопросы, подлежащие разработке:

|  |  |
| --- | --- |
| 1. Описание предметной области, постановка цели и задач | Октябрь (1 декада) |
| 1. Построение инфологической (концептуальной) модели предметной области | Октябрь (2 декада) |
| 1. Проектирование логической и физической структур информационной системы | Октябрь (3 декада) |
| 1. Разработка прототипа | Ноябрь - декабрь |
| 1. Предзащита | Декабрь – январь |
| 1. Подготовка технологического раздела пояснительной записки | Ноябрь - февраль |
| 1. Реализация проекта в конкретной среде | Февраль – март |
| 1. Разработка основных программных модулей продукта |
| 1. Выбор стратегии тестирования и анализ программного продукта | Март – апрель |
| 1. Подготовка проектного раздела пояснительной записки |
| 1. Оформление проекта |
| 1. Защита | Апрель (3 декада) |

Дата выдачи задания «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 201\_г.

Срок представления законченной работы «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 201\_г.

Научный руководитель курсового проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия руководителя)

Задание получил «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 201\_г.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия студента)

Пояснительная записка

курсового проекта

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc517519162)

[1. Технологический раздел 5](#_Toc517519163)

[2.1 описание объекта автоматизации 5](#_Toc517519164)

[2.2 Создание информационной модели 7](#_Toc517519165)

[2. ПРОЕКТНЫЙ РАЗДЕЛ 13](#_Toc517519166)

[2.1. Формирование базы знаний и выбор инструментов разработки 13](#_Toc517519167)

[2.2. Разработка программного продукта 14](#_Toc517519168)

[ЗАКЛЮЧЕНИЕ 19](#_Toc517519169)

[ИНФОРМАЦИОННЫЕ РЕСУРСЫ 20](#_Toc517519170)

[ПРИЛОЖЕНИЯ 21](#_Toc517519171)

# ВВЕДЕНИЕ

База данных MySQL — это самая популярная в мире база данных с открытым кодом. Благодаря своей проверенной производительности, надежности и относительной простоте использования база данных MySQL наиболее часто используется для веб-приложений. Кроме того, ее особенно часто выбирают в качестве встроенной базы данных, распространяемой тысячами поставщиков программного обеспечения и производителей оборудования.

MySQL использует интерфейс командной строки, что вызывает сложности в обучении у современных пользователей, привыкших к удобным графическим интерфейсам. Для облегчения освоения студентами этого мощного инструмента и предназначен "Конструктор запросов MySQL".

**Цель:**

Создать веб-интерфейс для облегчения освоения MySQL

**Задачи:**

1. Собрать базу знаний о MySQL.
2. Разработать информационную модель.
3. Выбрать и освоить инструменты разработки
4. Реализовать модуль запросов на действия с выводом собранного запроса в отдельном поле.

# Технологический раздел

## **описание объекта автоматизации**

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Сообществом разработчиков MySQL созданы различные ответвления кода, такие как Drizzle, OurDelta, Percona Server и MariaDB. Все эти ответвления уже существовали на момент поглощения компании Sun корпорацией Oracle.

О происхождении MySQL:

MySQL возникла как попытка применить mSQL к собственным разработкам компании: таблицам, для которых использовались ISAM — подпрограммы низкого уровня. В результате был выработан новый SQL-интерфейс, но API-интерфейс остался в наследство от mSQL. Откуда происходит название «MySQL» — доподлинно неизвестно. Разработчики дают два варианта: либо потому, что практически все наработки компании начинались с префикса My, либо в честь девочки по имени My, дочери Майкла Монти Видениуса, одного из разработчиков системы.

Логотип MySQL в виде дельфина носит имя «Sakila». Он был выбран из большого списка предложенных пользователями «имён дельфина». Имя «Sakila» было отправлено Open Source-разработчиком Ambrose Twebaze.

Лицензирование:

MySQL имеет двойное лицензирование. MySQL может распространяться в соответствии с условиями лицензии GPL. Однако по условиям GPL, если какая-либо программа использует библиотеки (или включает в себя другой GPL-код) MySQL, то она тоже должна распространяться по лицензии GPL. Это может расходиться с планами разработчиков, не желающих открывать исходные тексты своих программ. Для таких случаев предусмотрена коммерческая лицензия, которая также обеспечивает качественную сервисную поддержку. Для свободного программного обеспечения Oracle предоставляет отдельное исключение из правил, явным образом разрешающее использование и распространение MySQL вместе с ПО, распространяемым под лицензией из определённого Oracle списка.

Максимальные размеры таблиц:

Максимальный размер таблиц в MySQL 3.22 до 4 гигабайт, в последующих версиях ограничений нет.

Размер таблицы ограничен её типом. В общем случае тип MyISAM ограничен предельным размером файла в файловой системе операционной системы. Например, в NTFS этот размер теоретически может быть до 32 эксабайт. В случае InnoDB одна таблица может храниться в нескольких файлах, представляющих единое табличное пространство. Размер последнего может достигать 64 терабайт.

В отличие от MyISAM, в InnoDB имеется значительное ограничение на количество столбцов, которое можно добавить в одну таблицу. Размер страницы памяти по умолчанию составляет 16 килобайт, из которых под данные отведено 8123 байта. Размер указателя на динамические поля составляет 20 байт. Таким образом, в случае использования динамического формата строки (ROW\_FORMAT=DYNAMIC), одна таблица может вместить максимум 409 столбцов типа blob или text.

MySQL использует интерфейс командной строки, устаревший в наше время, но значительно более надежный в сравнении с графическими интерфейсами. Этот факт вызывает сложности в обучении у современных программистов и веб-мастеров, привыкших к удобным и понятным графическим интерфейсам. Эту проблему призван решить наш "Конструктор запросов MySQL".

## **Создание информационной модели**

Прототипирование программного обеспечения — этап разработки программного обеспечения (ПО), процесс создания прототипа программы — макета (черновой, пробной версии) программы, обычно — с целью проверки пригодности предлагаемых для применения концепций, архитектурных и/или технологических решений, а также для представления программы заказчику на ранних стадиях процесса разработки.

Создание прототипа программного продукта позволяет:

* Грамотно продумать логику работы программы.
* Оценить концепцию продукта и выявить изъяны.
* Продумать возможности взаимодействия пользователей с программой.

Кроме того, разработка прототипа существенно помогает сэкономить время разработки проекта и сократить число доработок, которые возникают при несоответствии функционала продукта ожиданиям заказчика. После представления прототипа клиент знает, чего ожидать в результате, а разработчик уверен в целях и требованиях.

Прототипирование начинается с анализа цели программного продукта и принятия решения о составе схем и диаграмм, реализующих информационную модель программного обеспечения.

Для нашей работы мы выбрали следующие схемы и диаграммы:

* функциональная диаграмма
* функциональная схема
* диаграмма переходов состояний
* диаграмма вариантов использования

Функциональными называют диаграммы, в первую очередь отражающие взаимосвязи функций разрабатываемого программного обеспечения.

Они создаются на ранних этапах проектирования систем, для того чтобы помочь проектировщику выявить основные функции и составные части проектируемой системы и, по возможности, обнаружить и устранить существенные ошибки. В нашей функциональной диаграмме главной функцией является, создание и вывод запросов (рис. 1).

Входными данными в нашем случае являются: база знаний MySQL. Выходными, запрос на действие и справочная информация. Механизм в нашем случае определяется техническим заданием, а управляется все это пользователем и скриптом конструктора запросов.

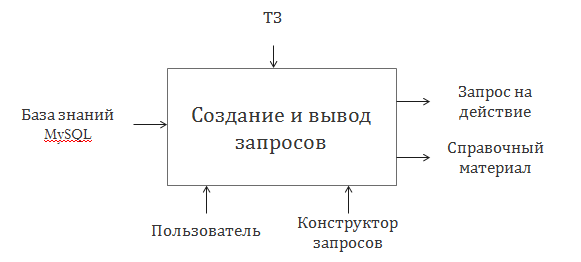


Рисунок 1. Функциональная диаграмма конструктора запросов

Функциональная схема или схема данных это — схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Для изображения функциональных схем используют специальные обозначения, установленные стандартом ГОСТ 19.701-90.

Наша схема отображает концепцию системы в целом (рис. 2). Сначала пользователь выбирает параметры для создания запроса, затем система выдает ему готовый запрос и справочную информацию, после чего пользователь может начать все заново или закончить работу с конструктором запросов.

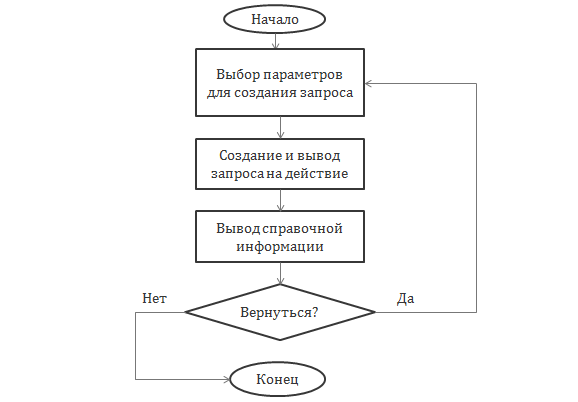


Рисунок 2. Функциональная схема конструктора запросов

Диаграмма переходов состояний это – известная технология описания поведения системы. В том или ином виде диаграмма состояний существует с 1960 года, и на заре объектно-ориентированного программирования они применялись для представления поведения системы. В объектно-ориентированных подходах вы рисуете диаграмму состояний единственного класса, чтобы показать поведение одного объекта в течение его жизни.

В нашем случае диаграмма показывает модуль запросов на действие, и отображает логику его работы (рис. 3). По ней мы видим что первое состояние после входа это отображение выбора типа действия. После чего пользователь может выбрать одну из трех форм:

* Параметры для запроса на действие с базой данных
* Параметры для запроса на действие с таблицей
* Параметры для запроса на действие с полем таблицы

Сделав свой выбор, пользователь вводит все необходимые данные и получает готовый запрос и справочную информацию. При этом он в любой момент может переместится в любую из предоставленных форм.

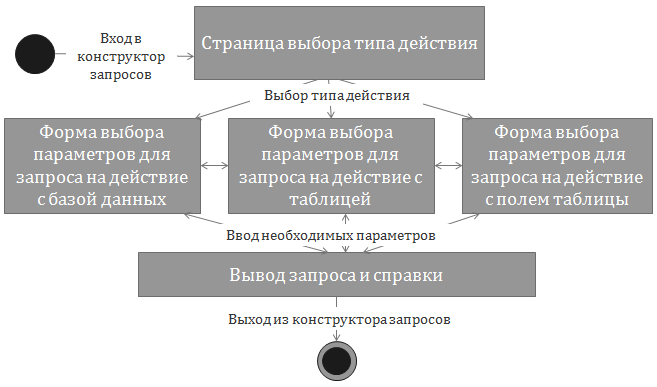


Рисунок 3. Диаграмма переходов состояний модуля запросов на действие

Основное назначение диаграммы вариантов использования — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы вариантов использования системный аналитик стремится:

* чётко отделить систему от её окружения;
* определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;
* определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть прецедентов).

В нашей диаграмме видно что есть один вариант использования конструктора запросов пользователем, для получения запроса и справочной информации, по средствам взаимодействия с формами конструктора (рис. 4).

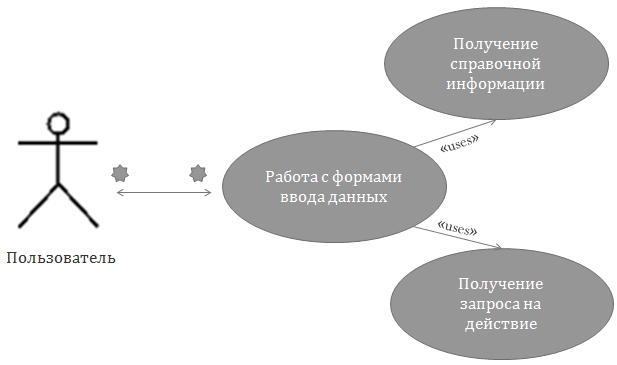


Рисунок 4. Диаграмма вариантов использования конструктора БД

Разработав информационную модель мы получили всю необходимую информацию для оценки объема предстоящих работ и выбора инструментов разработки. Теперь мы можем приступить к проектированию программного продукта.

# ПРОЕКТНЫЙ РАЗДЕЛ

## **Формирование базы знаний и выбор инструментов разработки**

Изучив существующие технологии по созданию баз данных MySQL, нами был выявлен факт того что есть множество web-платформ, с очень широким функционалом, таких как devart.com и mytaskhelper.ru. Они предоставляют пользователям обширный список возможностей, как в проектировании, так и в интеграции готовых баз данных MySQL. Однако все они имеют один общий недостаток: для неограниченного доступа ко всему функционалу требуется оформить платную подписку, или купить лицензию. Кроме того ни один из популярных ресурсов не имеет встроенного справочника. Что означает неспособность пользователя самостоятельно освоить MySQL и, используя интерфейс командной строки, создать базу данных и интегрировать в свой продукт. Несмотря на обширный функционал этих сервисов, они не могут быть использованы для разработки самостоятельных продуктов.

Для достижения поставленной задачи мы решили что наш программный продукт не будет самостоятельно применять запросы, он будет выводить MySQL запрос и справочную информацию, для того что бы пользователь мог самостоятельно освоить и применить эти знания.

Для реализации интерфейса мы решили использовать технологии, HTML и CSS. Они идеально подходят для разработки сайтов и могут быть, достаточно быстро, освоены, более того они входят в основной курс как отдельные предметы.

Скрипт, лежащий в основе конструктора, было решено реализовать на языке программирования JavaScript. Он отлично подходит для подобных задач, и тоже входит в основной курс.

Собрав и отсортировав справочную информацию о MySQL, мы проанализировали ее, и смогли выбрать наиболее подходящие инструменты разработки. Теперь можно приступать непосредственно к созданию нашего программного продукта.

## **2.2. Разработка программного продукта**

По мере освоения HTML и CSS, открывались все новые и новые возможности. В связи с этим менялась и концепция дизайна, изначально архаичный, он становился все современнее и удобней.

Параллельно с разработкой дизайна, проходил сбор всей возможной информации о MySQL и создание подробного справочника. Из за большого объема данных было решено выделить отдельные темы в отдельные страницы. Так же выделялась информация для коротких справок, и подсказок.

JavaScript основан на синтаксисе языка программирования C++, который был в учебной программе, что позволило быстро освоить его и начать реализовывать необходимые функции.

После первоначального освоения инструментов началась комплексная работа, с использованием всех технологий и вскоре был разработан финальный вариант интерфейса (рис. 5).

Выбор пал на минимализм и современность. Используя оттенки синего цвета на белом фоне, удалось добиться отличной контрастности, и при этом сохранить цветовой баланс. Ни один элемент не выбивается из общей картины, и не отвлекает, а благодаря минимализму и подсказкам, на сайте легко найти интересующую информацию.

Была проделана большая работа по реализации справочника, информация в нем разделена по темам и красиво оформлена, что облегчает восприятие. Кроме того, реализована ссылка в начало страницы, облегчающая работу с длинными статьями, а также ссылки перехода к следующей статье. В совокупности эти элементы значительно облегчили навигацию по статьям.

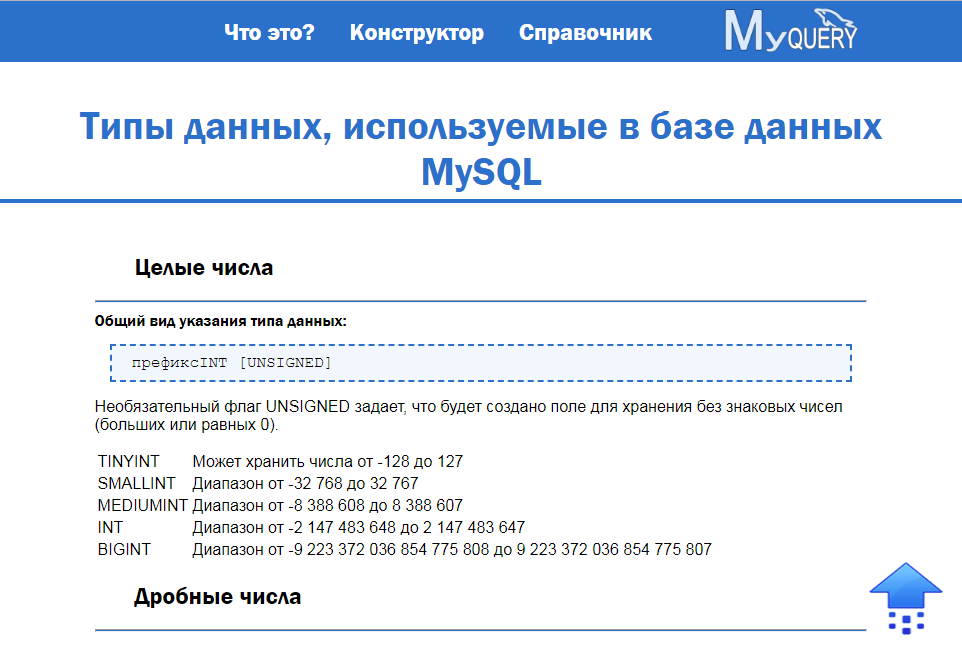


Рисунок 5. Дизайн.

С помощью JavaScript были реализованы вкладки (рис. 6), обеспечивающие, необходимое по плану разделение на три модуля (рис. 7):

* действия с БД
* действия с таблицами
* действия с полями

Использован метод замены стиля у объекта, это позволило добиться единого стиля с другими элементами дизайна.

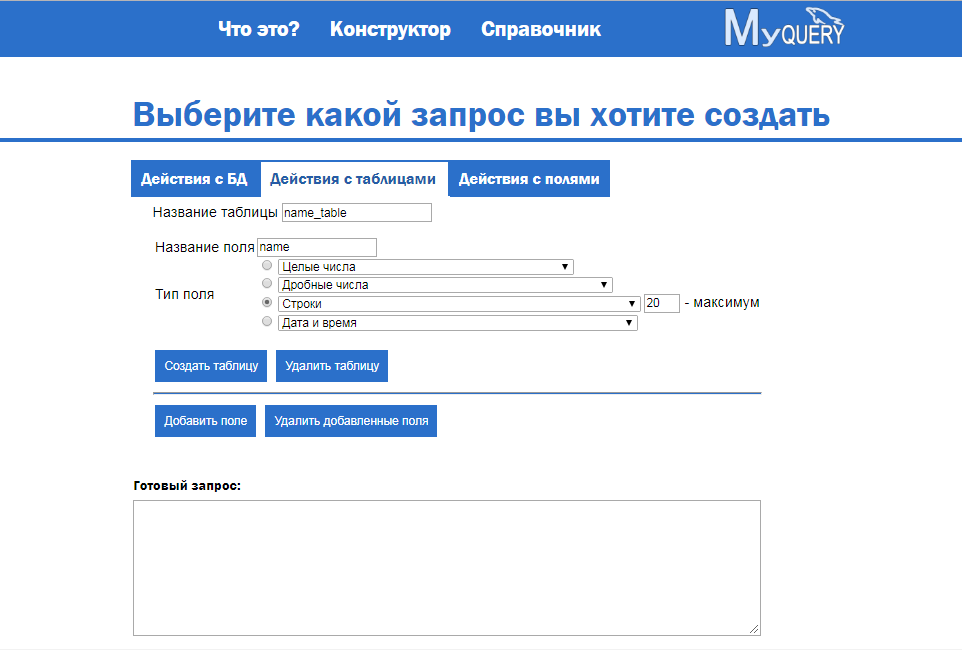


Рисунок 6. Реализация вкладок.

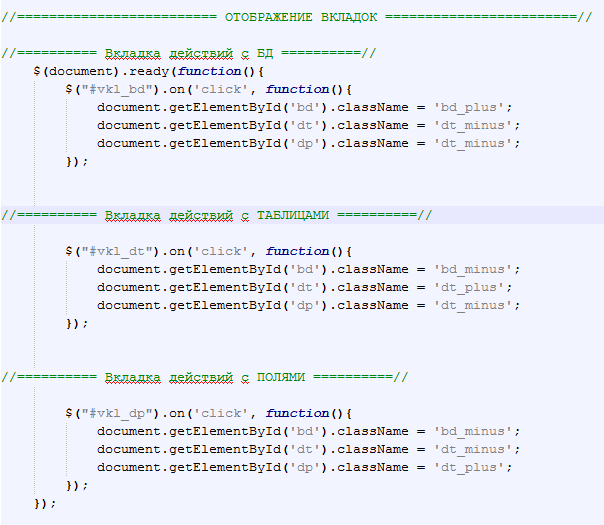


Рисунок 7. Реализация вкладок JavaScript.

Существует лишь два основных действия непосредственно с базами данных, это создание и удаление, поэтому были использованы две простые функции (рис. 8).

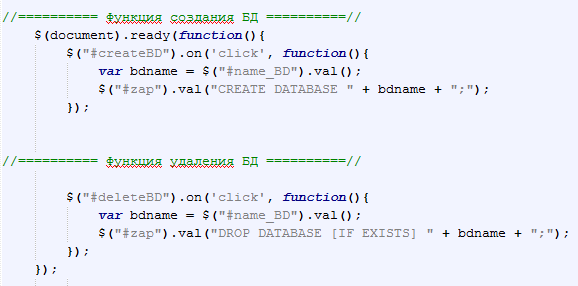


Рисунок 8. Скрипт запроса на действие с БД.

По техническому заданию необходимо было предоставить пользователю выбор типа поля таблицы, а так же реализовать добавление нескольких полей из за чего скрипт получился чрезмерно громоздким, поэтому было решено производить обработку нажатия отдельно от обработки введенной информации (рис. 9).

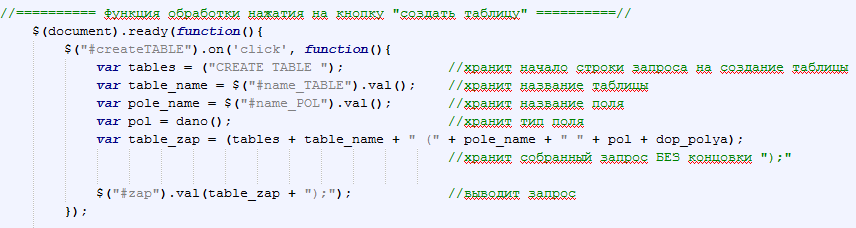


Рисунок 9. Функция обработки нажатия на кнопку "создать таблицу".

Активно используя возможности инструментов разработки, были реализованы все необходимые элементы, а так же подсказки, которые призваны помочь пользователю, быстрее изучить MySQL.  
Подсказки реализованы в виде окошка с короткой справкой и всплывающих подсказок.(рис. 11)

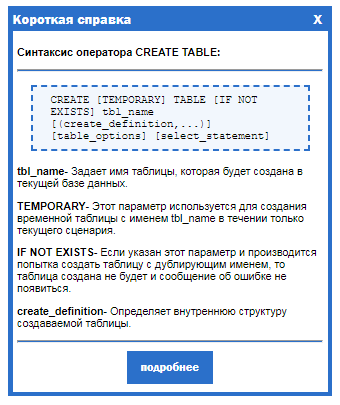


Рисунок 11. Окно справки.

После проделанной работы мы получили веб-сайт, с современным, интуитивно понятным дизайном и встроенным конструктором запросов, при работе с которым можно быстро получить справочную информацию по конкретным темам.

# ЗАКЛЮЧЕНИЕ

Целью нашего курсового проекта было создание веб-интерфейса для облегчения освоения MySQL. Для ее достижения мы выполнили следующие задачи:

Собрали базу знаний о MySQL и проанализировали ее.

На основе полученных выводов мы разработали информационную модель, и оценили объем предстоящих работ. Так же мы получили данные для грамотного выбора инструментов разработки.

На основе справочной информации и нашей информационной модели мы выбрали наиболее подходящие инструменты разработки.

Разработали веб-сайт, с современным, интуитивно понятным дизайном и встроенным конструктором запросов, при работе с которым можно быстро получить справочную информацию по конкретным темам.

В настоящее время сайт готов к вводу в эксплуатацию, но предполагает перспективу своего дальнейшего развития по следующим направлениям:

* добавление модуля запросов на выборку;
* возможность создания локальной базы данных;
* размещение на хостинге и продвижение.

# ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Рудаков А.В. Технология разработки программных продуктов. Практикум: учеб. пособие для студ. учреждений сред. проф. образования / А.В. Рудаков, Г.Н. Федорова. - 4-е изд., стер - М.: Издательский центр "Академия"; 2014. - 192 с.
2. Рудаков А.В. Технология разработки программных продуктов. учеб. пособие для студ. учреждений сред. проф. образования / А.В. Рудаков, - 2-е изд., стер - М.: Издательский центр "Академия"; 2006. - 208 с.
3. SpravkaWeb.Ru [Электронный ресурс] : информационный ресурс, посвященный программированию для Internet. — Электрон. дан. — М. : Интернет сайт SpravkaWeb.Ru, 2011. — Режим доступа: http//www.spravkaweb.ru/mysql, свободный. — Загл. с экрана. — Яз. рус.
4. ВикипедиЯ Свободная энциклопедия [Электронный ресурс] : свободная энциклопедия, которую может редактировать каждый. — Электрон. дан. — Режим доступа: https://ru.wikipedia.org, свободный. — Загл. с экрана. — Яз. рус., англ.
5. htmlbook.ru [Электронный ресурс] : информационный ресурс, посвященный технологиям HTML и CSS. — Электрон. дан. — М. : Интернет сайт SpravkaWeb.Ru, [2002 - 2018]. — Режим доступа: https//www. htmlbook.ru, свободный. — Загл. с экрана. — Яз. рус.
6. Современный учебник Javascript [Электронный ресурс] : информационный ресурс, посвященный технологии Javascript. — Электрон. дан. — М. : Интернет сайт learn.javascript.ru, [2007 - 2018]. — Режим доступа: https://learn.javascript.ru/, свободный. — Загл. с экрана. — Яз. рус.

# ПРИЛОЖЕНИЯ