МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное

учреждение высшего образования

«Казанский национальный исследовательский технический

университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

СПО ИКТЗИ

Колледж информационных технологий

Расчетно-пояснительная записка к

курсовой работе

по дисциплине «Технология разработки и защиты баз данных»

тема «Авиаперелет»

|  |  |
| --- | --- |
| Выполнил: | обучающийся группы 4238 |
|  | Халиков А.С. |
| Проверил: | Касимова А.Р. |

Курсовая работа зачтена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Казань 2021г.

Оглавление

[Введение 4](#_Toc72734957)

[Глава 1. Описание программных средств 5](#_Toc72734958)

[1.1 Программа XAMPP 5](#_Toc72734959)

[1.2 PHP, HTML, CSS 6](#_Toc72734960)

[Глава 2. Проектирование БД 7](#_Toc72734961)

[2.1 Концептуальное проектирование 7](#_Toc72734962)

[2.2 Логическое проектирование 8](#_Toc72734963)

[2.3 Физическое проектирование 9](#_Toc72734964)

[Глава 3. База данных. 10](#_Toc72734965)

[3.1 Создание БД 10](#_Toc72734966)

[3.2 Создание таблиц 11](#_Toc72734967)

[3.3 Создание связей 14](#_Toc72734968)

[3.4 Заполнение таблиц 15](#_Toc72734969)

[Глава 4. Создание Web-форм 16](#_Toc72734970)

[4.1 Работа с самолётами 16](#_Toc72734971)

[4.2 Работа с аэропортами 20](#_Toc72734972)

[4.3 Работа с рейсами 23](#_Toc72734973)

[4.4 Работа с вылетами 26](#_Toc72734974)

[4.5 Работа с пассажирами 31](#_Toc72734975)

[4.6 Регистрация 34](#_Toc72734976)

[Глава 5. Простейшие криптографические способы защиты базы данных 36](#_Toc72734977)

[Заключение 39](#_Toc72734978)

[Список использованных источников 40](#_Toc72734979)

[Приложения 41](#_Toc72734980)

[Приложение A 41](#_Toc72734981)

[Приложение B 41](#_Toc72734982)

[Приложение С1 42](#_Toc72734983)

[Приложение С2 43](#_Toc72734984)

[Приложение С3 45](#_Toc72734985)

[Приложение D1 46](#_Toc72734986)

[Приложение D2 47](#_Toc72734987)

[Приложение D3 48](#_Toc72734988)

[Приложение E1 50](#_Toc72734989)

[Приложение E2 51](#_Toc72734990)

[Приложение E3 53](#_Toc72734991)

[Приложение F1 54](#_Toc72734992)

[Приложение F2 55](#_Toc72734993)

[Приложение F3 57](#_Toc72734994)

[Приложение G1 58](#_Toc72734995)

[Приложение G2 60](#_Toc72734996)

[Приложение G3 61](#_Toc72734997)

[Приложение H 62](#_Toc72734998)

# Введение

Различные авиакомпании ежедневно организуют множество перелётов. Для хранения данных о каждом авиаперелёте используются базы данных поскольку они значительно упрощают работу с большим объёмом данных. В своей курсовой работе я буду использовать программу, позволяющую работать с веб-сервером, XAMPP и язык гипертекстовой разметки HTML и CSS для разработки сайта.

Объектом исследования является процесс разработки базы данных, а предметом курсовой работы является разработка БД по теме «Авиаперелёты» с пользовательским интерфейсом.

Цель курсовой работы – создание БД «Авиаперелёты» с пользовательским интерфейсом.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Изучить возможности программы XAMPP, а также языков php, html, css;

2. Проанализировать предметную область и спроектировать БД;

3. Создать БД с необходимыми таблицами и связями;

4. Разработать пользовательский интерфейс, используя web-формы;

5. Реализовать хеширование паролей и создать триггеры;

# Глава 1. Описание программных средств

## 1.1 Программа XAMPP

Для работы сайта нам нужно три программы: веб-сервер (Apache 2 или nginx), PHP-интерпретатор (PHP 7) и база данных (MySQL).

XAMPP — кроссплатформенная сборка локального веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер.

Преимущество XAMPP'а в том, что он экономит время: достаточно его скачать и установить и я сразу же могу начать разрабатывать на PHP, избежав настройку всех программ по отдельности.

В состав XAMPP входит несколько больше того, что я описал:

* Apache 2 — веб-сервер, который может работать на разных платформах (Windows, Linux и другие)
* MariaDB — замена базе данных MySQL, являющаяся достаточной для PHP-разработки и практически копией MySQL
* PHP — интерпретатор PHP последней или предпоследней версии
* phpMyAdmin — веб-интерфейс для управления базой данных MariaDB
* OpenSSL — программа для работы с SSL
* XAMPP Control Panel — панель управления самим XAMPP (через нее можно запускать и останавливать Apache, MySQL, Tomcat и др.)
* Webalizer — позволяет анализировать логи веб-сервера Apache (не используется при разработке)
* Mercury Mail Transport System — почтовый сервер
* FileZilla FTP Server — FTP-сервер
* Strawberry Perl — реализация Perl для Windows

## 1.2 PHP, HTML, CSS

PHP - это распространённый язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML.

Основная сфера применения языка PHP — это веб. Важно понимать, что веб — это не весь интернет, а только самая большая и популярная его часть. Кроме веба через интернет работает электронная почта, мессенджеры, файлообменные сервисы, сетевые игры и многое другое.

HTML — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами и полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

CSS — формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки. CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом.

# Глава 2. Проектирование БД

Базы данных – это программы, которые позволяют сохранять и получать большие объемы связанной информации. Базы данных состоят из таблиц, которые содержат информацию. Когда я создаю базу данных необходимо подумать о том, какие таблицы мне нужно создать и какие связи существуют между информацией в таблицах. Иначе говоря, мне нужно подумать о проекте моей базы данных. Хороший проект базы данных обеспечит целостность данных и простоту их обслуживания. Моя БД будет содержать 5 связанных между собой таблиц.

Для корректного проектирования сначала следует выделить основные сущности и выразить глаголами их взаимосвязь. Проделав логическое проектирование я получил логическую модель данных (Рисунок 2), в которой выделил 5 сущностей. Далее следует выделить основные поля для каждой таблицы. Среди них должно быть поле, представляющее первичный ключ. В результате я получил концептуальную модель данных (Рисунок 1). Следующий этап – это определить тип данных для каждого поля и организовать грамотные связи, используя поля. Как итог, я сформировал физическую модель данных (Рисунок 3).

## 2.1 Концептуальное проектирование

Концептуальное проектирование — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности.

Чаще всего концептуальная модель базы данных включает в себя:

* описание информационных объектов или понятий предметной области и связей между ними.
* описание ограничений целостности, то есть требований к допустимым значениям данных и к связям между ними.

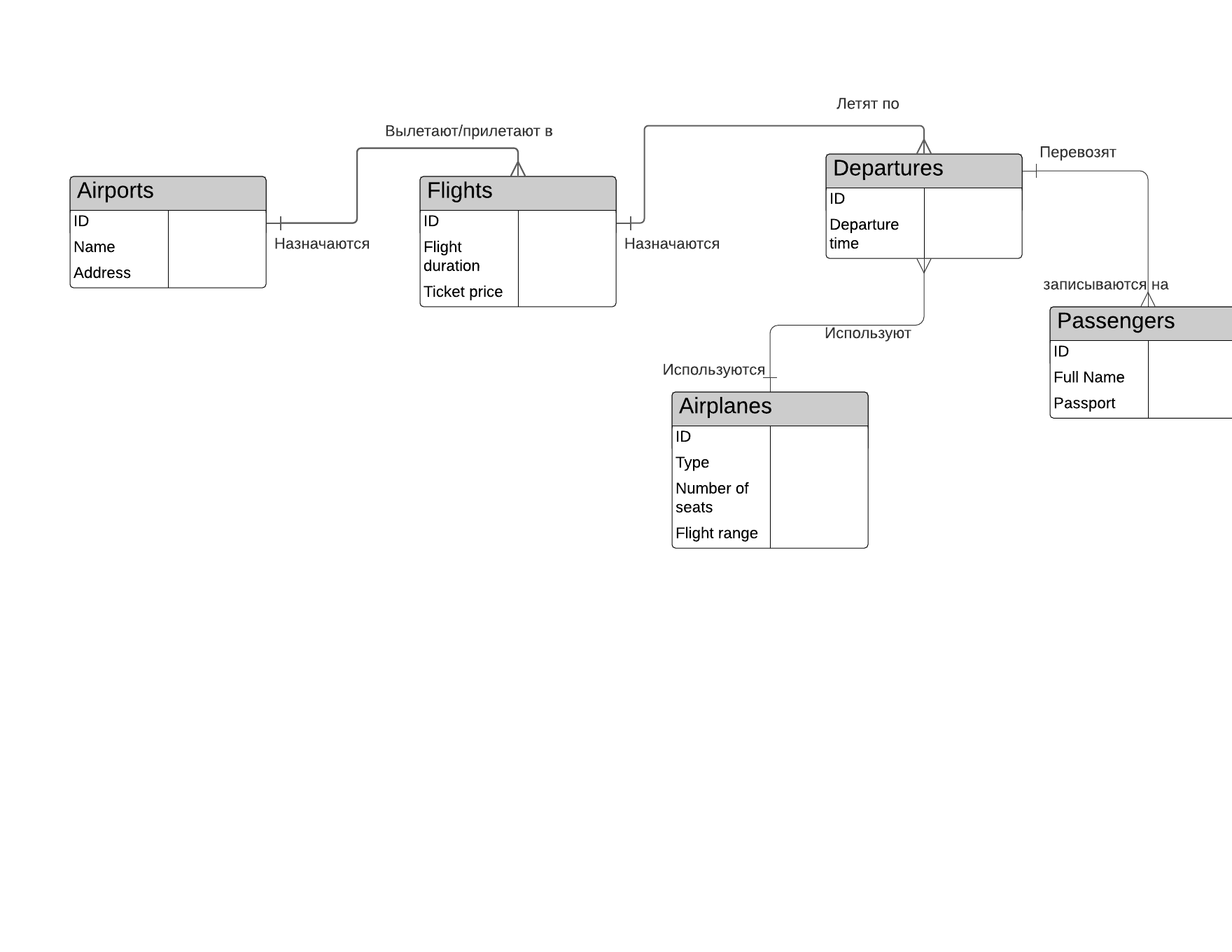


Рисунок 1. Концептуальная модель данных

## 2.2 Логическое проектирование

Логическое проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

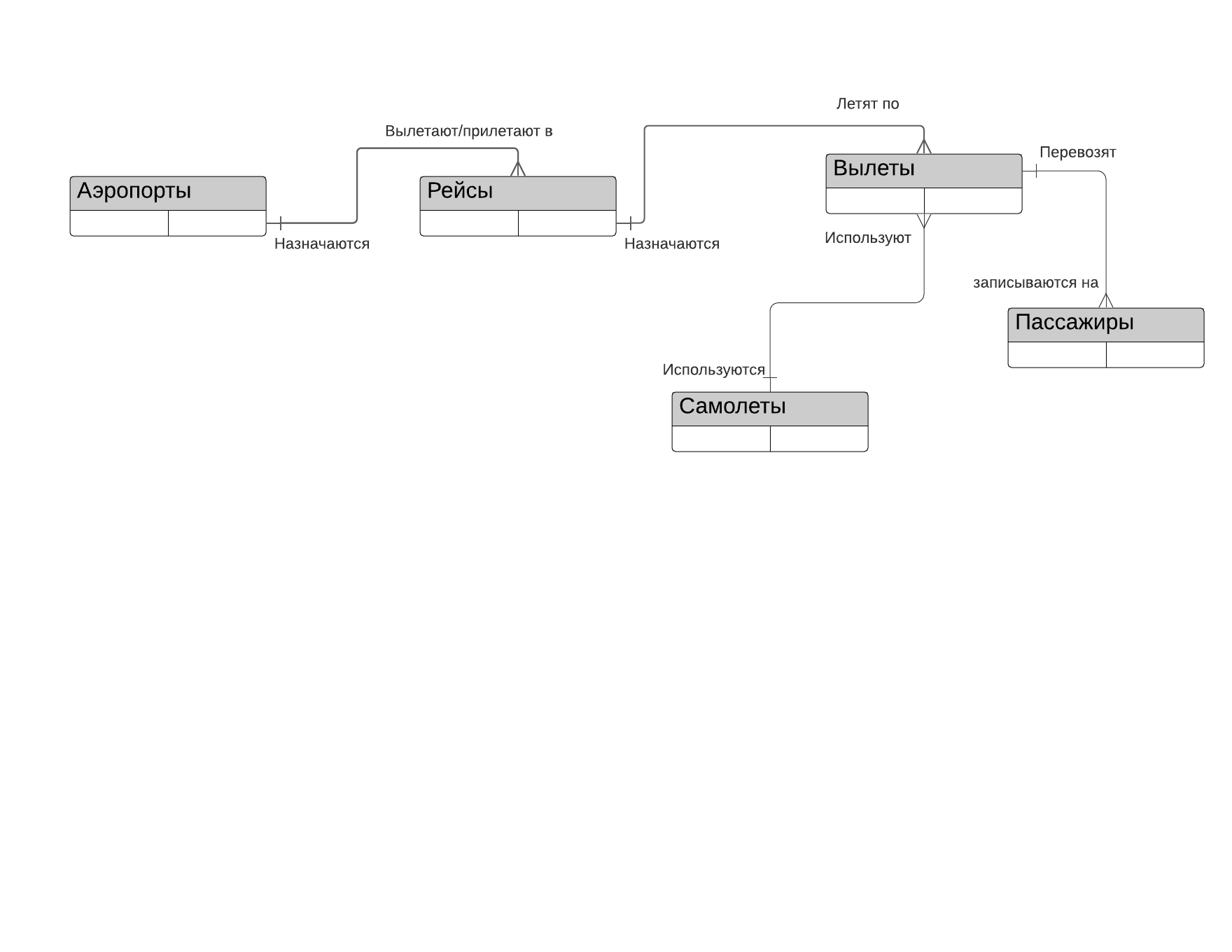


Рисунок 2. Логическая модель данных

## 2.3 Физическое проектирование

Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.

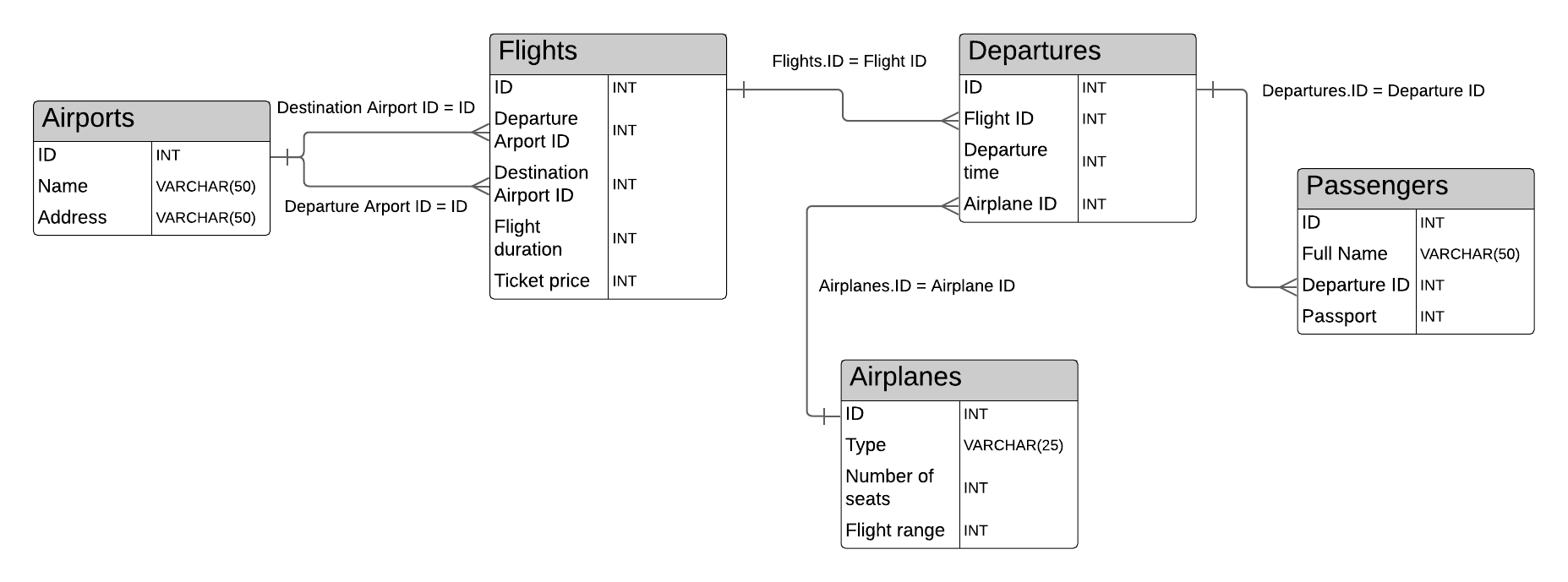


Рисунок 3. Физическая модель данных

# Глава 3. База данных.

База данных — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных. В этой главе будет идти речь о процессе создании базы данных на основе приложения XAMPP.

Для начала нужно запустить сервисы Apache и MySQL, нажав рядом с ними кнопку “Start”. Затем перейти на веб-интерфейс работы с приложением, нажав на кнопку “Admin” рядом с MySQL. Открылся браузер с интерфейсом самого приложения.

## 3.1 Создание БД

Для создания БД необходимо нажать на кнопку “Создать БД” (пункт 1 на Рисунок 4), после чего я ввёл имя своей базы данных и нажал на кнопку создать (пункты 2 и 3 на Рисунок 4). Успешно создав базу данных, можно переходить к созданию таблиц.

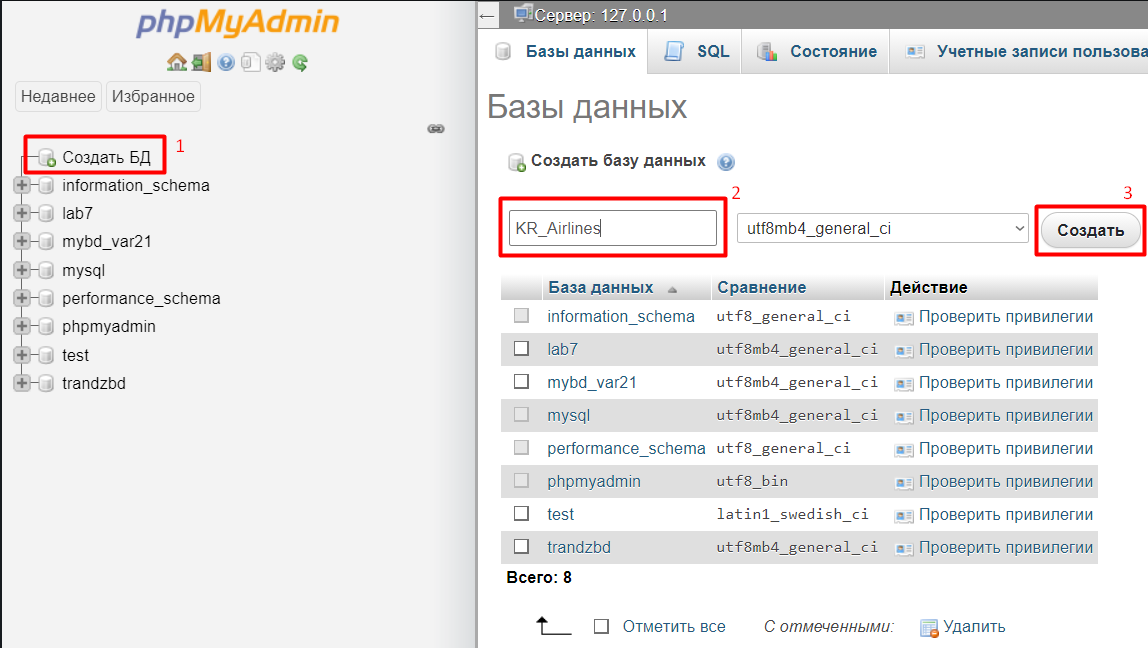


Рисунок 4. Создание БД

## 3.2 Создание таблиц

Для создания таблицы необходимо выбрать необходимую БД, после чего ввести название таблицы и количество столбцов. Затем следует нажать на кнопку “Вперёд”. Данные действия показаны на Рисунок 5.

Далее заполняем названия полей и выбираем из списка нужный тип данных для каждого поля. Для поля с первичным ключом устанавливаем атрибут PRIMARY и галочку возле параметра A\_I (AUTO\_INCREMENT).

Для создания таблицы Аэропортов заполняем три поля: Id типа данных INT с атрибутом PRIMARY KEY и AUTO\_INCREMENT; Name типа данных VARCHAR с количеством символов – 30; Address типа данных VARCHAR с количеством символов – 30 (Рисунок 6).

Для создания таблицы Рейсов заполняем пять полей: Id типа данных INT с атрибутом PRIMARY KEY и AUTO\_INCREMENT; Flight\_Duration типа данных INT; Ticket\_Price типа данных INT; Departure\_Airport\_Id типа данных INT, которое далее будет использовано в качестве внешнего ключа; Destination\_Airport\_Id типа данных INT, которое далее будет использовано в качестве внешнего ключа (Рисунок 7).

Для создания таблицы самолетов заполняем четыре поля: Id типа данных INT с атрибутом PRIMARY KEY и AUTO\_INCREMENT; Type типа данных VARCHAR с количеством символов – 30 (); Number\_Of\_Seats типа данных INT; Flight\_Range типа данных INT (Рисунок 8).

Для создания таблицы вылетов заполняем четыре поля: Id типа данных INT с атрибутом PRIMARY KEY и AUTO\_INCREMENT; Departure\_Time типа данных INT; Flight\_Id типа данных INT, которое далее будет использовано в качестве внешнего ключа; Airplane\_Id типа данных INT, которое далее будет использовано в качестве внешнего ключа (Рисунок 9).

Для создания таблицы пассажиров заполняем четыре поля: Id типа данных INT с атрибутом PRIMARY KEY и AUTO\_INCREMENT; Full\_Name типа данных VARCHAR с количеством символов – 50 (); Passport типа данных INT; Departure\_Id типа данных INT, которое далее будет использовано в качестве внешнего ключа (Рисунок 10).

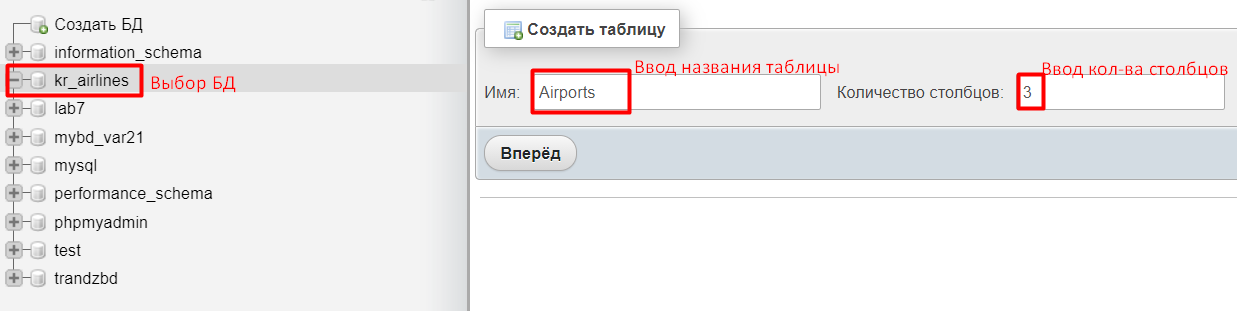


Рисунок 5. Создание таблицы

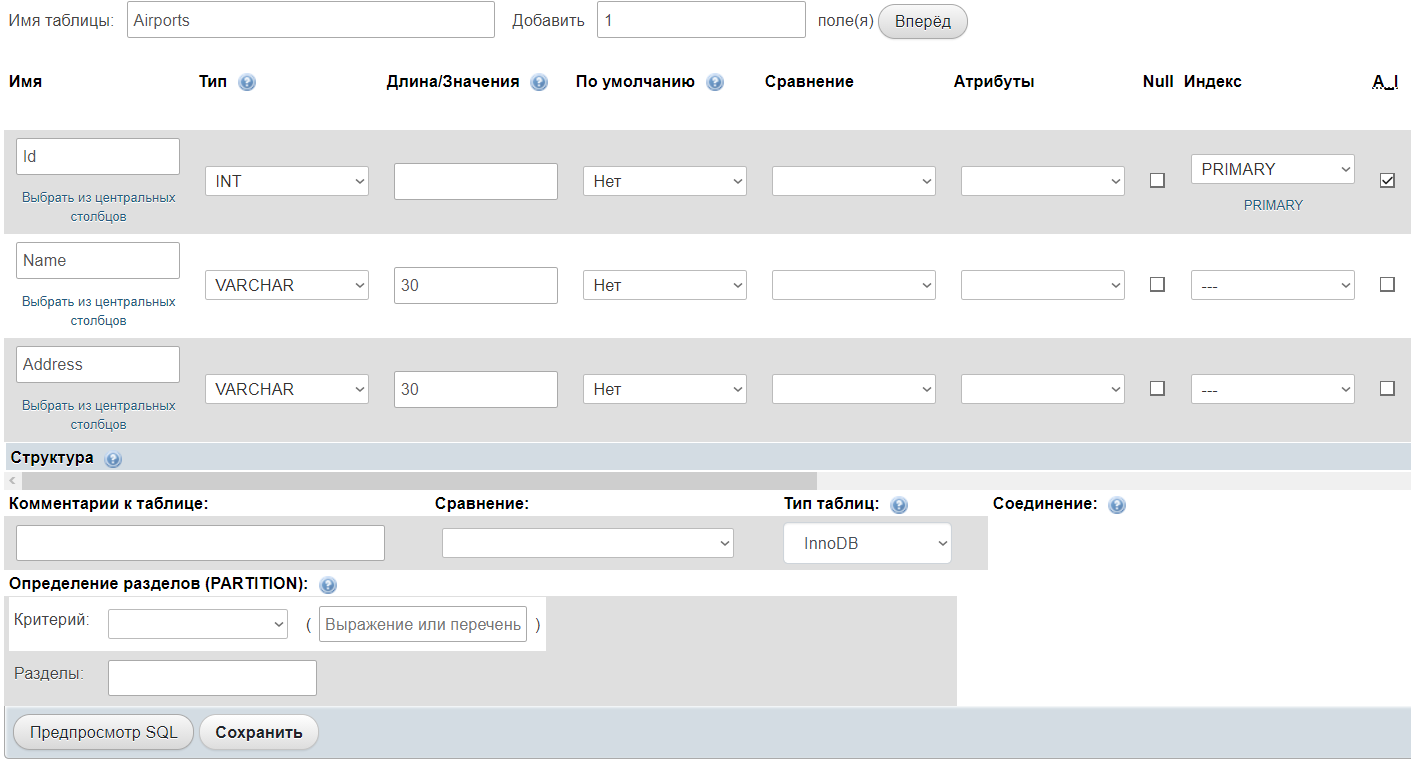


Рисунок 6. Создание таблицы Airports

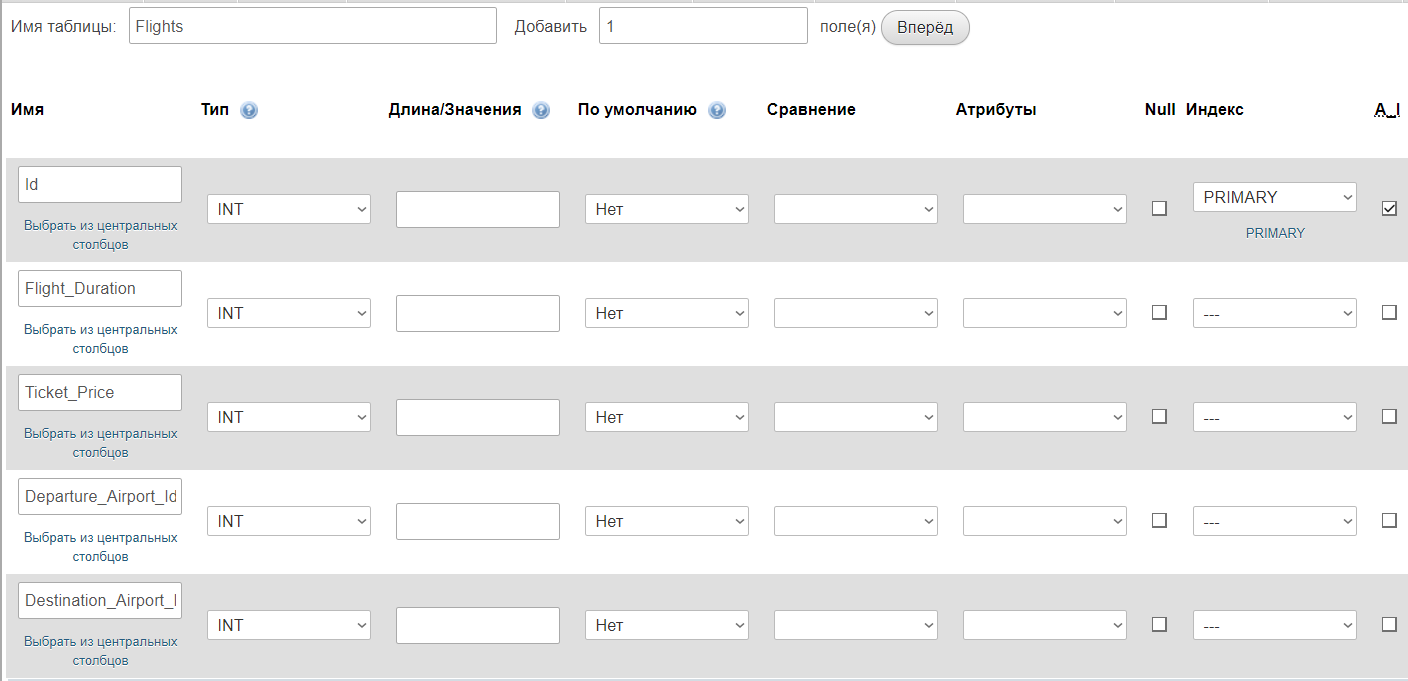


Рисунок 7. Создание таблицы Flights

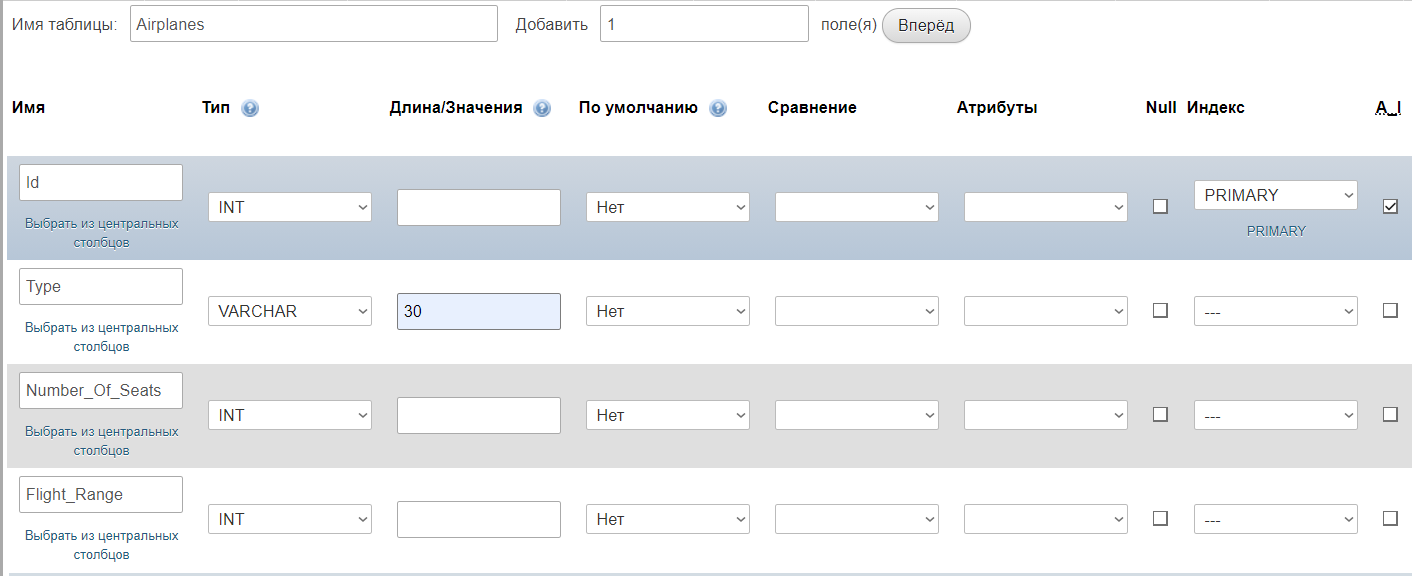


Рисунок 8. Создание таблицы Airplanes

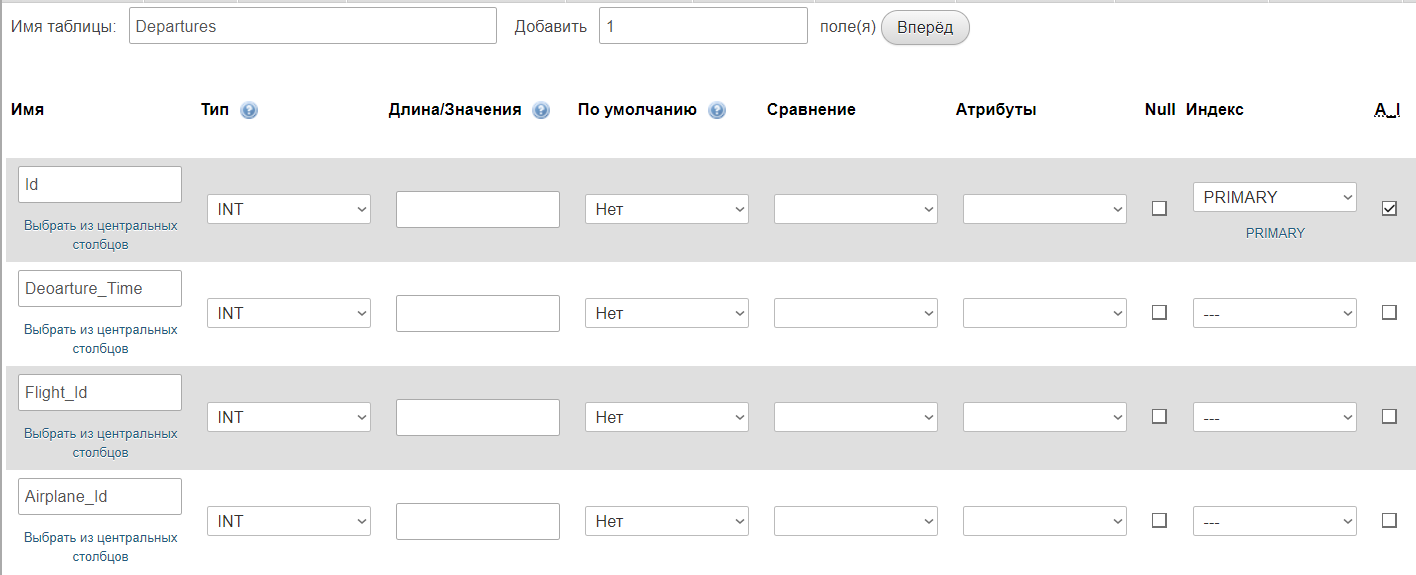


Рисунок 9. Создание таблицы Departures

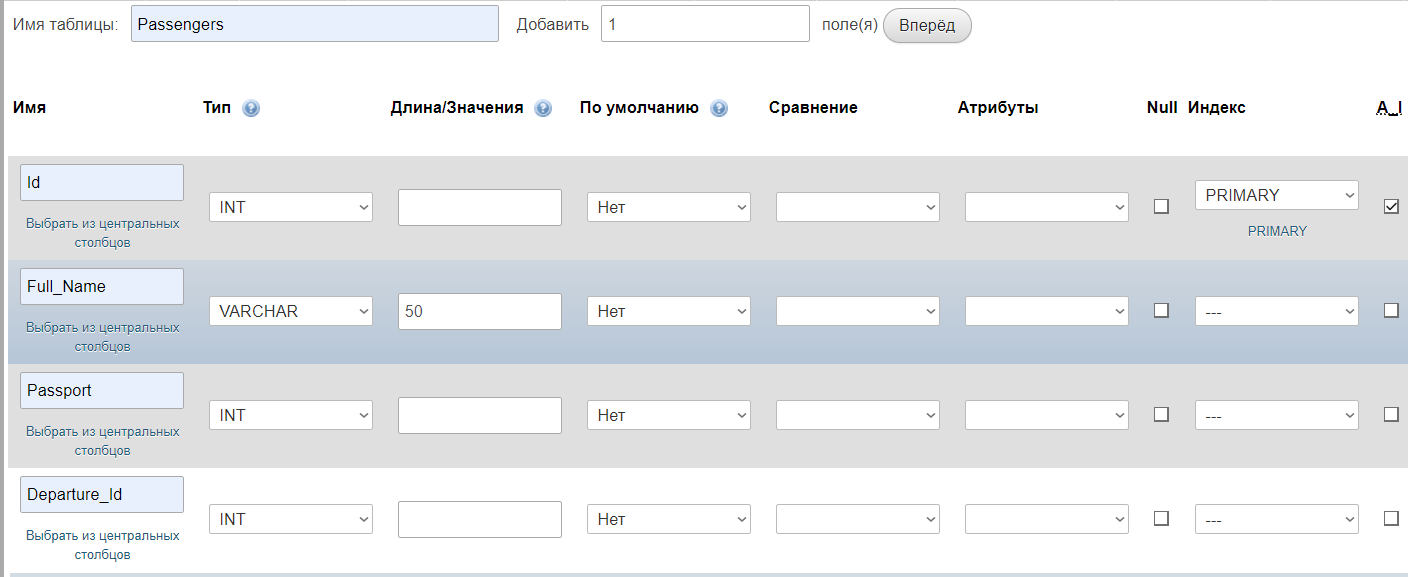


Рисунок 10. Создание таблицы Passengers

## 3.3 Создание связей

Для правильного взаимодействия таблиц в базе данных необходимо установить связи. Можно обойтись и без них, но в таком случае при удалении и обновлении некоторых записей придётся удалять и обновлять записи по отдельности в каждой таблице. Это усложнит процесс и повысит шансы допущения ошибок. Для создания связи необходимо перейти к таблице и выбрать раздел “Структура”. Далее нужно нажать на кнопку “Связи”, после чего заполнить нужные поля. Для создания внешних ключей в таблице выберем каскадное обновление/удаление, а затем соответствующее поле с первичным ключом в нужной таблице. На Рисунок 11 приведён пример создания связи между таблицами Аэропортов и Рейсов. Я проделал то же и с другими внешними ключами.

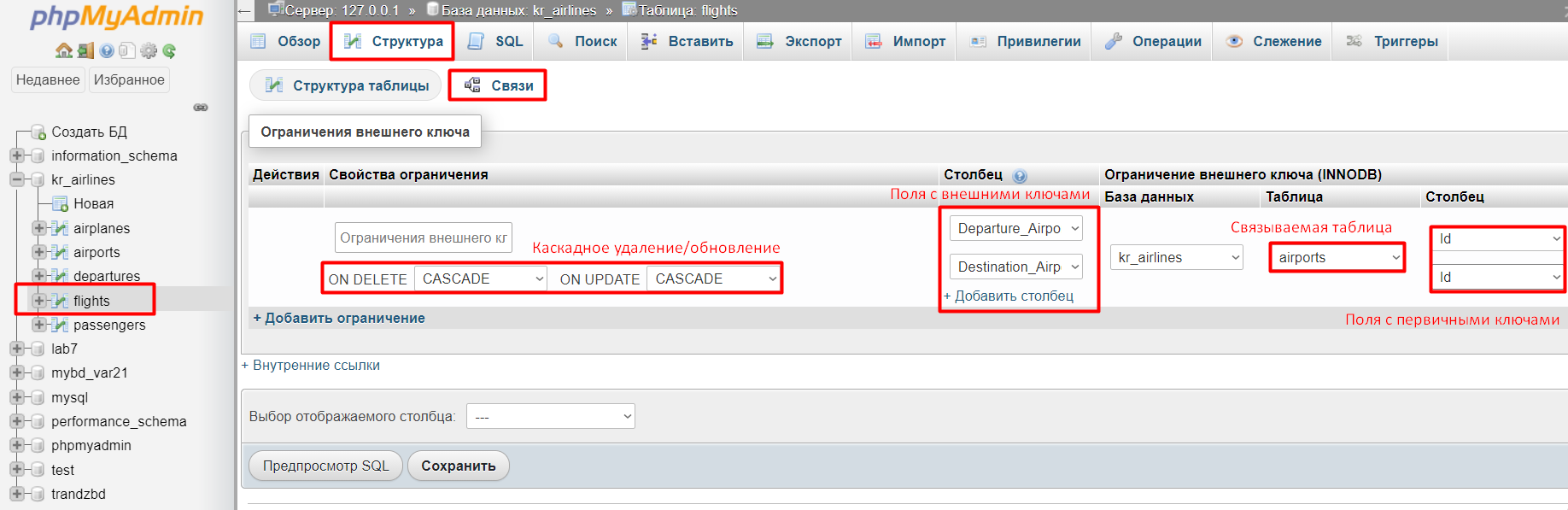


Рисунок 11. Создание связи

## 3.4 Заполнение таблиц

Для заполнения таблиц первоначальными данными можно применить два способа: используя интерфейс для работы с БД или используя SQL запрос. Я добавлю в каждую таблицу 15 записей, используя [генератор заполнения таблиц БД](https://www.mockaroo.com). В качестве примера добавим новую запись в таблицу Airports. Синтаксис запроса добавления новой записи можно посмотреть на Рисунок 12.

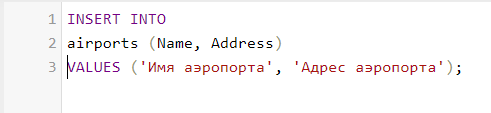


Рисунок 12. Добавление записи в таблицу

# Глава 4. Создание Web-форм

Используя PHP, HTML и CSS, я создал формы, образующие пользовательский интерфейс. Для начала я создал файл подключения к БД ([прил. A](#_Приложение_A)).

Далее написал файл главной страницы пользовательского интерфейса ([прил. B](#_Приложение_B)). Посмотреть его веб представление можно на **Рисунок 13**.

Из главной формы можно перейти к работе с любой таблицей. Для реализации основных операций я написал разные файлы для каждой таблицы.

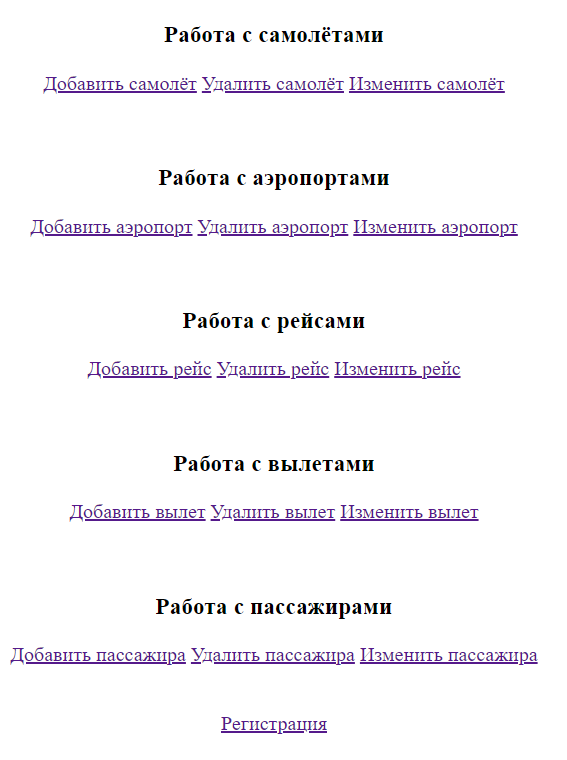


Рисунок 13. Форма - главное меню

## 4.1 Работа с самолётами

Перейдя по кнопке “Добавить самолёт”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно добавить новую запись в таблицу, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 14). Код данной формы можно просмотреть в [приложении C1](#_Приложение_С1).

Перейдя по кнопке “Удалить самолёт”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно удалить записи из таблицы по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 15). Код данной формы можно просмотреть в [приложении C2](#_Приложение_С2).

Перейдя по кнопке “Изменить самолёт”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно изменить записи в таблице по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 16). Код данной формы можно просмотреть в [приложении C3](#_Приложение_С3).

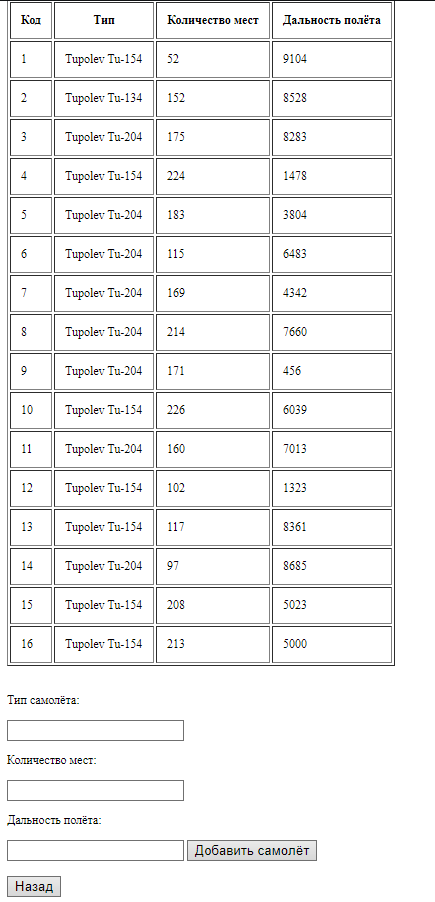


Рисунок 14. Форма - добавление самолёта

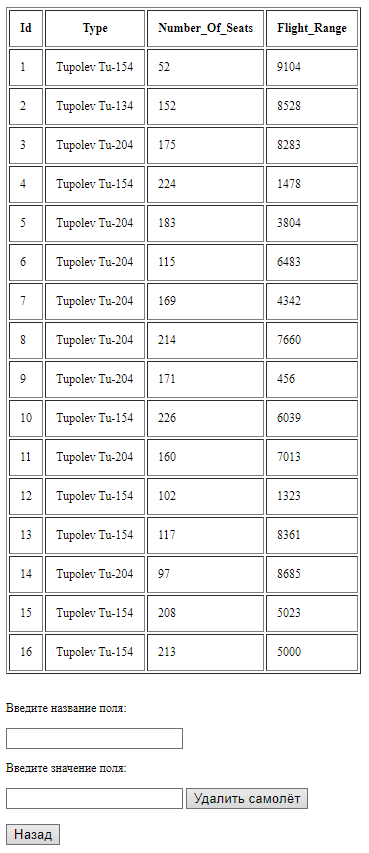


Рисунок 15. Форма - удаление самолёта

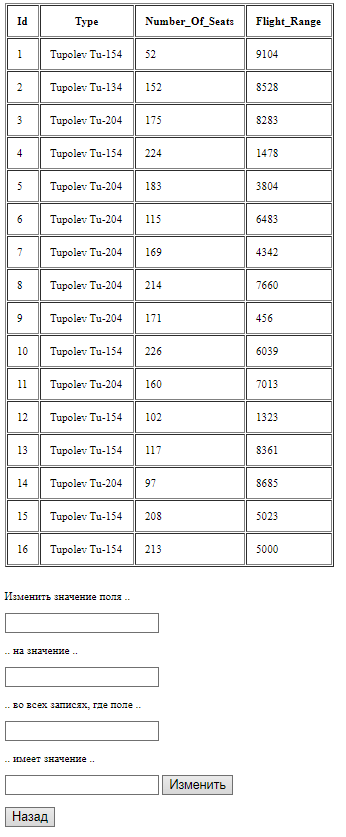


Рисунок 16. Форма - изменение самолёта

## 4.2 Работа с аэропортами

Перейдя по кнопке “Добавить аэропорт”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно добавить новую запись в таблицу, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 17). Код данной формы можно просмотреть в [приложении D1](#_Приложение_D1).

Перейдя по кнопке “Удалить аэропорт”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно удалить записи из таблицы по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 18). Код данной формы можно просмотреть в [приложении D2](#_Приложение_D2).

Перейдя по кнопке “Изменить аэропорт”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно изменить записи в таблице по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 19). Код данной формы можно просмотреть в [приложении D3](#_Приложение_D3).

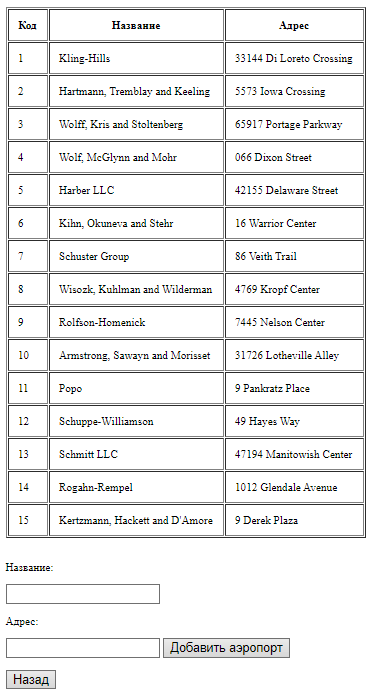


Рисунок 17. Форма - добавление аэропорта

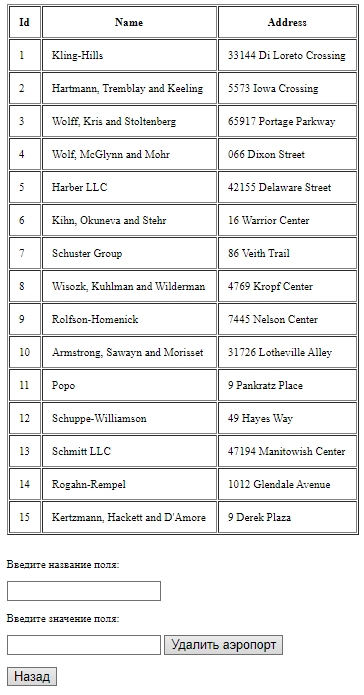


Рисунок 18. Форма - удаление аэропорта

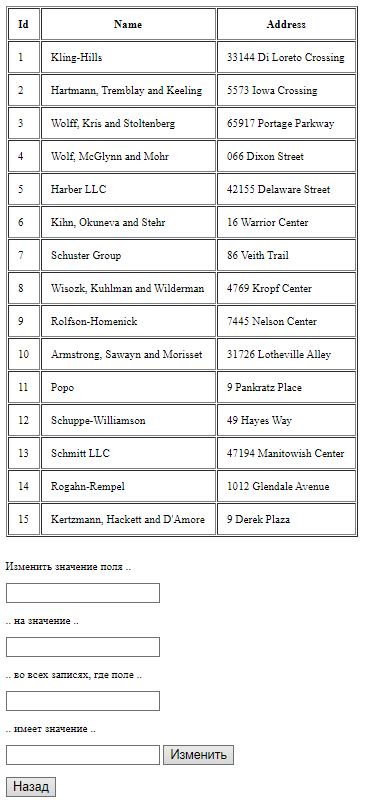


Рисунок 19. Форма - изменение аэропорта

## 4.3 Работа с рейсами

Перейдя по кнопке “Добавить рейс”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно добавить новую запись в таблицу, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 20). Код данной формы можно просмотреть в [приложении E1](#_Приложение_E1).

Перейдя по кнопке “Удалить рейс”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно удалить записи из таблицы по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 21). Код данной формы можно просмотреть в [приложении E2](#_Приложение_E2).

Перейдя по кнопке “Изменить рейс”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно изменить записи в таблице по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 22). Код данной формы можно просмотреть в [приложении E3](#_Приложение_E3).

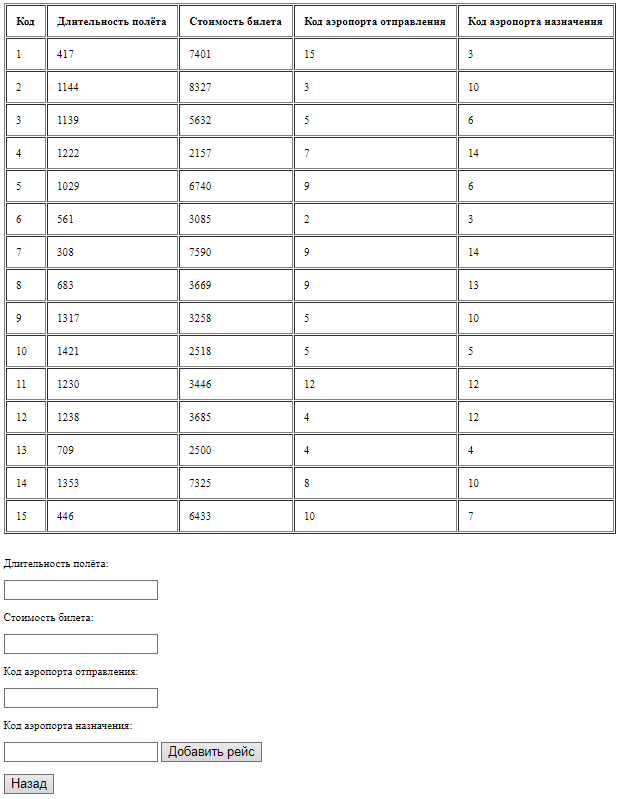


Рисунок 20. Форма - добавить рейс

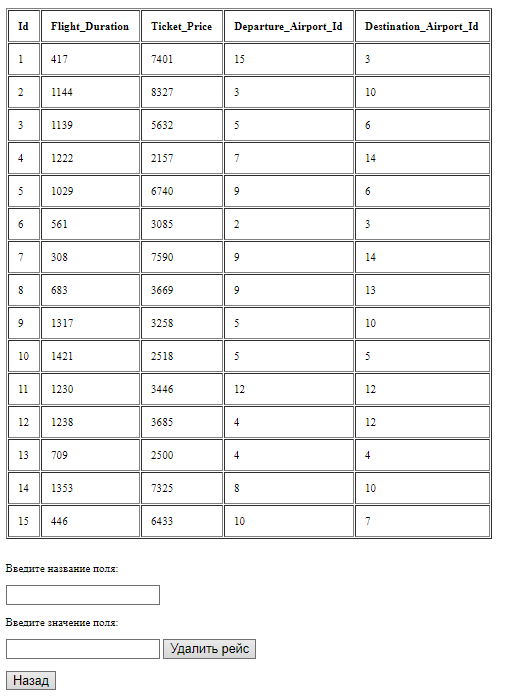


Рисунок 21. Форма - удалить рейс

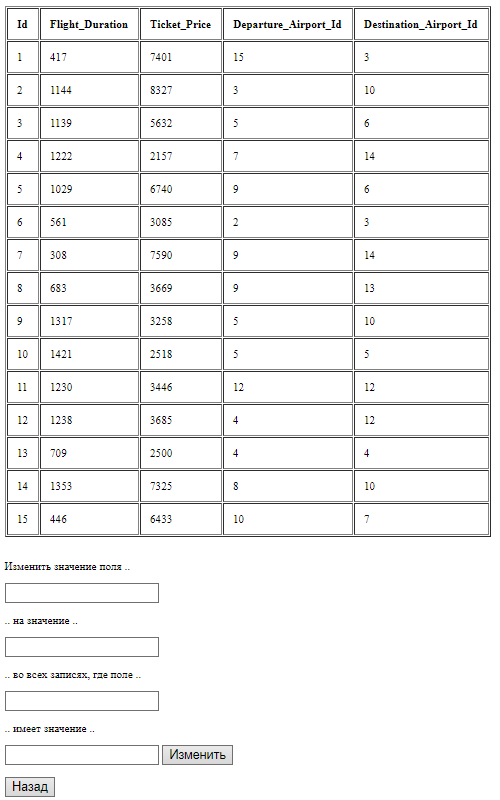


Рисунок 22. Форма - изменить рейс

## 4.4 Работа с вылетами

Перейдя по кнопке “Добавить вылет”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно добавить новую запись в таблицу, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 23). Код данной формы можно просмотреть в [приложении F1](#_Приложение_F1).

Перейдя по кнопке “Удалить вылет”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно удалить записи из таблицы по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 24). Код данной формы можно просмотреть в [приложении F2](#_Приложение_F2).

Перейдя по кнопке “Изменить вылет”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно изменить записи в таблице по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 25). Код данной формы можно просмотреть в [приложении F3](#_Приложение_F3).

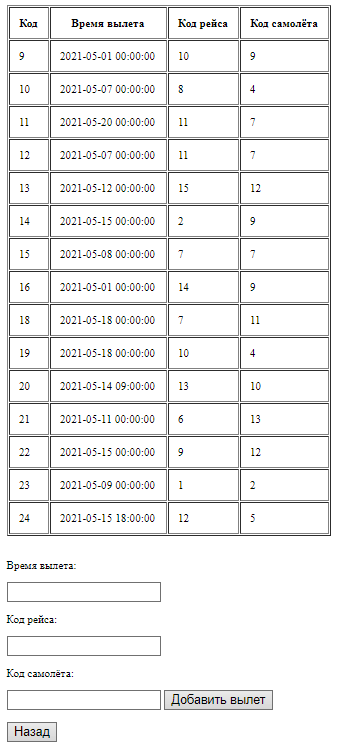


Рисунок 23. Форма - добавить вылет

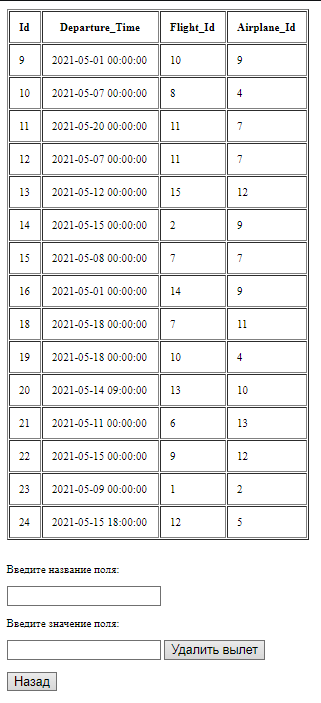


Рисунок 24. Форма - удалить вылет

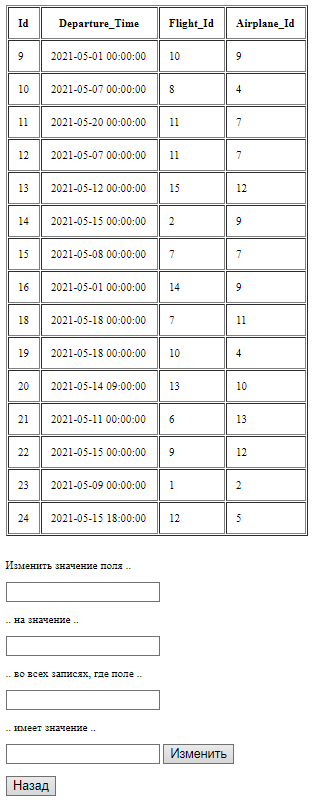


Рисунок 25. Форма - изменить вылет

## 4.5 Работа с пассажирами

Перейдя по кнопке “Добавить пассажира”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно добавить новую запись в таблицу, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 26). Код данной формы можно просмотреть в [приложении G1](#_Приложение_G1).

Перейдя по кнопке “Удалить пассажира”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно удалить записи из таблицы по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 27). Код данной формы можно просмотреть в [приложении G2](#_Приложение_G2).

Перейдя по кнопке “Изменить пассажира”, можно увидеть новую форму, в которой представлена таблица из БД. Ниже можно изменить записи в таблице по заданному полю, заполнив все необходимые поля и нажав на соответствующую кнопку (Рисунок 28). Код данной формы можно просмотреть в [приложении G3](#_Приложение_G3).

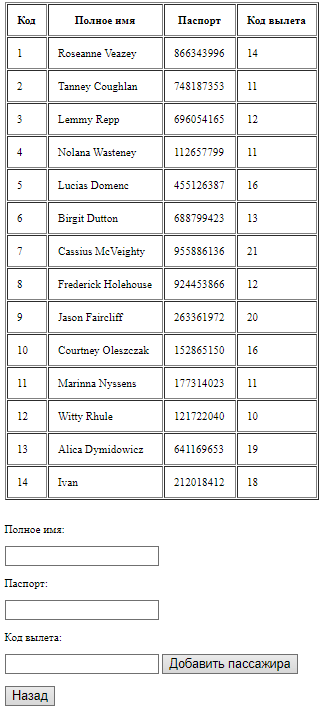


Рисунок 26. Форма - добавить пассажира

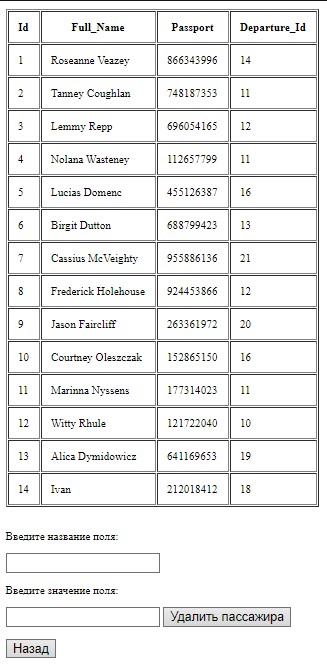


Рисунок 27. Форма - удалить пассажира



Рисунок 28. Форма - изменить пассажира

## 4.6 Регистрация

Также есть возможность регистрации пользователя. Перейдя по кнопке “Регистрация”, можно увидеть новую форму. В ней нужно ввести логин и пароль, после чего нажать на кнопку “Зарегистрироваться” (Рисунок 29). Код данной формы можно просмотреть в [приложении H](#_Приложение_H).

Для реализации этой функции я создал дополнительную таблицу users.

Для создания этой таблицы заполнил три поля: Id типа данных INT с атрибутом PRIMARY KEY и AUTO\_INCREMENT; Login типа данных VARCHAR с количеством символов – 50; Password типа данных VARCHAR (Рисунок 30).

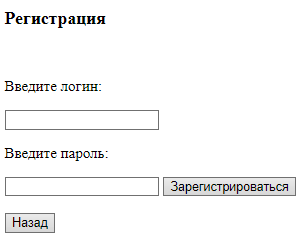


Рисунок 29. Форма - регистрация

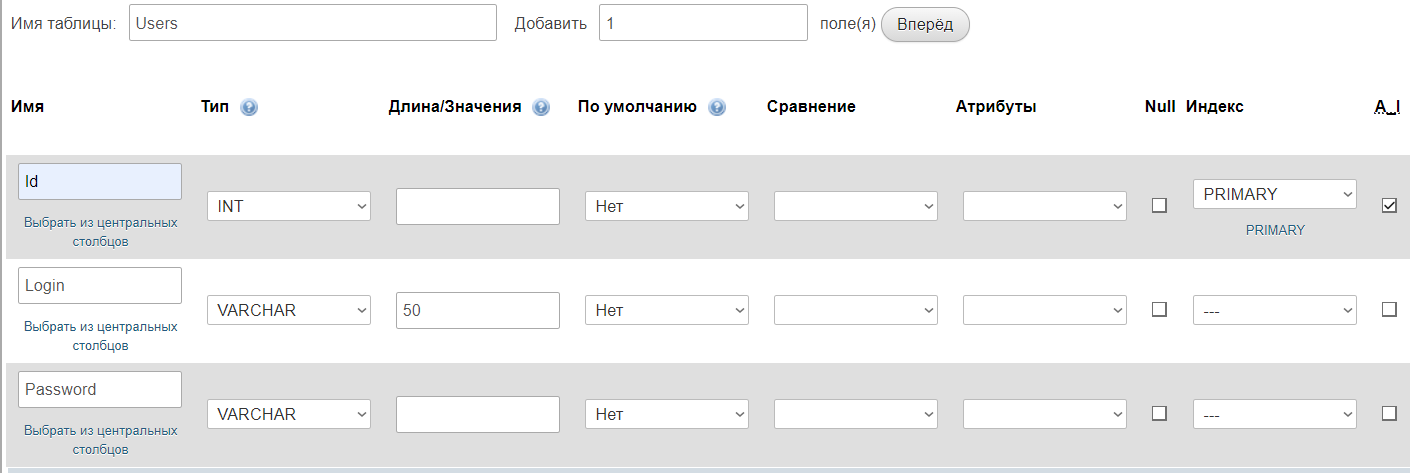


Рисунок 30. Создание таблицы users

# Глава 5. Простейшие криптографические способы защиты базы данных

В этой главе речь пойдет о способах защиты моей базы данных.

Я хешировал пароли с помощью md5 (см. [приложение H](#_Приложение_H)). MD5 является одним из алгоритмов хеширования на 128-битной основе. Под хешированием понимают преобразование входных данных по определенному алгоритму в битовую строку определенной длины. При этом полученный в ходе вычислений результат представлен в шестнадцатеричной системе исчисления. Она называется хешем, хеш-суммой или хеш-кодом. Процесс хеширования широко применяется в программировании и веб-индустрии. В основном для создания уникальных значений в ассоциативных массивах, идентификаторов.

Криптографические средства защиты информации. Одним из наиболее действенных способов защиты персональных данных является использование средств криптографии. Если упростить, то речь идет о шифровании текста с помощью цифрового кода. К криптографическим средствам относятся аппаратные, программные и комбинированные устройства и комплексы, способные реализовывать алгоритмы криптографического преобразования информации.

Создать хеш-код в php можно с помощью нескольких функций:

- md5() – в качестве одного из параметров принимает значение «соли»;

- crypt() – в отличие от предыдущей эта функция полностью автоматизирует весь процесс, в том числе и генерирование значения соли.

В своей курсовой работе я использовал первую функцию.

Алгоритм хеширования MD5 используется при заполнении данных в таблицу «users». На Рисунок 31 в таблице «users» хранится захешированный пароль, который мы получили из формы добавления клиента.

В своей курсовой работе я также использовал триггеры. Триггер – это хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено наступлением определенного события (действия) – по сути добавлением или удалением строки в заданной таблице, или модификацией данных в определенном столбце заданной таблицы реляционной БД. Триггеры применяются для обеспечения целостности данных и реализации сложной бизнес-логики. Триггер запускается сервером автоматически при попытке изменения данных в таблице, с которой он связан.

Триггер хранится вместе с таблицами в бинарном виде – однажды созданный, он не может быть изменен, возможно только удалить его и записать новый. Поддержка для триггеров включена в СУБД MySQL, начиная с версии MySQL 5.0.2.

Триггер активируется, когда происходит определенное событие для конкретной таблицы. Таких события три:

1. INSERT

2. DELETE

3. UPDATE

Каждое из трех поддерживаемых триггерами событий, указанных выше, условно подразделяются еще на два:

- начало события BEFORE;

- конец события AFTER.

Поэтому можно сказать имеется 6 событий для конкретной таблицы:

1. BEFORE INSERT.

2. AFTER INSERT.

3. BEFORE UPDATE.

4. AFTER UPDATE.

5. BEFORE DELETE.

6. AFTER DELETE.

Например, событие BEFORE INSERT произойдет перед вставкой новой строки в таблицу, а событие AFTER INSERT – после. Важно отметить, что событие AFTER INSERT может и не наступить, если во время выполнения запроса произошла ошибка.

На каждое из перечисленных шести событий для одной конкретной таблицы можно выполнить только один триггер.

Для применения триггера я создал ещё одну таблицу.

Для создания таблицы my\_log заполнил три поля: Id типа данных INT с атрибутом PRIMARY KEY и AUTO\_INCREMENT; User типа данных VARCHAR с количеством символов – 30; Query типа данных VARCHAR с количеством символов – 30; timestamp типа данных timestamp (Рисунок 32).

На Рисунок 33 показаны все имеющиеся триггеры в моей курсовой работе.

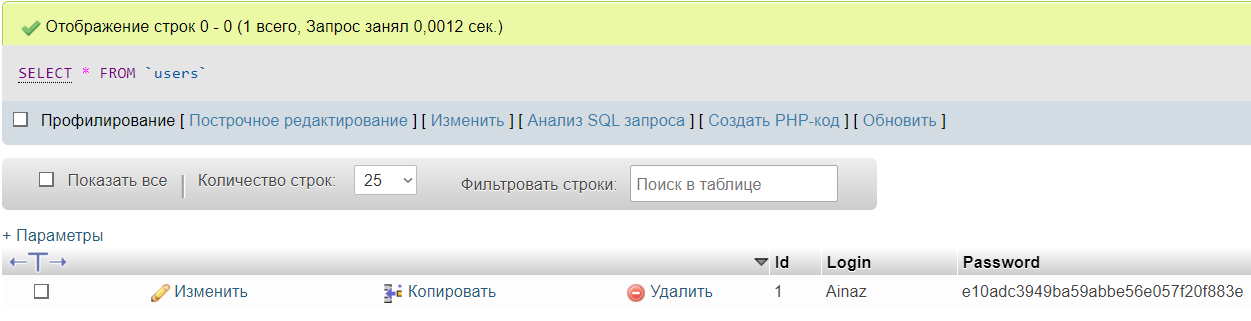


Рисунок 31. Содержимое таблицы users

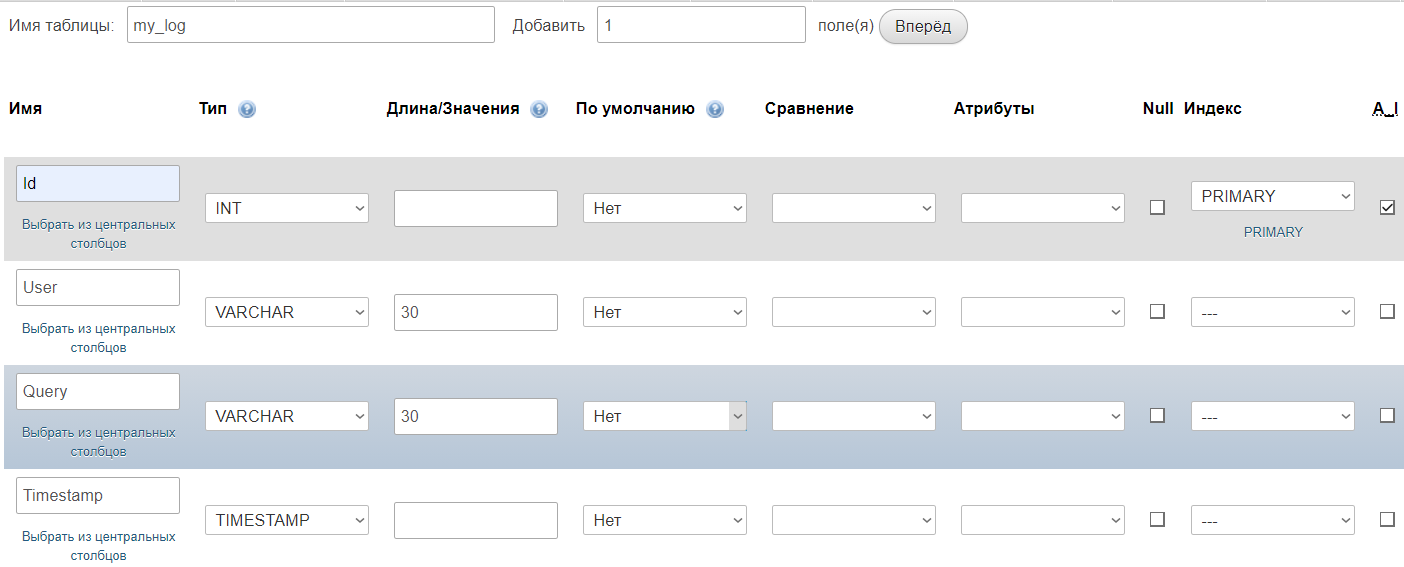


Рисунок 32. Создание таблицы my\_log

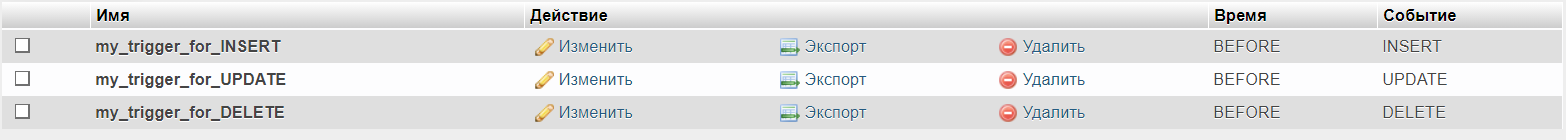


Рисунок 33. Триггеры таблицы departures

# ****Заключение****

В данной курсовой работе я:

* создал базу данных по теме – «Авиалинии»
* научился работать с помощью приложения XAMPP и понял основные принципы работы с ним
* понял, что будет представлять из себя моя база данных
* создал необходимые таблицы для курсовой работы
* разработал главное меню курсовой работы, формы для добавления, удаления и обновления данных для таблиц
* добавил возможность регистрации новых пользователей
* реализовал шифрование паролей, которые хранятся в таблице «users», в базе данных с помощью нового для меня алгоритма шифрования md5
* создал триггеры на всевозможные действия с данными (добавление, удаление, обновление), которые записывают данные в таблицу «my\_log» при работе с таблице “departures”

В итоге, завершив курсовую работу, я получил огромный опыт разработки, обслуживания и защиты базы данных с пользовательским интерфейсом.

# ****Список использованных источников****

1. Руководство пользователя XAMPP: <https://ipipe.ru/info/ustanovka-xampp>
2. Понятие ER-модели: <https://habr.com/ru/post/50312/>
3. Руководство по проектированию БД: <https://habr.com/ru/post/193136/>
4. Основные понятия БД: <https://ru.wikipedia.org/wiki/База_данных>
5. Основы CSS: <https://html5css.ru/html/html_css.php>
6. Защита БД: <https://searchinform.ru/services/outsource-ib/zaschita-informatsii/bezopasnost-baz-dannykh/zaschita-baz-dannykh/vidy-zaschity-baz-dannykh/>
7. Хэширование паролей: <https://www.php.net/manual/ru/faq.passwords.php>
8. Триггеры БД: <https://zametkinapolyah.ru/zametki-o-mysql/tema-13-triggery-v-sql-na-primere-bazy-dannyx-sqlite.html>

# ****Приложения****

## Приложение A

<?php

$connect = mysqli\_connect("localhost","root","", "kr\_airlines");

if(!$connect){

die("ErrorPIPI");

}

?>

## Приложение B

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Авиаперелёты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 16px;

text-align: center;

}

</style>

<body>

<h3 class="block-header">Работа с самолётами</h2>

<div class="airplanes">

<a class="navigation" href="addAirplane.php">Добавить самолёт</a>

<a class="navigation" href="deleteAirplane.php">Удалить самолёт</a>

<a class="navigation" href="updateAirplane.php">Изменить самолёт</a><br>

</div>

<br>

<br>

<h3 class="block-header">Работа с аэропортами</h2>

<div class="airports">

<a class="navigation" href="addAirport.php">Добавить аэропорт</a>

<a class="navigation" href="deleteAirport.php">Удалить аэропорт</a>

<a class="navigation" href="updateAirport.php">Изменить аэропорт</a><br>

</div>

<br>

<br>

<h3 class="block-header">Работа с рейсами</h2>

<div class="flights">

<a class="navigation" href="addFlight.php">Добавить рейс</a>

<a class="navigation" href="deleteFlight.php">Удалить рейс</a>

<a class="navigation" href="updateFlight.php">Изменить рейс</a><br>

</div>

<br>

<br>

<h3 class="block-header">Работа с вылетами</h2>

<div class="departures">

<a class="navigation" href="addDeparture.php">Добавить вылет</a>

<a class="navigation" href="deleteDeparture.php">Удалить вылет</a>

<a class="navigation" href="updateDeparture.php">Изменить вылет</a><br>

</div>

<br>

<br>

<h3 class="block-header">Работа с пассажирами</h2>

<div class="passengers">

<a class="navigation" href="addPassenger.php">Добавить пассажира</a>

<a class="navigation" href="deletePassenger.php">Удалить пассажира</a>

<a class="navigation" href="updatePassenger.php">Изменить пассажира</a><br>

</div>

<br>

<br>

<div class="users">

<a class="navigation" href="registration.php">Регистрация</a><br>

</div>

</body>

</html>

## Приложение С1

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Самолёты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Код</th>

<th>Тип</th>

<th>Количество мест</th>

<th>Дальность полёта</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `airplanes`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="addAirplane2.php" method="post">

<p>Тип самолёта: </p>

<input type="text" name="type">

<p>Количество мест: </p>

<input type="number" name="nos">

<p>Дальность полёта: </p>

<input type="number" name="fr">

<button type="submit">Добавить самолёт</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

<?php

require\_once 'connection.php';

$type = $\_POST['type'];

$nos = $\_POST['nos'];

$fr = $\_POST['fr'];

mysqli\_query($connect, "INSERT INTO `airplanes` (`type`, `Number\_Of\_Seats`, `Flight\_Range`) VALUES ('$type', '$nos', '$fr')")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="addAirplane.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение С2

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Самолёты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Type</th>

<th>Number\_Of\_Seats</th>

<th>Flight\_Range</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `airplanes`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="deleteAirplane2.php" method="post">

<p>Введите название поля: </p>

<input type="text" name="field">

<p>Введите значение поля: </p>

<input type="text" name="newMean">

<button type="submit">Удалить самолёт</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field = $\_POST['field'];

$newMean = $\_POST['newMean'];

mysqli\_query($connect, "DELETE FROM `airplanes` WHERE $field = '$newMean'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="deleteAirplane.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение С3

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Самолёты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Type</th>

<th>Number\_Of\_Seats</th>

<th>Flight\_Range</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `airplanes`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="updateAirplane2.php" method="post">

<p>Изменить значение поля ..</p>

<input type="text" name="field1">

<p>.. на значение ..</p>

<input type="text" name="param1">

<p>.. во всех записях, где поле ..</p>

<input type="text" name="field2">

<p>.. имеет значение ..</p>

<input type="text" name="param2">

<button type="submit">Изменить</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field1 = $\_POST['field1'];

$param1 = $\_POST['param1'];

$field2 = $\_POST['field2'];

$param2 = $\_POST['param2'];

mysqli\_query($connect, "UPDATE `airplanes` SET $field1='$param1' WHERE $field2='$param2'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="updateAirplane.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение D1

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Аэропорты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Код</th>

<th>Название</th>

<th>Адрес</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `airports`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="addAirport2.php" method="post">

<p>Название: </p>

<input type="text" name="name">

<p>Адрес: </p>

<input type="text" name="address">

<button type="submit">Добавить аэропорт</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

<?php

require\_once 'connection.php';

$name = $\_POST['name'];

$address = $\_POST['address'];

mysqli\_query($connect, "INSERT INTO `airports` (`name`, `address`) VALUES ('$name', '$address')")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="addAirport.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение D2

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Аэропорты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Name</th>

<th>Address</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `airports`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="deleteAirport2.php" method="post">

<p>Введите название поля: </p>

<input type="text" name="field">

<p>Введите значение поля: </p>

<input type="text" name="newMean">

<button type="submit">Удалить аэропорт</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field = $\_POST['field'];

$newMean = $\_POST['newMean'];

mysqli\_query($connect, "DELETE FROM `airports` WHERE $field = '$newMean'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="deleteAirport.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение D3

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Аэропорты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Name</th>

<th>Address</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `airports`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="updateAirport2.php" method="post">

<p>Изменить значение поля ..</p>

<input type="text" name="field1">

<p>.. на значение ..</p>

<input type="text" name="param1">

<p>.. во всех записях, где поле ..</p>

<input type="text" name="field2">

<p>.. имеет значение ..</p>

<input type="text" name="param2">

<button type="submit">Изменить</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field1 = $\_POST['field1'];

$param1 = $\_POST['param1'];

$field2 = $\_POST['field2'];

$param2 = $\_POST['param2'];

mysqli\_query($connect, "UPDATE `airports` SET $field1='$param1' WHERE $field2='$param2'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="updateAirport.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение E1

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рейсы</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Код</th>

<th>Длительность полёта</th>

<th>Стоимость билета</th>

<th>Код аэропорта отправления</th>

<th>Код аэропорта назначения</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `flights`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

<td><?=$key[4]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="addFlight2.php" method="post">

<p>Длительность полёта: </p>

<input type="number" name="fd">

<p>Стоимость билета: </p>

<input type="number" name="tp">

<p>Код аэропорта отправления: </p>

<input type="number" name="depai">

<p>Код аэропорта назначения: </p>

<input type="number" name="desai">

<button type="submit">Добавить рейс</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

<?php

require\_once 'connection.php';

$flightD = $\_POST['fd'];

$ticketP = $\_POST['tp'];

$depAI = $\_POST['depai'];

$desAI = $\_POST['desai'];

mysqli\_query($connect, "INSERT INTO `flights`(`Flight\_Duration`, `Ticket\_Price`, `Departure\_Airport\_Id`, `Destination\_Airport\_Id`) VALUES ('$flightD','$ticketP','$depAI','$desAI')")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="addFlight.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение E2

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рейсы</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Flight\_Duration</th>

<th>Ticket\_Price</th>

<th>Departure\_Airport\_Id</th>

<th>Destination\_Airport\_Id</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `flights`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

<td><?=$key[4]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="deleteFlight2.php" method="post">

<p>Введите название поля: </p>

<input type="text" name="field">

<p>Введите значение поля: </p>

<input type="text" name="newMean">

<button type="submit">Удалить рейс</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field = $\_POST['field'];

$newMean = $\_POST['newMean'];

mysqli\_query($connect, "DELETE FROM `flights` WHERE $field = '$newMean'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="deleteAirplane.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение E3

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рейсы</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Flight\_Duration</th>

<th>Ticket\_Price</th>

<th>Departure\_Airport\_Id</th>

<th>Destination\_Airport\_Id</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `flights`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

<td><?=$key[4]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="updateFlight2.php" method="post">

<p>Изменить значение поля ..</p>

<input type="text" name="field1">

<p>.. на значение ..</p>

<input type="text" name="param1">

<p>.. во всех записях, где поле ..</p>

<input type="text" name="field2">

<p>.. имеет значение ..</p>

<input type="text" name="param2">

<button type="submit">Изменить</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field1 = $\_POST['field1'];

$param1 = $\_POST['param1'];

$field2 = $\_POST['field2'];

$param2 = $\_POST['param2'];

mysqli\_query($connect, "UPDATE `flights` SET $field1='$param1' WHERE $field2='$param2'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="updateFlight.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение F1

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Вылеты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Код</th>

<th>Время вылета</th>

<th>Код рейса</th>

<th>Код самолёта</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `departures`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="addDeparture2.php" method="post">

<p>Время вылета: </p>

<input type="text" name="time">

<p>Код рейса: </p>

<input type="number" name="fi">

<p>Код самолёта: </p>

<input type="number" name="ai">

<button type="submit">Добавить вылет</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

<?php

require\_once 'connection.php';

$time = $\_POST['time'];

$fi = $\_POST['fi'];

$ai = $\_POST['ai'];

mysqli\_query($connect, "INSERT INTO `departures` (`Departure\_Time`, `Flight\_Id`, `Airplane\_Id`) VALUES ('$time', '$fi', '$ai')")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="addDeparture.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение F2

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Вылеты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Departure\_Time</th>

<th>Flight\_Id</th>

<th>Airplane\_Id</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `departures`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="deleteDeparture2.php" method="post">

<p>Введите название поля: </p>

<input type="text" name="field">

<p>Введите значение поля: </p>

<input type="text" name="newMean">

<button type="submit">Удалить вылет</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field = $\_POST['field'];

$newMean = $\_POST['newMean'];

mysqli\_query($connect, "DELETE FROM `departures` WHERE $field = '$newMean'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="deletePassenger.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение F3

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Вылеты</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Departure\_Time</th>

<th>Flight\_Id</th>

<th>Airplane\_Id</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `departures`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="updateDeparture2.php" method="post">

<p>Изменить значение поля ..</p>

<input type="text" name="field1">

<p>.. на значение ..</p>

<input type="text" name="param1">

<p>.. во всех записях, где поле ..</p>

<input type="text" name="field2">

<p>.. имеет значение ..</p>

<input type="text" name="param2">

<button type="submit">Изменить</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field1 = $\_POST['field1'];

$param1 = $\_POST['param1'];

$field2 = $\_POST['field2'];

$param2 = $\_POST['param2'];

mysqli\_query($connect, "UPDATE `departures` SET $field1='$param1' WHERE $field2='$param2'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="updateDeparture.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение G1

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Пассажиры</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Код</th>

<th>Полное имя</th>

<th>Паспорт</th>

<th>Код вылета</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `passengers`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="addPassenger2.php" method="post">

<p>Полное имя: </p>

<input type="text" name="name">

<p>Паспорт: </p>

<input type="number" name="pass">

<p>Код вылета: </p>

<input type="number" name="di">

<button type="submit">Добавить пассажира</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

<?php

require\_once 'connection.php';

$name = $\_POST['name'];

$pass = $\_POST['pass'];

$di = $\_POST['di'];

mysqli\_query($connect, "INSERT INTO `passengers` (`Full\_Name`, `Passport`, `Departure\_Id`) VALUES ('$name', '$pass', '$di')")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="addPassenger.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение G2

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Пассажиры</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Full\_Name</th>

<th>Passport</th>

<th>Departure\_Id</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `passengers`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="deletePassenger2.php" method="post">

<p>Введите название поля: </p>

<input type="text" name="field">

<p>Введите значение поля: </p>

<input type="text" name="newMean">

<button type="submit">Удалить пассажира</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field = $\_POST['field'];

$newMean = $\_POST['newMean'];

mysqli\_query($connect, "DELETE FROM `passengers` WHERE $field = '$newMean'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="deletePassenger.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение G3

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Пассажиры</title>

</head>

<style>

th,td{

padding: 10px;

}

body{

font-size: 12px;

}

</style>

<body>

<table border="1">

<tr>

<th>Id</th>

<th>Full\_Name</th>

<th>Passport</th>

<th>Departure\_Id</th>

</tr>

<?php

$worker = mysqli\_query($connect, query:"SELECT \* FROM `passengers`");

$worker = mysqli\_fetch\_all($worker);

foreach ($worker as $key) {

?>

<tr>

<td><?=$key[0]?></td>

<td><?=$key[1]?></td>

<td><?=$key[2]?></td>

<td><?=$key[3]?></td>

</tr>

<?php

}

?>

</table>

<br>

<form action="updatePassenger2.php" method="post">

<p>Изменить значение поля ..</p>

<input type="text" name="field1">

<p>.. на значение ..</p>

<input type="text" name="param1">

<p>.. во всех записях, где поле ..</p>

<input type="text" name="field2">

<p>.. имеет значение ..</p>

<input type="text" name="param2">

<button type="submit">Изменить</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</html>

<?php

require\_once 'connection.php';

$field1 = $\_POST['field1'];

$param1 = $\_POST['param1'];

$field2 = $\_POST['field2'];

$param2 = $\_POST['param2'];

mysqli\_query($connect, "UPDATE `passengers` SET $field1='$param1' WHERE $field2='$param2'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="updatePassenger.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

## Приложение H

<?php

require\_once 'connection.php';

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Регистрация</title>

</head>

<style>

th,td{

padding: 10px;

}

</style>

<body>

<h3 class="block-header">Регистрация</h2>

<br>

<form action="registration2.php" method="post">

<p>Введите логин: </p>

<input type="text" name="login">

<p>Введите пароль: </p>

<input type="text" name="password">

<button type="submit">Зарегистрироваться</button>

</form>

<br>

<form action="mainMenu.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>

<?php

require\_once 'connection.php';

$login = $\_POST['login'];

$password = $\_POST['password'];

$password = md5($password);

mysqli\_query($connect, "INSERT INTO `users` (`Login`, `Password`) VALUES ('$login', '$password'")

?>

<!doctype html>

<html Lang = "en">

<head>

<meta charset="utf-8">

<title>Рабочая форма</title>

</head>

<body>

<form action="registration.php" method="post">

<button type="submit">Назад</button>

</form>

</body>

</html>