Севастопольский национальный технический университет

Кафедра

Информационных систем

Курсовой проект по дисциплине

«Организация баз данных и знаний»

на тему

«Web-сайт турагентства»

Пояснительная записка

Проверил

\_\_\_\_\_\_\_\_\_ Тимофеева О.Л.

« \_\_\_» \_\_\_\_\_\_\_\_\_2014

Разработали

Студенты группы И–31д

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Глобин А.В.

\_\_\_\_\_\_\_\_\_\_\_\_Зосименко К.В.

« \_\_\_» \_\_\_\_\_\_\_\_\_2014

Севастополь

2014

СОДЕРЖАНИЕ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

Курсовой проект

Разраб.

Глобин А.В.

Зосименко К.В.

Провер.

Тимофеева О.Л

Н. Контр.

Утверд.

Пояснительная записка

Лит.

Листов

47

Кафедра ИС,

группа И-31д

Введение…………………………………………………………………...4

1. Постановка задачи……………………………………………………..5
2. Анализ предметной области…………………………………………..7
   1. Анализ отношений между объектами…………………………7
   2. Разработка ER-модели…………………………………………..9
3. Разработка логической схемы базы данных ………………………..10

3.1 Обоснование выбора реляционной модели…………….……..10

3.2 Построение концептуальной модели…………………..……..12

3.3 Нормализация исходной модели………………………...……13

3.3.1 Первая нормальная форма……………………………14

3.3.2..Вторая нормальная форма……………………………15

3.3.3..Третья нормальная форма……………...…………….18

1. Реализация базы данных …………………………………………....21
   1. Обоснование выбора языка СУБД……………………………21
   2. Разграничение прав доступа…………………………………...21
   3. Физическая реализация базы данных…………………………22
   4. Тестирование базы данных ……………………………………24
2. Исследование информационных параметров базы данных………...26
3. Описание интерфейса пользователя………………………………….28

Заключение………………………………………………………………..32

Библиографический список………………………………………………33

Приложение A……………………………………………………………..34

Введение

Концепция организации информации в виде баз данных является значительным фактором при создании различных систем автоматизированной обработки информации. Проектирование подобных программно-технических компонентов информационных систем является комплексной задачей, включающей широкий спектр вопросов, начиная от адекватного моделирования предметной области, до выбора необходимых технических и программных средств, написания эргономических интерфейсов и т.д.

В данной курсовой работе предлагается спроектировать базу данных, начиная от ее логического проектирования на бумаге, и до момента создания физической модели базы данных, которая сможет полностью реализовать принципы работы логической модели. Для достижения поставленной цели потребуется применить знания, полученные в области программирования, проектирования трансляторов и администрирования баз данных.

1. Постановка задачи

В данном курсовом проекте необходимо спроектировать и реализовать реляционную базу данных под web-сайт туристической фирмы. Web-интерфейс позволяет найти подходящий тур для пользователя и оставить информацию администратору о клиенте, забронировавший путёвку.

Непосредственно разработка базы данных подразумевает выполнение следующих этапов:

* Анализ предметной области (анализ отношений между объектами, разработка ER-модели);
* Разработка логической схемы базы данных (построение и нормализация реляционной модели);
* Реализация базы данных (обоснование выбора СУБД, разработка физической схемы базы данных, реализация и тестирование базы данных, запросов к ней, разграничение прав доступа);
* Исследование информационных параметров базы данных;
* Разработка клиентского приложения (обоснование выбора языка программирования, разработка клиентского приложения, описание интерфейса пользователя, тестирование системы).

Для корректной работы программы были введены информационные и системные ограничения.

Минимальными системными требованиями являются: Windows 2000 (с установленным ServicePack 1); NT 4.0 (с установленным ServicePack 6a или выше);ME;Windows 98; RedHat 6.2, 7; Mandrake 7.2; SuSE 7.0; TurboLinux 6.0;Solaris 2.6, 7;32 Мб оперативной памяти;10 Мб дискового пространства.

Выделим основные объекты предметной области, информация о которых должна храниться в базе данных:

а) Туры

б) Города

в) Страны

г) Отели

д) Рейсы

е) Заявки клиентов



Рисунок 1.1 — Связи между объектами

Web-интерфейс предполагает две группы пользователей: администратор и клиент. Администратор должен иметь возможность добавлять путёвки и просматривать заявки клиентов. Клиент может находить необходимый ему тур (по заданным критериям) и бронировать его.

2. Анализ предметной области

Исходя из постановки задачи, выделим сущности.Сущность – любой различимый объект, информацию о котором необходимо хранить в базе данных. Поэтому для нашей задачи можно определить следующие сущности, которые в дальнейшем будут использоваться для построения внешней (инфологической) модели проектируемой базы данных:

* Заявка;
* Тур;
* Отель;
* Рейс;
* Город.

Сущность тур характеризуется ценой, отелем, рейсами на отдых и с отдыха, количеством свободных путёвок, продолжительностью, количеством персон и определяется номером тура. Заявка характеризуется путёвкой, фамилией клиента, его телефоном и e-mail. Рейс имеет атрибуты: дата вылета, дата прилёта, город вылета, город прилёта. Город характеризуется названием города и страной; страна — названием страны. Отель имеет атрибуты: количество звёзд, название отеля, городом.

2.1 Анализ отношений между объектами

Сложность проектирования базы данных состоит в том, что присутствуют связи между элементами данных. Поэтому необходимо формирование структуры данных и ее описание.

Так как основными конструктивными элементами инфологических моделей являются сущности и связи между ними, а строят ее по аналогии с естественным языком, то можно описать внешнюю модель базы данных следующим образом:



Рисунок 2.1.1 – Внешняя модель

2.2 Построение ER- диаграммы

На основании выделенных сущностей определим для них свойства и отношения между ними, построив таким образом ER-диаграмму (entity-relationship diagram).

ER-диаграмма отображает внешнюю модель базы данных.



Рисунок 2.2.1 — ER-диаграмма базы данных

3. Разработка логической схемы базы данных

3.1 Обоснование выбора реляционной модели

Базам данных постоянно грозит опасность стать громоздкими и слишком сложными системами. Представление данных в виде деревьев, сетевых и списковых структур в общем случае препятствует многим изменениям данных, необходимых при росте базы. Рост базы может привести к нарушению логического представления данных и, как следствие, к искажению в прикладных программах.

Одним из простейших способов представления данных являются таблицы (плоские файлы). Они просты, удобны, наглядны, легко запоминаются.

Таблицы обладают следующими свойствами:

1. Каждый элемент таблицы представляет собой один элемент данных, повторяющиеся группы отсутствуют.
2. Все столбцы в таблице однородные – это означает, что элементы столбца имеют одинаковую природу.
3. Столбцам однозначно присвоены имена.
4. В таблице нет двух одинаковых строк.
5. В операциях с такой таблицей ее строки и столбцы могут просматриваться в любом порядке, в любой последовательности безотносительно к их информационному содержанию и смыслу.

Избежать растущей сложности древовидных и сетевых структур можно с помощью метода, называемого нормализацией. Принцип нормализации не связан с физическим представлением данных.

Реляционная структура - способ описания данных, который:

- понятен пользователю, не имеющему особых навыков в программировании;

- позволяет подсоединить новых пользователей без изменения существующей логической структуры и, следовательно, прикладных программ;

- допускает максимальную гибкость при формировании запросов с терминалов.

Задача преобразования древовидной структуры к табличной форме состоит во введении избыточности. В результате этого можно получить первую нормальную форму. Существует семь нормальных форм. Приведение к каждой из последующих нормальных форм связано с устранением избыточности.

В теории БД таблица называется отношением, а реляционную БД называют совокупностью отношений.

Таблица представляет собой множество кортежей. Если таблица имеет   
n - столбцов, то ее называют отношением степени n. Таблица, состоящая из 2-х столбцов, называется бинарным отношением; из 3-х столбцов - тернарным отношением; из n столбцов - n-арным отношением.

В математике отношение R определяется как множество строк на множествах S1, S2, ..., Sn, таких, что первый элемент каждой строки принадлежит S1, второй элемент - S2, ..., последний - Sn. Для описания таких таблиц используют обозначения реляционной алгебры и реляционных исчислений. С логической точки зрения реляционная БД - это множество двумерных таблиц с операциями извлечения и объединения столбцов.

Для сравнения древовидной, иерархической и реляционной структур данных существует два критерия:

1. Легкость использования. Основная часть издержек, особенно в малых БД, содержащих порядка нескольких тысяч записей, и средних БД, содержащих порядка нескольких десятков тысяч записей, приходится на написание прикладных программ и формирование запросов.
2. Эффективность. Для больших БД издержки приходятся на стоимость памяти и машинного времени, поэтому необходима эффективная реализация.

Исходя из этих критериев, для больших БД используют сетевую или иерархическую структуры, а для малых БД и средних БД - реляционную структуру. Так как разрабатываемая нами БД не столь велика, будем использовать реляционную модель.

3.2 Построение исходной концептуальной модели

Концептуальная модель применяется для структурирования ПО с учетом информационных потребностей самой ПО и информационных интересов пользователей системы и независима от конкретной СУБД.

После построения внешней модели базы данных переходим к следующему этапу проектирования – построению концептуальной модели. В результате перехода от внешней модели базы данных к концептуальной мы получили следующую структуру:



Рисунок 3.2.2 – Простая сетевая структура

Полученная структура является простой сетевой, так как в ней отсутствуют связи «многие ко многим». Её характерной чертой является то, что у потомков есть несколько родителей. У отношения Поставка потомки – Инструмент и Производитель, у отношения Покупка – Клиент и Продавец.

Иерархическая (древовидная) структура не допускает наличия нескольких родителей у одного потомка. Поэтому для приведения простой сетевой структуры к древовидной введем избыточность, продублировав отношения, которые являются потомками.



Рисунок 3.2.3 – Древовидная структура

3.3 Нормализация исходной модели

Нормализация – процесс приведения древовидной структуры к табличной форме.

Нормализация – это разбиение таблицы на две или более, обладающих лучшими свойствами при включении, изменении и удалении данных.

Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором каждый факт появляется лишь в одном месте, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных. Выделяют семь этапов нормализации.

Третья нормальная форма является достаточной при решении большинства практических задач, и процесс проектирования реляционной базы данных, как правило, заканчивается приведением к ней.

После получения древовидной структуры перейдем к преобразованию её к нормализованной структуре.

3.3.1 Первая нормальная форма

Таблица находится в первой нормальной форме (1НФ) тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения (условие атомарности) и ни одно из ее ключевых полей не пусто.

Приведение базы данных к 1-ой нормальной форме подразумевает устранение повторяющихся групп. Основой этого этапа нормализации является определение ключей. Свойства, которыми должен обладать ключ:

1. Однозначная идентификация кортежа;
2. Отсутствие избыточности (то есть никакой атрибут нельзя удалить из ключа, не нарушая при этом свойства однозначной идентификации.)

Условия приведения:

1. Уникальность всех PK;
2. Атомарность атрибутов;
3. Уникальность групп.

На основе анализа предметной области было определено, что первичные ключи, наследуемые из родительских таблиц не могут однозначно определять неключевые атрибуты, поэтому они были использованы не в качестве части первичного ключа, а в качестве внешнего ключа как атрибут. А так же, при спуске в атрибуты ключа код\_города из таблицы Город в таблицу Рейс и ключа номер\_рейса из таблицы Рейс в таблицу Тур, на основе анализа предметной области и ER-диаграммы, было добавлено не по 1, а по 2 внешних ключа: город\_вылета, город\_прибытия и рейс\_вылета, рейс\_возвращения.

Отношение Заявка:

Выделим первичным ключем поле номер\_заявки;

Отношение Тур:

Выделим первичным ключем поле номер\_тура;

Отношение Отель:

Выделим первичным ключем поле код\_отеля;

Отношение Рейс:

Выделим первичным ключем поле номер\_рейса;

Отношение Город:

Выделим первичным ключем поле код\_города;

Отношение Страна:

Выделим первичным ключем поле название\_страны;

Таким образом мы получаем следующие таблицы с полями:

Заявка (номер\_заявки, ФИО, e-mail, телефон, номер\_тура);

Заявка (номер\_заявки, ФИО, e-mail, телефон, номер\_тура);

Тур (номер\_тура, количество\_дней, количество\_людей, рейс\_вылета, рейс\_возвращения, количество\_путёвок, цена);

Тур (номер\_тура, количество\_дней, код\_отеля, количество\_людей, количество\_путёвок, цена);

Отель (код\_отеля, название\_отеля, количество\_звёзд, код\_города);

Рейс (номер\_рейса, город\_вылета, город\_прибытия, дата\_вылета, дата\_прибытия);

Город (код\_города, название, страна);

Страна (код\_страны, название);

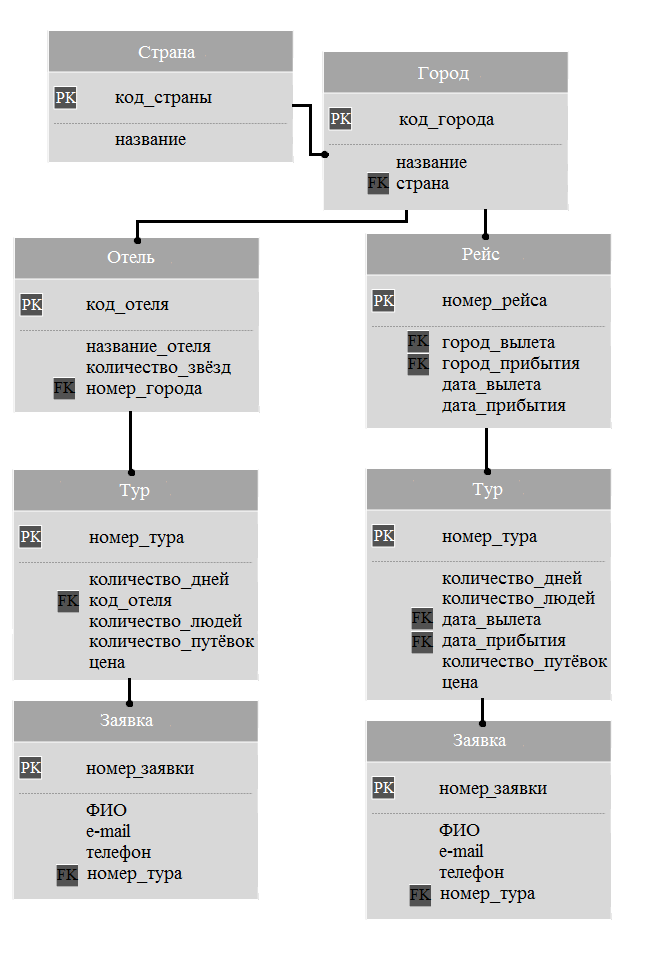


Рисунок 3.3.1 – Определение первичных ключей и атрибутов

Все первичные ключи однозначно идентифицируют кортежи, все поля атомарны. Отношения находятся в первой нормальной форме.

3.3.2 Вторая нормальная форма

Отношение задано во *второй нормальной форме*, если оно является отношением в первой нормальной форме и каждый атрибут, не являющийся первичным атрибутом в этом отношении, полностью зависит от любого возможного ключа этого отношения.

Если все возможные ключи отношения содержат по одному атрибуту, то это отношение задано во второй нормальной форме, так как в этом случае все атрибуты, не являющиеся первичными, полностью зависят от возможных ключей.

В рассматримаемой системе все полученные отношения полностью удовлетворяют требованиям второй нормальной формы, т.к. не имеется составных первичных ключей, т.е. все не ключевые поля связаны полной функциональной зависимостью с ключевыми.

3.3.3 Третья нормальная форма

Отношение задано в третьей нормальной форме, если оно задано во второй нормальной форме и каждый атрибут этого отношения, не являющийся первичным, не транзитивно зависит от каждого возможного ключа этого отношения.

Транзитивная зависимость выявляет дублирование данных в одном отношении. Если А, В и С – три атрибута одного отношения и С зависит от В, а В от А, то говорят, что С транзитивно зависит от А. Преобразование в третью нормальную форму происходит за счет разделения исходного отношения на два.

Проанализируем зависимости между атрибутами таблиц:

Заявка

номер\_заявки

ФИО

e-mail

телефон

номер\_путёвки

Тур

номер\_тура

количество\_дней

код \_отеля

количество\_людей

дата\_вылета

дата\_прибытия

количество\_путёвок

цена

Отель

код\_отеля\*

название\_отеля

количество\_звёзд

номер\_города

Рейс

номер\_рейса\*

город\_вылета

город\_прибытия

дата\_вылета

дата\_прибытия

Город

код\_города\*

название

cтрана

Страна

код страны

название

В результате анализа было выявлено, что транзитивные зависимости отсутствуют, т.е. требования третьей нормальной формы удовлетворены.

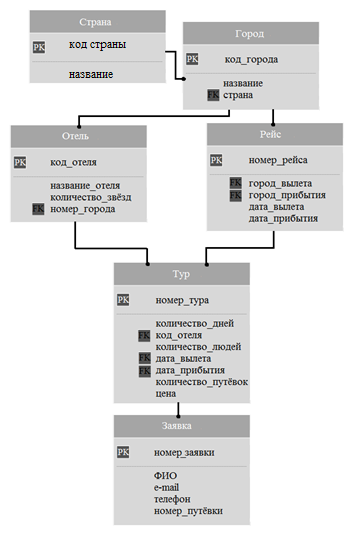


Рисунок 3.3.2 – Логическая схема базы данных

При приведении к иерархической структуре таблицы были продублированы, искусственно вводя избыточность, для того, чтобы у каждого дочернего элемента было не более одного родителя. Реляционная же модель допускает, чтобы у дочернего элемента было несколько родителей. Благодаря этому уменьшается избыточность базы данных, также это способствует целостности. Т.к. структура дублирующихся таблиц практически идентична, они были обьеденены.

4. Реализация базы данных

4.1 Обоснование выбора СУБД

В качестве СУБД была выбрана MySQL. MySQL — свободно распространяемая СУБД с открытым кодом. MySQL – кроссплатформенная система. Ее можно использовать практически во всех современных операционных системах, в том числе Windows, Linux, Mac OS, Solaris, HP-UX и др.

MySQL имеет отличные технические характеристики: многопоточность, многополь-зовательский доступ, быстродействие, масштабируемость (компания-разработчик приводит пример MySQL-сервера, который работает с 60 тыс. таблиц, содержащими приблизительно

5 млрд строк). Все эти особенности сделали MySQL универсальной СУБД, используемой на серверах хост-провайдеров и идеально подходит по нашу базу данных и web-сайт.

4.2 Разграничение прав доступа

Для работы с проектируемой базой данных было выделено 2 пользователя, имеющие различные уровни доступа – это администратор и клиент.

Администратор имеет возможность добавлять путёвки и просматривать заявки клиентов. Клиент может находить необходимый ему тур (по заданным критериям) и бронировать его.

4.3 Физическая реализация базы данных

Для хранения и манипулирования данными с помощью СУБД были созданы таблицы и установлены связи между ними:

Создание таблицы «Заявка»:

create table bid (

bid\_id int not null auto\_increment,

fio varchar (40) not null,

email varchar(50) not null,

phone int not null,

id\_tour int not null,

primary key(bid\_id))

ENGINE=MyISAM DEFAULT CHARSET=cp1251;

alter table bid add foreign key(id\_tour) references tour(id\_tour) on delete cascade on update cascade;

Создание таблицы «Тур»:

create table tour (

id\_tour int not null auto\_increment,

days int not null,

id\_hotel int not null,

count\_person int not null,

flight\_departure int not null,

flight\_arrival int not null,

count int not null,

price int not null,

primary key(id\_tour)

)

ENGINE=MyISAM DEFAULT CHARSET=cp1251;

alter table tour add foreign key(id\_hotel) references hotel(id\_hotel) on delete cascade on update cascade;

alter table tour add foreign key(flight\_departure) references flight(id\_flight) on delete cascade on update cascade;

alter table tour add foreign key(flight\_arrival) references flight(id\_flight) on delete cascade on update cascade;

Создание таблицы «Рейс»:

create table flight(

id\_flight int not null auto\_increment,

city\_departure int not null,

city\_arrival int not null,

date\_departure date not null,

date\_arrival date not null,

primary key(id\_flight)

)

ENGINE=MyISAM DEFAULT CHARSET=cp1251;

alter table flight add foreign key(city\_departure) references city(id\_city) on delete cascade on update cascade;

alter table flight add foreign key(city\_arrival) references city(id\_city) on delete cascade on update cascade;

Создание таблицы «Отель»:

create table hotel (

id\_hotel int not null auto\_increment,

hotel\_name varchar(25) not null,

star enum('2','3','4','5') not null,

city int not null,

primary key(id\_hotel))

ENGINE=MyISAM DEFAULT CHARSET=cp1251;

alter table hotel add foreign key(city) references city(id\_city) on delete cascade on update cascade;

Создание таблицы «Город»:

create table city (

id\_city int not null auto\_increment,

city\_name varchar(25) not null,

country int not null,

primary key(id\_city))

ENGINE=MyISAM DEFAULT CHARSET=cp1251;

alter table city add foreign key(country) references country(id\_country) on delete cascade on update cascade;

Создание таблицы «Страна»:

create table country (

id\_country int not null auto\_increment,

country\_name varchar(20),

primary key(id\_country))

ENGINE=MyISAM DEFAULT CHARSET=cp1251;

Таким образом, была создана база данных.

4.4 Тестирование базы данных

Выделим наиболее характерные запросы для нашей базы данных:

* "select id\_city from city join country on country=id\_country where country\_name='Россия' and city\_name='Симферополь'"
* "select id\_tour,price, hotel\_name, star from tour join hotel on tour.id\_hotel=hotel.id\_hotel where count>0 and days='9' and count\_person='1'and flight\_departure=' 5' and star='4' and price between '10000' and '$price[1]'15000"
* "insert into tour (id\_tour,days, id\_hotel,count\_person,flight\_departure,flight\_arrival,count, price) values(0,'9','3','5','6','7','5','50000')"
* "insert into flight (id\_flight, city\_departure, city\_arrival, date\_departure, date\_arrival) values(0,'5','6','2014-06-06','2014-06-07')"
* "insert into hotel(id\_hotel,hotel\_name,star, city) values(0,'Плаза','4','5')"
* "insert into bid(bid\_id,fio,email,phone, id\_tour) values(0,'Глобин Андрей Витальевич','globinav@rambler.ru','0508547802','255')"
* "select fio, email, phone, price, city\_name, hotel\_name, days from ((bid join tour on bid.id\_tour=tour.id\_tour) join hotel on tour.id\_hotel=hotel.id\_hotel join city on city.id\_city=hotel.city"

В ходе тестирования данного приложения ошибок не было выявлено. Приложение работает без сбоев и согласно поставленной задаче.

5. Исследование информационных параметров базы данных

Длина логической записи:

 [байт]

где Mj - число групп полей в записях, lij длина группы [байт], Lj и средняя длины записи m{ Lj }.

 [байт] ;

 [байт] ;

 [байт] ;

 [байт] ;

 [байт] ;

 [байт] ;



 [байт] ;

Объем памяти, необходимый для размещения информационного фонда без учёта системных данных и указателей : [байт]

где N - количество отношений реляционной базы данных, K j - количество записей j-го файла

 [байт] ;

 [байт] ;

 [байт] ;

 [байт] ;

 [байт] ;

 [байт] ;



 [байт] ;

Приращение информационного фонда

 [байт-1]

где  - число добавленных типов записей,  - интенсивность добавления записей в файл j -го типа.

 [байт-1];

 [байт-1];

 [байт-1];

 [байт-1];

 [байт-1];

 [байт-1];

 [байт-1];

 [байт-1];

Время заполнения информационного фонда:

 [время] , где

Vобщ. = 10 МБ = 10485760 байт;

Vпо = 250000 байт.

.

Время резервного копирования:





6. Описание интерфейса пользователя

Для доступа к системе клиенту необходимо знать адрес ресурса, на котором расположен сайт. Для тестирования web-сайта мы используем следующий url: <http://localhost/cursach>

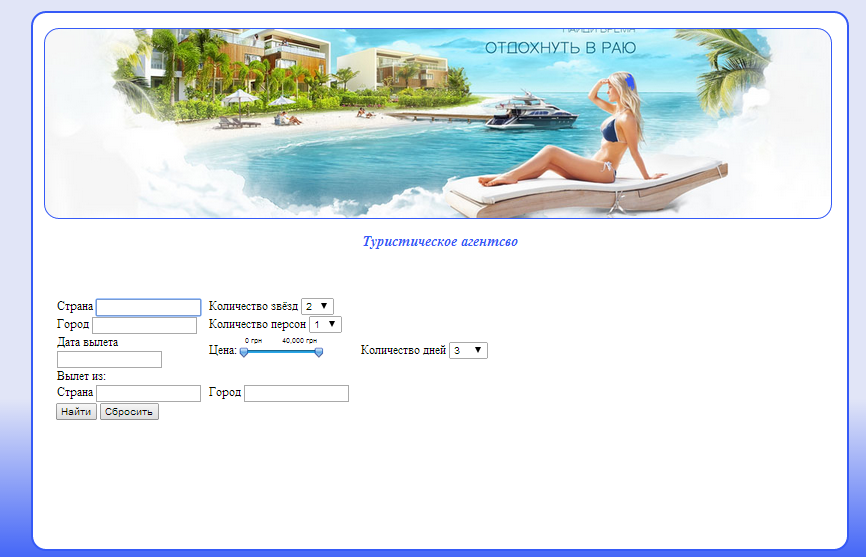


Рисунок 4.4.1 – Главная страница

После перехода по адресу открывается главная страница сайта, на которой сразу отображены элементы для поиска и бронирования туров.

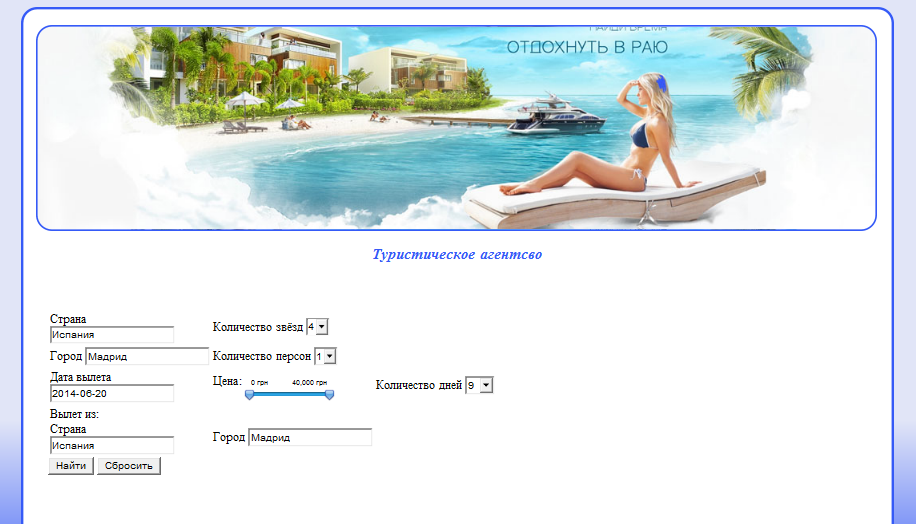


Рисунок 4.4.2 – Заполнение форм

На ней можно наблюдать элементы для выбора желаемой страны и города, в который планируется поездка, желаемой даты вылета, страны и города из которых будет осуществлён вылет, звёздности отеля, количества человек, участвующих в поездке, дипазона стоимости тура и количества дней отдыха.

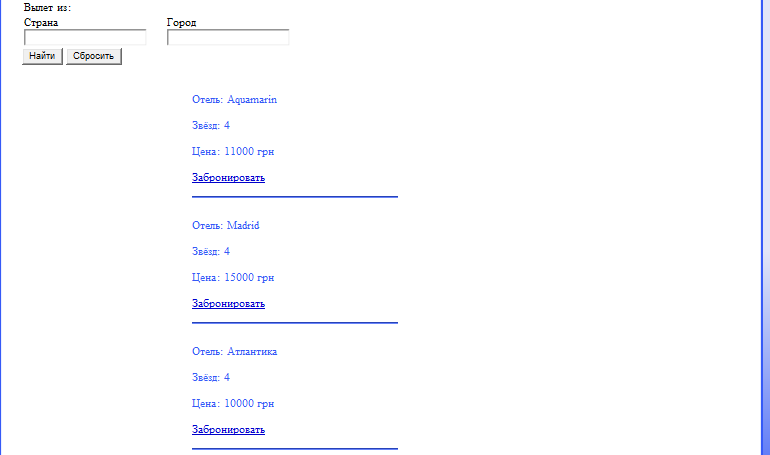


Рисунок 4.4.3 – Результаты запроса

После ввода данных формируется запрос, результатом которого служит список туров, удовлетворяющих заданным условиям, с краткой информацией о каждом, и предлагается выбрать наиболее подходящий нажатием кнопки «Забронировать».

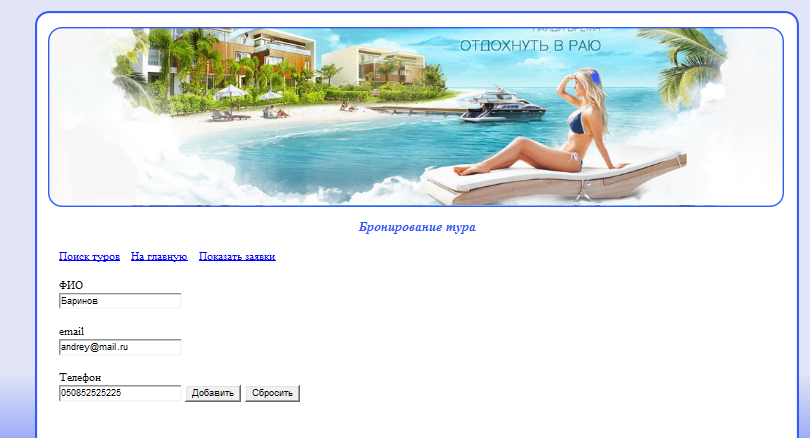


Рисунок 4.4.4 – Страница бронирования

Когда пользователь окончательно решил какой именно тур он хочет забронировать, ему нужно ввести свои личные контактные данные, а именно – ФИО, e-mail и телефон.

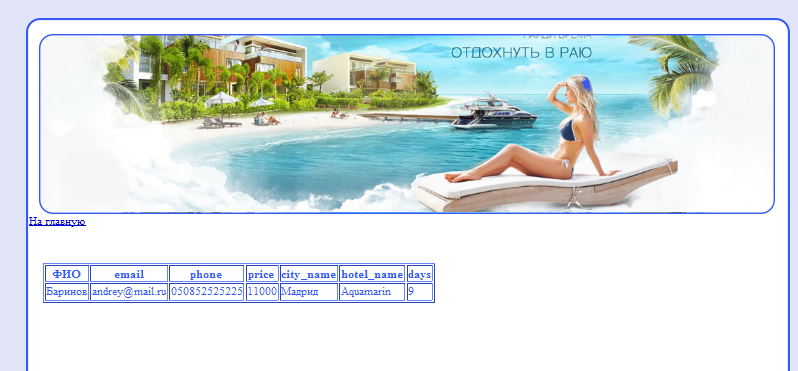


Рисунок 4.4.5 – Результирующая страница

После получения всех необходимых данных, пользователю выводится информация о заявке, сформированной с помощью введённых им данных. Остаётся лишь дождаться звонка от туроператора после обработки заявки.

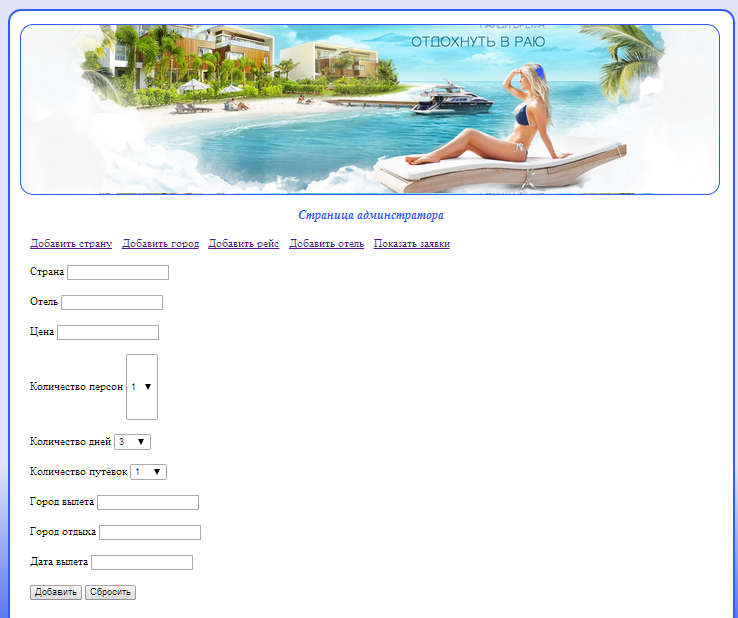


Рисунок 4.4.6 – Страница администратора

Кроме страниц, предназначенных для взаимодействия с клиентом, есть так же страницы для работы администратора. Они, в большей мере, реализованы для добавления новых записей в таблицы базы данных.

Заключение

В данном курсовом проекте была спроектирована и реализована реляционная база данных на тему «Web-Сайт туристической фирмы». На основе исходных данных о предметной области был произведен подробный анализ, в результате которого была разработана иерархическая модель БД, определены необходимые операции, выполняемые над БД.

На языке PHP был разработан web-интерфейс, реализующий обращение к базе данных, который выдаёт подходящие путёвки клиентам.

В ходе работы были закреплены навыки организации баз данных, разработки программного обеспечения для него и обеспечения секретности.

Библиографический список

1. Гарсиа-Молина Г. Системы баз данных. Полный курс: Пер. с англ./Г.Гарсиа-Молина, Д.Д.Ульман, Д.Уидом.– М.: Издательский дом «Вильямс», 2004.–1088с.

2. Дейт К.Дж. Введение в системы баз данных, 7-е издание.:Пер. с англ./ К.Дж.Дейт.– М.: Издательский дом «Вильямс», 2002.–1072с.

3. Ульман Д. Основы систем баз данных/Д.Ульман.– М.: Финансы и статистика, 1983. – 334 с.

Приложение A

//Главная Страница сайта

<!DOCTYPE html>

<html>

<head>

<title>Персональный сайт Андрея Глобина</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<style>

.label {

font-size: 16px;

color: black;

position: relative;

margin-left: -9px;

display: block;

bottom: 10px;

}

form {

margin-left:50px;

}

form input {

/\* margin-right: 40px;\*/

}

.data td {

margin-right: 20px;

width:200px;

}

.list {

margin-left:270px;

margin-top: 20px;

width: 300px;

}

</style>

<link rel="stylesheet" href="slider-range/css/jslider.css" type="text/css"/>

<link rel="stylesheet" href="slider-range/css/jslider.plastic.css" type="text/css"/>

<link rel="stylesheet" href="slider-range/css/jslider.round.css" type="text/css"/>

<link rel="stylesheet" href="slider-range/css/jslider.round.plastic.css" type="text/css"/>

<link rel="stylesheet" href="slider-range/css/jslider.blue.css" type="text/css"/>

<script type="text/javascript" src="slider-range/js/jquery-1.7.1.js"></script>

<script type="text/javascript" src="slider-range/js/jshashtable-2.1\_src.js"></script>

<script type="text/javascript" src="slider-range/js/jquery.numberformatter-1.2.3.js"></script>

<script type="text/javascript" src="slider-range/js/tmpl.js"></script>

<script type="text/javascript" src="slider-range/js/jquery.dependClass-0.1.js"></script>

<script type="text/javascript" src="slider-range/js/draggable-0.1.js"></script>

<script type="text/javascript" src="slider-range/js/jquery.slider.js"></script>

</head>

<body>

<div class="wrapper">

<div class="main">

<div class="header">

</div><!--end header-->

<div class="content">

<h3>Туристическое агентсво</h3>

<br /> <br />

<form method="post" action="<? echo $\_SERVER["PHP\_SELF"]?>" style="margin-left: 30px; color: black">

<table class="data">

<tr>

<td>Cтрана <input type="text" name="country\_to" /></td>

<td>

Количество звёзд

<select name="star">

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

</select>

</td>

</tr>

<tr><td >Город <input type="text" name="city\_to" /></td>

<td>

Количество персон

<select name="count\_person">

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

</select>

</td>

</tr>

<tr><td>

Дата вылета

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

</td>

<td>Цена:

<span style="display: inline-block; width: 100px; padding: 0 5px;">

<span class="label"> </span>

<input id="Slider1" type="slider" name="price" value="0;40000" /></span>

<script type="text/javascript" charset="utf-8">

jQuery("#Slider1").slider({ from: 0, to: 40000, step: 1000, smooth: true, round: 0, dimension: "&nbsp;грн", skin: "plastic" });

</script>

</td>

<td>

Количество дней

<select name="days">

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

<option value="8">8</option>

<option value="9">9</option>

<option value="10">10</option>

<option value="11">11</option>

<option value="12">12</option>

<option value="13">13</option>

<option value="14">14</option>

<option value="15">15</option>

<option value="16">16</option>

<option value="17">17</option>

<option value="18">18</option>

<option value="19">19</option>

<option value="20">20</option>

</select>

</td>

</tr>

<tr>

<td>

Вылет из:<br />

Cтрана <input type="text" name="country\_with"/>

</td>

<td>

<br />

Город <input type="text" name="city\_with"/>

</td>

</tr>

</table>

<input type="submit" value="Найти"/>

<input type="reset" value="Сбросить"/>

</form>

<?

if ($\_SERVER["REQUEST\_METHOD"]=="POST"){

include "Seach.php";

if(!search())

echo "<p align=center>Ничего не найдено</p>";

}

?>

</div><!--end content-->

</div><!--end main-->

</div><!--end wrapper-->

</body>

</html>

//Функция поиска путёвок

<?

function search() {

$dblocation = "localhost"; //Имя сервера

$dbuser = "root"; //Имя пользователя

$dbpasswd = ""; //Пароль

$dbname="wet";

//Осуществляем соединение с сервером базы данных

$mysqli = new mysqli($dblocation, $dbuser, $dbpasswd, $dbname);

//Объявление переменных

$city\_with=0; //город вылета

$city\_to=0; //город прилёта

$flight\_with=0; //рейс на отдых

/\*-------Поиск городов------------------------\*/

$do\_get = $mysqli->query("select id\_city from city join country on country=id\_country

where country\_name='${\_POST['country\_to']}' and city\_name='${\_POST['city\_to']}'");

$num\_rows = $do\_get->num\_rows;

if (!$num\_rows)

return false;

else {

$c=$do\_get->fetch\_array();

$city\_to=$c[0];

}

$do\_get = $mysqli->query("select id\_city from city join country on city.country=country.id\_country

where country\_name='${\_POST['country\_with']}' and city\_name='${\_POST['city\_with']}'");

$num\_rows = $do\_get->num\_rows;

if (!$num\_rows)

return false;

else {

$c=$do\_get->fetch\_array();

$city\_with=$c[0];

}

/\*------Поиск рейсов---------------\*/

$do\_get = $mysqli->query("select id\_flight from flight

where city\_departure='$city\_with' and city\_arrival='$city\_to' and date\_departure='${\_POST['date']}'");

$num\_rows = $do\_get->num\_rows;

if (!$num\_rows)

return false;

else {

$c=$do\_get->fetch\_array();

$flight\_with=$c[0];

}

//------Интеррвалы цены-------------

$price=explode(";",$\_POST['price']);

//Поиск туров

$do\_get =$mysqli->query("select id\_tour,price, hotel\_name, star from tour join

hotel on tour.id\_hotel=hotel.id\_hotel where count>0 and

days='${\_POST['days']}' and count\_person='${\_POST['count\_person']}'and flight\_departure=' $flight\_with'

and star='${\_POST['star']}'

and price between '$price[0]' and '$price[1]'");

$num\_rows = $do\_get->num\_rows;

if (!$num\_rows )

return false;

echo "<table class='list'>";

while ($results = $do\_get->fetch\_array()){

echo "<tr>";

echo "<td>";

echo "<p>Отель: $results[2]</p>";

echo "<p>Звёзд: $results[3]</p>";

echo "<p>Цена: $results[1] грн</p>";

echo "<p><a href='bid.php"."?id\_tour=$results[0]'>"."Забронировать"."</a> </p>";

echo "<hr>";

echo "</td>";

echo "<tr>";

}

echo "</table>";

return true;

}

?>

//Бронирование клиентом тура

<!DOCTYPE html>

<html>

<head>

<title>Персональный сайт Андрея Глобина</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<style>

.menu {

margin-left:20px;

}

.menu a{

margin-left:10px;

color: blue;

}

.menu a:hover{

color: purple;

}

</style>

</script>

</head>

<body>

<?

$dblocation = "localhost"; //Имя сервера

$dbuser = "root"; //Имя пользователя

$dbpasswd = ""; //Пароль

$dbname="wet";

//Осуществляем соединение с сервером базы данных

$mysqli = new mysqli($dblocation, $dbuser, $dbpasswd, $dbname);

if (isset($\_GET['id\_tour'])) $id\_tour=$\_GET['id\_tour']; else $id\_tour=0;

if ($\_SERVER["REQUEST\_METHOD"]=="POST"){

if ($\_POST['fio']!="" && $\_POST['email']!="" && $\_POST['phone']!="") {

$sql="insert into bid(bid\_id,fio,email,phone, id\_tour)

values(0,'${\_POST['fio']}','${\_POST['email']}','${\_POST['phone']}','$id\_tour')";

$mysqli->query($sql);

$sql="select count from tour where id\_tour='$id\_tour' ";

$d=$mysqli->query($sql);

$c=$d->fetch\_array();

if ($c[0]>0) {

$c[0]=$c[0]-1;

$sql="update tour set count= '$c[0]' where id\_tour=$id\_tour";

$mysqli->query($sql);

}

}

else {echo "<script> alert('не верные данные')</script>";}

}

?>

<div class="wrapper">

<div class="main">

<div class="header">

</div><!--end header-->

<div class="content">

<h3>Бронирование тура</h3>

<span class="menu">

<a href="index.php">Поиск туров</a>

<a href="index.php">На главную</a>

<a href="show\_bid.php">Показать заявки</a>

</span>

<br /> <br />

<form method="post" action="<? echo $\_SERVER["PHP\_SELF"."?id\_tour=$id\_tour"]?>" style="margin-left: 30px; color: black">

ФИО <br /><input type="text" name="fio"/>

<br /> <br />

email<br /><input type="text" name="email"/>

<br /> <br />

Телефон<br /><input type="text" name="phone"/>

<input type="submit" value="Добавить"/>

<input type="reset" value="Сбросить"/>

</form>

</div><!--end content-->

</div><!--end main--

</div><!--end wrapper-->

</body>

</html>

//Создание тура администратором

<!DOCTYPE html>

<html>

<head>

<title>Персональный сайт Андрея Глобина</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<style>

.menu {

margin-left:20px;

}

.menu a{

margin-left:10px;

}

.menu a:hover{

color: purple;

}

</style>

</script>

</head>

<body>

<?

function check() {

$flag=true;

if($\_POST['hotel']=="")

$flag=false;

if($\_POST['country']=="")

$flag=false;

if($\_POST['price']=="")

$flag=false;

if($\_POST['city\_with']=="")

$flag=false;

if($\_POST['city\_to']=="")

$flag=false;

if($\_POST['date']=="")

$flag=false;

return $flag;

}

$dblocation = "localhost"; //Имя сервера

$dbuser = "root"; //Имя пользователя

$dbpasswd = ""; //Пароль

$dbname="wet";

//Осуществляем соединение с сервером базы данных

$mysqli = new mysqli($dblocation, $dbuser, $dbpasswd, $dbname);

$id1=0;

$id2=0;

$id2=0;

$id4=0;

$id5=0;

if ($\_SERVER["REQUEST\_METHOD"]=="POST"){

if (check()) {

$do\_get4 = $mysqli->query("select id\_city from city where city\_name='${\_POST['city\_with']}'");

$num\_rows4= $do\_get4->num\_rows;

if ($num\_rows4) {

$c4=$do\_get4->fetch\_array();

$id4=$c4[0];

}

else {echo "<script> alert('нет такого города')</script>";

//return;

}

$do\_get5 = $mysqli->query("select id\_city from city where city\_name='${\_POST['city\_to']}'");

$num\_rows5= $do\_get5->num\_rows;

if ($num\_rows5) {

$c5=$do\_get5->fetch\_array();

$id5=$c5[0];

//echo "<span style='color=black'>$id5</span>";

}

else {echo "<script> alert('нет такого города')</script>";

//return;

}

$do\_get1 = $mysqli->query("select id\_flight, date\_arrival from flight where city\_departure='$id4' and date\_departure='${\_POST['date']}'");

$num\_rows1 = $do\_get1->num\_rows;

if($num\_rows1 ){

$c1=$do\_get1->fetch\_array();

$id1=$c1[0];

$date=$c1[1];

//echo "<span style='color=black'>$date</span>";

$do\_get3 = $mysqli->query("select adddate('$date','${\_POST['days']}')");

$c1=$do\_get3->fetch\_array();

//echo $c1[0];

$do\_get2 = $mysqli->query("select id\_flight from flight where city\_departure='$id5' and date\_departure=adddate('$date','${\_POST['days']}')");

$num\_rows2 = $do\_get2->num\_rows;

if ($num\_rows2) {

$c2=$do\_get2->fetch\_array();

$id2=$c2[0];

}

else echo "<script> alert('Не верные данные!!')</script>";

}

else echo "<script> alert('Не верные данные!')</script>";

$do\_get3 = $mysqli->query("select id\_hotel from hotel where hotel\_name='${\_POST['hotel']}'");

$num\_rows3= $do\_get3->num\_rows;

if ($num\_rows3) {

$c3=$do\_get3->fetch\_array();

$id3=$c3[0];

//echo "<span style='color=black'>$id3</span>";

}

else echo "<script> alert('Не верные данные!!!')</script>";

$sql="insert into tour (id\_tour,days, id\_hotel,count\_person,flight\_departure,flight\_arrival,count, price)

values(0,'${\_POST['days']}','$id3','${\_POST['count\_person']}','$id1','$id2','${\_POST['count']}','${\_POST['price']}')";

$mysqli->query($sql);

}

else echo "<script> alert('Не верные данные!!!!')</script>";

}

?>

<div class="wrapper">

<div class="main">

<div class="header">

</div><!--end header-->

<div class="content">

<h3>Страница aдминстратора</h3>

<span class="menu">

<a href="admin\_country.php">Добавить страну</a>

<a href="admin\_citi.php">Добавить город</a>

<a href="admin\_flight.php">Добавить рейс</a>

<a href="admin\_hotel.php">Добавить отель</a>

<a href="show\_bid.php">Показать заявки</a>

</span>

<br /> <br />

<form method="post" action="<? echo $\_SERVER["PHP\_SELF"]?>" style="margin-left: 30px; color: black">

Страна <input type="text" name="country"/>

<br /> <br />

Отель <input type="text" name="hotel"/>

<br /> <br />

Цена <input type="text" name="price"/>

<br /> <br />

Количество персон

<select name="count\_person" style="height: 100px; width: 100;">

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

</select>

<br /> <br />

Количество дней

<select name="days">

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

<option value="8">8</option>

<option value="9">9</option>

<option value="10">10</option>

<option value="11">11</option>

<option value="12">12</option>

<option value="13">13</option>

<option value="14">14</option>

<option value="15">15</option>

<option value="16">16</option>

<option value="17">17</option>

<option value="18">18</option>

<option value="19">19</option>

<option value="20">20</option>

</select>

<br /> <br />

Количество путёвок

<select name="count">

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

<option value="8">8</option>

<option value="9">9</option>

<option value="10">10</option>

</select>

<br /> <br />

Город вылета <input type="text" name="city\_with"/>

<br /> <br />

Город отдыха <input type="text" name="city\_to"/>

<br /> <br />

Дата вылета <input type="text" name="date"/>

<br /> <br />

<input type="submit" value="Добавить"/>

<input type="reset" value="Сбросить"/>

</form

</div><!--end content-->

</div><!--end main-->

</div><!--end wrapper-->

</body>

</html>

//Добавление отеля в базу данных

<!DOCTYPE html>

<html>

<head>

<title>Персональный сайт Андрея Глобина</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<style>

.menu {

margin-left:20px;

}

.menu a{

margin-left:10px;

color: blue;

}

.menu a:hover{

color: purple;

}

</style>

</script>

</head>

<body>

<?

$dblocation = "localhost"; //Имя сервера

$dbuser = "root"; //Имя пользователя

$dbpasswd = ""; //Пароль

$dbname="wet";

//Осуществляем соединение с сервером базы данных

$mysqli = new mysqli($dblocation, $dbuser, $dbpasswd, $dbname);

if ($\_SERVER["REQUEST\_METHOD"]=="POST"){

$do\_get = $mysqli->query("select id\_city from city where city\_name='${\_POST['city']}'");

$num\_rows = $do\_get->num\_rows;

if($num\_rows && $\_POST['hotel']!=""){

$c=$do\_get->fetch\_array();

$id=$c[0];

$sql="insert into hotel(id\_hotel,hotel\_name,star, city)

values(0,'${\_POST['hotel']}','${\_POST['star']}','$id')";

$mysqli->query($sql);

}

else echo "<script> alert('Неверные данные!')</script>";

}

?>

<div class="wrapper">

<div class="main">

<div class="header">

</div><!--end header-->

<div class="content">

<h3>Страница aдминстратора</h3>

<span class="menu">

<a href="admin\_country.php">Добавить страну</a>

<a href="admin\_citi.php">Добавить город</a>

<a href="admin\_flight.php">Добавить рейс</a>

<a href="admin\_hotel.php">Добавить отель</a>

<a href="show\_bid.php">Показать заявки</a>

</span>

<br /> <br />

<form method="post" action="<? echo $\_SERVER["PHP\_SELF"]?>" style="margin-left: 30px; color: black">

Город <input type="text" name="city"/>

<br /> <br />

Название Отеля <input type="text" name="hotel"/>

<br /> <br />

Звёзды

<select name="star">

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

</select>

<input type="submit" value="Добавить"/>

<input type="reset" value="Сбросить"/>

</form>

</div><!--end content-->

</div><!--end main-->

</div><!--end wrapper-->

</body>

</html>

//Добавление Города в базу данных администатором

<!DOCTYPE html>

<html>

<head>

<title>Персональный сайт Андрея Глобина</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<style>

.menu {

margin-left:20px;

}

.menu a{

margin-left:10px;

color: blue;

}

.menu a:hover{

color: purple;

}

</style>

</script>

</head>

<body>

<?

$dblocation = "localhost"; //Имя сервера

$dbuser = "root"; //Имя пользователя

$dbpasswd = ""; //Пароль

$dbname="wet";

//Осуществляем соединение с сервером базы данных

$mysqli = new mysqli($dblocation, $dbuser, $dbpasswd, $dbname);

if ($\_SERVER["REQUEST\_METHOD"]=="POST"){

$do\_get = $mysqli->query("select id\_country from country where country\_name='${\_POST['country']}'");

$num\_rows = $do\_get->num\_rows;

if($num\_rows){

$c=$do\_get->fetch\_array();

$id=$c[0];

$sql="insert into city(id\_city,country, city\_name)

values(0,'$id','${\_POST['city']}')";

$mysqli->query($sql);

}

else echo "<script> alert('Нет такой страны!')</script>";

}

?>

<div class="wrapper">

<div class="main">

<div class="header">

</div><!--end header-->

<div class="content">

<h3>Страница aдминстратора</h3>

<span class="menu">

<a href="admin\_country.php">Добавить страну</a>

<a href="admin\_citi.php">Добавить город</a>

<a href="admin\_flight.php">Добавить рейс</a>

<a href="admin\_hotel.php">Добавить отель</a>

<a href="show\_bid.php">Показать заявки</a>

</span>

<br /> <br />

<form method="post" action="<? echo $\_SERVER["PHP\_SELF"]?>" style="margin-left: 30px; color: black">

Страна <input type="text" name="country"/>

Город <input type="text" name="city"/>

<br /> <br />

<input type="submit" value="Добавить"/>

<input type="reset" value="Сбросить"/>

</form>

</div><!--end content-->

</div><!--end main-->

</div><!--end wrapper-->

</body>

<script type="text/javascript">

ShowDate();

</script>

</html>

//Добавление Рейса в базу данных

<!DOCTYPE html>

<html>

<head>

<title>Персональный сайт Андрея Глобина</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<style>

.menu {

margin-left:20px;

}

.menu a{

margin-left:10px;

color: blue;

}

.menu a:hover{

color: purple;

}

</style>

</script>

</head>

<body>

<?

$dblocation = "localhost"; //Имя сервера

$dbuser = "root"; //Имя пользователя

$dbpasswd = ""; //Пароль

$dbname="wet";

//Осуществляем соединение с сервером базы данных

$mysqli = new mysqli($dblocation, $dbuser, $dbpasswd, $dbname);

if ($\_SERVER["REQUEST\_METHOD"]=="POST"){

if ($\_POST['date\_with']!="" && $\_POST['date\_to']!="") {

$do\_get1 = $mysqli->query("select id\_city from city where city\_name='${\_POST['city\_with']}'");

$num\_rows1 = $do\_get1->num\_rows;

$do\_get2 = $mysqli->query("select id\_city from city where city\_name='${\_POST['city\_to']}'");

$num\_rows2 = $do\_get2->num\_rows;

if($num\_rows1 && $num\_rows2 ){

$c1=$do\_get1->fetch\_array();

$id1=$c1[0];

$c2=$do\_get2->fetch\_array();

$id2=$c2[0];

$sql="insert into flight (id\_flight, city\_departure, city\_arrival, date\_departure, date\_arrival)

values(0,'$id1','$id2','${\_POST['date\_with']}','${\_POST['date\_to']}')";

$mysqli->query($sql);

}

else echo "<script> alert('Не верные данные!')</script>";

}

else echo "<script> alert('Не верные данные!!')</script>";

}

?>

<div class="wrapper">

<div class="main">

<div class="header">

</div><!--end header-->

<div class="content">

<h3>Страница aдминстратора</h3>

<span class="menu">

<a href="admin\_country.php">Добавить страну</a>

<a href="admin\_citi.php">Добавить город</a>

<a href="admin\_flight.php">Добавить рейс</a>

<a href="admin\_hotel.php">Добавить отель</a>

<a href="show\_bid.php">Показать заявки</a>

</span>

<br /> <br />

<form method="post" action="<? echo $\_SERVER["PHP\_SELF"]?>" style="margin-left: 30px; color: black">

Город вылета <input type="text" name="city\_with"/>

Город прилёта<input type="text" name="city\_to"/>

<br /> <br />

Дата вылета <input type="text" name="date\_with"/>

<br /> <br />

Дата прилёта <input type="text" name="date\_to"/>

<br /> <br />

<input type="submit" value="Добавить"/>

<input type="reset" value="Сбросить"/>

</form>

</div><!--end content-->

</div><!--end main-->

</div><!--end wrapper-->

</body>

<script type="text/javascript">

ShowDate();

</script>

</html>

//Добавление страны в базу данных

<!DOCTYPE html>

<html>

<head>

<title>Персональный сайт Андрея Глобина</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<style>

.menu {

margin-left:20px;

}

.menu a{

margin-left:10px;

color: blue;

}

.menu a:hover{

color: purple;

}

</style>

</head>

<body>

<?

$dblocation = "localhost"; //Имя сервера

$dbuser = "root"; //Имя пользователя

$dbpasswd = ""; //Пароль

$dbname="wet";

//Осуществляем соединение с сервером базы данных

$mysqli = new mysqli($dblocation, $dbuser, $dbpasswd, $dbname);

if ($\_SERVER["REQUEST\_METHOD"]=="POST"){

if ($\_POST['country']!="") {

$sql="insert into country(id\_country, country\_name)

values(0,'${\_POST['country']}')";

$mysqli->query($sql);

}

else echo "<script> alert('Введите страну')</script>";

}

?>

<div class="wrapper">

<div class="main">

<div class="header">

</div><!--end header-->

<div class="content">

<h3>Страница aдминстратора</h3>

<span class="menu">

<a href="admin\_country.php">Добавить страну</a>

<a href="admin\_citi.php">Добавить город</a>

<a href="admin\_flight.php">Добавить рейс</a>

<a href="admin\_hotel.php">Добавить отель</a>

<a href="show\_bid.php">Показать заявки</a>

</span>

<br /> <br />

<form method="post" action="<? echo $\_SERVER["PHP\_SELF"]?>" style="margin-left: 30px; color: black">

Страна <input type="text" name="country"/>

<input type="submit" value="Добавить"/>

<input type="reset" value="Сбросить"/>

</form>

</div><!--end content-->

</div><!--end main-->

</div><!--end wrapper-->

</body>

</html>

//Показ заявок пользователя

<!DOCTYPE html>

<html>

<head>

<title>Персональный сайт Андрея Глобина</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<style>

table {

margin-left:20px;

margin-top: 50px;

</style>

</script>

</head>

<body>

<div class="wrapper">

<div class="main">

<div class="header">

</div><!--end header-->

<div class="content">

<a href="index.php">На главную</a>

<?

$dblocation = "localhost"; //Имя сервера

$dbuser = "root"; //Имя пользователя

$dbpasswd = ""; //Пароль

$dbname="wet";

//Осуществляем соединение с сервером базы данных

$mysqli = new mysqli($dblocation, $dbuser, $dbpasswd, $dbname);

$sql="select fio, email, phone, price, city\_name, hotel\_name,

days from ((bid join tour on bid.id\_tour=tour.id\_tour)

join hotel on tour.id\_hotel=hotel.id\_hotel)

join city on city.id\_city=hotel.city";

$do\_get = $mysqli->query($sql);

$num\_rows = $do\_get->num\_rows;

echo "<table border=1>";

echo " <tr><th>ФИО</th><th>email</th><th>phone</th><th>price</th>

<th>city\_name</th>

<th>hotel\_name</th>

<th>days</th>

</tr>";

if($num\_rows){

while ($results = $do\_get->fetch\_array()){

echo "<tr><td>$results[0]</td><td>$results[1]</td><td>$results[2]</td><td>$results[3]</td>

<td>$results[4]</td>

<td>$results[5]</td>

<td>$results[6]</td>

</tr>";

}

}else echo "<tr><td colspan=7>Сообщения отсутствуют в базе данных.</td></tr>"

?>

</div><!--end content-->

</div><!--end main-->

</div><!--end wrapper-->

</body>

</html>