

**Министерство образования, науки и культуры
Кыргызской Республики**

**Кыргызский Национальный Университет
Институт Интеграции Международных образовательных
программ**

Курсовой проект

По курсу: «Базы Данных»

**Тема: «База Данных для сайта авиакомпании
«Кыргызстан Аба Жолдору»**



Выполнили: Похилько А.Ф. (ЭСБ-4-00), Пузанов О.В. (КИС-4-00)
Принял: Бальджиков В.В

Бишкек 2003г.

Содержание:

СОДЕРЖАНИЕ:	2
ВВЕДЕНИЕ	3
ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ МОДЕЛИ	3
ER- моделирование.....	3
Построение ER-модели по заданию.....	3
Связи между сущностями	6
СРЕДА РАЗРАБОТКИ	6
MySQL - это система управления базами данных.....	6
Технические возможности СУБД MySQL	7
PHP	7
Краткая История PHP	8
PHP и MySQL.....	8
РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ	10
Построение структуры БД	10
Используемые типы данных MySQL.....	10
Административная часть	12
Рейсы.....	12
Пользовательская часть.....	14
Представительства.....	14
Тарифы.....	14
Рейсы.....	15
ЗАКЛЮЧЕНИЕ	17
ЛИТЕРАТУРА	17

Введение

От заказчика (НАП «Кыргызстан Аба Жолдору») поступил заказ на создание базы данных, объединяющей в себе информацию о рейсах компании, тарифах перевозок и представительствах. Отличительной особенностью явилось то, что эта БД должна находиться в сети Интернет на официальном сайте «Кыргызстан Аба Жолдору» (<http://www.ka.kg/>) для предоставления посетителям ресурса возможности быстро получить информацию о рейсах компании из любой точки мира. В качестве исходных данных были даны списки рейсов, представительств и тарифов в разрозненной форме, не объединенных каким-либо единым форматом. Также усложняющим фактором явилось постоянное взаимодействие с заказчиком (который всегда прав), в результате чего требования к функциям и оформлению БД менялись порой каждый день. Это наложило свой отпечаток на оптимальность системы и логичность построения информационной модели.

В целях безопасности некоторые имена файлов и параметров изменены.

Построение информационной модели

ER- моделирование

ER-моделирование это технология проектирования баз данных, основным принципом которой является построение модели БД на основе имеющейся информации путем выделения *сущностей (Entities)* и определения *связей (Relationships)* между ними. ER-моделирование помогает разработчику БД перейти от неформального описания потребностей заказчика к формальному описанию базы данных на выбранном языке реализации. Затем на основе ER-модели строится структура таблиц и связей между ними. ER-моделирование производится на основе словесных описаний или документов описывающих информационные потребности и правила организации. Результирующая модель является графическим представлением информационных потребностей и правил задания.

Построение ER-модели по заданию

На основе исходных данных выделяются следующие сущности и их атрибуты:

Города – города мира, в которые осуществляются полеты

- *ID* – первичный ключ таблицы
- *Название* – название города.
- *Страна* – страна, в которой расположен город
- *Латинский идентификатор* – необязательный параметр, применяющийся для передачи параметров через Интернет, где русские названия передать невозможно, а числовые параметры передавать нецелесообразно.

Группы тарифов – крупные группы тарифов, объединенных общими признаками.

- *ID* – первичный ключ таблицы
- *Название* – заголовок группы тарифов

Тарифы – стоимость полета в определенный город

- *ID* – первичный ключ таблицы
- *GID* – идентификатор группы тарифов. Внешний ключ, связанный с атрибутом *ID* сущности «Группы тарифов».
- *Город* - идентификатор города. Внешний ключ, связанный с атрибутом *ID* сущности «Города».
- *Название* – название тарифа (Пенсионный, Молодежный и т.д.)

- *В одну сторону* – стоимость полета в одну сторону
- *В обе стороны* – стоимость полета в обе стороны

Представительства – информация о представительствах компании в городах.

- *ID* – первичный ключ таблицы
- *Город* - идентификатор города. Внешний ключ, связанный с атрибутом ID сущности «Города».
- *Имя Представителя* – ФИО контактного лица.
- *Название параметра* - название параметра (адрес, телефон, факс, e-mail и т.д.).
- *Значение параметра* – значение заданного параметра заданного названием (собственно адрес, номер телефона, номер факса, адрес электронной почты и т.д.).

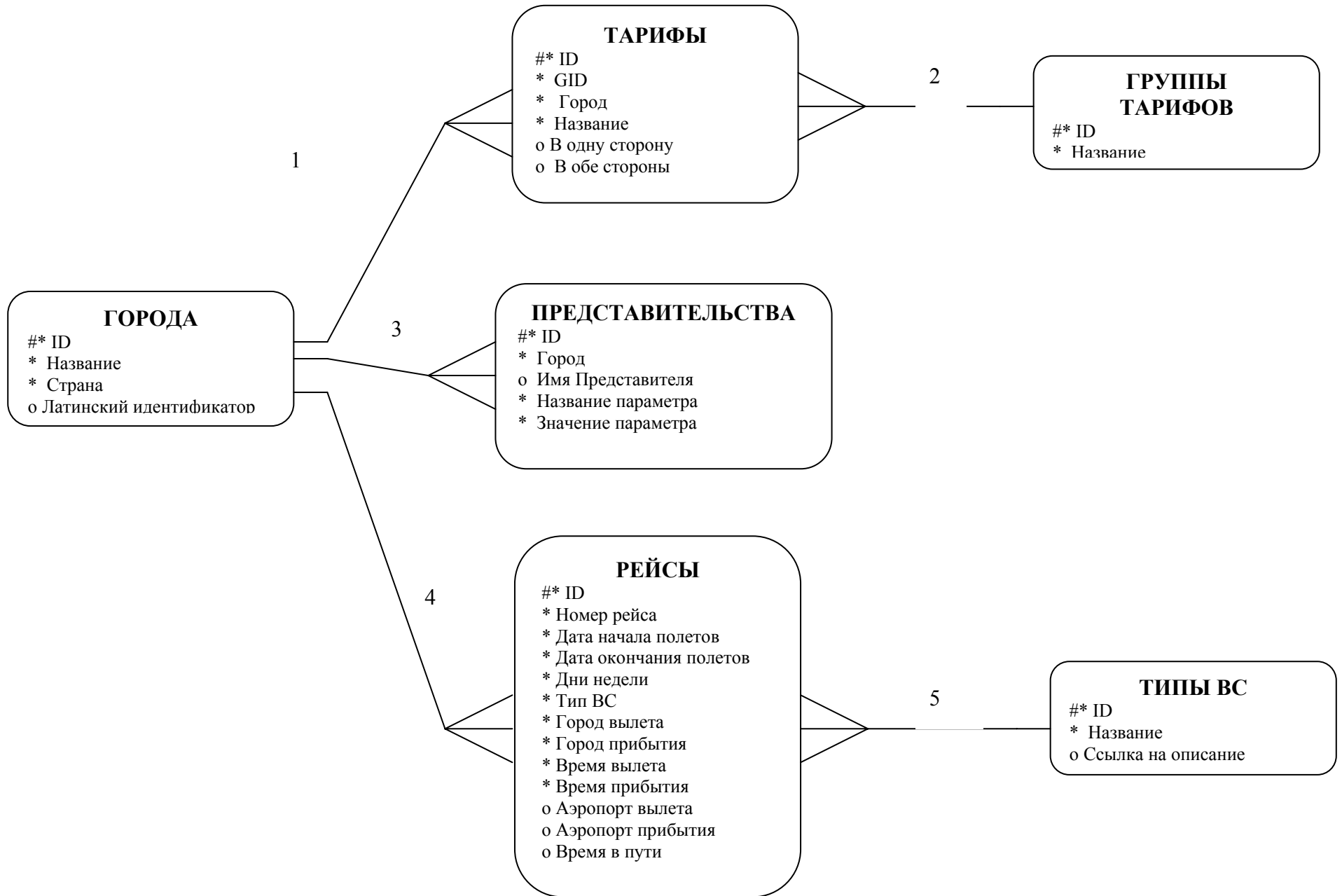
Типы ВС – типы воздушных судов, использующихся авиакомпанией.

- *ID* - первичный ключ таблицы.
- *Название* – название воздушного судна (например: ТУ-154, ЯК-40, Боинг-747) .
- *Ссылка на описание* – ссылка на статью с описанием данного воздушного судна (необязательный атрибут).

Рейсы – рейсы авиакомпании

- *ID* - первичный ключ таблицы.
- *Номер рейса* – международный номер рейса (R8 123, R8 456, R8 890, etc)
- *Дата начала полетов* – число и месяц года, когда начинается цикл полетов по этому рейсу.
- *Дата окончания полетов* - число и месяц года, когда заканчивается цикл полетов по этому рейсу.
- *Дни недели* – дни недели, по которым совершаются полеты
- *Тип ВС* - идентификатор типа ВС. Внешний ключ, связанный с атрибутом ID сущности «Типы ВС».
- *Город вылета* – идентификатор города вылета.
- *Город прибытия* - идентификатор города прибытия.
- *Время вылета* – местное время вылета.
- *Время прибытия* – местное время прибытия.
- *Аэропорт вылета* – название аэропорта, из которого происходит вылет.
- *Аэропорт прибытия* - название аэропорта прибытия.
- *Время в пути* – продолжительность полета

Ниже будет представлена полная ER-модель, со всеми сущностями и связями.



Связи между сущностями

В данной модели все связи между сущностями имеют тип «один-ко-многим».

Связь «один-ко-многим» является наиболее распространенной в реляционных базах данных, в этом случае одному экземпляру первой сущности могут соответствовать ноль или более экземпляров второй сущности, однако каждому экземпляру второй сущности соответствует не более одного экземпляра первой.

1. В каждый город билеты могут продаваться по одному или нескольким тарифам.
2. Каждому тарифу должна соответствовать одна и только одна группа тарифов.
3. Для каждого города может иметься одно или более описаний представительств.
4. В каждый город полеты могут осуществляться по одному или нескольким рейсам.
5. Каждый рейс осуществляется на одном и только одном типе ВС.

Среда разработки

MySQL - это система управления базами данных.

База данных представляет собой структурированную совокупность данных. Эти данные могут быть любыми - от простого списка предстоящих покупок до перечня экспонатов картинной галереи или огромного количества информации в корпоративной сети. Для записи, выборки и обработки данных, хранящихся в компьютерной базе данных, необходима система управления базой данных, каковой и является ПО MySQL. Поскольку компьютеры замечательно справляются с обработкой больших объемов данных, управление базами данных играет центральную роль в вычислениях. Реализовано такое управление может быть по-разному - как в виде отдельных утилит, так и в виде кода, входящего в состав других приложений.

MySQL - это система управления реляционными базами данных.

В реляционной базе данных данные хранятся не все скопом, а в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных.

В каких случаях следует отдавать предпочтение СУБД MySQL?

MySQL является очень быстрым, надежным и легким в использовании. Если вам требуются именно эти качества, попробуйте поработать с данным сервером. MySQL обладает также рядом удобных возможностей, разработанных в тесном контакте с пользователями. Сравнительные характеристики MySQL и других средств управления базами данных приведены на нашей странице тестов производительности. Первоначально сервер MySQL разрабатывался для управления большими базами данных с целью обеспечить более высокую скорость работы по сравнению с существующими на тот момент аналогами. И вот уже в течение нескольких лет данный сервер успешно используется в условиях промышленной эксплуатации с высокими требованиями. Несмотря на то, что MySQL постоянно совершенствуется, он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей доступности, скорости и безопасности MySQL очень хорошо подходит для доступа к базам данных по Internet.

Технические возможности СУБД MySQL

ПО MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API). Также сервер MySQL поставляется в виде многопоточной библиотеки, которую можно подключить к пользовательскому приложению и получить компактный, более быстрый и легкий в управлении продукт.

Доступно также большое количество программного обеспечения MySQL, разработанного сторонними разработчиками. Вполне возможно, что СУБД MySQL уже поддерживается вашим любимым приложением или языком.

MySQL правильно произносится как "Май Эс Кью Эль" (а не "майсиквел"), хотя никто не запрещает вам произносить эту аббревиатуру как "майсиквел" или еще каким-либо образом.

```
mysql> select * from apc.categories;
+----+-----+
| ID | name                |
+----+-----+
|  1 | Personal Pages     |
|  2 | E-courses          |
|  3 | Science            |
|  4 | Commercial         |
|  5 | Government         |
|  6 | Entertainment, Games |
|  7 | Computers and Internet |
|  8 | Health, Sport, Tourism, Recreation |
|  9 | Social             |
| 10 | Music & Movie      |
| 11 | Technology & Technique |
| 12 | Education          |
| 13 | GIS                |
| 14 | School             |
+----+-----+
14 rows in set (0.04 sec)

mysql>
```

Рис 1 Пример работы с MySQL через командную строку

Сам по себе MySQL является консольным приложением. То есть работает в режиме «запрос - ответ» через командную строку (CLI – command line interface). Для удобства MySQL чаще всего применяют как надстройку (модуль) к среде разработки и работают с этой СУБД через API (application program interface). Одной из этих сред является язык серверных сценариев PHP (PHP - Hypertext Preprocessor).

PHP

PHP - язык описания сценариев, выполняемых на сервере, встроенный в HTML.

В результате работы сценария, написанного на PHP, генерируется веб-страница, которая посылается пользователю, запросившему ее через Интернет.

Для разработки сценариев на PHP человечество еще не придумало мастеров (wizards) быстрого и не требующего специальных знаний в области программирования создания скриптов. Следовательно, программисту необходимо собственноручно писать код сценария, имея при этом полный контроль над программой, однако, неся ответственность за корректность ее работы.

Все сценарии набираются в любом текстовом редакторе, некоторые из которых специализированы для этой цели и имеют дополнительные функции, облегчающие работу программиста (расцветка кода, «умная» табуляция, списки подстановки и т.д.)

```
function daymonth($month,$day)
{
    return "'2003-$month-$day'";
}

function cities($var)
{
    $res=mysql_query("select ID,RUname,ENname from tblCities order by RUname;");
    while($tmp=mysql_fetch_row($res))
    {
        echo "<option ".((($tmp[0]==$var)?"selected":"")." value='$tmp[0]'>".$tmp[0]
    }
}

function days($var)
{
    for ($n=1;$n<32;$n++) {echo "<option ".((round($n)==round($var))?<option "selected":
}
```

Рис.2 PHP-сценарий в редакторе FAR-manager

Краткая История PHP

PHP был задуман где-то в конце 1994 года Расмусом Ледорфом(Rasmus Lerdorf). Ранние невыпущенные версии использовались на его домашней странице для того, чтобы следить за тем, кто просматривал его интерактивное резюме. Первая используемая версия стала доступна где-то в начале 1995 и была известна как Personal Home Page Tools. Она состояла из очень упрощенного движка синтаксического анализатора, который понимал только несколько специальных макрокоманд и ряд утилит, которые затем были в общем использовании на домашних страницах. Гостевые книги, счетчики и некоторые другие дополнения.

Довольно трудно дать какую-либо жесткую статистику, но отмечено, что к 1996 г. PHP/FI был использован по крайней мере на 15,000 веб-сайтах во всем мире. В середине 1997г. эта цифра выросла до более чем 50,000. В середине 1997г. также наблюдалось изменение в разработке PHP. Из частного любимого проекта Расмуса, которому способствовала горстка людей, это превратилось в намного более организованную рабочую группу. Синтаксический анализатор был заново переписан Зевом Сураски(Zeev Suraski) и Анди Гутмансом(Andi Gutmans), и этот новый синтаксический анализатор стал основой для PHP Версии 3.

Сегодня (в середине -1998г.) как PHP/FI так и PHP3 поставляется с рядом коммерческих продуктов типа C2's StrongHold web server и RedHat Linux и консервативной оценкой, основанной на экстраполяции чисел, предоставленных NetCraft было бы то, что PHP используется на 150,000 сайтах во всем мире. В перспективе, их больше чем сайтов, запущенных на Netscape's flagship Enterprise server в Интернете.

PHP и MySQL

Возможно самая сильная и значимая возможность в PHP - уровень интеграции с базами данных. Написание веб-страницы работающей с базой данных невероятно проста. В настоящее время поддерживаются следующие базы данных:

Oracle	
	Adabas D
Sybase	
	FilePro
mSQL	
	Velocis
MySQL	
	Informix
Solid	
	dBase
ODBC	
	Unix dbm
PostgreSQL	

Для взаимодействия PHP с MySQL используется набор встроенных функций, таких как:

- *mysql_connect()* – подключение к серверу MySQL с заданным именем пользователя и паролем;
- *mysql_select_db()* – (аналог команды «USE» командной строки сервера MySQL) выбор базы данных с которой будет в дальнейшем вестись работа.
- *mysql_query()* – посылка запроса серверу MySQL. Функция возвращает указатель на область памяти, куда загружен результат запроса.
- *mysql_num_rows* – возвращает количество строк в результате запроса.
- *mysql_fetch_array()* – функция возвращает очередную строку из результата функции *mysql_query()*;
- *mysql_close()* – завершение работы с сервером MySQL

Реализация базы данных

Интерфейс для работы с БД делится на две части – пользовательскую и административную.

Все формы создаются программистом средствами языка HTML и PHP динамически.

Пользовательская часть не имеет функций добавления и изменения данных и ориентирована на удобный и понятный просмотр и поиск информации.

Административная часть ориентирована на управление данными.

Построение структуры БД

Используемые типы данных MySQL

Tinyint – Целое число (1 байт).

Int - Целое (4 байта).

Varchar – Строка переменной длины (1 <= NUM <= 255)

Date - Сохраняет информацию о дате. Использует формат "YYYY-MM-DD".

Может модифицироваться как строка или число, хотя Вы, вероятно, используете контекст строки для времени и даты. MySQL тип DATE понимает по крайней мере следующие синтаксис.

- YYYY-MM-DD (Обратите внимание что '-' может фактически быть ЛЮБОЙ не цифрой)
- YY-MM-DD (Обратите внимание что '-' может фактически быть ЛЮБОЙ не цифрой)
- YYMMDD
- YYMM

Диапазон для этого типа данных от 0000-00-00 до 9999-12-31. Так что "проблема 2000" здесь не стоит. В отличие от TIMESTAMP, DATE принимает годы и в виде двух цифр от 0000 до 0099. Это не очень полезно в большинстве случаев. Используйте задание лет четырьмя цифрами в полях типа DATE. Тип DATE имеет длину 4 байта.

Запрос на создание таблицы для сущности «Тип ВС»

```
create table tblVSTypes
  (ID tinyint unsigned AUTO_INCREMENT primary key,
  RUname varchar(100),
  link varchar(100));
```

Запрос на создание таблицы для сущности «Города»

```
create table tblCities
  (ID tinyint unsigned AUTO_INCREMENT primary key,
  state varchar(100),
  RUname varchar(100),
  tarif varchar(100));
```

Запрос на создание таблицы для сущности «Рейсы»

```
create table tblRaces
  (ID int unsigned AUTO_INCREMENT primary key,
   race varchar(10),
   dateS date,
   dateF date,
   days varchar(20),
   VSType tinyint unsigned,
   cityS int(2),
   cityF int(2),
   timeS varchar(8),
   timeF varchar(8),
   portS varchar(100),
   portF varchar(100),
   length varchar(10),
   dayflag tinyint,
   singleflag tinyint);
```

Запрос на создание таблицы для сущности «Группы тарифов»

```
create table tblTarifGroups
  (ID int unsigned AUTO_INCREMENT primary key,
   title varchar(100));
```

Запрос на создание таблицы для сущности «Тарифы»

```
create table tblTarifs
  (ID int unsigned AUTO_INCREMENT primary key,
   GID tinyint,
   city int(2),
   title varchar(100),
   oneway int(2),
   twoway int(2));
```

Запрос на создание таблицы для сущности «Представительства»

```
create table tblRepres
  (ID int unsigned AUTO_INCREMENT primary key,
   city int(2),
   name varchar(100),
   attr varchar(100),
   value varchar(255));
```

По специальному параметру START административная часть интерфейса БД поочередно исполняет эти запросы через функцию `mysql_query()` и строит структуру таблиц. Также в таблицу `tblTarifGroups` автоматически вставляются три записи соответствующие стандартным группам тарифов у НАП.

```
mysql_query("insert into tblTarifGroups(title)
  values('Тариф зарубежный');");
mysql_query("insert into tblTarifGroups(title)
  values('Тарифы по СНГ');");
mysql_query("insert into tblTarifGroups(title)
  values('Тариф местный');");
```

Административная часть

Администратору предоставлены возможности создания, модификации и удаления информации по городам. Форма добавления городов содержит необходимые поля и используется как при добавлении города, так и при модификации ранее добавленных записей.

[View Races](#) | [Cities](#) | [Planes](#) | [AddRace](#) | [Tarifs](#) | [Representati](#)

Состояние: **Добавление городов**

Страна:

Город по-русски:

Город по-английски:

ID name:

- 1. Бишкек - Кыргызстан(bishkek) [Изменить](#) | [delete](#)
- 2. Ганновер - Германия(gannover) [Изменить](#) | [delete](#)
- 3. Франкфурт - Германия(frankfurt) [Изменить](#) | [delete](#)
- 4. Штуттгарт - Германия(shtuttgard) [Изменить](#) | [delete](#)
- 5. Лондон - Англия(london) [Изменить](#) | [delete](#)
- 6. Дели - Индия(deli) [Изменить](#) | [delete](#)

Рис. 3 Форма городов

Для удаления, добавления или модификации используется одна функция PHP, а выполняемое действие задается передаваемыми ей параметрами. Передача параметров, описывающих действия, производится через гиперссылки HTML.

После вывода на экран формы распечатывается информация о существующих городах:

```
$res=mysql_query("select * from tblCities order by ID;");//запрос на
выборку всех городов
while($tmp=mysql_fetch_array($res)) //последовательно выбрать все записи
из результата
{
    echo "<li>$tmp[0]. ".$tmp['RUname'].
        " - ".$tmp['state']. "(".$tmp['tarif'].") <a
href=./db.php?mode=city&act=modify&ID=".$tmp[0]."> Изменить </a> |
<a href=./db.php?mode=city&act=delete&ID=".$tmp[0].">delete
</a><br>";// распечатка строки результата, с соответствующими
ссылками на модификацию и удаление
}
```

Для остальных таблиц используется практически точная копия функции городов, изменения лишь в названиях таблиц и атрибутов в запросах, и в форме ввода/изменения.

Рейсы

Форма рейсов имеет существенные отличия от предыдущих, так как в ней используются не только числовые, скрытые поля и выпадающие списки, но и группа флажков. Для организации выпадающих списков работает несколько функций, которые делают выборку из соответствующей таблицы и формируют выпадающий список на языке HTML.

Состояние: **Добавление рейсов**

№ Рейса	<input type="text" value="R8"/>
Маршрут	1) Бишкек / <input type="text"/> - 1) Бишкек / <input type="text"/>
Период	с 1 <input type="text"/> Января <input type="text"/> По 1 <input type="text"/> Января <input type="text"/>
Дни отправления	<input type="checkbox"/> Понедельник <input type="checkbox"/> Вторник <input type="checkbox"/> Среда <input type="checkbox"/> Четверг <input type="checkbox"/> Пятница <input type="checkbox"/> Суббота <input type="checkbox"/> Воскресенье
Время отпр	<input type="text"/>
Время приб	<input type="text"/>
Аэропорт отпр	<input type="text"/>
Аэропорт приб	<input type="text"/>
Время в пути	<input type="text"/>
Тип ВС	Ту-154 <input type="text"/>
<input type="button" value="Add"/>	

Рис. 4. Форма рейсов

Пример: выпадающий список для городов

```
function cities($var)
{
    $res=mysql_query("select ID,RUname,ENname from tblCities;");
    while($tmp=mysql_fetch_row($res))
    {
        echo "<option ".((($tmp[0]==$var)?"selected":""))."
value=' $tmp[0]'>$tmp[0]) ". $tmp[1]." / ". $tmp[2];
    }
}
```

Рейсы не выводятся после формы добавления (из-за их большого количества), а выдаются (с кнопками модификации и удаления) по нажатию на соответствующую ссылку

Состояние: **Добавление**

40	R8 111	2003- 03-30	2003- 10-25	1,2,3,4,5,6,7	8	Ош	Бишкек	8:30	9:30	Ош	Манас		modify	delete
39	R8 112	2003- 03-30	2003- 10-25	1,2,3,4,5,6,7	8	Бишкек	Ош	11:20	12:25	Манас	Ош		modify	delete
44	R8 117	2003- 04-05	2003- 10-25	6	8	Ош	Бишкек	15:00	16:00	Ош	Манас	1 час	modify	delete
43	R8 118	2003- 04-05	2003- 10-25	6	8	Бишкек	Ош	17:20	18:25	Манас	Ош		modify	delete
42	R8 119	2003- 04-01	2003- 10-21	2	8	Ош	Бишкек	7:45	8:45	Ош	Манас	1 час	modify	delete
41	R8 120	2003- 04-01	2003- 10-21	2	8	Бишкек	Ош	18:30	19:35	Манас	Ош		modify	delete
56	R8 121	2003- 05-14	2003- 10-25	3	8	Ош	Бишкек	14:30	15:30	Ош	Манас	1 час	modify	delete
55	R8 122	2003- 05-14	2003- 10-25	4	8	Бишкек	Ош	20:00	21:05	Манас	Ош	1 час	modify	delete
46	R8 143	2003- 03-30	2003- 05-04	1,3,4,5,7	8	Ош	Бишкек	15:30	16:30	Ош	Манас	1 час	modify	delete
48	R8 143	2003- 05-05	2003- 10-25	1,5,7	8	Ош	Бишкек	15:30	16:30	Ош	Манас	1 час	modify	delete
45	R8 144	2003- 03-30	2003- 05-04	1,3,4,5,7	8	Бишкек	Ош	18:00	19:05	Манас	Ош	1 час	modify	delete
47	R8 144	2003- 05-05	2003- 10-25	1,5,7	8	Бишкек	Ош	18:00	19:05	Манас	Ош	1 час	modify	delete

Рис 5. Список рейсов

Пользовательская часть

Представительства

По адресу <http://www.ka.kg/ru/inquiry/representatives/> доступна информация по всем представительствам НАП за рубежом. Для генерации этой страницы делаются последовательно запросы на информацию о представительстве для каждого города (\$gorod):

```
select * from tblRepres
where city=$gorod
order by name desc;
```

По результату запроса строится таблица, в которой в левом столбце перечислены параметры представительства, а в правом их значения. Причем ФИО представителя (указанное для каждого города лишь раз) по правилу сортировки будет в первой записи.

Ганновер - Германия

Акматов Эрик Бекболотович	Тел	+49511 7261734
	Моб. тел.	+49 175 2067179
	Факс	+49511 7261959
	Моб. факс	+49 175 2079961
	E-mail:	kyrgyzair@t-online.de
	E-mail	hannover@kga-r8.de
	Адрес	Kyrgyzstan Airlines, Konrad-Adenauer-Strasse 80, D-30853 Langenhagen/Hannover
	Сайт Генерального Представительства в Германии	www.kga-r8.de

Рис 6. Таблица вывода представительства

Тарифы

По адресу <http://www.ka.kg/ru/tarif/> находится информация о стоимости всех полетов рейсами авиакомпании. Для удобства пользователей создана навигация внутри страницы по «якорям», позволяющая перемещаться на группы тарифов или сразу на искомый город.

Информация о тарифах выбирается из таблицы tblTarifs по городам и выводится таблицами, снабженными заголовками.

Тариф зарубежный, Тарифы по СНГ, Тариф местный

Дели, Душанбе, Екатеринбург, Франкфурт, Ганновер, Лондон, Москва,
Новосибирск, Омск, Джалал-Абад, Ош, Шарджа, Штуттгарт, Ташкент,
Урумчи

Тариф зарубежный

Лондон - Англия

Тариф	В одну сторону, цена \$ США	В обе стороны, цена \$ США
Эконом класс на 1 месяц	530	779
Эконом класс на 3 месяца	622	916
Эконом класс на 12 месяцев	747	1099
Бизнес класс	2412	3548

Рис 7. Локальная навигация и таблица тарифов

Рейсы

Полный перечень рейсов НАП доступен по адресу <http://www.ka.kg/ru/timetable/> где все рейсы выбираются из базы и выдаются одной большой таблицей.

Для удобства пользователей на главной странице, и странице расписания размещены формы поиска по рейсам авиакомпании. Пользователь указывает города отправления и прибытия, одно- или двунаправленность билета, а также дату желаемого вылета и запрашивает поиск. Форма организована так, что при выборе режима «в одну сторону» выбор даты отправления обратно становится неактивным. Также дата вылета «туда» ставится по умолчанию текущей, а «обратно» - через неделю. Для визуального разнообразия город прибытия по умолчанию выбирается случайным образом.

В результате поиска выдаются все рейсы в радиусе двух недель от запрашиваемой даты (даты в прошлом, естественно, не учитываются), стоимость полета и информация о представительстве.

Города отправления и прибытия

Откуда Бишкек **Туда** 20 Сентября

Куда Франкфурт **Обратно** 7 Января

Только туда Туда и обратно

Расписание с 30.03.2003 г. по 25.10.2003 г.

Туда

Направление: **Бишкек – Франкфурт (Рейс R8 571)**

	Города	Дата	Время	Аэропорт
Отправка	Бишкек	26.09	14:05	Манас
Прибытие	Франкфурт	26.09	16:45	Франкфурт

Тип самолета: Ту-154

Время во всех аэропортах указано местное

Рис 8. Результат поиска

Запрос на поиск формируется динамически на основе указанных пользователем данных:

```
$w="where (cityS=\"\".$q['cityS'].\"\" AND cityF=\"\".$q['cityF'].\"\" AND ( (date_sub($dateS,interval 7 day) between dates AND dateF) or ($dateS between dates AND dateF) or (date_add($dateS,interval 7 day) between dates AND dateF))) ";
```

Здесь \$w – условие для поиска, которое передается в функцию вывода записей по рейсам из БД.

Заключение

Данная база данных была успешно сдана заказчику в срок и на данный момент активно используется посетителями сайта (<http://www.ka.kg>).

В процессе разработки программного продукта нами были решены многие локальные проблемы связанные с работой сайта и взаимодействием языка скриптов PHP с СУБД MySQL . Так же мы существенно повысили свой профессиональный уровень, наработали методы и навыки, которые на сей день активно применяем в текущих проектах, таких как сайт компании «Континент», сайт телерадиокомпании «КООРТ» и т. д.

Литература

1. Курс лекций по «Банкам данных»- В.В. Бальджиков.
2. Руководство разработчика PHP (<http://www.php.net>).
3. Руководство разработчика My_SQL (<http://www.mysql.ru>).