

УДК 334.02:330.47

ОПТИМИЗАЦИЯ РАБОТЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПРЕДПРИЯТИЯ МАЛОГО БИЗНЕСА ПО ОКАЗАНИЮ КОНСУЛЬТАЦИОННЫХ УСЛУГ

Одияко Н.Н., Голодная Н.Ю., Реуцкий Р.С.

*Владивостокский государственный университет экономики и сервиса, Владивосток,
e-mail: odiako@yandex.ru, xas56@mail.ru, Roman.Reutskiy@vvsu.ru*

В развитии технологии Интернет/интранет основной акцент пока что делается на разработке инструментальных программных средств. В то же время наблюдается отсутствие развитых средств разработки приложений, работающих с базами данных. Компромиссным решением для создания удобных и простых в использовании и сопровождении информационных систем, эффективно работающих с базами данных, стало объединение Интернет/интранет-технологии с многоуровневой архитектурой. Внедрение информационной системы позволяет предприятию повысить эффективность деятельности работников, автоматизировать процессы, что позволит сократить время выполнения работником операций, которые до этого выполнялись вручную, уменьшить время, затрачиваемое менеджером на поиск информации или оповещение клиента. Появляется возможность структурированного хранения информации. Анализ этой информации позволяет делать прогнозы, которые, в свою очередь, дают возможность ставить перед компанией цели в краткосрочной и долгосрочной перспективах. Целью данной работы является создание адаптированной под предприятие информационной системы по работе с клиентами. Во время работы были рассмотрены все виды моделей баз данных, и из всех видов была выбрана та, которая использует организацию данных в виде двумерных таблиц (реляционная модель). Для решения задачи по созданию базы данных был использован локальный сервер Denver и язык SQL. С помощью языков HTML, PHP, JavaScript были решены задачи по созданию интерфейса для управления базой данных, а также для оформления информационной системы.

Ключевые слова: база данных, интерфейс взаимодействия, языки программирования, оптимизация работы

OPTIMIZATION OF THE ENTERPRISE INFORMATION SYSTEM SMALL BUSINESS ADVISORY SERVICES

Odiako N.N., Golodnaya N.Y., Reutskiy R.S.

*Vladivostok State University Economics and Service, Vladivostok, e-mail: odiako@yandex.ru,
xas56@mail.ru, Roman.Reutskiy@vvsu.ru*

In the development of technologies Internet/Intranet main focus so far is on the development of software tools. At the same time there is a lack of developed application development tools, working with databases. A compromise solution for the creation of user-friendly and easy to use and maintenance of information systems, effectively working with databases, was the consolidation of Internet / intranet architecture with multilevel technology. The implementation of the information system allows the company to increase the employee performance, automate processes to reduce execution time employee of the operations that were performed manually to reduce the time it takes a Manager to seek information or alert the client. You receive the opportunity of a structured information storage. Analysis of this information allows us to make predictions, which in turn provide an opportunity to set the company objectives in the short and long term. The aim of this work is to create adapted to the enterprise information system to work with clients. During operation, all types of database models were considered, and all kinds of was chosen the one that uses a data organization in the form of two-dimensional tables (relational model). Denver and the SQL language local server was used to solve the problem on the database. With languages HTML, PHP, JavaScript task of creating an interface for database management have been solved, as well as for the design of information systems.

Keywords: database, interface, programming languages, optimization

В настоящее время информация становится одним из наиболее весомых и ценных продуктов человеческой деятельности. Эффективность работы предприятия в значительной степени зависит от наличия соответствующей информации, методологии ее использования и защиты информационных систем.

Автономная некоммерческая организация Дальневосточный визовый центр «ДальВиза» является не имеющей членства некоммерческой организацией, учрежденной на основе добровольных имущественных взносов, преследующей общественно

полезные цели. Целью организации является укрепление межкультурных связей посредством предоставления консультационных услуг по оформлению различных виз (туристской, учебной, эмиграционной) и проведению различных международных образовательных программ. Всего существует восемь видов программ:

- иммиграция в Канаду;
- визы в Канаду;
- обучение в Канаде;
- рабочие визы в Канаду;
- иммиграция в провинцию Манитоба;
- иммиграция в провинцию Онтарио;

- визы в США;
- визы в другие страны.

Корпоративные информационные системы являются развитием систем для рабочих групп, они направлены на крупные компании и могут поддерживать территориально разнесенные узлы или сети. В основном, они имеют иерархическую структуру из нескольких уровней. Для таких систем характерна архитектура клиент-сервер со специализацией серверов или же многоуровневая архитектура [1, 2]. При разработке таких систем могут использоваться те же серверы баз данных, что и при разработке групповых информационных систем. Однако в крупных информационных системах наибольшее распространение получили серверы Oracle, DB2 и Microsoft SQL Server [3].

Для групповых и корпоративных систем значительно повышаются требования к надежности функционирования и сохранности данных [6]. Эти свойства обеспечиваются поддержкой целостности данных, ссылок и транзакций в серверах баз данных.

Важными требованиями для них являются следующие:

- высокая производительность обработки транзакций;
- доставка информации при удаленном доступе к БД по телекоммуникациям.

Объектами разработки в файл-серверной системе являются компоненты приложения, определяющие логику диалога PL, а также логику обработки BL и управления данными DL. Разработанное приложение реализуется либо в виде законченного загрузочного модуля, либо в виде специального кода для интерпретации.

Однако такая архитектура имеет существенный недостаток: при выполнении некоторых запросов в базе данных клиенту могут передаваться большие объемы информации, загружающие сеть и приводящие к непредсказуемости времени реакции. Значительный сетевой трафик особенно сильно сказывается при организации удаленного доступа к базам данных на файл-сервере через низкоскоростные каналы связи. Одним из вариантов устранения данного недостатка является удаленное управление файл-серверным приложением в сети. При этом в локальной сети размещается сервер приложений, совмещенный с сервером (обычно называемым сервером доступа), в среде которого выполняются обычные файл-серверные приложения. Особенность состоит в том, что диалоговый ввод-вывод поступает от удаленных клиентов через телекоммуникации. Приложения не должны быть слишком сложными, иначе велика ве-

роятность перегрузки сервера, или же нужна очень мощная платформа для сервера приложений [10].

Информационная система должна соответствовать требованиям гибкости, надежности, эффективности и безопасности.

Иерархическая модель данных представляет собой совокупность элементов данных, расположенных в порядке их подчинения и образующих по структуре перевернутое дерево. К основным понятиям иерархической модели данных относятся: уровень, узел и связь [4].

Иерархическая модель данных имеет некоторые недостатки:

- ограничения в организации отношений между сущностями. Иерархическая модель позволяет организовать последовательную связь «один ко многим» между данными, но не в состоянии реализовать отношения «многие ко многим»;
- структурная зависимость. Иерархическая структура предполагает, что физические данные также станут храниться в виде дерева;
- сложность разработки программного обеспечения. Разработчик программ должен знать особенности физического хранения данных, иначе он мог просто заблудиться в запутанной системе указателей.

Ко всему прочему иерархическая модель не была стандартизирована. Как следствие, всегда существовала проблема переносимости данных между приложениями различных разработчиков.

Сетевая модель данных основана на тех же основных понятиях (уровень, узел, связь), что и иерархическая модель, но в сетевой модели каждый узел может быть связан с любым другим узлом.

По сравнению с иерархической, сетевая модель имеет одно серьезное преимущество: она позволяет назначить произвольное количество связей между узлами графа. Поэтому можно создавать базы данных, более точно отражающие связи реального мира, в частности, в сетевых БД без особого труда можно формировать отношения «многие ко многим» или замыкать связь узла на себя самого [4].

К сожалению, сетевая модель также не свободна от недостатков, к которым относятся:

- большое количество произвольных связей повышает сложность схемы БД и, как следствие, вызывает дополнительные трудности при обеспечении целостности данных;
- сложность разработки прикладного программного обеспечения.

Реляционная модель данных использует организацию данных в виде двумерных таблиц.

Реляционная модель данных далеко не идеальна. По существу, это компромиссное решение между потребностью отражать сущности реального мира и связи между ними и ограниченными возможностями математической теории множеств, переложенной на программный код.

Основными объектами любой базы данных являются таблицы. Таблицы базы данных создаются таким образом, чтобы каждая из них содержала информацию об одном информационном объекте. Между таблицами должны быть установлены реляционные связи. Установка таких связей делает возможным выполнение одновременной обработки данных из нескольких таблиц. Первичный ключ реляционной таблицы – это поле или группа полей.

Различают персональные и многопользовательские Системы управления базами данных (СУБД). Основополагающие функции СУБД: доступность данных; управление параллельностью; транзакции; восстановление данных; контроль за доступом к данным.

Управление параллельностью – реализация механизма для многопользовательского доступа к обрабатываемым данным с гарантией корректного обновления этих данных. Умение предоставить нескольким пользователям одновременный доступ к разделяемым ресурсам – едва ли не самая сложная задача, решаемая СУБД. СУБД должна суметь избежать конфликта совместного доступа двух или большего числа пользователей к одним и тем же строкам таблицы или, по крайней мере, исключить какие-либо нежелательные последствия при возникновении конфликта.

Все операторы транзакции должны быть выполнены корректно и полностью, только тогда в БД будут зафиксированы изменения, которые вводил пользователь. В противном случае система осуществляет автоматический откат транзакции, т. е. состояние БД будет восстановлено на момент времени, предшествующий вызову транзакции в базе данных системы [9].

Все данные, хранимые в БД, должны быть корректными и непротиворечивыми. Это означает, что данные в таблицах могут модифицироваться только в соответствии с установленными правилами.

В самом общем случае можно говорить о существовании трех правил поддержания целостности данных: целостность доменов, целостность отношений, целостность связей между отношениями. Кроме того, разработчик имеет возможность описывать свои собственные бизнес-правила, которые мы будем называть корпоративными ограничениями.

В случае непредвиденных ошибок и сбоев, приведших к повреждению или разрушению данных, СУБД должна обладать возможностью восстанавливать пострадавшие данные. В первую очередь эта функция реализуется с помощью процедур резервного копирования. СУБД обязана поддерживать современные сетевые технологии и предоставлять доступ к БД удаленных персональных компьютеров [11].

Программисты совместно с администратором работают над физическим созданием проекта БД. Но прикладных программистов в большей степени интересует не сама концепция проекта БД, а способы донесения этой концепции до конечного пользователя. Поэтому область интересов программистов смещена в сторону разработки клиентских приложений и отчетов. Основным инструментом прикладного программиста выступают многочисленные среды проектирования, как правило, 4-го поколения (4 Generation Level, 4GL). В первую очередь это программные пакеты Embarcadero RAD Studio (в состав которого и входит язык Delphi) и Microsoft Visual Studio. Кроме того, некоторые СУБД (например, Microsoft Access и FoxPro) обладают встроенными средствами проектирования, но обычно возможности таких средств существенно ограничены.

Основным потребителем услуг СУБД выступает обычный пользователь: в его интересах создаются БД и прикладное программное обеспечение. Основным средством общения между людьми и базой данных выступает структурированный язык запросов SQL. Поняв, что угрозы целостности нет, процессор передает команду оптимизатору запросов. Задача оптимизатора состоит в поиске наиболее эффективного способа выполнения поступивших команд. Наконец, оптимизированная команда компилируется и передается во власть системы управления транзакциями.

Для создания информационной системы рассмотрено несколько видов локальных серверов, из которых выбран один, отвечающий требованиям системы. Также определены языки программирования, с помощью которых будет создаваться база и интерфейс.

OpenServer – это, пожалуй, наилучший комплект на сегодняшний день для создания и тестирования локальных сайтов под Windows. Удобное меню, система запуска, простое создание виртуальных хостов, внимание к мелочам позволяют легко работать начинающим веб-разработчикам, а входящие в комплект компоненты устроят и профессиональных программистов.

Включает в себя Apache, nginx, PHP, MySQL, phpMyAdmin, FTP-сервер FileZilla. OpenServer занимает практически 50 Мб, что раза в три превосходит объём других комплектов, при этом он содержит довольно внушительный набор компонент. Для удобства имеется русская версия.

Несмотря на практичность и комфортность работы, комплект имеет и некоторые недостатки. Он запускается только под Windows XP SP3, Windows Vista или Windows 7 с административными правами. Также может возникнуть несовместимость с другими установленными программами, которые потребуют отдельного отлаживания [11].

Для выборки записей из базы данных разработан специализированный язык – SQL (Structured Query Language, структурированный язык запросов). С помощью этого языка можно создавать базы данных и таблицы, добавлять, изменять и удалять данные, получать данные по запросу.

Сам по себе SQL не является Тьюринг-полным языком программирования, но его стандарт позволяет создавать для него процедурные расширения, которые расширяют его функциональность до полноценного языка программирования.

С начала работы у организации для каждого клиента создавался отдельный Word файл, в котором и хранились все данные. Сохранение данных клиента занимает очень много времени, а поиск клиента сводится к обычному поиску по алфавиту. Также становился трудным и долгим процесс создания какой-либо статистики.

Оптимальным вариантом решения данной проблемы явилось создание информационной системы по работе с клиентами. Менеджер сможет быстро вносить новых клиентов, искать уже внесенных, вносить изменения в данные клиентов [7].

Данное программное средство должно выполнять следующие функции:

- внесение данных клиента;
- редактирование данных клиента;
- быстрый поиск нужного клиента или просмотр всех;
- внесение данных о заказе услуги;
- редактирование данных заказа;
- быстрый поиск нужного заказа;
- возможность отправлять e-mail;
- обеспечение возможности в параметре «секторы экономики» выбрать как одно, так и несколько значений, вплоть до всех имеющих значения.

Программное средство должно работать с базой данных в соответствии с алгоритмом функционирования.

Для предотвращения некорректной работы программы необходимо было реализовать:

- проверку о заполнении всех полей при добавлении нового клиента;
- проверку о заполнении всех полей при добавлении нового заказа;
- оповещение о действиях, которые совершает система при нажатии пользователем на кнопки;
- вывод сообщений об ошибках;
- проверку формата и размера загружаемой картинки.

Данная программа должна работать под операционной системой Windows 7 (32-bit)/7 (64-bit) и выше.

Контроль и приемка разработки осуществляется на основе испытаний контрольно-отладочных примеров. При этом проверялось выполнение всех функций программы.

Самым простым является создание базы данных с использованием phpMyAdmin. Был создан интуитивно понятный интерфейс в совокупности с широкой функциональностью.

phpMyAdmin – веб-приложение, которое написали на языке PHP. Представляет собой веб-интерфейс для администрирования СУБД MySQL. phpMyAdmin позволяет через браузер осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных.

phpMyAdmin – веб-приложение, которое написали на языке PHP. Представляет собой веб-интерфейс для администрирования СУБД MySQL. phpMyAdmin позволяет через браузер осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных [11].

База данных была создана с помощью языка SQL. Все таблицы создавались SQL запросами. На рис. 1 представлены основные SQL запросы, используемые для создания таблиц данных.

С помощью языков HTML, PHP, JavaScript была создана оболочка для управления базой данных. Для оформления оболочки использовались те же самые языки. Программирование производилось в среде Notepad++.

Информационная система выглядит как база данных, созданная в веб-приложении phpMyAdmin, и интерфейс, написанный в редакторе текстовых файлов Notepad++, с помощью языка разметки HTML, языка программирования PHP с элементами JavaScript и специализированным языком SQL. Все четыре таблицы создавались с помощью SQL запросов, поэтому перенос базы с компьютера, где база разрабатывалась, на компьютер, на котором будет использоваться, не составляет труда. Достаточно создать

базу «dalviza» и в ней ввести запросы на создание таблиц. Единственное, что необходимо будет сделать вручную, это связать таблицы «clients» и «order1» по ключевому полю «id_client».

Также необходимо внести логины и пароли менеджеров, которые будут работать

с базой. Это действие может выполнить только администратор, чтобы исключить несанкционированный доступ к интерфейсу управления и всей базе данных в целом [8].

Созданная база получила название «Дальвиза». Основные элементы этой базы данных представлены на рис. 2.

```
create table clients
(
  id integer (9) not null auto_increment primary key,
  surname varchar(30) not null,
  name varchar(30) not null,
  patronymic varchar(30) not null,
  dateofbirth varchar(30) not null,
  country varchar(30) not null,
  city varchar(30) not null,
  address varchar(50) not null,
  phone varchar(30) not null,
  email varchar(50) not null,
  photo varchar(30) not null
);

create table order
(
  id_order int (9) not null auto_increment key,
  program varchar(50) not null,
  agreement varchar(30) not null,
  id_client int (9) not null,
  status varchar(50) not null,
  paid varchar(30) not null,
  costof varchar(30) not null,
);

create table authentication
(
  id_m int (9) not null auto_increment key,
  name_m varchar(30) not null,
  pass_m varchar(30) not null
);

create table sessions
(
  sess_id int (9) not null auto_increment key,
  sess_name varchar(30) not null,
  sess_pass varchar(30) not null,
  date_touched varchar(50) not null
);
```

Рис. 1. Примеры SQL запросов

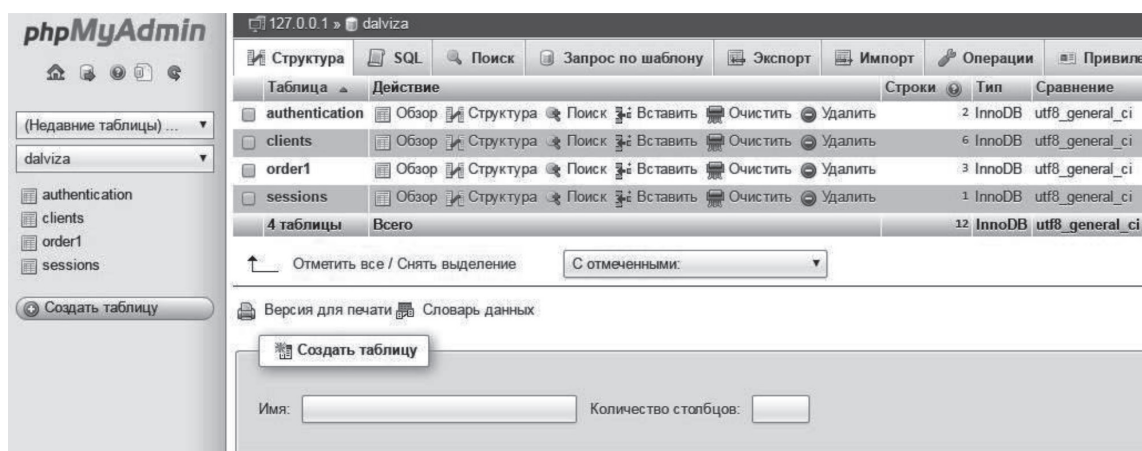


Рис. 2. База данных «Дальвиза»

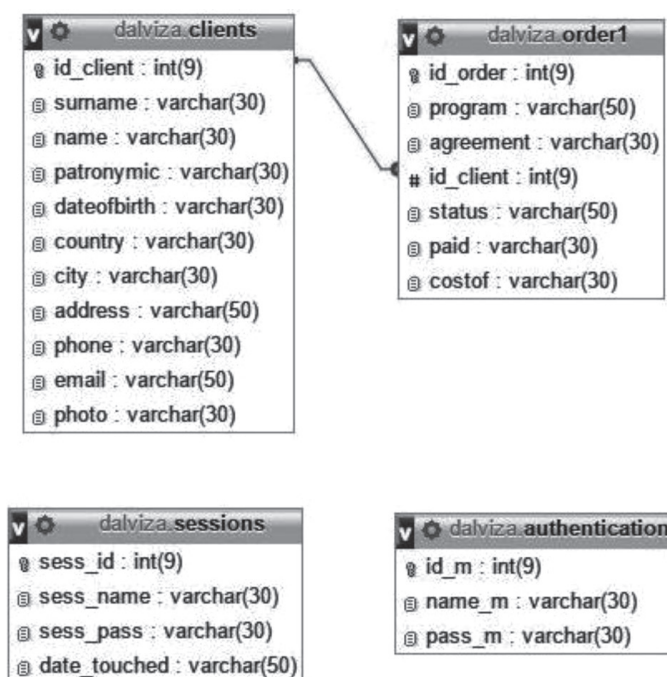


Рис. 3. Схема данных

По схеме данных видно, что в базе четыре таблицы «clients», «order1», «sessions» и «authentication», все четыре таблицы изображены рис. 3. Поле «id_client» является уникальным ключом в таблице «clients», в таблице «order1» уникальный ключ – это поле «id_order». Связь между двумя этими таблицами построена на основе поля «id_client» и называется «один ко многим». Суть связи заключается в том, чтобы присоединять разным заказам людей, которые уже когда-то пользовались услугами компании.

Таблица «clients» хранит в себе все данные о клиенте, приходящем в компанию. Таблица «order1» содержит данные о заказе клиентов, в «sessions» заносятся данные о том, под каким логином и паролем и когда происходил вход в систему. Таблица «authentication» хранит логины и пароли пользователей, которые имеют доступ. При вводе правильных логина и пароля форма вводит в таблицу «sessions» логин и пароль, введенные в поля, а также время, установленное на компьютере.

Главное меню было создано максимально простым, чтобы пользователю было легко перемещаться по системе и быстро переходить от одной формы к другой.

Поиск клиента осуществляется в отдельном окне, представленном на рис. 4. Поиск осуществляется по фамилии кли-

ента, также в выпадающем списке можно выбрать пункт «Все клиенты», который позволяет показать абсолютно всех клиентов в алфавитном порядке.

Форма добавления клиента позволяет вводить данные о клиенте в базу. Она состоит из десяти полей, которые необходимы компании для обслуживания клиента. Фотография в базу не загружается, так как это сделало бы ее очень тяжелой. Фотография хранится в отдельной папке, куда загружается автоматически, в базе хранится только ее имя и ссылки на нее.

Так как могут быть допущены ошибки, то была создана страница по редактированию данных клиента. Внешний вид страницы изображен на рис. 5.

Поиск осуществляется точно так же, как на странице поиска клиента. Чтобы исключить случаи путаницы из-за одинаковых имен и фамилий клиентов, в таблицу выводятся остальные данные. В конце каждой записи присутствуют две кнопки «Редактировать» и «Изменить фотографию».

Редактирование происходит на основе формы «Добавление клиента» с учетом того, что все данные из базы уже введены в поля. Для изменения необходимо просто ввести коррективы в нужные поля и нажать кнопку «Сохранить изменения», после этого форма автоматически обновится и выдаст уже измененные данные.

Клиент:

Поиск

Назад

№	Фамилия	Имя	Отчество	Дата рождения	Страна	Город	Адрес	Телефон	Email	Фото
1)	Иванов	Иван	Иванович	15.05.1985	Россия	Владивосток	Ладыгина 11, кв. 109	89000000000	ivan.ivanov@gmail.com	
2)	Петров	Петр	Петрович	17.08.1978	Россия	Владивосток	Ул. Адмирала Фокина 9 кв. 32а	89000000001	petr.petrov@gmail.com	

Рис. 4. Поиск клиента по фамилии

Клиент: Поиск Назад									
№	Фамилия	Имя	Отчество	Дата рождения	Адрес	Телефон	Фото	Действия	
1)	Иванов	Иван	Иванович	15.05.1985	Ладыгина 11, кв. 109	89000000000		Редактировать	Изменить фотографию
2)	Петров	Петр	Петрович	17.08.1978	Ул. Адмирала Фокина 9 кв. 32а	89000000001		Редактировать	Изменить фотографию

Рис. 5. Страница поиска клиента для редактирования

Изменение фотографии не вызывает трудностей, необходимо просто выбрать нужную фотографию и нажать на кнопку «Изменить фотографию».

Найти необходимый заказ можно по нескольким критериям: «Программа», «Статус» и «Клиент». Из-за того, что менеджер не может помнить все заказы и клиентов, то все три критерия поиска не связаны.

Многие клиенты приходят семьями, и часто они одновременно получают другой статус. Чтобы отдельно у каждого такого клиента не редактировать данные, создана страница изменения статуса. Она позволяет изменять статус одновременно у нескольких клиентов, что в разы экономит время менеджера.

Форма добавления заказа позволяет вводить данные о заказе той или иной услуги. Для исправления случайных ошибок и отображения внесенных за услуги денег создана страница поиска заказа для редактирования.

Поиск происходит по двум полям одновременно, то есть, чтобы найти необходимого клиента, нужно знать, по какой программе он проходит в организации и его статус на данный момент.

Само редактирование происходит на основе формы «Добавление заказа», с учетом того, что все данные из базы уже введены в поля. Для изменения необходимо просто ввести коррективы в нужные поля и нажать кнопку «Сохранить изменения», после этого форма автоматически обновится и вы-

даст уже измененные данные. Поле «Клиент» – это выпадающий список, который динамически обновляется при добавлении нового клиента в таблицу «clients».

Последняя кнопка переводит на страницу рассылки писем, она автоматически генерируется из таблицы «clients». Для отправки почты необходимо поставить одну или несколько галочек в необходимых строках. После расстановки «галочек» происходит выделение тех клиентов, которым необходимо отправить почту.

Заключение

Средства, использованные для разработки системы, показывают хорошую надежность. Разработанная информационная система потребляет мало ресурсов, поэтому может стабильно работать на компьютерах малой мощности.

Для обеспечения безопасности каждому менеджеру присвоен свой логин и пароль, под которым он заходит в систему, а также все заносится в журнал, фиксирующий, кто и когда производил вход в систему. Данная функция позволяет увеличить уровень безопасности базы.

Во время работы были рассмотрены все виды моделей баз данных, и из всех видов была выбрана та, которая использует организацию данных в виде двумерных таблиц (реляционная модель).

Для решения задачи по созданию базы данных был использован локальный сервер Denver и язык SQL. С помощью языков

HTML, PHP, JavaScript были решены задачи по созданию интерфейса для управления базой данных, а также для оформления информационной системы.

Информационная система, созданная для предприятия, позволит работнику тратить меньше времени на получение данных клиентов, даст возможность быстрого поиска клиентов в базе. При этом меньше времени будет занимать создание заказа на предоставление услуг, что позволит работнику с легкостью отслеживать, на какой стадии находится заказ, сколько нужно оплатить за заказ, быстро изменять статус одного или сразу нескольких заказов в случаях, когда клиенты приходят всей семьей.

В дальнейшем предполагается разработка следующей версии информационной системы. На следующем уровне предполагается расширить объем текстовой информации. Для ее обработки планируется применение специальной информационной технологии обработки качественной информации [12].

Список литературы

1. Батаев А.В. Информатика. Технологии баз данных в информационных экономических системах. учебное пособие / А.В. Батаев – СПб: Изд-во Политехнического ун-та, 2006. – 77 с.
2. Батаев А.В. Оценка развития информационных технологий в финансовой сфере России // Молодой ученый. – 2016. – № 13. – С. 375–378.
3. Информационный ресурс для программистов [Электронный ресурс] – Режим доступа: <http://www.sql.ru/> (дата обращения: 17.11.2016).
4. Мартишин С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: учеб. пособие для студентов вузов / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. – М.: ФОРУМ: ИНФРА-М, 2012. – 160 с.
5. Мартышенко С.Н., Егоров Е.А. Информационная технология повышения эффективности обработки качественной информации // Информационные технологии моделирования и управления. – 2009. – № 6 (58). – С. 753–760.
6. Мартышенко С.Н., Мартышенко Н.С. Методы обнаружения ошибок в эмпирических данных // Известия высших учебных заведений. Северо-Кавказский регион. Серия: Технические науки. – 2008. – № 1. – С. 11–14.
7. Одияко Н.Н., Голодная Н.Ю., Гузенко А.Г., Гусев Е.Г. Оптимизация деятельности предприятий малого бизнеса как основа повышения качества обслуживания населения // Экономика и предпринимательство. – 2016. – № 6 (71). – С. 735–738.
8. Одияко Н.Н., Ермак И.В. Создание интерфейса взаимодействия пользователя с базой данных предприятия // В сборнике: Интеллектуальный потенциал вузов – на развитие Дальневосточного региона России и стран АТР. Материалы XVIII Международной конференции студентов, аспирантов и молодых ученых. – Владивосток, 2016. – том 2 – С. 147–149.
9. Осипов Д.Л. Базы данных и DELPHI. Теория и практика / Д.Л. Осипов. – СПб: БХВ-Петербург, 2011. – 733 с.
10. Прохоренко Н.А., Дронов В.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. – 4-е изд., перераб. И доп. / Н.А. Прохоренко, В.А. Дронов. – СПб.: БХВ-Петербург, 2015. – 768 с.
11. Сайт офисных продуктов компании Microsoft SQL [Электронный ресурс] – Режим доступа: <http://office.microsoft.com/ru-ru/access-help/> (дата обращения: 17.11.2016).

References

1. Bataev A.V. Informatika. Tehnologii baz dannyh v informacionnyh jekonomicheskikh sistemah. uchebnoe posobie / A.V. Bataev SPb: Izd-vo Politehnicheskogo un-ta, 2006. 77 p.
2. Bataev A.V. Ocenka razvitiya informacionnyh tehnologij v finansovoj sfere Ros-sii // Molodoy uchenyj. 2016. no. 13. pp. 375–378.
3. Informacionnyj resurs dlja programmistov [Jelektronnyj resurs] Rezhim dostupa: <http://www.sql.ru/> (data obrashhenija: 17.11.2016).
4. Martishin S.A. Proektirovanie i realizacija baz dannyh v SUBD MySQL s ispolzovaniem MySQL Workbench: ucheb. posobie dlja studentov vuzov / S.A. Martishin, V.L. Simonov, M.V. Hrapchenko. M.: FORUM: INFRA-M, 2012. 160 p.
5. Martysenko S.N., Egorov E.A. Informacionnaja tehnologija povyshenija jeffektivnosti obrabotki kachestvennoj informacii // Informacionnye tehnologii modelirovaniya i upravleniya. 2009. no. 6 (58). pp. 753–760.
6. Martysenko S.N., Martysenko N.S. Metody obnaru-zheniya oshibok v jempiricheskikh dannyh // Izvestija vysshih uchebnyh zavedenij. Severo-Kavkazskij region. Serija: Teh-niche-skie nauki. 2008. no. 1. pp. 11–14.
7. Odijako N.N., Golodnaja N.Ju., Guzenko A.G., Gu-sev E.G. Optimizacija dejatelnosti predpriyatij malogo biznesa kak osnova povyshenija kachestva obsluzhivaniya naselenija // Jekonomika i predprinimatelstvo. 2016. no. 6 (71). pp. 735–738.
8. Odijako N.N., Ermak I.V. Sozdanie interfejsa vzaimod-ejstvija polzovatelja s bazoj dannyh predprijatija // V sbornike: Intellektualnyj potencial vuzov na razvitie Dal-nevostochnogo regiona Rossii i stran ATR. Materialy HVIII Mezhdunarodnoj konferencii studentov, aspirantov i molodyh uchenyh. Vladiv-ostok, 2016. tom 2. pp. 147–149.
9. Osipov D.L. Bazy dannyh i DELPHI. Teorija i praktika / D.L. Osipov. SPb: BHV-Piterburg, 2011. 733 p.
10. Prohorenok N.A., Dronov V.A. HTML, JavaScript, PHP i MySQL. Dzhentlmenskij nabor Web-mastera. 4-e izd., pererab. I dop. / N.A. Prohorenok, V.A. Dronov. SPb.: BHV-Peterburg, 2015. 768 p.
11. Sajt ofisnyh produktov kompanii Microsoft SQL [Jele-ktronnyj resurs] Rezhim do-stupa: <http://office.microsoft.com/ru-ru/access-help/> (data obrashhenija: 17.11.2016).