Содержание:

Введение

Актуальность выполнения данной работы обусловлена тем, что современное информационное пространство предполагает обработку большого количества разрозненной информации. Использование разрозненных информационно-коммуникационных технологий уже недостаточно для обеспечения эффективной работы организации и пользователя, в частности. Перевод сервисов в облака позволяет предоставить организациям более широкий спектр услуг и надежность обработки информации. И при этом, использование технологии клиент-сервер позволяет достичь максимального эффекта обработки информации.

Технология клиент-сервер представляют собой специализированную информационную модель по обеспечению пользователям удобного сетевого доступа по требованию к определенному общему фонду конфигурируемых ресурсов, которые могут быть в оперативном порядке освобождены и предоставлены с минимальными затратами или обращениями к провайдеру предоставления услуг.

Объект исследования – интернет-технологии.

Предмет исследования – технология «клиент-сервер».

Целью данной работы является изучение вариантов архитектуры «клиент-сервер».

В соответствии с целью была определена необходимость постановки и решения следующих *задач*:

– дать характеристику технологии «клиент-сервер»;

– изучить модели взаимодействия «клиент-сервер»;

– описать варианты архитектур «клиент-сервер»;

– представить программную реализацию предметной задачи.

1. Технология «клиент-сервер»

1.1. Понятие технологии «клиент-сервер»

«Клиент–сервер» представляет собой вычислительную или сетевую архитектуру, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер – это специальное программное обеспечение [1].

Поскольку одна программа-сервер может выполнять запросы от множества программ-клиентов, её размещают на специально выделенной вычислительной машине, настроенной особым образом, как правило, совместно с другими программами-серверами, поэтому производительность этой машины должна быть высокой. Из-за особой роли такой машины в сети, специфики её оборудования и программного обеспечения, её также называют сервером, а машины, выполняющие клиентские программы, соответственно, клиентами.

Как правило, компьютеры и прикладные программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами, другие имеют возможность обращаться к этим ресурсам. Компьютер, управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер). Клиент и сервер какого-либо ресурса могут находиться как в рамках одной вычислительной системы, так и на различных компьютерах, связанных сетью.

Основной принцип технологии «клиент-сервер» заключается в разделении функций приложения на три группы:

– ввод и непосредственное отображение данных (взаимодействие с пользователем);

– прикладные функции, которые являются характерными для данной предметной области;

– функции управления оперативными ресурсами вычислительной системы [4].

Поэтому, в любом прикладном программном приложении выделяются следующие компоненты:

– компонент представления данных;

– прикладной компонент;

– компонент управления ресурсом.

Связь между компонентами осуществляется по определенным правилам, которые называют «протокол взаимодействия».

Преимущества практического использования технологии «клиент-сервер»:

– отсутствие дублирования кода программы-сервера прикладными программами-клиентами;

– так как все вычисления выполняются на сервере, то требования к персональным компьютерам, на которых установлен клиент, существенно снижаются;

– все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще может быть организован контроль полномочий, чтобы разрешать доступ к данным только клиентам с определенными правами доступа [7].

Недостатки практического использования технологии «клиент-сервер» заключаются в следующем:

– неработоспособность вычислительного сервера может сделать неработоспособной всю вычислительную сеть. Неработоспособным сервером следует считать сервер, производительности которого не хватает на обслуживание всех клиентов, а также сервер, который находится на ремонте, профилактике;

– поддержка работы данной системы требует отдельного специалиста – системного администратора;

– высокая стоимость требуемого оборудования.

1.2. Модели взаимодействия «клиент-сервер»

Компанией Gartner Group, специализирующейся в области исследования информационных технологий, предложена следующая классификация двухзвенных моделей взаимодействия клиент-сервер (двухзвенными эти модели называются потому, что три компонента приложения различным образом распределяются между двумя узлами).

Исторически первой появилась модель распределенного представления данных, которая реализовывалась на универсальной ЭВМ с подключенными к ней неинтеллектуальными терминалами. Управление данными и взаимодействие с пользователем при этом объединялись в одной программе, на терминал передавалась только «картинка», сформированная на центральном компьютере [10].

Затем, с появлением персональных компьютеров и локальных сетей, были реализованы модели доступа к удаленной базе данных. Некоторое время базовой для сетей ПК была архитектура файлового сервера. При этом один из компьютеров является файловым сервером, на клиентах выполняются приложения, в которых совмещены компонент представления и прикладной компонент [2].

Протокол обмена при этом представляет набор низкоуровневых вызовов операций файловой системы. Такая архитектура, реализуемая, как правило, с помощью персональных систем управления базами данных, имеет очевидные недостатки - высокий сетевой трафик и отсутствие унифицированного доступа к ресурсам.

С появлением первых специализированных серверов баз данных появилась возможность другой реализации модели доступа к удаленной базе данных. В этом случае ядро СУБД функционирует на сервере, протокол обмена обеспечивается с помощью языка SQL. Такой подход по сравнению с файловым сервером ведет к уменьшению загрузки сети и унификации интерфейса «клиент-сервер» [17]. Однако, сетевой трафик остается достаточно высоким, кроме того, по прежнему невозможно удовлетворительное администрирование приложений, поскольку в одной программе совмещаются различные функции.

Позже была разработана концепция активного сервера, который использовал механизм хранимых процедур. Это позволило часть прикладного компонента перенести на сервер (модель распределенного приложения). Процедуры хранятся в словаре базы данных, разделяются между несколькими клиентами и выполняются на том же персональном компьютере, что и SQL-сервер [14].

Преимущества такого подхода: возможно централизованное администрирование прикладных функций, значительно снижается сетевой трафик (т.к. передаются не структурированные SQL-запросы, а вызовы хранимых процедур). Недостаток - ограниченность специализированных средств разработки хранимых процедур по сравнению с языками общего назначения (C и Pascal).

В последнее время также наблюдается тенденция ко все большему использованию модели распределенного приложения. Характерной чертой таких приложений является логическое разделение приложения на две и более частей, каждая из которых может выполняться на отдельном компьютере. Выделенные части приложения взаимодействуют друг с другом, обмениваясь сообщениями в заранее согласованном формате. На практике сейчас обычно используются смешанный подход:

– простейшие прикладные функции выполняются хранимыми процедурами на сервере;

– более сложные реализуются на клиенте непосредственно в прикладной программе [11].

Сейчас ряд поставщиков коммерческих систем управления базами данных объявило о планах реализации механизмов выполнения хранимых процедур с использованием языка программирования Java. Это соответствует концепции «тонкого клиента», функцией которого остается только непосредственное отображение данных (модель удаленного представления данных).

1.3. Архитектура «клиент-сервер»

Архитектура клиент-сервер определяет лишь общие принципы непосредственного взаимодействия между компьютерами, детали взаимодействия определяют различные протоколы. Данная концепция говорит, что нужно разделять машины в сети на клиентские, которым всегда что-то надо и на серверные, которые дают то, что надо [23]. При этом взаимодействие всегда начинает клиент, а правила, по которым происходит взаимодействие описывает протокол.

Появление архитектуры клиент-сервер, как и многих других новых компьютерных технологий, сопровождалось рождением новой терминологии, которые и являются компонентами технологии «клиент-сервер»:

– прикладной программный интерфейс (Application Programming Interface, API) - набор функций и подпрограмм, обеспечивающих взаимодействие клиентов и серверов;

– клиент - объект, запрашивающий информацию по сети. Как правило, это персональный компьютер или рабочая станция, запрашивающая информацию у

сервера;

– промежуточное программное обеспечение - набор драйверов, прикладных программных интерфейсов и прочего программного обеспечения, позволяющего улучшить взаимодействие между клиентским приложением и сервером;

– база данных - база данных, в которой доступ к информации ограничен выбором строк, удовлетворяющих определенным критериям поиска;

– сервер - компьютер (как правило, высокопроизводительная рабочая станция, миникомпьютер или мэйнфрейм), хранящий информацию, с которой работают сетевые клиенты;

– язык структурированных запросов (Structured Query Language, SQL) - разработанный корпорацией IBM и стандартизованный институтом ANSI язык для создания, управления и изменения баз данных [6].

Каждый сервер в окружении клиент-сервер предоставляет клиентам набор услуг. Наиболее распространенным типом сервера в архитектуре клиент-сервер является сервер баз данных, как правило, управляющий реляционной базой данных. Высокопроизводительный сервер обеспечивает коллективный доступ нескольких клиентов к одной и той же базе данных [13].

Помимо клиентов и серверов в окружение клиент-сервер входит сеть. Вычислительная модель клиент-сервер по определению является распределенной. Пользователи, приложения и ресурсы располагаются на разных компьютерах и соединены общей локальной, глобальной или составной сетью.

Таким образом, архитектура «клиент-сервер» включает специализированные компоненты, например, специализированное программное обеспечение, использование сервера и прикладной интерфейс. Использование возможностей языка структурированных запросов SQL позволяет выполнить необходимую обработку информации базы данных программной системы.

2. Варианты архитектуры «клиент-сервер»

2.1. Двухуровневая архитектура «клиент-сервер»

В любой сети (даже одноранговой), построенной на современных сетевых технологиях, присутствуют элементы клиент-серверного взаимодействия, чаще всего на основе двухзвенной архитектуры. Двухзвенной (two-tier, 2-tier) она называется из-за необходимости распределения трех базовых компонентов между двумя узлами (клиентом и сервером).

Двухзвенная архитектура используется в клиент-серверных системах, где вычислительный сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные вычислительные ресурсы [3].

Т.е. сервер не вызывает сторонние сетевые прикладные программные приложения и не обращается к сторонним вычислительным ресурсам для выполнения какой-либо части пользовательского структурированного запроса (рис. 1).

Расположение компонентов на стороне клиента или сервера определяет следующие основные модели их взаимодействия в рамках двухзвенной архитектуры:

– сервер терминалов – распределенное представление оперативных данных;

– файл-сервер – доступ к удаленной базе данных и файловым вычислительным ресурсам;

– сервер базы данных – удаленное представление оперативных данных программы;

– сервер приложений – удаленное прикладное программное приложение [15].

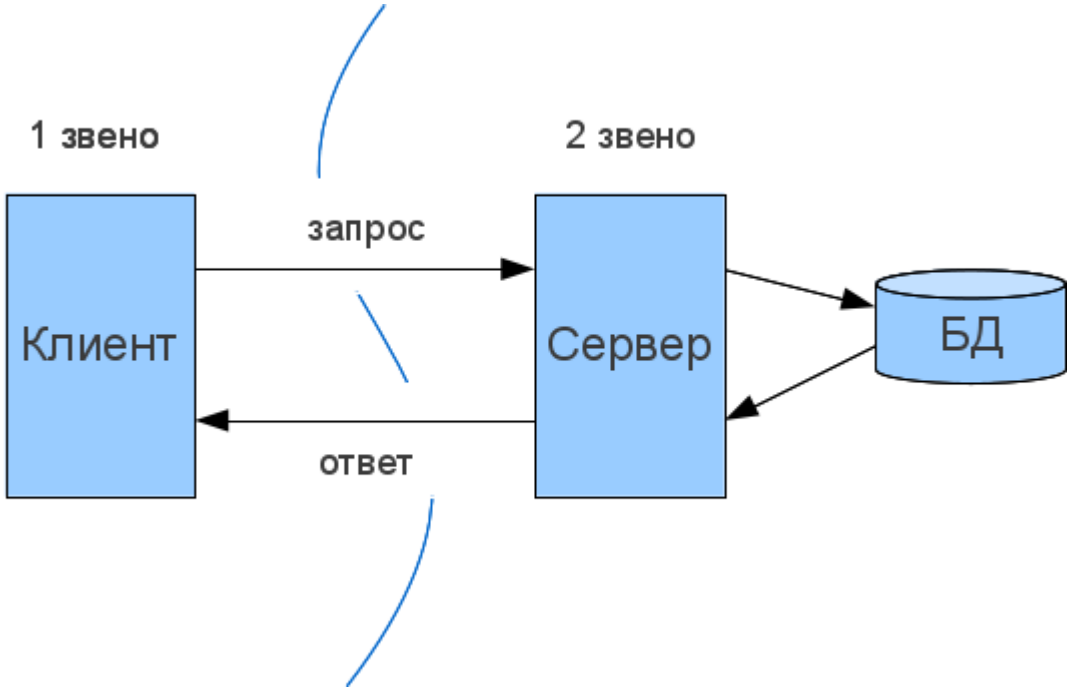

Рис.1. Двухзвенная клиент-серверная архитектура

В настоящее время намечается тенденция возврата к тому, с чего начиналась клиент-серверная архитектура – к централизации вычислений на основе использования модели терминал-сервера [18]. В современной реинкарнации терминалы отличаются от своих алфавитно-цифровых предков тем, что имея минимум программных и аппаратных средств, представляют мультимедийные возможности (в т.ч. графический пользовательский интерфейс) [5].

Работу терминалов обеспечивает высокопроизводительный сервер, куда вынесено все, вплоть до виртуальных драйверов устройств, включая драйверы видеоподсистемы.

2.2. Трехуровневая архитектура «клиент-сервер»

Еще одна тенденция в клиент-серверных технологиях связана со все большим использованием распределенных вычислений. Они реализуются на основе модели сервера приложений, где сетевое приложение разделено на две и более частей, каждая из которых может выполняться на отдельном компьютере [9].

Выделенные части приложения взаимодействуют друг с другом, обмениваясь сообщениями в заранее согласованном формате. В этом случае двухзвенная клиент-серверная архитектура становится трехзвенной (three-tier, 3-tier) [20].

Как правило, третьим звеном в трехзвенной вычислительной архитектуре становится сервер прикладных программных приложений, т.е. прикладные компоненты распределяются следующим образом (рис. 2):

– представление данных – на стороне клиента обработка оперативных данных;

– прикладной компонент – на выделенном сервере прикеладных программных приложений (как вариант, выполняющем функции промежуточного программного обеспечения);

– управление ресурсами – на сервере базы данных, который и представляет запрашиваемые данные [21].

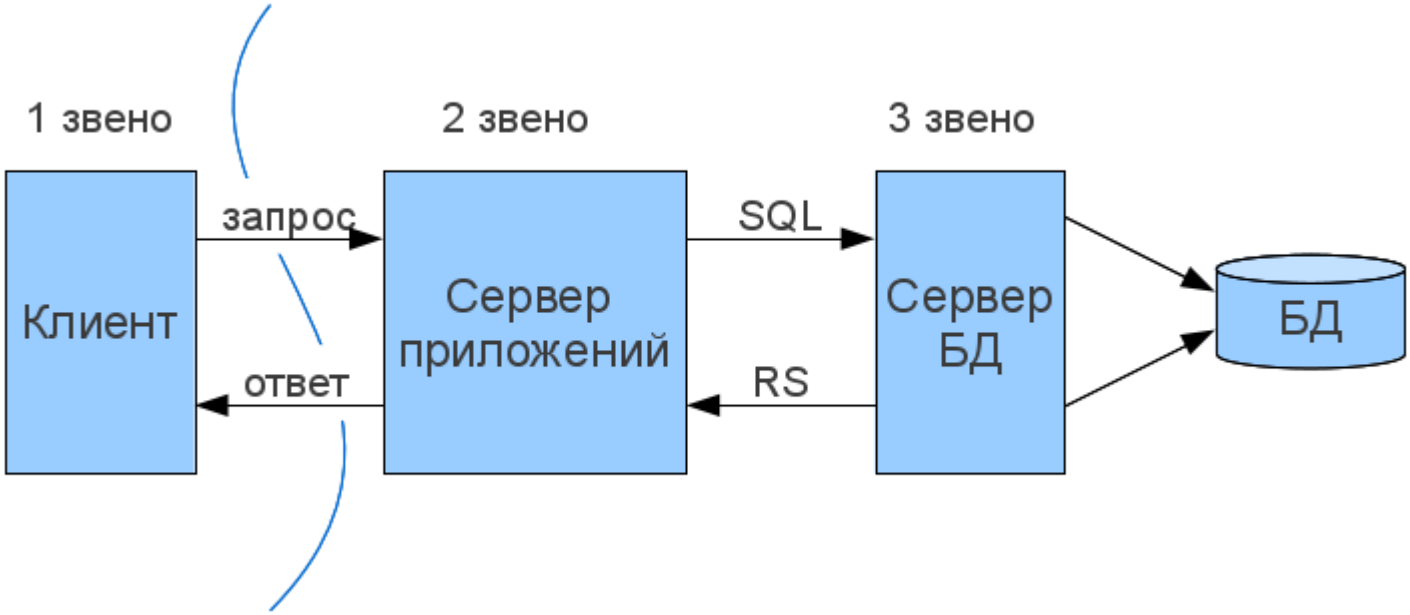

Рис. 2. Трехзвенная клиент-серверная архитектура

Трехзвенная архитектура может быть расширена до многозвенной (N-tier, Multi-tier) путем выделения дополнительных серверов, каждый из которых будет представлять собственные сервисы и пользоваться услугами прочих серверов разного уровня. Абстрактный пример многозвенной модели приведен на рис. 3.

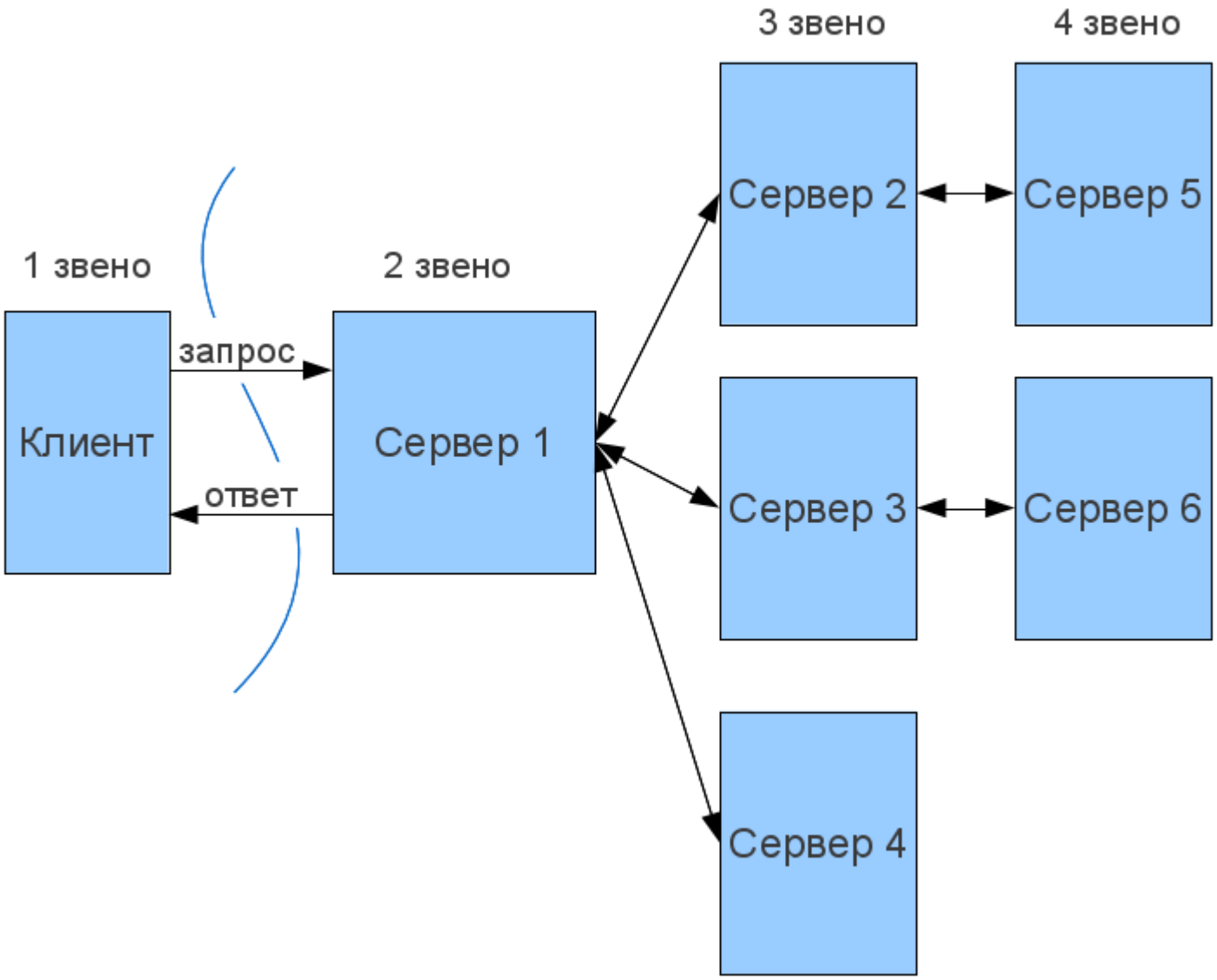

Рис. 3. Многозвенная (N-tier) клиент-серверная архитектура

Таким образом, трехуровневая архитектура «клиент-сервер» имеет уникальную структуру, позволяющая организовать высокоэффективную обработку информации. Данная архитектура позволяет помещать прикладные программы на отдельные серверы приложений, с которыми через API-интерфейс устанавливается связь клиентских рабочих станций. Работа клиентской части приложения сводится к вызову необходимых функций сервера приложения, которые называются «сервисами». Прикладные прогаммы в свою очередь обращаются к серверу баз данных с помощью SQL-запросов.

2.3. Многоуровневая архитектура «клиент-сервер»

В рамках многоуровневого представления вычислительных систем можно выделить несколько групп функций, которые ориентированы на решение различных подзадач:

– функции ввода и отображения данных (обеспечивающие необходимое взаимодействие с пользователем);

– прикладные функции, которые являются характерными для данной предметной области;

– функции управления вычислительными ресурсами (файловой системой, базой данных и т.д.) [12].

В основном выполнение перечисленных функций обеспечивается прикладными программными средствами, которые можно представить в виде взаимосвязанных компонентов, где:

– компонент представления отвечает за обработку пользовательского интерфейса;

– прикладной программный компонент реализует алгоритм решения конкретной вычислительной задачи;

– прикладной программный компонент управления ресурсом обеспечивает доступ к необходимым ресурсам [22].

Автономная система представляет все эти прикладные программные компоненты как на различных уровнях, так и на уровне прикладных программных приложений (не характерно для современных программ) [8].

Так же и вычислительная сеть – она представляет все эти прикладные компоненты, но, в общем случае, распределенные между узлами. Задача сводится к обеспечению сетевого взаимодействия между этими компонентами.

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты, потребители этих функций [16].

Практические реализации такой архитектуры называются клиент-серверными технологиями. Каждая технология определяет собственные или использует имеющиеся правила взаимодействия между клиентом и сервером, которые называются протоколом обмена (протоколом межсетевого взаимодействия).

Двухзвенная архитектура является намного проще, в отличие от трехзвенной, так как все получаемые запросы обслуживаются одним вычислительным сервером, но именно из-за этого она будет менее надежной и предъявляет повышенные требования к вычислительной производительности используемого сервера.

Трехзвенная архитектура сложнее, но благодаря тому, что функции распределены между серверами второго и третьего уровня, эта архитектура представляет:

– высокую степень гибкости и масштабируемости;

– высокую безопасность (т.к. защиту можно определить для каждого сервиса или уровня);

– высокую производительность (т.к. задачи распределены между серверами) [19].

Таким образом, была представлена характеристика технологии «клиент-сервер», которая представляет собой вычислительную или сетевую архитектуру, в которой задания или сетевая нагрузка распределяется между активными поставщиками услуг, которые называются серверами, и заказчиками услуг, называемыми клиентами. Были описаны достоинства и недостатки технологии клиент-сервер.

3. Разработка программы

3.1. Разработка базы данных

Целью разработки программы является автоматизация деятельности агента страховой компании. Обработка заявок на объекты страхования является сложной и многогранной задачей, требующей использования дополнительных средств автоматизации. В разрабатываемой базе данных были созданы следующие таблицы: заявка; клиент; валюта; образование; сотрудник; организация; должность; предмет. Схема данных базы данных информационной системы представлена в Приложении.

Таким образом, были представлены основные сведения о используемых реквизитах таблиц базы данных разрабатываемой информационной системы обработки оперативной информации компании. Представленные реквизиты позволяют в полной мере реализовать необходимый функционал разрабатываемой информационной системы.

3.2. Разработка алгоритмов функционирования системы

Подключение необходимых библиотек было выполнено путем их программного вызова в директории using

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Threading.Tasks;

using MySql.Data.MySqlClient;

Следующие операторы предоставляют доступ как для чтения, так и для изменение всех пользовательских дата атрибутов custom data attributes (data-*), установленных у элемента.

string connectionString = @"Data Source=.\\SQLEXPRESS; server=localhost; user=root; Initial Catalog=bank; charset=utf8";

string sql = "SELECT * FROM subject";

Функциональные возможности пользовательских форм информационной системы включают: удаление; добавление и обновление данных. Для каждой из перечисленных функций был разработан соответствующий программный код и разработаны функции доступа к ним.

Так, для осуществления операций удаления данных были реализованы следующие операторы, первый из которых вводит переменную с sql-запросом на удаление. Далее выполнен доступ к полям базы данных информационной системы

string query = "DELETE FROM subject" + " WHERE id_subject = @id_subject and name_subject = @name_subject and note = @note";

Для осуществления операций добавления данных в базу данных информационной системы были реализованы следующие операторы, первый из которых вводит переменную с sql-запросом на добавление.

Далее выполнен доступ к соответствующим полям базы данных информационной системы

query = "INSERT INTO subject (id_subject, name_subject, note) VALUES (@id_subject, @name_subject, @note)";

adapter.InsertCommand = new MySqlCommand(query, connection);

pc = adapter.InsertCommand.Parameters;

Для осуществления операций обновления данных в базу данных информационной системы были реализованы следующие операторы, первый из которых вводит переменную с sql-запросом на обновление.

Далее выполнен непосредственный доступ к полям базы данных информационной системы

query = "UPDATE subject SET id_subject = @id_subject, name_subject = @name_subject, note = @note WHERE id_subject = @iid_subject";

Непосредственный вызов обозначенных программных операторов выполняется при помощи практического использования соответствующих обработчиков функциональных кнопок (read, update, insert, delete) на пользовательских формах.

Таким образом, были представлены основные подходы к обработке оперативных данных информационной системы средствами языка программирования C#.

3.3. Описание порядка работы с системой

Работа с информационной системой начинается путем запуска исполняемого файла program.exe. Интерфейс главной пользовательской формы включает пользовательское меню, содержащее следующие пункты верхнего уровня:

– справочники;

– документы;

– анализ и обработка данных;

– завершение работы, рис. 4.

Первый пункт главного меню информационной системы «Справочники» включает такие справочники: организация; клиент; сотрудник; образование; должность; валюта; предмет.

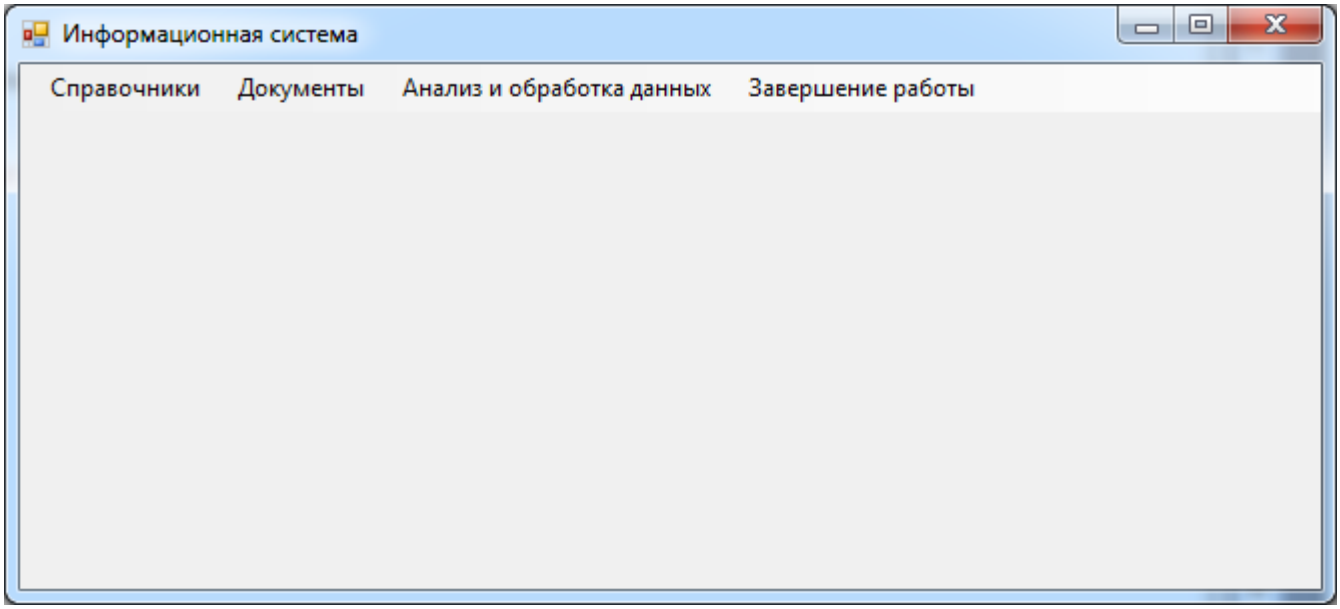

Рис. 4. Интерфейс главной формы

Ввод и редактирование данных о организации можно выполнить при помощи справочника «Организация», пользовательская форма которого представлена на рис. 5.

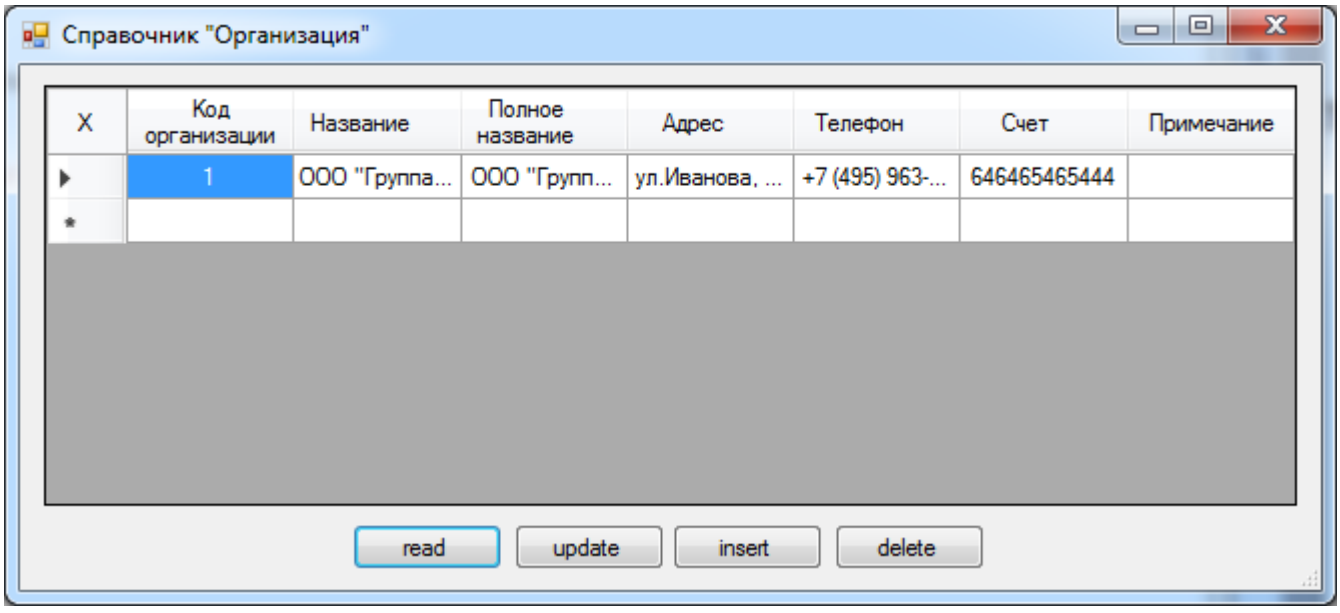

Рис. 5. Пользовательская форма «Организация»

Форма является основным объектом информационной системы, на ней будут располагаться другие информационные объекты доступа к базе данных. Каждая форма в период выполнения информационной системы соответствует отдельному окну.

Ввод и редактирование данных о клиентах организации можно выполнить при помощи справочника «Клиент», пользовательская форма которого представлена на рис. 6.

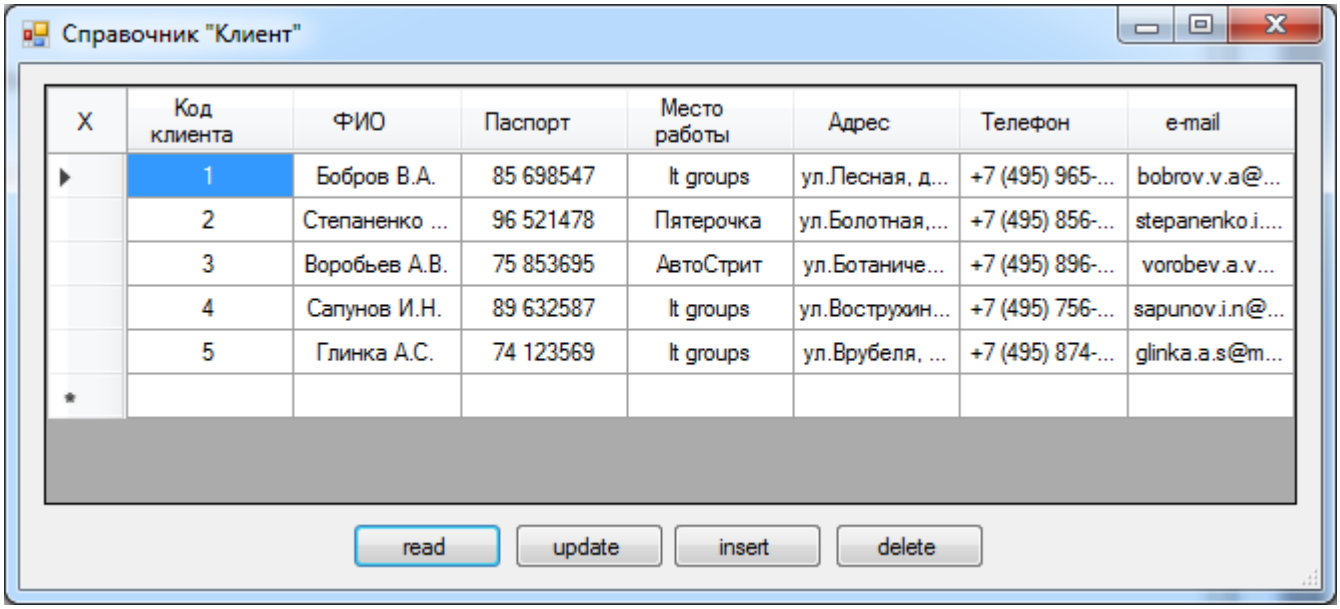

Рис. 6. Пользовательская форма «Клиент»

Ввод и редактирование данных о сотрудниках организации можно выполнить при помощи справочника «Сотрудник», пользовательская форма которого представлена на рис. 7.

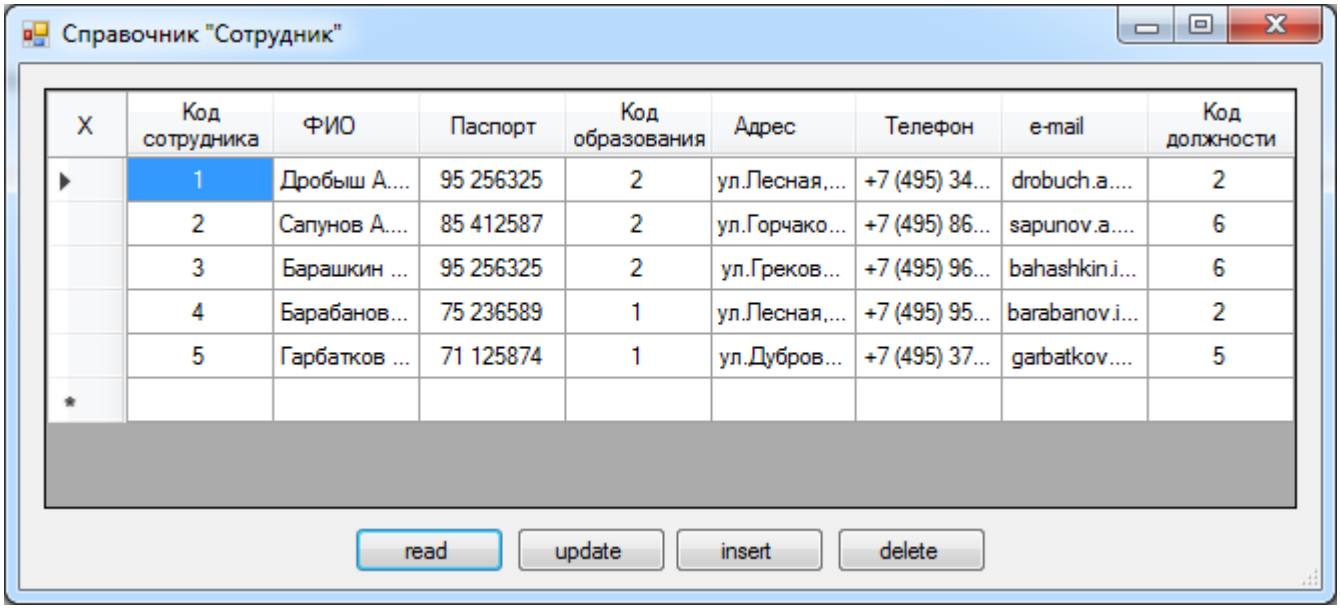

Рис. 7. Пользовательская форма «Сотрудник»

Ввод и редактирование данных о образовании сотрудников организации можно выполнить при помощи справочника «Образование», пользовательская форма которого представлена на рис. 8.

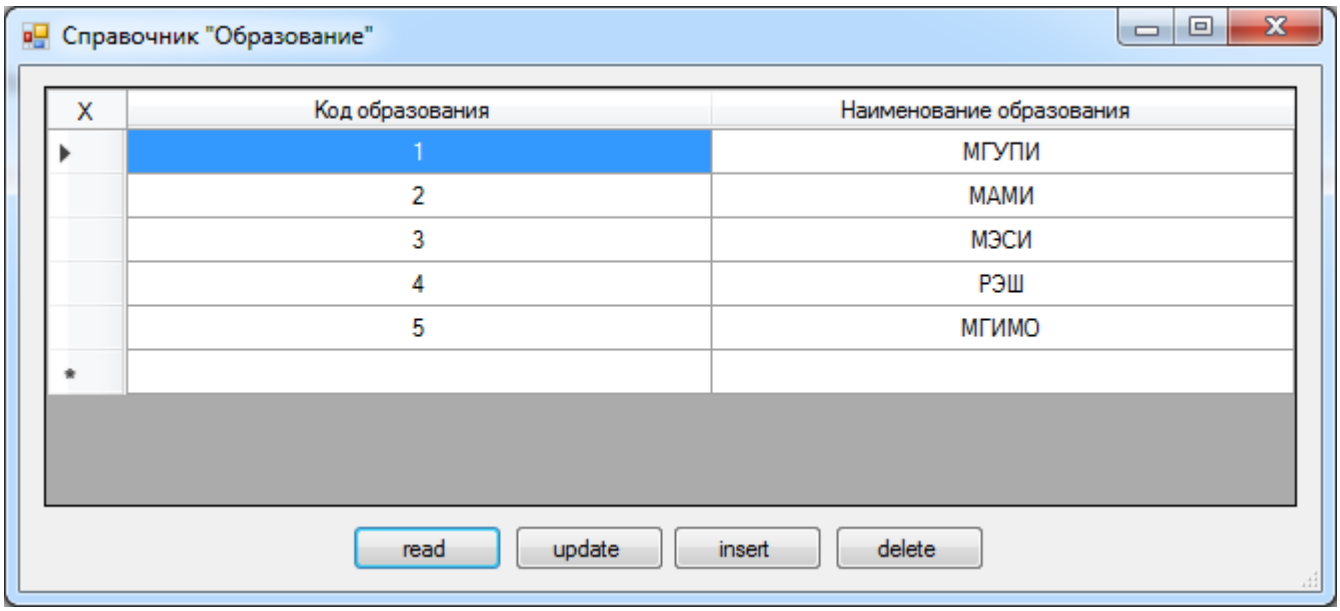

Рис. 8. Пользовательская форма «Образование»

Ввод и редактирование данных о должностях сотрудников организации можно выполнить при помощи справочника «Должность», пользовательская форма которого представлена на рис. 9.

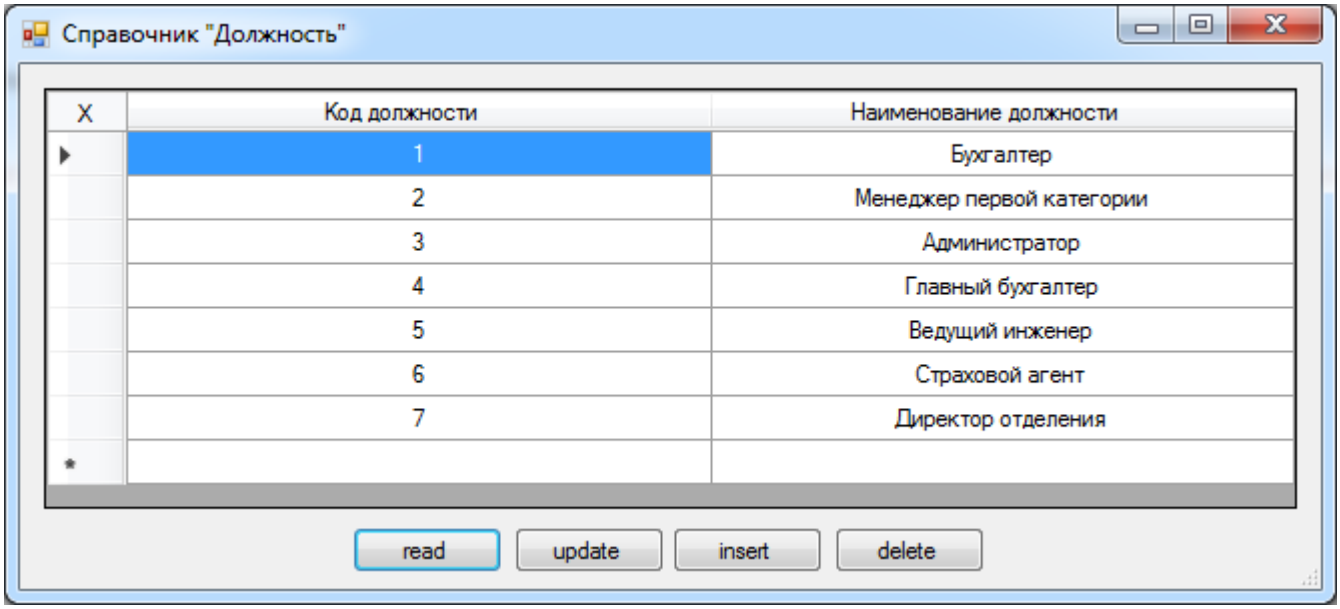

Рис. 9. Пользовательская форма «Должность»

Ввод и редактирование данных о валюте можно выполнить при помощи справочника «Валюта», пользовательская форма которого представлена на рис. 10.

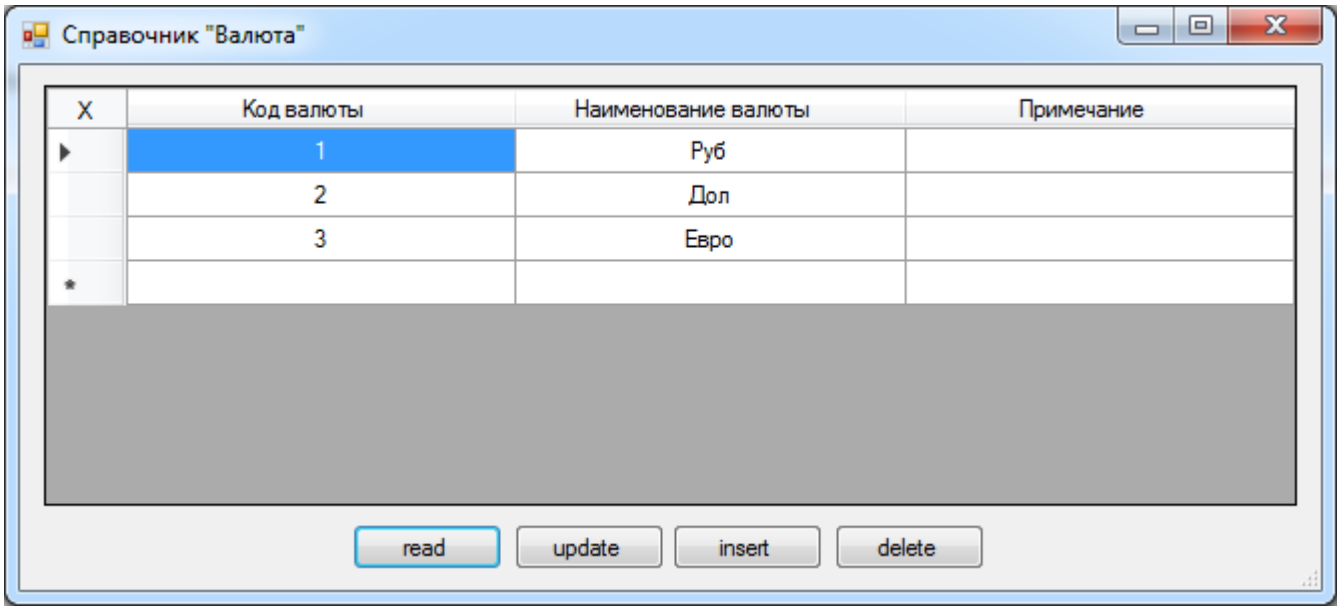

Рис. 10. Пользовательская форма «Валюта»

Ввод и редактирование данных о предметах страхования можно выполнить при помощи справочника «Предмет», пользовательская форма которого представлена на рис. 11.

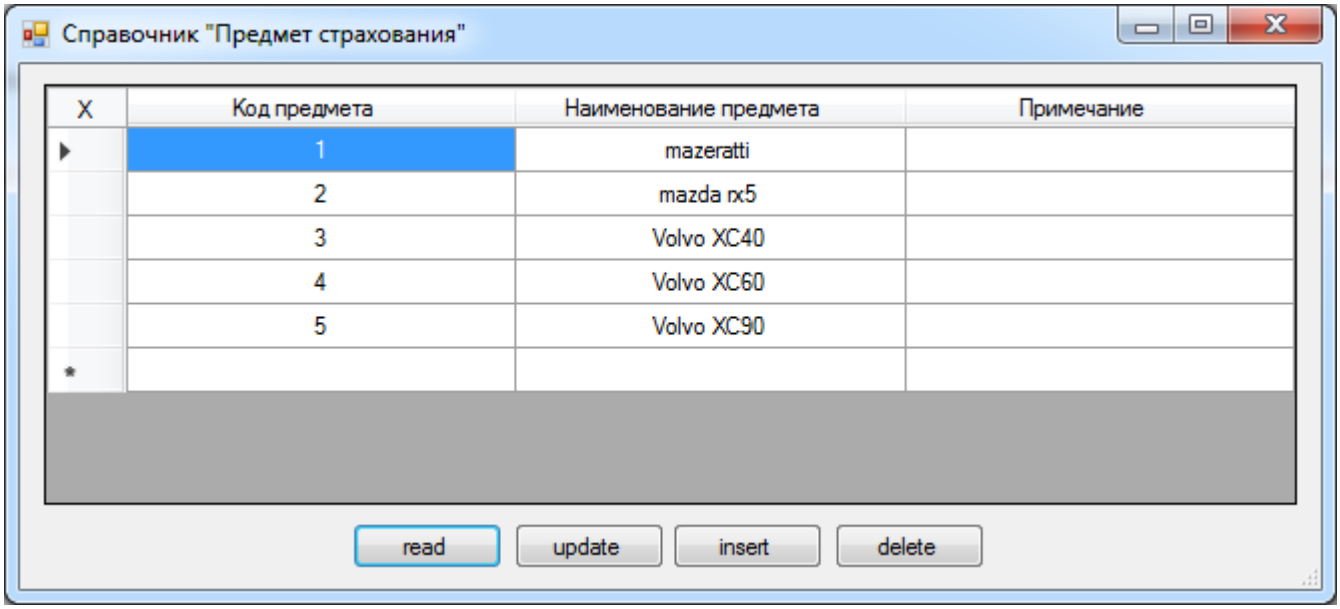

Рис. 11. Пользовательская форма «Предмет»

Все описанные пользовательские формы информационной системы имеют следующие функциональные возможности:

– read (считать оперативные данные из базы данных информационной системы);

– update (обновить данные после изменений или добавления, удаления записей);

– insert (добавить запись);

– delete (удалить запись).

Также, необходимо отметить, что после выполнения операций insert, delete необходимо нажать на кнопку update для обновления базы данных информационной системы.

Второй пункт главного меню информационной системы «Документы» включает такие документы:

– заявка.

Ввод и непосредственное редактирование оперативных данных о заявках на страхование можно выполнить при помощи практического использования документа «Заявка», пользовательская форма которого представлена на следующем рис. 12.

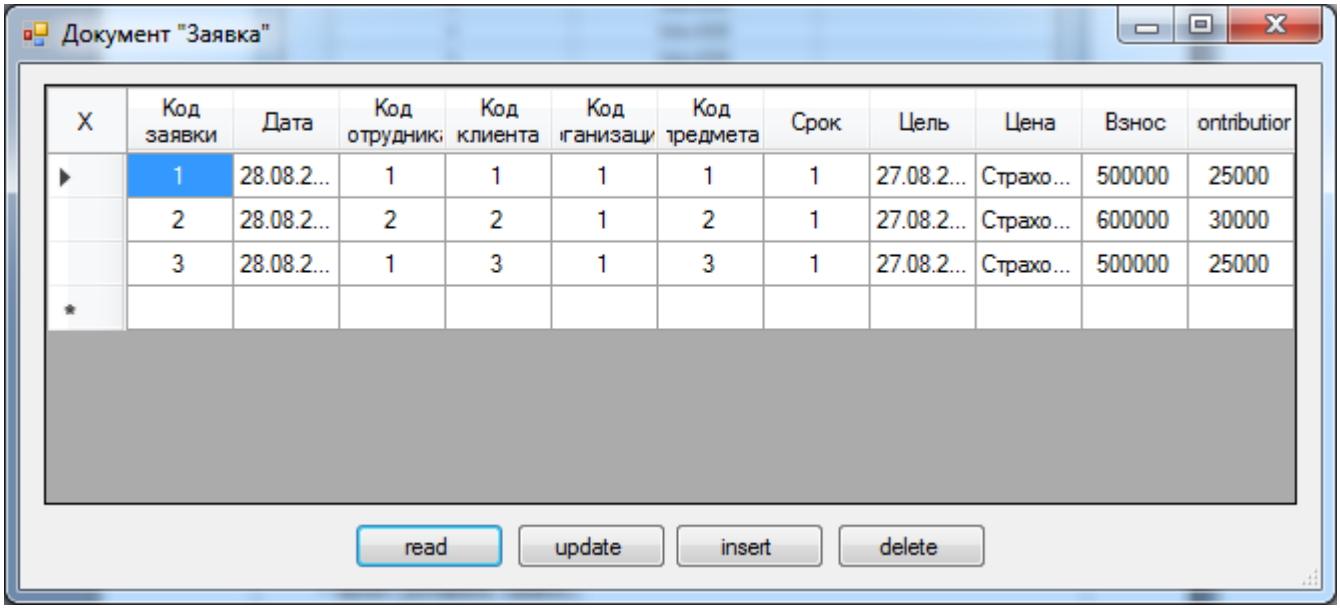

Рис. 12. Пользовательская форма «Заявка»

Третий пункт главного меню информационной системы «Анализ и обработка данных» включает следующие запросы и отчеты:

– информация о клиенте;

– информация о заявке;

– информация о предмете страхования;

– реестр заявок.

Первый запрос предоставляет пользователю информационной системы возможности непосредственного просмотра данных о необходимых клиентах компании. Для поиска информации о клиенте нужно выбрать из списка базы данных интересующую фамилию, после чего будет выведена информация о клиенте. Интерфейсная форма запроса «Информация о клиенте» представлена на рис. 13.

Следующий запрос предоставляет пользователю информационной системы возможности просмотра оперативных данных о заявках клиентов компании. Для поиска оперативной информации о заявке нужно выбрать из списка базы данных интересующий номер, после чего будет выведена информация о заявке клиента.

Интерфейсная форма запроса «Информация о заявке» представлена на рис. 14.

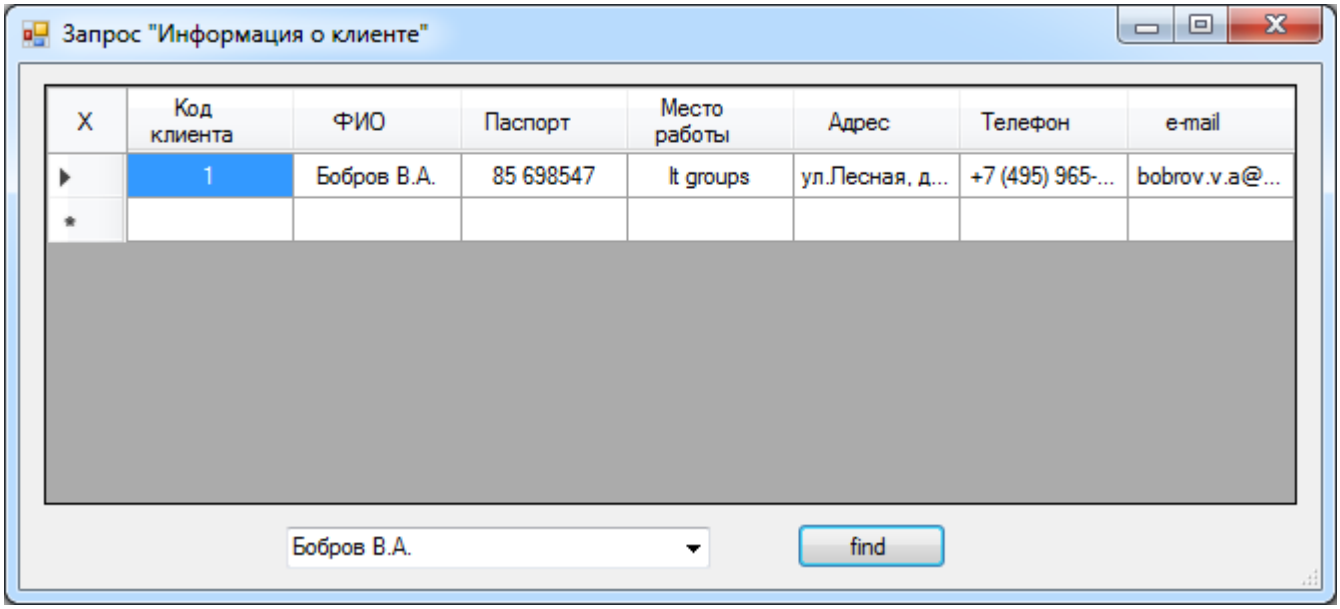

Рис. 13. Пользовательская форма «Информация о клиенте»

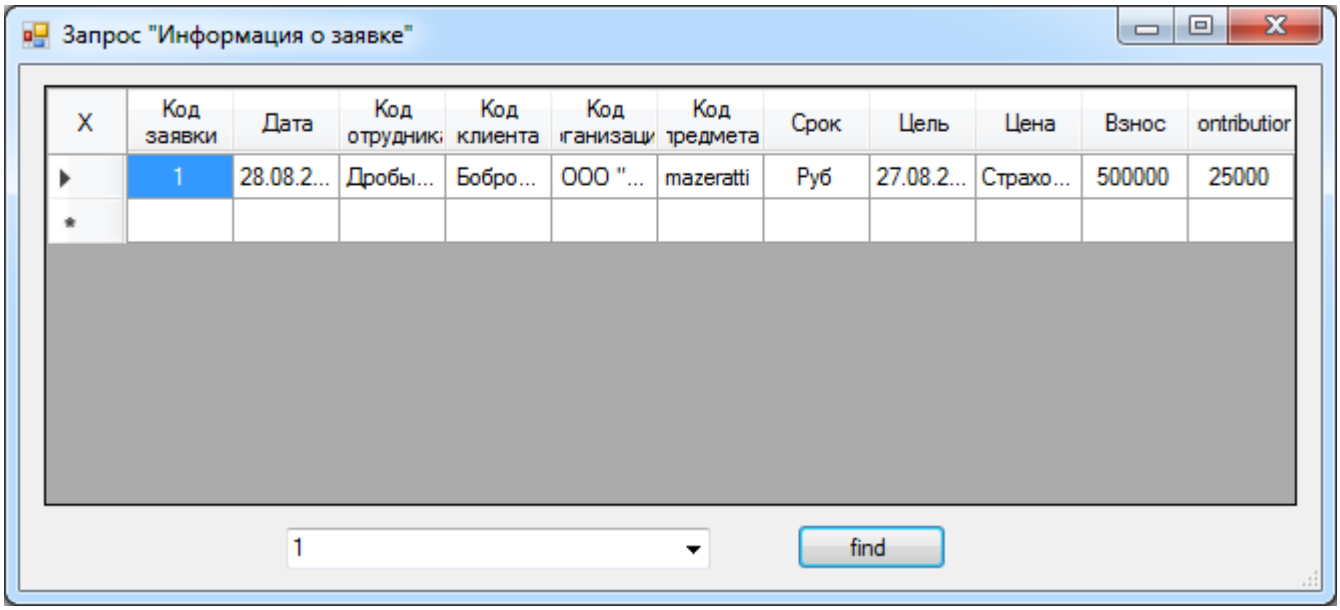

Рис. 14. Пользовательская форма «Информация о заявке»

Следующий запрос предоставляет пользователю информационной системы возможности просмотра оперативных данных о объектах страхования клиентов компании. Для поиска оперативной информации о нужном объекте страхования следует выбрать из списка базы данных интересующий объект, после чего будет выведена информация о нем.

Интерфейсная форма запроса «Информация о объекте страхования» представлена на рис. 15.

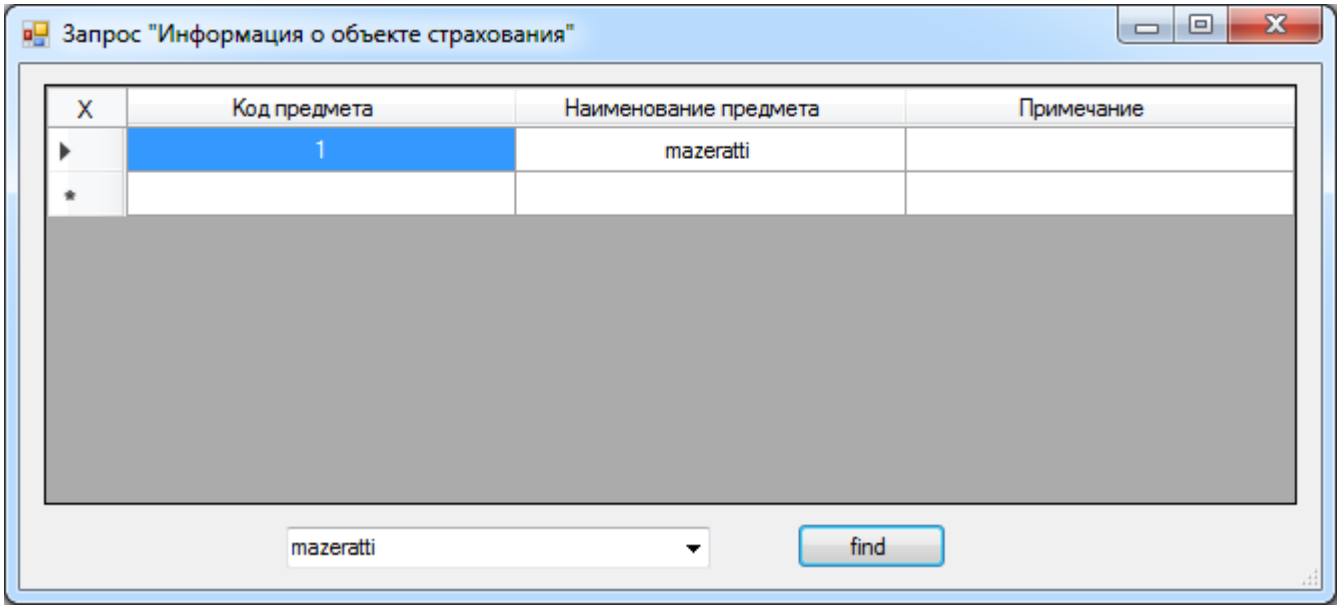

Рис. 15. Пользовательская форма «Информация о объекте страхования»

Таким образом, описанные объекты информационной системы отражают технологию работы с ней. Накопленная оперативная информация в базе данных информационной системы может быть использована для дальнейшей обработки существующими средствами автоматизации компании. Дальнейшим развитием информационной системы является ее оптимизация и добавление новых функциональных возможностей.

Заключение

В процессе выполнения данной работы были получены следующие результаты. Установлено, что «клиент-сервер» представляет собой вычислительную или сетевую архитектуру, в которой задания или сетевая нагрузка распределяется между активными поставщиками услуг, которые называются серверами, и заказчиками услуг, называемыми клиентами.

Среди преимуществ практического использования технологии «клиент-сервер» были отмечены: отсутствие дублирования кода программы-сервера прикладными программами-клиентами; так как все вычисления выполняются на сервере, то требования к персональным компьютерам, на которых установлен клиент, существенно снижаются; все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов.

Недостатки практического использования технологии «клиент-сервер» заключаются в следующем: неработоспособность вычислительного сервера может сделать неработоспособной всю вычислительную сеть; поддержка работы данной системы требует отдельного специалиста – системного администратора; высокая стоимость требуемого оборудования

Была описана двухзвенная архитектура, которая используется в клиент-серверных системах, где вычислительный сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные вычислительные ресурсы. Трехзвенная архитектура связана с все большим использованием распределенных вычислений. Они реализуются на основе модели сервера приложений, где сетевое приложение разделено на две и более частей, каждая из которых может выполняться на отдельном компьютере.

В качестве примера была реализована программа для автоматизации деятельности агента страховой компании. В разрабатываемой базе данных были созданы следующие таблицы: заявка; клиент; валюта; образование; сотрудник; организация; должность; предмет. Также, был разработан интерфейс пользователя.

Список использованной литературы

1. Валитов Ш.М. Современные системные технологии в отраслях экономики: Учебное пособие / Ш.М. Валитов, Ю.И. Азимов, В.А. Павлова. - М.: Проспект, 2016. – 504 с.
2. Венделева М.А. Информационные технологии в управлении.: Учебное пособие для бакалавров / М.А. Венделева, Ю.В. Вертакова. - Люберцы: Юрайт, 2016. – 462 с.
3. Гаврилов М.В. Информатика и информационные технологии: Учебник / М.В. Гаврилов, В.А. Климов. - Люберцы: Юрайт, 2016. – 383 с.
4. Грошев А.С. Информационные технологии : лабораторный практикум / А. С. Грошев. – 2-е изд. – М.-Берлин: Директ-Медиа, 2015. – 285 с.
5. Грошев А.С., Закляков П. В. Информатика: учеб. для вузов – 3-е изд., перераб. и доп. – М.: ДМК Пресс, 2015. – 588 с.
6. Дарков А.В. Информационные технологии: теоретические основы: Учебное пособие / А.В. Дарков, Н.Н. Шапошников. - СПб.: Лань, 2016. – 448 с.

7. Ерохин В.В. Безопасность информационных систем: учеб пособие / В.В. Ерохин, Д.А. Погонышева, И.Г. Степченко. - М.: Флинта, 2016. – 184 с.
8. Жданов С.А. Информационные системы: учебник / С.А. Жданов, М.Л. Соболева, А.С. Алфимова. - М.: Прометей, 2015. – 302 с.
9. Замятина О.М. Вычислительные системы, сети и телекоммуникации. моделирование сетей.: Учебное пособие для магистратуры / О.М. Замятина. - Люберцы: Юрайт, 2016. – 159 с.
10. Информатика для экономистов: учебник для академического бакалавриата/Под ред. В.П. Полякова.- М.: Юрайт, 2015. – 524 с.
11. Информационные системы и технологии: Научное издание. / Под ред. Ю.Ф. Тельнова. - М.: ЮНИТИ, 2016. – 303 с.
12. Информационные технологии: Учебное пособие / Л.Г. Гагарина, Я.О. Теплова, Е.Л. Румянцева и др.; Под ред. Л.Г. Гагариной - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015. – 320 с.
13. Корнеев И.К. Информационные технологии в работе с документами: Учебник / И.К. Корнеев. - М.: Проспект, 2016. – 304 с.
14. Корпоративные информационные системы управления : учебник / под ред. Н.М. Абдикеева, О.В. Китовой. - М. : ИНФРА-М, 2014. – 563 с.
15. Корячко В.П. Проектирование IP-систем: Учебное пособие для вузов / В.П. Корячко, Ю.М. Цыцаркин, Е.Ю. Скоз. - М.: РиС, 2015. – 224 с.
16. Косиненко Н.С. Информационные системы и технологии в экономике: Учебное пособие для бакалавров / Н.С. Косиненко, И.Г. Фризен. - М.: Дашков и К, 2015. – 304 с.
17. Кренке Д. Теория и Практика построения баз данных / Д. Кренке. - М.: СПб: Питер; Издание 9-е, 2017. – 858 с.
18. Лапшина С.Н. Информационные технологии в менеджменте : учебное пособие / С. Н. Лапшина, Н. И. Тебайкина. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 84 с.
19. Олифер В., Олифер Н. Компьютерные сети (принципы, технологии, протоколы). - СПб.: Питер, 5-е изд., 2016. – 992 с.
20. Советов Б.Я. Информационные технологии: теоретические основы: Учебное пособие / Б.Я. Советов, В.В. Цехановский. - СПб.: Лань, 2016. – 448 с.
21. Таланов В. М., Федосин С. А. Проектирование информационных систем и баз данных. Учеб. пособие. Изд.3. Переработанное и дополненное – Саранск: Изд-во СВМО, 2013. – 72 с.
22. Цуриков А.Н. Компьютерные системы и сети: учеб. пособие / А.Н. Цуриков; ФГБОУ ВО РГУПС. – Ростов н/Д, 2016. – 64 с.

23. Угринович Н. Информатика и информационные технологии. М.: Бином. Лаборатория Знаний, 2014. – 430 с.

Приложение

Схема данных