

СОДЕРЖАНИЕ

Глава 1. системный анализ предметной области.....	2
1.1 анализ объекта автоматизации.....	2
1.2 обзор информационных технологий.....	4
1.3 Обзор продуктов-аналогов.....	7
Глава 2. проектирование базы данных.....	9
2.1 Разработка инфологической модели.....	9
2.2 Обоснование выбора модели данных.....	10
2.3. Даталогическое проектирование.....	12
2.5. Нормализация, схема БД.....	12
Глава 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ.....	15
3.1. Анализ и выбор СУБД.....	15
3.2. Физическое проектирование базы данных в СУБД.....	22
3.3. Разработка представлений.....	27
3.4. РАЗРАБОТКА ФОРМ.....	31
3.5 РАЗРАБОТКА ОТЧЕТОВ.....	36
3.6. Реализация ограничений, автоматизация обработки данных в БД.....	39
3.7 Безопасность и контроль.....	43
Заключение.....	48
СПИСОК ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	50

ГЛАВА 1. СИСТЕМНЫЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 АНАЛИЗ ОБЪЕКТА АВТОМАТИЗАЦИИ

Агентство недвижимости представляет собой организацию, оказывающую все виды услуг по обмену, продаже, покупке и аренде жилья. В распоряжении компании - исчерпывающие базы данных, содержащие информацию обо всех актуальных предложениях на рынке недвижимости города, что позволяет в кратчайшие сроки предоставить клиенту информацию о предлагаемом объекте, полностью соответствующем его индивидуальным запросам.

Данное агентство недвижимости имеет главный офис (в Центральном районе города), а также дополнительные офисы в других районах города (рис. 1). Это значительно облегчает работу с клиентом, а также способствует более широкому охвату рынка недвижимости. В центральном офисе находится единая база данных, которая создается и корректируется с учетом информации от филиалов.



Миссия агентства недвижимости заключается в оказании клиентам риэлтерских услуг на уровне, соответствующем высшим профессиональным и этическим стандартам.

Агентство недвижимости выполняет следующие действия:

1. ведение и поддержка актуальности БД покупки, продажи, обмена и аренды недвижимости;
2. прием заявок от клиентов;
3. проведение переговоров с клиентами;
4. подбор подходящего варианта;
5. помощь в оформлении документов, поэтапная поддержка всего процесса купли-продажи, обмена и аренды недвижимости.

Схема функционирования агентства недвижимости представлена на (Рис. 2)



Рис. 2 Схема функционирования агентства недвижимости

1.2 ОБЗОР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Наиболее быстрая и эффективная разработка программных продуктов учёта операций по работе со счетами достигаются при использовании как средств визуального программирования, так и СУБД. В данной сфере безусловными лидерами рынка являются:

- Delphi.
- C++ Builder ?
- MS Visual Studio.

Delphi, ранее Borland Delphi и CodeGear Delphi, – интегрированная среда разработки ПО для Microsoft Windows на языке Delphi (ранее носившем название Object Pascal), созданная первоначально фирмой Borland, и на данный момент принадлежащая и разрабатываемая Embarcadero Technologies. Embarcadero Delphi является частью пакета Embarcadero RAD Studio и поставляется в четырёх редакциях: Starter, Professional, Enterprise и Architect. Delphi - программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке программирования Object Pascal.

C++ Builder– программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке программирования C++.

Изначально разрабатывался компанией BorlandSoftware, а затем её подразделением CodeGear, ныне принадлежащим компании EmbarcaderoTechnologies.

C++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты. Цикл разработки аналогичен Delphi. Большинство компонентов,

разработанных в Delphi, можно использовать и в C++ Builder без модификации, но обратное утверждение не верно.

C++ Builder содержит инструменты, которые при помощи drag-and-drop действительно делают разработку визуальной, упрощает программирование благодаря встроенному WYSIWYG — редактору интерфейса и пр.

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающие интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework и Microsoft Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов цикла разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Microsoft Access

Первая версия СУБД Access появилась в начале 90-х годов. Это была первая настольная реляционная СУБД для 16-разрядной версии Windows. Популярность этой СУБД резко возросла после ее включения в состав Microsoft Office.

В отличие от Visual FoxPro, фактически превратившегося в средство разработки приложений, Access ориентирован в первую очередь на пользователей Microsoft Office, в том числе и не знакомых с программированием. Это, в частности, проявилось в том, что вся информация, относящаяся к конкретной базе данных, а именно таблицы, индексы (естественно программно поддерживаемые), правила ссылочной целостности, список пользователей, а также формы, запросы и отчеты хранятся в одном файле, что в целом удобно для начинающих пользователей.

В состав версии Access-2013 входят:

- средства манипуляции данными Access и данными, доступными через ODBC (последние могут быть "присоединены" к базе данных Access);
- средства создания форм, отчетов и приложений, при этом отчеты могут быть экспортированы в формат Microsoft Word или Microsoft Excel, а для создания приложений используется язык Visual Basic for Applications, общий для всех составных частей Microsoft Office;
- средства публикации отчетов в Internet;
- средства создания интерактивных Web-приложений для работы с данными (Data Access Pages);
- средства доступа к данным серверных СУБД через OLE DB;
- средства создания клиентских приложений для Microsoft SQL Server;
- средства администрирования Microsoft SQL Server.

1.3 ОБЗОР ПРОДУКТОВ-АНАЛОГОВ

1. «1С:Риэлтор»

Конфигурация 1С для управления продажами недвижимости. Продукт предназначен для компаний, занимающихся сделками по купле-продаже объектов недвижимости как на первичном, так и на вторичном рынках. Решение позволяет повысить эффективность подготовки и проведения сделок с недвижимостью в строительных и девелоперских компаниях, в риэлторских компаниях и агентствах недвижимости.

2. «Intrum CRM»

Онлайн CRM система. База объектов и клиентов, прием входящих заявок и автоматический подбор для них объектов, ведение сделки по стадиям и формы для быстрого создания документов, выгрузка объектов недвижимости на сайты объявлений, возможность sms, email рассылок, звонки непосредственно из браузера через встроенную voip телефонию, интеграция с социальными сетями. Готовое отраслевое решение для агентств недвижимости.

3. «RP CRM»

Приложение для автоматизации современного агентства недвижимости. База объектов, отправка карточек объектов клиенту в 1 клик, клиенты и все сделки в одном месте, ежедневник с "напоминалкой" и смс-информированием, формы и образцы договоров.

4. «Plektan»

Украинская онлайн система для риэлторских агентств. Предоставляет единое рабочее пространство для агентов. В нем хранится вся информация о покупателях и продавцах недвижимости, необходимая для работы агенту. Система позволяет быстро подобрать предложения для клиентов.

Руководители подразделений или компании имеют доступ к полной аналитике.

Украинская онлайн система для риэлторских агентств Plektan - это набор инструментов, позволяющий добиться роста числа клиентов и увеличить конверсию при работе с ними. Plektan предоставляет единое рабочее пространство для агентов. В нем хранится вся информация о покупателях и продавцах недвижимости, необходимая для работы агенту. Система позволяет быстро подобрать предложения для клиентов. Руководители подразделений или компании имеют доступ к полной аналитике о работе сотрудников и предприятия в целом. Система Plektan дает возможность управлять конфликтами интересов между агентами и компанией.

1. «Лидер»

Бесплатная программа по недвижимости "Лидер" составляет основную часть комплекса Лидер Недвижимость (программного обеспечения для агентов и агентств) и является востребованным инструментом в сфере работы с недвижимостью, в связи с некоторыми специфичными фактами развития последней. Не стоит забывать, что объем информации об объектах недвижимости растет. Растет не только объем информации у каждого отдельно взятого агентства или риэлтора, растет само количество объектов на рынке. Растет количество жилых массивов, населенные пункты увеличиваются территориально, становятся ближе друг к другу, объединяясь и примыкая к более крупным городам. Информационные базы переживают все большую нагрузку, потому что обмен данными требует все большего количества ресурсов. Телефон уже не может удовлетворить потребностям информационного потока нового рынка недвижимости.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Проектирование базы данных — процесс создания схемы базы данных и определения необходимых ограничений целостности.

Основные задачи:

- Обеспечение хранения в БД всей необходимой информации.
- Обеспечение возможности получения данных по всем необходимым запросам.
- Сокращение избыточности и дублирования данных.
- Обеспечение целостности базы данных.

2.1 РАЗРАБОТКА ИНФОЛОГИЧЕСКОЙ МОДЕЛИ

В простейшем виде инфологическая модель может быть отображена в виде взаимосвязей между бизнес-компонентами и бизнес-процессами, как это показано на рисунке 3. В практике проектирования информационных систем такие схемы получили название ER-диаграмм (Entity-relationship diagram (ERD) – диаграмма «Сущность-связь»). ER-диаграммы хорошо вписываются в методологию структурного анализа и проектирования информационных систем. Такие методологии обеспечивают строгое и наглядное описание проектируемой системы, которое начинается с ее общего обзора и затем уточняется, давая возможность получить различную степень детализации объекта с различным числом уровней.

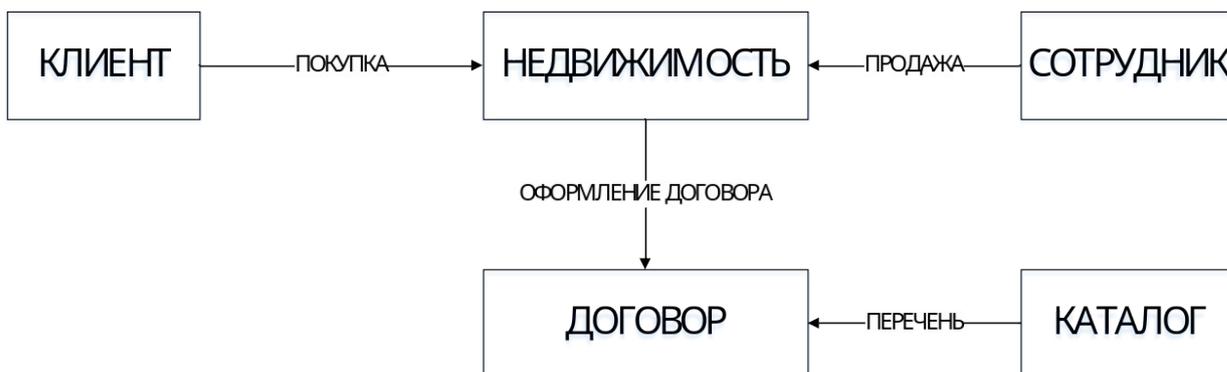


Рисунок 1 – Диаграмма взаимосвязей между бизнес-компонентами и бизнес-процессами (ER-диаграмма)

2.2 ОБОСНОВАНИЕ ВЫБОРА МОДЕЛИ ДАННЫХ

При проектировании концептуальной модели все усилия разработчика должны быть направлены в основном на структуризацию данных и выявление взаимосвязей между ними без рассмотрения особенностей реализации и вопросов эффективности обработки. Проектирование концептуальной модели основано на анализе решаемых на этом предприятии задач по обработке данных. Концептуальная модель включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области и выявляемых в результате анализа данных. Имеются в виду данные, используемые как в уже разработанных прикладных программах, так и в тех, которые только будут реализованы (рис. 4)

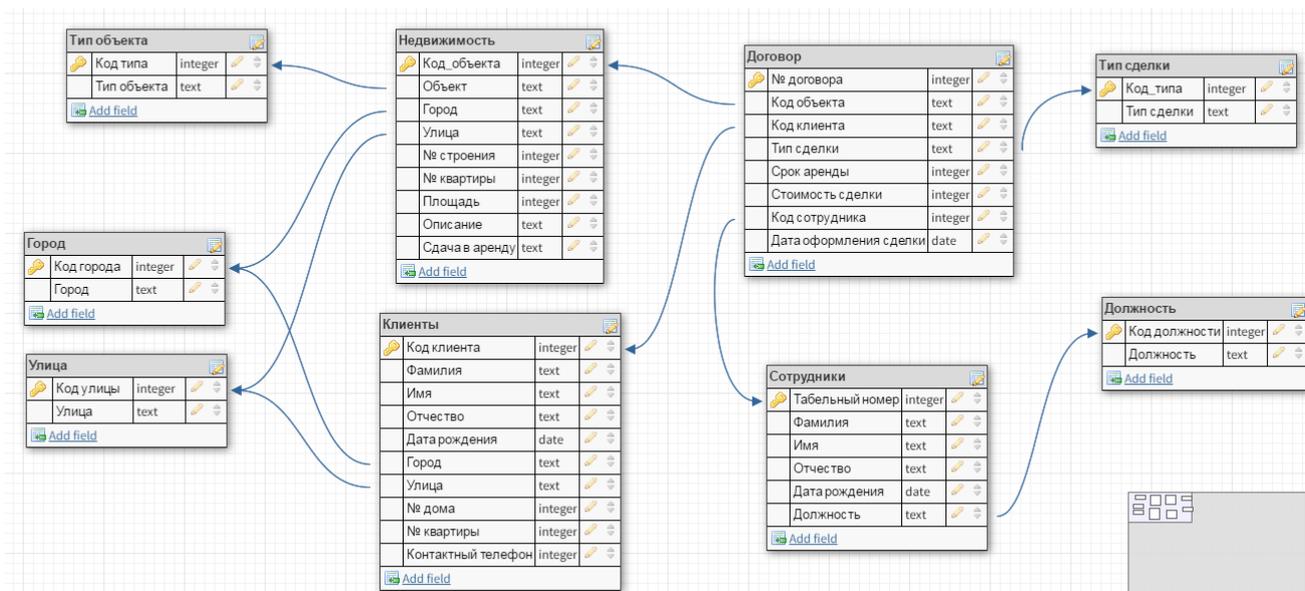


Рисунок 4. Концептуальная модель АИС «Агентства недвижимости»

В каждой таблице выделены первичные и внешние ключи для связей. Целостность данных обеспечивается наличием внешних ключей [3].

Таблицы «Улицы» и «Город» связаны с таблицами «Клиенты» и «Недвижимость» связью «один-ко многим».

Таблица «Тип объекта» связана с таблицей «Недвижимостью» связью «один-ко многим».

Таблица «Должность» связана с таблицей «Сотрудник» связью «один-ко многим».

Таблицы «Тип сделки», «Недвижимость», «Сотрудники» и «Клиенты» связаны с таблицей «Договор» связью «один-ко многим».

Средством разработки базы данных была выбрана СУБД Microsoft Access 2013.

2.3. ДАТАЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Даталогическая модель базы данных является моделью логического уровня и строится для конкретной СУБД, в среде, в которой проектируется база данных, в данном случае это СУБД Access.

При даталогическом моделировании необходимо спроектировать структуру таблиц с учетом требований к реляционным моделям в среде СУБД Access.

Обычно исходная реляционная модель формируется из ER-модели путем преобразования классов объектов и процессов в самостоятельные отношения – таблицы.

В результате моделирования может быть получена даталогическая модель следующего вида:

Должность (Код должности, Должность)

Сотрудники (Табельный номер, Фамилия, Имя, Отчество, Дата рождения, Улица, Дом, Квартира, Телефон, Должность)

Город (Код города, Город)

Улица (Код улицы, Улицы)

Клиенты (Код клиента, Фамилия, Имя, Отчество, Дата рождения, Город, Улица, № дома, № квартиры, Контактный телефон)

Тип объекта (Код типа, Тип объекта)

Тип сделки (Код сделки, Тип сделки)

Недвижимость (Код объекта, Объект, Город, Улица, № строения, № квартиры, Площадь, Описание, Сдача в аренду).

2.5. НОРМАЛИЗАЦИЯ, СХЕМА БД

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.

На использовании разновидностей ER-модели основано большинство современных подходов к проектированию баз данных (главным образом, реляционных). Модель была предложена Ченом (Chen) в 1976 г. Моделирование предметной области базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов. В связи с наглядностью представления концептуальных схем баз данных ER-модели получили широкое распространение в системах CASE, поддерживающих автоматизированное проектирование реляционных баз данных. Среди множества разновидностей ER-моделей, одна из наиболее развитых применяется в системе CASE фирмы ORACLE.

Преимущество ER-диаграмм в том, что они позволяют наглядно представить взаимосвязь объектов предметной области и при этом нет необходимости в манипулировании множеством атрибутов при нормализации, как это имеет место в методе декомпозиции.

Ключевыми для данного метода являются понятия сущность и связь.

Сущность – это реальный или представляемый объект, информация о котором должна сохраняться и быть доступна. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности - это имя типа, а не некоторого конкретного экземпляра этого типа. Для большей выразительности и лучшего понимания имя сущности может сопровождаться примерами конкретных объектов этого типа.

Связь – это графически изображаемая ассоциация, устанавливаемая между двумя сущностями. Эта ассоциация всегда является бинарной и может существовать между двумя разными сущностями или между сущностью и ей

же самой (рекурсивная связь). В любой связи выделяются два конца (в соответствии с существующей парой связываемых сущностей), на каждом из которых указывается имя конца связи, степень конца связи (сколько экземпляров данной сущности связывается), обязательность связи (класс принадлежности) т.е. любой ли экземпляр данной сущности должен участвовать в данной связи.

Для применения метода ER-диаграмм необходимо определить сущности, которые являются необходимыми (стержневыми) для проектируемой базы данных, а также их ключевые атрибуты. После этого строятся ER-диаграммы. Затем согласно степени связи и класса принадлежности объекта строятся отношения, которые необходимо проверить на соответствие 3НФ. Если хотя бы одно полученное отношение не удовлетворяет требованиям 3НФ, то необходимо пересмотреть ER-диаграмму, соответствующую этому отношению.

Используя вышеописанные правила, применим метод ER-диаграмм для нормализации исходного универсального отношения.

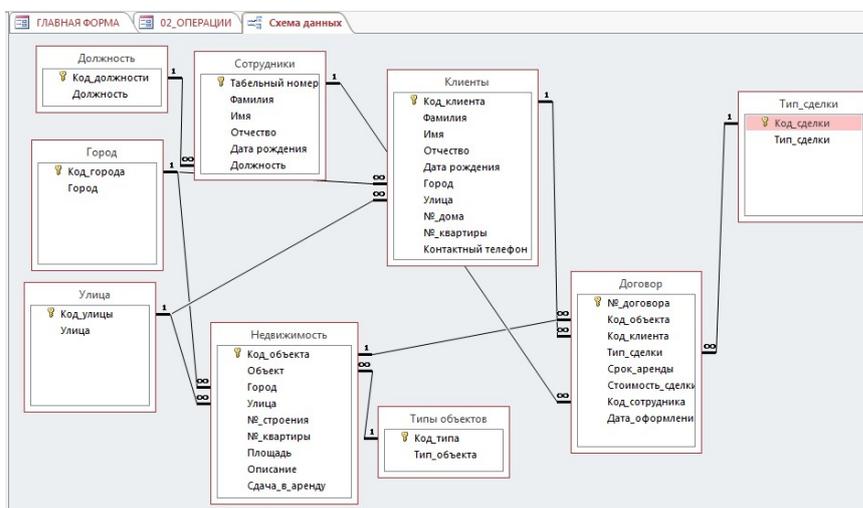


Рисунок 5. Схема данных АИС «Агентства недвижимости»

ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1. АНАЛИЗ И ВЫБОР СУБД

Современные СУБД в основном являются приложениями Windows, так как данная среда позволяет более полно использовать возможности персональной ЭВМ, нежели среда DOS. Снижение стоимости высокопроизводительных персональных компьютеров обусловило не только широкий переход к среде Windows, где разработчик программного обеспечения может в меньшей степени заботиться о распределении ресурсов, но также сделал программное обеспечение ПК в целом и СУБД в частности менее критичными к аппаратным ресурсам электронно-вычислительной машины.

Среди наиболее ярких представителей систем управления базами данных можно отметить: Lotus Approach, Microsoft Access, Borland dBase, Borland Paradox, Microsoft Visual FoxPro, Microsoft Visual Basic, а также базы данных Microsoft SQL Server и Oracle, используемые в приложениях, построенных по технологии “клиент-сервер”. Фактически, у любой современной СУБД существует аналог, выпускаемый другой компанией, имеющий аналогичную область применения и возможности, любое приложение способно работать со многими форматами представления данных, осуществлять экспорт и импорт данных благодаря наличию большого числа конвертеров. Общепринятыми, также, являются технологии, позволяющие использовать возможности других приложений, например, текстовых процессоров, пакетов построения графиков и т.п., и встроенные версии языков высокого уровня (чаще – диалекты SQL и/или VBA) и средства визуального программирования интерфейсов разрабатываемых приложений. Поэтому уже не имеет существенного значения на каком языке и на основе какого пакета написано конкретное приложение, и какой формат данных в нем используется. Более того, стандартом “де-факто” стала

“быстрая разработка приложений” или RAD (от английского Rapid Application Development), основанная на широко декларируемом в литературе “открытом подходе”, то есть необходимость и возможность использования различных прикладных программ и технологий для разработки более гибких и мощных систем обработки данных. Поэтому в одном ряду с “классическими” СУБД все чаще упоминаются языки программирования Visual Basic 4.0 и Visual C++, которые позволяют быстро создавать необходимые компоненты приложений, критичные по скорости работы, которые трудно, а иногда невозможно разработать средствами “классических” СУБД. Современный подход к управлению базами данных подразумевает также широкое использование технологии “клиент-сервер”.

Таким образом, на сегодняшний день разработчик не связан рамками какого-либо конкретного пакета, а в зависимости от поставленной задачи может использовать самые разные приложения. Поэтому, более важным представляется общее направление развития СУБД и других средств разработки приложений в настоящее время.

Рассмотрим более подробно программные продукты компании Microsoft, а именно Visual FoxPro 3.0, Paradox, Visual Basic 4.0, Visual C++, Access 7.0.

Наиболее интересной чертой этих пакетов являются их большие возможности интеграции, совместной работы и использования данных, так как данные пакеты являются продуктами одного производителя, а также используют сходные технологии обмена данными.

FoxPro (фирма Fox Software) обладала исключительно высокими скоростными характеристиками и в этом отношении заметно выделялась среди интерпретирующих систем. Сравнительно с dBaseIV ее скорость в несколько раз выше и не уступает скорости систем-компиляторов. Практически по всем показателям Fox-программы работают значительно быстрее Clipper-программ. (Напоминаем - речь пока о версии для DOS'а.) Набор команд и функций, предлагаемых разработчиками FoxPro, по мощи и

гибкости отвечает любым требованиям к представлению и обработке данных. Может быть реализован максимально удобный и эффективный пользовательский интерфейс. В FoxPro поддерживаются разнообразные всплывающие и многоуровневые меню, работа с окнами и мышью, реализованы функции низкоуровневого доступа к файлам, управление цветами, настройками принтера, данные могут быть представлены в виде «электронных таблиц» и много еще приятностей и удобностей. В «довиндовскую» эпоху FoxPro был самой быстрой, самой удобной и самой мощной СУБД для компьютеров стандарта IBM PC.

версии 3.0 – процессор 468DX, Windows 3.1, 95, NT, объем оперативной памяти 8 (12) Мб, занимаемый объем на ЖМД 15-80 Мб, а для Visual FoxPro версии 5.0 (выпущена в 1997 году) – Windows 95 или NT, 486 с тактовой частотой 50 МГц, 10 Мб ОЗУ, от 15 до 240 Мб на ЖМД.

Paradox был разработан компанией Ansa Software, и первая его версия увидела свет в 1985 году. Этот продукт был впоследствии приобретен компанией Borland. С июля 1996 года он принадлежит компании Corel и является составной частью Corel Office Professional. В конце 80-х - начале 90-х годов Paradox, принадлежавший тогда компании Borland International, был весьма популярной СУБД, в том числе и в нашей стране, где он одно время занимал устойчивые позиции на рынке средств разработки настольных приложений с базами данных.

Принцип хранения данных в Paradox сходен с принципами хранения данных в dBase - каждая таблица хранится в своем файле (расширение *.db), MEMO- и BLOB-поля хранятся в отдельном файле (расширение *.md), как и индексы (расширение *.rx).

Однако, в отличие от dBase, формат данных Paradox не является открытым, поэтому для доступа к данным этого формата требуются специальные библиотеки. Например, в приложениях, написанных на C или Pascal, использовалась некогда популярная библиотека Paradox Engine, ставшая основой Borland Database Engine. Эта библиотека используется ныне

в приложениях, созданных с помощью средств разработки Borland (Delphi, C++Builder), в некоторых генераторах отчетов (например, Crystal Reports) и в самом Paradox. Существуют и ODBC-драйверы к базам данных, созданным различными версиями этой СУБД.

Отметим, однако, что отсутствие «открытости» формата данных имеет и свои достоинства. Так как в этой ситуации доступ к данным осуществляется только с помощью «знающих» этот формат библиотек, простое редактирование подобных данных по сравнению с данными открытых форматов типа dBase существенно затруднено. В этом случае возможны такие недоступные при использовании «открытых» форматов данных сервисы, как защита таблиц и отдельных полей паролем, хранение некоторых правил ссылочной целостности в самих таблицах - все эти сервисы предоставляются Paradox, начиная с первых версий этой СУБД.

По сравнению с аналогичными версиями dBase ранние версии Paradox обычно предоставляли разработчикам баз данных существенно более расширенные возможности, такие как использование деловой графики в DOS-приложениях, обновление данных в приложениях при многопользовательской работе, визуальные средства построения запросов, на основе интерфейса QBE - Query by Example (запрос по образцу), средства статистического анализа данных, а также средства визуального построения интерфейсов пользовательских приложений с автоматической генерацией кода на языке программирования PAL (Paradox Application Language).

Windows-версии СУБД Paradox, помимо перечисленных выше сервисов, позволяли также манипулировать данными других форматов, в частности dBase и данными, хранящимися в серверных СУБД. Такую возможность пользователи Paradox получили благодаря использованию библиотеки Borland Database Engine и драйверов SQL Links. Это позволило использовать Paradox в качестве универсального средства управления различными базами данных (существенно облегченная версия Paradox 7 под названием Database Desktop по-прежнему входит в состав Borland Delphi и

Borland C++Builder именно с этой целью). Что же касается базового формата данных, используемого в этом продукте, то он обладает теми же недостатками, что и все форматы данных настольных СУБД, и поэтому при возможности его стараются заменить на серверную СУБД, даже сохранив сам Paradox как средство разработки приложений и манипуляции данными.

Текущая версия данной СУБД - Paradox 9, поставляется в двух вариантах - Paradox 9 Standalone Edition и Paradox 9 Developer's Edition. Первый из них предназначен для использования в качестве настольной СУБД и входит в Corel Office Professional, второй - в качестве как настольной СУБД, так и средства разработки приложений и манипуляции данными в серверных СУБД. Обе версии содержат:

- Средства манипуляции данными Paradox и dBase.
- Средства создания форм, отчетов и приложений.
- Средства визуального построения запросов.
- Средства публикации данных и отчетов в Internet и создания Web-клиентов.
- Corel Web-сервер.
- ODBC-драйвер для доступа к данным формата Paradox из Windows-приложений.
- Средства для доступа к данным формата Paradox из Java-приложений.
- Run-time-версию Paradox для поставки вместе с приложениями.
- Средства создания дистрибутивов.

Драйверы SQL Links для доступа к данным серверных СУБД.

Access - в переводе с английского означает “доступ”. MS Access - это функционально полная реляционная СУБД. Кроме того, MS Access одна из самых мощных, гибких и простых в использовании СУБД. В ней можно создавать большинство приложений, не написав ни единой строки программы, но если нужно создать нечто очень сложное, то на этот случай

MS Access предоставляет мощный язык программирования - Visual Basic Application.

Популярность СУБД Microsoft Access обусловлена следующими причинами:

- Access является одной из самых легкодоступных и понятных систем как для профессионалов, так и для начинающих пользователей, позволяющая быстро освоить основные принципы работы с базами данных;
- система имеет полностью русифицированную версию;
- полная интегрированность с пакетами Microsoft Office: Word, Excel, Power Point, Mail;
- идеология Windows позволяет представлять информацию красочно и наглядно;
- возможность использования OLE технологии, что позволяет установить связь с объектами другого приложения или внедрить какие-либо объекты в базу данных Access;
- технология WYSIWIG позволяет пользователю постоянно видеть все результаты своих действий;
- широко и наглядно представлена справочная система;
- существует набор “мастеров” по разработке объектов, облегчающий создание таблиц, форм и отчетов.

Предназначен для создания отчетов произвольной формы на основании различных данных и разработки некоммерческих приложений. Минимальные ресурсы ПК: процессор 468DX, Windows 3.1, 95, NT, объем оперативной памяти 12 (16) Мб, занимаемый объем на ЖМД 10-40 Мб.

Visual Basic – это универсальный объектно-ориентированный язык программирования, диалекты которого встроены в Access, Visual FoxPro. Преимущества: универсальность, возможность создания компонентов OLE, невысокие требования к аппаратным ресурсам ЭВМ. Применяется для создания приложений средней мощности, не связанных с большой интенсивностью обработки данных, разработки компонентов OLE,

интеграция компонентов Microsoft Office. Минимальные ресурсы ПК: процессор 368DX, Windows 3.1, 95, NT, объем оперативной памяти 6 (16) Мб, занимаемый объем на ЖМД 8-36 Мб.

Указанные программные продукты имеют возможности визуального проектирования интерфейса пользователя, то есть разработчик из готовых фрагментов создает элементы интерфейса, программирует только их изменения в ответ на какие-либо события.

Компания Borland всегда была широко известна профессиональным разработчикам как фирма, предлагающая компиляторы C и Pascal, систему управления базами данных Paradox. Имея по всему миру около шести миллионов пользователей, dBASE остается индустриальным стандартом, применимым к различным операционным платформам, среди которых MS-DOS, UNIX, VAX/VMS и MS-Windows. Продукты, развиваемые в классе языков программирования - Borland C++ 4.5 и Delphi - с уникальным сочетанием классических принципов и современной технологии.

Совершенно новый продукт Borland Delphi for Windows - система скоростной разработки приложений, основанная на объектно-ориентированном Паскале. Delphi объединяет визуальные средства быстрой разработки приложений, высокопроизводительный компилятор объектно-ориентированного языка, масштабируемый механизм доступа к данным и другие последние достижения в области компьютерных технологий.

Visual C++ – наиболее мощный объектно-ориентированный язык программирования, обладает неограниченной функциональностью. Предназначен для создания компонентов приложений для выполнения операций, критичных по скорости.

В качестве СУБД используется Microsoft Access 2013 т.к. в Access в полной мере реализовано управление реляционными базами данных. Система поддерживает первичные и внешние ключи и обеспечивает целостность данных на уровне ядра (что предотвращает несовместимые операции обновления или удаления данных). Кроме того, таблицы в Access

снабжены средствами проверки допустимости данных, предотвращающими некорректный ввод вне зависимости от того, как он осуществляется, а каждое поле таблицы имеет свой формат и стандартные описания, что существенно облегчает ввод данных. Access поддерживает все необходимые типы полей, в том числе текстовый, числовой, счетчик, денежный, дата/время, MEMO, логический, гиперссылка и поля объектов OLE. Если в процессе специальной обработки в полях не оказывается никаких значений, система обеспечивает полную поддержку пустых значений.

3.2. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ В СУБД

Цель физического проектирования – преобразование логической модели с учетом синтаксиса, семантики и возможностей выбранной целевой СУБД.

Физическая модель БД определяет способ размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне. Исторически первыми системами хранения и доступа были файловые структуры и системы управления файлами (СУФ), которые фактически являлись частью операционных систем. СУБД создавала над этими файловыми моделями свою надстройку, которая позволяла организовать всю совокупность файлов таким образом, чтобы она выглядела как единое целое и получала централизованное управление от СУБД. Однако непосредственный доступ осуществлялся на уровне файловых команд, которые СУБД использовала при манипуляции всеми файлами, составляющие хранимые данные одной или нескольких баз данных.

С учетом типов данных и ограничений, принятых в MS Access, опишем требования к таблицам.

Таблица 1

Таблица Сотрудники

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
Табельный номер	Счётчик	Размер поля	Длинное целое
		Индексированное поле	Да (совпадения не допускаются)
Фамилия	Текстовый	Размер поля	25
		Обязательное поле	Нет
Имя	Текстовый	Размер поля	10
		Обязательное поле	Нет
Отчество	Текстовый	Размер поля	15
		Обязательное поле	Нет
Дата рождения	Дата и время	Размер поля	Краткий формат даты
		Индексированное поле	Нет
Должность	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет

Таблица 2

Таблица Должность

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
Код должности	Счётчик	Размер поля	Длинное целое
		Индексированное поле	Да (совпадения не допускаются)
Должность	Текстовый	Размер поля	30
		Обязательное поле	Нет

Таблица 3

Таблица Тип сделки

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
Код типа	Счётчик	Размер поля	Длинное целое
		Индексированное поле	Да (совпадения не допускаются)
Тип сделки	Текстовый	Размер поля	25
		Обязательное поле	Нет

Таблица 4

Таблица Тип объекта

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
Код объекта	Счётчик	Размер поля	Длинное целое
		Индексированное поле	Да (совпадения не допускаются)
Тип объекта	Текстовый	Размер поля	55
		Обязательное поле	Нет

Таблица 5

Таблица Улица

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
Код улицы	Счётчик	Размер поля	Длинное целое
		Индексированное поле	Да (совпадения не допускаются)
Улица	Текстовый	Размер поля	35
		Обязательное поле	Нет

Таблица 6

Таблица Город

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
Код города	Счётчик	Размер поля	Длинное целое
		Индексированное поле	Да (совпадения не допускаются)
Город	Текстовый	Размер поля	35
		Обязательное поле	Нет

Таблица 7

Таблица Клиенты

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
Код клиента	Счётчик	Размер поля	Длинное целое
		Индексированное поле	Да (совпадения не допускаются)
Фамилия	Текстовый	Размер поля	35
		Обязательное поле	Нет
Имя	Текстовый	Размер поля	10
		Обязательное поле	Нет
Отчество	Текстовый	Размер поля	25
		Обязательное поле	Нет
Дата рождения	Дата и время	Размер поля	Краткий формат даты
		Индексированное поле	Нет
Город	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Улица	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
№ дома	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
№ квартиры	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет

Контактный телефон	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет

Таблица 8

Таблица Недвижимость

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
Код_объекта	Счетчик	Размер поля	Длинное целое
		Индексированное поле	Да (совпадения не допускаются)
Объект	Текстовый	Размер поля	35
		Обязательное поле	Нет
Город	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Улица	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
№_строения	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
№_квартиры	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Площадь	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Описание	Текстовый	Размер поля	255
		Обязательное поле	Нет
Сдача_в_аренду	Логический	Формат поля	Да/Нет
		Индексированное поле	Нет

Таблица Договор

Имя поля	Тип данных	Свойства полей	
		Свойство	Значение
1	2	3	4
№_договора	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Код_объекта	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Код_клиента	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Тип_сделки	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Срок_аренды	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Стоимость_сделки	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Код_сотрудника	Числовой	Размер поля	Длинное целое
		Индексированное поле	Нет
Дата_оформление договора	Дата и время	Размер поля	Краткий формат даты
		Индексированное поле	Нет

3.3. РАЗРАБОТКА ПРЕДСТАВЛЕНИЙ

Выбор структуры диалога — это первый из этапов, который должен быть выполнен при разработке интерфейса. Рассмотренные ниже четыре варианта структуры диалога являются разновидностями структуры типа «вопрос — ответ», тем не менее каждая из них имеет свои особенности и наиболее удобна для определенного класса задач.

Развитие диалога во времени можно рассматривать как последовательность переходов системы из одного состояния в другое. Очевидно, что ни одно из этих состояний не должно быть «тупиковым», т.е. пользователь должен иметь возможность перейти из любого текущего состояния диалога в требуемое (за один или несколько шагов). Для этого в ходе разработки интерфейса необходимо определить все возможные состояния диалога и пути перехода из одного состояния в другое. Другими словами, необходимо разработать сценарий диалога.

MS Access позволяет организовывать удобный и интуитивно понятный интерфейс пользователя для работы с данными с помощью форм. Формами называются настраиваемые диалоговые окна, сохраняемые в базе данных в виде объектов специального типа. Формы содержат так называемые элементы управления, с помощью которых осуществляется доступ к данным в таблицах. Элементами управления являются текстовые поля для ввода и правки данных, кнопки, флажки, переключатели, списки, надписи, а также рамки объектов для отображения графики. Создание форм, содержащих необходимые элементы управления, существенно упрощает процесс ввода данных и позволяет предотвратить ошибки. Формы позволяют выполнять проверку корректности данных при вводе, проводить вычисления, и обеспечивают доступ к данным в связанных таблицах с помощью подчиненных форм.

Главная кнопочная форма создается с целью навигации по базе данных, т.е. она может использоваться в качестве главного меню БД. Элементами главной кнопочной формы являются объекты форм и отчетов.

Для одной базы данных можно создать несколько кнопочных форм. Кнопки следует группировать на страницах кнопочной формы таким образом, чтобы пользователю было понятно, в каких кнопочных формах можно выполнять определенные команды (запросы, отчеты, ввода и редактирования данных). Необходимо отметить, что на подчиненных

кнопочных формах должны быть помещены кнопки возврата в главную кнопку форму.

Из основного окна (рис.5) возможен переход к окну «Справочники» (рис.6), а также переход к окну «Операции» (рис. 7), «Запросы» (рис.8) и «Отчеты» (рис. 9).



Рисунок 5. Главная форма.

Основное окно состоит из следующих кнопок:

- Справочники
- Операции
- Запросы
- Отчеты
- Выход

При выборе меню «Справочники» пользователю будет доступна форма, представленная в соответствии с рисунком 6.

СПРАВОЧНИКИ

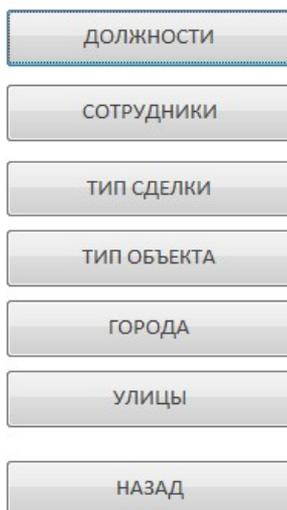


Рисунок 6. Подчиненная форма «01_Справочники»

Форма «Справочники» содержит управляющее меню, состоящее из:

- Должность;
- Сотрудники;
- Тип сделки;
- Тип объекта;
- Город;
- Улица;
- Кнопка «Назад»

При выборе меню «Операции» пользователю будет доступна форма, представленная в соответствии с рисунком 7.

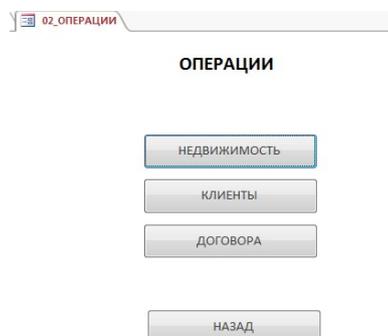


Рисунок 7. Подчиненная форма «02_Операции»

Форма «Операции» содержит управляющее меню, состоящее из:

- Недвижимость;
- Клиенты;
- Договора;
- Кнопка «Назад»

При выборе меню «Запросы» пользователю будет доступна форма, представленная в соответствии с рисунком 8.

03_ЗАПРОСЫ

**ЗАПРОСЫ / ПОИСК
ИНФОРМАЦИИ**

Кол-во договоров по типам объектов

Кол-во типов сделок за указанный период

Перечень договоров оформленный в
указанный период

Перечень клиентов арендуемые жилье

НАЗАД

Рисунок 8. Подчиненная форма «03_Запросы»

При выборе меню «Отчеты» пользователю будет доступна форма, представленная в соответствии с рисунком 9.

04_ОТЧЕТЫ

ОТЧЕТЫ

Кол-во договоров по типам объектов

Кол-во типов сделок за указанный период

Перечень договоров оформленный в
указанный период

Перечень клиентов арендуемые жилье

НАЗАД

Рисунок 9. Подчиненная форма «04_Отчеты»

3.4. РАЗРАБОТКА ФОРМ

Формы в Microsoft Access создают для того, чтобы облегчить ввод и редактирование данных, обеспечить их вывод в удобном для пользователя представлении. С помощью подобных объектов можно делать доступной только часть данных, автоматически выбирать информацию из связанных таблиц, вычислять значения выражений и т. д.

Формы — это объекты, предназначенные, в основном, для ввода и отображения данных на экране, хотя они могут быть распечатаны и содержать так называемые элементы управления, такие как поля, списки, флажки, переключатели и др.

В формы можно помещать командные кнопки для открытия других форм, выполнения запросов или команд меню, фильтрации выводимых на экран данных, организации вывода сообщений или печати информации (в частности, можно установить разные наборы опций для вывода формы на экран и на печать). Таким образом, формы позволяют управлять ходом выполнения приложения и являются основным средством организации интерфейса пользователя в Microsoft Access.

Записи базы данных (БД) можно просматривать и редактировать в виде таблицы или в виде формы. Представление БД в виде таблицы позволяет наблюдать сразу несколько записей одновременно, однако часто вид Таблица не позволяет полностью видеть всю информацию на экране. Формы позволяют забыть о неудобствах, возникающих при работе с таблицами, имеющими большое количество полей. В режиме Форма можно все внимание уделить одной записи, не пользуясь прокруткой для поиска нужного поля в длинной череде столбцов таблицы, а имея перед глазами всю информацию.

Чтобы создать форму, необходимо в окне БД выбрать в списке Создание, необходимо выбрать режим создания формы, а также таблицу (в

раскрывающемся списке), которая будет использоваться в качестве источника данных.

Основными режимами создания любой формы являются Конструктор и Мастер форм. Рассмотрим создание формы с помощью режима Мастер форм.

С помощью мастера можно создавать формы на основе одной таблицы и более сложные формы на основе нескольких таблиц и запросов. Намного проще и быстрее создавать формы с помощью мастера, а затем усовершенствовать их в режиме Конструктора. Чтобы лучше представить, как создавать формы с помощью Мастера форм, опишем процедуру создания формы. Эта форма предназначена для оформления товарного чека. Для создания этой формы:

1. Выбираем *Другие формы* — *Мастер форм*.
2. Появилось первое диалоговое окно. В поле со списком *Таблицы и запросы* отображены имена всех таблиц и запросов базы данных, которые могут использоваться в качестве источника данных для форм. Раскрыв этот список, мы выбираем запрос «Расписка».
3. В списке *Доступные поля* этого диалогового окна отображаются все поля выбранной таблицы или запроса. Чтобы добавить в создаваемую форму нужные поля, мы нажимаем кнопку «>». Нажимаем кнопку *Далее* для отображения второго диалогового окна *Мастера форм*.
4. Во втором диалоговом окне мастера можно определить вид формы. После выбора подходящего режима отображения данных в форме мы нажимаем кнопку *Далее* для отображения следующего диалогового окна.
5. Третье диалоговое окно *Мастера форм* предназначено для выбора стиля оформления новой формы. *Мастер* предлагает несколько стилей оформления. Мы выбираем стиль и нажимаем кнопку *Далее*.
6. В последнем диалоговом окне *Мастера форм* требуется указать название формы. Чтобы отобразить созданную мастером форму в режиме *Формы*, выбираем переключатель *Открыть форму* для просмотра и ввода данных и нажимаем кнопку *Готово*.

В результате мастером будет создана форма в соответствии с выбранными параметрами и сохранена с указанным именем, затем эта форма будет открыта в заданном режиме. После редактируем форму в режиме Конструктора.

Форма "Должность"

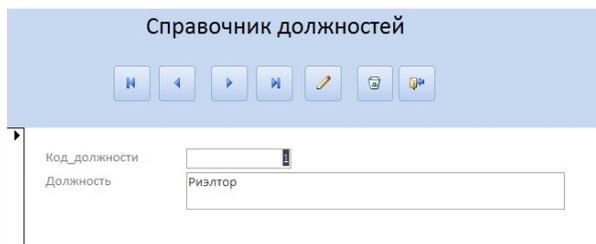


Рисунок 10 – Форма «Должность»

Форма «Сотрудники»

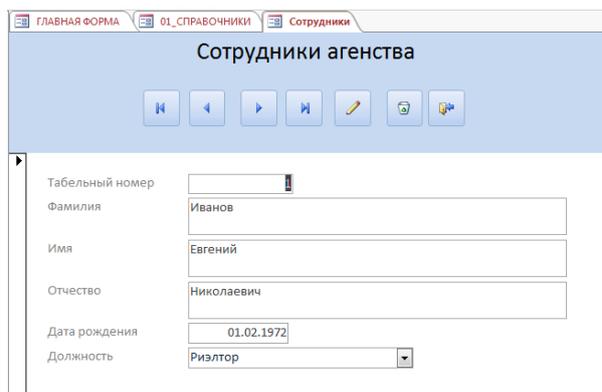


Рисунок 11 – Форма «Сотрудники»

Форма «Тип сделки»

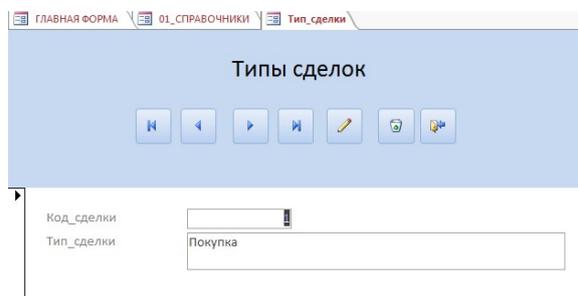


Рисунок 12 – Форма «Тип сделки»

Форма «Тип объекта»

Скриншот формы «Типы объектов». В заголовке окна «Типы объектов» расположены кнопки для навигации и редактирования. В основной области формы есть поле «Код_типа» и поле «Тип_объекта» со значением «1-комнатная квартира».

Рисунок 13 – Форма «Тип объекта»

Форма «Города»

Скриншот формы «Справочник городов». В заголовке окна «Справочник городов» расположены кнопки для навигации и редактирования. В основной области формы есть поле «Код_города» и поле «Город» со значением «Москва».

Рисунок 14 – Форма «Города»

Форма «Улицы»

Скриншот формы «Справочник улиц». В заголовке окна «Справочник улиц» расположены кнопки для навигации и редактирования. В основной области формы есть поле «Код_улицы» и поле «Улица» со значением «Мира».

Рисунок 15 – Форма «Улицы»

Форма «Недвижимость»

Скриншот формы «Список объектов агенства». В заголовке окна «Список объектов агенства» расположены кнопки для навигации и редактирования. В основной области формы есть поля для ввода данных: «Код_объекта», «Объект», «Город» (Уфа), «Улица» (Молодежная), «№_строения» (15), «№_квартиры» (1), «Площадь» (150), «Описание» и «Сдача_в_аренду» (отмечено галочкой).

Рисунок 16 – Форма «Недвижимость»

Форма «Клиенты»

Код_клиента	
Фамилия	Патрикеева
Имя	Светлана
Отчество	Васильевна
Дата рождения	20.05.1963
Город	Уфа
Улица	Ленина
№_дома	15
№_квартиры	1
Контактный телефон	89177609043

Рисунок 17 – Форма «Клиенты»

Форма «Договора» пользователю будет доступна форма, представленная в соответствии с рисунком 18.

№_договора	
Код_объекта	1-комнатная квартира
Код_клиента	Патрикеева
Тип_сделки	Сдача в аренду
Срок_аренды	12
Стоимость_сделки	72000
Код_сотрудника	Иванов
Дата_оформление_догов	01.06.2017

Рисунок 18 – Форма «Договора»

3.5 РАЗРАБОТКА ОТЧЕТОВ

Отчет – это форматированное представление данных, которое выводится на экран, в печать или файл. Они позволяют извлечь из базы нужные сведения и представить их в виде, удобном для восприятия, а также предоставляют широкие возможности для обобщения и анализа данных.....

В MS Access можно создавать отчеты различными способами:

- Конструктор

- Мастер отчетов
- Автоотчет: в столбец
- Автоотчет: ленточный
- Мастер диаграмм
- Почтовые наклейки

Для решения поставленных Заказчиком задач и на основании имеющихся запросов нами были сформированы следующие виды отчетов.

Отчеты были созданы при помощи "Мастера отчетов"

Мастер позволяет создавать отчеты с группировкой записей и представляет собой простейший способ создания отчетов. Он помещает выбранные поля в отчет и предлагает шесть стилей его оформления. После завершения работы Мастера полученный отчет можно доработать в режиме Конструктора. Воспользовавшись функцией Автоотчет, можно быстро создавать отчеты, а затем вносить в них некоторые изменения.

Для создания Автоотчета необходимо выполнить следующие действия:

1. В окне базы данных щелкнуть на вкладке Отчеты и затем щелкнуть на кнопке Создать. Появится диалоговое окно Новый отчет.
2. Выделить в списке пункт Автоотчет: в столбец или Автоотчет: ленточный.
3. В поле источника данных щелкнуть на стрелке и выбрать в качестве источника данных таблицу или запрос.
4. Щелкнуть на кнопке ОК.
5. Мастер автоотчета создает автоотчет в столбец или ленточный (по выбору пользователя), и открывает его в режиме Предварительного просмотра, который позволяет увидеть, как будет выглядеть отчет в распечатанном виде.
6. В меню Файл щелкнуть на команде Сохранить. В окне Сохранение в поле Имя отчета указать название отчета и щелкнуть на кнопке ОК.

Отчет "Кол-во договоров по типам объектов"

The screenshot shows a report design interface with a grid layout. At the top, there are tabs for '04_ОТЧЕТЫ', '03_ЗАПРОСЫ', and the active report 'Кол-во договоров по типам объектов'. Below the tabs is a horizontal ruler with numbers 1 through 20. The report design is organized into several sections:

- Заголовок отчета:** A large blue-shaded header cell containing the text 'Кол-во договоров по типам объектов'.
- Верхний колонтитул:** A row of fields containing 'Код_типа', 'Тип_объекта', and 'Count-№_договора'.
- Область данных:** A row of fields containing 'Код_типа', 'Тип_объекта', and 'Count-№_договора'.
- Нижний колонтитул:** A row of fields containing '=Now()' and a page number format string '="Стр. " & [Page] & " из " & [Pages]'.
- Примечание отчета:** A final empty row at the bottom.

Рисунок 19 – Отчет «Кол-во договоров по типам объектов»

Отчет " Кол-во типов сделок за указанный период"

Кол-во типов сделок за указанный период																		
Верхний колонтитул																		
Тип_сделки										Count-№_договора								
Область данных																		
Тип_сделки										Count-№_договора								
Нижний колонтитул																		
=Now()										="Стр. " & [Page] & " из " & [Pages]								
Примечание отчета																		

Рисунок 20 – Отчет «Кол-во типов сделок за указанный период»

Отчет " Перечень договоров оформленный в указанный период"

Перечень договоров оформленный в указанный период																															
Верхний колонтитул																															
№_договора				Код_объекта				Код_клиента				Тип_сделки				Срок_арен				Стоимость_сд				Код_сотрудника				Дата_оформлен			
Область данных																															
№_договора				Код_объекта				Код_клиента				Тип_сделки				Срок_арен				Стоимость_сд				Код_сотрудника				Дата_оформлен			
Нижний колонтитул																															
=Now()										="Стр. " & [Page] & " из " & [Pages]																					
Примечание отчета																															

Рисунок 21 – Отчет «Перечень договоров оформленный в указанный период»

Отчет " Перечень клиентов арендуемые жилье"

Перечень клиентов арендуемые жилье																											
Верхний колонтитул																											
Код_клиент				Фамилия								Имя								Отчество							
Область данных																											
Код_клиент				Фамилия								Имя								Отчество							
Нижний колонтитул																											
=Now()										="Стр. " & [Page] & " из " & [Pages]																	
Примечание отчета																											

Рисунок 22 – Отчет «Перечень клиентов арендуемые жилье»

3.6. РЕАЛИЗАЦИЯ ОГРАНИЧЕНИЙ, АВТОМАТИЗАЦИЯ ОБРАБОТКИ ДАННЫХ В БД

Запросы в Access являются основным инструментом выборки, обновления и обработки данных в таблицах базы данных. Access в соответствии с концепцией реляционных баз данных для выполнения запросов использует язык структурированных запросов SQL (Structured Query Language). С помощью инструкций языка SQL реализуется любой запрос в Access.

Основным видом запроса является запрос на выборку. Результатом выполнения этого запроса является новая таблица, которая существует до закрытия запроса. Записи формируются путем объединения записей таблиц, на которых построен запрос. Способ объединения записей таблиц указывается при определении их связи в схеме данных или при создании запроса. Условия отбора, сформулированные в запросе, позволяют фильтровать записи, составляющие результат объединения таблиц.

В Access может быть создано несколько видов запроса:

– запрос на выборку — выбирает данные из одной таблицы или запроса или нескольких взаимосвязанных таблиц и других запросов. Результатом является таблица, которая существует до закрытия запроса. Формирование записей таблицы результата производится в соответствии с заданными условиями отбора и при использовании нескольких таблиц путем объединения их записей;

– запрос на создание таблицы — выбирает данные из взаимосвязанных таблиц и других запросов, но, в отличие от запроса на выборку, результат сохраняет в новой постоянной таблице;

– запросы на обновление, добавление, удаление — являются запросами действия, в результате выполнения которых изменяются данные в таблицах.

Любой запрос хранится в базе данных в формате (на английском - Structured Query Language) расшифровывается как «язык структурированных запросов».

В реляционных базах данных язык запросов служит:

- для создания данных;
- для модификации данных;
- для управления данными.

Основное достоинство этого языка состоит в том, что он является стандартом для большинства реляционных СУБД.

Для решения задач, поставленных Заказчиком были созданы следующие запросы:

1. «Кол-во договоров по типам объектов»

Данный запрос позволяет вывести количество договоров по типам объектов

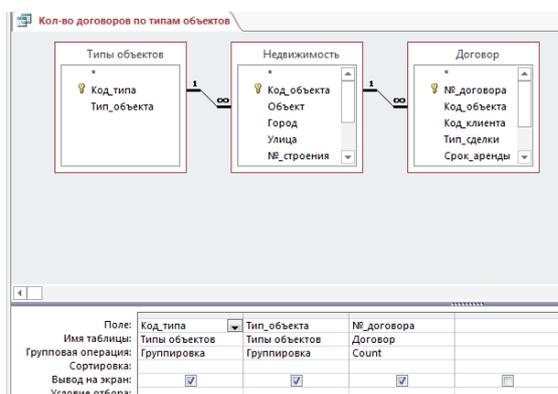


Рисунок 23. Запрос «Кол-во договоров по типам объектов» в режиме конструктора

Структура запроса на языке SQL:

```
SELECT [Типы объектов].Код_типа, [Типы  
объектов].Тип_объекта, Count(Договор.[№_договора]) AS [Count-  
№_договора]
```

```
FROM [Типы объектов] INNER JOIN (Недвижимость INNER  
JOIN Договор ON Недвижимость.Код_объекта = Договор.Код_объекта)  
ON [Типы объектов].Код_типа = Недвижимость.Объект  
GROUP BY [Типы объектов].Код_типа, [Типы объектов].Тип_объекта;
```

2. «Кол-во типов сделок за указанный период»

Данный запрос позволяет вывести количество сделок за указанный период

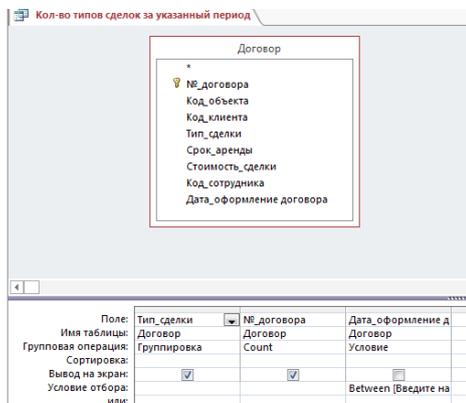


Рисунок 24. Запрос «Кол-во типов сделок за указанный период» в режиме конструктора

Структура запроса на языке SQL:

```
SELECT Договор.Тип_сделки, Count(Договор.[№_договора]) AS [Count-№_договора]
FROM Договор
WHERE (((Договор.[Дата_оформление_договора]) Between [Введите начало периода] And [Введите окончание периода]))
GROUP BY Договор.Тип_сделки;
```

3. «Перечень договоров оформленный в указанный период»

Данный запрос позволяет вывести перечень договоров указанных за указанный период

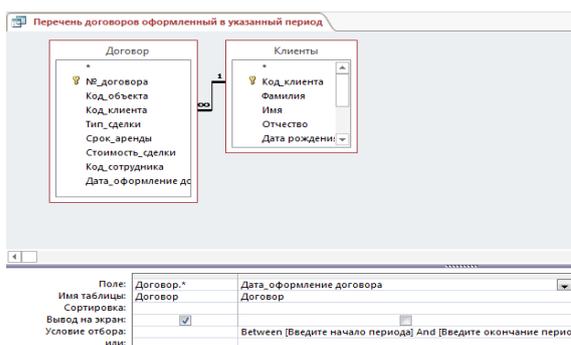


Рисунок 25. Запрос «Перечень договоров оформленный в указанный период» в режиме конструктора

Структура запроса на языке SQL:

```
SELECT Договор.*
FROM Клиенты INNER JOIN Договор ON Клиенты.Код_клиента =
Договор.Код_клиента
WHERE (((Договор.[Дата_оформление_договора]) Between [Введите
начало периода] And [Введите окончание периода]));
```

4. «Перечень клиентов арендуемые жилье»

Данный запрос позволяет вывести перечень клиентов которые арендуют жилье

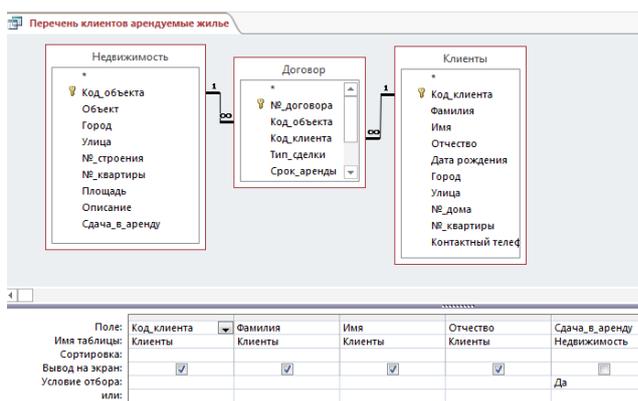


Рисунок 26. Запрос «Перечень клиентов арендуемые жилье» в режиме конструктора

Структура запроса на языке SQL:

```
SELECT Клиенты.Код_клиента, Клиенты.Фамилия, Клиенты.Имя,
Клиенты.Отчество
FROM Недвижимость INNER JOIN (Клиенты INNER JOIN Договор ON
Клиенты.Код_клиента = Договор.Код_клиента) ON
Недвижимость.Код_объекта = Договор.Код_объекта
WHERE (((Недвижимость.Сдача_в_аренду)=Yes));
```

3.7 БЕЗОПАСНОСТЬ И КОНТРОЛЬ

В базе данных реализовано разграничение доступа средствами СУБД Microsoft Access [5]. Доступ к БД осуществляется вводом логина и пароля сотрудника в форме авторизации (см. 2.3.2). Каждому пользователю в соответствии с его должностью предоставляется определенный набор прав доступа к объектам в базе данных. Это позволяет ограничить пользователям доступ к информации во избежание неправомерных действий. Также для защиты от НСД информация таблицы «Пользователи», содержащей информацию о паролях, хранится в виде результат хэш-функции.

Организована парольная защита данной базы средствами Microsoft Access. При попытке открыть базу, будет предложено ввести пароль (рис. 27).

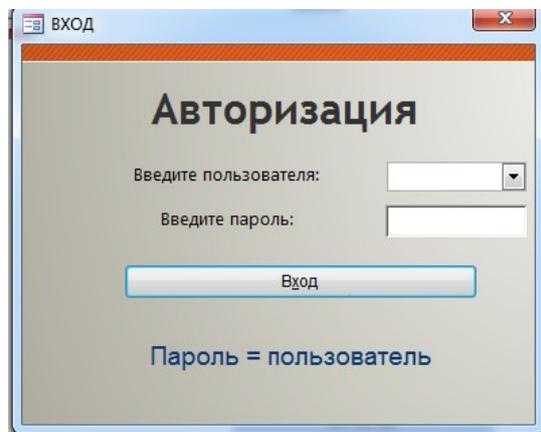


Рисунок 27 - Вход в базу данных

При введении неверного пароля будет выдана ошибка и будет повторно предложено ввести пароль.

Так же, следует упомянуть, что обойти окно авторизации весьма проблематично, так как, при попытке закрыть окно авторизации СУБД MS Access закрывается целиком, а обработка исключений сильно затруднит доступ к базе данным злоумышленникам.

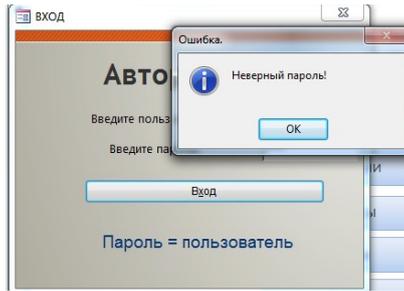


Рисунок 28 – Ошибка ввода пароля

В случае ввода правильного пароля автоматически откроется «Главная форма».

Для организации резервного копирования была использована сторонняя программа Handy Backup.

Данная программа может хранить данные (в том числе и нашу БД) в разных местах по выбору – облако, папка на компьютере, флеш–накопители или сервера (рисунке 29).

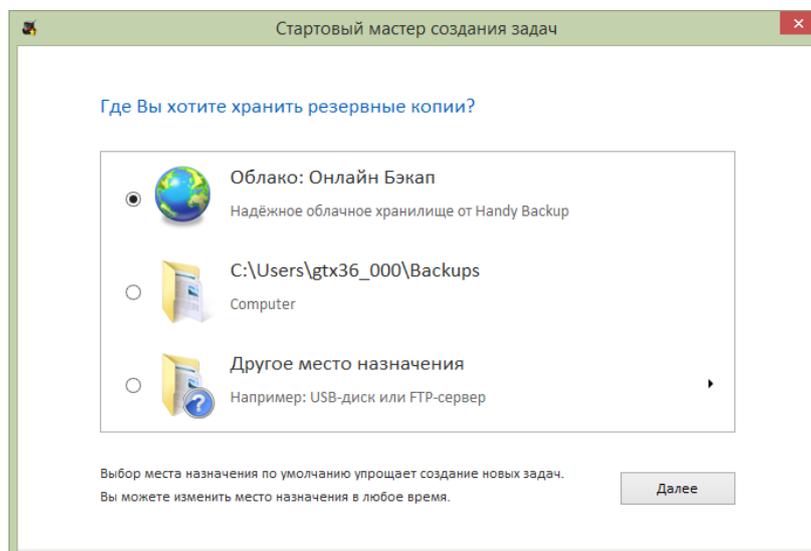


Рисунок 29 – Выбор места хранения резервной копии

Программа может как создавать резервные копии, так и восстанавливать данные из данных копий (рисунок 30).

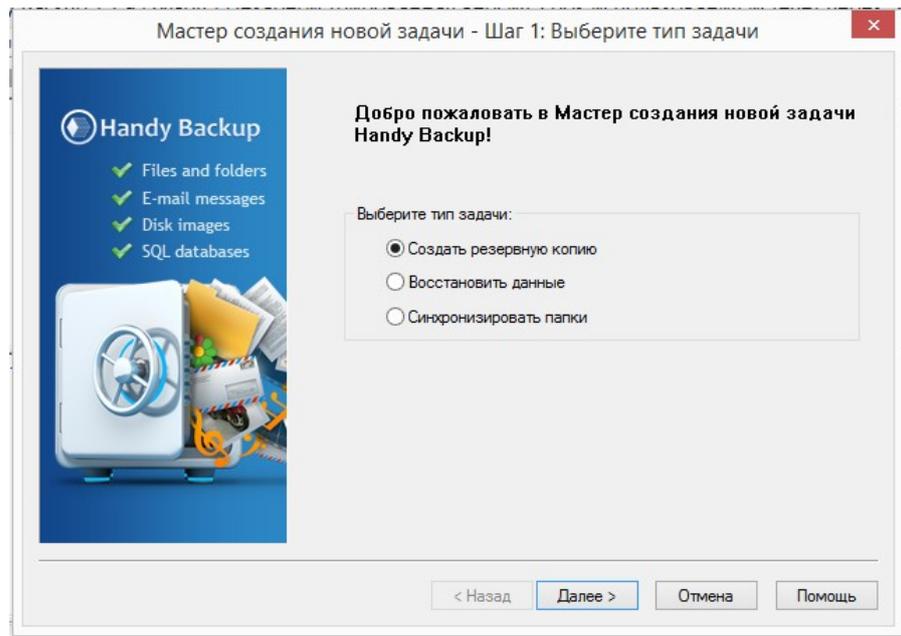


Рисунок 30 – Выбор задачи

Выбираем копирование базы данных, а затем место, куда будет сохраняться резервная копия.

Следующим шагом мы выбираем тип копирования – полное, инкрементное, дифференциальное либо смешанное (рисунок 31). Также мы можем выбрать, будет ли хранимая резервная копия единственной, или же будет храниться несколько резервных копий разных дат (рисунок 32). Можно выбрать также длительность хранения старых версий. Так как зачастую базы весят немало, хранить множество их резервных копий слишком затратно в плане объемов памяти.

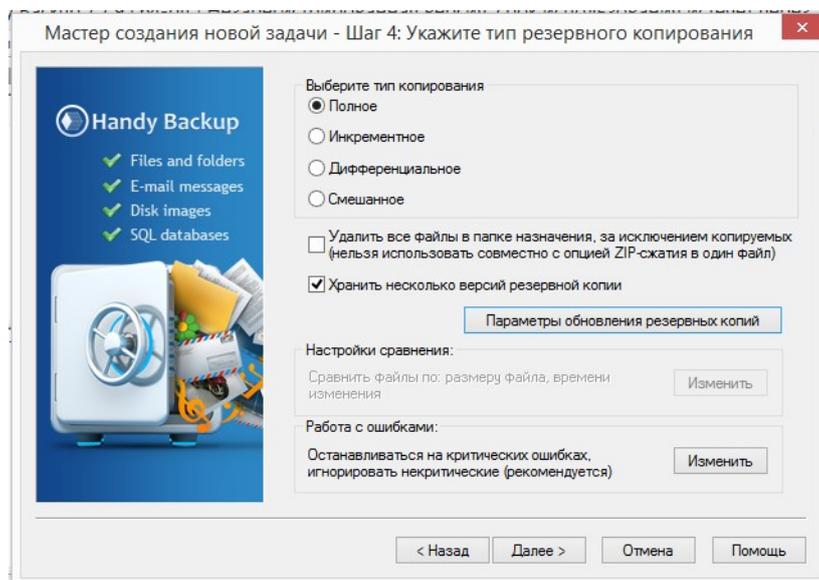


Рисунок 31 – Выбор типа копирования

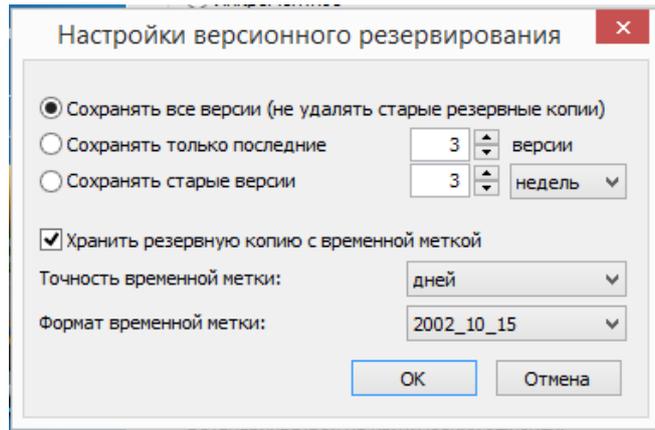


Рисунок 32 – Хранение различных версий копий

При больших объемах файл может быть сжат, а также дополнительно зашифрован паролем, чтобы злоумышленник не мог получить доступа к копиям (рисунок 33).

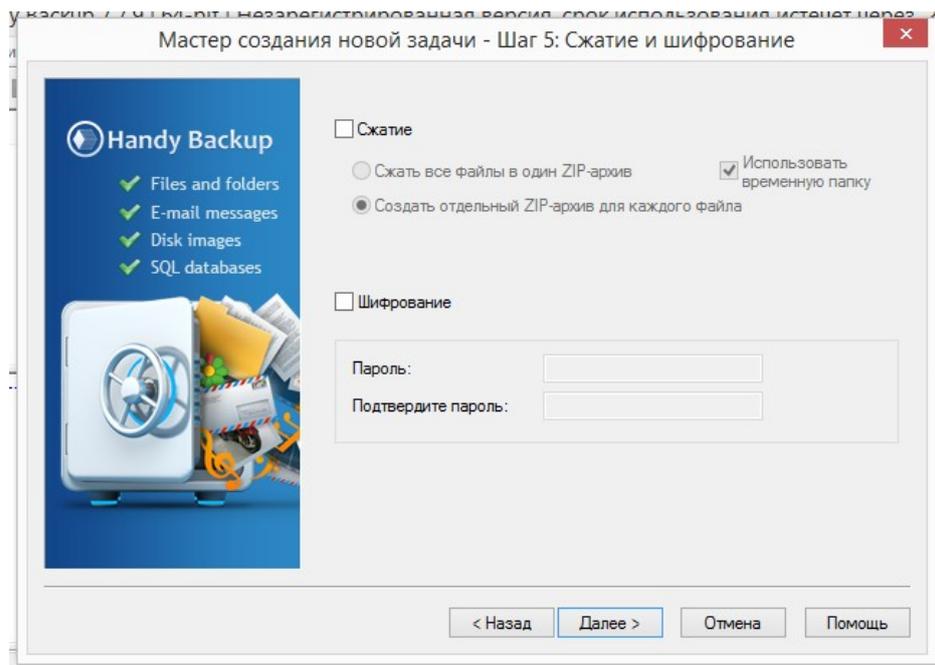


Рисунок 33 – Сжатие и шифрование

Одним из главных преимуществ данной программы является автоматическое резервное копирование (рисунок 34). Можно установить расписание по которому база будет копироваться, что будет являться защитой от человеческого фактора во время ручного копирования базы – администратор может просто забыть сделать вовремя резервную копию базы.

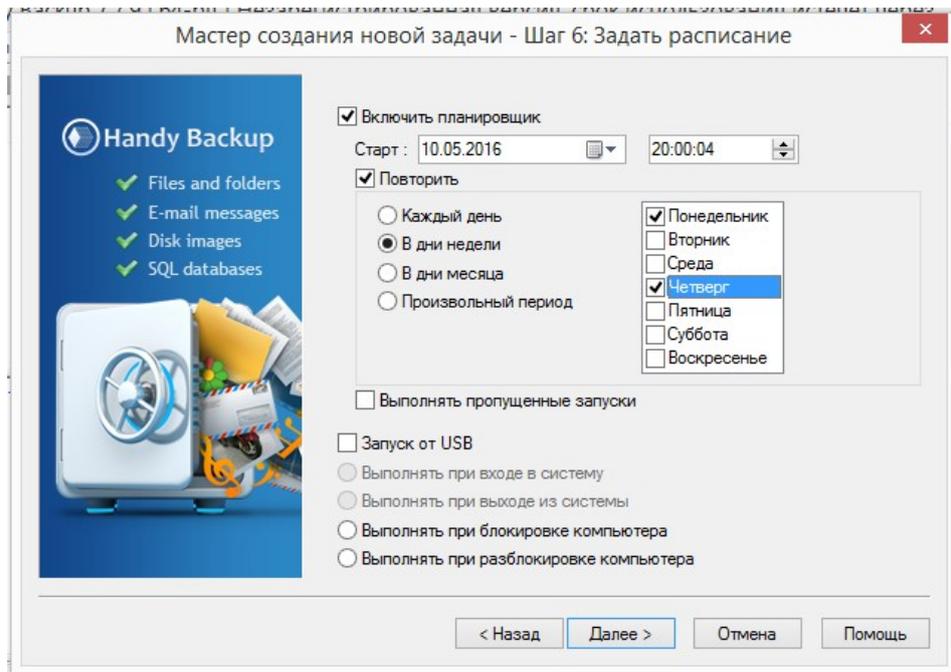


Рисунок 34 – Расписание копирования

Еще один плюс программы – она может сделать копию вне зависимости, работаем ли мы сейчас с базой данных или нет ("Горячее копирование").

ЗАКЛЮЧЕНИЕ

СУБД MS Access является в настоящее время одной из самых популярных среди настольных систем. Среди причин такой популярности следует отметить:

- удобство ввода и редактирования данных таблиц, т.к. программа создает интерфейс по выбору пользователя;
- производит поиск данных в таблицах по определенным критериям;
- контролирует ключевые поля;
- создает любые формы отчетов, в которых можно менять содержание и стиль оформления;
- дает возможность пополнять базу данных новыми таблицами и решать новые задачи, т.е. составлять новые отчеты и формы.

Все это позволяет считать Access надежной программой.

Автоматизированная информационная система «Агентство недвижимости» является достаточно простой в обращении и отвечает всем требованиям к базе данных:

- 1) Быстродействие.
- 2) Простота обновления данных.
- 3) Независимость данных.
- 4) Безопасность данных.
- 5) Дружественный интерфейс.

Основой баз данных являются таблицы, содержащие всю необходимую информацию. Отсутствует дублирование информации, что исключает избыточность данных. Уникальность записей и быстродействие базы данных достигнуто при помощи использования первичных ключей. Для целостности связанных данных, находящихся в разных таблицах, используются первичные и внешние ключи.

Для просмотра, ввода и редактирования данных, хранящихся в

таблицах, предусмотрено наличие форм, что облегчает работу с таблицами. Также в АИС «Агентство недвижимости» используются поисковые формы, которые облегчают поиск записей в таблицах. Для наглядного представления информации и вывода её на печать или в файл используются отчёты.

СПИСОК ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Григорьев, Ю.А. Банки данных: учебник для вузов / Ю.А. Григорьев, Г.И. Ревунков. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с.
2. Брешенков, А.В. Проектирование объектов баз данных в среде Access : учеб. пособие / А.М. Губарь, А.В. Брешенков. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2006. – 183 с.
3. Булдакова, Т.И. Технологии разработки баз данных [Электрон. ресурс] : учеб. пособие по дисциплине "Системы управления базами данных" / Т. И. Булдакова. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2012.
4. Белоус, В.В. Основы реляционных баз данных. Практикум по SQL [Электрон. ресурс] : метод. указ. к лаб. работам по курсу "Базы данных" / В.В. Белоус, Н.В. Пивоварова. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2014.
5. Кренке Д. Теория и практика построения баз данных: [пер.с англ] / Д. Кренке. – 9 - е изд. – СПб.: Питер, 2015. – 858 с.
6. Кригель, А. SQL. Библия пользователя / А. Кригель, Б. Труханов.–М.: Вильямс, 2010.–752 с.
7. Грабер, М. Справочное руководство по SQL /М. Грабер// ЛОРИ. – 2011. – 680 с.
8. Мейер, М. Теория реляционных баз данных/ М. Мейер.–М.: Мир, 2012. – 530 с.
9. Туманов, В.Е. Основы проектирования реляционных баз данных / В.Е. Туманов. – М.: Бином-Пресс, 2012. – 420 с.