Министерство образования и науки Российской Федерации

Федеральное агентство образования

Курский институт социального образования

(филиал)

Российского государственного социального университета

Кафедра информационных систем и ЕНД

**Курсовой проект**

по дисциплине «Автоматизированные системы организационно-административного управления»

на тему «АИС почтовое отделение»

КУРСК 2009

**СОДЕРЖАНИЕ:**

# ВВЕДЕНИЕ ……………………………………………………………………. 2

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ …………………………………… 4
   1. Описание предметной области ……………………………………... 4
   2. Анализ требований к базе данных ………………………………….. 6
2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ …………………………………. 9
   1. Методология проектирования …………………………………….… 10
      1. Метод нормальных форм …………………………………….. 13
      2. Метод сущность-связь ……………………………………….. 16
   2. Проектирование базы данных «Почтовые отделения» …………… 21
3. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ СРЕДСТВАМИ MS ACCESS ………. 25
   1. Обзор системы управления базами данных MS Access …………… 25
   2. Формирование исходных данных ………………………………….. 28
   3. Бизнес-план……………………………………………………………30
   4. Разработка объектов базы данных …………………………………35

3.5Построение модели в программе BPwin» ….………………..………..46

3.6Организационная структура…………………………………………….48

ЗАКЛЮЧЕНИЕ ……………………………………………………………….. 49

СПИСОК ЛИТЕРАТУРЫ …………………………………………………….. 51

###### ВВЕДЕНИЕ

В настоящие время в связи с развитием компьютерной техники появилась возможность автоматизировать многие процессы, также увеличился объем обрабатываемой информации. И возникла объективная необходимость автоматизировать большую часть сферы человеческой деятельности.

Для автоматизации обработки данных в начале 70-х годов были предложены программы, специально предназначенные для управления данными – системы управления базами данных (СУБД).

В самом общем смысле **база данных**– совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объекта и их взаимосвязей в рассматриваемой предметной области. В базе данных может храниться множество таблиц с однотипными записями. Один из типов баз данных - это документы, набранные с помощью текстовых редакторов и сгруппированные по темам. Другой тип - файлы электронных таблиц, объединяемые в группы по характеру их использования.

С ростом популярности СУБД в 70-80-х годах появилось множество различных моделей данных. У каждой из них имелись свои достоинства и недостатки, которые сыграли ключевую роль в развитии реляционной модели данных, появившейся во многом благодаря стремлению упростить и упорядочить первые модели данных.

В зависимости от способа установления связей между данными с компьютерно–ориентированным описанием на языке конкретной СУБД, разрабатывались различные модели логической организации данных: иерархическая, сетевая и реляционная.

Использование баз данных и информационных систем становится неотъемлемой составляющей деловой деятельности современного человека и функционирования преуспевающих организаций. В связи с этим большую актуальность приобретает освоение принципов построения и эффективного применения соответствующих технологий и программных продуктов: систем управления базами данных, CASE-систем автоматизации проектирования, средств администрирования и защиты баз данных и других.

От правильного набора инструментальных средств создания информационных систем, определения подходящей модели данных, обоснования рациональной схемы построения баз данных, организация запросов к хранимым данным и ряда других моментов во многом зависят эффективность функционирования разрабатываемых систем. Все это требует осознанного применения теоретических положений и инструментальных средств разработки баз данных и информационных систем.

Цель моей работы заключается в проектировании и разработке базы данных «Почтовые отделения». Разрабатываемая мною база данных может быть использована для создания единой информационной системы почтовых отделений. В ней можно будет отслеживать пересылку писем, бандеролей, закупку печатных изданий у типографий. Моя база данных будет ограничена закупкой печатных изданий у различных типографий. В базе данных будет отслеживаться информация об известных печатных изданиях, типографиях и почтовых отделениях. Создаваемая база данных облегчит поиск почтовым отделениям необходимых изданий, а также можно будет выбрать типографию, закупка у которой данного печатного издания будет наиболее выгодной. Далее созданную базу данных можно будет расширить для использования в других целях.

Достижение цели осуществляется посредством комплекса задач:

* проектирование и создание таблиц для хранения данных;
* ввод данных;
* разработка других элементов базы, предназначенных для просмотра, редактирования и вывода информации.

1. **АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Сфера деятельности почтовых отделений характеризуется большими массивами информации и объёмом выполняемых работ.

Задача почтовых отделений заключается в своевременной доставке газет, журналов, писем, телеграмм, бандеролей жителям своих районов. Для этого им необходима единая информационная система, в которой будет отслеживаться поступление и последующая доставка всех писем, телеграмм и бандеролей населению и различным организациям, а также доставка газет и журналов по подписке на потовых отделениях населению и организациям. Качество и своевременность доставки писем, телеграмм и бандеролей будет зависеть от того, насколько продуманы каналы обмена ими между почтовыми отделениями и непосредственная их доставка получателям. А доставка газет и журналов жителям и организациям, которые подписались на их получение, зависит от своевременного заказа необходимых газет и журналов на типографиях, оплаты и доставки их на почтовые отделения, а также от своевременной их доставки получателям.

Моя цель – разработать базу данных для почтовых служащих, работающих в единой системе почтовых отделений. Она должна отображать весь перечень газет и журналов, которые имеются в наличии, а также производство их на типографиях и заказы почтовых отделений газет и журналов у этих типографий с последующей оплатой и получением необходимых партий.

* 1. **Описание предметной области**

В базе данных должны храниться сведения о почтовых отделениях, газетах и журналах, поступающих в почтовое отделение, и типографиях, их выпускающих.

Сведения о почтовых отделениях должны включать в себя следующую информацию:

* уникальный шифр издания;
* название газеты или журнала;
* Ф.И.О. редактора газеты или журнала.

Сведения о типографиях, выпускающих данные газеты или журналы, должны содержать:

* уникальный номер типографии;
* адрес типографии;
* Ф.И.О. директора типографии.

Также в этой базе данных должны храниться сведения о заказах, сделанных почтовыми отделениями на определённые издания. Они должны включать:

* Уникальный номер заказа;
* Номер почтового отделения, делающего заказ;
* Номер типографии, у которой делают заказ;
* Код издания, на которое делают заказ;
* Количество заказов;
* Цена доставки изданий.

В базе данных должны содержаться сведения о тиражах выпускаемых изданий определёнными типографиями:

* Номер типографии;
* Код издания;
* Цена за один экземпляр издания;
* Код заказа;
* Тираж.

**1.2. Анализ требований к базе данных**

На информацию, хранящуюся в базе данных, накладываются следующие ограничения:

1. по изданиям:

* один и тот же человек не может быть редактором нескольких изданий одновременно;
* у нескольких изданий не может быть одинакового кода издания;

1. по типографиям:

* каждая типография должна иметь свой уникальный номер;
* несколько типографий не могут располагаться по одному почтовому адресу;
* один и тот же человек не может быть директором нескольких типографий одновременно;

1. по почтовым отделениям:

* каждое почтовое отделение должно иметь свой уникальный номер;
* несколько почтовых отделений не могут располагать по одному почтовому адресу;
* один и тот же человек не может быть директором нескольких почтовых отделений одновременно.

С базой данных должны работать следующие группы пользователей:

* редакторы изданий;
* служащие и начальники типографий;
* служащие и начальники почтовых отделений.

Редакторам изданий может потребоваться следующая информация:

* об изданиях определенного типа с сортировкой их по популярности, которая основана на заказах почтовых отделений.

Служащим или начальнику типографии может потребоваться следующая информация:

* о тираже изданий, выпускаемых данной типографией, с указанием ФИО их редактора;
* об общем объеме заказа почтовыми отделениями каждого издания, выпускаемого на данной типографии;
* о заказах почтовыми отделениями изданий у данной типографии.

Служащим или начальнику почтового отделения может потребоваться следующая информация:

* об изданиях, заказываемых у типографий, данным почтовым отделением;
* о заказах данным почтовым отделением изданий у типографий с указанием ФИО директора типографии;
* ФИО начальника почтового отделения, на которое поступает наибольшее общее число изданий.

Пользователям этой базы данных может потребоваться также следующая информация:

* об изданиях, учет которых ведется в базе данных;
* о типографии, в которых печатается данное издание и его тираж;
* о почтовых отделениях, в которые поступает данное издание и в каком количестве.

В базу данных могут вноситься следующие изменения:

1. редакторы изданий могут добавлять информацию о своем издании, изменять или удалять ее из базы данных;
2. служащие или начальник типографии могут:

* добавлять информацию о своей типографии, изменять или удалять ее из базы данных;
* добавлять информацию о выпуске на своей типографии изданий, изменять или удалять ее из базы данных;

1. служащие или начальник почтового отделения могут:

* добавлять информацию о своем почтовом отделении, изменять или удалять ее из базы данных;
* добавлять информацию о своем заказе на типографии изданий, изменять или удалять ее из базы данных.

Пользователям, работающим с данной базой данных, необходимы следующие отчеты:

* 1. отчет по известным печатным изданиям, содержащий следующие данные:
* шифр, название, ФИО редактора этого издания;
* тираж и цена этого издания в различных типографиях;

1. отчет о работе типографий, содержащий следующие данные:

* номер, адрес и ФИО директора типографии;
* шифры, названия, тиражи и цены выпускаемых изданий;
* адреса рассылки (номера и адреса почтовых отделений) для каждого печатного издания;
* объем заказа и цену доставки для каждого почтового отделения;

1. отчет о работе почтовых отделений, содержащий следующие данные:

* номер, адрес и ФИО начальника почтового отделения;
* шифр, название, объем заказа и цена доставки покупаемых изданий;
* номера и адреса типографий, в которых заказаны издания.

1. отчет о работе определенного почтового отделения, содержащий следующие данные:

* номер, адрес и ФИО начальника почтового отделения;
* шифр, название, объем заказа и цена доставки покупаемых изданий;
* номера и адреса типографий – поставщиков.

**2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

Процесс проектирования базы данных представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к нормализованному описанию объектов предметной области в терми­нах некоторой модели.

Проектирование БД является очень важным этапом, от которого зависят последующие этапы разработки СУБД (системы управления базами данных). Время, затраченное разработчиком на проектирование БД, обычно окупается высокой скоростью реализации проекта.

Перед созданием базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, иметь всю необходимую информацию для удовлетворения предполагаемых запросов пользователя и определить потребности в обработке данных.

На основе такого описания на этапе проектирования базы данных осуществляется определение состава и структуры, данных предметной области, которые должны находиться в базе данных и обеспечивать выполнение необходимых запросов и задач пользователя. Структура данных предметной области может отображаться информационно-логической моделью. На основе этой модели легко создается реляционная база данных.

Информационно-логическая модель отображает данные предметной области в виде совокупности информационных объектов и связей между ними. Эта модель представляет данные, подлежащие хранению в базе данных.  
При разработке модели данных могут использоваться два подхода. В первом подходе сначала определяются основные задачи, для решения которых строится база, и выявляются потребности задач в данных. При втором подходе сразу устанавливаются типовые объекты предметной области. Наиболее рационально сочетание обоих подходов. Это связано с тем, что на начальном этапе, как правило, нет исчерпывающих сведений обо всех задачах. Использование такой технологии тем более оправдано, что гибкие средства создания реляционной базы данных в Access позволяют на любом этапе разработки внести изменения в базу данных и модифицировать ее структуру без ущерба для введенных ранее данных.

**2.1. Методология проектирования**

Широко известные методы проектирования баз данных (БД) появились в процессе разработки все более сложных Информационных Систем (ИС), которые должны были рассматривать потребности не одного пользователя, но больших групп и коллективов. Одна такая интегрированная БД создавалась для решения многих задач, каждая из которых использовала только "свою" часть данных, обычно, пересекающуюся с частями, используемыми в других задачах. Поэтому главнейшими методами проектирования стали методы исключения избыточности в данных. Эти методы связывались с другими средствами обеспечения логической целостности данных.

Было сформулировано принципиальное требование отделения программ от интегрированных данных. Этот принцип направлен на отчуждение данных в качестве ресурса предприятия, важен также тем, что консервативные по характеру данные отделялись от прикладных программ, которые могли часто подвергаться изменениям.

Другой важной проблемой проектирования БД явилось обеспечение нужных эксплуатационных параметров, таких как объем внешней памяти или время выполнения различных операций. Известны и другие требования. Например, информация не должна потеряться не только из-за отказов оборудования, но и вследствие ошибки пользователя. Это отличается от того положения, при котором тот, кто решает некую задачу, сам и отвечает за сохранность данных для этой задачи.

Сформировалось понимание интегрированной БД как общего информационного ресурса предприятия. Хранимые данные стали аналогичны большому компьютеру, который одновременно используется многими пользователями с различными целями и должен быть все время работоспособен.

**Временные характеристики и транзакции.**

Обеспечение эксплуатационных характеристик БД - по-прежнему непростая задача, несмотря на повышение удельной мощности компьютеров и снижение удельной стоимости памяти. При этом определение временных характеристик работы с БД и сохранение этих характеристик в процессе эксплуатации БД относится к труднейшим проектным задачам. На этапах проектирования для определения рациональной физической схемы БД от способов определения временных характеристик нужно следующее:

1. возможности сравнения временных параметров вариантов реализации разных вариантов схемы БД, на некоторой СУБД;
2. возможности сравнения параметров вариантов реализации одной схемы БД на разных СУБД;
3. возможности сравнения параметров реализации одной схемы БД на разных аппаратных серверах БД;
4. возможности предсказания временных параметров работы различных прикладных программ и служебных программ-утилит.

Задача сравнения временных параметров разных СУБД рассматривается как самостоятельная. Однако, она часто должна решаться как часть проектной задачи выбора СУБД для проектируемой БД и в процессе этого проектирования.   
Понятие транзакции было введено для определения законченной совокупности действий над БД, которая переводит БД из одного целостного в логическом смысле состояния в другое. На его базе строились, прежде всего, механизмы корректной актуализации и восстановления БД. Однако, затем на этой основе стали базироваться и другие механизмы и методы.

Временные оценки СУБД наиболее популярных тестов последнее время даются в виде числа транзакций определенного стандартизованного вида в единицу времени. Распределенная обработка строится на основе мониторов транзакций.   
Нужно будет обнаруживать пределы возможностей такого деления работ на достаточно мелкие порции. Здесь отметим очень важный эффект: практика ориентации на "транзакционный подход" тесно связана с классической методологией проектирования БД, которая развивалась, в основном, как методология проектирования так называемы "операционных" БД, то есть баз данных, которые должны фиксировать отдельные совершаемые операции и хранить модель текущего фактического состояния объекта или ПрО.

При проектировании структур данных для автоматизированных систем выполняется сбор информации об объектах предметной области в рамках одной таблицы (отношения), а потом выполняется последующая ее декомпозиция на несколько взаимосвязанных таблиц на основе процедуры нормализации отношений. При определении структур данных в отношениях реляционной модели существуют две основные проблемы: избыточное дублирование данных и аномалии.

Следует различать простое (неизбыточное) и избыточное **дублирование****данных**. Наличие простого дублирования допускается в базе данных, а избыточное может приводить к проблемам при обработке данных.

**Аномалиями** называют такую таблицу в базе данных, которая приводит к противоречиям в базе данных или существенно усложняет обработку данных. Выделяют три основных вида аномалий: аномалии модификации, удаления и добавления. Аномалии модификации проявляются в том, что изменение одного значения может повлечь за собой просмотр всей таблицы и соответствующие изменения других записей таблицы. Аномалии удаления состоят в том, что при удалении какого-либо значения из таблицы может пропасть и другая информация, которая не связана на прямую с удаляемым значением. Аномалии добавления возникают в случаях, когда информацию в таблицу нельзя поместить до тех пор, пока она не полная, либо вставка новой записи требует дополнительного просмотра таблицы.

От избыточности данных и различных аномалий можно избавиться с помощью нормализации отношений.

**2.1.1. Метод нормальных форм**

# Нормализация – это приведение, к лучшей форме относительно включения, удаление и модификации.

Процесс проектирования баз данных с использованием метода нормальных форм является итерационным и заключается в последовательном переводе отношений из первой нормальной формы в нормальные формы более высокого порядка по определенным правилам. Каждая следующая нормальная форма ограничивает определенный тип функциональных зависимостей, устраняет соответствующие аномалии при выполнении операций над отношениями базы данных и сохраняет свойства предшествующих нормальных форм.

Для определения структуры каждой таблицы необходимо выполнить анализ функциональных зависимостей, т. е. выяснить какие поля зависят от других полей, а затем поля с одинаковой зависимостью организовать в отдельную таблицу. Таким образом, в одну и туже таблицу не нужно включать поля, являющиеся произвольными от других полей. В результате количество необходимых таблиц определяется числом функциональных зависимостей. Формально нормализация данных обеспечена, если набор таблиц удовлетворяет первым трем правилам, которые называются нормальными формами.

Этот метод является классическим методом проектирования реляционной базы данных. Он основан на понятии **зависимости между атрибутами** отношений. Существуют следующие основные виды зависимостей: функциональные, транзитивные и многозначные.

Атрибут В **функционально** зависит о атрибута А, если каждому значению А соответствует в точности одно значение В (обозначение: А→В). Функциональная взаимозависимость атрибутов А и В означает, что имеется взаимнооднозначное соответствие, т.е. А→В и В→А (А↔В). Функциональная частичная зависимость – зависимость неключевого атрибута от части составного ключа. Полная функциональная зависимость – зависимость неключевого атрибута от всего составного ключа.

Атрибут С зависит от атрибута А **транзитивно**, если для атрибутов А,В,С выполняется условие А→В и В→С, но обратная зависимость отсутствует.

Атрибут В **многозначно** зависит от атрибута А, если каждому значению А соответствует множество значений В, не связанных с другими атрибутами. Эти зависимости могут быть «один ко многим», «многие к одному» или «многие ко многим» (1:m, m:1, m:m соответственно), обозначаемые соответственно А⇒В, В⇒А и А⇔В.

Два и более атрибута называются **взаимнонезависимыми,** если не один из этих атрибутов не является функционально-зависимым от других атрибутов (В не зависит от А: А¬ →В; взаимнонезависимы: А¬ =В).

Процесс проектирования базы данных с использованием метода **нормальных форм** является итерационным и заключается в последовательном переводе отношения из первой нормальной формы (НФ) в нормальную форму более высокого порядка по определенным правилам. Перевод отношения в следующую нормальную форму осуществляется методом декомпозиции без потерь, основной операцией этого метода является проекция. Каждая следующая нормальная форма ограничивает определенный тип зависимостей, устраняет соответствующие аномалии и сохраняет свойства предшествующих нормальных форм.

Выделяют следующую последовательность нормальных форм:

* 1. первая нормальная форма (1НФ);
  2. вторая нормальная форма (2НФ);
  3. третья нормальная форма (3НФ);
  4. усиленная нормальная форма или нормальная форма Бойса-Кодда (БКНФ);
  5. четвертая нормальная форма (4НФ);
  6. пятая нормальная форма (5НФ).

Отношение находится в **1НФ**, если все его атрибуты являются простыми (имеют единственное значение). Исходное отношение строится таким образом, чтобы оно было в 1НФ.

Отношение находится в **2НФ**, если оно находится в 1НФ, и каждый неключевой атрибут функционально-полно зависит от первичного ключа.

Отношение находится в **3НФ**, если оно находится в 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа. Отношение находится в **3НФ** в том и только в том случае, если все неключевые атрибуты отношения взаимнонезависимы и полностью зависят от первичного ключа.

Если в отношении имеется зависимость атрибутов составного ключа от неключевых атрибутов, то нужно перейти к БКНФ. Отношение находится в **БКНФ**, если оно находится в 3НФ, и в нем отсутствуют зависимости ключей от неключевых атрибутов.

**4НФ**. В произвольном отношении R(A,B,C) может одновременно существовать многозначная зависимость А⇒В и А⇒С. Это обстоятельство обозначим как А⇒В|С, т.е. атрибуты В и С многозначно зависят от А.

**Теорема Фейджина**: отношение R(A,B,C) можно спроецировать без потерь в отношение R1(A,B) и R2(A,C) в том и только в том случае, когда существует зависимость А⇒В|С.

Отношение R находится в **4НФ** в том и только в том случае, когда существует многозначная зависимость А⇒В, а все остальные атрибуты R функционально зависят от А.

**5НФ**. Результатом нормализации предыдущих отношений были два новых отношения, иногда это сделать не удается или отношения заведомо имеют нежелательные свойства. В этом случае выполняют декомпозицию отношения более чем на два отношения.

Отношение R(X,Y,…,Z) удовлетворяет зависимости соединения, которое обозначим как \*(X,Y,…,Z), в том и только в том случае, если R восстанавливается без потерь путем соединения своих проекций на X,Y,…,Z.

Отношение R находится в **5НФ** или нормальной форме проекции-соединения (PJ/NF) в том и только в том случае, когда любая зависимость соединения в R следует из существования некоторого возможного ключа в R.

Эта форма является последней из известных, условия ее получения довольно нетривиальны и поэтому она почти не используется на практике. Более того, она имеет определенные недостатки, поэтому на практике обычно ограничиваются структурой базу данных, соответствующей 3НФ или БКНФ.

**2.1.2. Метод сущность-связь**

Моделирование структуры базы данных при помощи алгоритма нормализации, описанного в предыдущих главах, имеет серьезные недостатки:

Первоначальное размещение всех атрибутов в одном отношении является очень неестественной операцией. Интуитивно разработчик сразу проектирует несколько отношений в соответствии с обнаруженными сущностями. Даже если совершить насилие над собой и создать одно или несколько отношений, включив в них все предполагаемые атрибуты, то совершенно неясен смысл полученного отношения.

Невозможно сразу определить полный список атрибутов. Пользователи имеют привычку называть разными именами одни и те же вещи или наоборот, называть одними именами разные вещи.

Для проведения процедуры нормализации необходимо выделить зависимости атрибутов, что тоже очень нелегко, т.к. необходимо **явно выписать все зависимости**, даже те, которые являются очевидными.

В реальном проектировании структуры базы данных применяются другой метод - так называемое, **семантическое моделирование**. Семантическое моделирование представляет собой моделирование структуры данных, опираясь на смысл этих данных. В качестве инструмента семантического моделирования используются различные варианты **диаграмм сущность-связь** (*ER - Entity-Relationship*).

Первый вариант модели сущность-связь был предложен в 1976 г. Питером Пин-Шэн Ченом . В дальнейшем многими авторами были разработаны свои варианты подобных моделей (нотация Мартина, нотация IDEF1X, нотация Баркера и др.). Кроме того, различные программные средства, реализующие одну и ту же нотацию, могут отличаться своими возможностями. По сути, все варианты диаграмм сущность-связь исходят из одной идеи - рисунок всегда нагляднее текстового описания. Все такие диаграммы используют графическое изображение сущностей предметной области, их свойств (атрибутов), и взаимосвязей между сущностями.

Мы опишем работу с ER-диаграммами близко к нотации Баркера, как довольно легкой в понимании основных идей. Данная глава является скорее иллюстрацией методов семантического моделирования, чем полноценным введением в эту область.

Кроме метода нормальных форм Кодда для проектирования больших баз данных используют метод ER-диаграмм (метод сущность-связь). На последнем этапе метода ER-диаграмм полученные отношения анализируют на принадлежность к БКНФ.

Основными понятиями этого метода являются следующие: сущность; атрибут сущности; ключ сущности; связь между сущностями, степень связи, класс принадлежности экземпляров сущности; диаграммы ER-экземпляров и диаграммы ER-типа.

**Сущность** представляет собой объект, информация о котором хранится в базе данных. Экземпляры сущности отличаются друг от друга и однозначно идентифицируются. Названиями сущности являются, как правило, существительные, например, ИЗДАНИЕ, ТИПОГРАФИЯ и т.д. **Атрибут сущности** – свойство сущности. **Ключ сущности** – атрибут или набор атрибутов, используемый для идентификации экземпляра сущности. **Связь сущности** – зависимость между атрибутами этих сущностей, название связи обычно представляется глаголами.

С целью повышения наглядности и удобства проектирования используют следующие графические средства: диаграммы ER-экземпляра; диаграммы ER-типов или ER-диаграммы. При построении диаграмм ER-типов учитывается степень связи и класс принадлежности, которые выявляются на основе анализа диаграмм ER-экземпляров. **Степень связи** характеризует связь между сущностями, которая может быть один к одному (1:1), один ко многим (1:М), много к одному (М:1), много ко многим (М:М).. Степень означается символами на линии связи. **Класс принадлежности** может быть обязательным или необязательным. Класс принадлежности является обязательным в том случае, если все экземпляры этой сущности обязательно участвуют в этой связи. В противном случае класс принадлежности является необязательным.

Под каждым блоком, соответствующим некоторой сущности указывается ее ключ, выделяемый подчеркиванием. Многоточие за ключевыми атрибутами означает, что возможны другие атрибуты сущности, но ни один из них не может быть частью ее ключа. Эти атрибуты выявляются после формирования отношения.

Этапы проектирования

Процесс проектирования базы данных является итерационным, т.е. допускающим возврат к предыдущим этапам для пересмотра ранее принятых решений, и включает следующие этапы:

1. выявление сущностей и связей между ними;
2. построение диаграмм ER-типа с учетом всех сущностей и их связей;
3. формирование набора предварительных отношений с указанием предполагаемого первичного ключа для каждого отношения;
4. добавление неключевых атрибутов в отношения;
5. приведение предварительных отношений к БКНФ с помощью метода нормальных форм;
6. пересмотр ER-диаграмм в следующих случаях:

* некоторые отношения не приводятся к БКНФ;
* некоторым атрибутам не находится логически-обоснованных мест в предварительных отношениях.

После преобразования ER-диаграмм осуществляется повторное выполнение предыдущих этапов проектирования.

Правила формирования отношений:

1. Если степень связи 1:1 и класс принадлежности обеих сущностей обязательны, то формируется одно отношение. Первичным ключом этого отношения может быть ключ любой сущности.
2. Если степень связи 1:1 и класс принадлежности первой сущности обязателен, а другой нет, то под каждую из сущностей выделяется по отношению с первичными ключами, являющимся ключами соответствующих сущностей. Далее к отношению, сущность которого имеет обязательный класс принадлежности, добавляется в качестве атрибута ключ сущности с необязательным классом принадлежности.
3. Если степень связи 1:1 и класс принадлежности обеих сущностей необязателен, то нужно использовать три отношения: два отношения соответствуют связываемым сущностям, ключи которых являются первичными в этих отношениях; третье отношение является связным между первыми двумя, поэтому его ключ объединяет ключевые атрибуты связываемых отношений.
4. Если степень связи между сущностями 1:М или М:1 и класс принадлежности М-связной обязательный, то достаточно формирования двух отношений, по одному на каждую из сущностей. Каждое отношение будет иметь свои первичные ключи, и, кроме того, ключ 1-связной сущности добавляется в отношение М-связной сущности.
5. Если степень связи 1:М или М:1 и класс принадлежности М-связной сущности является необязательным, то необходимо формирование трех отношений: два отношения соответствуют связываемым сущностям, ключи которых являются первичными в этих отношениях; третье отношение является связным и его ключ объединяет ключи первых двух отношений.
6. Если степень связи М:М, то независимо от класса принадлежности сущностей формируется три отношения: два отношения соответствуют связываемым сущностям; а третье является связным и объединяет ключи первых двух отношений.

**2.2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ «ПОЧТОВЫЕ ОТДЕЛЕНИЯ»**

**Первый этап проектирования**

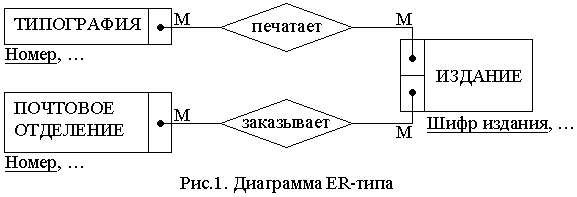
В результате анализа предметной области работы почтовых отделений выделим следующие сущности:

* ИЗДАНИЕ (Шифр издания, название, Ф.И.О. редактора);
* ТИПОГРАФИЯ (Номер, адрес, Ф.И.О. директора);
* ПОЧТОВОЕ ОТДЕЛЕНИЕ (Номер, адрес, Ф.И.О. директора).

Между этими сущностями существуют следующие связи:

* Типографии печатают издания;
* Почтовые отделения заказывают издания.

**Второй этап проектирования**



**Третий этап проектирования**

На основе анализа построенной диаграммы ER-типа и правил формирования отношений построим предварительные отношения.

Связь «Печатает» удовлетворяет условиям правила №6, в соответствии с которым получаем 3 отношения:

* ИЗДАНИЕ (Шифр издания, название, Ф.И.О. редактора);
* ТИПОГРАФИЯ (Номер, адрес, Ф.И.О. директора);
* ТИРАЖ (Шифр издания, Номер типографии, тираж, цена одного экземпляра, код заказа).

Связь «**Заказывают**» удовлетворяет условиям правила №6, в соответствии с которым получаем 3 отношения:

* ПОЧТОВОЕ ОТДЕЛЕНИЕ (Номер, адрес, Ф.И.О. директора);
* ИЗДАНИЕ (Номер, название, Ф.И.О. редактора);
* ЗАКАЗ (Код заказа, номер почтового отделения, шифр издания, номер типографии, количество заказов, цена доставки).

Четвертый этап проектирования

Добавим неключевые атрибуты к полученным отношениям, и тогда они примут следующий вид:

* ИЗДАНИЕ (Шифр издания, Название, ФИО\_редактора);
* ТИПОГРАФИЯ (Номер, Адрес, ФИО\_директора);
* ПОЧТОВОЕ ОТДЕЛЕНИЕ (Номер, Адрес, ФИО\_начальника);
* ТИРАЖ (Шифр издания, Номер типографии, Тираж, Цена\_одного\_экземп-ляра);
* ЗАКАЗ (Код заказа, Номер почтового отделения, Шифр издания, Номер типографии, Количество заказов, Цена\_доставки).

**Пятый этап проектирования**

С помощью метода нормальных форм приведем исходные отношения к БКНФ. Все атрибуты всех исходных отношений являются простыми, следовательно, все эти отношения уже находятся в 1НФ.

Зависимости между атрибутами в отношениях:

* 1. ИЗДАНИЕ (Шифр издания, Название, ФИО\_редактора):
  2. Шифр издания → Название;
  3. Шифр издания → ФИО\_редактора;
  4. ТИПОГРАФИЯ (Номер, Адрес, ФИО\_директора):
  5. Номер → Адрес;
  6. Номер → ФИО\_директора;
  7. ПОЧТОВОЕ ОТДЕЛЕНИЕ (Номер, Адрес, ФИО\_начальника):
  8. Номер → Адрес;
  9. Номер → ФИО\_начальника;
  10. ТИРАЖ (Шифр издания, Номер типографии, Тираж, Цена\_одного\_экземп-ляра):
  11. Шифр издания, Номер типографии → Тираж;
  12. Шифр издания, Номер типографии → Цена\_одного\_экземпляра;
  13. ЗАКАЗ (Код заказа, Номер почтового отделения, Шифр издания, Номер типографии, Количество заказов, Цена\_доставки):
  14. Код заказа → Количество\_заказа;
  15. Код заказа → Цена\_доставки.

Каждый из неключевых атрибутов всех отношений функционально-полно зависит от первичного ключа своего отношения, и это значит, что все эти отношения находятся в 2НФ. Все неключевые атрибуты каждого из отношений взаимно-независимы и полностью зависят от своего первичного ключа, следовательно, все эти отношения находятся в 3НФ. Все атрибуты первичных ключей каждого из отношений не имеют зависимости от неключевых атрибутов, и это значит, что все эти отношения находятся в БКНФ. Так как все отношения приведены к БКНФ и все атрибуты имеют логически обоснованные места в предварительных отношениях, то проектирование этой базы данных не требует проведения шестого этапа проектирования.

**3. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ СРЕДСТВАМИ MS ACCESS**

**3.1. Обзор системы управления базами данных MS Access**

Microsoft Access – система управления базами данных (СУБД), которую фирма Microsoft включает в состав профессиональной редакции (Professional Edition) Microsoft Office. Удобство использования и мощность встроенных средств делает диапазон информационных систем, построенных с применением Access, весьма широким – от локальных разработок для домашнего применения до серьезных профессиональных проектов распределенных по сети баз данных, включающих сотни тысяч записей и сложнейшие правила обработки данных.

СУБД Access поддерживает реляционную модель представления данных. Она функционирует под управлением операционных систем Windows 95/98, Windows NT и выше. СУБД Access имеет стандартизованный интерфейс приложений Windows. Access позволяет создать рабочую информационную систему практически без единой строки программного кода, лишь с помощью визуального проектирования, встроенных мастеров и шаблонов. В ней реализованы возможности программирования с использованием структурированного языка запросов Structured Query Language (SQL) и языка Visual Basic for Application (VBA).

Access поддерживает традиционные для офисных приложений механизмы связывания и встраивания объектов Object Linking and Embedding (OLE) и динамического обмена данными Dynamic Data Exchange (DDE). Это позволяет Access работать с любыми объектами из библиотеки типов другого приложения пакета Microsoft Office и предоставлять свои объекты для других приложений. В Access 2002 появились возможности использования расширяемого языка разметки – extensible Markup Language (XML), играющего роль стандарта обмена данными между приложениями.

Большинство действий по работе с элементами баз данных в среде Access можно выполнить с помощью следующих средств: команд основного меню, кнопок панелей инструментов, команд контекстного меню и комбинаций клавиш.

База данных является основным компонентом проекта приложения Access. К основным элементам базы данных можно отнести таблицы, запросы, отчеты, макросы и модули.

Для работы с БД в Access имеется стандартное окно, из которого можно вызвать любой ее объект для просмотра, выполнения, разработки или модификации. Пользователь может для работы с БД разработать свой интерфейс, основу которого обычно составляют формы.

Текущая (открытая в данный момент) БД может взаимодействовать с внешними БД, которые используются как источники таблиц при импорте или присоединении, а также как получатели при экспорте данных из текущей базы данных.

**Таблица** представляет собой основную единицу хранения данных в базе. Понятие таблицы в Access соответствует аналогичному понятию реляционной модели данных. В произвольной базе данных Access обычно имеется совокупность связанных между собой таблиц.

С таблицей в целом можно выполнять следующие операции:

* создание (определение структуры);
* изменение структуры (реструктуризация);
* переименование;
* удаление.

При создании таблицы задается структура и имя таблицы. При сохранении таблицы на диске вся необходимая информация размещается в файле базы данных. При изменении структуры таблицы в ней могут измениться названия и характеристики полей, состав и названия ключа и индексов, ограничения. Для изменения структуры и переименования таблиц используются утилиты (программы), предназначенные для работы с таблицами БД.

Между двумя таблицами можно устанавливать связи типа 1:1 и 1:М в окне описания схемы данных. Основными операциями над таблицами являются: просмотр и обновление (ввод, модификация и удаление), сортировка, фильтрация и печать.

**Запрос** представляет собой формализованное требование на отбор данных из таблиц или на выполнение определенных действий с данными. Запрос позволяет создать набор записей из данных, находящихся в разных таблицах, и использовать его как источник данных для формы или отчета. В Access можно создавать и выполнять следующие основные типы запросов: выборка, обновление, удаление или добавление данных. С помощью запросов можно также создавать новые таблицы, используя данные из одной или нескольких существующих таблиц. Описание запроса можно выполнить с помощью бланка QBE или инструкции языка SQL.

**Форма** представляет собой объект базы данных Access, в котором разработчик размещает элементы управления, принимающие действия пользователей или служащие для ввода, отображения и изменения данных в полях. На формах можно разместить различные элементы, такие как: поля таблиц, поля со списком, кнопки, раскрывающиеся списки, выключатели, переключатели, флажки, рисунки, надписи, подчиненные формы и т.д. За кнопками обычно закрепляют вызов функций. Функции обработки информации во время работы с БД задаются с помощью макросов или программ на языке VBA.

**Отчет** представляет собой объект базы данных, предназначенный для отображения содержащейся в базе данных информации. Он позволяет осуществлять необходимую группировку данных, отображать расчетные и итоговые данные. Отчет можно просмотреть на экране или распечатать через принтер на бумагу.

**Макрос** представляет последовательность команд встроенного в Access языка VBA, задающих автоматическое выполнение некоторых операций.

**Модуль** представляет совокупность описаний, инструкций и процедур на языке VBA, сохраненную под общим именем. В Access используются модули трех типов: формы, отчета и стандартный. Модули форм и отчетов содержат программы, являющиеся локальными для этих объектов. Процедуры из стандартного модуля, если они не описаны явно как локальные для содержащего их модуля, распознаются и могут вызываться процедурами из других модулей той же или другой базы данных Access.

**3.2.Формирование исходных данных**

Суть любой базы данных заключается в том, что она должна хранить данные, адекватно отражающие соответствующее состояние предметной области. Поэтому реальная база данных для почтовых работников должна содержать фактические данные, соответствующие данной предметной области. Но моя база данных является учебной, и поэтому данные, хранящиеся в ней, будут условными.

Тем не менее, исходные данные этой базы данных должны соответствовать требованиям, которые были выявлены при анализе предметной области, т.е. данные будут заполняться согласно запросам, формам и отчетам, которые могут быть необходимы пользователям этой базы данных. Кроме того, исходные данные должны проверить соответствие спроектированной базы данных данной предметной области, а также оценить правильность формирования отношений, выявление первичных ключей и связей между отношениями.

Исходные данные, необходимые для данной предметной области, можно просмотреть в Приложении 3.

3.3**.Составление бизнес – плана**

**Бизнес-план** *-* это комплект документов, позволяющий оценить эффективность и надежность вложения средств для превращения бизнес-идеи в реальное дело, приносящее прибыль.

Содержащуюся в бизнес-плане информацию можно условно разделить на информацию описательного характера и информацию, полученную в результате финансовых расчетов моделируемой ситуации.

Расчетная информация может быть получена и систематизирована с помощью программной системы Project Expert.

**Project Expert** - автоматизированная система планирования и анализа эффективности инвестиционных проектов на базе имитационной модели денежных потоков. Используемый в данной системе сценарный подход (возможность построения модели и проигрыш вариантов ее развития) является наиболее эффективным в условиях неопределенности многих значимых для инвестиционного проекта факторов, например, таких как: инфляция, прогноз объема продаж, задержки платежей и т.п. Модель проекта с максимальной достоверностью имитирует реальную операционную деятельность предприятия в условиях, когда вводимые исходные данные и варьируемые параметры описывают суть принимаемых тех или иных управленческих решений. В результате расчетов оцениваются последствия принимаемых решений в виде общепринятых финансовых показателей.

**Отчёт о прибылях и убытках** позволяет определить из каких составляющих складывается прибыль предприятия.

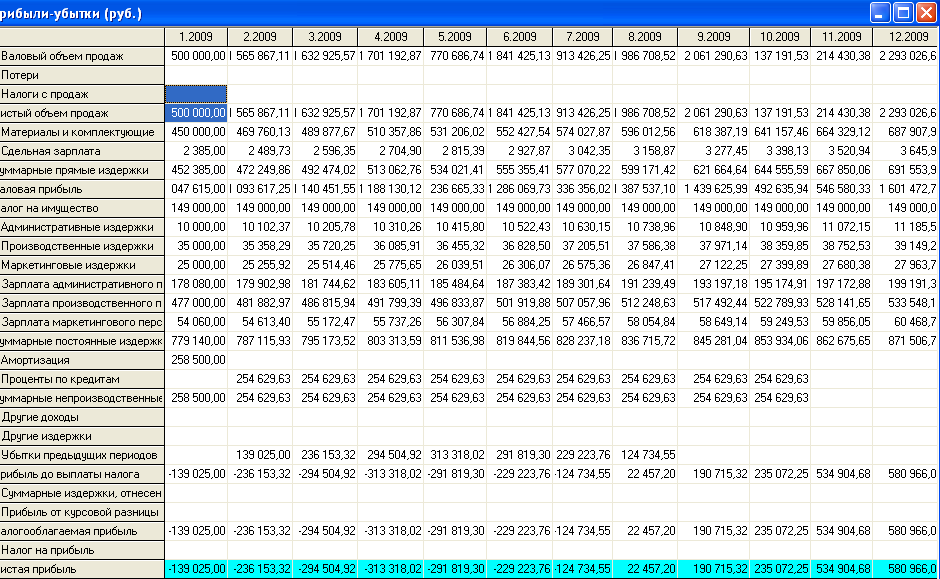
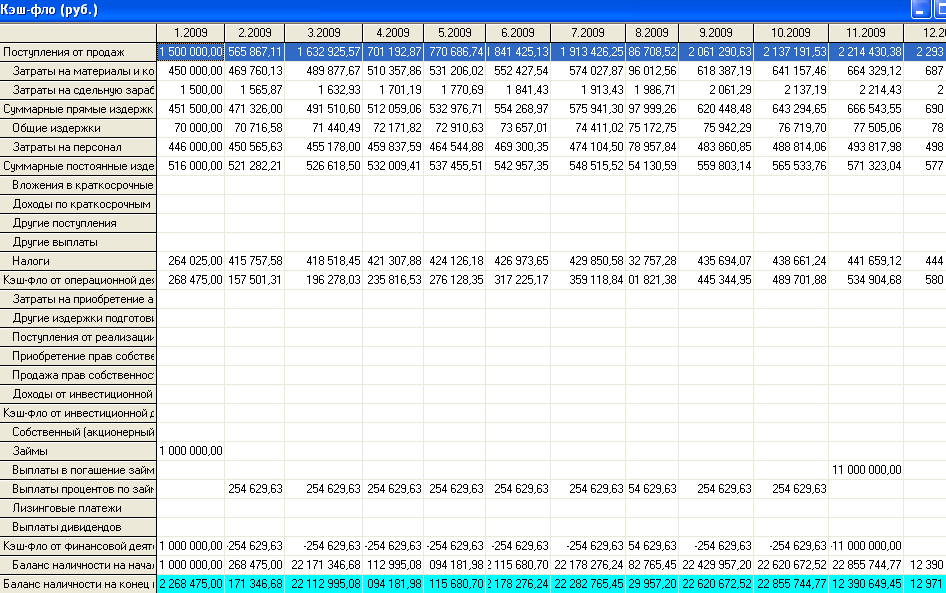
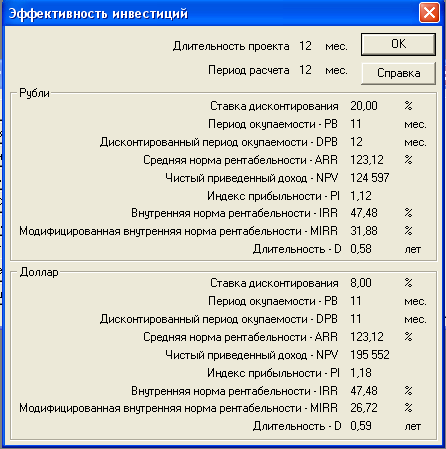


Таблица Кэш-фло



Эффективность инвестиций



**3.4. Разработка объектов базы данных**

MS Access предоставляет два способа создания базы данных: создание типовой БД с помощью мастера и создание пустой БД, которая не содержит данных, но хранит информацию о структуре. Первый способ обеспечивает быстрое создание БД и позволяет получить типовую базу данных с образцами информации в таблицах. Он применим в случаях, когда пользователю подходит одна из десяти предлагаемых СУБД Access типовых баз данных. Второй способ отличается большей гибкостью и одновременно большей трудоемкостью, поскольку требует отдельного определения каждого элемента базы данных. Независимо от способа создания базу данных можно в любое время изменить и расширить.

При создании базы данных «Почтовые отделения» я буду пользоваться вторым способом создания БД. Для создания новой базы данных с помощью СУБД MS Access необходимо: открыть ее с помощью щелчка по соответствующей пиктограмме.

Далее необходимо создать файл новой БД, и для этого нужно выполнить следующие действия:

1. В открывшемся рабочем окне Microsoft Access появится диалоговое окно, в котором будут следующие варианты: *Создать новую БД*, *Создать БД с помощью мастера* или *Открыть БД*. Из них необходимо выбрать *Создать новую БД* и нажать кнопку *ОК*.
2. Появится диалоговое окно создания новой БД. В нем необходимо определить имя нового файла и указать его местоположение. По умолчанию Access присвоит создаваемой базе данных имя *db1* и предложит сохранить ее в папке *Мои документы*. Так как я создаю базу данных, спроектированную во второй главе курсовой работы, то логично назвать ее «Почтовые отделения». После ввода имени в поле И*мя файла* необходимо щелкнуть на кнопке *Создать*.
3. Открывается диалоговое окно созданной базы данных, в котором предлагается выбрать элемент БД и способ его создания.

В открывшемся окне открыта закладка *Таблицы*, для создания новой таблицы необходимо нажать кнопку *Создать*. В открывшемся окне «Новая таблица» предлагаются следующие режимы создания новой таблицы: *Режим таблицы*, *Конструктор*, *Мастер таблиц*, *Импорт таблиц*, *Связь с таблицами*. Для создания моей базы данных я буду использовать режим *Конструктор*, поэтому выбираю в этом окне *Конструктор* и нажимаю на кнопку *OK*.

Открывшееся окно содержит три раздела:

* *Имя поля* – обязательный раздел;
* *Тип данных* – условно-обязательный раздел;
* *Описание* – необязательный раздел.

В разделе *Имя поля* нужно указать имена всех необходимых полей таблицы. Чтобы начать работу с разделом *Тип данных*, следует в нем в строке, соответствующей создаваемому полю, щелкнуть мышью. После появления кнопки со стрелкой вниз щелкнуть на ней и выбрать подходящий тип данных из появившегося списка. Для каждого поля можно дополнительно задать свойства полей в разделе *Свойства полей*, который делится на две вкладки *Общие* и *Подстановка*. Необходимо обозначить ключевые поля таблицы, это необходимо для идентификации каждой записи таблицы. Для этого необходимо выделить их и нажать на панели инструментов кнопку *Ключ*. После ввода всех необходимых параметров на панели инструментов необходимо выбрать в меню *Файл* пункт *Сохранить*, а далее в появившемся окне ввести имя создаваемой таблицы и нажать кнопку *OK*.

СУБД MS Access поддерживает следующие типы данных: текстовый; поле MEMO; числовой; дата/время; денежный; счетчик; логический; поле объекта OLE; гиперссылка; мастер подстановок.

СУБД MS Access позволяет заполнить следующие свойства полей, которые определяют параметры ввода, отображения и хранения данных в таблицах:

* размер поля: используется только текстовым и числовым типами данных, для текстовых полей он может быть от 1 до 255 символов, а у числовых – целое, длинное целое, одинарное с плавающей точкой и другие;
* маска ввода: устанавливается формат, используемый при вводе данных;
* подпись: задает имя полей, которое будет использоваться для отображения в формах и отчетах;
* значение по умолчанию: определяет данные, автоматически добавляемые в новую запись;
* условие на значение: является логическим выражением, определяющим реакцию Access на ввод данных;
* обязательное поле: определяет необходимость ввода данных в поле;
* индексированное поле: указывает, нужно ли создавать индекс на поле.

Таким образом, создаем следующие таблицы, которые соответствуют отношениям, выявленным на этапе проектирования базы данных: Издание, Типография, Почтовое отделение, Тираж, Заказ. Структура этих таблиц приведена в приложении 2.

Для создания запроса необходимо переключиться в созданной БД на закладку *Запросы* и нажать кнопку *Создать*. В открывшемся окне «Новый запрос» можно выбрать один из следующих способов создания запроса: *Конструктор*, *Простой запрос*, *Перекрестный запрос*, *Повторяющиеся записи*, *Записи без подчиненных*. Для создания запросов к моей БД я буду использовать *Конструктор*, поэтому необходимо выбрать *Конструктор* и нажать кнопку *OK*. В результате откроется окно нового запроса и в нем диалоговое окно «Добавление таблицы», в котором можно выбрать таблицы, которые будут нужны для формирования запросов, а также уже созданные запросы. Открывшееся окно *Конструктора запросов* разделено на две части: в верхней отображаются все выбранные для запроса таблицы со списками входящих в них полей и существующие межтабличные связи; в нижней находится бланк запроса, служащий для определения параметров запроса.

Далее необходимо выбрать тип запроса: *Выборка*, *Перекрестный*, *Создание таблицы*, *Обновление*, *Добавление*, *Удаление*. По умолчания будут создаваться запросы на выборку, которые я и буду использовать в своей базе данных.

Для формирования бланка запроса необходимо из верхней части окна перетаскивать в него необходимые поля таблиц. Бланк запроса содержит следующую информацию по каждому использующемуся полю: *Поле*, *Имя таблицы*, *Сортировка*, *Групповая операция*, *Вывод на экран*, *Условие отбора*. В строке *Сортировка* можно задать одну из следующих сортировок для каждого поля: *по возрастанию* или *по убыванию*. В строке *Групповая операция* можно выбрать одну из следующих операций: суммирование, выбрать минимальное или максимальное значение, посчитать их количество и другие. В строке *Вывод на экран* можно поставить или убрать отметку о выводе на экран, по умолчанию для всех полей она будет поставлена. В строке *Условие отбора* можно задать следующие типы условий: определенное значение, диапазон значений, можно использовать операторы сравнения, выбирать записи с нулевыми и ненулевыми значениями в данном поле и другие.

Для сохранения этого запроса необходимо выбрать в меню *Файл* пункт *Сохранить*, в открывшемся окне набрать имя нового запроса и нажать кнопку *ОК*. Таким образом, создаем следующие запросы.

Запрос «1 – адреса типографий, в которых печатаются газеты или журналы данного наименования» отображает информацию по следующим полям: Издание.name, Типография.adress. В этом запросе в поле Издание.name производится отбор записей, у которых значение в этом поле начинается с введенной цифры. А в поле Типография.adress производится групповая операция – суммирование (Sum), для того, чтобы показать совокупный заказ всеми почтовыми отделениями данного издания, и сортировка записей по убыванию. Будет производиться сортировка записей по убыванию в этом же поле. Этот запрос создан для показа редакторам изданий определенного типа популярности изданий, основанной на объеме заказа почтовыми отделениями.

Запрос «2-1 - Информация об изданиях на типографии» отображает информацию по следующим полям: Тираж.shifr\_izd, Издание.nazv\_izd, Тираж.zena\_ekz\_izd, Тираж.tirag\_izd, Заказ.obem\_zakaza. Для этого запроса производится отбор записей по номеру типографии, который будет введен для поля Типография.nom\_tipograf. Этот запрос создан для показа работникам определенной типографии информации о печатаемых типографией изданиях, а также тираж и общий объем заказа издания почтовыми отделениями у типографии.

Запрос «2-2 - Заказы изданий почтовыми отделениями у типографии» отображает информацию по следующим полям: Издание.nazv\_izd, Заказ.nom\_pocht\_otd, Заказ.obem\_zakaza, а так же поле, в котором производится вычисление по следующему выражению для определения цены заказа с учетом цены доставки: Тираж.zena\_ekz\_izd \* Заказ.obem\_zakaza + Заказ.zena\_dostav-ki\_partii. Для этого запроса производится отбор записей по номеру типографии, который будет введен для поля Типография.nom\_tipograf. Этот запрос создан для показа работникам определенной типографий информации о заказах почтовыми отделениями изданий.

Запрос «3-1 - Информация об изданиях на почтовом отделении» отображает информацию по следующим полям: Тираж.shifr\_izd, Издание.nazv\_izd, Заказ.obem\_zakaza. А в поле Заказ.obem\_zakaza производится групповая операция – суммирование (Sum), для того, чтобы показать совокупный заказ этим почтовым отделением определенного издания. Для этого запроса производится отбор записей по номеру почтового отделения, который будет введен для поля Почтовое\_отделение.nom\_pocht\_otd. Этот запрос создан для показа работникам определенного почтового отделения информации о совокупных заказах изданий.

Запрос «3-2 - Заказы изданий почтовым отделением у типографий» отображает информацию по следующим полям: Издание.nazv\_izd, Заказ.nom\_tipograf, Типография.adres\_tipograf, Типография.fio\_direktora\_tipograf, Заказ.obem\_zakaza, а так же поле, в котором производится вычисление по следующему выражению для определения цены заказа с учетом цены доставки: Тираж.zena\_ekz\_izd \* Заказ.obem\_zakaza + Заказ.zena\_dostavki\_partii. Для этого запроса производится отбор записей по номеру почтового отделения, который будет введен для поля Почтовое\_отделение.nom\_pocht\_otd. Этот запрос создан для показа работникам определенного почтового отделения информации о заказах изданий у типографий.

Запрос «3-3 - Почтовое отделение с наибольшим общим числом изданий» отображает информацию по следующим полям: Почтовое\_отделение.nom\_-pocht\_otd, Почтовое\_отделение.adres\_pocht\_otd, Почтовое\_отделение.fio\_nach\_-pocht\_otd, Заказ.obem\_zakaza. А в поле Заказ.obem\_zakaza производится групповая операция – суммирование (Sum), для того, чтобы показать общее число изданий, заказанное почтовым отделением. Этот запрос создан для показа информации о почтовом отделении с наибольшим общим числом заказываемых изданий.

Запрос «4-1 - Список изданий» отображает информацию по следующим полям: Издание.shifr\_izd, Издание.nazv\_izd, Издание.fio\_redaktora\_izd. Этот запрос создан для показа информации об известных печатных изданиях.

Запрос «4-2 - Типографии где печатается издание» отображает информацию по следующим полям: Тираж.nom\_tipograf, Типография.adres\_tipograf, Типография.fio\_direktora\_tipograf, Тираж.tirag\_izd. Для этого запроса производится отбор записей по шифру издания, который будет введен для поля Тираж.shifr\_izd. Этот запрос создан для показа типографий, где печатается определенное издание.

Запрос «4-3 - Почтовые отделения которые заказывают издание» отображает информацию по следующим полям: Заказ.nom\_pocht\_otd, Почтовое\_отделение.adres\_pocht\_otd, Почтовое\_отделение.fio\_nach\_pocht\_otd, Заказ.obem\_zakaza. А в поле Заказ.obem\_zakaza производится групповая операция – суммирование (Sum), для того, чтобы показать совокупный заказ этим почтовым отделением определенного издания. Для этого запроса производится отбор записей по шифру издания, который будет введен для поля Заказ.shifr\_izd. Этот запрос создан для показа почтовых отделений, которые заказывают определенное издание.

Все эти запросы приведены в Приложении 4.

Для создания форм необходимо переключиться в созданной БД на закладку *Формы* и нажать кнопку *Создать*. MS Access позволяет создавать формы следующим способами: в режиме *Автоформы*, с помощью *Мастера форм*, с помощью *Конструктора форм*. Кроме того, возможно построение разнообразных форм с помощью запросов и расширение возможностей форм на основе включения в них диаграмм, картографических данных и рисунков. Для создания форм к моей БД я буду использовать *Мастер форм*, и для этого необходимо в открывшемся окне «Новая форма» необходимо выбрать *Мастер форм* и нажать кнопку *ОК*.

Откроется первое окно мастера *Создание формы*. В нем есть список *Таблицы и запросы*, в котором отображены все таблицы и запросы, созданные в этой БД и которые можно использовать как источники данных, нажав на стрелку прокрутки и выбрав из списка. При этом в области *Доступные поля* будут отражаться поля выбранной таблицы или запроса. Необходимые поля можно перенести в область *Выбранные поля* с помощью специальных кнопок. После выбора необходимых полей перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

В левой части открывшегося окна дается перечень используемых таблиц, одну из которых выбирают как основной источник данных. В правой же части окна находится область предварительного просмотра и можно выбрать вариант формы: *Одиночная форма*, *Подчиненная форма* или *Связанная форма*. После осуществления необходимого выбора перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

В правой части открывшегося окна находится область определения внешнего вида формы, в левой – область предварительного просмотра. Можно выбрать один из следующих видов форм: *в один столбец*, *ленточный*, *табличный*, *выровненный* и другие. После осуществления необходимого выбора перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

В правой части открывшегося окна можно выбрать стиль формы, в левой части находится область предварительного просмотра. Можно выбрать один из следующих стилей формы: *Стандартный*, *Официальный*, *Чертеж* и другие. После осуществления необходимого выбора перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

Откроется последнее окно *Мастера формы*, в котором можно изменить имя формы и подчиненной формы (если такая будет создана). В нем также предлагается сделать выбор дальнейших действий: *Открыть форму для просмотра и ввода данных* или *Изменить макет формы*. Можно также отметить пункт *Вывести справку по работе с формой*. После осуществления необходимых изменений нажимаем кнопку *Готово*. Если в качестве дальнейшего действия выбран вариант *Изменить макет формы*, то откроется конструктор созданной формы, в котором можно сделать необходимые изменения.

Таким образом, создаем следующие формы:

* «1-1 - Список изданий» с полями: Издание.shifr\_izd, Издание.nazv\_izd, Издание.fio\_redaktora\_izd; эта форма имеет «ленточный» вид;
* «2-1 - Список типографий» с полями: Типография.nom\_tipograf, Типография.adres\_tipograf, Типография.fio\_direktora\_tipograf; эта форма имеет «ленточный» вид;
* «2-2 - Печать изданий типографиями» с полями: Типография.nom\_tipograf, Типография.adres\_tipograf, Типография.fio\_direktora\_tipograf, Издание.-shifr\_izd, Издание.nazv\_izd, Издание.fio\_redaktora\_izd, Тираж.tirag\_izd, Тираж.zena\_ekz\_izd; эта форма имеет вид «в один столбец»;
* «3-1 - Список почтовых отделений» с полями: Почтовое\_отделение.nom\_-pocht\_otd, Почтовое\_отделение.adres\_pocht\_otd, Почтовое\_отделение.fio\_-nach\_pocht\_otd; эта форма имеет «ленточный» вид;
* «3-2 - Заказ изданий почтовыми отделениями у типографий»: Почтовое\_отделение.nom\_pocht\_otd, Почтовое\_отделение.adres\_pocht\_otd, Почтовое\_отделение.fio\_nach\_pocht\_otd, Типография.nom\_tipograf, Типография.-adres\_tipograf, Типография.fio\_direktora\_tipograf, Издание.shifr\_izd, Изда-ние.nazv\_izd, Издание.fio\_redaktora\_izd, Заказ.obem\_zakaza, Заказ.zena\_dos-tavki\_partii; эта форма имеет вид «в один столбец».

В приложении 5 приведен пример формы на основе формы «3-2 - Заказ изданий почтовыми отделениями у типографий».

Для создания отчета необходимо переключиться в созданной БД на закладку *Отчеты* и нажать кнопку *Создать*. MS Access позволяет создавать отчеты следующими способами: с помощью *Автоотчета*, с помощью *Мастера отчетов*, с помощью *Конструктора*. Для создания отчетов к моей БД я буду использовать *Мастер отчетов*, и для этого в открывшемся окне «Новый отчет» необходимо выбрать *Мастер отчетов* и нажать кнопку *OK*.

Откроется первое окно мастера *Создание отчета*. В нем есть список *Таблицы и запросы*, в котором отображены все таблицы и запросы, созданные в этой БД и которые можно использовать как источники данных, нажав на стрелку прокрутки и выбрав из списка. При этом в области *Доступные поля* будут отражаться поля выбранной таблицы или запроса. Необходимые поля можно перенести в область *Выбранные поля* с помощью специальных кнопок. После выбора необходимых полей перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

В левой части открывшегося окна дается перечень используемых таблиц, одну из которых выбирают как основной источник данных, а в правой – область предварительного просмотра. После осуществления необходимого выбора перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

В левой части открывшегося окна дается перечень всех отобранных полей и можно выбирать уровни группировки, выбирая поле, по которому будет проводиться группировка, и используя специальные кнопки. Изменять уровни группировки можно с помощью кнопок *Уровень*. В правой же части окна находится область предварительного просмотра, в котором показываются все изменения с группировками в новом отчете. После выполнения необходимых изменений перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

В открывшемся окне можно задать сортировку записей по возрастанию или по убыванию, включающую до четырех полей, а также вычислить итоговые значения (сумма, среднее, наибольшее и наименьшее). Для выполнения вычислений итоговых значений следует нажать кнопку *Итоги*. Раскроется окно, в котором перечислены все поля таблиц и запросов, используемых в отчете с данными числового и денежного типов, а также перечень операций. Выбор операции осуществляется путем установки флажка в соответствующем окне строки с именем поля. Для продолжения работы с мастером необходимо нажать кнопку *OK*. После выполнения необходимых изменений перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

В правой части открывшегося окна можно выбрать макет отчета и ориентацию страниц (книжная или альбомная). Предлагается сделать выбор одного из следующих макет отчета: *ступенчатый*, *блок*, *структура 1*, *по левому краю 1* и другие. В левой части этого окна находится область предварительного просмотра, где отображаются все изменения с макетом отчета. После осуществления необходимого выбора перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

В правой части открывшегося окна можно выбрать стиль создаваемого отчета, а в левой находится область предварительного просмотра, где отображаются все изменения со стилем отчета. Можно выбрать один из следующих стилей отчета: *Деловой*, *Обычный*, *Строгий* и другие. После осуществления необходимого выбора перемещаемся в следующее окно мастера, нажав кнопку *Далее*.

Откроется последнее окно *Мастера отчета*, в котором можно изменить имя отчета. В нем также предлагается сделать дальнейший выбор: *Просмотреть отчет* или *Изменить форму отчета*. Можно также отметить пункт *Вывести справку по работе с отчетом*. После осуществления необходимых изменений нажимаем кнопку *Готово*. Если в качестве дальнейшего действия выбран вариант *Изменить форму отчета*, то откроется конструктор созданного отчета, в котором можно сделать необходимые изменения.

Таким образом, создаем следующие отчеты:

* «1-1 - Информация об изданиях» с полями: Издание.shifr\_izd, Издание.-nazv\_izd, Издание.fio\_redaktora\_izd, Типография.nom\_tipograf, Типография.-adres\_tipograf, Типография.fio\_direktora\_tipograf, Тираж.tirag\_izd, Тираж.-zena\_ekz\_izd;
* «2-1 - Информация о типографиях» с полями: Типография.nom\_tipograf, Типография.adres\_tipograf, Типография.fio\_direktora\_tipograf, Издание.shifr\_-izd, Издание.nazv\_izd, Тираж.tirag\_izd, Тираж.zena\_ekz\_izd, Почтовое\_отде-ление.nom\_pocht\_otd, Почтовое\_отделение.adres\_pocht\_otd, Заказ.obem\_-zakaza, Заказ.zena\_dostavki\_partii;
* «3-1 - Информация о почтовых отделениях» с полями: Почтовое\_отделе-ние.nom\_pocht\_otd, Почтовое\_отделение.adres\_pocht\_otd, Почтовое\_отделе-ние.fio\_nach\_pocht\_otd, Издание.shifr\_izd, Издание.nazv\_izd, Тираж.zena\_-ekz\_izd, Заказ.obem\_zakaza, Заказ.zena\_dostavki\_partii, Типография.nom\_ti-pograf, Типография.adres\_tipograf;
* «3-2 - Отчет по работе почтового отделения» с полями: Почтовое\_отделение.nom\_pocht\_otd, Почтовое\_отделение.adres\_pocht\_otd, Почтовое\_отделение.fio\_nach\_pocht\_otd, Издание.shifr\_izd, Издание.nazv\_izd, Заказ.obem\_zakaza, Тираж.zena\_ekz\_izd, Заказ.zena\_dostavki\_partii, вычисляемое поле «Цена партии с доставкой» выражением Тираж.zena\_ekz\_izd \* Заказ.obem\_zakaza + Заказ.zena\_dostavki\_partii, Типография.nom\_tipograf, Типография.adres\_tipograf. Этот отчет в отличие от предыдущего выводит информация по одному почтовому отделению, номер которого будет запрошен для поля Почтовое\_отделение.nom\_pocht\_otd.

В приложении 6 приведен пример отчета на основе отчета «3-2 - Отчет по работе почтового отделения».

**Изучение основных функций пакета BPwin**

BPwin позволяет аналитику создавать сложные модели бизнес-процессов при минимальных усилиях. BPwin поддерживает три методологии – IDEF0, IDEF3 и DFD. Каждая из них призвана решать свои специфические задачи.

Модель в BPwin рассматривается как совокупность работ, каждая из которых оперирует с некоторым набором данных. Работы изображаются в виде прямоугольников (блоков), данные – в виде стрелок (дуг).

Модели на основе стандарта IDEF0 необходимы для описания технологических процессов. Диаграммы потоков данных (DFD) описывают функции обработки информации, документы, объекты, а также сотрудников и подразделения. При этом используется набор элементов для источников, приемников и хранилищ данных. Для логики взаимодействия информационных потоков используется IDEF3. С его помощью дают характеристику как отдельной постановке реализации бизнес-процесса, так и полной последовательности действий.

В IDEFO различают пять типов стрелок:

**Вход (Input)** - материал или информация, которые используются или преобразуются работой для получения результата (выхода). Допус­кается, что работа может не иметь ни одной стрелки входа. Каждый тип стрелок подходит к определенной стороне прямоугольника, изобра­жающего работу, или выходит из нее. Стрелка входа рисуется как вхо­дящая в левую грань работы.

**Управление (Control)** - правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления. Стрелка управления рисуется как входящая в верхнюю грань работы.

**Выход (Output)** - материал или информация, которые производятся работой. Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла и не должна моделироваться. Стрелка выхода рисуется как исходящая из правой грани работы.

**Механизм (Mechanism)** - ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т. д. Стрелка механизма рисуется как входящая в нижнюю грань работы.

**Вызов (Call)** - специальная стрелка, указывающая на другую модель работы. Стрелка механизма рисуется как исходящая из нижней грани работы. Стрелка вызова исполь­зуется для указания того, что некоторая работа выполняется за преде­лами моделируемой системы. В BPwin стрелки вызова используются в механизме слияния и разделения моделей.

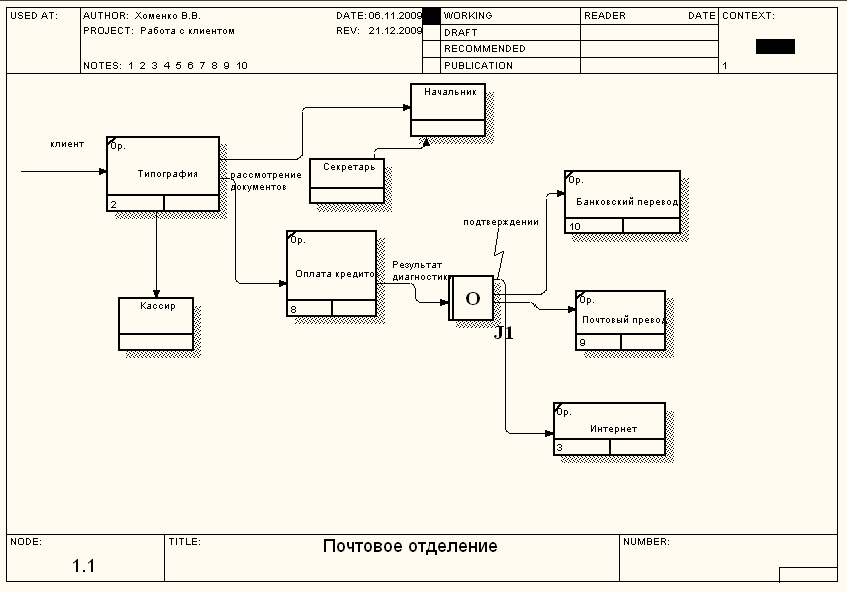
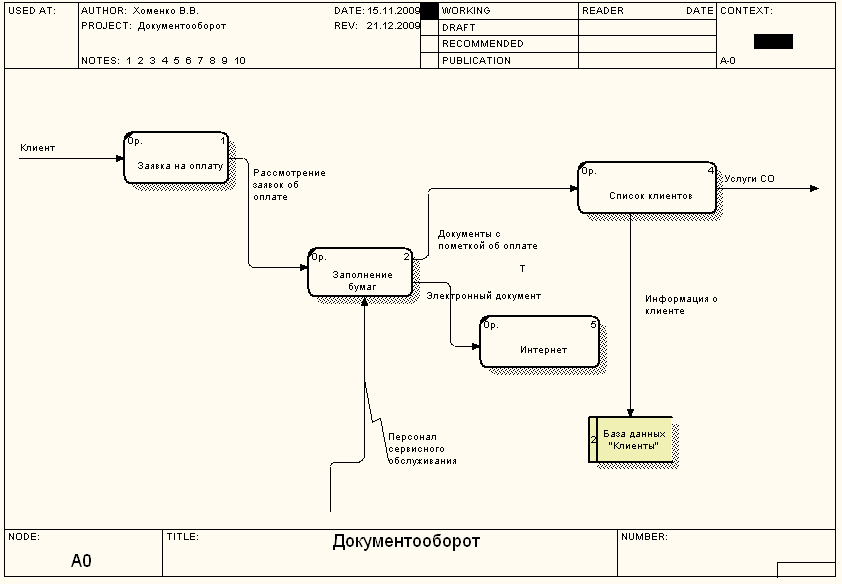
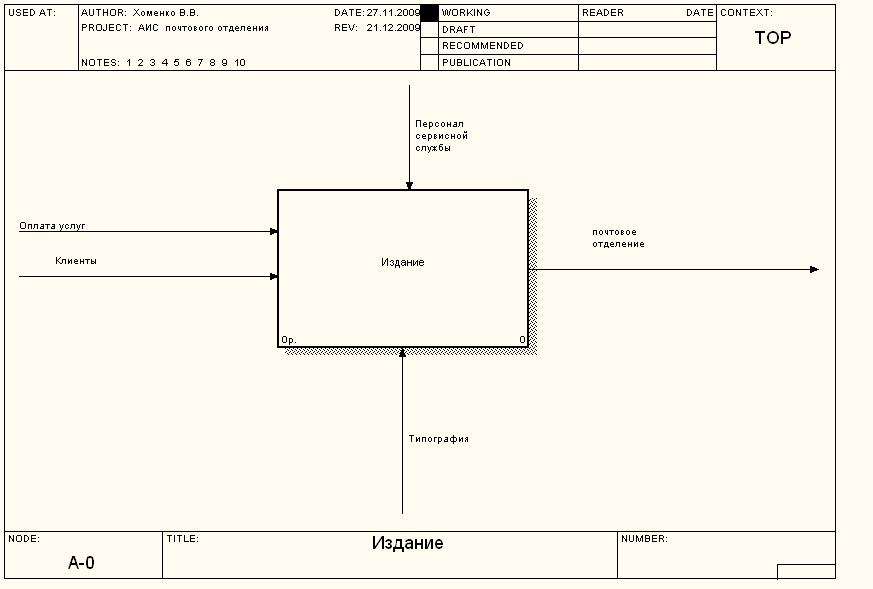
**Граничные стрелки.** Стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы диаграммы и заканчиваться у работы, и наоборот. Такие стрелки называются граничными.

Диаграммы потоков данных (Data flow diagramming, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпора­тивных системах обработки информации. DFD описывает:

1. функции обработки информации (работы);
2. документы (стрелки, arrow), объекты, сотрудников или отделы,  
   которые участвуют в обработке информации;
3. внешние ссылки (external references), которые обеспечивают интер­  
   фейс с внешними объектами, находящимися за границами модели­  
   руемой системы;
4. таблицы для хранения документов (хранилище данных, data store).

В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, DFD рассматривает систему как совокупность предметов.

Наличие в диаграммах DFD элементов для описания источников, приемников и хранилищ данных позволяет более эффективно и наглядно описать процесс документооборота. Однако для описания логики взаимо­действия информационных потоков более подходит IDEF3, называемая также workflow diagramming - методологией моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов. Диаграммы Workflow могут быть использованы в моделиро­вании бизнес-процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сот­рудников организации, например последовательность обработки заказа события, которые необходимо обработать за конечное время.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Финансово-экономический отдел |  | Начальник узла почтовой связи |  | Отдел информатизации и эксплуатации почтовой связи |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отдел кадров |  | Заместитель начальника |  | Отдел сортировки, обработки и доставки почты |

|  |  |  |
| --- | --- | --- |
| Коммерческий отдел |  | Отдел перевозки почты |

|  |  |  |
| --- | --- | --- |
| Хозяйственный отдел |  | Транспортный участок |

|  |
| --- |
| Отделения почтовой связи |

Рис.1. Организационно - производственная структура узла почтовой связи

ЗАКЛЮЧЕНИЕ

Проектирование базы данных представляет собой длительный и трудоемкий процесс. Качество созданной базы данных зависит от анализа предметной области и выбранной методологии проектирования. При неполном анализе предметной области в процессе эксплуатации созданной базы данных может возникать избыточное дублирование данных, а так же различные аномалии, что, скорее всего, приведет к потере необходимых данных и повторному проектированию базы данных. Процесс последующего проектирования базы данных не менее ответственный, так как необходимо четко выявить необходимые сущности и согласно связям между ними сформировать отношения. Процесс реализации базы данных средствами СУБД является преобразованием выполненного проектирования на ЭВМ.

Моей целью являлось проектирование базы данных «Почтовые отделения» средствами СУБД MS Access.

Первый этап проектирования заключается в тщательном анализе предметной области работы почтовых отделений. Задача почтовых отделений заключается в закупке печатных изданий. Эти печатные издания составляет редактор, типографии их печатают и поставляют согласно заказам почтовых отделений. В ходе анализа предметной области были выявлены группы параметров для каждого из объектов, информация о которых будет храниться в проектируемой базе данных. Были также выявлены:

* ограничения, накладываемые на информацию, которая будет храниться в базе данных;
* перечень запросов на предоставление справочной информации;
* возможные изменения информации, которая будет храниться в базе данных, а также добавление новой и удаление ненужной информации;
* перечень отчетов, которые будут необходимы работникам, которые будут обслуживать проектируемую базу данных.

Результаты проведенного анализа должны быть сгруппированы должным образом и представлены в виде отчета на бумажном носителе.

Второй этап проектирования делится на две фазы, первая из них это выбор методологии проектирования, а вторая – проектирование базы данных «Почтовые отделения». Для проектирования моей базы данных я выбрал метод «сущность-связь» и метод нормальных форм. С помощью первого из них формируются сущности, выявляются связи между ними и строится диаграмм ER-типов. Далее из сущностей и связей между ними формируются отношения согласно правилам формирования отношений, изложенным выше. С помощью второго метода уже сформированные отношения приводятся к БКНФ.

Третий этап проектирования заключается в реализации базы данных «Почтовые отделения» с помощью СУБД Microsoft Access. Выбор для реализации базы данных СУБД MS Access обосновывается следующим. Во-первых, эта СУБД входит в состав наиболее распространенного пакета прикладных программ по работе с документами Microsoft Office. Из MS Access возможен экспорт информации в другие приложения, входящие в MS Office (MS Word, MS Excel). Во-вторых, MS Access может являться дополнением к другим приложениям, в частности и СУБД, таким как Delphi, dBase и другие. Например, файлы MS Access могут являться основой для построения базы данных с помощью Delphi. Также MS Access может являться клиентом по отношению к Microsoft SQL Server. В-третьих, MS Access поддерживает механизм OLE, т.е. может связывать и внедрять объекты.

Полученную базу данных можно использовать в экспериментальных или показательных целях для проектирования полнофункциональной информационной системы с единой базой данных для работы почтовых отделений, а также типографий.

СПИСОК ЛИТЕРАТУРЫ

1. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.
2. Хомоненко А.Д., Гридин В.В. Microsoft Access. Быстрый старт. – СПб.: БХВ-Петербург, 2003. – 304 с.
3. Золотова С.И. Практикум по Access. – М.: Финансы и статистика, 2004. – 144 с.
4. Тиори Т., Фрай Дж. Проектирование структур баз данных: В 2-х кн. Кн. 1. Пер. с англ. – М.: Мир, 1985. – 287 с.
5. Чамберлин Д.Д., Астрахан М.М., Эсваран К.П., Грифитс П.П., Лори Р.А., Мел Д.В., Райшер П., Вейд Б.В. SEQUEL 2: унифицированный подход к определению, манипулированию и контролю данных //СУБД. - 1996. - №1. - С.144-159.
6. Чаудхари С. Методы оптимизации запросов в реляционных системах //СУБД. - 1998. - №3. - С.22-36.
7. Чен П. Модель "сущность-связь" - шаг к единому представлению о данных //СУБД. - 1995. - №3. - С.137-158.