

## **Титульный лист**

## Содержание

|   |    |
|---|----|
| Введение .....  | 3  |
| 1 глава. Аналитическая часть.....                               | 5  |
| 1.1.Описание предметной области. Постановка задачи .....        | 5  |
| 1.2.Выбор средств / методологии проектирования. Выбор СУБД..... | 8  |
| 1.3.Проектирование логической структуры базы данных .....       | 13 |
| 1.4.Проектирование физической структуры базы данных .....       | 17 |
| 2 глава. Практическая часть .....                               | 20 |
| 2.1.Инструкции по работе с базой данных .....                   | 20 |
| 2.2.Разработка интерфейса и реализация проекта.....             | 25 |
| 2.3.Оценка экономической эффективности проекта.....             | 33 |
| Заключение .....  | 38 |
| Список литературы .....   | 40 |

## Введение

Восприятие и понимание реального мира можно сравнить с последовательностью различных, но иногда взаимосвязанных, явлений. Очень давно люди делали попытки описать эти явления (это происходило даже тогда, когда люди не могли их никак понять). Такое описание называется данными. Существует еще одно определение: данные – это ведомости об окружающей среде, которые входят в систему, обрабатываются в ней и выходят из нее для последующей обработки.

Увеличение объема и сложности хранимых данных, увеличения круга пользователей в информационных системах привели к широкому распространению простых и наиболее удобных для понимания табличных СУБД.

Для обеспечения доступа к информации некоторого множества пользователей, что нередко расположены на большом расстоянии от места хранения информации или друг от друга, создаются базы данных, которые основываются на реляционной структуре.

В этих объектах различными путями можно решить специфические проблемы для безопасности, параллельных процессов и целостности данных, а также актуальную проблему санкционирования доступа.

В связи с тем, что почти всё нынешнее программное обеспечение включает в себя базы данных для хранения различной информации, то данная работа является актуальной.

Цель курсовой работы – точно и качественно высветлить понятие баз данных на примере СУБД Access при разработке БД для работника склада.

В работе необходимо решить такие основные задачи:

- провести анализ литературы с теории баз данных;
- описать предметную область;
- построить информационно-логическую модель предметной области;
- выполнить физическое проектирование базы данных в среде MS

Access;

– разработать интерфейса пользователя.

Объект исследования: сфера продажи и хранения товаров.

Предмет исследования: склад торгового предприятия.

Основные методы исследования: композиция, анализ утверждений, связанных с хранениям данных в табличном виде.

Курсовая работа состоит из введения, двух глав, заключения, списка литературы и базы данных.

## 1 глава. Аналитическая часть

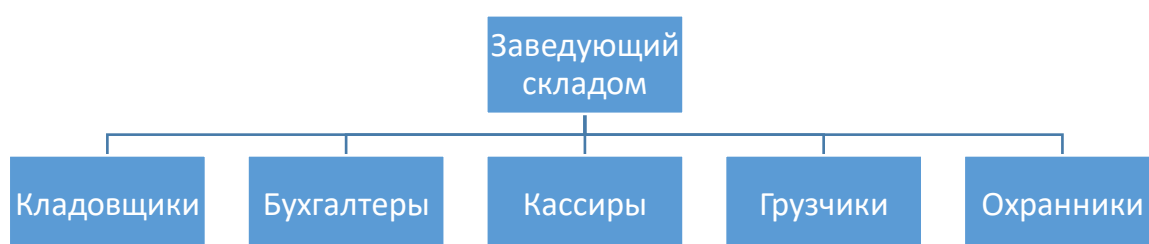
### 1.1. Описание предметной области. Постановка задачи

Предметная область выполнения курсовой работы – склад. С точки зрения проектирования базы данных в данной предметной области имеются следующие конечные пользователи: заведующий складом, кладовщик, бухгалтер, кассир.

Отдельное место в любой организационной структуре занимает склад, на территории которого храниться весь товар, который привезен для продажи от производителей или посредников. На складе ответственным лицом является кладовщик, а всеми организационными вопросами техники безопасности, складирования и иными занимается заведующий складом.

Целью создания БД является автоматизация основных процессов, которые протекают на оптовых складах, уменьшения времени на обработку больших объёмов информации по перемещению товаров по складу.

Структура аппарата склада и численность работников определяются видом склада, ассортиментом товаров, объемом товарооборота, сложностью и объемом выполняемых складских операций. Функционирование склада осуществляется под общим руководством торгового (коммерческого) отдела оптовой организации.



**Рисунок 1 – Типовая организационная структура оптового склада**

Основные функции и задачи сотрудников:

1. Заведующий складом обязан:

– Руководить работой склада по приему, хранению и отпуску товарно-материальных ценностей, по их размещению с учетом наиболее

рационального использования складских площадей, облегчения и ускорения поиска необходимых материалов, инвентаря и т.п.

- Обеспечивать сохранность складироваемых товарно-материальных ценностей, соблюдение режимов хранения, учет складских операций.

- Обеспечивать соблюдение правил оформления и сдачи приходно-расходных документов, составлять установленную отчетность по предприятию.

- Следить за наличием и исправностью противопожарных средств, состоянием помещений, оборудования и инвентаря на складе и обеспечивать их своевременный ремонт.

- Организовывать проведение погрузочно-разгрузочных работ на складе с соблюдением правил охраны труда, техники безопасности, производственной санитарии и противопожарной защиты, сбор, а так же хранение и своевременный возврат поставщикам погрузочного реквизита.

- Участвовать в проведении инвентаризаций товарно-материальных ценностей.

Функции бухгалтера склада:

- Осуществляет организацию бухгалтерского учета хозяйственно-финансовой деятельности склада и контроль экономного использования материальных, трудовых и финансовых ресурсов;

- Готовит рабочие планы счетов, форм первичных учетных документов, применяемых для оформления хозяйственных операций, по которым не предусмотрены типовые формы;

- Обеспечивает проведение инвентаризаций, контролирует проведение хозяйственных операций, соблюдение технологии обработки бухгалтерской информации и порядка документооборота;

- Обеспечивает рациональную организацию бухгалтерского учета и отчетности на складе;

– Принимает меры по предупреждению недостатков, незаконного расходования товарно-материальных ценностей, нарушений финансового и хозяйственного законодательства.

Функции кладовщика:

– Организация и осуществление приема, отпуска, хранения и сортировки, перемещения, переработки товарно-материальных и иных ценностей.

– Знать правила, порядок и условия складирования и хранения товарно-материальных и иных ценностей, с которыми приходится работать.

– Рационально использовать складские помещения и мощности.

– Должен обеспечить перемещение поступивших на склад ценностей к местам их хранения.

Кассир обязан:

- осуществлять операции, связанные с приемом и выдачей наличных денежных средств через кассу предприятия, обеспечивать сохранность всех принятых ценностей;

- получать в учреждениях банка наличность;

- заполнять кассовую книгу на основании приходных и расходных кассовых документов;

- подписывать приходные кассовые ордера и расходные документы сразу после получения или выдачи по ним наличности.

Грузчик имеет следующие обязанности:

– Осуществляет погрузку и выгрузку.

– Устанавливает лебедки, подъемные блоки, устройство временных скатов и другие приспособления для погрузки и выгрузки грузов.

– Осуществляет крепление и укрытие грузов на складах и транспортных средствах.

– Переносит щиты и трапы.

Охранник должен:

– Нести службу по охране объектов и материальных ценностей.

– Осуществлять проверку документов у проходящих на охраняемый объект (выходящих с объекта) лиц и контроль за ввозом и вывозом (выносом) материальных ценностей.

– Принимать под охрану от материально ответственных лиц оборудованные сигнализацией обособленные помещения.

Требуется создать многофункциональную БД для работника склада, с целью решения вопросов, связанных с хранением и обработкой информации о полученном от поставщика товаре, его реализации клиентам и т.д.

Основными носителями информации при автоматизированной обработке являются входные и выходные документы, т. е. утвержденной формы носители информации, имеющие юридическую силу.

Входная документация содержит первичную, не обработанную информацию, отражающую состояние объекта управления; заполняется вручную либо при помощи технических средств.

Выходная документация включает сводно-группировочные данные, полученные в результате автоматизированной обработки и изготавливается на печатающих устройствах машины.

В результате получим такие выходные документы:

- отчет о спросе на определенные товары;
- отчет о выборке из общего числа клиентов наиболее выгодных по объемам сделок купли-продажи;
- – отчет о выборке из общего числа клиентов наиболее выгодных по суммам сделок купли-продажи.

Создаваемая база данных – однопользовательская.

## **1.2.Выбор средств / методологии проектирования. Выбор СУБД**

Для проектирования базы данных использовать только некоторую СУБД не достаточно. Нужны также и специальные вспомогательные программы для моделирования предметной области и т.д.



ERwin является одним из CASE средств, позволяющих моделировать бизнес процессы. Он относится к категории I - CASE.

ERwin обеспечивает интеграцию моделей верхнего уровня с моделями нижнего уровня. Модели верхнего уровня разрабатываются на начальных стадиях проектирования информационных систем. Модели нижнего уровня разрабатываются на этапе создания программного кода и тестирования.

Модели данных, которые создаются помощью ERwin data modeler, разделяются на две категории:

Логические модели. Эти модели предназначены для представления элементов в терминах бизнес процессов. Для моделирования применяются диаграмма «сущность-связь», К - В модель (модель, основанная на ключах) и FA модель (полностью определенная модель).[9]

Физические модели. Эти модели предназначены для создания базы данных информационной системы. Они обеспечивают перевод понятий бизнес процессов на уровень понятий ИТ. Для моделирования применяются два вида моделей – трансформационная модель и схематичная модель базы данных (DBMS модель).

В связи с тем, что данная программа является наилучшей в своем секторе программного обеспечения, для проектирования БД работника склада будем использовать ее.

Для создания приложения по автоматизации маркетинговой деятельности необходимо выбрать СУБД, в которой будет создана информационная система.

В наше время одними из самых популярных СУБД являются SQL Server и MS Access. Рассмотрим их главные характеристики.

SQL Server – это СУБД, которая используя язык Transact SQL, реализует пересылку сообщений между компьютерами клиента и компьютером, где работает SQL Server, создавая при этом сетевую структуру.

Рассматриваемая СУБД состоит из механизма баз данных и компонентов СУБД, которая в свою очередь организует данные в виде таблиц

которые. SQL Server отвечает за поддержку структуры БД и решает следующие задачи:

- поддержка связей между данными;
- обеспечение корректного хранения данных, выполнение правил, которые регламентируют связи между ними;
- восстановление данных после аварии системы, перевод их в состояние, зафиксированное до сбоя (откат базы данных).

В первых версиях СУБД сходный код SQL Server основывался на коде Sybase SQL. Это позволило корпорации Microsoft выйти на рынок БД для предприятий, где монополистами были IBM и Oracle.

Microsoft SQL Server для разработки запросов использует версию языка SQL, получившую название T-SQL (Transact-SQL), что является реализацией SQL-93 (стандарт ISO) с множественными расширениями. Данная версия позволяет использовать дополнительные элементы синтаксиса для хранимых процедур, обеспечивать поддержку транзакций.

SQL Server поддерживает по трем сценариям избыточное дублирование данных:[8]

- Синхронизация с различными серверами: БД нескольких серверов могут синхронизироваться между собой. При этом изменения всех БД происходят друг от друга независимо, а при синхронизации выполняется сверка данных. Такой тип дублирования предусматривает возможности разрешения противоречий между разными БД.

- История изменений: все изменения, которые вносятся в базу данных непрерывно передаются конечным пользователям.

- Снимок: производится так называемый «снимок» БД, который отправляется сервером получателем.

Microsoft SQL Server 2012 обладает рядом возможностей, которые обеспечивают легкость установки, эксплуатации и развертывания, а также поддерживающих следующие функции:[7]

- масштабируемость;

- создание хранилищ данных;
- системную интеграцию с различным серверным программным обеспечением.

В состав СУБД входит огромное количество функций и инструментов, упрощающих процесс установки, управления или использования БД. SQL Server предоставляет администраторам полный набор инструментов, которые необходимы для тонкой настройки СУБД в составе промышленных онлайн-систем. SQL Server также корректно работает в однопользовательских системах небольшого масштаба, при этом издержки на их администрирование являются минимальными.[7]

Microsoft Access - настольная СУБД реляционного типа. В отличие от остальных СУБД, Access хранит всю информацию в одном файле, но распределяет их по таблицам, как и необходимо в реляционных БД. К таким данным относится не лишь информация в таблицах, а и другие объекты базы, которые будут ниже описаны.

Одним из достоинств Access является наличие очень простого графического интерфейса, который позволяет легко создать собственную базу данных, разрабатывать приложения, использовать при этом встроенные средства.

Это помогает избежать рутинных действий, облегчает работу неопытному пользователю.

Создание многопользовательской базы и получение нескольких пользователей одновременного доступа к общей базе возможно в сетях с одноранговой структурой или с файловым сервером. Локальная сеть обеспечивает программную и аппаратную поддержку обмена информацией между компьютерами. MS Access следит за распределением доступа различных пользователей к базе данных и обеспечивает их защиту.

В плане обеспечения целостности данных MS Access отвечает лишь моделям средней сложности. В нем не используются такие объекты как

хранимые процедуры и триггеры, что заставляет разработчиков создавать клиентские программы для поддержания бизнес-логики БД.

Однако, при указанных недостатках MS Access располагает большим количеством преимуществ.

В первую очередь отметим распространенность, что обусловлена принадлежностью СУБД компании Microsoft, операционные системы и программное обеспечение которой использует множество пользователей ПК. MS Access абсолютно совместим с ОС Windows, постоянно обновляется, поддерживает различные языки.[6]

MS Access предоставляет за небольшую стоимость огромное количество возможностей. Также отметим ориентированность на пользователей с различной профессиональной подготовкой. Это выражается в наличии различных вспомогательных средств, развитую справочную систему и интуитивно понятный интерфейс. Данные средства облегчают проектирование и создание БД, выборку из нее данных.[5]

В распоряжение непрограммирующим пользователям MS Access предоставляет разнообразные диалоговые средства, что позволяют им создавать приложения, при этом не прибегая к непосредственной разработке запросов с помощью языка SQL или к программированию вспомогательных модулей или макросов на языке VBA.

Также Access обладает большими возможностями по экспорту/импорту данных в разнообразные форматы через механизм ODBC: от текстовых файлов и таблиц Excel до любой серверной СУБД.

Еще одним немаловажным преимуществом MS Access является встроенные средства разработки приложений. Большое количество приложений, которые распространяемые среди пользователей, содержат некоторый объем кода языка Visual Basic for Applications.

VBA – единственное средство для выполнения различных стандартных задач в MS Access (построение команд SQL, обработка ошибок, работа с

переменными, использование Windows API), для создания сложных приложений.

Отметим, что Access предоставляет и дополнительные возможности по сравнению со стандартными СУБД. То есть, это не только простая и гибкая в использовании система, но и платформа для разработки приложений, работающих с базами данных.

Одной из самых необходимых функций каждой СУБД является защита информации, которая размещена в таблицах базы данных. [7]

СУБД MS Access хранит данные о защите в двух местах. При установке программа Setup создает в папке Program Files\Microsoft\_Office\Office стандартный файл для рабочей группы – System.mdw, который далее при запуске Access используется по умолчанию. Этот файл содержит информацию обо всех группах и пользователях. При создании новой базы данных MS Access сохраняет данные о правах, которые предоставляются конкретным группам и пользователям, непосредственно в файле базы данных.

Расположение файла рабочей группы находится в реестре Windows. Можно также использовать служебную программу операционной системы Windows – Wkadm.exe (администрирование рабочих групп) для редактирования текущего или создания новой рабочей группы. Также можно выбрать необходимый файл рабочей группы при выполнении приложения, задав при этом соответствующий параметр в командной строке.

### **1.3.Проектирование логической структуры базы данных**

Сущность (объектное множество, таблица) - это собирательное понятие, абстракция реально существующего процесса, объекта или явления, о котором необходимо хранить информацию.

Связь - ассоциирование двух и более сущностей. Если бы назначением БД было только хранение отдельных, не связанных между собой данных, то ее структура могла быть очень простой. Однако одно из основных требований к организации базы данных - это обеспечение возможности отыскания одних

сущностей по назначениям других, для чего необходимо установить между ними определенные связи.

Взаимосвязи между таблицами БД могут быть типизированы по следующим основным видам:

1. отношение “один к одному” (1:1) означает, что каждая запись одной таблицы соответствует только одной записи в другой таблице;

2. отношение “один ко многим” (1:M) возникает, когда одна запись взаимосвязана со многими другими;

3. отношение “многие к одному” означает, что многие записи связаны с одной (M:1);

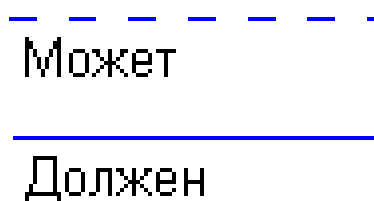
4. отношение “многие ко многим” (M:N) возникает между двумя таблицами в тех случаях, когда:

– одна запись из первой таблицы может быть связана более чем с одной записью из второй таблицы;

– одна запись из второй таблицы может быть связана более чем с одной записью из первой таблицы.

Результатом концептуального проектирования базы данных является ER-модель.

Каждая связь может иметь одну из двух модальностей связи (рис.2):



### **Рисунок 2 – Модальность связей**

Модальность «может» означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, может быть и не связан ни с одним экземпляром.

Модальность «должен» означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.[2]

При построении инфологической модели предметной области устанавливаются связи между выявленными информационными объектами.

Рассмотрим следующие сущности: Поставщики, Покупатели, Товар на складе, Сделки о продаже и определим между ними связи.

Для решения исходной задачи, необходимы такие входные документы:

- список поставщиков (код поставщика, название фирмы-поставщика, адрес, телефон);

- список покупателей (код покупателя, название фирмы-покупателя, адрес, телефон);

- список товаров (код товара, поставщик, название товара, единицы измерения, количество, цена покупки за единицу товара, цена продажи за единицу товара);

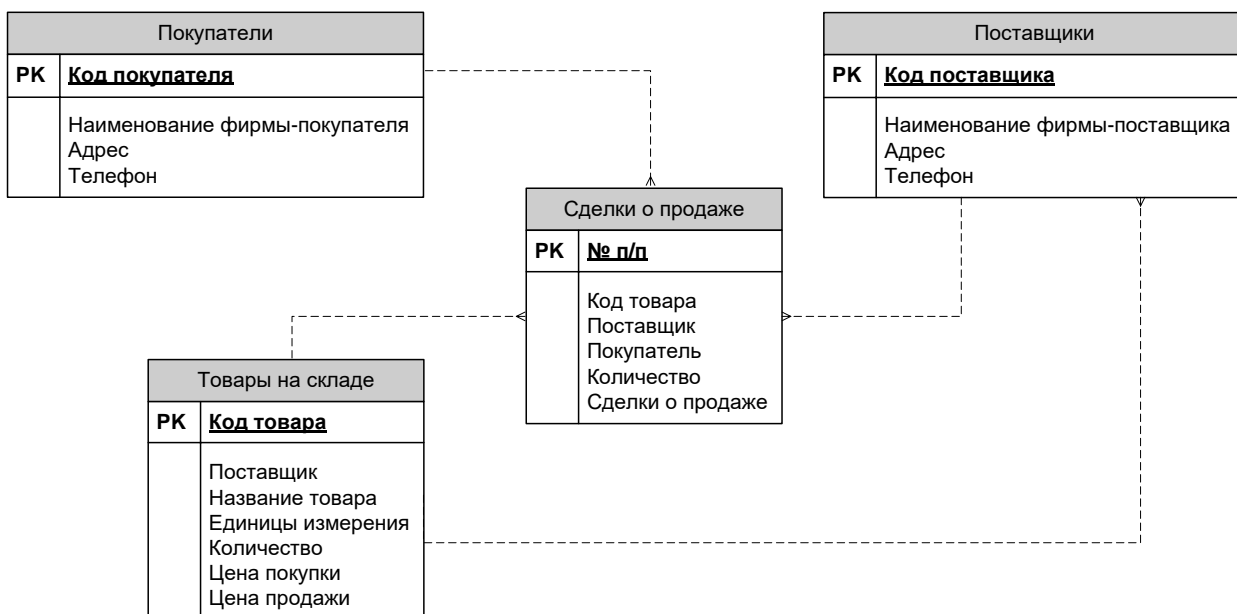
- перечень сделок (код товара, поставщик, покупатель, количество проданного товара).

У каждого товара, который имеет свой уникальный код, есть определенный поставщик, а один поставщик может поставлять разные товары. Поэтому связь между сущностями Товары на складе и Поставщик многие к одному (М:1).[3]

Каждый покупатель может купить несколько разных товаров, а один товар может быть куплен только определенным покупателем. Поэтому связь между сущностями Покупатели и Сделки о продаже один ко многим (1: М).

Каждый поставщик может поставлять несколько разных товаров, а один покупатель может несколько товаров разных поставщиков. Поэтому связь между сущностями Покупатели и Поставщики многие ко многим (N: М).

В результате получим ER-диаграмму:



**Рисунок 3 – ER - диаграмма**

Стоит отметить, что в построенной модели есть ограничения целостности. Рассмотрим основные из них:

– Количество товаров на складе должно быть неотрицательным.

Это достигается установкой свойства Правило проверки (рис.4):

| Общие                   | Подстановка |
|-------------------------|-------------|
| Размер поля             | Длинно      |
| Формат поля             |             |
| Число десятичных знаков | Авто        |
| Маска ввода             |             |
| Подпись                 |             |
| Значение по умолчанию   | 0           |
| Правило проверки        | >=0 ...     |
| Сообщение об ошибке     |             |
| Обязательное поле       | Нет         |
| Индексированное поле    | Нет         |
| Выравнивание текста     | Общее       |

**Рисунок 4 – Установка правила проверки**

– Цена покупки товара и цена продажи – в рублях. Это достигается установкой необходимого типа данных при проектировании физической структуры данных.[4]

–Количество проданного товара не должно превышать суммарное количество товара на складе.

– Названия покупателей должны использоваться только из соответственного справочника.



– Названия поставщиков должны использоваться только из соответственного справочника.

Стоит отметить, что для всех полей установлены необходимые типы данных и их свойства (формат, длина и т.д.)

Информация в таблицах должна удовлетворять таким принципам.[9]

1. Атрибуты (их значения) должны быть атомарными (то есть, каждое значение, которое содержится на пересечении колонки и строки, не должно расчленяться на несколько значений).[8]

2. Значения всех атрибутов должны принадлежать одному типу.

3. Каждое поле имеет уникальное имя.

4. Каждая запись в таблице есть уникальной.

5. Последовательность записей и полей в таблице не является существенной.

#### **1.4.Проектирование физической структуры базы данных**

Рассмотрим применение MS Access для создания ИС для оптового склада и на практике продемонстрируем основные возможности СУБД.

Для создаваемой базы данных требуются следующие информационные объекты (таблицы):

«Покупатели» – для хранения информации о покупателях.

«Поставщики» – для хранения информации о поставщиках.

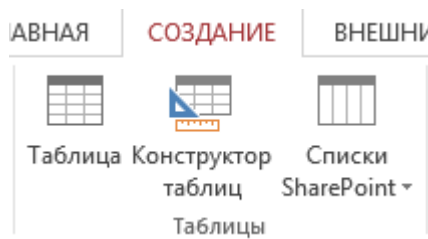
«Товары на складе» - для хранения информации о товарах на складе.

«Сделки о продаже» - для хранения информации о сделках о продаже.

В процессе создания таблиц, имеющих ссылки на записи из других таблиц, устанавливается связь. Некоторые таблицы называют справочниками, если они содержат информацию, которая может быть использована при заполнении других таблиц. Они по структуре простые и имеют первичный ключ, их называют главными.

Рассмотрим процесс создания таблицы в СУБД MS Access 2013, а именно с помощью конструктора таблиц. [7]

После запуска СУБД и создания базы данных нужно нажать на ленту «Создание» и выбрать в разделе «Таблицы» Конструктор таблиц (рис.5.)



**Рисунок 5 – Выбор конструктора таблиц.**

После этого откроется окно в котором нужно указать название поля, тип данных, описание (в случае необходимости), свойства полей, ключевое поле. Внешний вид окна конструктора таблицы Покупатели показан на рис. 6 – 9:

| Покупатели                |                |
|---------------------------|----------------|
| Имя поля                  | Тип данных     |
| Код покупателя            | Числовой       |
| Название фирмы-покупателя | Короткий текст |
| Адрес                     | Короткий текст |
| Телефон                   | Короткий текст |

**Рисунок 6 – Таблица Покупатели в режиме конструктора**

| Поставщики                |                |
|---------------------------|----------------|
| Имя поля                  | Тип данных     |
| Код поставщика            | Числовой       |
| Название фирмы-поставщика | Короткий текст |
| Адрес                     | Короткий текст |
| Телефон                   | Короткий текст |

**Рисунок 7 – Таблица Поставщики в режиме конструктора**

| Товар на складе             |                |
|-----------------------------|----------------|
| Имя поля                    | Тип данных     |
| Код товара                  | Числовой       |
| Поставщик                   | Числовой       |
| Название товара             | Короткий текст |
| Единицы измерения           | Короткий текст |
| Количество                  | Числовой       |
| Цена покупки за единицу тов | Денежный       |
| Цена продажи за единицу тов | Денежный       |

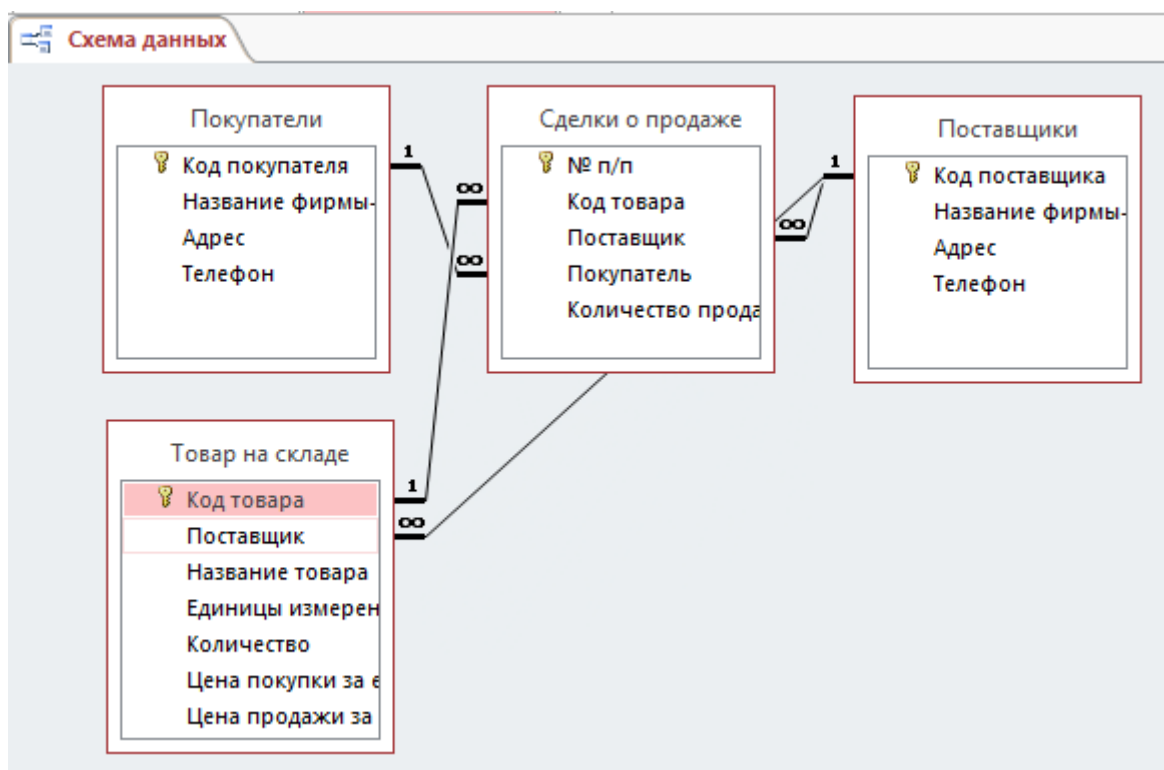
**Рисунок 8 – Таблица Товары на складе в режиме конструктора**

| Сделки о продажах            |            |
|------------------------------|------------|
| Имя поля                     | Тип данных |
| № п/п                        | Счетчик    |
| Код товара                   | Числовой   |
| Поставщик                    | Числовой   |
| Покупатель                   | Числовой   |
| Количество проданного товара | Числовой   |

**Рисунок 9 – Таблица Сделки по продажам в режиме конструктора**  
 Далее перейдем к вводу данных в таблицы. Заполненная данными таблица Покупатели имеет такой вид (рис. 7):

| Покупатели |                |                           |             |           |
|------------|----------------|---------------------------|-------------|-----------|
|            | Код покупателя | Название фирмы-покупателя | Адрес       | Телефон   |
| +          | 100            | Пилот                     | Ленина,2    | 544-54-55 |
| +          | 101            | Орлан                     | Чапаева,1   | 123-24-68 |
| +          | 102            | ЧП "Иванович"             | Мира,43     | 845-12-13 |
| +          | 103            | Лина-3                    | Шелянова,1  | 124-65-45 |
| +          | 104            | Ашан                      | Пушкина,43  | 216-54-56 |
| +          | 105            | ЧП "Суровский"            | Цветочная,4 | 521-46-58 |

**Рисунок 7 – Заполненная таблица Заявки**  
 Создадим связи между таблицами:



**Рисунок 8 – Схема данных**

## 2 глава. Практическая часть

### 2.1.Инструкции по работе с базой данных

Формы – это элементы базы данных, которые используются для ввода и коррекции данных.

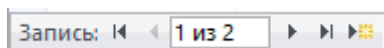
Рассмотрим форму Покупатели и введем с ее помощью данные в таблицы (рис.9):

| № п/п | Код товара | Поставщик | Количество |
|-------|------------|-----------|------------|
| 1     | 408        | Пеперс    | 15         |
| 11    | 405        | Свинтесс  | 20         |
| * (№) | 0          |           | 0          |

**Рисунок 9 – Составная форма Покупатели**

Для ввода данных в форму нужно сначала ввести (или найти) покупателя в главной форме, а потом в подчиненной ввести купленные им товары.

Стоит отметить, что для ввода данных можно использовать кнопки перехода (рис.10):



**Рисунок 10 – Кнопки перехода**

Аналогично можно ввести или откорректировать данные для других таблиц с помощью таких форм (рис.11 – 13):

Товар на складе

Товар на складе

Код товара: 400

Поставщик: Свинтесс

Название товара: Сок Добрый

Единицы измерения: л

Количество: 200

Цена покупки за единицу товара: 98,00 Р

Цена продажи за единицу товара: 120,00 Р

| № п/п | Поставщик | Покупатель | Количество |
|-------|-----------|------------|------------|
| 3     | Свинтесс  | Орлан      | 20         |
| *     | (№)       |            | 0          |

**Рисунок 11 – Форма Товары на складе**

Поставщики

Поставщики

Код поставщика: 200

Название фирмы-поставщика: ЧП "Макеев"

Адрес: Мира,12

Телефон: 456-64-55

**Рисунок 12 – Форма Поставщики**

Сделки о продаже

Сделки о продаже

№ п/п: 1

Код товара: 408

Поставщик: Пеперс

Покупатель: Пилот

Количество проданного товара: 15

**Рисунок 13 – Форма Сделки о продаже**

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.[3]

При печати таблиц и запросов информация выдается практически в том виде, в котором хранится. Часто возникает необходимость представить данные в виде отчетов, которые имеют традиционный вид и легко читаются. Подробный отчет включает всю информацию из таблицы или запроса, но содержит заголовки и разбит на страницы с указанием верхних и нижних колонтитулов.

В Microsoft Access можно создавать отчеты различными способами:

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

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. [1]

Создадим отчет для таблицы Товары на складе с помощью мастера.

Для этого вызовем мастер создания отчетов (Создание – Отчеты – Мастер отчетов). Далее нужно:

- выбрать таблицу Товары на складе и переместить все поля кроме Код товара в категорию Выбранные;
- в уровень группировки выбрать поле Поставщик;
- выберем итоговую операцию Сумма;
- выполним сортировку по полю Название товара;

- отметим макет Структура, ориентация Альбомная и нажмем кнопку Далее;
- в последнем окне задаем имя Товары.

В результате получим отчет, показанный ниже:

| Товар на складе |   |                 |            |                                |                                |  |
|-----------------|---|-----------------|------------|--------------------------------|--------------------------------|--|
| Поставщик       | Название товара                         | Единицы измерен | Количество | Цена покупки за единицу товара | Цена продажи за единицу товара |  |
| 200             | Сок Дар                                 | л               | 650        | 68,50 Р                        | 86,00 Р                        |  |
|                 | Халва Колос                             | кг              | 240        | 150,00 Р                       | 180,00 Р                       |  |
|                 | Итого для 'Поставщик' = 200 (2 записей) |                 |            |                                |                                |  |
| Мак             |   |                 | 650        |                                |                                |  |
| 201             | Кофе Арабика                            | кг              | 250        | 190,00 Р                       | 212,00 Р                       |  |
|                 | Лепша                                   | кг              | 600        | 130,00 Р                       | 155,00 Р                       |  |
|                 | Итого для 'Поставщик' = 201 (2 записей) |                 |            |                                |                                |  |
| Мак             |   |                 | 600        |                                |                                |  |
| 202             | Мюсли Семья                             | кг              | 450        | 69,00 Р                        | 78,00 Р                        |  |
|                 | Сок Добрый                              | л               | 200        | 98,00 Р                        | 120,00 Р                       |  |
|                 | Итого для 'Поставщик' = 202 (2 записей) |                 |            |                                |                                |  |
| Мак             |   |                 | 450        |                                |                                |  |
| 203             | Квас Тарас                              | л               | 350        | 52,00 Р                        | 69,50 Р                        |  |
|                 | Зефир Наш                               | кг              | 400        | 80,00 Р                        | 105,00 Р                       |  |
|                 | Итого для 'Поставщик' = 203 (2 записей) |                 |            |                                |                                |  |
| Мак             |   |                 | 400        |                                |                                |  |
| 204             | Чипсы Люкс                              | пач             | 300        | 25,80 Р                        | 32,10 Р                        |  |
|                 | Мёд Сладкий                             | л               | 500        | 45,00 Р                        | 60,00 Р                        |  |
|                 | Итого для 'Поставщик' = 204 (2 записей) |                 |            |                                |                                |  |
| Мак             |   |                 | 500        |                                |                                |  |

**Рисунок 14 – Отчет Товар на складе**

Воспользовавшись функцией Автоотчет, создадим отчет для таблицы Поставщики.

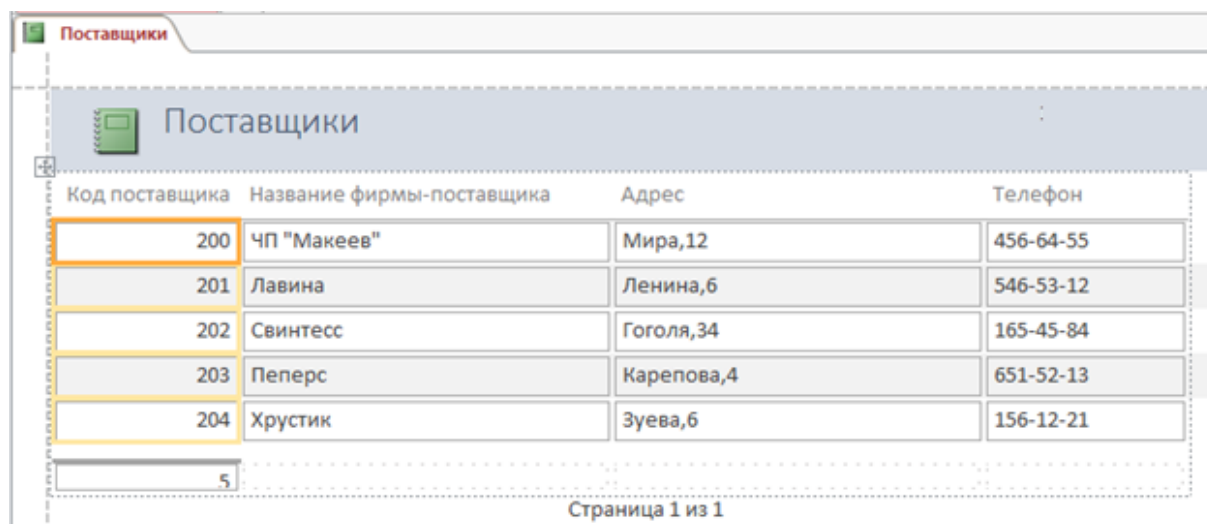
Для создания Автоотчета необходимо выполнить следующие действия:

- В окне базы данных щелкнуть на вкладке Отчеты и затем щелкнуть на кнопке Создать. Появится диалоговое окно Новый отчет.
- Выделить в списке пункт Автоотчет: в столбец или Автоотчет: ленточный.
- В поле источника данных щелкнуть на стрелке и выбрать в качестве источника данных таблицу Поставщики.
- Щелкнуть на кнопке ОК.

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.

В меню Файл щелкнуть на команде Сохранить. В окне Сохранение в поле Имя отчета указать название отчета – Поставщики и щелкнуть на кнопке ОК.



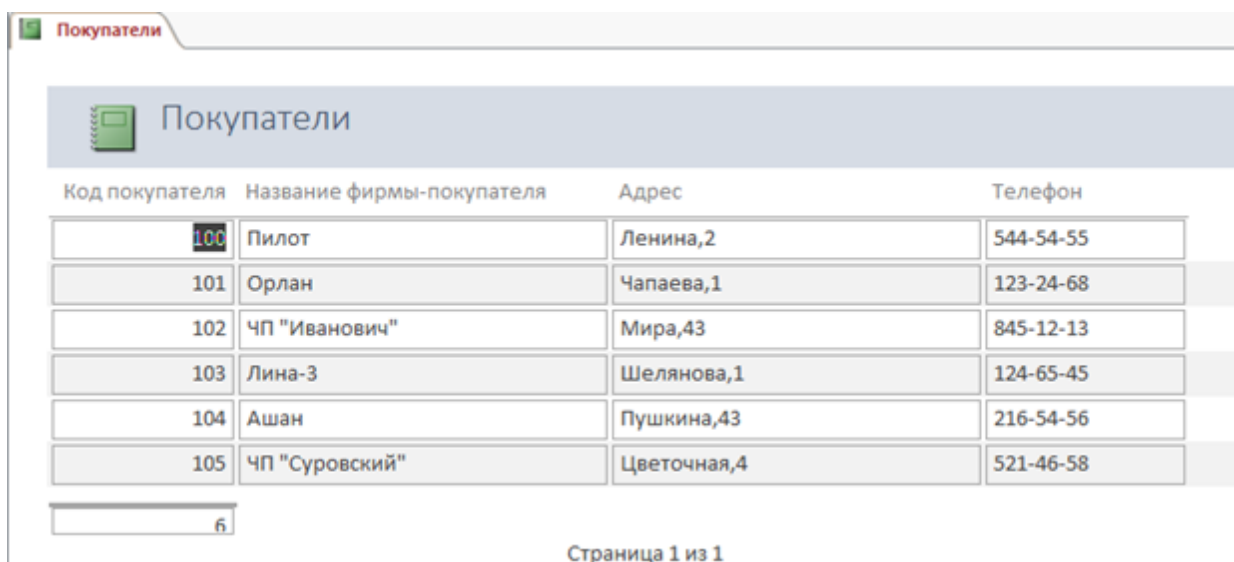
| Код поставщика | Название фирмы-поставщика | Адрес     | Телефон   |
|----------------|---------------------------|-----------|-----------|
| 200            | ЧП "Макеев"               | Мира,12   | 456-64-55 |
| 201            | Лавина                    | Ленина,6  | 546-53-12 |
| 202            | Свинтесс                  | Гоголя,34 | 165-45-84 |
| 203            | Пеперс                    | Кареева,4 | 651-52-13 |
| 204            | Хрустик                   | Зуева,6   | 156-12-21 |

5

Страница 1 из 1

**Рисунок 15 – Отчет Поставщики**

Аналогично создадим отчеты для оставшихся таблиц:



| Код покупателя | Название фирмы-покупателя | Адрес       | Телефон   |
|----------------|---------------------------|-------------|-----------|
| 100            | Пилот                     | Ленина,2    | 544-54-55 |
| 101            | Орлан                     | Чапаева,1   | 123-24-68 |
| 102            | ЧП "Иванович"             | Мира,43     | 845-12-13 |
| 103            | Лина-3                    | Шелянова,1  | 124-65-45 |
| 104            | Ашан                      | Пушкина,43  | 216-54-56 |
| 105            | ЧП "Суровский"            | Цветочная,4 | 521-46-58 |

6

Страница 1 из 1

**Рисунок 16 – Отчет Покупатели**



| Сделки о продаже |            |             |                |                              |    |
|------------------|------------|-------------|----------------|------------------------------|----|
| № п/п            | Код товара | Поставщик   | Покупатель     | Количество проданного товара |    |
| 1                | 408        | Пеперс      | Пилот          |                              | 15 |
| 2                | 402        | Лавина      | ЧП "Иванович"  |                              | 14 |
| 3                | 400        | Свинтесс    | Орлан          |                              | 20 |
| 4                | 402        | Лавина      | Лина-3         |                              | 30 |
| 5                | 401        | ЧП "Макеев" | ЧП "Суровский" |                              | 40 |
| 6                | 405        | Свинтесс    | Ашан           |                              | 15 |
| 7                | 408        | Пеперс      | Орлан          |                              | 25 |
| 8                | 407        | Лавина      | Лина-3         |                              | 20 |
| 9                | 406        | ЧП "Макеев" | Орлан          |                              | 30 |
| 10               | 401        | ЧП "Макеев" | ЧП "Суровский" |                              | 35 |
| 11               | 405        | Свинтесс    | Пилот          |                              | 20 |
| 12               | 403        | Пеперс      | Лина-3         |                              | 24 |
| 13               | 404        | Хрустик     | ЧП "Иванович"  |                              | 27 |

**Рисунок 17 – Отчет Сделки о продаже**

## 2.2. Разработка интерфейса и реализация проекта

Запрос (query) – это средство выбора необходимой информации из базы данных. Вопрос, сформированный по отношению к базе данных, и есть запрос. Применяются два типа запросов: по образцу (QBE – Query by example) и структурированный язык запросов (SQL – Structured Query Language).

QBE - запрос по образцу – средство для отыскания необходимой информации в базе данных. Он формируется не на специальном языке, а путем заполнения бланка запроса в окне Конструктора запросов.

SQL – запросы – это запросы, которые составляются (программистами) из последовательности SQL – инструкций. Эти инструкции задают, что надо сделать с входным набором данных для генерации выходного набора. Все запросы Access строит на основе SQL – запросов, чтобы посмотреть их, необходимо в активном окне проектирования запроса выполнить команду Вид/SQL.

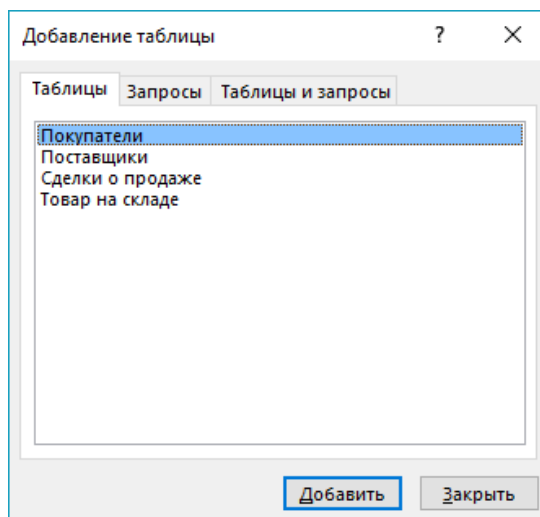
Существует несколько типов запросов: на выборку, на обновление, на добавление, на удаление, перекрестный запрос, создание таблиц. Наиболее

распространенным является запрос на выборку. Запросы на выборку используются для отбора нужной пользователю информации, содержащейся в таблицах. Они создаются только для связанных таблиц.

Рассмотрим создание запросов для реализации задач заказчика.

Создадим запрос Спрос на товар с помощью режима QBE.

Для этого нужно вызвать конструктор запросов и добавить необходимые таблицы с окна, которое откроется:



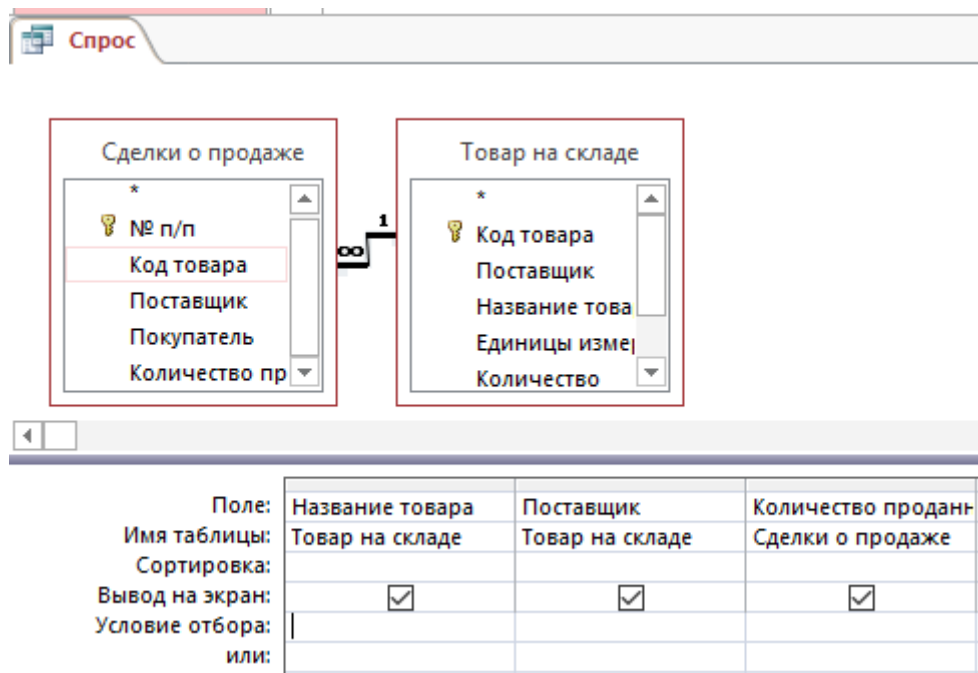
**Рисунок 18 – Окно Добавление таблицы**

Окно Конструктора состоит из двух частей – верхней и нижней. В верхней части окна размещается схема данных запроса, которая содержит список таблиц – источников и отражает связь между ними.

В нижней части окна находится Бланк построения запроса QBE (Query by Example), в котором каждая строка выполняет определенную функцию:

- Поле – указывает имена полей, которые участвуют в запросе.
- Имя таблицы – имя таблицы, с которой выбрано это поле.
- Сортировка – указывает тип сортировки.
- Вывод на экран – устанавливает флажок просмотра поля на экране.
- Условия отбора – задаются критерии поиска.
- Или – задаются дополнительные критерии отбора.

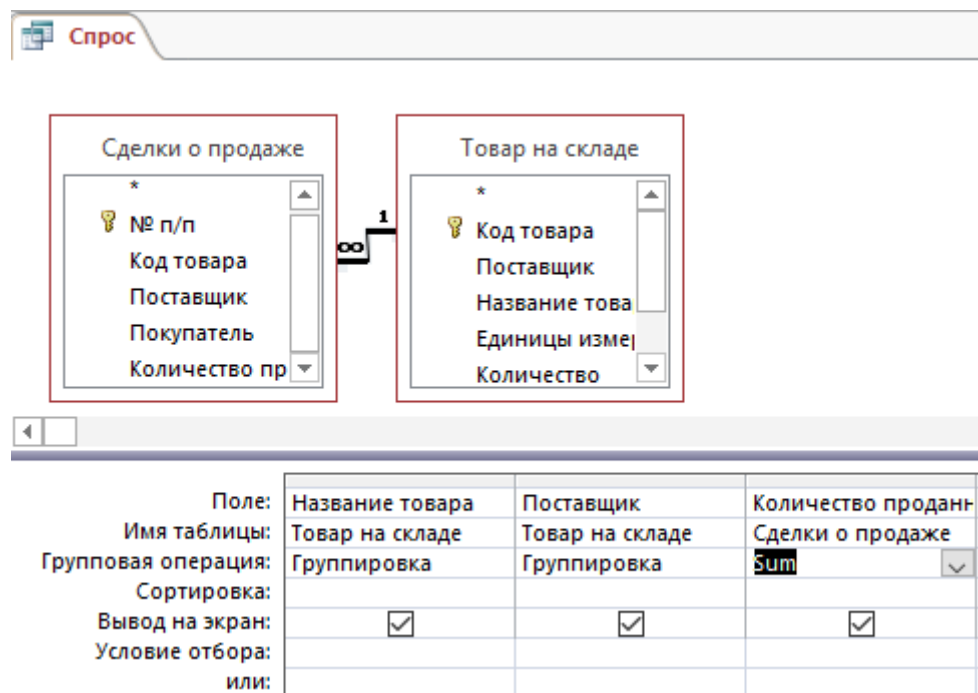
Введем информацию в конструктор запросов так, как показано ниже:



**Рисунок 19 – Ввод полей в конструктор запросов**

Для вычисления суммы количества проданного товара нужно задать групповые операции. Для этого нужно нажать правую кнопку мыши в строке Поле и выбрать пункт Итоги.

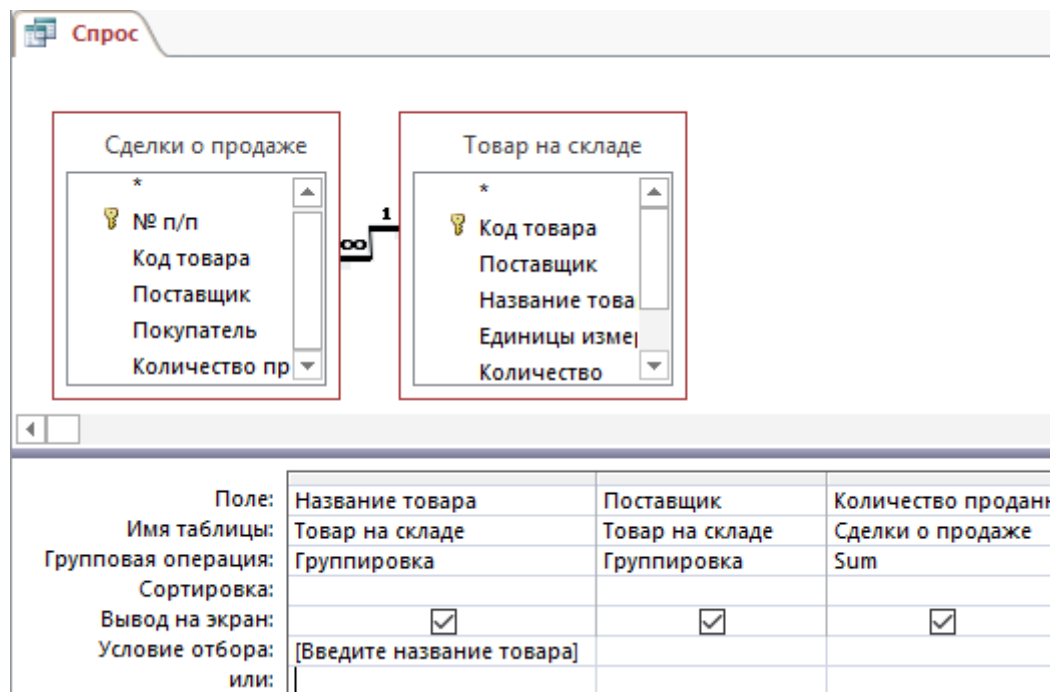
Далее нужно выбрать необходимую функцию (Sum()) в строке группировка:



**Рисунок 20 – Реализация группировки данных**

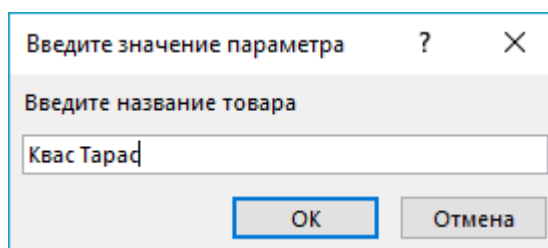
При запуске запроса нужно, чтобы пользователь вводил конкретный товар и по нему отображалась бы информация. Для этого создадим окно для ввода параметра, записав в строке Условие отбора текст:

[Введите название товара]



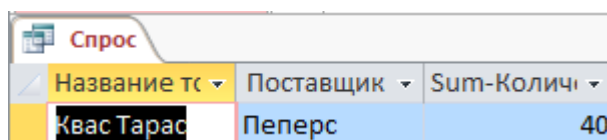
**Рисунок 21 – Создание параметрического запроса**

Запустим запрос на выполнения и введем название товара:



**Рисунок 22 – Диалоговое окно параметрического запроса**

В результате получим:



**Рисунок 23 – Запрос Спрос**

Создадим запрос Выгодные по объемам, который выполняет отбор из общего числа клиентов наиболее выгодных по объемам сделок купли-продажи.

Для этого используем SQL-выражения, которые имеют следующий синтаксис:

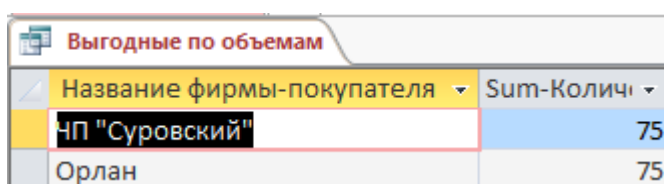
```
SELECT Список столбцов  
FROM Список таблиц  
[WHERE Условия выбора строк]  
[ORDER BY Спецификация сортировки];
```

В квадратных скобках указываются предложения, которые могут быть опущены.

Созданный запрос будет иметь такой код:

```
SELECT TOP 1 Покупатели.[Название фирмы-покупателя],  
Sum([Сделки о продаже].[Количество проданного товара]) AS [Sum-  
Количество проданного товара]  
FROM Покупатели INNER JOIN [Сделки о продаже] ON  
Покупатели.[Код покупателя] = [Сделки о продаже].Покупатель  
GROUP BY Покупатели.[Название фирмы-покупателя]  
ORDER BY Sum([Сделки о продаже].[Количество проданного товара])  
DESC;
```

В результате получим:



| Название фирмы-покупателя | Sum-Колич |
|---------------------------|-----------|
| ЧП "Суровский"            | 75        |
| Орлан                     | 75        |

**Рисунок 24 – Результат выполнения запроса**

То есть, есть 2 покупателя, которые купили максимальное количество товаров.

Аналогично создадим запрос Выгодные по суммам, который выполняет отбор из общего числа клиентов наиболее выгодных по суммам сделок купли-продажи.

```
SELECT TOP 1 Покупатели.[Название фирмы-покупателя],  
Sum([Количество проданного товара]*[Цена продажи за единицу  
товара]) AS Сумма
```

**FROM** [Товар на складе] **INNER JOIN** (Покупатели **INNER JOIN** [Сделки о продаже] **ON** Покупатели.[Код покупателя] = [Сделки о продаже].Покупатель) **ON** [Товар на складе].[Код товара] = [Сделки о продаже].[Код товара]

**GROUP BY** Покупатели.[Название фирмы-покупателя], [Сделки о продаже].[Количество проданного товара], [Товар на складе].[Цена продажи за единицу товара]

**ORDER BY** Sum([Количество проданного товара]\*[Цена продажи за единицу товара]) **DESC**;

В результате получим:

| Название фирмы-покупателя | Сумма      |
|---------------------------|------------|
| ЧП "Суровский"            | 7 200,00 Р |

**Рисунок 25 – Результат выполнения запроса**

Опишем процедуру создания главной кнопочной формы как основного элемента интерфейса.

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

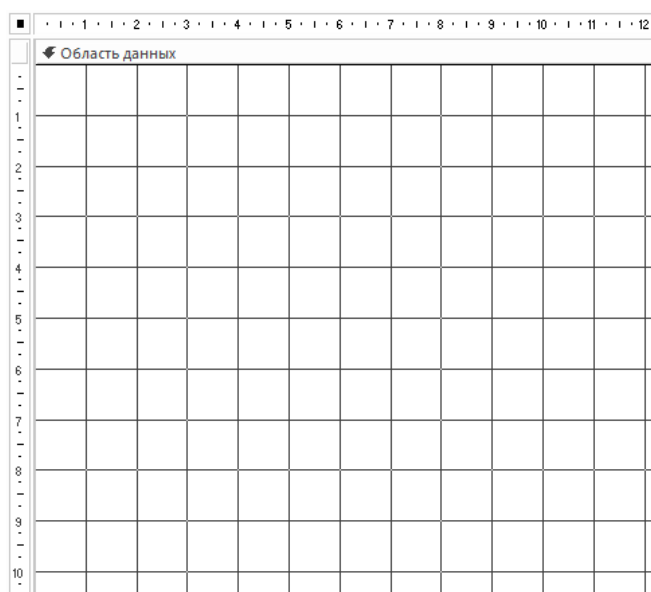
Запросы и таблицы не являются элементами главной кнопочной формы. Поэтому для создания кнопок Запросы или Таблицы на кнопочной форме можно использовать макросы. Сначала в окне базы данных создают макросы «Открыть Запрос» или «Открыть Таблицу» с уникальными именами, а затем в кнопочной форме создают кнопки для вызова этих макросов.

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).

Технология создания кнопочных форм следующая:

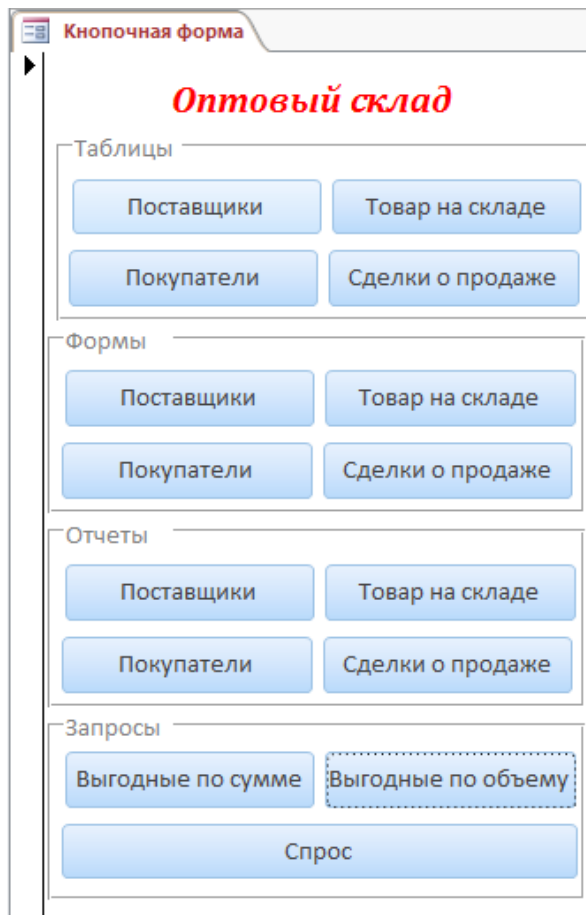
- создать страницу главной кнопочной формы (ГКФ);
- создать необходимое количество страниц подчиненных кнопочных форм (например, формы для ввода данных, для отчетов, для запросов и т.д.);
- создать элементы главной кнопочной формы;
- создать элементы для кнопочных форм отчетов и форм ввода или изменения данных;
- создать макросы для запросов или для таблиц с уникальными именами;
- создать элементы для кнопочных форм запросов или таблиц.

Добавим пустую форму:



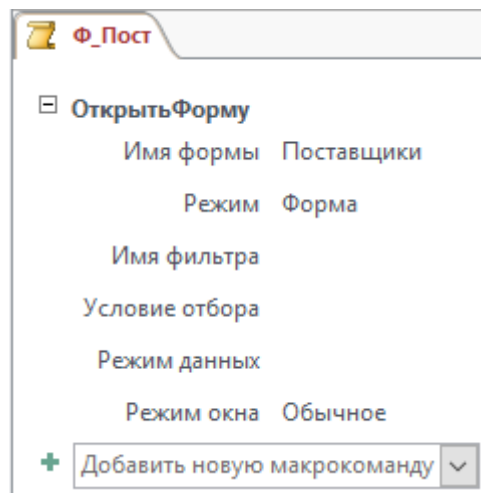
**Рисунок 26 – Пустая форма**

Разместим на ней надпись «Оптовый склад» и группы кнопок в фреймах:



**Рисунок 27 –Интерфейс кнопочной формы**

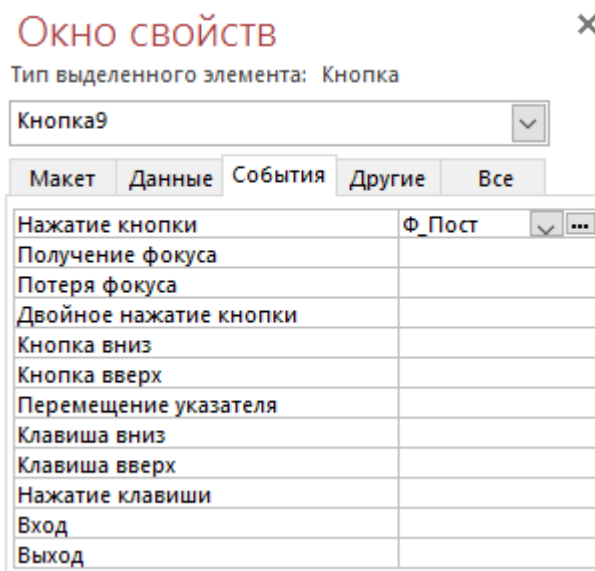
Далее создадим макросы для автоматизации формы. Например, макрос для открытия формы Поставщики имеет вид:



**Рисунок 28 – Макрос для формы Поставщики**

После создания всех макросов назначим из каждой кнопки в панели свойств формы:





**Рисунок 28 – Назначение макроса кнопке**

После этого можно использовать кнопочную форму по назначению.

### 2.3. Оценка экономической эффективности проекта

Для расчета экономической эффективности БД, выбран метод сопоставления базового и проектного варианта, анализ затрат, которые необходимы для всех операций процесса.[4]

Экономическая эффективность программного продукта складывается из эффекта, который описывается через снижение трудовых показателей.

К показателям труда относят:

- абсолютное снижение затрат на труд ( $\Delta T$ ), что рассчитывается по формуле:

$$\Delta T = T_0 - T_1, \quad (1)$$

где  $T_0$  – это трудовые затраты по обработке данных базового варианта,  $T_1$  – трудовые затраты по проектному варианту.

Рассчитаем абсолютное снижение затрат труда.

– 7 часов в день – затраты труда по базовому варианту на обработку данных. Рассчитаем сколько именно часов на месяц составляют затраты по труду на обработку данных:

$$8 \text{ часов/день} * 24 \text{ раб. дня} = 192 \text{ часов/месяц}$$

Узнаем сколько именно составляют затраты труда в часах за год:

$$192 \text{ часов/месяц} * 12 \text{ месяцев} = 2304 \text{ часов/год.}$$

То есть,  $T_0 = 2304$  часов/год составляют затраты труда на обработку информации по базовому варианту.

– 3 часа в день составляют затраты труда для обработки данных по предлагаемому варианту. Трудовые затраты по рассматриваемому варианту на обработку данных равны:

$$3 \text{ часов/день} * 24 \text{ рабочих дня} = 72 \text{ часов/месяц.}$$

Узнаем сколько составляют затраты на труд в часах в год:

$$72 \text{ часов/месяц} * 12 \text{ месяцев} = 1728 \text{ часов/год.}$$

То есть,  $T_1 = 1728$  часов/год – это затраты труда на обработку информации по варианту.

Получим, что  $T_0 = 2304$  часов/год,  $T_1 = 1728$  часов /год, и можно вычислить абсолютное снижение затрат на труд:

$$\Delta T = 2304 - 1728 = 576 \text{ часов/год.}$$

• коэффициент относительного снижения для затрат труда рассчитывается по следующей формуле:

$$K_{ЗТ} = \frac{\Delta T}{T_0} \times 100\% \quad (2)$$

Получим:

$$K_{ЗТ} = 576 / 2304 * 100\% = 25\%.$$

То есть,  $K_{ЗТ} = 25\%$  – коэффициент относительного снижения для затрат труда.

• индекс повышения производительности труда (ППТ) вычисляется по формуле:

$$ППТ = \frac{T_0}{T_1} \quad (3)$$

В результате имеем:

$$ППТ = \frac{2304}{1728} = 1,33$$

Информационное обеспечение снабжает пользователей ИС данными, необходимыми для выполнения своих профессиональных обязанностей. ИС имеет возможность распределенного обработки и хранения информации, накопления данных в базах в местах использования, предоставления автоматизированного доступа к данным пользователям, одноразового ее ввода и многоцелевого, многократного использования.

Затраты на выполнение проекта состоят из затрат на заработную плату исполнителям, затрат на закупку или аренду оборудования, затрат на организацию рабочих мест, и затрат на накладные расходы (4).

$$K = C_{\text{ЗАРП}} + C_{\text{ОБ}} + C_{\text{ОРГ}} + C_{\text{НАКЛ}} \quad (4)$$

где  $C_{\text{ЗАРП}}$  - заработная плата исполнителей,  $C_{\text{ОБ}}$  -затраты на обеспечение необходимым оборудованием,  $C_{\text{ОРГ}}$  - затраты на организацию рабочих мест,  $C_{\text{НАКЛ}}$ - накладные расходы.

Стоит отметить, что при разработке проекта затраты на обеспечение необходимым оборудованием, затраты на организацию рабочих мест не были зафиксированы по причине их отсутствия.

Затраты на выплату исполнителям заработной платы определяется следующим соотношением:

$$C_{\text{ЗАРП}} = C_{\text{З.ОСН}} + C_{\text{З.ДОП}} + C_{\text{З.ОТЧ}} \quad (5)$$

где  $C_{\text{З.ОСН}}$  - основная заработанная плата,  $C_{\text{З.ДОП}}$  - дополнительная заработная плата,  $C_{\text{З.ОТЧ}}$  - отчисление с заработной платы.

Средняя зарплата системного администратора в 2016 г. составила приблизительно 60000 руб., дополнительная заработная плата в районе 20% от основной, отчисления с заработной платы в районе 26%.

Отчисления с заработной платы состоят в настоящее время в уплате единого социального налога. Согласно налоговому кодексу РФ применяются ставки налога для отчисления в фонд социального страхования, пенсионный фонд РФ, фонды обязательного медицинского страхования (федеральный и территориальный фонды).

Получим:

$$C_{\text{зарп}} = 60000 + 12000 - 18720 = 53280 \text{ руб.}$$

Затраты, связанные с обеспечением работ оборудованием, следует начать с определения состава оборудования и определения необходимости его закупки или аренды.

Расчет затрат, связанных с организацией рабочих мест для исполнителей проекта следует провести ориентируясь на требования СНИПа (санитарные нормы и правила) и на стоимость годичной аренды помещения требуемого уровня сервиса.

В обосновании затрат на размещение исполнителей следует сначала определить количество исполнителей, работающих на компьютерах и количество принтеров, а затем установить площадь для рабочего помещения.

Накладные расходы, связанные с выполнением проекта, следует вычислить, ориентируясь на расходы по основной заработной плате. Обычно они составляют от 60% до 100% расходов на основную заработанную плату. Используя соотношение, следует определить:

$$C_{\text{НАКЛ}} = 0.6 \cdot C_{\text{з.осн}} \quad (6)$$

Для рассматриваемого примера  $C_{\text{НАКЛ}}$  составит 36000 руб.

Результаты, полученные в процессе вычислений затрат следует подставить в соотношение (6), что и определит суммарные затраты на реализацию целей проекта.

$$K = 53280 + 36000 = 89580 \text{ руб.}$$

Для наглядности проиллюстрируем структуру затрат на выполнение проекта, используя круговую диаграмму (рис. 29).



**Рисунок 29 – Структура расходов**

В ряде случаев продажа ПО предполагает его настройку под условия эксплуатации, анализ условий эксплуатации, выдача рекомендаций для конкретного использования ПО и др. Вся совокупность затрат на эти мероприятия определяется как затраты на внедрение ПО.

В результате анализа были вычислены следующие показатели:

- общие затраты труда для разработки и внедрения программного обеспечения – 232 чел/час;
- количество необходимых исполнителей данного проекта – 1 чел.;
- абсолютное снижение затрат на труд – 576 часов/год.;
- коэффициент относительного снижения для затрат труда – 25%;
- индекс повышения производительности труда – 1,33;
- затраты на выполнение проекта – 89580 руб.

## Заключение

Основным достоинством БД является совместимость с популярным языком запросов SQL. Выполнив единственный запрос можно соединить несколько таблиц в одну временную и вырезать из нее нужные столбцы и строки.

Поскольку табличная структура реляционной БД интуитивно понятна пользователям, то и SQL-запросы являются простыми и легкими для изучения.

Реляционная модель имеет внушительный теоретический фундамент, с помощью которого была основана эволюция реляционных баз данных.

В процессе написания курсовой работы были реализованы следующие задачи:

- проведено анализ литературы с теории баз данных;
- описано предметную область;
- построено информационно-логическую модель предметной области;
- выполнено физическое проектирование базы данных в среде MS Access;
- разработано интерфейс пользователя.

В процессе анализа вышеизложенной информации выявлены следующие достоинства рассмотренной баз данных:

- удобность введения и просмотра информации таблиц базы данных пользователем;
- структурированное отображение данных в таблицах;
- применение различных видов связи между таблицами, что позитивно влияет на обработку данных;
- возможность создания на основе табличной модели данных различных вспомогательных элементов базы данных.

Но в данной модели хранения данных есть и несколько недостатков:

- невозможность хранения многомерных структур данных;

– плохая совместимость с сетевыми и иерархичными моделями данных.

Несмотря на имеющиеся недостатки, реляционная модель данных является наиболее распространенной и удобной.

## Список литературы

1. Мартынова В.П. Базы данных. Распределенные и удаленные БД. Т.1: Учебник/В.П. Мартынова.–М.:ИД ФОРУМ,НИЦ ИНФРА-М, – 2013. – 272 с.
2. Мартынова В.П. Базы данных. Распределенные и удаленные БД. Т.1 / В.П. Мартынова.– М.: ИД ФОРУМ,НИЦ ИНФРА-М,2013. – 352 с.
3. Ракован О.Л. Базы данных / О.Л. Ракован – М.:Форум, 2004. – 352 с.
4. Ракован О.Л.Базы данных: Учебное пособие/О.Л. Ракован. - М.:Форум, 2012.–400 с.
5. Малевич И.П. Базы данных:Учебное пособие /И.П. Малевич. - СПб.:Питер, 2013.– 240 с.
6. Кирилов В.В. Введение в реляционные базы данных./В.В. Кирилов.– СПб.: БХВ-Петербург, 2012.–464 с.
7. Кошечелев В.Е. Базы данных в ACCESS 2007: Эффективное использование /В.Е. Кошечелев.–М.: Бином-Пресс, 2014.–592 с.
8. Кузина А.В. Базы данных:Учебное пособие для студентов высш. учеб. заведений /А.В. Кузина.– М.: ИЦ Академия, 2012.–320 с.
9. Ливенар С.В. Материалы базы данных "Склад"/С.В. Ливенар.–М.: ИНФРА-М, 2015.–51 с.
10. Пирогова В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие/В.Ю. Пирогова.–СПб.: БХВ-Петербург, 2013.–528 с.