

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕНЕДЖМЕНТЕ

*Методические указания к курсовой работе
для студентов бакалавриата направления 38.03.02*

**САНКТ-ПЕТЕРБУРГ
2019**

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
Санкт-Петербургский горный университет

Кафедра информатики и компьютерных технологий

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕНЕДЖМЕНТЕ

*Методические указания к курсовой работе
для студентов бакалавриата направления 38.03.02*

САНКТ-ПЕТЕРБУРГ
2019

УДК 004.67(073)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕНЕДЖМЕНТЕ: Методические указания к курсовой работе / Санкт-Петербургский горный университет. Сост.: *В.В. Беляев, Т.Р. Косовцева, Л.Г. Муста*. СПб, 2019. 42 с.

Изложены теоретические сведения по формированию запросов из источников структурированных данных различных форматов средствами приложения *MSQuery* и обработке полученных данных в табличном процессоре *MSExcел*. Сформированы задания к курсовой работе, алгоритм выполнения и требования к оформлению.

Методические указания предназначены для студентов бакалавриата направления 38.03.02 «Менеджмент» по профилю «Экономика предприятия и организации».

Научный редактор доц. *А.Б. Маховиков*

Рецензент *К.В. Столяров* (Корпорация «Телум Инж»)

ВВЕДЕНИЕ

В соответствии с учебным планом курсовая работа выполняется на первом курсе осеннего семестра и является заключительным этапом в изучении дисциплины “Информационные технологии в менеджменте. Курсовая работа позволяет закрепить навыки работы с табличным процессором MS Excel.

Каждый студент выполняет индивидуальное задание, которое ориентировано на работу с заранее подготовленными и хранящимися на компьютере базами данных. Для выполнения задания требуются уверенные практические навыки работы на персональном компьютере, понимание основ построения баз данных, умение разрабатывать алгоритм решения задачи и ее постановку.

Большое внимание уделяется оформлению отчета по курсовой работе, который полностью выполняется на персональном компьютере и демонстрирует практические навыки работы студента с современным программным обеспечением.

Задание выдается кафедрой на третьей неделе семестра, а оформленная работа представляется на проверку до 1 декабря. После этого назначается срок защиты. В процессе выполнения курсовой работы каждый студент должен ежемесячно отчитываться о проделанной работе и предоставлять преподавателю промежуточные материалы.

ЗАДАНИЕ К КУРСОВОЙ РАБОТЕ ПО ИНФОРМАТИКЕ

Цель работы - овладеть навыками импорта данных (*MSQuery*) и обработки полученных таблично организованных данных средствами сводной таблицы (*Pivot*) в табличном процессоре MS Excel.

Индивидуальные задания для выполнения работы и заранее подготовленные файлы баз данных выдаются преподавателем.

Описание структуры баз данных приведено ниже.

В соответствии с индивидуальным заданием с помощью *MS Query* следует сформировать запрос на выборку, предварительно объединив эти базы данных. Результат выборки направить в

MS Excel, где обработать переданную информацию средствами сводной таблицы по заданию.

Задача 1

На компьютере хранится информация о работе перегрузочных складов нерудных строительных материалов в базе данных STOCKS.DBF (табл.1) и свойствах щебней в базе данных PROPERTY.DBF формата dBASE (табл.2).

Таблица 1

Структура базы STOCKS.DBF

Имя поля	Содержание	Тип поля
DET	Шифр щебня	текст
DETRITUS	Горная порода	текст
TRANS	Перегрузочный склад	текст
DATE	Дата	дата
PRICE	Цена, руб./т	число
GROUP	Фракция, мм	текст
RETURN	Оборот, тыс.т.	число

Таблица 2

Структура базы PROPERTY.DBF

Имя поля	Содержание	Тип поля
DET	Шифр щебня	текст
GROUP	Фракция, мм	текст
CRUSH	Марка по дробимости	число
FROST	Марка по морозостойкости	текст
BREAM	Лещадность, не выше %	число
RADIO	Радиоактивность, Бк/кг	число

Задача 2

На компьютере хранится информация об основных фондах горного предприятия в базах данных OF10.DBF, OF12.DBF, OF13.DBF и OF14.DBF по группам подразделений. В базах данных

OF10_M.DBF, OF12_M.DBF, OF13_M.DBF, OF14.DBF и OF14_M.DBF по соответствующим группам подразделений имеется информация об ответственных хранителях материальных ценностей. Все базы данных созданы в формате dBASE.

Таблица 3

Структура базы OF .DBF

Имя поля	Содержание	Тип поля
СЕН	Номер подразделения	число
DDAT	Дата постановки ОС на учет	дата
INV	Инвентарный номер	число
NAIM	Наименование	текст
SUM	Первоначальная стоимость, тыс.руб.	число
IZN	Износ, тыс.руб.	число
NOR	Норма амортизации, %	число

Таблица 4

Структура базы OF M.DBF

Имя поля	Содержание	Тип поля
СЕН	Номер подразделения	число
NAME	Наименование подразделения	текст
PERSON	Фамилия, имя, отчество материально ответственного лица	текст

Задача 3

В базе данных STORE.DBF формата dBASE хранится информация о оборотах на складе готовой продукции предприятия по производству фарфора. База данных DESIGN.DBF, также формата dBASE содержит информацию о наименовании изделий и их авторах.

Таблица 5

Структура базы STORE.DBF

Имя поля	Содержание	Тип поля
NSKL	Номер склада	число
OBOZN	Шифр изделия	текст

Продолжение табл.5

Имя поля	Содержание	Тип поля
NAIM	Наименование изделия	Текст
KOLVO	Количество	число
DT	Дата поступления на склад	дата
EDIZM	Единица измерения	текст
PRICE	Цена, руб.	число
NDS	Налог на добавленную стоимость, %	число
SORT	Сорт	число
NUMITEM	Количество предметов в изделии	число

Таблица 6

Структура базы DESIGN.DBF

Имя поля	Содержание	Тип поля
NAIM	Наименование изделия	текст
DESIGNER	Фамилия дизайнера	текст

АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

Для импорта данных из внешних источников следует:

- 1) Открыть MS Excel. Установить курсор в левый верхний угол листа.
- 2) Выполнить команду **Данные Δ Импорт внешних данных Δ Создать запрос**.
- 3) Извлечь нужную информацию с помощью приложения **MS Query**. Последовательность операций:
 - определить источник данных (мастер запросов не использовать);
 - указать путь к файлам с базами данных;
 - добавить необходимые таблицы;
 - установить связи между таблицами;
 - выбрать тип связи между таблицами;
 - сформировать условия (критерии) выборки;

- выбрать нужные поля из обеих таблиц;
 - сохранить запрос (текст запроса) на языке SQL в виде рисунка (скриншота) окна «Запрос SQL»;
 - скопировать текст запроса на языке SQL и вставить в отчет в виде текста;
 - вернуть данные в MS Excel.
- 4) Создать сводную таблицу в следующем порядке:
- скопировать полученные из Query данные на новый лист книги и работать с ним;
 - установить необходимые форматы;
 - определить русские названия колонок в таблице;
 - добавить вычисляемые поля, если это необходимо;
 - сформулировать название таблицы;
 - создать таблицу в соответствии с заданием.

**ПОСЛЕДОВАТЕЛЬНОСТЬ ИЗЛОЖЕНИЯ МАТЕРИАЛА В ПОЯСНИТЕЛЬНОЙ
ЗАПИСКЕ ПО КУРСОВОЙ РАБОТЕ**

- Аннотация
- Оглавление
- Введение (цель и средства).
- Вариант задания (определить все понятия и термины задачи).
- Постановка задачи.
- Алгоритм решения задачи (работа с *MS Query* и сводной таблицей *MS Excel*).
- Результаты (текст запросов, промежуточные таблицы, сводные таблицы).
- Библиографический список.

ТРЕБОВАНИЯ ПО ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Пояснительная записка по курсовой работе должна быть оформлена в соответствии с действующими стандартами [1].

Объем отчета не должен превышать 20 страниц.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

БАЗЫ ДАННЫХ

Базы данных представляют собой информационные модели, содержащие данные об объектах и их свойствах. Базы данных хранят информацию о группах объектов с одинаковым набором свойств.

Например, база данных « Библиотечный каталог» хранит информацию о книгах, каждая из которых имеет название, автора, год издания и так далее.

Информация в базах данных хранится в упорядоченном виде. Так, в библиотечном каталоге - либо по алфавиту – алфавитный каталог), либо по области знания (предметный каталог).

База данных (БД) - это информационная модель, позволяющая в упорядоченном виде хранить данные о группе объектов, обладающих одинаковым набором свойств.

Существует несколько различных структур информационных моделей и соответственно различных типов баз данных: табличная, сетевая, иерархическая.

ИЕРАРХИЧЕСКИЕ БАЗЫ ДАННЫХ

Иерархическая модель данных — это модель данных, где используется представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней. Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня.

Иерархические базы данных графически могут быть представлены как перевернутое дерево, состоящее из объектов различных уровней. Верхний уровень (корень дерева) занимает один объект, второй — объекты второго уровня и так далее.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект, более близкий к корню) к потомку(объект более низкого уровня), при этом объект-предок может не иметь потомков или иметь их несколько, тогда как

объект-потомок обязательно имеет только одного предка. Объекты, имеющие общего предка, называются близнецами.

Иерархической базой данных является файловая система, состоящая из корневого каталога, в котором имеется иерархия подкаталогов и файлов.

Например: иерархической базой данных является *Каталог папок Windows*, с которым можно работать, запустив приложение *Проводник*. Верхний уровень занимает папка *Рабочий стол*. На втором уровне находятся папки *Мой компьютер*, *Мои документы*, *Сетевое окружение* и *Корзина*, которые являются потомками папки *Рабочий стол*, а между собой является близнецами. В свою очередь, папка *Мой компьютер* является предком по отношению к папкам третьего уровня - папкам дисков (например, DVD RW дисковод (E:), (C:), (D:), (F:)) и системным папкам (*Принтеры*, *Панель управления* и др.) Иерархия папок ОС MS Windows как пример иерархической базы данных приведена на рисунке 1.

Иерархическая база данных - это хранилище, применимое для тех систем, которым изначально свойственна древовидная структура. Для них выбирать подобное моделирование - логично. Примерами иерархической базы данных с изначально систематизированными степенями могут быть

- воинские подразделения, в которых, четко определены ранги;
- учебные заведения с явно выраженной структурой подчинения, как это показано на рисунке 1.

Для моделирования таких систем и приведения их к виду рассматриваемой БД нет необходимости в *декомпозиции*. Тем не менее такая ситуация складывается не всегда. Кроме того, существует тенденция, при которой направленный вниз по структуре запрос проще, чем аналогичный вверх.

Иерархическая модель базы данных основана на структуре, имеющей сходство с перевернутым деревом, где от ствола отходят ветви, от которых в свою очередь отходят другие ветви. В такой структуре просто проследить компоненты СУБД, и существующие между ними связи.

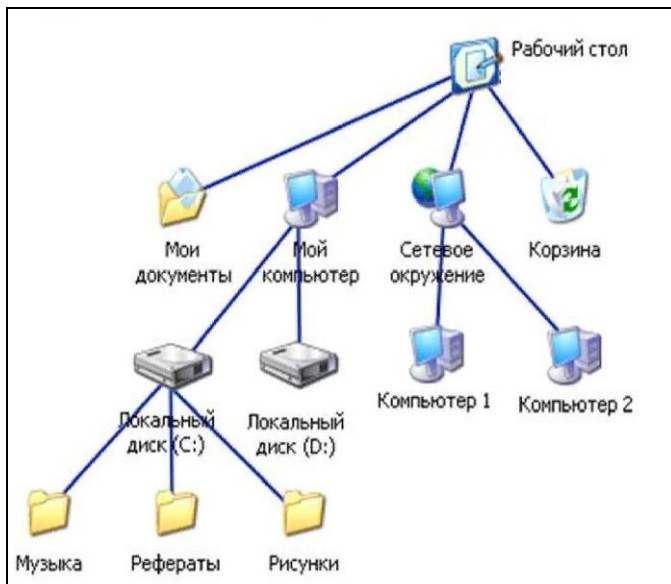


Рисунок - 1. Иерархия папок ОС MS Windows

Недостатки:

- сложность реализации и управления;
- сложность программирования и использования приложений;
- малая стандартизация.

Преимущества:

- простота идеи;
- безопасность;
- независимость и целостность данных
- эффективность.

Иерархическая модель данных строится по принципу иерархии типов объектов, то есть один тип объекта является главным, а остальные находятся на низших уровнях иерархии - подчиненными.

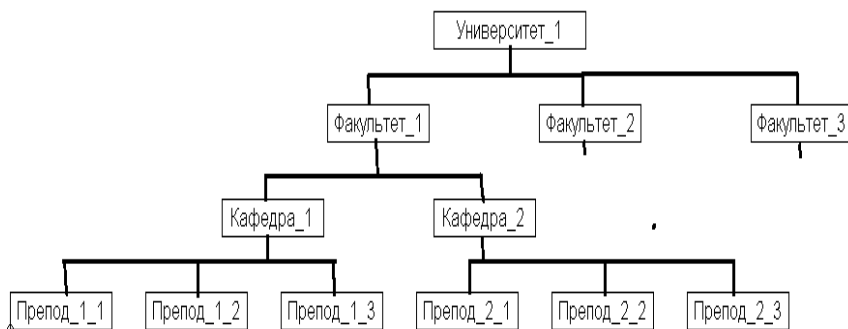


Рисунок - 2. Структура учебного заведения. Пример иерархической базы данных

СЕТЕВЫЕ БАЗЫ ДАННЫХ

Сетевая модель – это структура, у которой любой элемент может быть связан с любым другим элементом.

Сетевая база данных является обобщением иерархической за счет допущения объектов, имеющих более одного предка. Сетевая база данных состоит из наборов записей, которые связаны между собой. При этом записи могут содержать ссылки на другие наборы записей, образуя таким образом сеть. На связи между объектами в сетевых моделях не накладывается никаких ограничений. Пример сетевой базы данных приводится на рисунке 3.

Сетевой базой данных фактически является *Всемирная паутина* глобальной компьютерной сети Интернет. Гиперссылки связывают между собой сотни миллионов документов в единую распределенную сетевую базу данных.

Основное достоинство сетевой модели – это высокая эффективность по затратам памяти и оперативность доступа к данным. Недостаток – сложность и жесткость схемы базы, а также сложность понимания. Кроме того, в этой модели ослаблен контроль целостности, так как в ней допускается устанавливать произвольные связи между записями

В целом иерархические и сетевые модели обеспечивают достаточно быстрый доступ к данным. Поскольку в сетевых базах основная структура представления информации имеет форму сети, в

которой каждая вершина (узел) может иметь связь с любой другой, то данные в сетевой базе более равноправны, чем в иерархической, так как доступ к информации может быть осуществлен, начиная с любого узла.

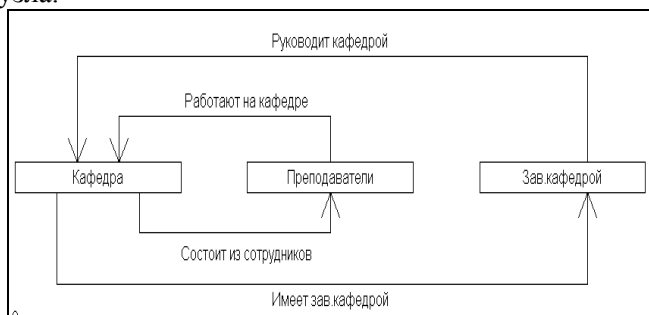


Рисунок - 3. Пример сетевой базы данных

Следует отметить, что жесткость организации данных в иерархических и сетевых моделях определяет особенности работы с такими базами. Доступ к информации осуществляется только в соответствии со связями, определенными при проектировании структуры конкретной базы данных. Базы данных с такими моделями сложно реорганизовывать. Недостатком этих моделей является и сложность механизма доступа к данным, а также необходимость на физическом уровне четко определять связи данных. А поскольку каждый элемент данных должен содержать ссылки на некоторые другие элементы, то для этого требуются значительные ресурсы памяти ЭВМ. Кроме того, для таких моделей характерна сложность реализации систем управления базами данных.

В сетевой модели данных понятие главного и подчиненного объектов несколько расширены. Любой объект может быть главным и подчиненным. Один и тот же объект может выступать и в роли "владельца" и в роли "подчиненного".

Графовые (иерархические и сетевые) модели реализованы в качестве моделей данных в системах управления базами данных, работающих на больших ЭВМ. Для персональных компьютеров больше распространены реляционные базы данных, хотя имеются и

системы управления базами данных, поддерживающих сетевую модель.

РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

Реляционная модель данных – это структура, состоящая из совокупности взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа.

Таблица содержит перечень объектов одного типа, то есть объектов, имеющих одинаковый набор свойств. Основным объектом такой базы данных является двумерная таблица, которую удобно представлять в виде двумерной таблицы состоящей из совокупности строк и столбцов. В каждой ее строке последовательно размещаются значения свойств одного из объектов; каждое значение свойства — в своем столбце, озаглавленном именем свойства.

Рассмотрим, например, базу данных **Картотека игрушек**, состоящую из одной таблицы (рис.4):

Картотека игрушек

Артикул	Наименование	Материал	Цвет	Цена
A01	Кубики	Дерево	Красный	5
A02	Пирамида	Пластмасса	Синий	10
A03	Мячик	Резина	Красный	6

Рисунок 4 – Пример базы данных

Поле базы данных - это столбец таблицы, содержащий значения определенного свойства каждое поле характеризуется своим именем (именем соответствующего свойства) и типом данных, представляющих значения данного свойства.

Запись базы данных - это строка таблицы, содержащая набор значений свойств, размещенный в полях базы данных.

Строки таблицы являются записями об объекте; эти записи разбиты на поля столбцами таблицы, поэтому каждая запись представляет собой набор значений, содержащихся в полях.

Записи, т. е. строки таблицы, имеют одинаковую структуру — они состоят из полей, хранящих атрибуты объекта. Каждое поле, т. е. столбец, описывает только одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те

же поля, только в них отображаются различные информационные свойства объекта.

Ключевое поле - это поле, значение которого однозначно, определяет запись в таблице.

При реализации реляционной модели в качестве ключевого поля часто используют поле, имеющее тип данных **счетчик**. Однако иногда удобнее в качестве ключевого поля таблицы использовать другие поля: *код товара, инвентарный номер* и т. п. Обозначается первичный ключ, как primary key (**Primary key - РК**), очень важен для каждой таблицы поскольку:

- РК не позволяет создавать одинаковых записей (строк) в таблице;
- РК обеспечивают логическую связь между таблицами одной базы данных (для реляционных БД).

Принципы организации логической связи между таблицами описаны ниже.

Картотека игрушек

<i>Имена полей</i>	Артикул	Наименование	Материал	Цвет	Цена
<i>Запись</i>	A01	Кубики	Дерево	Красный	5
<i>Запись</i>	A02	Пирамида	Пластмасса	Синий	10
<i>Запись</i>	A03	Мячик	Резина	Красный	6
	<i>Ключевое поле</i>	<i>Поле</i>	<i>Поле</i>	<i>Поле</i>	<i>Поле</i>

Тип поля определяется типом данных, которые оно содержит. Поля могут содержать данные следующих основных типов:

- *счетчик* — целые числа, которые задаются автоматически при вводе записей. Эти числа не могут быть изменены пользователем;
- *текстовый* — тексты, содержащие до 255 символов;
- *числовой* — числа;
- *дата/время* — дата или время;

- *денежный* — числа в денежном формате;
- *логический* — значения *Истина* (Да) или *Ложь* (Нет);
- *поле объекта OLE* - изображение или рисунок;
- *гиперссылка* — ссылки на информационный ресурс в Интернете (например, Web-сайт).

Поле каждого типа имеет свой набор свойств. **Наиболее важными свойствами полей** являются:

- *размер поля* - определяет максимальную длину текстового или числового поля;
- *формат поля* - устанавливает формат данных;
- *обязательное поле* - указывает на то, что данное поле обязательно надо заполнить

Каждая таблица должна содержать, по крайней мере, одно ключевое поле, содержимое которого уникально для каждой записи в этой таблице. Ключевое поле позволяет однозначно определять (идентифицировать) каждую запись в таблице. По значению ключа можно отыскать единственную запись. Ключи служат также для упорядочивания информации в БД.

Связи между таблицами бывают следующих типов:

- **Один к одному (One to one);**
- **Один к многим (One to many);**
- **Многие ко многим (Many to many).**

Один к одному

Это тип связи, когда одной строке таблицы соответствует не более одной строки в другой таблице (Гражданин – Номер паспорта). Такой вид связи используется редко, поскольку, в этом случае, таблицы можно объединить в одну.

Связь один ко многим

Это наиболее часто используемый тип связей. В этом типе связей несколько строк из дочерней таблицы зависят от одной

строки в родительской таблице. Например, несколько игрушек могут иметь один и тот же цвет. В этом случае таблица «Цвета» является родительской, а таблица с игрушками «Картотека» - дочерней. В то же время (в нашей модели), одна игрушка одновременно может иметь только один цвет. То есть один цвет - много игрушек

Многие ко многим

Это тип связи, когда одна строка первой таблицы связана с несколькими строками второй таблицы, и одна строка второй таблицы связана с несколькими строками первой (предмет - преподаватель). В реляционной модели данных этот тип связей напрямую реализован быть не может. Он реализуется путём введения промежуточной таблицы.

Таблицы реляционной БД должны отвечать требованиям нормализации отношений.

Нормализация отношений — это формальный аппарат ограничений на формирование таблиц, который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение базы данных.

Проведем анализ таблицы **Картотека игрушек**, содержащую следующие поля: **Артикул**, **Наименование**, **Материал**, **Цвет**, **Цена**. Такая организация хранения информации будет иметь ряд недостатков:

- дублирование информации (материал и цвет), что приводит к увеличению объема БД;
- процедура обновления информации в таблице затрудняется из-за необходимости редактирования каждой записи таблицы.

Нормализация таблиц предназначена для устранения этих недостатков.

Целью нормализации является устранение недостатков структуры базы данных, приводящих к вредной избыточности в данных, которая в свою очередь потенциально приводит к различным аномалиям и нарушениям целостности данных.

Для устранения указанных недостатков таблицу **Картотека игрушек** целесообразно реорганизовать следующим способом:

1. Создать две дополнительные таблицы, содержащие информацию
 - а) о материале (таблица Материалы);
 - б) о цвете (таблица Цвета).
2. В таблице **Картотека** изменить тип полей
 - а) Материал (для кода материала из таблицы Материалы);
 - б) Цвет (для кода цвета из таблицы Цвета).
3. Создать связи между всеми тремя таблицами.

При реализации указанных изменений в СУБД Access¹ таблицы примут вид, приведенный на рисунке 5.

Таблицы в этом случае должны быть связаны между собой. Наличие связей между таблицами определяет термин «реляционные база данных», поскольку слово «relation» и обозначает термин «связь», «отношение», «зависимость». Наличие связей между таблицами графически может быть изображено, например в MS Access в виде «Схемы данных», как это показано на рисунке 6.

Для установления логической связи между таблицами используют внешние ключи.

Для обеспечения однозначной логической связи, между таблицами одной БД используется *внешний ключ* (Foreign key, кратко FK).

Внешний ключ представляет один или несколько столбцов из одной таблицы, который одновременно является потенциальным ключом из другой таблицы.

Как правило, внешний ключ из зависимой таблицы указывает на первичный ключ из главной таблицы.

¹ Реализация в СУБД Access приведена здесь и ниже исключительно в целях визуального представления принципов построения реляционных баз данных

The image shows three tables from an MS Access database. The top table, 'Картотека', contains 7 records with columns: Артикул, Наименование, Материал, Цвет, and Цена. Below it are two smaller tables: 'Материалы' and 'Цвета', each with 3 records and columns: КодМатериала/КодЦвета and Наименование.

Артикул	Наименование	Материал	Цвет	Цена
A01	Кубики	1	1	5,00 Р
A02	Пирамида	1	2	10,00 Р
A03	Мячик	2	1	6,00 Р
A04	Мячик большой	2	2	7,00 Р
A05	Кукла Маша	3	3	18,00 Р
A06	Кукла Даша	3	3	19,00 Р
		0	0	0,00 Р

КодМатериала	Наименование
1	Дерево
2	Резина
3	Пластмасса
0	

КодЦвета	Наименование
1	Красный
2	Синий
3	Телесный
0	

Рисунок - 5. Пример реализации реляционной базы данных в MS Access

Например, поле «Материал» является внешним ключом (FK) в таблице «Картотека», и содержит значение поля «КодМатериала», которое является ключевым полем (PK) таблицы «Материалы».

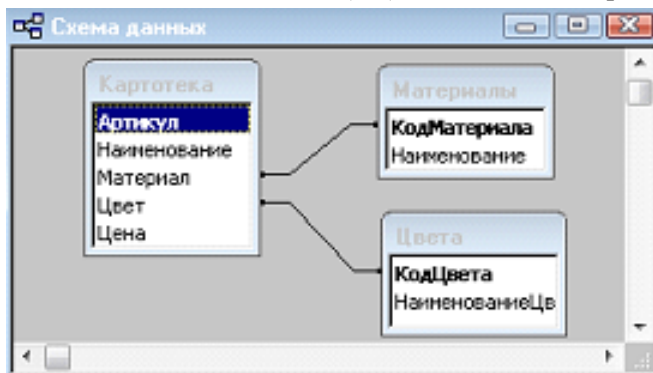


Рисунок - 6. Схемы данных в MS Access

ИМПОРТ ДАННЫХ С ПОМОЩЬЮ MS-QUERY

ОСНОВНЫЕ ПОНЯТИЯ

MS Query - приложение, которое просматривает, отбирает и организует данные из базы данных. Это не база данных, а только инструмент для просмотра и отбора определённых данных. Полученные данные можно копировать в *MS Excel* и приложения *Windows*.

С помощью приложения *MS Query* можно производить следующие операции:

- обрабатывать данные различных форматов (*dBase* или *Paradox*, *Microsoft Access*, *FoxPro*), а также любых других форматов, доступ к которым возможен через интерфейс *ODBC* (*Open Data Base Connectivity*), и передавать результаты в *MS Excel*;
- указывать записи или отдельные поля записей, для которых будет выполнен тот или иной запрос *QBE* (*Query by Example*), а также определять критерии выбора данных;
- использовать для обработки запросов специальный структурированный язык запросов *SQL* (*Structured Query Language*).

Применение всех команд будем иллюстрировать примерами, в которых данные берутся из совокупности таблиц, описывающих продажу фруктов некоторым торговым предприятием.

СТРУКТУРА БАЗЫ ДАННЫХ

В работе исходными данными являются файлы типа *dbf*. Одним из вариантов реализации реляционной модели являются информационные системы, использующие такой тип данных. Файлы типа *dbf* исторически называются файлами базы данных. *DBF* (*Data Base File*) — формат хранения данных, используемый в качестве одного из стандартных способов хранения информации в системах управления базами данных.

Не всякая программа, в том числе и *MS Excel*, может правильно прочитать и корректно записать произвольный файл *DBF*. Для результативной и успешной работы с файлами *DBF*

необходимо иметь представление об их структуре. Упрощенно можно полагать, что такой файл имеет структуру, похожую на таблицу. Можно считать, что в файле есть столбцы, которые называют полями, и строки, которые называются записями. Структура файла DBF определяется именами полей, их типом и длиной. Для иллюстрации приемов работы с файлами DBF рассмотрим информационную систему (ИС) «Фруктовая база», состоящую из четырех файлов. Система предназначена для автоматизации учета процесса продажи фруктов.

Каждой продаже (заказу) соответствует документ - накладная, которую можно представить себе в виде шапки (заголовка) и перечня позиций (деталей заказа).

Информация о продаже фруктов хранится в файле INVOICE.DBF, содержащим заголовки заказов, и в файле ITEMS.DBF, содержащим детали заказа. При этом наименование фруктов, их свойства хранятся в файле, который имеет имя Фрукты.dbf.

Файл с именем Категор.dbf содержит сведения о категориях, к которым можно отнести фрукты.

Структура файлов приведена в таблицах 1-4. Фрагменты файлов иллюстрируют их содержание (рис.7 - рис.10).

Таблица 1

Структура базы «INVOICE.dbf»

Имя поля	Содержание	Тип поля
Код заказа	Ключевое поле-Уникальный номер	Числовой
Код покупателя		Числовой
Дата заказа		Дата
Подарок		Логический
Фамилия		Текстовый
Имя		Текстовый
Адрес		Текстовый
Город		Текстовый
Индекс		Числовой
Страна		Текстовый

Таблица 2

Структура базы «ITEMS.dbf»

Имя поля	Содержание	Тип поля
Код Описания	Ключевое поле-Уникальный номер	Числовой
Код заказа	Внешний ключ. Для связи с INVOICE.dbf	Числовой
Количество		Числовой
Код фрукта	Внешний ключ. Для связи с Фрукты.dbf	Числовой

Таблица 3

Структура базы «Фрукты.dbf»

Имя поля	Содержание	Тип поля
Код фрукта	Ключевое поле - Уникальный номер	Числовой
Название фрукта		Текстовый
Описание		Текстовый
Категория	Внешний ключ. Для связи с Категор.dbf	
Цена		Денежный

Таблица 4

Структура базы «Категор.dbf»

Имя поля	Содержание	Тип поля
Код категории	Ключевое поле-Уникальный номер	Числовой
Категория	Наименование категории	Текстовый

	A	B	C	D	E
1	Код фрукта	Название фрукта	Описание фрукта	Категория	Цена
2	1	абрикос	Канадские абрикосы из Харглоу	1	2
3	2	вишни	Свежие вишни из Северной Калифорнии	1	3
4	3	персик	Свежие персики с запада США	1	2

Рисунок 7 - Фрагмент файла Фрукты.dbf

	A	B
1	КОД_КАТЕГО ,N,19,5	КАТЕГОРИЯ, C,50
2		1 <u>Свежие</u>
3		2 <u>Сушеные</u>
4		3 <u>Засахаренные</u>
5		4 В шоколаде

Рисунок 8 - Файл Категор.dbf

	A	B	C	E	F	G	H	I	J
1	КОД_З АКАЗА ,N,19,5	КОД_П ОКУПА ,N,19,5	ДАТА_З АКАЗ,D	ФАМИЛИЯ, C,254	ИМЯ,С,25 4	АДРЕС,С,254	ГОРОД,С,254	ИНДЕКС,С, 254	СТРАНА,С,2 54
2	1	23	28.10.16	Иванов	Михаил	Солнечная 93-3	Выборг	83554	Россия
3	2	33	28.10.16	Минтон	Светлана	Павловская 39-58	Выборг	02151-	Россия
4	3	158	28.10.16	Хрустов	Никита	Агрономов 92-2	Воронеж	58103-	Россия
5	4	332	29.10.16	Мозгорово	Павел	Шкиперский пер., 4-90	Жодино		Белоруссия
6	5	21	29.10.16	Мионов	Александр	Мельничная, 23-1	Краснофлотск	98303	Россия
7	6	1	29.10.16	Собачин	Альберт	Сабельная 87-12 Suite 5	Кострома	941171234	Россия
8	7	275	30.10.16	Лотов	Андрей	Беринга 65-25	Гдов	99508-	Россия
9	8	231	30.10.16	Полес	Мария	пр. Шекспира, 2-1	Ярега	92354-	Россия
10	9	348	30.10.16	Сееледкина	Елена	Солонная 2-90	Потопталово		Белоруссия

Рисунок 9 - Фрагмент файла INVOICE.dbf

	A	B	C	D
1	КОД_ОПИСАН ,N,19,5	КОД_ЗАКАЗА ,N,19,5	КОЛИЧЕСТВО, N,19,5	КОД_ФРУКТА, N,19,5
2		1	1	10
3		2	34	10
4		3	32	10
5		4	3	3
6		5	29	3
7		6	12	2
8		7	1	10

Рисунок 10 - Фрагмент файла ITEMS.dbf

С учетом описания типа полей, приходим к выводу что все связи между таблицами в ИС имеют тип «один ко многим». ИС будет иметь схему, приведенную на рис.11.

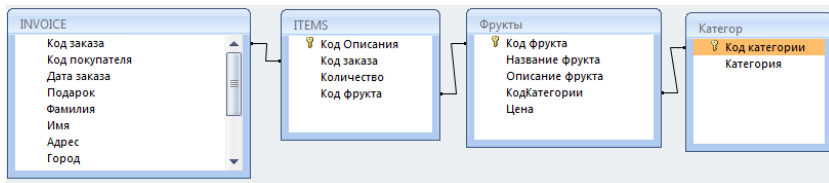


Рисунок 11 - Схема данных

ФОРМИРОВАНИЕ ЗАПРОСА

Пример 1

Выбрать все записи из таблицы «Фрукты». Результат выборки направить в MS Excel.

Вызвать MS Query можно с помощью команды *Получение внешних данных* - из других источников - из *Microsoft Query* на ленте в пункте меню *Данные* (рис. 12).

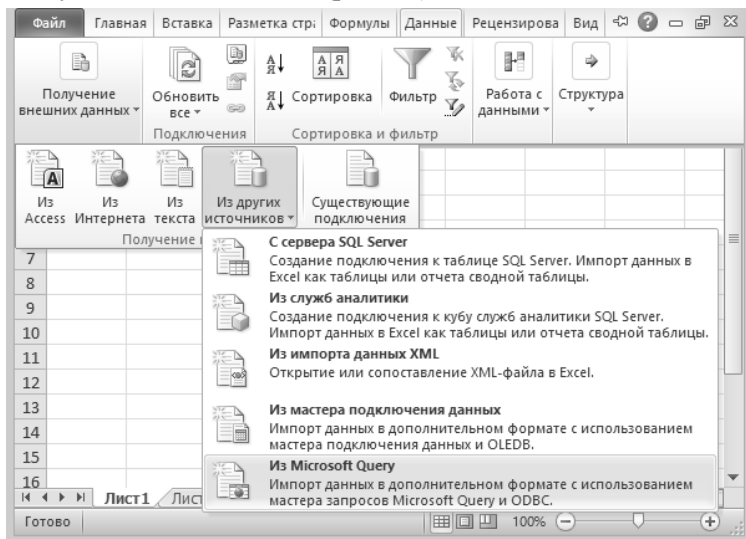


Рисунок 12 - Получение внешних данных в MS Excel

Откроется диалоговое окно «Выбор источника данных», в котором необходимо определить источник получения данных с учетом типа данных и места его расположения (рис.13).

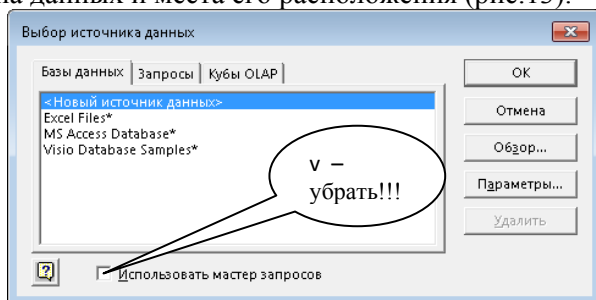


Рисунок 13 - Диалоговое окно *Выбор источника данных*

Создадим новый источник данных, при этом флажок «Использовать мастер запросов» следует убрать.

Создание нового запроса:

1. Дадим запросу имя, например, «myConnect-2019»;
2. Выберем драйвер для требуемого типа базы данных, в нашем случае это «Microsoft dBase Driver (*.dbf)»;
3. Нажмем кнопку «Связь» (рис.14).

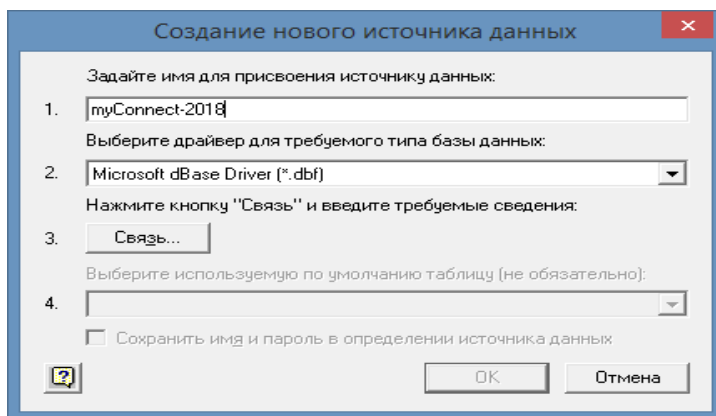


Рисунок 14 – Создание источника данных

В появившемся окне убираем отметку «Использовать текущий каталог» (рис.15).

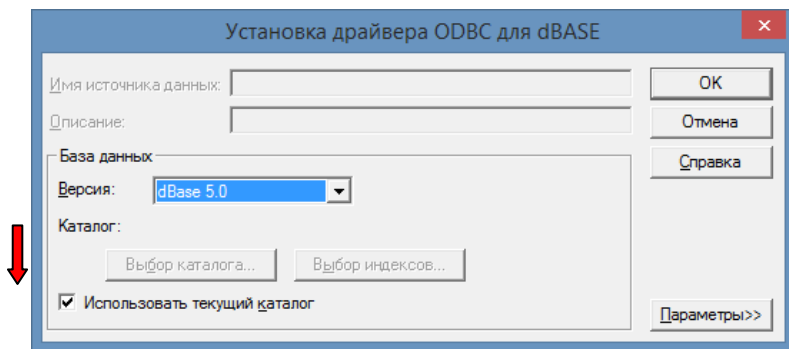


Рисунок 15 - Работа с диалоговым окном *Установка драйвера* (начало)

После этого кнопка «Выбор каталога» будет активна (рис.16).

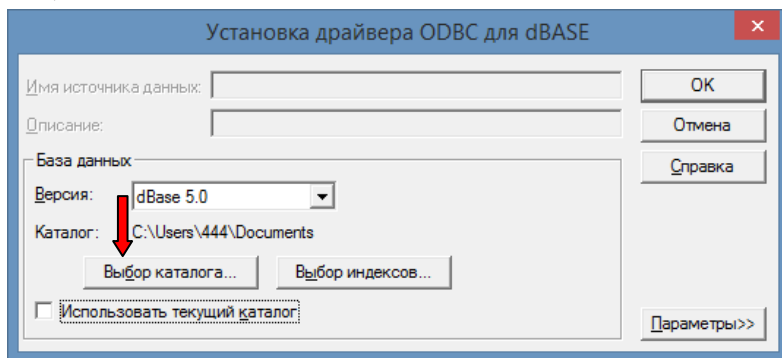


Рисунок 16 - Работа с диалоговым окном *Установка драйвера* (продолжение)

Выберем каталог, в котором лежат заданные таблицы (в нашем случае это «Bases\Fruits»), при этом в listbox-е будут видны искомые файлы Фрукты.dbf и Категор.dbf (рис.17).

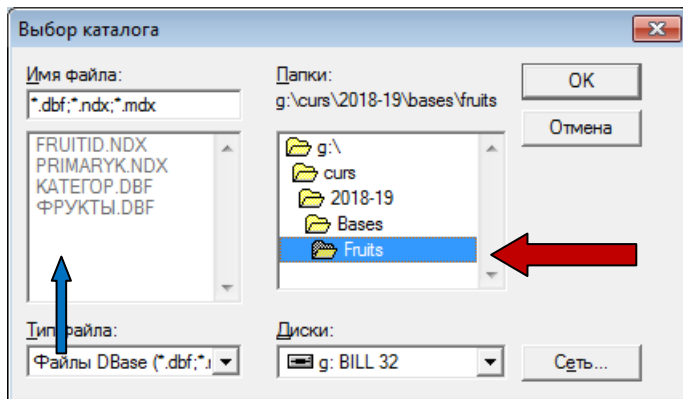


Рисунок 17- Выбор каталога

Нажимаем кнопку «ОК» пока не появится диалоговое окно «Создание нового источника данных», в которой выберем искомую таблицу «ФРУКТЫ» (рис.18).

Создание источника данных завершаем нажатием кнопки «ОК»

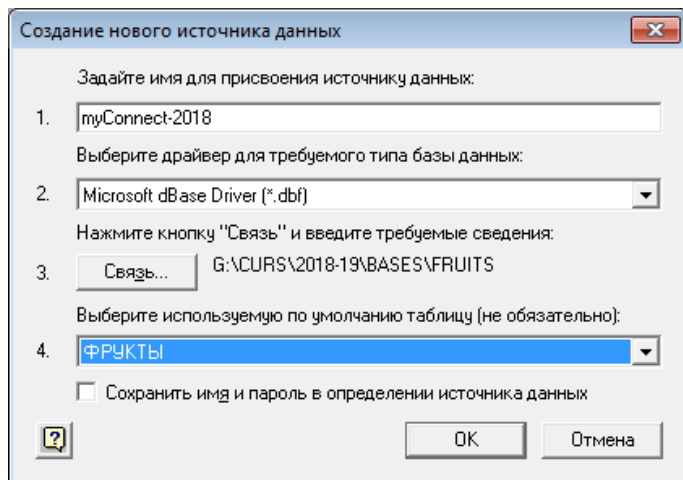


Рисунок 18 – Выбор таблицы *Фрукты*

Запрос с именем «myConnect-2018» будет отображаться в окне «Выбор источника данных» (рис.19).

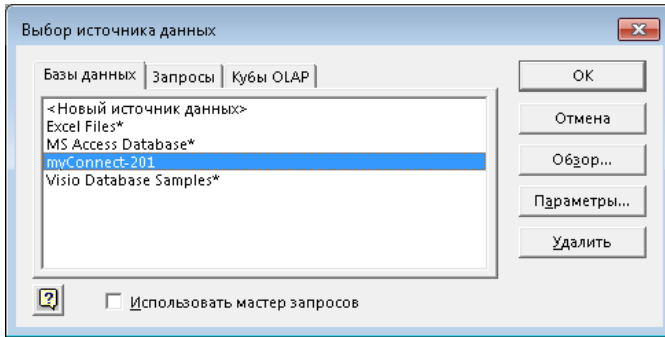


Рисунок 19 - Запрос с именем «myConnect-2018» в списке

Активизируем созданный источник данных «myConnect-2018».

Для этого выберем его и нажмем кнопку (OK). Это вызовет появление окна *MS Query*, которое показано на рисунке 20. Оно разделено на две части. В верхней части расположено окно с именами полей, которые есть в исходной базе данных, а в нижней будут размещены поля, которые участвуют в определении запроса.

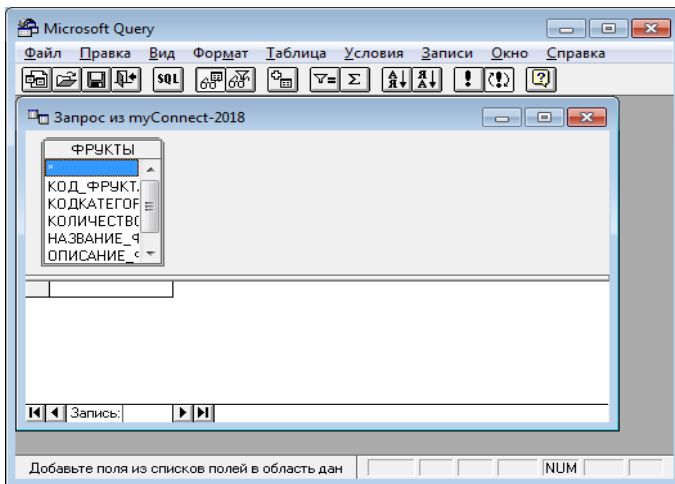


Рисунок 20 - Окно *MS Query*

Определить эти поля можно одним из следующих способов:

- поставить указатель "мыши" на имя поля, нажать левую клавишу "мыши" и, не отпуская ее, протащить в нижнюю часть окна;
- дважды щелкнуть "мышью" по имени поля;
- открыть список в нижней части окна, щелкнув "мышью" на верхней строке пустого столбца и выбрав из списка нужное имя;
- выполнить команду **Записи > Добавить столбец**.

Если в запросе нужны все поля исходной базы, необходимо щелкнуть в качестве имени поля звездочку (*), которая находится в первой строке списка полей. В данном примере необходимо получить все поля в результате работы запроса, поэтому нажимаем символ «*» (рис.21).

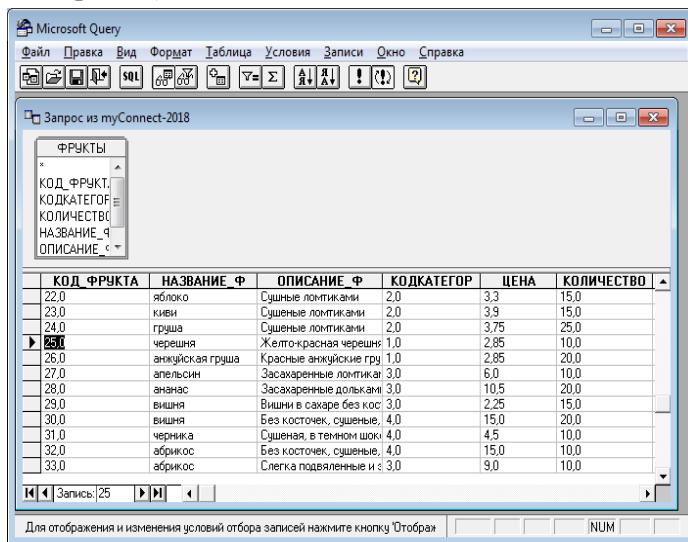


Рисунок 21 - Окно запроса с одной таблицей "Фрукты".

Любому запросу, сформированному с помощью MS Query, соответствует команда (инструкция), написанная на языке SQL (рис. 22). Чтобы увидеть это выражение необходимо активировать окно «Запрос SQL», нажав кнопку «SQL».

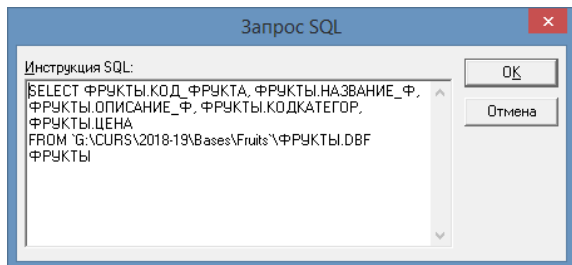


Рисунок 22 - Окно «Запрос SQL» с текстом запроса для решения примера 1

Подробное описание языка SQL приведено ниже. Выбранные данные можно вернуть в MS Excel.

ВОЗВРАТ В EXCEL

Для того, чтобы передать данные в MS Excel, следует выполнить команду *Δ Файл Δ Вернуть данные* или нажать соответствующую кнопку на панели инструментов (рис.23).

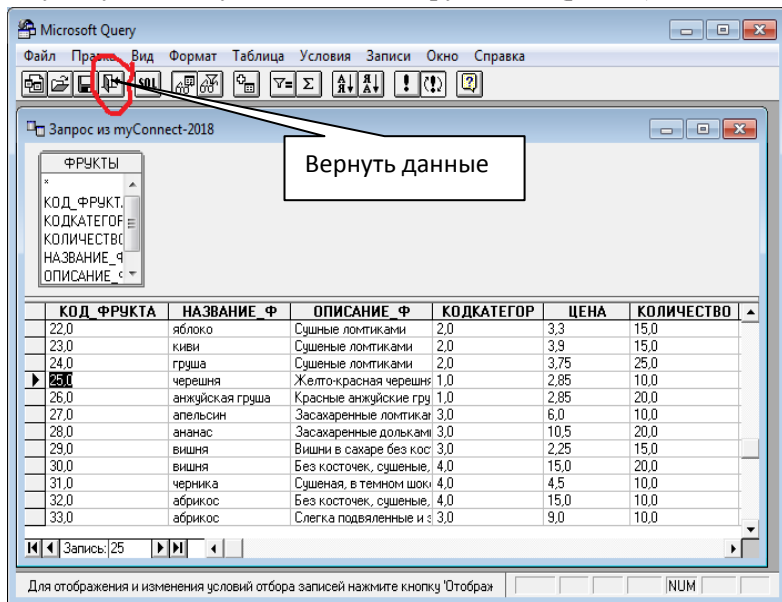


Рисунок 23 - Окно запроса с одной таблицей "Фрукты". Активизация диалога Импорт данных

При этом открывается диалоговое окно *Импорт данных* (рис.24).

Если передаваемые в MS Excel данные не требуют предварительной обработки, то из диалога *Импорт данных* можно перейти к мастеру создания сводных таблиц.

MS Query передает в MS Excel копию оригинальных данных. При изменении исходных данных соответствующих изменений в MS Excel не происходит. Актуализацию можно провести с помощью команды *Данные–Обновить данные*.

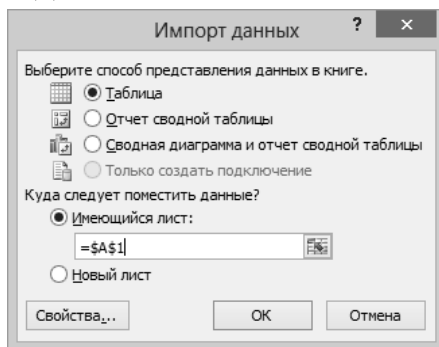


Рисунок 24 - Диалоговое окно *Импорт данных*

Результат импорта данных представлен на рис.25.

	A	B	C	D	E	F	G
1	ГОРОД	ДАТА_ЗАКАЗ	КОЛИЧЕСТВО	ЦЕНА	НАЗВАНИЕ_Ф	КОД_КАТЕГО	КАТЕГОРИЯ
2	Выборг	19.06.2016	3	1,5	инжир		2 Сушеные
3	Выборг	22.06.2016	5	2,85	черешня		1 Свежие
4	Выборг	25.06.2016	7	2,85	абрикос		2 Сушеные
5	Выборг	28.06.2016	10	1,5	инжир		2 Сушеные
36	Выборг	26.02.2017	25	2,85	черешня		1 Свежие
37	Санкт-Петер	04.04.2016	10	1,5	инжир		2 Сушеные
38	Санкт-Петер	05.04.2016	10	4,8	постные фрукты		1 Свежие
39	Санкт-Петер	06.04.2016	3	2,1	персик		1 Свежие
40	Санкт-Петер	07.04.2016	10	1,5	инжир		2 Сушеные
41	Санкт-Петер	08.04.2016	3	2,1	персик		1 Свежие
42	Санкт-Петер	09.04.2016	5	1,5	инжир		2 Сушеные
43	Санкт-Петер	10.04.2016	7	2,85	черешня		1 Свежие
44	Санкт-Петер	11.04.2016	10	1,5	инжир		2 Сушеные
45	Санкт-Петер	12.04.2016	3	2,85	абрикос		2 Сушеные

Рисунок 25. Результат импорта данных в MS Excel

РАБОТА С ЗАПИСЯМИ

MS Query предоставляет следующие возможности для непосредственной работы с записями. Переход к нужной записи осуществляется кнопками со стрелками в левом нижнем углу окна MSQuery. Для быстрого перехода к конкретной записи достаточно ввести ее номер в поле запись.

ОБРАБОТКА ЗАПИСЕЙ

В начале каждой записи в окне *MS Query* находится маркер записи:

- первая свободная запись обозначается звездочкой,
- черный треугольник указывает на текущую запись,
- во время обработки записи в маркере записи появляется символ карандаша.

Запись сохраняется на носителе, как только указатель перемещается на другую запись, т.е. сохранение не требует никаких дополнительных действий.

ВЫДЕЛЕНИЕ ПОЛЕЙ И ЗАПИСЕЙ

Чтобы удалить или скопировать поле или запись, их необходимо предварительно выделить. Поле считается выделенным в следующих случаях:

- выделен текст в этом поле,
- курсор ввода находится в этом поле,
- указатель "мыши" установлен на левой границе этого поля и произведен щелчок.

Запись выделена, если произведен щелчок "мышью" на маркере записи.

УДАЛЕНИЕ ПОЛЕЙ И ЗАПИСЕЙ

Если записи или поля предварительно выделены, их можно удалить стандартными способами: **Δ Правка Δ Удалить** или клавишей **Del**.

СОХРАНЕНИЕ ЗАПРОСА

Чтобы запомнить текущие параметры форматирования, а также для использования запроса в другое время или из другой книги, запрос можно сохранить. Для этого:

- нажать кнопку *Сохранить*;
- ввести имя в поле *Имя файла*;
- в диалоговом окне *Сохранить_как MS Query* автоматически выбирает тип *Файлы запросов (*.dqu)*; Для передачи данных в MS Excel не требуется сохранение запроса.

СОРТИРОВКА

Данные, для которых выполняется запрос, могут быть отсортированы. Для сортировки данных по возрастанию или по убыванию можно воспользоваться соответствующими кнопками на панели инструментов.

Для этого в окне *MS Query* следует выделить нужный столбец и вызвать команду **ΔЗаписи ΔСортировать**. Откроется диалоговое окно для определения критериев сортировки. Можно выполнить сортировку по нескольким критериям, если указать их в списке. Сортировка большого объема данных по нескольким критериям занимает много времени.

Довольно часто при работе с базами данных необходимо получить по некоторому полю только уникальные значения. Например, в предложенной базе следует получить названия фирм или имена клиентов, которым были выписаны счета. Естественно, что некоторые названия в базе встречаются несколько раз. Такую задачу можно решить, используя команду **Δ Вид ΔСвойства запроса** окна MS Query. Предварительно следует сделать выборку только по одному полю. В раскрывшемся окне ΔВидΔСвойства запроса надо установить флажок «Только уникальные значения».

УСЛОВИЯ ВЫБОРА

Для выбора записей удовлетворяющим каким-либо логическим условиям используется команда **ΔУсловия ΔДобавить условие** окна *MS Query*.

ЛОГИЧЕСКИЕ СВЯЗИ

Задавая условия выбора, можно обрабатывать те данные, которые удовлетворяют тем или иным критериям. Во многих случаях одного условия бывает недостаточно. При определении нескольких условий необходимо хорошо представлять их

взаимодействие. Определяются два основных вида логических связей:

- логическое **"И"**: записи выбираются только тогда, когда все условия выполняются;
- логическое **"ИЛИ"**: запись выбирается, если выполнено хотя бы одно условие.

ЗАПРОСЫ ИЗ НЕСКОЛЬКИХ ТАБЛИЦ. СВЯЗИ.

Запросы из нескольких таблиц часто используются в реляционных базах данных, в силу их конструктивных особенностей *MS Query* позволяет делать выборку данных из нескольких таблиц. Как правило, таблицы необходимо связать между собой.

В *MS Query* можно устанавливать различные свойства связей. По умолчанию устанавливается связь, при которой используются записи из левой и правой таблиц, связанные заданным соотношением. Такое объединение (связь) называется **внутренним объединением** INNER JOIN (синоним JOIN, ключевое слово INNER можно опустить). Выбираются только те данные, которые содержат совпадающие поля из объединяемых таблиц.

Можно установить и такой тип связи, при котором в результирующую таблицу будут включены все записи из левой таблицы и только те записи из правой, которые связаны с записями из левой указанными соотношениями, и наоборот. Такие объединения называются **левым внешним объединением** (LEFT OUTER JOIN) и **правым внешним объединением** (RIGHT OUTER JOIN) соответственно.

Установка типа связей в *MS Query* осуществляется в диалоговом окне «Объединения».

Вернемся к нашей ИС «Фруктовая база» и вспомним пример 1, в котором были выбраны все записи из файла Фрукты.dbf. При этом в результате была получена выборка (таблица), в которой категория каждого фрукта указана как код, т.е. в виде числа, что не совсем удобно.

Чтобы разрешить эту проблему рассмотрим следующий пример.

Пример 2

Выбрать все записи из файла Фрукты.dbf с явным указанием категории (расшифровкой), с помощью таблицы (файла) с именем Категор.dbf.

С помощью команды **ΔТаблицаΔДобавить таблицу** вызовем две таблицы Фрукты.dbf и Категор.dbf и установим между ними связь по двум полям (Категория и Код категории) следующим образом: выделим нужное поле в одной из таблиц и установим указатель "мыши", нажмем левую кнопку и, не отпуская ее, переместим указатель на имя соответствующего столбца в другой таблице. Между выбранными именами двух полей этих таблиц появится линия, которая отражает связь между ними.

Для того, чтобы изменить свойства связи, надо щелкнуть на линии связи между таблицами два раза. Откроется диалог Объединения (рис.26), в котором можно определить один из типов связи между таблицами.

После выбора необходимых полей из двух таблиц итоговый вид сформированного запроса будет иметь форму показанную на рисунке 27.

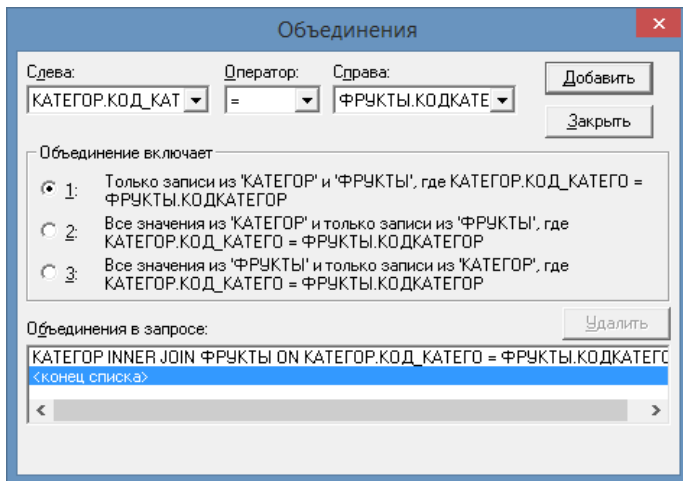


Рисунок 26 - Диалоговое окно для выбора типа объединения

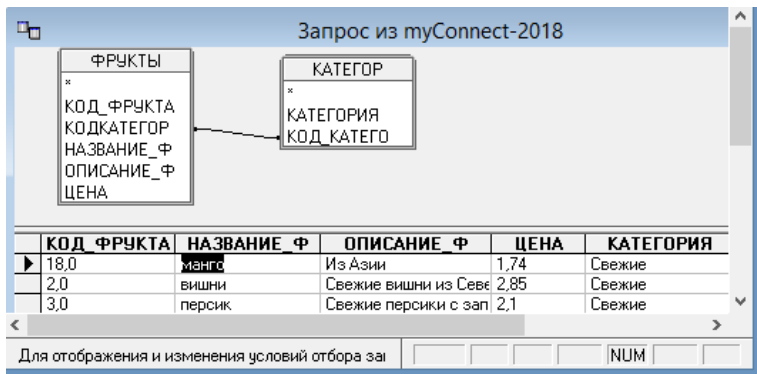


Рисунок 27 - Диалоговое окно *MS Query* с двумя связанными таблицами

Для иллюстрации необходимости объединения таблиц в реляционной модели данных рассмотрим формирование отчета о покупках фруктов определенных категорий для выбранных городов.

Пример 3

Выбрать информацию о всех фруктах, кроме сушеных, реализованных в заказах в городах 'Выборг' и 'Санкт-Петербург'. Информация должна включать: город (место реализации), дату заказа, название фрукта, количество, цену, категорию (фрукта). Выбранные данные упорядочить по городам.

Для решения поставленной задачи необходимо использовать данные содержащиеся во всех четырех таблиц, входящих в ИС «Фруктовая база». Таблицы должны быть связаны в соответствии с описанием ИС (рис.28).

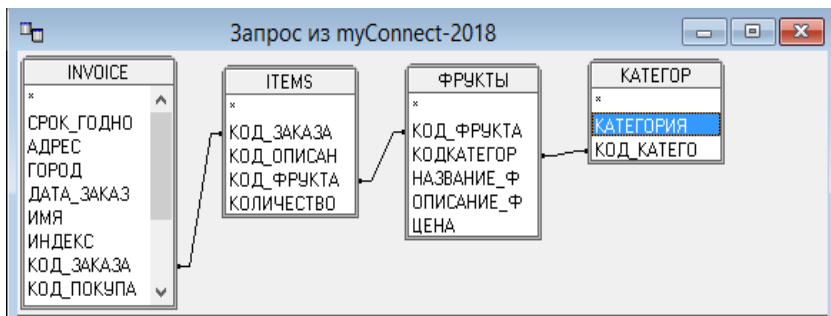


Рисунок 28 - Диалоговое окно *MS Query* с четырьмя связанными таблицами

После выбора необходимых полей из таблиц установим условия отбора записей, относящихся к городам 'Выборг' или 'Санкт-Петербург'.

Выберем команду Δ Условия Δ Добавить условие окна MS Query. Откроется диалоговое окно «Добавление условия» (рис.29а). Значения полей и операторов выбираем из раскрывающихся списков. Определим критерий по полю ГОРОД таблицы INVOICE. В поле «Значение» поставим значение 'Выборг'. Чтобы добавить в запрос данные, относящиеся ко второму городу, продолжим работу с текущим диалоговым окном. Нажимаем кнопку «Добавить», и активизируем радиокнопку «Или», после чего в поле «Значение» подставим значение 'Санкт-Петербург' (рис. 29б).

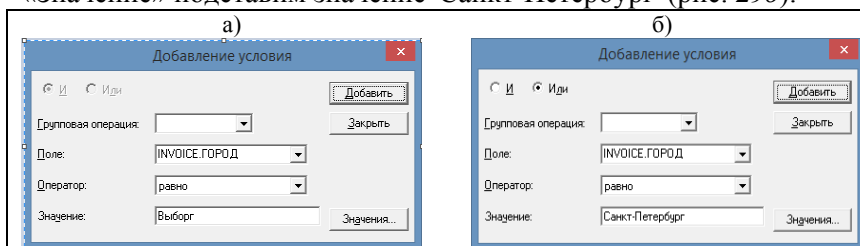


Рисунок 29 - Работа с диалоговым окном «Добавление условия».

Для добавления условия на категорию фруктов продолжим диалог, нажав на кнопку «Добавить», в обновленном окне устанавливаем критерий по полю КАТЕГОРИЯ таблицы КАТЕГОР. В поле «Оператор» поставим значение 'не равно'. В поле «Значение» поставим значение 'Сушёные', (рис. 30). Завершаем диалог, нажав на кнопку «Закреть».

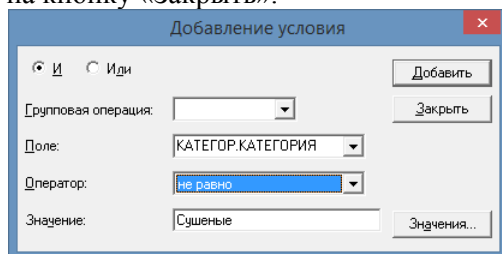


Рисунок 30 - Продолжение работы с диалоговым окном «Добавление условия».

Чтобы упорядочить записи по полю «Город» выделяем столбец «Город», щелкнув по его заголовку и нажимаем клавишу



на панели инструментов. Заметим, что более сложную сортировку (по нескольким полям) можно выполнить, используя команду Δ Записи Δ Сортировать.

Итоговый вид сформированного запроса будет иметь форму, показанную на рисунке 31.

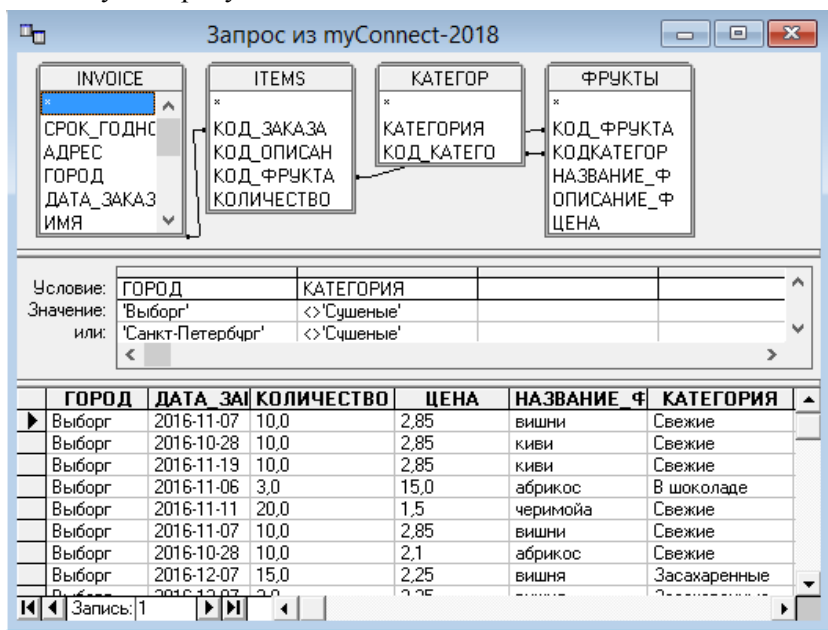


Рисунок 31 - Диалоговое окно MS Query с отображением критериев выборки.

То, что критерии по полям в центральной части окна запроса расположены рядом, показывает работу логической связи "И". Если работает логическая связь "ИЛИ", следующий критерий располагается строкой ниже.

Для понимания работы *MS Query* следует помнить, что любому запросу, сформированному с помощью диалогового окна *MS Query*, соответствует выражение на языке SQL, которое принято называть *SQL-запросом*. Посмотреть это выражение можно с

помощью активации окна «Запрос SQL», для этого необходимо нажать кнопку «SQL» или выполнить команду **ΔПросмотрΔSQL**. Окно, содержащее SQL запрос, соответствующий решению примера 3, показано на рис.32. Содержимое этого окна содержит плоский текст, который может быть отредактирован, скопирован и, при необходимости, вставлен в текст программы или текстовый документ.

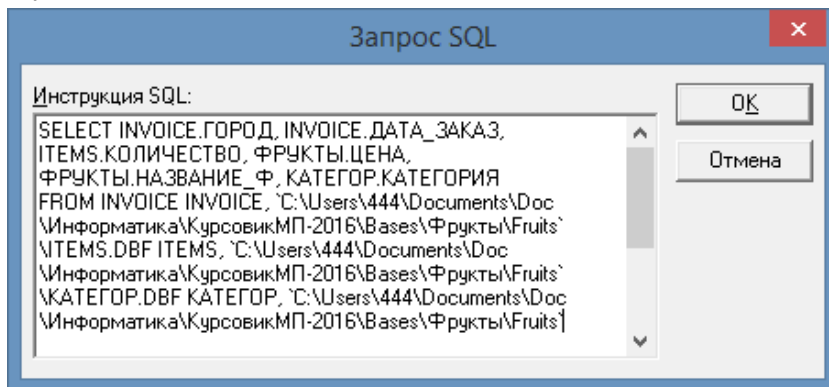


Рисунок 32 - Окно «Запрос SQL» с текстом запроса для решения примера 3

Текст запроса может иметь существенную длину и не всегда отображаться целиком в диалоговом окне, поэтому его необходимо разместить полностью в курсовой работе в виде текста. Запрос в виде текста приведен на рис.33.

```
SELECT INVOICE.ГОРОД, INVOICE.ДАТА_ЗАКАЗ, ITEMS.КОЛИЧЕСТВО,
ФРУКТЫ.ЦЕНА, ФРУКТЫ.НАЗВАНИЕ_Ф, КАТЕГОР.КАТЕГОРИЯ
FROM INVOICE INVOICE, ITEMS.DBF ITEMS, КАТЕГОР.DBF КАТЕГОР,
ФРУКТЫ.DBF ФРУКТЫ
WHERE КАТЕГОР.КОД_КАТЕГО = ФРУКТЫ.КОДКАТЕГОР AND
ФРУКТЫ.КОД_ФРУКТА = ITEMS.КОД_ФРУКТА AND ITEMS.КОД_ЗАКАЗА =
INVOICE.КОД_ЗАКАЗА AND ((INVOICE.ГОРОД='Выборг') AND
(КАТЕГОР.КАТЕГОРИЯ<>'Сушеные') OR (INVOICE.ГОРОД='Санкт-Петербург')
AND (КАТЕГОР.КАТЕГОРИЯ<>'Сушеные'))
```

Рисунок 33 - Текст запроса для решения примера 3

Любое изменение в диалоговом окне **MS Query** приводит к изменению SQL-запроса и наоборот, любое корректное изменение в

окне SQL-запроса приводит к изменению в диалоговом окне **MS Query**.

SQL-запросы записываются на языке SQL с соблюдением определенных правил.

ЯЗЫК ЗАПРОСОВ

SQL (*Structured Query Language*) -структурированный язык запросов является прежде всего информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных.

Первоначально SQL был разработан фирмой IBM. Целью разработки было создание простого непроцедурного языка, которым мог воспользоваться любой пользователь, даже не имеющий навыков программирования. Теперь он стандартизирован, но многие разработчики программных продуктов расширили его собственными элементами.

Каждая реализация языка SQL называется диалектом. Компания Microsoft реализует в своих продуктах два диалекта: Transact-SQL в продуктах MS SQL Server и Jet SQL в MS Access

С помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Чем сложнее сконструирован запрос, тем больше он допускает вариантов написания, различных по скорости выполнения, но одинаковых по итоговому набору данных.

Преимуществом языка SQL является его независимость от конкретной СУБД. Несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов могут быть достаточно легко перенесены из одной СУБД в другую.

Анализ данных с помощью Мастера сводных таблиц

Если передаваемые в MS Excel данные не требуют предварительной обработки, то из диалога **Импорт данных** можно перейти к созданию сводных таблиц.

Фрагмент листа Excel с результатами работы SQL-запроса представлен на рисунке 34.

	A	B	C	D	E	F	G
1	ГОРОД	ДАТА_ЗАКАЗ	КОЛИЧЕСТВО	ЦЕНА	НАЗВАНИЕ_Ф	КОД_КАТЕГО	КАТЕГОРИЯ
2	Выборг	19.06.2016	3	1,5	инжир		2 Сушеные
29	Выборг	11.11.2016	20	2,85	абрикос		2 Сушеные
30	Выборг	19.11.2016	10	1,5	инжир		2 Сушеные
31	Выборг	22.11.2016	12	2,85	черешня		1 Свежие
32	Выборг	23.11.2016	16	4,8	постные фрукты		1 Свежие
33	Выборг	26.11.2016	10	2,85	черешня		1 Свежие
34	Выборг	07.12.2016	3	2,1	персик		1 Свежие
35	Выборг	07.12.2016	15	4,8	постные фрукты		1 Свежие
36	Выборг	26.02.2017	25	2,85	черешня		1 Свежие
37	Санкт-Петербург	04.04.2016	10	1,5	инжир		2 Сушеные
38	Санкт-Петербург	05.04.2016	10	4,8	постные фрукты		1 Свежие
39	Санкт-Петербург	06.04.2016	3	2,1	персик		1 Свежие
40	Санкт-Петербург	07.04.2016	10	1,5	инжир		2 Сушеные

Рисунок 34 - Фрагмент листа Excel с результатами работы SQL-запроса

Пример 4

Составить сводную таблицу по полученной выборке, сформированной в примере 3. В таблице строками должны быть *даты продажи*, столбцами - *категории фруктов*.

В качестве фильтра использовать наименования городов. Поле значений должно содержать суммарную стоимость проданных фруктов в день продажи с учетом категории фруктов. Работа со сводными таблицами описана [4].

Результат работы Мастера сводных таблиц приводится на рисунке 35.

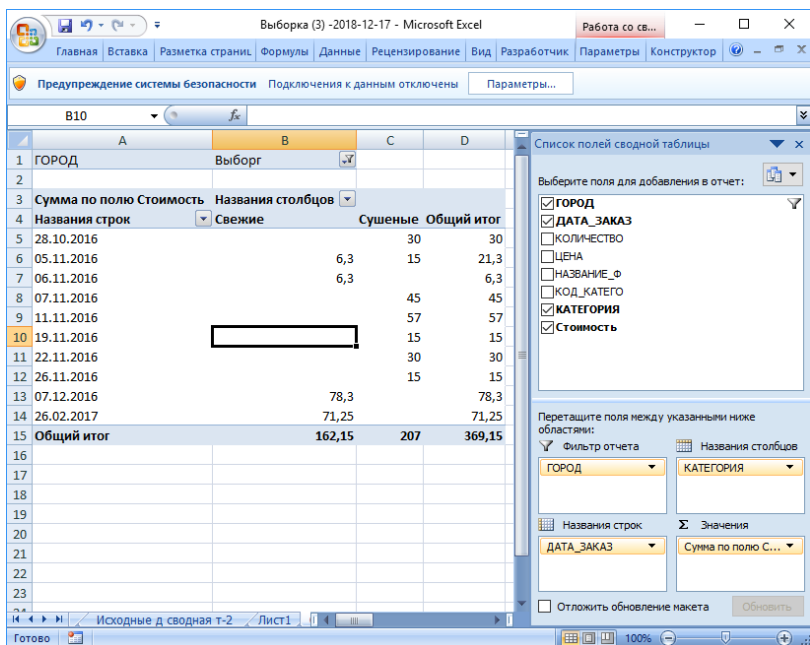


Рисунок 35 - Результат работы Мастера сводных таблиц

РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам. Единая система конструкторской документации (с Изменением №1). Режим доступа: <http://docs.cntd.ru/document/gost-2-105-95-eskd>
2. *Сергеев А.П.* Microsoft office 2010. Самоучитель. – М.: Вильямс, 2010. – 642 с.
3. *Музычкин П., Романова Ю.* EXCEL в экономических расчетах. Серия: Высшее Экономическое образование Издательство: Эксмо, 2009 г. – 304 с.
4. *Косовцева Т.Р.* Информационные технологии в менеджменте. Обработка списков. Сводные таблицы /Т.Р.Косовцева, Г.А. Прудинский, Л.Г. Муста. Метод. указания к лабораторным работам. СПб: РИЦ Горного университета, 2016. – 54 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ЗАДАНИЕ К КУРСОВОЙ РАБОТЕ ПО ИНФОРМАТИКЕ	3
АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ.....	6
ПОСЛЕДОВАТЕЛЬНОСТЬ ИЗЛОЖЕНИЯ МАТЕРИАЛА В ПОЯСНИТЕЛЬНОЙ ЗАПИСКЕ ПО КУРСОВОЙ РАБОТЕ.....	7
ТРЕБОВАНИЯ ПО ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ	7
КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	8
БАЗЫ ДАННЫХ.....	8
ИЕРАРХИЧЕСКИЕ БАЗЫ ДАННЫХ.....	8
СЕТЕВЫЕ БАЗЫ ДАННЫХ	11
РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ	13
ОДИН К ОДНОМУ	15
СВЯЗЬ ОДИН КО МНОГИМ.....	15
МНОГИЕ КО МНОГИМ.....	16
ИМПОРТ ДАННЫХ С ПОМОЩЬЮ MS-QUERY	19
ОСНОВНЫЕ ПОНЯТИЯ	19
СТРУКТУРА БАЗЫ ДАННЫХ.....	19
ФОРМИРОВАНИЕ ЗАПРОСА	23
ВОЗВРАТ В EXCEL.....	29
РАБОТА С ЗАПИСЯМИ	31
ОБРАБОТКА ЗАПИСЕЙ.....	31
ВЫДЕЛЕНИЕ ПОЛЕЙ И ЗАПИСЕЙ	31
УДАЛЕНИЕ ПОЛЕЙ И ЗАПИСЕЙ	31
СОХРАНЕНИЕ ЗАПРОСА.....	31
СОРТИРОВКА.....	32
УСЛОВИЯ ВЫБОРА	32
ЛОГИЧЕСКИЕ СВЯЗИ.....	32
ЗАПРОСЫ ИЗ НЕСКОЛЬКИХ ТАБЛИЦ, СВЯЗИ.	33
ЯЗЫК ЗАПРОСОВ	39
РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК	41

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕНЕДЖМЕНТЕ

*Методические указания к курсовой работе
для студентов бакалавриата направления 38.03.02*

Сост.: *В.В. Беляев, Т.Р. Косовцева, Л.Г. Муста*

Печатается с оригинал-макета, подготовленного кафедрой
информатики и компьютерных технологий

Ответственный за выпуск *В.В. Беляев*

Лицензия ИД № 06517 от 09.01.2002

Подписано к печати 03.09.2019. Формат 60×84/16.

Усл. печ. л. 2,4. Усл.кр.-отт. 2,4. Уч.-изд.л. 2,0. Тираж 75 экз. Заказ 736. С 253.

Санкт-Петербургский горный университет
РИЦ Санкт-Петербургского горного университета
Адрес университета и РИЦ: 199106 Санкт-Петербург, 21-я линия, 2