МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

ГОУВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономической кибернетики»

КУРСОВАЯ РАБОТА

на тему: «Информационная система торговой организации»

по дисциплине

«Базы данных»

Выполнил: студент группы БИ-21  
Скиндер Иван Павлович

Проверил: к.т.н., доц. Харитонов Ю.Е.

Донецк 2022 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc121142148)

[1. ПОСТАНОВКА ЗАДАЧИ 5](#_Toc121142149)

[2. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 8](#_Toc121142150)

[3. СХЕМА ДАННЫХ 10](#_Toc121142151)

[4. ТАБЛИЦЫ 12](#_Toc121142152)

[5. ФОРМЫ 21](#_Toc121142153)

[6. ЗАПРОСЫ 29](#_Toc121142154)

[7. ОТЧЕТЫ 41](#_Toc121142155)

[ЗАКЛЮЧЕНИЕ 45](#_Toc121142156)

[ЛИТЕРАТУРА 46](#_Toc121142157)

ВВЕДЕНИЕ

В настоящее время жизнь человека зависит от различного рода информации, для управления которой требуются создания огромного количества баз данных различного назначения.

Понятие базы данных (БД) можно применять к любой связанной между по определенному признаку информации, хранимой и ограниченной особым образом- что, выполняется в СУБД MS ACCESS в виде таблиц.

Данная курсовая работа рассматривает создание базы данных Строительной организации. В курсовой работе разрабатывается База Данных, с помощью которой, можно будет вести отчет о деятельности каждой бригады, составления графика работы, учета материалов.

База данных (БД)- совокупность взаимосвязанных и организованных по определенным правилам данных, которое отражает состояние объектов какой-то конкретной предметной области и отношения между этими объектами.

При создании БД важно учитывать следующие параметры:

* Целостность БД — правильность данных в любой момент времени
* Избыточность БД — дублирование данных в нескольких таблицах (избыточность влечет за собой противоречивость данных).

Цель проектирования БД — сокращение избыточных данных. Эта цель достигается с помощью теории нормализации баз данных.

Нормализация — это разбиение таблиц на два или более, обладающими лучшими свойствами при внесении или изменении данных. В результате получается БД, в которой каждый факт появляется лишь в одном месте, что исключает избыточность информации.

Поддержание целостности данных. Настраивая свойства полей, связывая таблицы между собой и настраивая ограничения, мы сможем увеличить надежность наших данных.

Access предоставляет максимальную свободу в задании типа данных — текст, числовые данные, даты, время, денежные значения, рисунки, звук, документы, электронные таблицы. Имеется возможность задавать форматы хранения (длина строки, точность представления чисел и даты времени) и представления этих данных при выводе на экран или печать[1-4]. СУБД Access для работы с данными использует процессор баз данных Microsoft Jet, объекты доступа к данным и средство быстрого построения интерфейса — конструктор форм. Для получения распечаток используется конструктор отчётов. Автоматизация рутинных операций может выполняться с помощью макрокоманд либо при помощи встроенного языка VBA (Visual Basic for Applications).

1. ПОСТАНОВКА ЗАДАЧИ

Целью курсовой является разработка базы данных для строительной организации.

Для достижения цели необходимо решить следующие задачи:

* описать предметную область;
* разработать схему данных;
* составить таблицы;
* сделать формы для работы с базой данных;
* сделать запросы, представленные в предметной области;
* сделать отчеты;
* сделать кнопочную форму и закончить оформление базы данных.

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Строительная организация занимается строительством различного рода объектов: жилых домов, больниц, школ так и бань, гаражей и т.д. по договорам с заказчиками. Рабочие (каменщики, бетонщики, отделочники, сварщики, электрики, шофера, слесари, и пр.) объединяется в бригады, которыми руководят бригадиры. Бригадиры выбираются из числа рабочих. На каждом объекте работу ведут одна или несколько бригад. Закончив работу, бригада переходит к другому объекту. Технология строительства того или иного объекта предполагает выполнение определенного набора видов работ, необходимых для сооружения данного типа объекта. Например, для жилого дома - это возведение фундамента, кирпичные работы, прокладка водоснабжения и т.д. Каждый вид работ на объекте выполняется одной бригадой. Для организации работ на объекте составляется графики работ, указывающие в каком порядке и в какие сроки выполняются те или иные работы, а также смета, определяющая какие строительные материалы и в каких количествах необходимы для сооружения объекта. По результатам выполнения работ составляется отчет с указанием сроков выполнения работ и фактических расходов материалов.

Виды запросов в информационной системе:

1. Получить перечень строительных управлений и/или участков и их руководителей.

2. Получить список специалистов инженерно-технического состава обозначенного участка или строительного управления с указанием их должностей.

3. Получить перечень объектов, возводимых указанным строительным управлением и/или участком, и графики их возведения.

4. Получить состав бригад, работавших (работающих) на строительстве указанного объекта.

5. Получить перечень строительной техники, приданной указанному строительному управлению.

6. Получить перечень строительной техники, выделенной на указанный объект либо работавшей там в течение указанного периода времени.

7. Получить график и смету на строительство указанного объекта.

8. Получить отчет о сооружении указанного объекта.

9. Получить перечень объектов, возводимых в некотором строительном управлении или в целом по организации, на которых в обозначенный период времени выполнялся указанный вид строительных работ.

10.Получить перечень видов строительных работ, по которым имело место превышение сроков выполнения на указанном участке, строительном управлении или в целом по организации.

11.Получить перечень строительных материалов, по которым имело место превышение по смете на указанном участке, строительном управлении или в целом по организации.

12.Получить перечень видов строительных работ, выполненных указанной бригадой в течение обозначенного периода времени с указанием объектов, где эти работы выполнялись.

13. Получить перечень бригад, выполненных указанный вид строительных работ в течение обозначенного периода времени с указанием объектов, где эти работы выполнялись.

1. СХЕМА ДАННЫХ

Схема данных является графическим образом БД. Она используется различными объектами Access для определения связей между несколькими таблицами. На рисунке 1 выше изображена схема данных в которой происходят различные связи между таблицами.

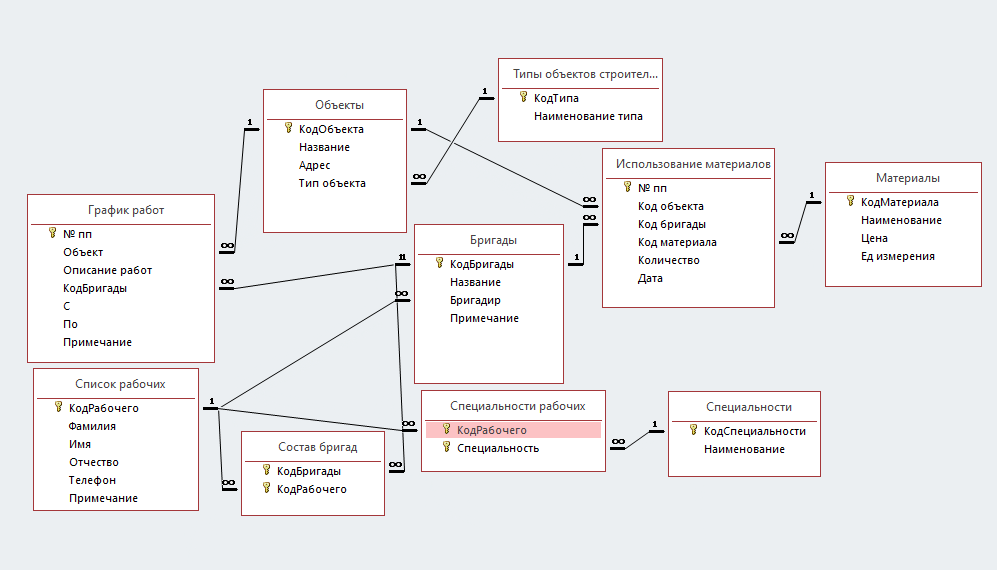


Рис. 1 — *Схема данных*

В табл. 1 проанализированы все связи между таблицами.

Табл. 1 *Таблица связей*

|  |  |
| --- | --- |
| «Один» | «Много» |
| Список рабочих: КодРабочего | Специальности рабочих: КодРабочего |
| Список рабочих: КодРабочего | Состав бригад: КодРабочего |
| Список рабочих: КодРабочего | Бригады: Бригадир |
| Специальности: КодСпециальности | Специальности рабочих: Специальность |
| Бригады: КодБригады | Состав бригад: КодБригады |
| Бригады: КодБригады | График работ: КодБригады |
| Бригады: КодБригады | Использование материалов: Код бригады |
| Объекты: КодОбъекта | График работ: Объект |
| Объекты: КодОбъекта | Использование материалов: Код объекта |
| Типы объектов строительства: КодТипа | Объекты: Тип объекта |
| Материалы: КодМатериала | Использование материалов: Код материала |

Информационная система содержит следующие сущности:

1. **Бригада** — содержит информацию о бригадире (любой рабочий может стать бригадиром).

* КодБригады — первичный ключ таблицы
* Название — название бригады
* Бригадир — является ссылкой (внешним ключом) для поля «КодРабочего» в таблице **«Список рабочих»**
* Примечание — примечание бригады

1. **Список рабочих** — содержит информацию о рабочих

* КодРабочего — первичный ключ таблицы
* Фамилия — фамилия рабочего
* Имя — имя рабочего
* Отчество — отчество рабочего
* Телефон — телефон рабочего
* Примечание — примечание рабочего

1. **Состав бригад** — соединительная таблица

* КодБригады, КодРабочего — составной первичный ключ, который состоит из двух внешних ключей, которые ссылаются на первичные ключи таблиц **«Бригада»** и **«Список рабочих»**.

1. **Специальности** — содержит информацию о специальностях

* КодСпециальности — первичный ключ таблицы
* Наименование — имя специальности

1. **Специальности рабочих** — соединительная таблица

* КодРабочего, Специальность — составной первичный ключ, который состоит из двух внешних ключей, которые ссылаются на первичные ключи таблиц **«Список рабочих»** и **«Специальности»**

1. **Объекты** — содержит информацию об объектах строительства

* КодОбъекта — первичный ключ таблицы
* Название — название объекта строительства
* Адрес — адрес объекта строительства
* Тип объекта — является ссылкой (внешним ключом) для поля «КодТипа» в таблице **«Типы объектов строительства»**

1. **Типы объектов строительства** — содержит информацию о типе объекта

* КодТипа — первичный ключ таблицы
* Наименование типа — название типа объекта строительства

1. **График работ —** содержит информацию о том, в какие сроки выполняются те или иные работы.

* № пп — первичный ключ таблицы
* Объект — является ссылкой (внешним ключом) для поля «КодОбъекта» в таблице **«Объекты»**
* Описание работ — какие работы проводятся на объекте (отделочные, монтажные, изоляционные, бетонные и пр.)
* КодБригады — является ссылкой (внешним ключом) для поля «КодБригады» в таблице **«Бригады»**
* С — Дата начала работ
* По — Дата окончания работ
* Примечание — Примечание по работе

1. **Использование материалов** — информация, касающаяся об использовании материалов определенной бригадой на определенном объекте строительства.

* № пп — первичный ключ таблицы
* Код объекта — является ссылкой (внешним ключом) для поля «КодОбъекта» в таблице **«Объекты»**.
* Код бригады — является ссылкой (внешним ключом) для поля «КодБригады» в таблице **«Бригады»**.
* Код материала — является ссылкой (внешним ключом) для поля «КодМатериала» в таблице **«Материалы»**.
* Количество — Количество материалов
* Дата — дата использования материала

1. **Материалы** — содержит информацию о материалах

* КодМатериала — первичный ключ таблицы
* Наименование — название материала
* Цена — цена материала
* Ед измерения — единица измерения (кг, руллон)

1. ТАБЛИЦЫ

Таблица - это структура, предназначенная для хранения информации в базе данных. Современные СУБД позволяют хранить в таблицах данные разных типов: числа, тексты, даты, денежные значения, графические изображения, гиперссылки и др. Строки таблицы называются записями, а столбцы - полями. Существуют информационные таблицы и справочные таблицы [2-4].

Нужно создать информационные и справочные таблицы, соответствующие предметной области.

При выполнении работы были созданы следующие таблицы:

Таблица «Бригады», представленная на рисунке 2, содержит в себе список бригад.

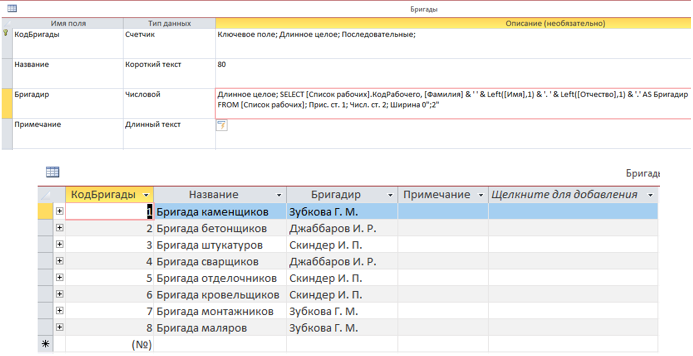


Рис. 2 *Таблица Бригады в режиме конструктора и просмотра*

Таблица «График работ», представленная на рисунке 3, хранит информацию о бригаде, объекте на котором они проводят определенные строительные работы, в том числе в какие сроки выполняются те или иные работы.

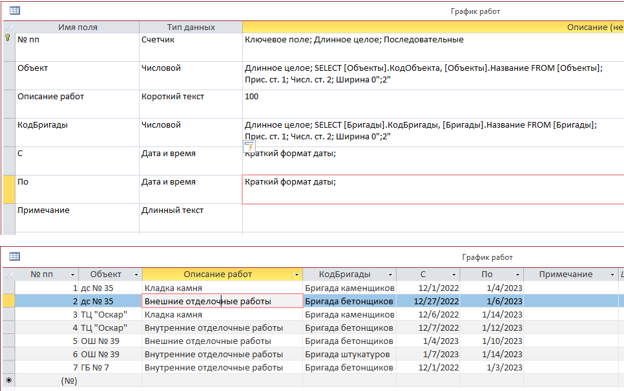


Рис. 3 *Таблица График работ в режиме конструктора и просмотра*

Таблица «Использование материалов» (рис. 4) содержит в себе материалы, необходимые для осуществления деятельности бригады на определенном строительном объекте, количество этих материалов и даты использования материала.

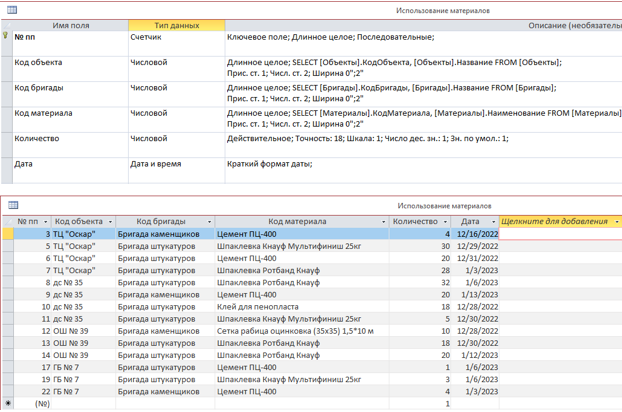


Рис. 4 *Таблица Использование материалов в режиме конструктора и просмотра*

Таблица «Материалы», представленная на рисунке 5, предназначена для хранения данных строительных материалов.

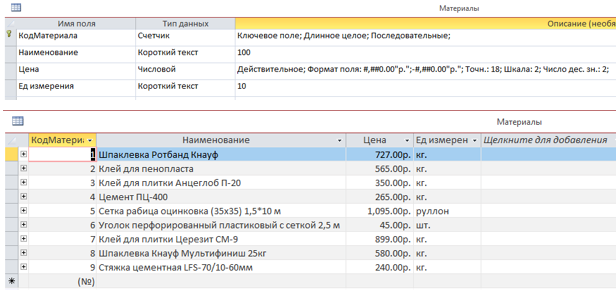


Рис. 5 *Таблица Материалы в режиме конструктора и просмотра*

Таблица «Объекты», представленная на рисунке 6, содержит в себе данные об адресе, типе объекта(жилой дом, больница, школа и пр.)

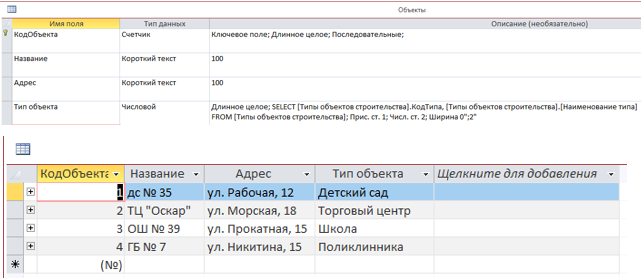


Рис. 6 *Таблица Объекты в режиме конструктора и просмотра*

Таблица «Состав бригад», представленная на рисунке 7, предназначена для хранения данных о составе бригад.

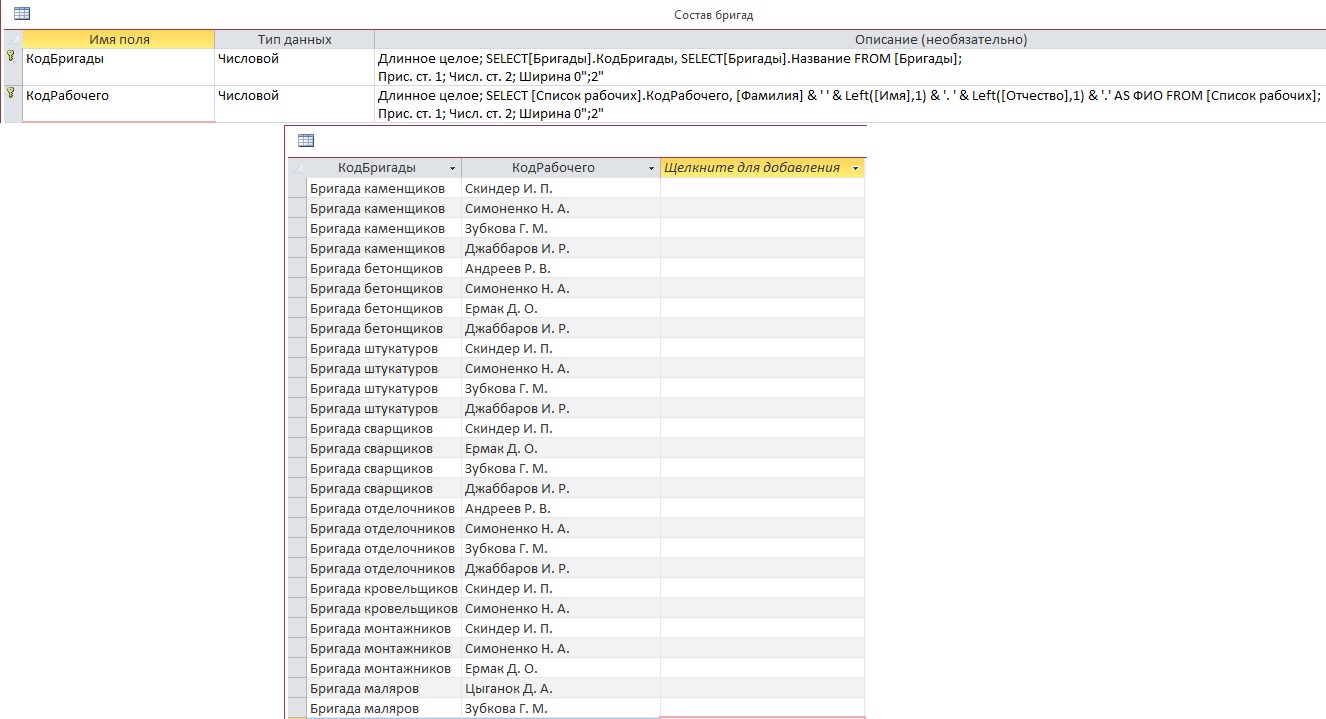


Рис. 7 *Таблица Состав бригад в режиме конструктора и просмотра*

Таблица «Специальности» (рис. 8) содержит в себе информацию специализации каждого рабочего.



Рис. 8 *Таблица Специальности в режиме конструктора и просмотра*

Таблица «Специальности рабочих», представленная на рисунке 9, содержит в себе данные о специальностях каждого рабочего.

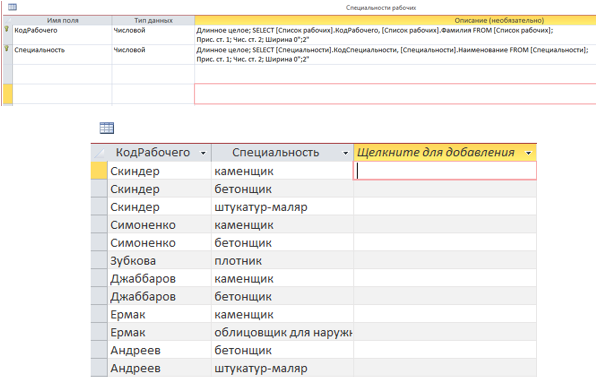


Рис. 9 *Таблица Специальности рабочих в режиме конструктора и просмотра*

Таблица «Список рабочих» (рис.10) предназначена для хранения данных о контактной информации рабочих.

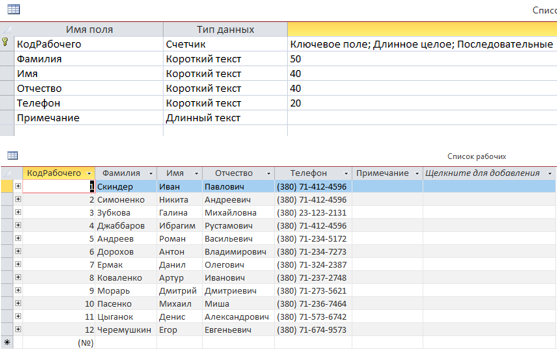
**

Рис. 10 *Таблица Список рабочих в режиме конструктора и просмотра*

Таблица «Типы объектов строительства» (рис.11) содержит в себе список возможных типов объектов строительства

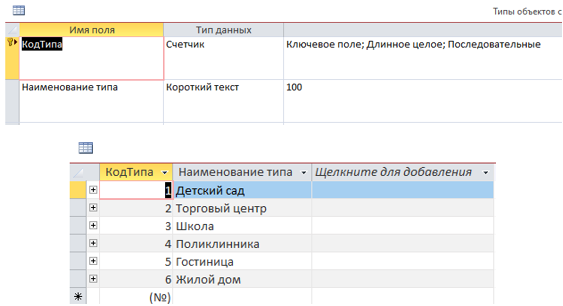


Рис. 11 *Таблица Типы объектов строительства в режиме конструктора и просмотра*

Таблица Switchboard Items (рис.12) создается автоматически при создании кнопочной формы через мастер кнопочных форм.

Кнопки кнопочной формы работают с помощью макрокоманд(табл.2).

Табл. 2 *Таблица макрокоманд кнопочной формы*

|  |  |
| --- | --- |
| Command | Макрокоманда |
| 1 | Перейти к кнопочной форме |
| 2 | Открыть форму для добавления |
| 3 | Открыть форму для изменения |
| 4 | Открыть отчет |
| 5 | Конструктор приложения |
| 6 | Выйти из приложения |
| 7 | Выполнить макрос |
| 8 | Выполнить программу |

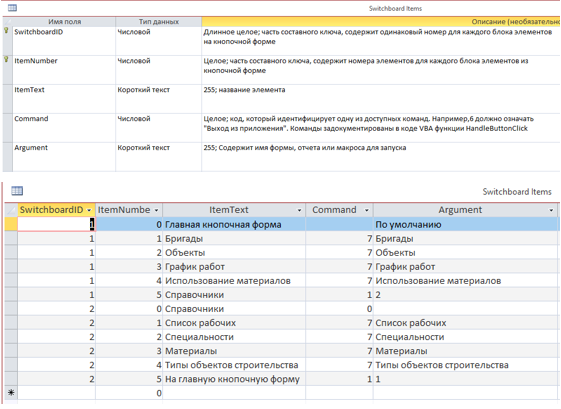


Рис. 12 *Таблица Switchboard Items в режиме конструктора и просмотра*

Справочные и информационные таблицы, созданные в данной базе данных полностью соответствуют требованиям предметной области.

1. ФОРМЫ

Формы, используются для ввода и просмотра таблиц в окне формы и позволяющие ограничить объем информации, отображаемой на экране в требуемом виде[2-4].

Для данной базы данных требуется создать формы, необходимые для ввода и просмотра данных всех таблиц базы данных.

Форма «Бригады» (рис. 13) предназначена для формирования состава бригад. Источник данных для нее — таблица «Бригады». Источник данных для подчиненной формы - таблица «Состав бригад». Связана подчиненная форма с основной по полю «Бригадир». Элементы управления формы:

— Кнопка «График работы» открывает форму «График работ» с условием отбора =” Формы![График работ]![КодБригады] = ” & Формы![Бригады]![КодБригады]

Код выполнения скрипта в VBA:

Private Sub ГрафикРабот\_Click()

Dim Answer As Integer

If Not ValidationModule.isFilled(Me![КодБригады]) Then

MsgBox "Бригада не выбрана!", vbOKOnly

Exit Sub

End If

Dim rst As Recordset

Set rst = CurrentDb.OpenRecordset("График работ", dbOpenDynaset)

rst.FindFirst "[КодБригады] = " & Me![КодБригады]

If rst.NoMatch Then

Answer = MsgBox("График по данной бригаде не составлен!" & Chr(13) & Chr(13) & "Создать?", vbOKCancel)

If Answer = 1 Then

DoCmd.OpenForm "График работ"

DoCmd.GoToRecord , , acNewRec

Forms("График работ")![КодБригады] = Me![КодБригады]

DoCmd.Close acForm, "Бригады", acSaveYes

End If

Exit Sub

End If

DoCmd.OpenForm "График работ", , , "[КодБригады] = " & Me![КодБригады]

DoCmd.Close acForm, "Бригады", acSaveYes

End Sub

— Кнопка «Используемые материалы» открывает форму «Использование материалов» с условием отбора = ”Формы![Использование материалов]![Код бригады] = ” & Формы![Бригады]![КодБригады]

Код выполнения скрипта в VBA:

Private Sub ИспользуемыеМатериалы\_Click()

Dim Answer As Integer

If Not ValidationModule.isFilled(Me![КодБригады]) Then

MsgBox "Бригада не выбрана!", vbOKOnly

Exit Sub

End If

Dim rst As Recordset

Set rst = CurrentDb.OpenRecordset("Использование материалов", dbOpenDynaset)

rst.FindFirst "[Код бригады] = " & Me![КодБригады]

If rst.NoMatch Then

Answer = MsgBox("У бригады отсутствуют материалы . . ." & Chr(13) & Chr(13) & "Добавить материал?", vbOKCancel)

If Answer = 1 Then

DoCmd.OpenForm "Использование материалов"

DoCmd.GoToRecord , , acNewRec

Forms("Использование материалов")![Код бригады] = Me![КодБригады]

DoCmd.Close acForm, "Бригады", acSaveYes

End If

Exit Sub

End If

DoCmd.OpenForm "Использование материалов", , , "[Код бригады] = " & Me![КодБригады]

DoCmd.Close acForm, "Бригады", acSaveYes

End Sub

— Кнопка «Информация о бригаде» открывает форму «Использование материалов» с условием отбора = “Формы![Список рабочих]![КодРабочего] = ” & Формы![Бригады]![Бригадир]

Код выполнения скрипта в VBA:

Private Sub Бригадир\_\_Click()

If Not ValidationModule.isFilled(Me![Бригадир]) Then

MsgBox "Бригадир не выбран!", vbOKOnly

Exit Sub

End If

DoCmd.OpenForm "Список рабочих", , , "[КодРабочего] = " & Me![Бригадир]

DoCmd.Close acForm, "Бригады", acSaveYes

End Sub

— Кнопка «Поиск» формирует отчет графика работ за период, указанный предварительно пользователем, в случае не заполнения одного из полей, появится диалоговое окно с уведомлением о том, что поля не были заполнены.

Код выполнения скрипта в VBA:

Private Sub Кнопка55\_Click()

Dim rst As Recordset

Dim Counter As Integer

If Not ValidationModule.isFilled(Me![ДатаС], Me![ДатаПо]) Then

MsgBox "Поля дат должны быть заполнены!", vbOKOnly

Exit Sub

End If

If Not ValidationModule.isFilled(Me![КодБригады]) Then

MsgBox "Бригада не выбрана!", vbOKOnly

Exit Sub

End If

Set rst = CurrentDb.OpenRecordset("SELECT Бригады.КодБригады, Бригады.Название, [Список рабочих]![Фамилия] & ' ' & [Список рабочих]![Имя] & ' ' & [Список рабочих]![Отчество] AS Бригадир, [График работ].С, [График работ].По, Объекты.Название, Объекты.Адрес, [График работ].[Описание работ] " & \_

"FROM [Список рабочих] INNER JOIN (Объекты INNER JOIN (Бригады INNER JOIN [График работ] ON Бригады.КодБригады = [График работ].КодБригады) ON Объекты.КодОбъекта = [График работ].Объект) ON [Список рабочих].КодРабочего = Бригады.Бригадир " & \_

"WHERE Бригады.КодБригады = " & Me![КодБригады] & " AND [График работ]![С] - " & Format(Me![ДатаС], "\#mm\/dd\/yyyy\#") & " >= 0 AND " & Format(Me![ДатаПо], "\#mm\/dd\/yyyy\#") & " - [График работ]![По] >= 0", dbOpenDynaset, dbReadOnly)

Counter = ValidationModule.RecCount(rst)

If Counter = 0 Then

MsgBox "Не найдено ни одной записи в промежутке от " & Me![ДатаС] & " до " & Me![ДатаПо], vbOKOnly

Else

DoCmd.OpenReport "График работ бригады за период", acViewReport

End If

rst.Close

End Sub

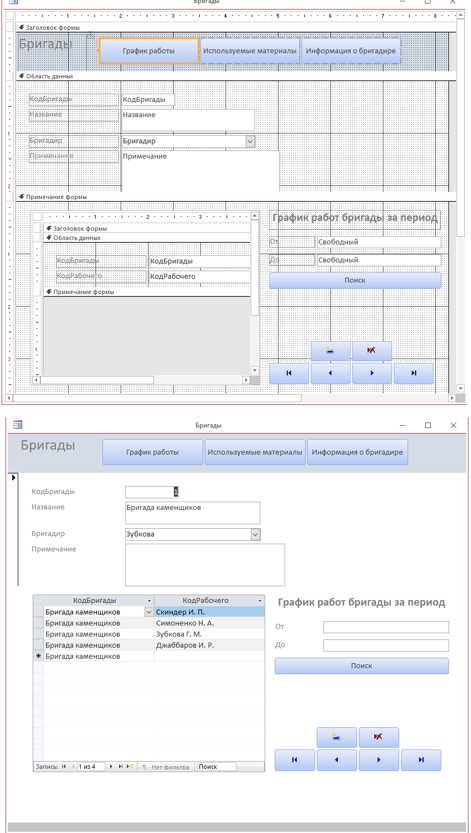


Рис. 13 *Форма Бригады в режиме Конструктора и формы*

Форма «График работ» (рис. 14) предназначена для составления графика работ на объекте строительства. Источник данных — таблица «График работ».

Элементы управления формы:

— Кнопка «Поиск»(График работ за период) формирует отчет графика работ каждой бригады на любом объекте за период, указанный предварительно пользователем.

Код выполнения скрипта в VBA:

Private Sub Кнопка62\_Click()

Dim rst As Recordset

Dim Counter As Integer

If Not ValidationModule.isFilled(Me![ДатаС], Me![ДатаПо]) Then

MsgBox "Поля дат должны быть заполнены!", vbOKOnly

Exit Sub

End If

Set rst = CurrentDb.OpenRecordset("SELECT [График работ].С, [График работ].По, Объекты.Название, Объекты.Адрес, [График работ].[Описание работ], Бригады.Название, [Список рабочих]![Фамилия] & ' ' & Left([Список рабочих]![Имя],1) & '. ' & Left([Список рабочих]![Отчество],1) & '.' AS Бригадир " & \_

"FROM [Список рабочих] INNER JOIN (Объекты INNER JOIN (Бригады INNER JOIN [График работ] ON Бригады.КодБригады = [График работ].КодБригады) ON Объекты.КодОбъекта = [График работ].Объект) ON [Список рабочих].КодРабочего = Бригады.Бригадир " & \_

"WHERE [График работ]![С] - " & Format(Me![ДатаС], "\#mm\/dd\/yyyy\#") & " >= 0 AND " & Format(Me![ДатаПо], "\#mm\/dd\/yyyy\#") & " - [График работ]![По] >= 0", dbOpenDynaset, dbReadOnly)

Counter = ValidationModule.RecCount(rst)

If Counter = 0 Then

MsgBox "Не найдено ни одной записи в промежутке от " & Me![ДатаС] & " до " & Me![ДатаПо], vbOKOnly

Else

DoCmd.OpenReport "График работ за период", acViewReport

End If

rst.Close

End Sub

— Кнопка «Поиск»(График работ на объекте) формирует отчет графика работ за период, указанный предврарительно пользователем, с условием отбора = “Объекты.КодОбъекта = ” & Формы![График работ]![Объект]

Код выполнения скрипта в VBA:

Private Sub Кнопка80\_Click()

Dim rst As Recordset

Dim Counter As Integer

If Not ValidationModule.isFilled(Me![ДатаС\_Объект], Me![ДатаПо\_Объект]) Then

MsgBox "Поля дат должны быть заполнены!", vbOKOnly

Exit Sub

End If

If Not ValidationModule.isFilled(Me![Объект]) Then

MsgBox "Объект не выбран!", vbOKOnly

Exit Sub

End If

If Not ValidationModule.isFilled(Me![С], Me![По]) Then

MsgBox "Поле 'С' или Поле 'По' не заполнены в форме!", vbOKOnly

Exit Sub

End If

Set rst = CurrentDb.OpenRecordset("SELECT Объекты.КодОбъекта, Объекты.Название, Объекты.Адрес, Объекты.[Тип объекта], [График работ].[Описание работ], Бригады.Название, Бригады.Бригадир, [График работ].С, [График работ].По " & \_

"FROM Объекты INNER JOIN (Бригады INNER JOIN [График работ] ON Бригады.КодБригады = [График работ].КодБригады) ON Объекты.КодОбъекта = [График работ].Объект " & \_

"WHERE Объекты.КодОбъекта = " & Me![Объект] & " AND [График работ]![С] - " & Format(Me![ДатаС\_Объект], "\#mm\/dd\/yyyy\#") & " >= 0 AND " & Format(Me![ДатаПо\_Объект], "\#mm\/dd\/yyyy\#") & " - [График работ]![По] >= 0", dbOpenDynaset, dbReadOnly)

Counter = ValidationModule.RecCount(rst)

If Counter = 0 Then

MsgBox "Не найдено ни одной записи в промежутке от " & Me![ДатаС\_Объект] & " до " & Me![ДатаПо\_Объект], vbOKOnly

Else

DoCmd.OpenReport "График работ на объекте", acViewReport

End If

rst.Close

End Sub

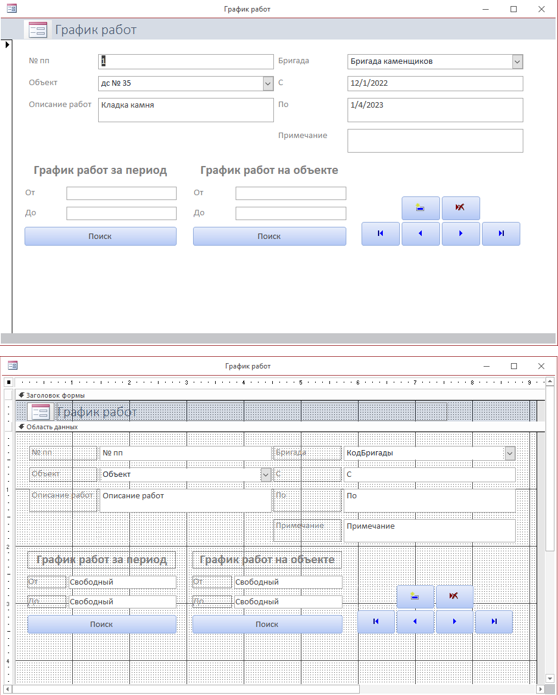


Рис. 14 *Форма График работ в режиме Конструктора и формы*

Форма «Использование материалов» (рис.15) предназначена для внесения, редактирования и удаления информации материалов каждой бригадой на определенном объекте. Источник данных для нее – таблица «Использование материалов». Элементы управления формы:

— Кнопка «Объект» открывает форму «Объекты», с условием отбора “Формы![Объекты]![КодОбъекта] = “ & Формы![Использование материалов][Код объекта]; В случае отсутствия в поле формы “Код объекта” значения, то появится соответствующее диалоговое окно.

Код выполнения скрипта в VBA:

Private Sub Объект\_Click()

If Not ValidationModule.isFilled(Me![Код объекта]) Then

MsgBox "Объект не выбран!", vbOKOnly

Exit Sub

End If

DoCmd.OpenForm "Объекты", , , "[КодОбъекта] = " & Me![Код объекта]

DoCmd.Close acForm, "Использование материалов", acSaveYes

End Sub

— Кнопка «Бригадир» открывает форму «Бригады», с условием отбора “Формы![Бригады]![КодБригады] = “ & Формы![Использование материалов][Код бригады]; Аналогично валидируется поле формы “Код бригады”.

Код выполнения скрипта в VBA:

Private Sub Бригадир\_Click()

If Not ValidationModule.isFilled(Me![Код бригады]) Then

MsgBox "Бригада не выбрана!", vbOKOnly

Exit Sub

End If

DoCmd.OpenForm "Бригады", , , "[КодБригады] = " & Me![Код бригады]

DoCmd.Close acForm, "Использование материалов", acSaveYes

End Sub

— Кнопка «Материал» открывает форму «Материалы», с условием отбора “Формы![Материалы]![КодМатериала] = “ & Формы![Использование материалов][Код материала]; Аналогично валидируется поле формы “Код материала”.

Код выполнения скрипта в VBA:

Private Sub Материал\_Click()

If Not ValidationModule.isFilled(Me![Код материала]) Then

MsgBox "Материал не выбран!", vbOKOnly

Exit Sub

End If

DoCmd.OpenForm "Материалы", , , "[КодМатериала] = " & Me![Код материала]

DoCmd.Close acForm, "Использование материалов", acSaveYes

End Sub

— Кнопка «Поиск»(Использование материалов за период) формирует список материалов, испозуемых в процессе деятельности бригады на объектах. Отбор записей — [Использование материалов].Дата Between [Формы]![Использование материалов]![ДатаС] And [Формы]![Использование материалов]![ДатаПо];

Код выполнения скрипта в VBA:

Private Sub Кнопка37\_Click()

Dim rst As Recordset

Dim Counter As Integer

If Not ValidationModule.isFilled(Me![ДатаС], Me![ДатаПо]) Then

MsgBox "Поля дат должны быть заполнены!", vbOKOnly

Exit Sub

End If

Set rst = CurrentDb.OpenRecordset("SELECT Материалы.КодМатериала, Материалы.Наименование, Материалы.[Ед измерения], Объекты.Название, Объекты.Адрес, Бригады.Бригадир, Материалы.Цена, [Использование материалов].Количество, Sum([Использование материалов]![Количество]\*[Материалы]![Цена]) AS Стоимость, [Использование материалов].Дата " & \_

"FROM Объекты INNER JOIN (Бригады INNER JOIN (Материалы INNER JOIN [Использование материалов] ON Материалы.КодМатериала = [Использование материалов].[Код материала]) ON Бригады.КодБригады = [Использование материалов].[Код бригады]) ON Объекты.КодОбъекта = [Использование материалов].[Код объекта] " & \_

"GROUP BY Материалы.КодМатериала, Материалы.Наименование, Материалы.[Ед измерения], Объекты.Название, Объекты.Адрес, Бригады.Бригадир, Материалы.Цена, [Использование материалов].Количество, [Использование материалов].Дата " & \_

"HAVING [Использование материалов].Дата Between " & Format(Me![ДатаС], "\#mm\/dd\/yyyy\#") & " And " & Format(Me![ДатаПо], "\#mm\/dd\/yyyy\#"), dbOpenDynaset, dbReadOnly)

Counter = ValidationModule.RecCount(rst)

If Counter = 0 Then

MsgBox "Не найдено ни одной записи в промежутке от " & Me![ДатаС] & " до " & Me![ДатаПо], vbOKOnly

Else

DoCmd.OpenReport "Использование материалов за период", acViewReport

End If

rst.Close

End Sub

— Поле Стоимость: Данные=[Цена]\*[Количество];

Формат поля: #,##0.00"р.";-#,##0.00"р."

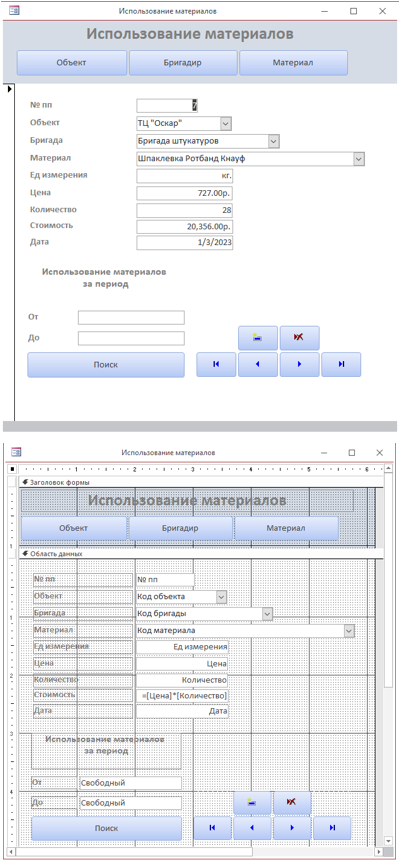


Рис. 15 *Форма Использование материалов в режиме Конструктора и формы*

Форма «Материалы» (рис.16) предназначена для формирования отчета по использованию определенного материала за период, предварительно указанной пользователем, также для добавления, редактирования и удаления материалов. Источник данных — таблица Материалы. Элементы управления:

Рис. 16 *Форма Продажи в режиме Конструктора и формы*

Форма «Сотрудники» (рис.19) предназначена для внесения, редактирования и удаления информации про сотрудников. Источник данных для нее – таблица Сотрудники. Элементы управления:

-кнопка «Выработка продавца 2022» открывает запрос «4\_1ВыработкаПродавца2022»;

-кнопка «Сотрудники по указанной ТТ» открывает запрос «7\_1СотрудникиПоТорговойТочке»;

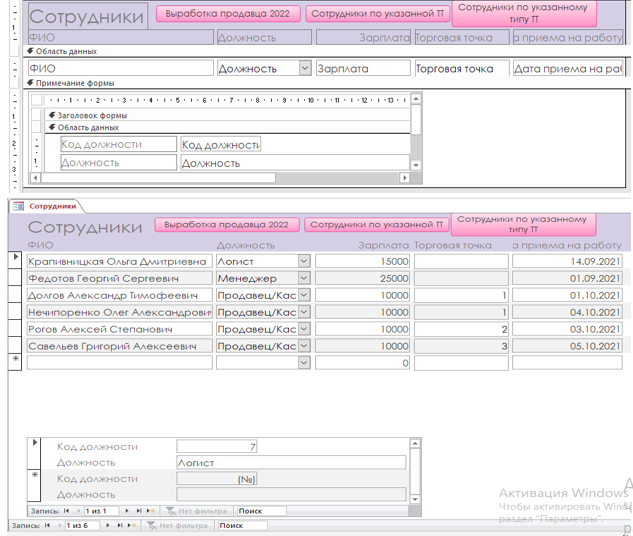
-кнопка «Сотрудники по указанному типу ТТ» открывает запрос «7\_2СотрудникиПоТипуТоргТочки».

Рис. 19 *Форма Сотрудники в режиме Конструктора и формы*

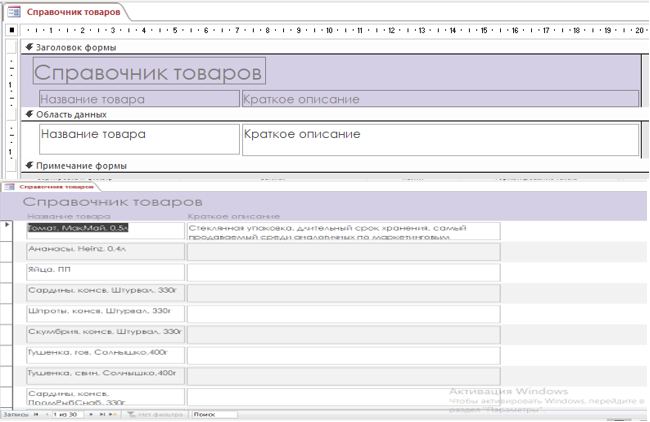
Форма «Справочник товаров» (рис.20) предназначена для внесения, редактирования и удаления информации про товары. Источник данных для нее – таблица Справочник товаров.

Рис. 20 *Форма Справочник товаров в режиме Конструктора и формы*

Форма «Торговые точки» (рис.21) предназначена для внесения, редактирования и удаления информации про торговые точки. Источник данных для нее – таблица Торговые точки. Элемент управления:

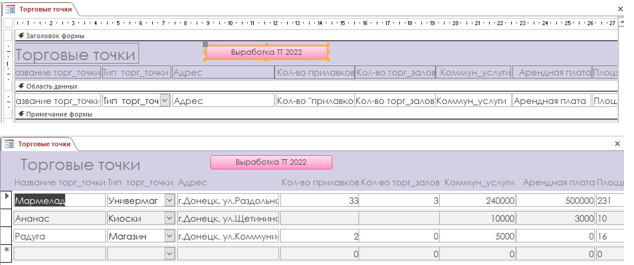
-кнопка «Выработка ТТ 2022» открывает запрос «4\_2ВыработкаТоргТочки2022».

Рис. 21 *Форма Торговые точки в режиме Конструктора и формы*

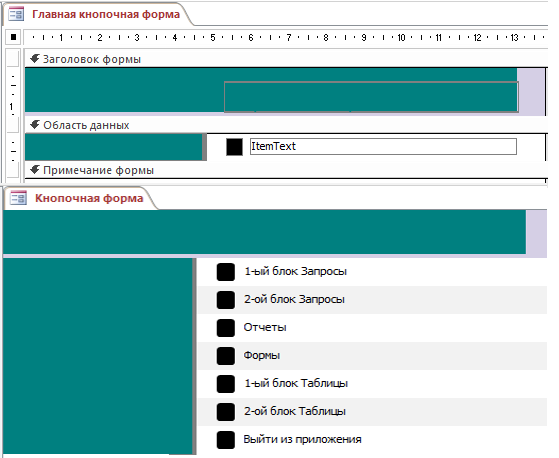
«Главная кнопочная форма» (рис.22) предназначена для удобной работы пользователя. Источник данных для нее – таблица Switchboard items.

Рис. 22 *Главная кнопочная форма в режиме Конструктора и формы*

Подводя итог, можно сказать что были созданы все необходимые формы для ввода и просмотра данных всех таблиц базы данных.

1. ЗАПРОСЫ

Запросы - это объект базы данных, который служит для извлечения данных из таблиц и предоставления их пользователю в удобном виде. Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную таблицу. Применение запросов позволяет избежать дублирования данных в таблицах и обеспечивает максимальную гибкость при поиске и отображении данных в базе данных[2-4].

Для данной базы данных необходимо создать запросы, соответствующие описанию предметной области.

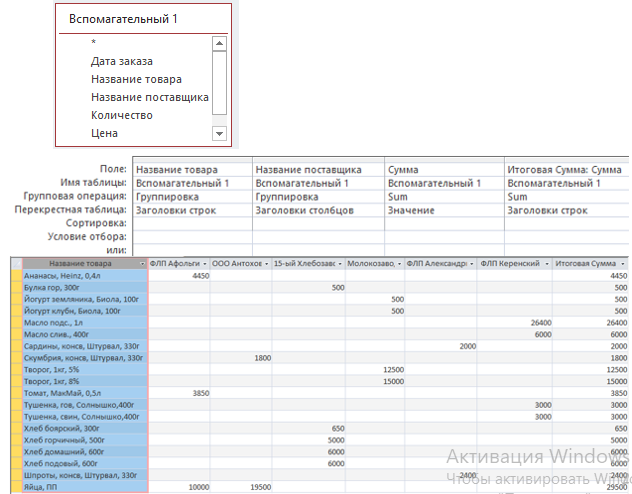
Запрос «1\_1Заказанные товарыОбъем2022Поставщики» (рис. 23) предназначен для просмотра объемов заказа определенного вида товара у каждого из поставщиков.

Рис. 23 *Запрос 1\_1Заказанные товарыОбъем2022Поставщики в режиме конструктора и просмотра*

Sql инструкция данного запроса:

TRANSFORM Sum([Вспомагательный 1].Сумма) AS [Sum-Сумма]

SELECT [Вспомагательный 1].[Название товара], Sum([Вспомагательный 1].Сумма) AS [Итоговая Сумма]

FROM [Вспомагательный 1]

GROUP BY [Вспомагательный 1].[Название товара]

PIVOT [Вспомагательный 1].[Название поставщика];

Sql инструкция вспомогательного запроса:

SELECT [Заказы поставок].[Дата заказа], [Справочник товаров].[Название товара], Поставщики.[Название поставщика], [Товары в заказе].Количество, [Товары в заказе].Цена, [Количество]\*[Цена] AS Сумма

FROM [Справочник товаров] INNER JOIN ([Заказы поставок] INNER JOIN (Поставщики INNER JOIN [Товары в заказе] ON Поставщики.[Код поставщика] = [Товары в заказе].[Поставщик]) ON [Заказы поставок].[Код заказа] = [Товары в заказе].[Код заказа]) ON [Справочник товаров].[Код товара] = [Товары в заказе].[Название товара]

WHERE ((([Заказы поставок].[Дата заказа]) Like "\*.\*.2022"))

ORDER BY [Количество]\*[Цена];

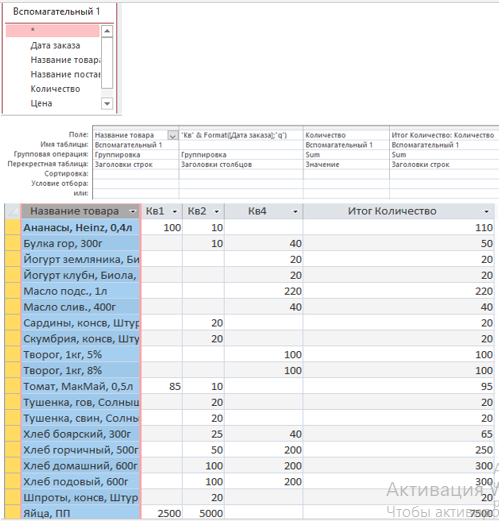
Запрос «1\_2Заказанные товарыКол-воПоКварталам» (рис. 24) предназначен для просмотра количества заказанных товаров по кварталам.

Рис. 24 *Запрос 1\_2Заказанные товарыКол-воПоКварталам в режиме конструктора и просмотра*

Sql инструкция данного запроса:

TRANSFORM Sum([Вспомагательный 1].Количество) AS [Sum-Количество]

SELECT [Вспомагательный 1].[Название товара], Sum([Вспомагательный 1].Количество) AS [Итог Количество]

FROM [Вспомагательный 1]

GROUP BY [Вспомагательный 1].[Название товара]

PIVOT "Кв" & Format([Дата заказа],"q");

Sql инструкция вспомогательного запроса:

SELECT [Заказы поставок].[Дата заказа], [Справочник товаров].[Название товара], Поставщики.[Название поставщика], [Товары в заказе].Количество, [Товары в заказе].Цена, [Количество]\*[Цена] AS Сумма

FROM [Справочник товаров] INNER JOIN ([Заказы поставок] INNER JOIN (Поставщики INNER JOIN [Товары в заказе] ON Поставщики.[Код поставщика] = [Товары в заказе].[Поставщик]) ON [Заказы поставок].[Код заказа] = [Товары в заказе].[Код заказа]) ON [Справочник товаров].[Код товара] = [Товары в заказе].[Название товара]

WHERE ((([Заказы поставок].[Дата заказа]) Like "\*.\*.2022"))

ORDER BY [Количество]\*[Цена];

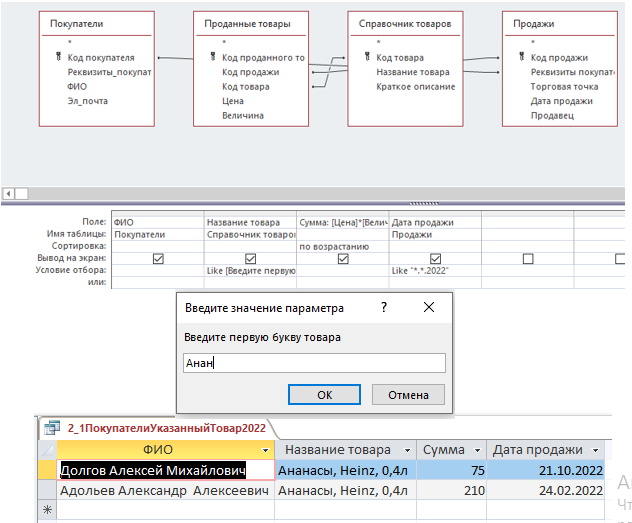
Запрос «2\_1ПокупателиУказанныйТовар2022» (рис.25) предназначен для просмотра покупателей указанного пользователем товара.

Рис. 25 *Запрос 2\_1ПокупателиУказанныйТовар2022 в режиме конструктора и просмотра*

Sql инструкция данного запроса:

SELECT Покупатели.ФИО, [Справочник товаров].[Название товара], [Цена]\*[Величина] AS Сумма, Продажи.[Дата продажи]

FROM (Покупатели INNER JOIN Продажи ON Покупатели.[Код покупателя] = Продажи.[Реквизиты покупателя]) INNER JOIN ([Справочник товаров] INNER JOIN [Проданные товары] ON [Справочник товаров].[Код товара] = [Проданные товары].[Код товара]) ON Продажи.[Код продажи] = [Проданные товары].[Код продажи]

WHERE ((([Справочник товаров].[Название товара]) Like [Введите первую букву товара] & "\*") AND ((Продажи.[Дата продажи]) Like "\*.\*.2022"))

ORDER BY [Цена]\*[Величина];

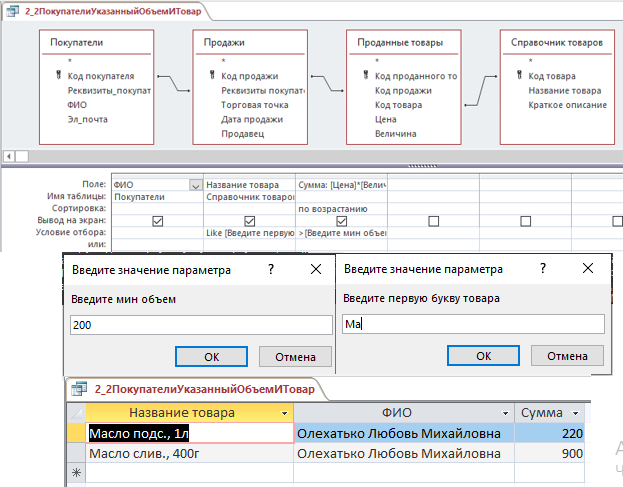
Запрос «2\_2ПокупателиУказанныйОбъемИТовар» (рис. 26) предназначен для просмотра покупателей минимального объема указанного пользователем товара.

Рис. 26 *Запрос 2\_2ПокупателиУказанныйОбъемИТовар в режиме конструктора и просмотра*

Sql инструкция данного запроса:

PARAMETERS [Введите мин объем] IEEEDouble, [Введите первую букву товара] Text ( 255 );

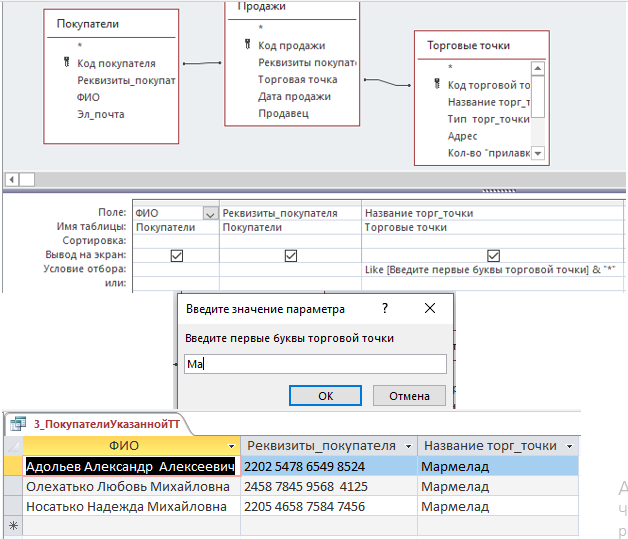
SELECT Покупатели.ФИО, [Справочник товаров].[Название товара], [Цена]\*[Величина] AS Сумма

FROM [Справочник товаров] INNER JOIN ((Покупатели INNER JOIN Продажи ON Покупатели.[Код покупателя] = Продажи.[Реквизиты покупателя]) INNER JOIN [Проданные товары] ON Продажи.[Код продажи] = [Проданные товары].[Код продажи]) ON [Справочник товаров].[Код товара] = [Проданные товары].[Код товара]

WHERE ((([Справочник товаров].[Название товара]) Like [Введите первую букву товара] & "\*") AND (([Цена]\*[Величина])>[Введите мин объем]))

ORDER BY [Цена]\*[Величина];

Запрос «3\_ПокупателиУказаннойТТ» (рис.27) предназначен для просмотра покупателей по указанной пользователем торговой точке.

Рис. 27 *Запрос 3\_ПокупателиУказаннойТТ в режиме конструктора и просмотра*

Sql инструкция данного запроса:

PARAMETERS [Введите первые буквы торговой точки] Text ( 255 );

SELECT Покупатели.ФИО, Покупатели.Реквизиты\_покупателя, [Торговые точки].[Название торг\_точки]

FROM [Торговые точки] INNER JOIN (Покупатели INNER JOIN Продажи ON Покупатели.[Код покупателя] = Продажи.[Реквизиты покупателя]) ON [Торговые точки].[Код торговой точки] = Продажи.[Торговая точка]

WHERE ((([Торговые точки].[Название торг\_точки]) Like [Введите первые буквы торговой точки] & "\*"));

Запрос «4\_1ВыработкаПродавца2022» (рис. 28) предназначен для определения выработки каждого продавца, которая находится как сумма проданного продавцом товара.

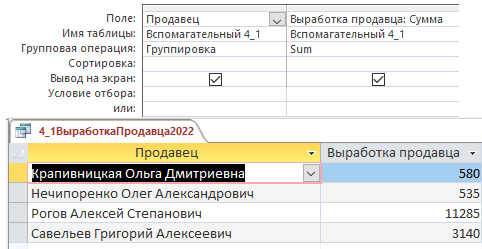


Рис. 28 *Запрос 4\_1ВыработкаПродавца2022* *в режиме конструктора и просмотра*

Sql инструкция к данному запросу:

SELECT DISTINCTROW [Вспомагательный 4\_1].Продавец, Sum([Вспомагательный 4\_1].Сумма) AS [Выработка продавца]

FROM [Вспомагательный 4\_1]

GROUP BY [Вспомагательный 4\_1].Продавец;

Sql инструкция к вспомогательному запросу:

SELECT Продажи.Продавец, [Цена]\*[Величина] AS Сумма, Продажи.[Дата продажи]

FROM Продажи INNER JOIN [Проданные товары] ON Продажи.[Код продажи] = [Проданные товары].[Код продажи]

WHERE (((Продажи.[Дата продажи]) Like "\*.\*.2022"));

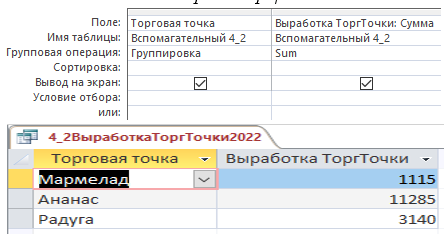
Запрос «4\_2ВыработкаТоргТочки2022» (рис. 29) предназначен для определения выработки каждой торговой точки, которая находится как сумма проданного в данной торговой точке товара.

Рис. 29 *Запрос 4\_2ВыработкаТоргТочки2022 в режиме конструктора и просмотра*

SELECT DISTINCTROW [Вспомагательный 4\_2].[Торговая точка], Sum([Вспомагательный 4\_2].Сумма) AS [Выработка ТоргТочки]

FROM [Вспомагательный 4\_2]

GROUP BY [Вспомагательный 4\_2].[Торговая точка];

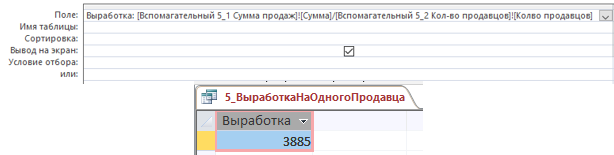
Запрос «5\_ВыработкаНаОдногоПродавца» (рис.30) предназначен для определения средней выработки на одного продавца.

Рис. 30 *Запрос 5\_ВыработкаНаОдногоПродавца в режиме конструктора и просмотра*

Sql инструкция данного запроса:

SELECT [Вспомагательный 5\_1 Сумма продаж]![Сумма]/[Вспомагательный 5\_2 Кол-во продавцов]![Колво продавцов] AS Выработка

FROM [Вспомагательный 5\_1 Сумма продаж], [Вспомагательный 5\_2 Кол-во продавцов];

Sql инструкция первого вспомогательного запроса:

SELECT Sum([Цена]\*[Величина]) AS Сумма

FROM [Проданные товары];

Sql инструкция второго вспомогательного запроса:

SELECT Count(Сотрудники.[Торговая точка]) AS [Колво продавцов]

FROM Сотрудники

GROUP BY Сотрудники.Должность

HAVING (((Count(Сотрудники.[Торговая точка]))<>0));

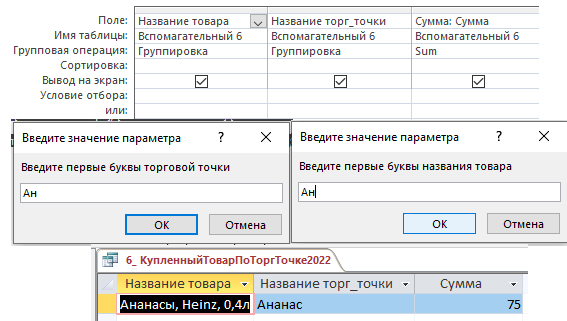
**Запрос «6\_ КупленныйТоварПоТоргТочке2022» (рис.31) предназначен для просмотра покупок определенного товара по указанной торговой точке.

Рис. 31 *Запрос 6\_ КупленныйТоварПоТоргТочке2022 в режиме конструктора и просмотра*

Sql инструкция данного запроса:

SELECT DISTINCTROW [Вспомагательный 6].[Название товара], [Вспомагательный 6].[Название торг\_точки], Sum([Вспомагательный 6].Сумма) AS Сумма

FROM [Вспомагательный 6]

GROUP BY [Вспомагательный 6].[Название товара], [Вспомагательный 6].[Название торг\_точки];

Sql инструкция вспомогательного запроса:

PARAMETERS [Введите первые буквы торговой точки] Text ( 255 ), [Введите первые буквы названия товара] Text ( 255 );

SELECT Продажи.[Дата продажи], [Справочник товаров].[Название товара], [Торговые точки].[Название торг\_точки], [Величина]\*[Цена] AS Сумма

FROM [Торговые точки] INNER JOIN ([Справочник товаров] INNER JOIN ((Покупатели INNER JOIN Продажи ON Покупатели.[Код покупателя] = Продажи.[Реквизиты покупателя]) INNER JOIN [Проданные товары] ON Продажи.[Код продажи] = [Проданные товары].[Код продажи]) ON [Справочник товаров].[Код товара] = [Проданные товары].[Код товара]) ON [Торговые точки].[Код торговой точки] = Продажи.[Торговая точка]

WHERE (((Продажи.[Дата продажи]) Like "\*.\*.2022") AND (([Справочник товаров].[Название товара]) Like [Введите первые буквы названия товара] & "\*") AND (([Торговые точки].[Название торг\_точки]) Like [Введите первые буквы торговой точки] & "\*"))

ORDER BY [Величина]\*[Цена];

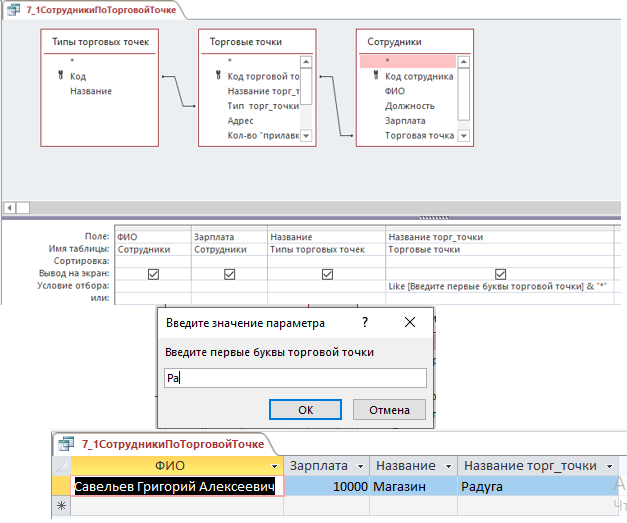
Запрос «7\_1СотрудникиПоТорговойТочке» (рис.32) предназначен для просмотра сотрудников по указанной торговой точке.

Рис. 32 *Запрос 7\_1СотрудникиПоТорговойТочке в режиме конструктора и просмотра*

Sql инструкция к данному запросу:

PARAMETERS [Введите первые буквы торговой точки] Text ( 255 );

SELECT Сотрудники.ФИО, Сотрудники.Зарплата, [Типы торговых точек].Название, [Торговые точки].[Название торг\_точки]

FROM ([Типы торговых точек] INNER JOIN [Торговые точки] ON [Типы торговых точек].[Код] = [Торговые точки].[Тип торг\_точки]) INNER JOIN Сотрудники ON [Торговые точки].[Код торговой точки] = Сотрудники.[Торговая точка]

WHERE ((([Торговые точки].[Название торг\_точки]) Like [Введите первые буквы торговой точки] & "\*"));

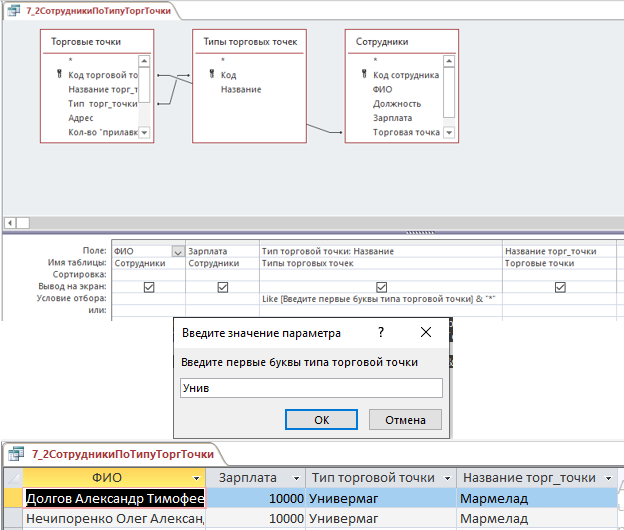
Запрос «7\_2СотрудникиПоТипуТоргТочки» (рис. 33) предназначен для просмотра сотрудников торговых точек указанного типа. 

Рис. 33 *Запрос 7\_2СотрудникиПоТипуТоргТочки в режиме конструктора и просмотра*

Sql инструкция для данного запроса:

PARAMETERS [Введите первые буквы типа торговой точки] Text ( 255 );

SELECT Сотрудники.ФИО, Сотрудники.Зарплата, [Типы торговых точек].Название AS [Тип торговой точки], [Торговые точки].[Название торг\_точки]

FROM ([Типы торговых точек] INNER JOIN [Торговые точки] ON [Типы торговых точек].Код = [Торговые точки].[Тип торг\_точки]) INNER JOIN Сотрудники ON [Торговые точки].[Код торговой точки] = Сотрудники.[Торговая точка]

WHERE ((([Типы торговых точек].Название) Like [Введите первые буквы типа торговой точки] & "\*"));

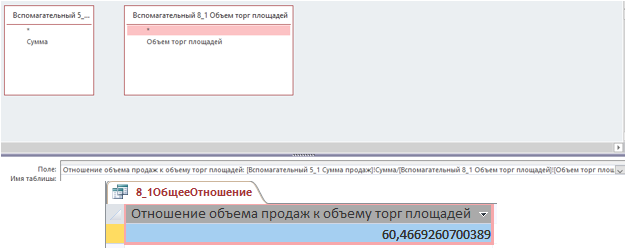
Запрос «8\_1ОбщееОтношение» (рис.34) предназначен для нахождения отношения общего объема продаж к сумме торговых площадей. 

Рис. 34 *Запрос 8\_1ОбщееОтношение в режиме конструктора и просмотра*

Sql инструкция для данного запроса:

SELECT [Вспомагательный 5\_1 Сумма продаж]!Сумма/[Вспомагательный 8\_1 Объем торг площадей]![Объем торг площадей] AS [Отношение объема продаж к объему торг площадей]

FROM [Вспомагательный 5\_1 Сумма продаж], [Вспомагательный 8\_1 Объем торг площадей];

Sql инструкция для вспомогательного запроса:

SELECT DISTINCTROW Sum([Торговые точки].[Площадь торг\_точки]) AS [Объем торг площадей]

FROM [Торговые точки];

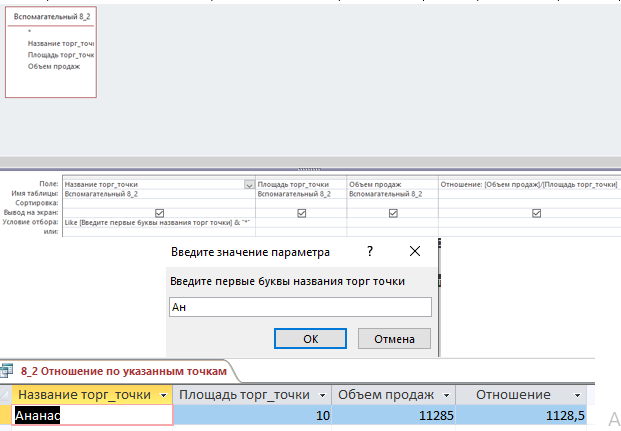
Запрос «8\_2 Отношение по указанным точкам» (рис.35) предназначен для нахождения отношения объема продаж к объему торговых площадей для указанной торговой точки.

Рис. 35 *Запрос 8\_2 Отношение по указанным точкам в режиме конструктора и просмотра*

Sql инструкция для данного запроса:

PARAMETERS [Введите первые буквы названия торг точки] Text ( 255 );

SELECT [Вспомагательный 8\_2].[Название торг\_точки], [Вспомагательный 8\_2].[Площадь торг\_точки], [Вспомагательный 8\_2].[Объем продаж], [Объем продаж]/[Площадь торг\_точки] AS Отношение

FROM [Вспомагательный 8\_2]

WHERE ((([Вспомагательный 8\_2].[Название торг\_точки]) Like [Введите первые буквы названия торг точки] & "\*"));

Sql инструкция для вспомогательного запроса:

SELECT [Торговые точки].[Название торг\_точки], [Торговые точки].[Площадь торг\_точки], Sum([Цена]\*[Величина]) AS [Объем продаж]

FROM [Торговые точки] INNER JOIN (Продажи INNER JOIN [Проданные товары] ON Продажи.[Код продажи] = [Проданные товары].[Код продажи]) ON [Торговые точки].[Код торговой точки] = Продажи.[Торговая точка]

GROUP BY [Торговые точки].[Название торг\_точки], [Торговые точки].[Площадь торг\_точки];

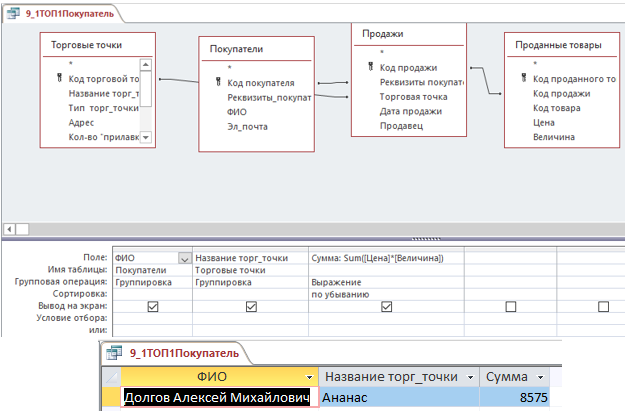
Запрос «9\_1ТОП1Покупатель» (рис.36) предназначен для определения самого активного покупателя для торговой организации.

Рис. 36 *Запрос 9\_1ТОП1Покупатель в режиме конструктора и просмотра*

Sql инструкция для данного запроса:

SELECT TOP 1 Покупатели.ФИО, [Торговые точки].[Название торг\_точки], Sum([Цена]\*[Величина]) AS Сумма

FROM ([Торговые точки] INNER JOIN (Покупатели INNER JOIN Продажи ON Покупатели.[Код покупателя] = Продажи.[Реквизиты покупателя]) ON [Торговые точки].[Код торговой точки] = Продажи.[Торговая точка]) INNER JOIN [Проданные товары] ON Продажи.[Код продажи] = [Проданные товары].[Код продажи]

GROUP BY Покупатели.ФИО, [Торговые точки].[Название торг\_точки]

ORDER BY Sum([Цена]\*[Величина]) DESC;

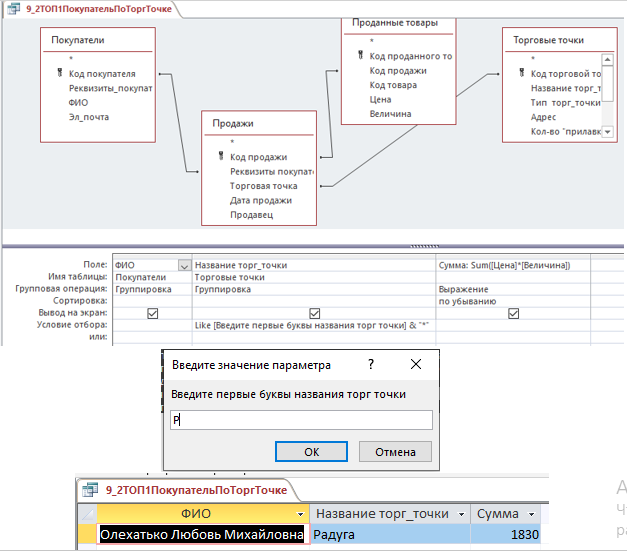
Запрос «9\_2ТОП1ПокупательПоТоргТочке» (рис.37) предназначен для определения самого активного покупателя для указанной торговой точки. 

Рис. 37 *Запрос 9\_2ТОП1ПокупательПоТоргТочке в режиме конструктора и просмотра*

Sql инструкция для данного запроса:

PARAMETERS [Введите первые буквы названия торг точки] Text ( 255 );

SELECT TOP 1 Покупатели.ФИО, [Торговые точки].[Название торг\_точки], Sum([Цена]\*[Величина]) AS Сумма

FROM [Торговые точки] INNER JOIN ((Покупатели INNER JOIN Продажи ON Покупатели.[Код покупателя] = Продажи.[Реквизиты покупателя]) INNER JOIN [Проданные товары] ON Продажи.[Код продажи] = [Проданные товары].[Код продажи]) ON [Торговые точки].[Код торговой точки] = Продажи.[Торговая точка]

GROUP BY Покупатели.ФИО, [Торговые точки].[Название торг\_точки]

HAVING ((([Торговые точки].[Название торг\_точки]) Like [Введите первые буквы названия торг точки] & "\*"))

ORDER BY Sum([Цена]\*[Величина]) DESC;

Для данной базы данных были созданы все запросы, соответствующие описанию предметной области.

1. ОТЧЕТЫ

Отчет - это объект базы данных, который предназначен для вывода информации из баз данных, прежде всего для печати[2-4].

Требуется создать отчеты для заказанных и проданных товаров.

Отчет Заказы товаров (рис.38 и рис. 39) создан для получения сгруппированных данных заказов по каждому товару.

Источник записей: SELECT [Товары в заказе].[Поставщик], [Товары в заказе].[Количество], [Товары в заказе].[Цена], [Товары в заказе].[Код заказа], [Справочник товаров].[Название товара], [Справочник товаров].[Код товара] FROM [Справочник товаров] INNER JOIN [Товары в заказе] ON [Справочник товаров].[Код товара] =[Товары в заказе].[Название товара].

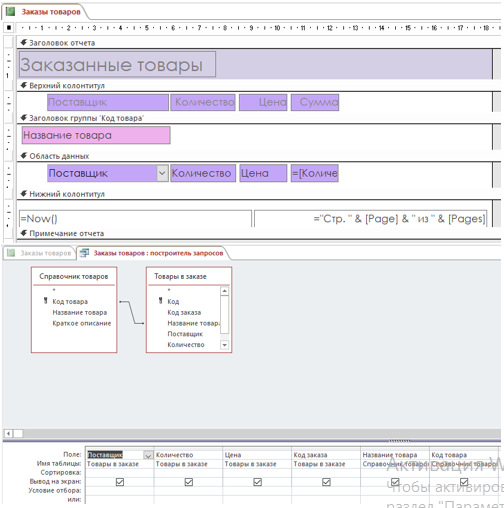
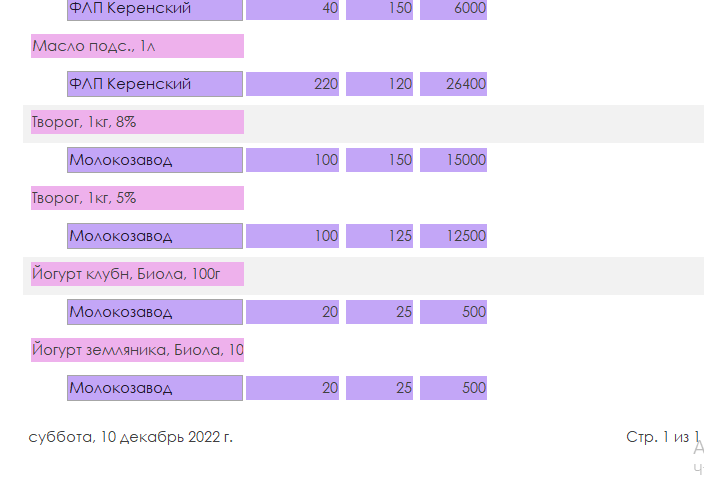
Группировка по названию товара. Ориентация книжная. Ступенчатый макет.

Рис. 38 *Отчет Заказы товаров в режиме конструктора*

**Рис. 39 *Результат работы отчета*

Отчет Продажи товаров (рис.40 и рис.41) создан для получения сгруппированных данных о продажах компанией каждого вида товара.

Источник записей: SELECT [Справочник товаров].[Название товара], [Проданные товары].[Код продажи], [Проданные товары].[Цена], [Проданные товары].[Величина], [Справочник товаров].[Код товара] FROM [Справочник товаров] INNER JOIN [Проданные товары] ON [Справочник товаров].[Код товара] =[Проданные товары].[Код товара];

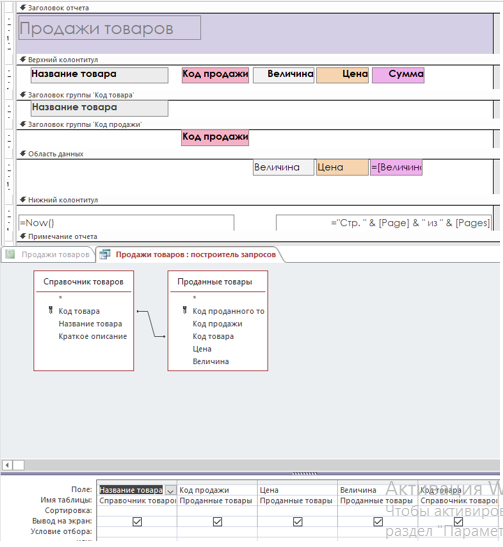
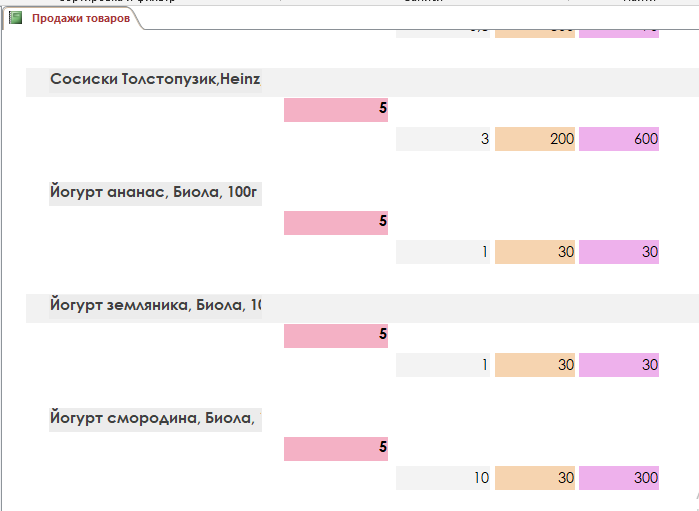
Группировка по названию товара и коду продажи. Ориентация книжная. Ступенчатый макет. Поле Сумма имеет следующие данные: =[Величина]\*[Цена]

Рис. 40 *Отчет Заказы товаров в режиме конструктора*

**Рис. 41 *Результат работы отчета*

Мы создали 2 отчета для отслеживания заказанных и проданных товаров.

ЗАКЛЮЧЕНИЕ

Цель и задачи курсовой были выполнены: разработана база данных для торговой организации.

Все задачи, которые требовалось решить для достижения цели были решены:

• описана предметная область;

• разработана схема данных;

• составлены таблицы;

• созданы формы для работы с базой данных;

• созданы и внедрены кнопками в формы запросы, представленные в предметной области;

• сделаны отчеты;

• создана кнопочная форма и закончено оформление базы данных..

СУБД позволяет получать данные о товарах, продажах, поставках, о работающих сотрудниках.

В ходе выполнения курсовой была изучена программа Microsoft Access и все её функции.

ЛИТЕРАТУРА

1. [Электронный ресурс]. – Режим доступа: <https://infopedia.su/13xfd5e.html>
2. Кузнецов, С. Д. Основы баз данных / С.Д. Кузнецов. - М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2017. - 488 c.
3. Стружкин, Н. П. Базы данных. Проектирование. Учебник / Н.П. Стружкин, В.В. Годин. - М.: Юрайт, 2016. - 478 c.
4. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли. - М.: Вильямс И.Д., 2017. - 1440