



# **КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

**Лабораторный практикум  
для студентов экономических специальностей  
В двух частях. Часть II**

**Составители: Вардомацкая Е.Ю., Багрецова Н.А.**

**Витебск  
2011**



## ПРЕДИСЛОВИЕ

Настоящее издание представляет собой лабораторный практикум по второй части дисциплины “Компьютерные информационные технологии” для студентов экономических специальностей дневной формы обучения.

Вторая часть курса “Компьютерные информационные технологии” (КИТ) является промежуточной в 3-х семестровом цикле подготовки студентов экономических специальностей в области использования информационных технологий в профессиональной деятельности. В этой части дисциплины изучаются технологии структуризации, хранения, и обработки экономической информации средствами баз данных. Получаемые студентами знания систематизируются в соответствие с требованиями времени и действующими стандартами. В результате изучения этой части дисциплины студент должен научиться представлять экономическую информацию в виде баз и хранилищ данных, проектировать, создавать, использовать базы данных.

Настоящий лабораторный практикум содержит 17 лабораторных работ. В методических указаниях к каждой лабораторной работе кратко изложен теоретический материал, разобраны типовые примеры и представлены варианты заданий для самостоятельной работы. Отдельными разделами практикума выделены вопросы для подготовки к тестовому контролю знаний и варианты заданий для проектирования индивидуальных баз данных.

Практикум полностью соответствует утвержденной учебной программе курса «Компьютерные информационные технологии» № ТД-258 /тип., 2007 г. для экономических специальностей высших учебных заведений и может быть использован при проведении лабораторных работ по соответствующим темам курса на дневном и на заочном отделении. Кроме того, данный практикум может быть использован студентами всех специальностей, магистрантами, аспирантами, как при изучении дисциплин информационного направления, так и для самоподготовки.

Авторы-составители считают своим долгом высказать глубокую благодарность и признательность студентам экономического факультета *Алаховой В., Видус Ю., Гуделевой Н., Зецманис А., Ивановой Я., Ильясовой Н., Коновалюк Е., Симоненко В., Тодосиевой Н., Тодрик А., Пушкаревой Е., Швындиковой Е.*, оказавшим неоценимую помощь в процессе подготовки и отладки учебных примеров лабораторных работ.

## Лабораторная работа № 1

# ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ С РЕЛЯЦИОННОЙ СУБД MS ACCESS

**Цель работы:** получить начальные навыки работы с СУБД Access; приобрести опыт модификации имеющейся базы данных (БД).

### Порядок выполнения работы

1. Запустить систему управления реляционными базами данных MS Access.
2. Открыть учебную базу данных "Борей".
3. Изучить состав и количество объектов СУБД MS Access в окне *БД Борей* (таблицы, запросы, отчеты, формы, макросы, модули).
4. Используя справку MS Access (раздел "Введение в MS Access", подразделы "Общие сведения"), изучить назначение объектов MS Access.
5. Используя справку по учебной базе "Борей", изучить назначение учебной БД, состав и содержание объектов.
6. Просмотреть свойства объектов (*Вид/Свойства*). Упорядочить таблицы по дате создания, дате изменения (*Вид/Упорядочить значки*).
7. Изучить структуру БД "Борей": таблицы и их взаимосвязи (*Сервис/Схема данных*).
8. Определить количество записей в таблице "Сотрудники" (в режиме таблицы).
9. Добавить новую запись в таблицу "Сотрудники":
  - в режиме таблицы
  - с использованием формы "Сотрудники".
10. Для таблицы "Сотрудники" в режиме таблицы выполнить скрывание/показ полей *Должность*, *Обращение*.
11. Выполнить закрепление поля *Фамилия*. Отобразить на экране поля *Примечание*, *Подчиняется* (используя *полосы прокрутки* или *< Tab >*).
12. Вернуться к началу таблицы. Отменить закрепление поля *Фамилия*.
13. Поменять местами поля *Фамилия* и *Код сотрудника*. Поместить поле *Обращение* после поля *Код сотрудника*.
14. Продемонстрировать изменение ширины полей и высоты строк открытой таблицы. Закрыть таблицу *Сотрудники* без сохранения изменений макета.
15. Удалить добавленную запись в режиме таблицы и режиме формы.
16. Открыть таблицу *Сотрудники* в режиме конструктора. Изучить окно Конструктора.
17. Завершить работу с БД *Борей*.

## Лабораторная работа № 2

### ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

**Цель работы:** создать информационно-логическую модель предметной области; создать логическую модель базы данных.

#### Порядок выполнения работы

**Задание:** спроектировать реляционную базу данных, описывающую предметную область Банк и включающую сведения о видах услуг, предоставляемых банком: клиентах, открытых счетах, менеджерах и др.

**Этап 1. Концептуальное проектирование** - анализ предметной области "БАНК". В результате анализа можно выявить следующие документы-источники данных для создания базы данных:

- документ 1 "Клиенты"
- документ 2 "Менеджеры"
- документ 3 "Счета"
- документ 4 "Клиенты-счета"
- Документ 5 «Информация по вкладам»

#### Документ 1 "Клиенты"

Код клиента	ФИО клиента	Адрес
A(5)	A(25)	A(35)

Для реквизитов документов указаны форматы их значений. Формат A(5) означает, что значения алфавитно-цифровые, содержащие максимум 5 символов, далее форматы A(25) и A(35) аналогично.

#### Документ 2 "Менеджеры"

№ менеджера	ФИО менеджера	№ филиала	Комиссионные %
A(5)	A(20)	A(4)	5(2)

Формат 5(2) указывает на то, что значения числовые, десятичные, максимум двузначные.

#### Документ 3 "Счета"

Вид счета	% ставка	Срок месяц	Код счета
-----------	----------	------------	-----------

A(30)	5(2)	A(3)	A(5)
-------	------	------	------

#### Документ 4 "Клиенты-счета"

№ счета	Код клиента	Код счета	Дата открытия счета	Сумма на счете	Вид валюты	Начисления в месяц	№ менеджера
A(целое)	A(5)	A(5)	Дата/время	12(2)	A(10)	12(2)	A(5)

#### Документ 5 «Информация по вкладам»

Вклад	Срок размещения, мес.	% ставка (годовых)	Минимальная сумма, ден. ед.	Возможность досрочного изъятия денег, дней.
Валютный депозит	6	10	500 у.е.	60
Валютный срочный	12	12	100 у.е.	-
Валютный текущий	12	3	1 у.е.	30
Рублевый депозит	36	22	500000 бел.р.	360
Рублевый срочный	6	25	200000 бел.р.	90
Рублевый текущий	12	10	100 бел.р.	30

На основании анализа документов выделяются информационные объекты. Для каждого объекта определяется ключевой реквизит. Ключевой реквизит однозначно идентифицирует экземпляры объекта.

Например, реквизит "Код клиента" в Информационном объекте «Клиенты» однозначно идентифицирует определенного клиента.

Таблица 1. Информационные объекты предметной области

Информационный объект	Наименование реквизита	Имя реквизита
Клиенты	Код клиента	Код клиента
	ФИО клиента	ФИО клиента
	Адрес	Адрес
Менеджеры	№ менеджера	№ мен
	ФИО менеджера	ФИО мен
	№ филиала	№ филиала
	Комиссионные %	Комисс %
Счета	Вид счета	Вид счета
	% ставка	% ставка
	Срок месяц	Срок мес
	Код счета	Код счета
Клиенты-счета	№ счета	№ счета
	Код клиента	Код клиента

	<b>Код счета</b>	<b>Код счета</b>
	<b>Дата открытия счета</b>	<b>Дата откp сч</b>
	<b>Сумма на счете</b>	<b>Сумма на сч</b>
	<b>Вид валюты</b>	<b>Вид валюты</b>
	<b>Начисления в месяц</b>	<b>Начисл в мес</b>
	<b>№ менеджера</b>	<b>№ мен</b>

В таблице ключевые реквизиты выделены жирно и подчеркнуты.

Затем определяются связи между информационными объектами. Связь устанавливается между двумя информационными объектами. Наличие связи и ее тип обуславливаются природой реальных объектов, процессов, явлений, отображаемых информационными объектами. Как известно, различают связи следующих типов:

- один к одному (1:1)
- один ко многим (1:M)
- многие ко многим (M:M) – (см. таблицу 2).

*Таблица 2. Типы связей между информационными объектами*

<b>Связь</b>	<b>Ключ связи</b>	<b>Тип связи</b>	<b>Пояснения</b>
КЛИЕНТЫ- Клиенты-Счета	Код клиента	1:M	У одного клиента может быть открыто несколько счетов разных видов на разную сумму, он может быть обслужен разными менеджерами.
СЧЕТА- Клиенты-Счета	Код счета	1: M	Счета одного вида могут быть открыты несколькими клиентами; при открытии определённого счета, разных клиентов могут обслужить разные менеджеры.
МЕНЕДЖЕРЫ- Клиенты-Счета	№ мен	1: M	Определенный менеджер может обслужить разных клиентов, открыть различные счета в разные дни.

**Этап 2. Логическое проектирование** - построение информационно-логической модели предметной области. Ее графическое представление приведено на рис. 1.

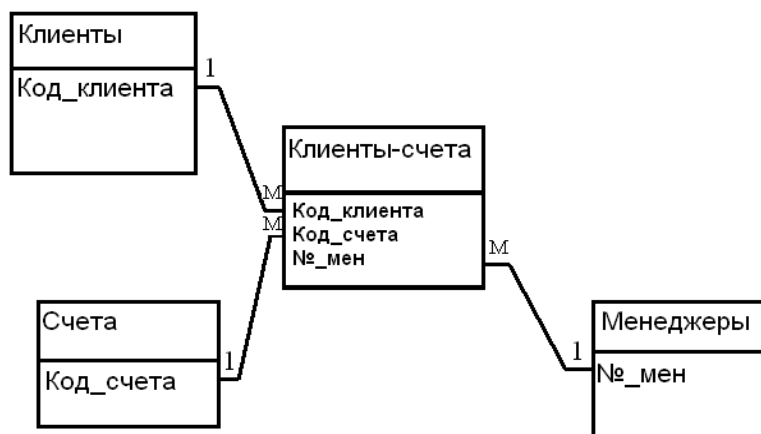


Рис.1. Информационно-логическая модель предметной области Банк.

На этом этапе выбирается СУБД для создания базы данных и информационно-логическую модель предметной области следует отобразить в виде логической модели, основанной на структурных единицах той базы, которая создается выбранной СУБД. Так как в нашем примере выбирается реляционная СУБД Access, то каждый информационный объект следует представить определенной таблицей и установить связи между таблицами. Графическое изображение логической модели базы данных приведено на рис.2.

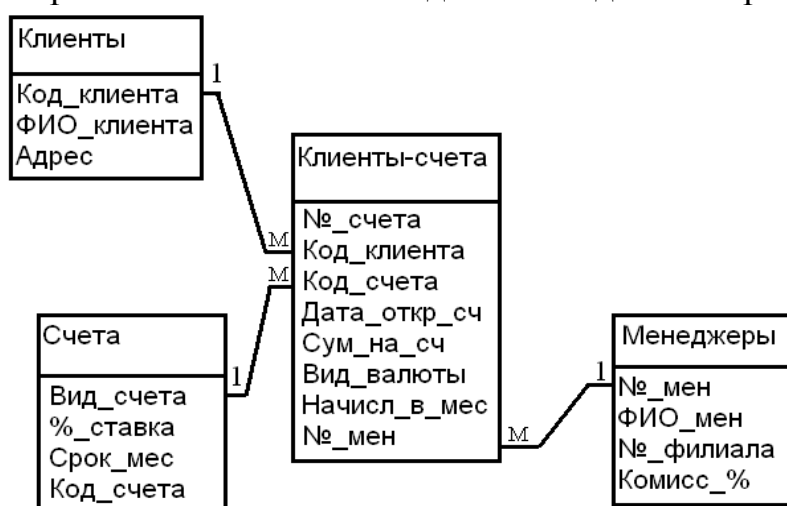


Рис.2. Логическая модель базы данных

Спроектированная база данных является нормализованной, так как ее таблицы-отношения соответствуют требованиям нормализации.

**Этап 3. Физическое проектирование.** На этом этапе база данных создается на внешних носителях информации (см. лабораторную работу №3).

## Лабораторная работа № 3

### СОЗДАНИЕ И КОРРЕКТИРОВКА БАЗЫ ДАННЫХ

**Цель работы:** Освоить принципы работы с таблицами базы данных: создать файл базы данных, описать структуры таблиц в режиме Конструктора, установить связи между таблицами, заполнить таблицы данными, выполнить корректировку базы данных.

#### Порядок выполнения работы

**Задание 1.** Создать файл базы данных с именем **БАНКЪ** в своей папке.

Выполнение задания 1

- Запустите СУБД Access.
- установите в стартовом окне переключатель в положении *Новая база данных* и нажмите [ОК]. Если это окно не отображено, то выполните команду *Файл/Создать* и в появившемся окне *Создание* на вкладке *Общие* произведите двойной щелчок по значку «База данных»;
- в окне *Файл новой базы данных* укажите, на каком диске, в какой папке требуется создать файл базы данных и введите имя файла **БАНКЪ**, а затем нажмите кнопку [Создать].

**Задание 2.** На основании информационно-логической модели, разработанной в лабораторной работе № 2, создать структуру таблиц СЧЕТА, КЛИЕНТЫ, МЕНЕДЖЕРЫ, КЛИЕНТЫ-СЧЕТА базы данных БАНКЪ.

#### Выполнение задания 2

❖ **Таблица КЛИЕНТЫ** (структура представлена в таблице 3.1).

Таблица 3.1. Описание полей таблицы КЛИЕНТЫ

Имя поля	Тип поля	Свойства поля, подлежащие изменению	Устанавливаемое значение поля
Код клиента*	Текстовый	-	5
ФИО клиента	Текстовый	Размер поля	25
		Обязательное поле	Нет
		Индексированное поле	Нет
Адрес	Текстовый	Размер поля	35
		Обязательное поле	Нет
		Индексированное поле	Нет

\* в таблице КЛИЕНТЫ поле *Код клиента* является ключевым согласно логической модели базы данных.



Для описания структуры таблицы в режиме *Конструктора* в окне *База данных* откройте окно *Конструктора таблицы* (ярлык «Создание таблицы в режиме конструктора») или выполните последовательность действий:

окно *База данных* => объект *Таблицы* => [Создать] => окно *Новая таблица* => выбрать *Конструктор* => [ОК].

В окне *Конструктор таблиц* опишите поле *Код клиента* следующим образом:

- введите в столбец «*Имя поля*» текст *Код клиента*;
- щелкните в столбце «*Тип данных*» и выберите из раскрывающегося списка тип этого поля - *Текстовый*. В разделе окна «*Свойства поля*» отобразились свойства, присущие данному типу, и их значения заданы по умолчанию.

*Примечание.* Если значение некоторого свойства не совпадает со значением, принятым по умолчанию, то его можно изменить:

- щелкнув в строке свойства и выбрав из раскрывающегося списка;
- введя с клавиатуры;
- задав с помощью *Построителя выражений*, который вызывается щелчком по кнопке с изображением *трёх точек*;
- введите при желании комментарий к полю в столбце «*Описание поля*», например, *Личный код клиента*
- сделайте поле *Код клиента* ключевым, введя команду *Правка/Ключевое поле*. В результате для поля *Код клиента* будут заданы изменения свойств. Их следует оставить без изменения.

- опишите остальные поля таблицы **КЛИЕНТЫ** в соответствии с таблицей 3.1;

- закройте окно *Конструктора* и на вопрос о сохранении структуры таблицы нажмите кнопку [Да], после чего введите имя таблицы **КЛИЕНТЫ**. В окне *База данных* появится значок таблицы **КЛИЕНТЫ**.

*Примечания.* При необходимости изменения структуры таблицы, таблицу надо открыть в режиме *Конструктора*. Из окна *базы данных* это делается нажатием кнопки [Конструктор], а из окна таблицы – по команде *Вид/Конструктор* или щелчком по кнопке [Вид] панели инструментов.

Аналогично создайте и сохраните структуру следующих таблиц:

❖ **Таблица СЧЕТА** (структура представлена в таблице 3.2).

Таблица 3.2. Описание полей таблицы **СЧЕТА**

Имя поля	Тип поля	Свойство поля, подлежащее изменению	Устанавливаемое значение поля
Вид счета	Текстовый	Размер поля	30
% ставка	Числовой	Размер поля	Байт
Срок мес	Числовой	Размер поля	Байт

Код счета*	Текстовый	Размер поля	5
------------	-----------	-------------	---

\* в таблице СЧЕТА поле Код счета является ключевым согласно логической модели базы данных.

❖ **Таблица МЕНЕДЖЕРЫ** (структура представлена в таблице 3.3).

Таблица 3.3. Описание полей таблицы МЕНЕДЖЕРЫ

Имя поля	Тип поля	Свойство поля, подлежащее изменению	Устанавливаемое значение поля
№ менеджера*	Текстовый	Размер поля	5
ФИО менеджера	Текстовый	Размер поля	20
№ филиала	Числовой	Размер поля	Байт
Комиссионные %	Числовой	Размер поля	Байт

\* в таблице МЕНЕДЖЕРЫ поле № менеджера является ключевым согласно логической модели базы данных.

❖ **таблица КЛИЕНТЫ-СЧЕТА** (см. в таблицу 3.4)

Таблица 3.4. Описание полей таблицы КЛИЕНТЫ-СЧЕТА

Имя поля	Тип поля	Свойства поля, подлежащие изменению	Устанавливаемое значение поля
№ счета	Числовой	Размер поля	Целое
Код клиента	Текстовый	Размер поля	5
		Индексированное поле	Нет
Код счета	Мастер подстановок*	Размер поля	5
		Индексированное поле	Нет
Дата открытия счета	Дата/ Время	Формат поля	Краткий формат даты
Сумма на счете	Числовой	Размер поля	Одинарное с плавающей точкой
Вид валюты	Текстовый	Размер поля	10
Начисления в месяц	Числовой	Размер поля	Одинарное с плавающей точкой
		Число десятичных знаков	1
№ менеджера	Мастер подстановок*	Размер поля	5

Для удобства ввода данных поле *Код счета* опишем как *поле со списком*. Для этого при определении *типа данных* этого поля в режиме *конструктора таблиц* выполним следующие действия:

- В окне *Конструктора таблиц* откроем вкладку *Подстановка*,
- в строке *Тип элемента управления* выберем *Поле со списком* (см. рис. 3.1)

\* Оформить как поле со списком с помощью *мастера подстановок*. *Мастер подстановок* позволяет создать для указанных полей список, из которого выбирается значение.

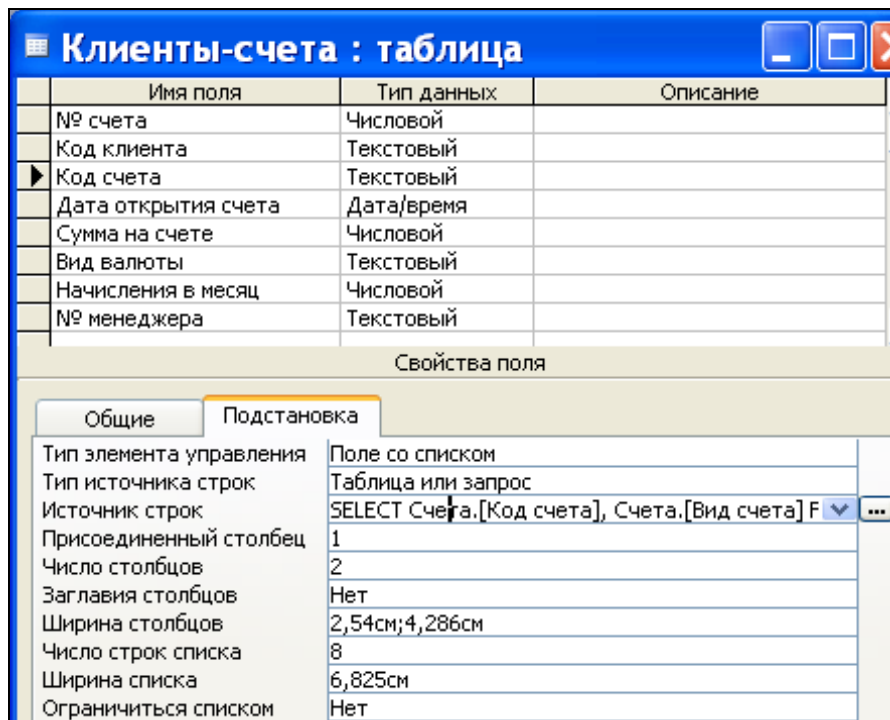


Рис. 3.1 – Создание поля со списком

- в строке *Источник строк* активизируем *Конструктор выражений*,
- автоматически откроется *окно Конструктора запросов*, в котором составим простейший запрос, выбрав из таблицы СЧЕТА поля для подстановки. Это поля *Код счета* и *Вид счета* (см. рис. 3.2). Текст SQL - запроса можно просмотреть, выбрав в меню *Вид режим SQL*:  
 SELECT [Код счета], [Вид счета] FROM Счета;

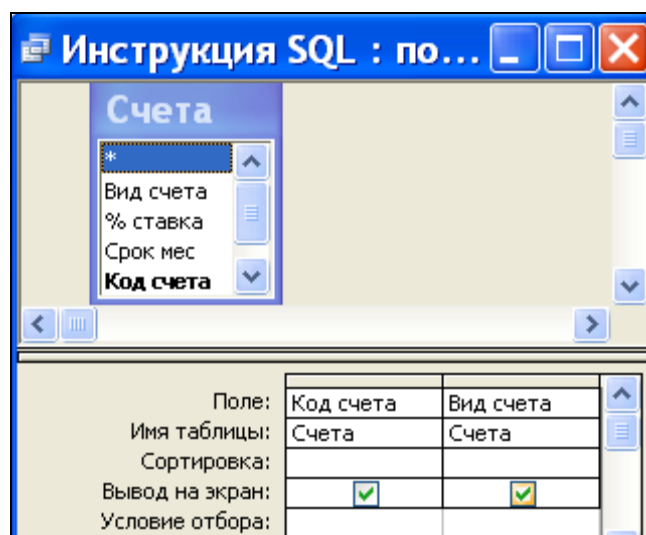


Рис.3.2 – Окно конструктора запросов

- В поле *Присоединенный столбец* введем 1,
- В поле *Число столбцов* введем 2.

- Остальные поля Access заполнит автоматически => ОК.

Теперь при вводе данных в поле *Код счета* можно будет выбирать нужное значение из выпадающего списка, элементами в первого столбца которого будут обозначения *кодов счетов*, второго – *названия счетов*.

**Задание 3.** Установить связи между таблицами КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ, СЧЕТА с обеспечением целостности данных в соответствии с логической моделью базы данных.

*При создании связи между таблицами необходимо учесть следующее:*

**Связь 1:1** создаётся, когда оба связываемых поля имеют уникальные значения. Уникальные значения имеют ключевое поле или поле, для которого задано свойство «Индексированное поле» со значением *Да*(*Совпадения не допускаются*).

**Связь 1:М** создаётся в том случае, когда только одно из связываемых полей является уникальным – ключевым или имеющим свойство «Индексируемое поле» со значением *Да* (*Допускаются совпадения*).

Таким образом,

- связь между таблицами КЛИЕНТЫ и КЛИЕНТЫ-СЧЕТА должна быть установлена по полю *Код клиента* и иметь тип *1:М*;
- связь между таблицами МЕНЕДЖЕРЫ и КЛИЕНТЫ-СЧЕТА должна быть установлена по полю *№ менеджера* и иметь тип *1:М*;
- связь между таблицами СЧЕТА и КЛИЕНТЫ-СЧЕТА должна быть установлена по полю *Код счета* и иметь тип *1:М*.

### **Выполнение задания 3**

Закройте (если не закрыты) таблицы, между которыми устанавливаются связи, и выполните следующее:

- введите команду *Сервис/ Схема данных* или нажмите кнопку [*Схема данных*]. Появится окно *Схема данных* с отображением списков полей таблиц КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ, СЧЕТА;

- введите команду *Связи/ Добавить таблицу*;

- в окне *Добавление таблицы* выделите таблицу КЛИЕНТЫ и нажмите кнопку [*Добавить*], а затем выделите таблицу КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ, СЧЕТА, нажмите [*Добавить*] и [*Заккрыть*];

*Примечание.* Для удаления из окна *Схема данных* лишнего списка полей таблицы, не связанной с другими таблицами, щелкают по нему и нажимают клавишу *Delete*. В случае удаления связанной таблицы прежде удаляют линии связи, а затем список полей таблицы. Для удаления линии связи ее выделяют (она становится жирной), вводят команду *Правка/Удалить* и подтверждают удаление.

- установите связь между таблицами КЛИЕНТЫ и КЛИЕНТЫ-СЧЕТА. Для этого:

- в окне *Схема данных* перетащите ключевое поле *Код клиента* из таблицы *КЛИЕНТЫ* на поле *Код клиента* подчиненной таблицы *КЛИЕНТЫ-СЧЕТА*;

- в окне *Изменение связей* установите флажок «*Обеспечение целостности данных*». Затем установите флажки «*Каскадное обновление связанных полей*» (изменение значения ключа в записи главной таблицы приведет к автоматическому изменению значений внешнего ключа в подсчитанных записях) и «*Каскадное удаление связанных записей*» (удаление записи из главной таблицы приведет к автоматическому удалению всех связанных записей).

Тип связи 1:М между таблицами *КЛИЕНТЫ* и *КЛИЕНТЫ-СЧЕТА* определен автоматически, исходя из произведенного описания поля *Код клиента* в этих таблицах;

- в окне *Изменение связей* нажмите кнопку [*Создать*]. Из окна *Изменение связей* можно открыть с помощью кнопки [*Объединение*] окно *Параметры объединения*, в котором выбрать нужный способ объединения записей связываемых таблиц. В рассматриваемом примере для всех таблиц принимается первый способ, задаваемый по умолчанию, и не требуется изменять параметры объединения.

- установите аналогично связь между таблицами *МЕНЕДЖЕРЫ* и *КЛИЕНТЫ-СЧЕТА* по полю *№\_менеджера*, *СЧЕТА* и *КЛИЕНТЫ-СЧЕТА* по полю *Код счета*.

*Примечание.* Если связь не устанавливается, перейдите в режим конструктора таблицы и проверьте типы данных ключевых полей. В результате схема данных должна иметь вид, представленный на рис. 3.3.



Рис.3.3. Схема данных базы данных Банк

#### Задание 4. Заполнить таблицы данными.

##### **Выполнение задания 4**

- Откройте таблицу КЛИЕНТЫ в режиме таблицы: ==> *Окно базы данных* ==> кнопка [Открыть] или двойной щелчок по значку таблицы. Либо *окно Конструктора таблицы* ==> команда Вид/ Режим таблицы или щелчок по кнопке [Вид] панели инструментов.
- Введите записи таблицы. В числовые и текстовые поля *Код клиента, ФИО клиента, Адрес* введите значения с клавиатуры, используя данные таблицы 3.5.

*Таблица 3.5. КЛИЕНТЫ*

Код клиента	ФИО клиента	Адрес
K1	Артемов	Витебск, Чкалова
K2	Привалов	Витебск, Бровки
K3	Мацкевич	Витебск, Московский
K4	Алексеева	Витебский район
K5	Григорьева	Витебский район
K6	Зайцева	Витебский район
K7	Терентьев	Витебский район
K8	Коробкина	Витебский район
K9	Сергеева	Витебск, Чкалова
K10	Малышева	Витебск
K11	Шитиков	Витебск
K12	Шарай	Витебск
K13	Осипова	Витебский район
K14	Смирнова	Витебский район
K15	Руцкая	Витебск
K16	Шорец	Витебск, Ленина
K17	Соловьева	Витебский район

*Примечание.* Процесс ввода числовых, текстовых данных можно ускорить, если повторяющиеся значения в полях не набирать на клавиатуре, а вводить их из предыдущей записи нажатием комбинации <CTRL><"> (кавычки).

- По завершении заполнения таблицы закройте ее окно.
- Аналогично заполните данными таблицы СЧЕТА, МЕНЕДЖЕРЫ и КЛИЕНТЫ-СЧЕТА на основании значений, приведенных в таблицах 3.6-3.8.

*Таблица 3.6. СЧЕТА*

Вид счета	% ставка	Срок мес	Код счета
Валютный депозит	10	6	ВД
Валютный депозит	12	12	ВС
Валютный текущий	3	12	ВТ
Рублевый депозит	22	36	РД
Рублевый срочный	25	6	РС

Рублевый текущий	10	12	РТ
------------------	----	----	----

**Таблица 3.7. МЕНЕДЖЕРЫ**

№ менеджера	ФИО менеджера	№ филиала	Комиссионные %
М1	Николаев	1	5
М2	Полякова	1	8
М3	Гусева	2	6
М4	Львова	2	8

**Таблица 3.8. КЛИЕНТЫ-СЧЕТА**

№ счета	Код клиента	Код счета	Дата открытия счета	Сумма на счете	Вид валюты	Начисления в месяц	№ менеджера
1001	К1	ВД	01.08.2009	5000			М1
1002	К1	РД	01.06.2009	1000000			М2
1003	К2	РД	01.06.2009	5000000			М4
1004	К3	РС	14.07.2009	10000000			М2
1005	К4	ВС	22.06.2009	10000			М3
1006	К4	ВС	22.06.2009	5000			М3
1007	К5	ВТ	10.05.2009	10000			М4
1008	К5	ВС	20.05.2009	5000			М3
1009	К6	РТ	01.06.2009	25000000			М1
1010	К6	РС	10.06.2009	15000000			М1
1011	К10	ВС	02.09.2009	6000			М2
1012	К11	ВД	05.09.2009	5000			М2
1013	К14	ВД	05.09.2009	8000			М1
1014	К14	РС	05.09.2009	10000000			М1
1015	К7	РС	15.09.2009	5000000			М3
1016	К8	РД	06.09.2009	8000000			М3
1017	К9	ВС	05.07.2009	5000			М4
1018	К12	ВД	25.09.2009	7000			М4
1019	К12	РС	25.09.2009	7000000			М1
1020	К10	ВД	04.09.2009	10000			М4
1021	К17	ВС	05.06.2010	10000			М2
1022	К15	РС	10.06.2010	10000000			М1
1023	К16	ВД	05.06.2010	6000			М4

**Задание 5.** Просмотрите содержимое таблиц базы данных и при необходимости откорректируйте значения записей. При желании дополните новыми значениями таблицы КЛИНТЫ и КЛИЕНТЫ-СЧЕТА (задайте их самостоятельно).

**Задание 6.** На основании Документа 5 «Информация по вкладам» (см. лабораторную работу №2) создайте два текстовых документа: «Информация по рублевым вкладам» и «Информация по валютным вкладам» (см. таблицы 3.9, 3.10). Добавьте в таблицу Счета поле

Информация, содержащимся которого должна быть гиперссылка на соответствующий документ.

**Таблица 3.9. Информация по рублевым вкладам**

Вклад	Срок размещения, мес.	% ставка (годовых)	Минимальная сумма, ден. ед.	Возможность досрочного изъятия денег, дней.
Рублевый депозит	36	22	500000 бел.р.	360
Рублевый срочный	6	25	200000 бел.р.	90
Рублевый текущий	12	10	100 бел.р.	30

Управляющий банком

И.И. Иванов

**Таблица 3.10. Информация по валютным вкладам**

Вклад	Срок размещения, мес.	% ставка (годовых)	Минимальная сумма, ден. ед.	Возможность досрочного изъятия денег, дней.
Валютный депозит	6	10	500 у.е.	60
Валютный срочный	12	10	100 у.е.	-
Валютный текущий	12	3	1 у.е.	30

Управляющий банком

И.И. Иванов



---

---

## Лабораторная работа № 4

### ВЕДЕНИЕ БАЗЫ ДАННЫХ: ПОИСК, ЗАМЕНА И ФИЛЬТРАЦИЯ ДАННЫХ

---

---

**Цель работы:** изучить возможности СУБД ACCESS по поиску, замене и вводу информации в схемы данных, получить навыки работы с таблицами.

#### Порядок выполнения работы

**Задание 1.** Поиск и замена значений в полях таблиц и форм: в таблице **КЛИЕНТЫ** заменить фамилию клиента Мацкевич на Галай.

#### Выполнение задания

1. Откройте таблицу **КЛИЕНТЫ** ⇒ *Правка* ⇒ *Найти* ⇒ в диалоговом окне *Поиск и замена* на вкладке *Поиск* заполните поле ввода *Образец* значением *Мацкевич* ⇒ в раскрывающемся списке *Поиск* выберите в качестве области поиска поле таблицы **ФИО\_клиента**.
2. В раскрывающемся списке *Совпадение* выберите степень совпадения значений – *Поля целиком*.
3. В раскрывающемся списке *Просмотр* выберите направление поиска, которое может принимать значения *Все, Вверх, Вниз* ⇒ *Найти далее*.
4. Для получения каждого следующего значения, удовлетворяющего условию поиска, нажмите кнопку *Найти далее*.
5. Для внесения нового значения в найденное поле выберите в диалоговом окне вкладку *Замена*.
6. В появившейся дополнительно строке *Заменить на* введите новое значение *Галай*.
7. Восстановите исходные значения полей.
8. Нажмите кнопку *Заменить*.
9. Откройте таблицу **СЧЕТА** и замените названия счетов **Валютный текущий** и **Рублевый текущий** на **Валютный накопительный** и **Рублевый накопительный**.
10. Восстановите исходные значения полей.

## Задание 2. Сортировка записей

### **Выполнение задания**

Откройте таблицу КЛИЕНТЫ. Для сортировки по возрастанию или убыванию по любому из полей необходимо установить курсор в соответствующее поле и выполнить команду *Записи* ⇒ *Сортировка* ⇒ *Сортировка по возрастанию* (*Записи* ⇒ *Сортировка* ⇒ *Сортировка по убыванию*). Аналогичным образом сортируются данные по нескольким полям. Предварительно необходимо выделить совокупность полей которые выбраны для сортировки и повторить действия предыдущего пункта.

## Задание 3. Отбор записей по выделенному фрагменту

### **Выполнение задания**

1. Откройте таблицу КЛИЕНТЫ в режиме таблицы. Для отбора всех записей о клиентах, фамилии которых начинаются с заданной буквы (например с буквы «М») следует выделить эту букву в поле ФИО\_клиента одной из записей, где фамилия начинается с этой буквы.

2. Выполните команду меню *Записи* ⇒ *Фильтр* ⇒ *Фильтр по выделенному* или нажмите соответствующую кнопку панели инструментов.

3. Отберите записи о клиентах, проживающих в Витебском районе.

4. Отберите записи о клиентах, проживающих по улице Ленина, фамилии которых начинаются на «Ш» (выполняется в 2 этапа).

5. Для отбора записей не содержащих выделенного значения в поле, после выделения выполните меню *Записи* ⇒ *Фильтр* ⇒ *Исключить выделенное*. Тоже самое можно выполнить щелкнув правой кнопкой мыши в поле выделения и выбрав соответствующую команду в появившемся меню.

6. Выберите всех клиентов проживающих (не проживающих) в *Витебске*.

7. Для работы со всеми записями таблицы, отмените действие фильтра, выполнив команду *Записи* ⇒ *Удалить фильтр*.

8. Для повторного использования фильтра выполните команду *Записи* ⇒ *Применить фильтр*.

## Задание 4. Создание, применение и уничтожение обычного фильтра

### **Выполнение задания**

1. Откройте таблицу КЛИЕНТЫ ⇒ *Записи* ⇒ *Фильтр* ⇒ *Изменить фильтр*.

2. В окне *КЛИЕНТЫ: фильтр* удалите при необходимости ранее использованный фильтр, выполнив команду *Правка* ⇒ *Очистить бланк*.

3. Задайте условие отбора. На внешней вкладке *Найти* в списке Адрес выберите значение «*Витебский район*». Второе значение задайте в поле ФИО клиента - «*С\**» (фамилии клиентов, начинающиеся на букву С).

Выполните фильтрацию таблицы по команде *Фильтр* ⇒ *Применить фильтр*.

4. Для просмотра всех записей таблицы снимите фильтр выполнив команду *Записи* ⇒ *Удалить фильтр*.

5. Для того, чтобы сделать недоступным старый фильтр в окне: *КЛИЕНТЫ фильтр* выполните команду *Правка* ⇒ *Очистить бланк* и затем выполните команду *Записи* ⇒ *Применить фильтр*.

6. Откройте таблицу *КЛИЕНТЫ-СЧЕТА* ⇒ *Записи* ⇒ *Фильтр* ⇒ *Изменить фильтр*.

7. Задайте условие отбора. На внешней вкладке *Найти* в списке Код счета выберите значение «ВД». Перейдите на вкладку *Или* и в списке Код счета выберите значение «ВС». Второе значение задайте в поле «Сумма на счете» - «>5000». Выполните фильтрацию таблицы по команде *Фильтр* ⇒ *Применить фильтр*.

8. Для просмотра всех записей таблицы снимите фильтр выполнив команду *Записи* ⇒ *Удалить фильтр*.

9. Отфильтруйте информацию по вкладкам типа «ВД» за 05.09.2009 и вкладкам типа «ВС» за 22.06.2009.

10. Удалите фильтр.

## **Задание 5. Создание, применение и сохранение расширенного фильтра**

### **Выполнение задания**

1. Откроем таблицу *КЛИЕНТЫ-СЧЕТА* в режиме таблицы ⇒ *Записи* ⇒ *Фильтр* ⇒ *Расширенный фильтр*

2. В окне фильтра перетащите из списка полей таблицы поле *Дата открытия счета* в строку *Поле*. В *строку отбора* введите вопрос, с которым система будет обращаться к пользователю при выполнении фильтрации [*За какую дату?*].

3. Выполнить фильтрацию ⇒ *Фильтр* ⇒ *Применить фильтр*.

4. В диалоговом окне запроса введите значение интересующей Вас даты (например, 25.09.2009) ⇒ ОК.

5. Для сохранения подготовленного фильтра выполните щелчок правой клавишей мыши по окну фильтра и выберите команду *Сохранить как запрос*.

6. В открывшемся диалоговом окне *Сохранение в виде запроса* введите имя сохраняемого фильтра *Фильтр1*.

7. Чтобы уничтожить текущий запрос, необходимо выполнить команду *Записи* ⇒ *Фильтр* ⇒ *Расширенный фильтр*. Чтобы перейти в окно фильтра необходимо выполнить команду *Правка* ⇒ *Очистить бланк* и затем ⇒ *Записи* ⇒ *Применить фильтр*. После этого действия в режиме таблицы команда *Применить фильтр* становится недоступной.

8. Для подключение существующего фильтра к таблице активизируйте *Записи* ⇒ *Фильтр* ⇒ *Расширенный фильтр* и выполните команду *Файл* ⇒ *Загрузить из запроса*. После чего выполните команду *Записи* ⇒ *Применить фильтр*, либо активизируйте запрос *Фильтр1*.

**Задание 6** (выполните самостоятельно)

Используя дополнительную информацию представленную ниже курсивным шрифтом создайте расширенный фильтр к таблице КЛИЕНТЫ-СЧЕТА, чтобы выводилась информация о валютных вклада, сделанных после 01.09.2009 г., суммы на которых превышают 5000 у.е. Результаты сохраните под именем *Фильтр 2*.

## Лабораторная работа № 5

### ЗАПРОСЫ ДЕЙСТВИЯ

**Цель работы:** Изучить возможности СУБД ACCESS по формированию запросов на внесение изменений в базу данных.

#### Теоретические сведения

Для быстрого изменения, вставки, создания или удаления наборов данных из базы данных используются управляющие запросы или запросы-действия. СУБД ACCESS отмечает запросы-действия специальными значками:



— запрос на *добавление* записей в таблицу;



— запрос на *обновление* записей в таблице;



— запрос на *создание* таблицы;



— запрос на *удаление* данных из таблицы.

К командам языка определения данных (DDL) относятся:

- **CREATE** – предназначена для *создания* таблиц. Имя запроса не должно совпадать с именем создаваемой таблицы. Формат:

```
CREATE TABLE <имя таблицы>  
(<имя столбца 1>< тип данных>[(<размер>)]),  
(<имя столбца 2>< тип данных>[(<размер>)]...);
```

- **ALTER** – предназначена для *изменения структуры* таблицы. Формат команды, предназначенный для добавления столбца в таблицу:

```
ALTER TABLE <имя таблицы> ADD < имя столбца >  
<тип данных>[<размер>];
```

- **DROP** – предназначена для *удаления* одной или нескольких таблиц. Формат:

```
DROP TABLE <имя таблицы>;
```

К командам языка манипулирования данными (DML) относятся:

- **INSERT** - позволяет *добавлять строки* в базу данных с помощью ключевого слова **VALUES** или с помощью оператора **SELECT**. Формат:

**INSERT INTO** <имя таблицы>

[(<имя столбца 1>,<имя столбца 2>...<имя столбца>)]

**VALUES** (<значение1>, < значение2>... < значениеN>);

- **UPDATE** – предназначена для изменения существующих в таблице данных. Формат:

**UPDATE** <имя таблицы>

**SET** <имя столбца> = <список полей>

[**WHERE** <предикат>];

- **DELETE** – предназначена для работы с одной или несколькими строками. Формат:

**DELETE FROM** <имя таблицы>

[**WHERE** <предикат>];

Все, что заключено в квадратные скобки [...], является необязательным.

Элемент <предикат> – условие, которое может быть истинным или ложным для каждого столбца или комбинация столбцов из таблицы (таблиц), определенных предложение FROM.

К командам языка управления данными (DCL) относятся:

- **GRANT** – разрешает указанным пользователям выполнять определенные действия;
- **REVOKE** – отбирает у пользователя права, для осуществления операций над объектом.

## Порядок выполнения работы

### Задание 1 (использование оператора CREATE TABLE)

Требуется создать новую таблицу ФИЛИАЛЫ, включающую поля: [№ филиала], [Название филиала], адрес, телефон, [Код банка], УНП, [Описание услуг]. Поля [Название филиала], адрес, телефон имеют символьный тип с размерами 100, 50,20,10 ,10 символов соответственно, поля [№ филиала], [Код банка], УНП имеют целый тип, поле [Оказываемые услуги] - тип MEMO.

Запросы действия в СУБД ACCESS создаются в режиме SQL , для активизации которого следует выполнить следующие действия:

- В окне базы данных выбрать вкладку *Запросы ==> Создать ==> Новый запрос ==> Конструктор => ОК.*
- Появляется диалоговое окно *Добавление таблицы*, которое нужно закрыть, не выбирая никакой таблицы. В результате откроется окно конструктора запроса на выборку.

- Преобразуйте этот запрос на выборку в *запрос на создание таблицы*, выбрав тип запроса *Создание таблицы* (рис. 5.1):

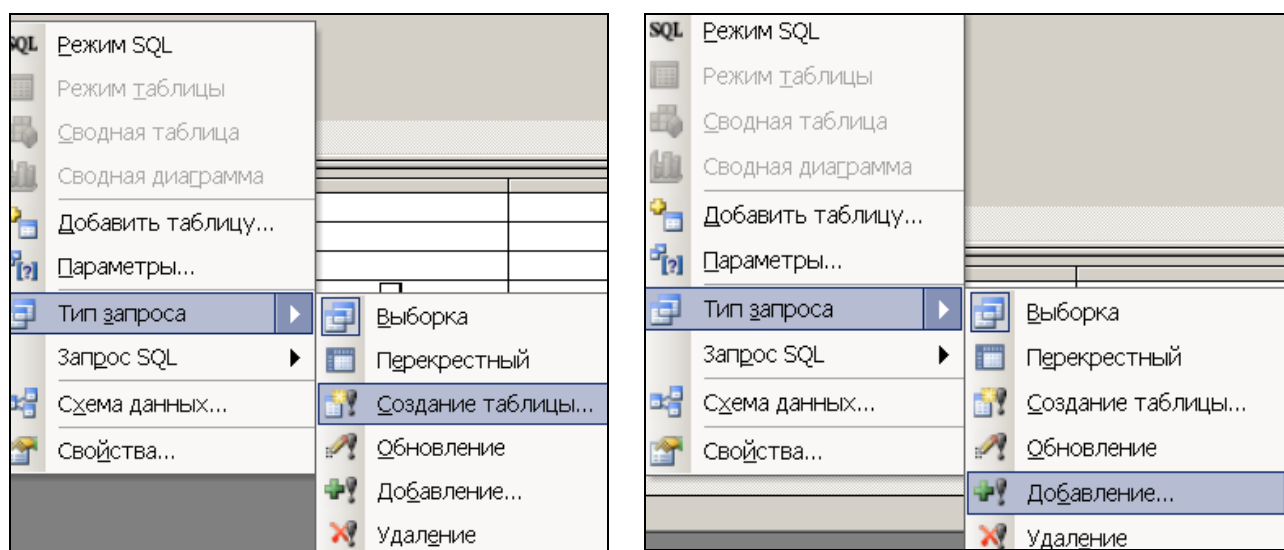


Рис. 5.1 – Выбор типа запроса

- В открывшемся окне *Создание таблицы* задайте имя создаваемой таблицы **ФИЛИАЛЫ**, перейдите в режим SQL и введите инструкцию SQL для этого запроса, которая будет иметь вид:

```
CREATE TABLE Филиалы ([№ филиала] INT,[Название филиала] CHAR(100), адрес CHAR(50), телефон CHAR(20), [Код банка] INT, УНП INT,[Описание услуг] MEMO);
```

- Запустите запрос на выполнение *Запрос ==>Запуск* (или соответствующая кнопка пиктографического меню).

### **Задание 2 (использование оператора INSERT INTO).**

В таблицу **ФИЛИАЛЫ** добавить новую запись.

**№ филиала:** 2

**Название филиала:** ФИЛИАЛ АКЦИОНЕРНОГО БАНКА ГАЗОВОЙ ПРОМЫШЛЕННОСТИ

**адрес:** г.Витебск ул. Ленина 10/3

**телефон:**8-0212-47-87-12

**Код банка:** 150816137

**УНП:** 300229956

**Оказываемые услуги:** прием платежей, операции с ценными бумагами, расчетно-кассовое обслуживание юридических лиц, операции по дорожным чекам, продажа драгоценных металлов, оформление рублёвых пластиковых карточек, выдача наличных по пластиковым карточкам, обмен валюты

Используйте методику создания запроса, описанную в Задании 1, но в этом случае преобразуйте запрос на выборку в *запрос на добавление*, выбрав тип запроса *Добавление* (рис.5.1):

Инструкция SQL для этого запроса будет иметь вид:

```
INSERT INTO Филиалы ([№ филиала], [Название филиала], адрес, телефон, [Код банка], УНП, [Описание услуг])  
VALUES (2, "ФИЛИАЛ АКЦИОНЕРНОГО БАНКА ГАЗОВОЙ ПРОМЫШЛЕННОСТИ", "г.Витебск ул. Ленина 10/3", "8-0212-47-87-12", 150816137, 300229956, "прием платежей, операции с ценными бумагами, расчетно-кассовое обслуживание юридических лиц, операции по дорожным чекам, продажа драгоценных металлов, оформление рублёвых пластиковых карточек, выдача наличных по пластиковым карточкам, обмен валюты");
```

или

```
INSERT INTO Филиалы  
VALUES (2, "ФИЛИАЛ АКЦИОНЕРНОГО БАНКА ГАЗОВОЙ ПРОМЫШЛЕННОСТИ", "г.Витебск ул. Ленина 10/3", "8-2-0212-27-87-12", 150816137, 300229956, "обслуживание юридических и физических лиц");
```

С помощью данного запроса в конец таблицы Филиалы добавляется новая запись, в первое поле которой заносится значение 2, во второе поле – “ФИЛИАЛ АКЦИОНЕРНОГО БАНКА ГАЗОВОЙ ПРОМЫШЛЕННОСТИ”, в третье поле – “ г.Витебск ул. Ленина 10/3”, в четвертое поле – значение "8-2-0212-27-87-12", в пятое и шестое поле соответственно - 150816137, 300229956 и в седьмое поле - "обслуживание юридических и физических лиц"

**Самостоятельно** внесите информацию по филиалу 1:

```
№ филиала: 1  
Название филиала: филиал ОАО «Банк Москва-Минск»  
адрес: г.Витебск ул. Фрунзе 35-1  
телефон:8-0212-49-37-12  
Код банка:150662351  
УНП:300229952  
Оказываемые услуги: прием платежей, операции с ценными бумагами, расчетно-кассовое обслуживание юридических лиц, операции по дорожным чекам, продажа драгоценных металлов, оформление рублёвых пластиковых карточек, выдача наличных по пластиковым карточкам, обмен валюты, денежные переводы
```

### **Задание 3 (использование оператора ALTER TABLE).**

В структуру таблицы КЛИЕНТЫ добавить новое поле *Телефон*, символьного типа размером 20 символов.



### **ALTER TABLE Клиенты ADD Телефон CHAR(20)**

Перейдите в режим *Таблица* и заполните данное поле.

В таблицу СЧЕТА добавить новое поля типа MEMO, для краткого описания вида счета.

### **ALTER TABLE Счета ADD Описание MEMO**

Перейдите в режим *Таблица* и заполните данное поле.

В таблицу *Клиенты* добавить два новых поля типа OLE, для хранения в базе фотографии клиента и образца его подписи.

### **ALTER TABLE Клиенты ADD Фотография IMAGE, Подпись IMAGE**

Перейдите в режим *Таблица* и заполните данные поля.

### **Задание 4 (использование оператора ALTER TABLE)**

Увеличить размер поля *ФИО менеджера* в таблице МЕНЕДЖЕРЫ до 50 СИМВОЛОВ.

### **ALTER TABLE Менеджеры ALTER COLUMN [ФИО менеджера] CHAR(50)**

**Самостоятельно** добавить в таблицу МЕНЕДЖЕРЫ два новых поля типа OLE, для хранения в базе фотографии менеджера и образца его подписи.

### **Задание 5 (использование оператора UPDATE)**

Используйте методику создания запроса, описанную в Задании 1, но в этом случае преобразуйте запрос на выборку в *запрос на обновление*, выбрав тип запроса *Обновление* (рис.5.2):

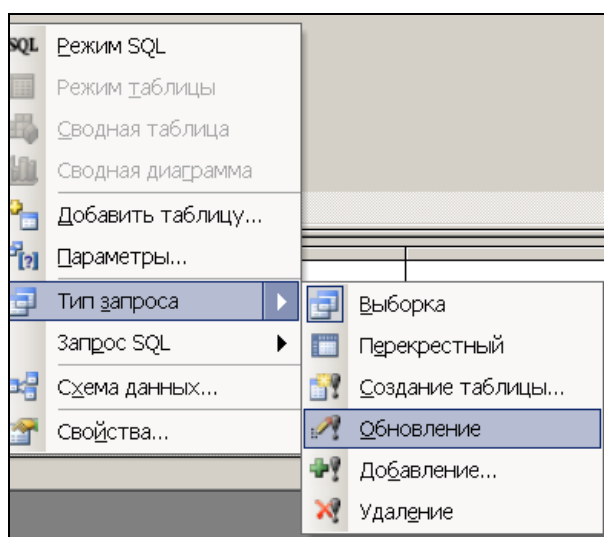


Рис. 5.2 –Выбор типа запроса

Заменить фамилию менеджера *Николаева* полным ФИО.

```
UPDATE Менеджеры SET [ФИО менеджера] = "Николаев Николай Николаевич" WHERE [N° менеджера]="М1";
```

### Задание 6 (использование оператора UPDATE).

Банки решили поднять процентную ставку по всем счетам на 10% по отношению к ранее установленным. Необходимо внести изменения в таблицу СЧЕТА.

```
UPDATE Счета SET [% ставка] = 1.1*[% ставка];
```

### Задание 7 (использование оператора UPDATE).

Заполнить для рублевых счетов поле *Вид валюты* таблицы КЛИЕНТЫ-СЧЕТА значением "бел. р."

```
UPDATE [Клиенты-счета] SET [Вид валюты] = "бел. р." WHERE [Код счета] Like "P*";
```

### Задание 8 (использование оператора UPDATE).

Рассчитать значения поля *[Начисления в месяц]* для счетов всех видов, %-ые ставки по которым размещены в таблице СЧЕТА.

```
UPDATE Счета, [Клиенты-счета] SET [Начисления в месяц]=[% ставка]/100*[Сумма на счете]/12 WHERE счета.[Код счета]=[Клиенты-счета].[Код счета];
```

### Задание 9 (использование оператора DELETE FROM).

Из таблицы ФИЛИАЛЫ удалить сведения по филиалу 1.

```
DELETE FROM Филиалы WHERE [N° филиала]=1
```

Самостоятельно: из таблицы ФИЛИАЛЫ удалите сведения по филиалу 2.

### Задание 10 (использование оператора DROP TABLE).

Удалите таблицу ФИЛИАЛЫ из базы данных.

```
DROP TABLE Филиалы
```

## Задания для самостоятельной работы

### Задание 1

- Сконструировать запрос на создание таблицы КРЕДИТЫ. Эта таблица должна содержать следующие поля:

Название поля	Тип данных	Размер
Код кредита	INT	4
Название	CHAR	100
Описание	MEMO	
% ставка	INT	4
Период	INT	4

- С помощью запроса на добавление заполнить таблицу КРЕДИТЫ следующей информацией:

Код кредита	Название	Описание	% ставка	Период( мес)
2569	Кредит на покупку жилья	Гражданам, нуждающимся в улучшении жилищных условий	15	180
1259	Экспресс-кредит	Сумма кредита - в пределах кредитоспособности с учетом установленного коэффициента от 1 млн. до 10 млн. белорусских рублей.	26	24
3354	Кредит на потребительские нужды	Кредит может быть предоставлен путем выдачи наличных денежных средств, перечислением на текущий или карт-счет заявителя, на счета третьих лиц.	22	36

- Снизить % ставку *Экспресс-кредита* на 1%, а срок кредитования на строительство и покупку жилья увеличить на 60 месяцев.
- В таблицу КРЕДИТЫ добавить новое поле *Вид валюты* символьного типа размером 10 символов. С помощью запроса на обновление заполнить данное поле значением «*бел.р.*»
- Удалить таблицу КРЕДИТЫ из базы *Банкз* с помощью запроса.

## Задание 2

- Сконструировать запрос на создание таблицы ОТДЕЛЫ КРЕДИТОВАНИЯ. Эта таблица должна содержать следующие поля:

Название поля	Тип данных	Размер
Код отдела	INT	4
Название банка	CHAR	20
Телефон	CHAR	10
Адрес	CHAR	50
Начальник отдела	CHAR	50

- С помощью запроса на добавление заполнить таблицу ОТДЕЛЫ КРЕДИТОВАНИЯ, следующей информацией:

Код отдела	Название банка	Телефон	Адрес	Начальник отдела
1111	Белинвестбанк	8-0212-55-55-66	г. Витебск ул.Ленина 12	Иванов Иван Иванович
2222	Белгазпромбанк	8-0212-47-89-96	г. Витебск пр-т Строителей 123	Петров Петр Петрович
3333	Агропромбанк	8-0212-55-77-99	г. Витебск пр-т Московский 64	Сидоров Сидор Сидорович

- В таблицу ОТДЕЛЫ КРЕДИТОВАНИЯ добавить новое поле *Описание услуг* типа MEMO. С помощью запроса на обновление заполнить данное поле значением «*Выдача кредитов физическим и юридическим лицам, операции с пластиковыми карточками*».
- Удалить таблицу ОТДЕЛЫ КРЕДИТОВАНИЯ из базы *Банк* с помощью запроса.

### Задание 3

- Сконструировать запрос на создание таблицы ВКЛАДЫ. Эта таблица должна содержать следующие поля:

<i>Название поля</i>	<i>Тип данных</i>	<i>Размер</i>
Код вклада	INT	4
Название	CHAR	50
Период	INT	4
% ставка	INT	4
Выплата дохода	MEMO	

- С помощью запроса на добавление заполнить таблицу ВКЛАДЫ следующей информацией:

Код вклада	Название вклада	Период	% ставка	Выплата дохода
1564	Гарантированный доход	36	15	По окончании срока возврата вклада. Присоединение к остатку вклада при наступлении срока возврата вклада.
8523	Престиж	3	16	В период до наступления срока возврата вклада, капитализация дохода по истечении каждого месяца хранения вклада.
7745	Детский	60	20	До дня наступления срока возврата вклада основной доход присоединяется к остатку вклада по истечении каждого месяца хранения вклада.

- Снизить % ставку вклада «*Престиж*» на 1%, а период вклада «*Детский*» снизить на 5 месяцев.
- В таблицу ВКЛАДЫ добавить новое поле *Вид валюты* символьного типа размером 10 символов. С помощью запроса на обновление заполнить данное поле значением «*бел.рубль*».
- Удалить таблицу ВКЛАДЫ из базы *Банк* с помощью запроса.

---

---

## Лабораторная работа № 6

### ОДНОТАБЛИЧНЫЕ ЗАПРОСЫ НА ВЫБОР ДАННЫХ

---

---

**Цель работы:** изучить возможности СУБД ACCESS по формированию запросов и поиску информации на их основе; получить навыки работы с запросами.

#### Теоретические сведения

Запрос – команда, которая формулируется для СУБД и требует предоставить определённую указанную информацию. Все запросы на выбор данных из базы формируются с помощью одной команды **SELECT**, которая используется в формате:

```
SELECT <что выводится>  
FROM <откуда> [INTO <куда>  
WHERE<условия отбора записей>  
GROUP BY <колонки(поле), по которым осуществляется группировка>  
HAVING<условие группировки записей в одну строку>  
ORDER BY<в каком порядке выводить данные>]
```

Запросы на выбор данных среде СУБД ACCESS можно создавать в режиме конструктора запросов или в режиме SQL.

#### Порядок выполнения работы

**Задание 1.** Создание запросов на выборку с логическими операциями и специальными операторами в условии отбора.

**Запрос 1.** Вывести сведения по тем из менеджеров, которые обслужили валютные срочные счета от 10000 у.е. и выше и валютные депозиты от 8000 у.е. и выше. В результате запроса вывести поля № менеджера, № счета, Код счета, Сумма на счете. Использовать таблицу КЛИЕНТЫ-СЧЕТА.

#### Выполнение задания

1. Для создания запроса в окне базы данных выберите объект *Запросы* ⇒ *Создать*.
2. В окне *Новый запрос* выберите *Конструктор* ⇒ *ОК*. Появляется диалоговое окно *Добавление таблицы*. В диалоговом окне выберите таблицу *КЛИЕНТЫ-СЧЕТА* ⇒ *Добавить*. Выбранная таблица будет

отображена в области схемы данных запроса. Закройте окно *Добавление таблицы*.

3. В окне конструктора перетащите из списка полей таблицы **КЛИЕНТЫ-СЧЕТА** поля *№ менеджера*, *№ счета*, *Код счета*, *Сумма на счете* в столбцы запроса в строку *Поле*.
4. В строке *Вывод на экран* отметьте поля, иначе они не будут видны при выполнении запроса.
5. Сформулированные в задаче условия требуют формирования следующего логического выражения:

([Код счета]="BC" AND [Сумма на счете]>=10000)  
OR ([Код счета]="ВД" AND [Сумма на счете]>=8000).

Текст запроса на языке SQL имеет вид:

```
SELECT [№ менеджера],[№ счета], [Код счета], [Сумма на счете]
FROM [Клиенты-счета]
WHERE ([Код счета]="BC" AND [Сумма на счете]>=10000
      OR [Код счета]="ВД" AND [Сумма на счете]>=8000);
```

6. Запишите условие из первых скобок в соответствующие поля *Код счета* и *Сумма на счете* первой строки *Условия отбора*. Между условиями находящимися в разных полях одной строки выполняется логическая операция *AND*. Условия из вторых скобок запишите в соответствующих полях *Код счета* и *Сумма на счете* второй строки *Условия отбора*. Между условиями записанными в разных строках, выполняется логическая операция *OR*. Бланк запроса 1 представлен на рис. 6.1.

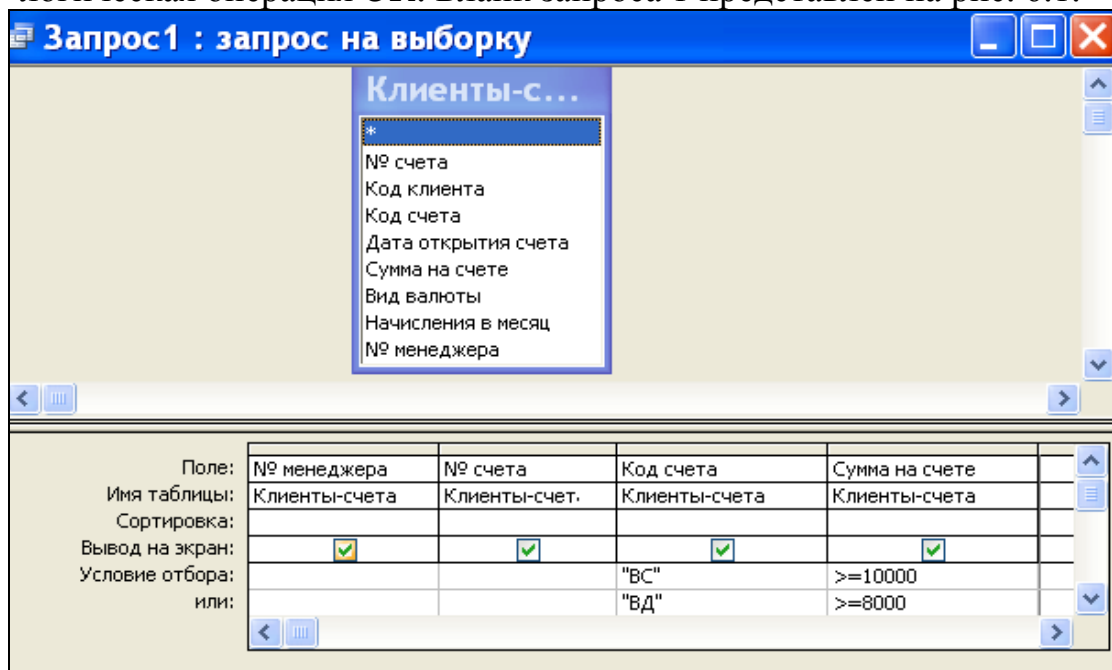


Рис.6.1. Бланк запроса 1

7. В главном меню выполните команду *Запрос* ⇒ *Запуск*. На экране появится окно запроса в режиме таблицы с записями из таблицы **КЛИЕНТЫ-СЧЕТА**, отвечающими заданным условиям отбора.
8. Перейдите из режима конструктора запросов в режим SQL и просмотрите текст запроса 1 на языке SQL. Сравните SQL-запросы.
9. Сохраните запрос 1 по команде меню *Файл* ⇒ *Сохранить* и задав ему имя **Запросб\_1**.
10. Закройте текущий запрос по команде меню *Файл* ⇒ *Закреть*.

**Запрос 2.** Вывести сведения о рублевых счетах, на которых размещены суммы от 10 млн. бел.руб. до 20 млн. бел.руб. В результате запроса вывести поля № счета, Код счета, Сумма на счете. Использовать таблицу **КЛИЕНТЫ-СЧЕТА**.

### Выполнение задания

1. Для составления условия отбора в этом запросе необходимо использовать специальные операторы: оператор подобия **Like** и оператор **Between**.

Текст запроса 2 на языке SQL будет иметь вид:

```
SELECT [№ счета], [Код счета], [Сумма на счете]
FROM [Клиенты-счета]
WHERE [Код счета] Like "P*" AND [Сумма на счете]
Between 10000000 And 20000000;
```

2. Используя методику, описанную выше, заполните бланк запроса для выбора данных в соответствии с условием запроса 2, как показано на рис. 6.2

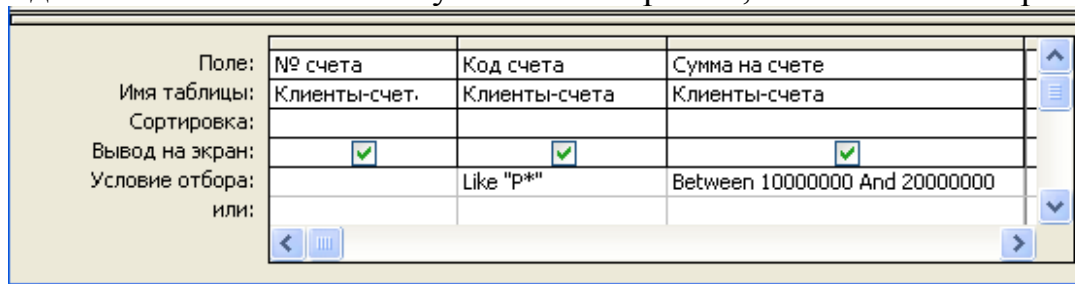


Рис.6.2. Бланк запроса 2

3. Перейдите из режима конструктора запросов в режим SQL и просмотрите текст запроса 2 на языке SQL. Сравните SQL-запросы.
4. Выполните запрос 2 и проанализируйте результат.
5. Сохраните запрос 2 по команде меню *Файл* ⇒ *Сохранить* и задав ему имя **Запросб\_2**.
6. Закройте текущий запрос по команде меню *Файл* ⇒ *Закреть*.

### Задание 2. Использование в условии отбора выражений с именами полей

**Запрос 3.** Пусть необходимо получить информацию по вкладам, начисления в месяц по которым удовлетворяют определенным условиям. Например, требуется вывести список открытых валютных счетов, начисления в месяц по которым составляют 1% и выше и список рублевых счетов, начисления в месяц по которым составляют менее 2%.

По запросу должны отбираться только те записи, в которых значение в поле *Начисления в месяц* удовлетворяет указанным условиям.

### Выполнение задания

1. Для реализации запроса необходимо, прежде всего, составить условие отбора записей:

([Код счета] Like "B\*" AND [Начисления в месяц] $\geq$ 0.01\*[Сумма на счете]) OR ([Код счета] Like "P\*" AND [Начисления в месяц] $<$ 0.02\*[Сумма на счете]);

Текст запроса 3 на языке SQL будет иметь вид:

```
SELECT[№ счета], [Код счета], [Начисления в месяц],[Сумма на счете],
FROM [Клиенты-счета]
WHERE
([Код счета] Like "B*" AND [Начисления в месяц] $\geq$ 0.01*[Сумма на счете]
OR ([Код счета] Like "P*" AND [Начисления в месяц] $<$ 0.02*[Сумма на счете]));
```

2. Используя методику, описанную выше, заполните бланк запроса для выбора данных в соответствии с условием запроса 3, как показано на рис. 6.3

Поле:	№ счета	Код счета	Начисления в месяц	Сумма на счете
Имя таблицы:	Клиенты-сче	Клиенты-счет	Клиенты-счета	Клиенты-счета
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Like "B*"	$\geq$ 0,01*[Сумма на счете]	
или:		Like "P*"	$<$ 0,02*[Сумма на счете]	

Рис.6.3. Бланк запроса 3

3. Перейдите из режима конструктора запросов в режим SQL и просмотрите текст запроса 2 на языке SQL. Сравните SQL-запросы.

4. Выполните запрос и проанализируйте результат.

5. Сохраните запрос 3 по команде меню *Файл*  $\Rightarrow$  *Сохранить* и задав ему имя *Запрос6\_3*.

6. Закройте текущий запрос по команде меню *Файл*  $\Rightarrow$  *Закреть*.

### Задание 3. Создание запроса с вычисляемым полем

**Запрос 4.** Предположим, что необходимо найти записи по валютным вкладам, для которых начисления по процентам за год (за 12 месяцев) составят более 1000 у.е. По запросу должны отбираться только те записи, в которых рассчитанное значение выше указанного в условии.



### Выполнение задания

1. Для решения этой задачи следует рассчитать сумму начислений по процентам, накапливаемую за год (12 месяцев) по каждому валютному вкладу и включить в ответ только те записи, для которых эта сумма будет выше 1000 у.е.

Условие отбора записей в этом случае запишется следующим образом:  
 $12 * [\text{Начисления в месяц}] > 1000 \text{ AND } [\text{Вид валюты}] = \text{"у. е"}$ .

2. Текст запроса 4 на языке SQL будет иметь вид:

```
SELECT [№ счета],[Код счета], [Начисления в месяц], [Вид валюты]
12*[Начисления в месяц]
FROM [Клиенты-счета]
WHERE 12*[Начисления в месяц]>1000 AND [Вид валюты]="у. е";
```

3. Далее следует создать в режиме конструктора запрос на выборку для таблицы Клиенты-счета (либо сразу ввести текст SQL-запроса в режиме SQL):

- Перетащите в бланк запроса поля [№ счета], [Код счета], [Сумма на счете], [Начисления в месяц], [Вид валюты].
- Для подсчета суммы начислений по процентам за год создайте вычисляемое поле в пустой ячейке строки *Поле*, записав выражение:  $12 * [\text{Начисления в месяц}]$ .
- Для отбора искомых записей в строку *Условие отбора* введите условие  $>1000$ , как показано на рисунке 6.4.

Поле:	№ счет	Код счета	Начисления в месяц	Выражение1: 12*[Начисления в месяц]	Вид валюты
Имя таблицы:	Клиент	Клиенты-	Клиенты-счета		Клиенты-сче
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				>1000	"у. е"
или:					

Рис.6.4 – Бланк запроса 4

4. После ввода выражения система по умолчанию формирует имя вычисляемого поля – *выражение 1:*. Это имя ставится перед формулой  $12 * [\text{Начисления в месяц}]$ . Для изменения имени установите курсор в вычисляемом поле бланка запроса и нажмите правую кнопку мыши. В контекстном меню выберите *Свойства* и в строку *Подпись поля* введите новое имя поля – *% за год, ден\_ед.*

5. Для задания точности вычисляемых значений в окне *Свойства* в строке *Формат поля* выберите *Фиксированный*, в строке *Число десятичных знаков* – 2.

6. Сохраните запрос под именем – Запрос 6\_4.

**Запрос 5.** Предположим, необходимо узнать, какая сумма по процентам к концу 2010 года накопится у каждого из вкладчиков, открывших вклады в 2009 году.

## Выполнение задания

1. В этом случае следует рассчитать начисления по процентам для каждого вклада на конец 2010 года. Поскольку вклады размещались в разные периоды (месяцы) года (поле Дата открытия), следует определить количество полных месяцев, начиная с даты открытия, в течение которых каждый вклад находится в банке и умножить это значение на значение поля [Начисления в месяц].

- Количество полных месяцев за 2009 год можно определить, используя функцию Month()<sup>1</sup>: 12-Month([Дата открытия счета]).
- Тогда всего месяцев за 2009 и 2010 год:  
12-Month([Дата открытия счета]) +12 или  
24-Month([Дата открытия счета]).
- Сумма начислений по процентам за период 2009-2010 год составит:  
(24-Month([Дата открытия счета]))\*[Начисления в месяц].

2. Текст запроса 5 на языке SQL будет иметь вид:

```
SELECT [№ счета], [Код счета], [Дата открытия счета], 12-Month([Дата открытия
счета]) + 12 AS Выражение1, [Начисления в месяц], (24-Month([Дата открытия
счета]))*[Начисления в месяц] AS Выражение2, [Вид валюты]
FROM [Клиенты-счета]
WHERE ((([Клиенты-счета].[Дата открытия счета])<=#1/1/2010#))2;
```

3. Используя методику, описанную при создании запроса 4, заполните бланк запроса в соответствии с условием запроса 5, как показано на рис. 6.5.

Дата открытия	Выражение1: 12-Month([Дата открытия счета])+12	Выражение2: (24-Month([Дата открытия счета]))*[Начисления в месяц]
Клиенты-счета		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<=#01.01.2010#		

Рис.6.5 – Фрагмент бланка запроса 5

4. Измените имена вычисляемых полей следующим образом:

- Поле *Выражение 1* – До конца 2010 г., мес.
- Поле *Выражение 2* – % на конец 2010 г., ден\_ед.

5. Задайте точность вычисления поля % на конец 2010 г. равной 2 знакам.

6. Сохраните запрос 5 под именем Запрос 6\_5.

## Задание 4. Параметрические запросы<sup>3</sup>

Параметрические запросы представляют собой варианты базового запроса на выборку, в которых указывается критерий, который может

<sup>1</sup> Для работы с данными типа Дата/время SQL располагает специальными функциями: Day – выделяет число - номер дня месяца введенной даты, Month - выделяет число – номер месяца даты, Year - выделяет число – номер года даты.

<sup>2</sup> Параметр AS вводит имя вычисляемого поля. По умолчанию – Выражение N.

<sup>3</sup> Запросы к заданию 4 разработаны студентами гр. Мк-20 Симоненко В. и Тодосиевой Н.

изменяться по заказу пользователя. Критерий заключается в квадратные скобки и указывается в строке Условие отбора бланка запроса. Иногда условие отбора удобнее записывать непосредственно в режиме SQL .

**Запрос 6.** Вывести сведения по вкладам определенного вида. Код счета (формальный критерий отбора) задавать по желанию пользователя.

### Выполнение задания

1. Создайте обычный запрос на выбор данных в соответствии с условием.
2. В строке *Условие отбора* бланка запроса задайте *формальный критерий*, например текст *[Введите код счета]*. Эта фраза и будет выдаваться в виде приглашения в диалоговом окне при запуске запроса на выполнении. Бланк запроса 6 может иметь вид, как показано на рисунке 6.6:

Поле:	№ счета	Код клиента	Код счета	Дата открытия с-	Сумма на счете	Вид валюты
Имя таблицы:	КЛИЕНТЫ-СЧЕТ	КЛИЕНТЫ-СЧЕТА	КЛИЕНТЫ-СЧЕТА	КЛИЕНТЫ-СЧЕТА	КЛИЕНТЫ-СЧЕТА	КЛИЕНТЫ-СЧЕТА
Сортировка:						
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			[Введите код счета]			
или:						

Рис.6.6 – Бланк запроса 6

3. В меню Запрос выберите подпункт Параметры. В появившемся окне Параметры запроса в столбце Параметр введите то же сообщение, но без квадратных скобок. В столбце Тип данных задайте тип Текстовый, и выполните щелчок по кнопке ОК (рисунок 6.7).

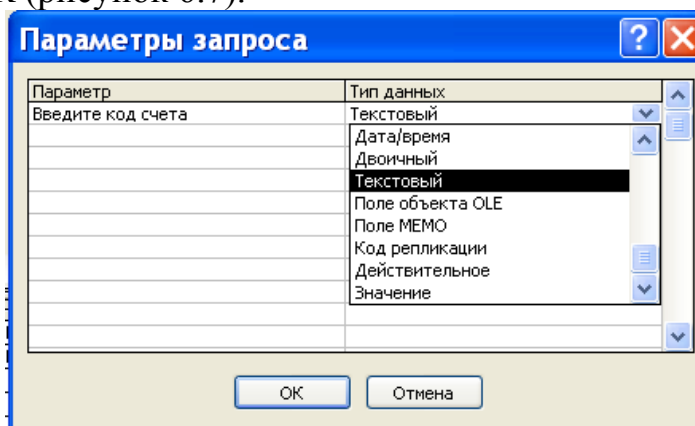


Рис.6.7– Окно Параметры запроса

4. Текст запроса 6 на языке SQL (его можно увидеть, если переключиться в режим SQL) будет иметь вид:

```
PARAMETERS [Ведите код счета] Text (255);
SELECT [№ счета], [Код клиента], [Код счета], [Дата открытия счета], [Сумма на
счете],[Вид валюты]
FROM [КЛИЕНТЫ-СЧЕТА]
WHERE [Код счета]=[Введите код счета]
```

Первая строка PARAMETERS [Введите код счета] Text (255); определяет имя и тип *формального параметра*, указываемого далее в критерии отбора. В тексте запроса эта строка не является обязательной. Если ее удалить, запрос будет обрабатываться корректно.

**5.** Запустите запроса на выполнение. Появится диалоговое окно (рис. 6.8), в которое нужно ввести значение параметра *Код Счета* (например, ВД).

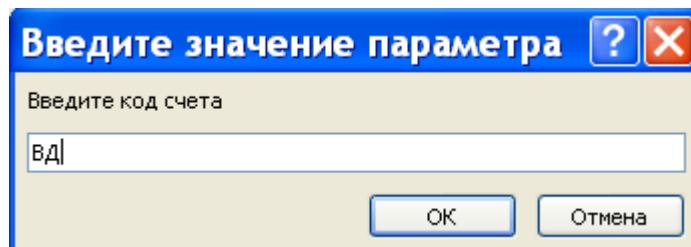


Рис. 6.8 – Диалоговое окно для ввода параметра

В результате будут отобраны все записи с информацией по счетам типа валютный депозит.

**6.** Сохраните запрос под именем Запрос 6\_6.

**Запрос 7.** Вывести список клиентов, проживающих в Витебском районе, фамилии которых начинаются с заданной буквы.

### Выполнение задания

**1.** Составьте условие отбора по полю *ФИО клиента*. Это условие может выглядеть следующим образом:

*Like [Введите первую букву фамилии] + «\*».*

Здесь текст [Введите первую букву фамилии] является *формальным параметром* и будет выдаваться в виде приглашения в диалоговом окне при запуске запроса на выполнении.

**2.** Составьте условие отбора по полю *Адрес*. Это условие отбора может выглядеть следующим образом: *Like «Вит\*район».*

**3.** Создайте обычный запрос на выбор данных в соответствии с условиями отбора, как описано выше. Бланк запроса представлен на рис. 6.9.

Поле:	ФИО клиента	Код клиента	Адрес
Имя таблицы:	Клиенты	Клиенты	Клиенты
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Like [Введите первую букву фамилии]+ "*"		Like "Вит*район"
или:			

Рис. 6.9 – Бланк запроса для запроса 7

**4.** Откройте окно *Параметры запроса* (Запрос ==> Параметры) и задайте *формальный параметр* – текст [Введите первую букву ФИО].

**5.** Просмотрите текст запроса 7 на языке SQL:

```
PARAMETERS [Введите первую букву фамилии] Text ( 255 );
SELECT[ФИО клиента], [Код клиента], Адрес
FROM Клиенты
WHERE [ФИО клиента] Like [Введите первую букву фамилии]+"*"
AND Адрес) Like "Вит*район";
```

6. Запустите запрос на выполнение и отберите сведения о клиентах (например, фамилии которых начинаются на букву Г).

7. Откройте таблицу Клиенты и убедитесь в правильности отбора данных.

8. Сохраните запрос 7 под именем Запрос 6\_7.

### Задание 5. Конструирование запросов с группировкой данных

Часто нужно видеть не каждую строку таблицы, а только итоговые значения по группам данных. Групповые операции позволяют выделить группы записей с одинаковыми значениями в указанных полях и использовать для некоторых полей этих групп одну из статистических функций (*подробнее эта тема рассмотрена в материале лекций и специальной литературе*).

Для группировки данных используется параметр **GROUP BY** в формате: **GROUP BY** <поле, по которому осуществляется группировка>.

**Запрос 8.** Определим количество клиентов, привлеченных банком к открытию счетов каждого вида.

#### Выполнение задания

1. Создайте в режиме конструктора запрос на выборку для таблицы *Клиенты-счета*, включив в него поля *Код счета* и *Код клиента*. По полю *Код счета* будет производиться группировка. По полю *Код клиента* будет вычисляться функция *Count* для подсчета числа клиентов.

2. Выполните команду *Вид* ⇒ *Групповые операции*. В бланке запросов появится строка *Групповые операции*, в которой для обоих полей записано *Группировка*

3. Замените слово *группировка* в столбце *Код клиента* на *Count*. Для этого вызовите список и выберите эту функцию.

4. В строке сортировка задайте сортировку по возрастанию. Бланк запроса 8 представлен на рисунке 6.10.

Поле:	Код счета	Код клиента
Имя таблицы:	Клиенты-счета	Клиенты-счета
Групповая операция:	Группировка	Count
Сортировка:	по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		

Рис. 6.10 – Бланк запроса для запроса 8

5. Перейдите в режим SQL и просмотрите текст запроса:

```
SELECT [Код счета], Count([Код клиента]) AS [Количество клиентов]
```

```
FROM [Клиенты-счета]
GROUP BY [Код счета]
ORDER BY [Код счета];
```

**6.** Выполните запрос *Запрос* ⇒ *Запуск*.

**7.** Замените подпись *Код клиента* на *Количество клиентов*. Для ввода этой подписи в бланке запроса установите курсор мыши на поле *Код клиента* и нажмите правую кнопку. В контекстном меню выберите *Свойства*. В окне *Свойства поля* наберите в строке *Подпись* – *Количество клиентов*.

**8.** Сохраните запрос на выборку под именем *Запрос 6\_8*.

**Запрос 9.** Подвести итоги за каждый день работы банка (количество открытых счетов в валюте, количество открытых счетов бел. рублях, общую сумму средств, принятых банком на хранение в валюте и общую сумму средств, принятых банком на хранение в бел. рублях).

### Выполнение задания

**1.** Создайте в режиме конструктора запрос на выборку для таблицы *Клиенты-счета*, включив в него поля *Дата открытия счета*, *Сумма на счете*, *Код счета*, *Вид валюты*. По полям *Дата открытия счета* и *Вид валюты* будет производиться группировка: сначала по дате, затем по виду валюты за каждую дату. По полю *Сумма на счете* будет вычисляться функция *Sum* для подсчета общей суммы средств, принятых банком на хранение в валюте и в бел. Рублях за каждую дату. По полю *Код счета* будет вычисляться функция *Count* для подсчета количества открытых счетов за каждую дату.

**2.** Выполните команду *Вид* ⇒ *Групповые операции*.

**3.** Замените слово *группировка* в столбце *Сумма на счете* на *Sum*, в столбце *Код счета* на *Count* – вызовите список и выберите эту функцию.

**4.** В строке сортировка для поля *Дата открытия счета* задайте сортировку по возрастанию. Бланк запроса 9 представлен на рисунке 6.11.

Поле:	Дата открытия счета	Сумма на счете	Код счета	Вид валюты
Имя таблицы:	Клиенты-счета	Клиенты-счета	Клиенты-счета	Клиенты-счета
Групповая операция:	Группировка	Sum	Count	Группировка
Сортировка:	по возрастанию			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				

Рис. 6.11 – Бланк запроса для запроса 9

**5.** Перейдите в режим *SQL* и просмотрите текст запроса:

```
SELECT [Дата открытия счета], Sum([Сумма на счете]) AS [Sum-Сумма на
счете], Count([Код счета]) AS [Count-Код счета],[Вид валюты]
FROM [Клиенты-счета]
GROUP BY [Дата открытия счета], [Вид валюты]
ORDER BY [Дата открытия счета];
```

**6.** Выполните запрос *Запрос* ⇒ *Запуск*.

7. Замените подписи *Сумма на счете* на *Итого по счетам* и *Код счета* на *Количество счетов*.
8. Сохраните запрос на выборку под именем Запрос 6\_9.

**Запрос 10.** Подвести итоги по каждому менеджеру.

### Выполнение задания

1. По описанной выше методике создайте запрос на выборку, включив в него нужные поля таблицы *Клиенты-счета*.
2. Задайте группировку данных и выберите функции *Sum*, *Count*, *Min*, *Max*.
3. Задайте сортировку данных по полю *№ менеджера*. Бланк запроса может иметь вид как на рис. 6.12.

№ менеджера	Вид валюты	Сумма на счете	Сумма на счете	Сумма на счете
Клиенты-счета	Клиенты-счет	Клиенты-счета	Клиенты-счета	Клиенты-счета
Группировка	Группировка	Count	Sum	Max
по возрастанию		по убыванию		
✓	✓	✓	✓	✓

Рис. 6.12 – Фрагмент бланка запроса для запроса 10

4. Перейдите в режим SQL и просмотрите текст запроса:

```
SELECT [№ менеджера], [Вид валюты], Count([Сумма на счете]) AS [Count-Сумма на счете], Sum([Сумма на счете]) AS [Sum-Сумма на счете], Max([Сумма на счете]) AS [Max-Сумма на счете], Min([Сумма на счете]) AS [Min-Сумма на счете], Avg([Сумма на счете]) AS [Avg-Сумма на счете]
FROM [Клиенты-счета]
GROUP BY [№ менеджера], [Вид валюты]
ORDER BY [№ менеджера], Count([Сумма на счете]) DESC;
```

5. Замените подписи полей на соответствующие по смыслу.
6. Сохраните запрос на выборку под именем *Итоги по менеджерам*.
7. Замените подписи *Сумма на счете* на *Итого по счетам* и *Код счета* на *Количество счетов*.
8. Сохраните запрос на выборку под именем Запрос 6\_10.

### Задание 6. Задание условий отбора в запросах с групповыми операциями.

Для выбора данных из групп, объединенных общим значением поля, используется параметр **HAVING** в формате:

**HAVING**<условие группировки записей в одну строку>

**Запрос 11.** Вывести итоговую информацию по тем из менеджеров, у которых общая сумма на счетах выше заданной величины (15 000 000 в бел. рублях и 15 000 в валюте).

### Выполнение задания

1. Сначала составьте условие выбора из групп:  
*(Sum([Сумма на счете])>15000000 And [Вид валюты] Like 'б\*') Or (Sum([Сумма на счете])>15000 And [Вид валюты] Like 'у\*')*. Это условие должно использоваться в качестве аргумента параметра *HAVING*.

Тогда текст запроса на SQL будет иметь вид:

```
SELECT [№ менеджера], [Вид валюты], Sum([Сумма на счете]) AS Итого,
Count([Сумма на счете]) AS [Количество счетов],
Max([Сумма на счете]) AS [Max счет], Min([Сумма на счете]) AS [Min счет]
FROM [Клиенты-счета]
GROUP BY [№ менеджера], [Вид валюты]
HAVING (Sum([Сумма на счете])>15000000 And [Вид валюты] Like 'б*')
Or (Sum([Сумма на счете])>15000 And [Вид валюты] Like 'у*');
```

2. Откройте бланк запроса 10.

3. В строке условия отбора добавьте условия выбора из групп: в столбце *Сумма на счете* напишите *>15000000*, в столбце *Вид валюты* этой же строки напишите *б\**. Перейдите в строку *или* бланка запроса и в столбце *Сумма на счете* добавьте условие *>15000*, в столбце *Вид валюты* этой же строки напишите *у\**.

4. Бланк запроса должен иметь вид как на рис. 6.13.

Поле:	№ менеджера	Вид валюты	Сумма на счете	Код счета	Сумма на счете
Имя таблицы:	Клиенты-счета	Клиенты-счета	Клиенты-счета	Клиенты-счет	Клиенты-счета
Групповая операция:	Группировка	Группировка	Sum	Count	Max
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Like 'б*'	>15000000		
или:		Like 'у*'	>15000		

Рис. 6.13 – Фрагмент бланка запроса для запроса 11

5. Выполните запрос, проанализируйте результаты.

6. Измените имена полей в соответствии со смыслом (это можно сделать и в режиме SQL).

7. Перейдите в режим SQL и просмотрите текст запроса.

8. Сохраните запрос под именем Запрос 6\_11.

### Задание 7. Вложение запросов

SQL позволяет вкладывать запросы друг в друга. Сначала выполняется внутренний запрос. Он генерирует значения, которые тестируются на предмет истинности предиката.

**Запрос 12.** Предположим, необходимо получить информацию по валютным счетам, суммы на которых превышают среднее значение валютных вкладов по банку.

### **Выполнение задания**



В этом случае необходимо сформировать подзапрос, в котором вычислить среднее значение валютных вкладов по банку за весь период:

```
SELECT Avg([Сумма на счете])  
FROM [Клиенты-счета] WHERE [Код счета] Like 'B*'
```

затем вложить этот подзапрос в основной запрос, который должен сравнивать значение поля *Сумма на счете* каждой записи с результатом, вычисляемым в подзапросе:

Текст запроса 12 на языке SQL будет иметь вид:

```
SELECT [№ счета], [Сумма на счете], [Вид валюты]  
FROM [Клиенты-счета]  
WHERE [Код счета] like 'B*' And [Сумма на счете]>  
      (SELECT Avg([Сумма на счете]) FROM [Клиенты-счета]  
       WHERE [Код счета] Like 'B*');
```

Бланк запроса составляется обычным способом в режиме конструктора запросов, текст подзапроса записывается в строке *Условие отбора*.

Выполните запрос и сохраните его под именем Запрос 6\_12.

### **Задание 8. Вставка результатов запросов**

С помощью запросов на выбор данных можно получать или выбирать значений из одной таблицы и помещать их в другую. Для этих целей используется команда INSERT (см. лабораторную работу №5), в которой предложение VALUES заменяется на соответствующий запрос.

### **Запрос 13**

Выбрать сведения о счетах типа ВД и поместить их в отдельную таблицу [Счета ВД].

### **Выполнение задания**

**1.** Создайте таблицу [Счета ВД] с помощью конструктора таблиц (методика создания таблиц БД изложена в *лабораторной работе № 3*). Структура этой таблицы должна соответствовать структуре таблицы [Клиенты-счета], т.е. соответствующие столбцы должны иметь одинаковые типы данных, но могут иметь разные имена.

**2.** Составьте SQL-запрос на выбор данных, которые должны быть отражены в таблице [Счета ВД]. Текст такого SQL-запроса может иметь вид:

```
SELECT [Код счета], [№ счета], [Сумма на счете],[Вид валюты],  
[№ менеджера]  
FROM [Клиенты-счета]  
WHERE [Код счета]="ВД";
```

**3.** Составьте SQL-запрос на внесение в таблицу [Счета ВД] записей, соответствующих условиям отбора (т.е. отобранных в результате запроса сформулированного выше в п.2). Текст этого SQL-запроса имеет вид:

```
INSERT INTO [Счета ВД] ([Код счета], [№ счета], [Сумма на счете],  
[Вид валюты],[№ менеджера])  
SELECT [Код счета], [№ счета], [Сумма на счете],[Вид валюты], [№ менеджера]  
FROM [Клиенты-счета]  
WHERE [Код счета]="ВД";
```

4. Войдите в режим конструктора запросов, далее ==> *меню Запрос* ==> *Обновление* ==> задайте в открывшемся окне имя таблицы [Счета ВД] (без скобок) ==> заполните бланк запроса в соответствии с условием или ==> *меню Вид* ==> *режим SQL* ==> внесите текст запроса на языке SQL .

5. Выполните запрос, проанализируйте результат. Убедитесь, что таблица [Счета ВД] заполнена.

6. Сохраните запрос под именем Счета\_ВД.

### Задания для самостоятельной работы

*Рекомендуется:* вначале составить текст запроса на языке SQL, затем заполнить бланк конструктора запросов, выполнить запрос, проанализировать результат, открыть режим SQL и сравнить свой текст SQL–запроса с запросом, сгенерированным СУБД Access.

**Запрос 1.** Вывести список фамилий клиентов в указанном буквенном диапазоне.

**Запрос 2.** Найти записи по валютным вкладам, начисления по которым не равно 1% в месяц от суммы, хранящейся на счете.

По запросу должны отбираться только те записи, в которых значение в поле Начисления в месяц выше или ниже, указанного в условии.

**Запрос 3.** Найти записи по рублевым вкладам, начисления по которым не равно 2% в месяц от суммы, хранящейся на счете.

По запросу должны отбираться только те записи, в которых значение в поле Начисления в месяц выше или ниже, указанного в условии.

*Указание:* для реализации этих запросов следует рассчитать разность между реальными начислениями в месяц по каждому виду вкладов и указанными в условии значениями (1% от суммы на счете по валютным вкладам и 2% от суммы на счете по рублевым вкладам). В ответ включить только те записи, для которых эта разность не равна нулю.

Условие отбора записей в случае запроса 1:

$[Начисления\ в\ месяц] - 0,01 * [Сумма\ на\ счете] <> 0$ , если  $[Вид\ валюты] = y. e$

Условие отбора записей в случае запроса 2:

$[Начисления\ в\ месяц] - 0,02 * [Сумма\ на\ счете] <> 0$ , если  $[Вид\ валюты] = бел. р.$

**Запрос 4.** Найти записи по рублевым вкладам, для которых начисления по процентам за год составят более 1,5млн. бел. руб. Сформировать вычисляемое поле *Начисления по % за год*. По запросу должны отбираться

только те записи, в которых рассчитанное значение выше указанного в условии.

**Запрос 5.** Узнать, какая сумма по процентам к концу 2010 года накопится у каждого из вкладчиков, открывших вклады в 2010 году.

**Запрос 6.** Вывести сведения по вкладам определенного менеджера (*№ менеджера* представить формальным параметром).

**Запрос 7.** Вывести сведения по вкладам, открытым за определенную дату (*Дату открытия счета*).

**Запрос 8.** Вывести сведения по вкладам, за определенный месяц 2009 года. (*Месяц* представить формальным параметром).

**Запрос 9.** Вывести список клиентов, у которых номер телефона начинается с заданного сочетания цифр (*Сочетание цифр*, с которых начинается номер телефона представить формальным параметром).

*Указание.* Предварительно в таблицу *Клиенты* добавить поле *№ телефона* (см. лабораторную работу № 5).

**Запрос 10.** По *№ менеджера* вывести о нем полную информацию (*№ менеджера* представить формальным параметром).

**Запрос 11.** Проанализировать популярность у вкладчиков вкладов разных видов за весь период работы банка. (Подвести итоги по каждому виду вклада):

**Запрос 12.** Подвести итоги по тем менеджерам, у которых общая сумма на рублевых счетах удовлетворяет условию - находится в заданных пределах:

**Запрос 13.** Проанализировать работу банка за каждый месяц 2009г, то есть подвести итоги по валютным и рублевым вкладам за каждый месяц.

**Запрос 14.** Проанализировать работу банка за сентябрь месяц 2009г, то есть подвести итоги по валютным и рублевым вкладам за этот месяц.

**Запрос 15.** Проанализировать работу банка за I квартал (первое полугодие) 2010 года.

**Запрос 16.** Получить информацию по рублевым счетам, суммы на которых являются *максимальными* по банку.

**Запрос 17.** Получить информацию по рублевым счетам, суммы на которых *ниже среднего значения* рублевых счетов, открытых в *летние месяцы 2009 года*.

**Запрос 18.** Получить в одной таблице среднее значение и количество открытых валютных счетов жителями г.Витебска и среднее значение и количество открытых валютных счетов жителями Витебского района. (использовать параметр UNION).

**Запрос 19.** Сформировать таблицу *Итоги* (с полями *[Код счета]*, *[Итого по счетам]*, *[Количество]*, в которой отразить итоговую информацию по счетам каждого вида.

**Запрос 20.** Сформировать таблицу *Итоги* (с полями *[Код счета]*, *[Итого по счетам]*, *[Количество]*, в которой отразить итоговую информацию по счетам каждого вида за каждый месяц 2009 года.

---

## Лабораторная работа № 7

### МНОГОТАБЛИЧНЫЕ ЗАПРОСЫ НА ВЫБОРКУ ДАННЫХ

---

**Цель работы:** Изучить возможности СУБД ACCESS по формированию запросов из нескольких таблиц и поиску информации на их основе. Изучить способы объединения записей из разных таблиц в многотабличных запросах.

#### Порядок выполнения работы

##### Задание 1. Объединение записей в многотабличном запросе

Для любой пары связанных таблиц реляционной базы данных может быть выбран один из трех способов объединения записей:

- **способ 1** – объединение только тех записей, в которых связанные поля обеих таблиц совпадают. В этом случае SQL ядра Access объединяет их, используя операцию *INNER JOIN* как часть параметра *FROM*. Объединение производится при условии равенства полей, приведенных после ключевого слова *ON*, в записях таблиц, указанных в операции *INNER JOIN*. Например,

*FROM Менеджеры INNER JOIN [Клиенты-счета]*

*ON Менеджеры.[№ менеджера] = [Клиенты-счета].[№ менеджера]*

- **способ 2** – объединение тех записей, в которых связанные поля обеих таблиц совпадают, а также объединение всех записей из первой таблицы, для которых нет связанных во второй, с пустой записью второй таблицы. В этом случае SQL ядра Access объединяет их, используя операцию *LEFT JOIN* (левое внешнее объединение) как часть параметра *FROM*. Объединение производится при условии равенства полей, приведенных после ключевого слова *ON*, в записях таблиц, указанных в операции *LEFT JOIN*. Например,

*FROM Менеджеры LEFT JOIN [Клиенты-счета]*

*ON Менеджеры.[№ менеджера] = [Клиенты-счета].[№ менеджера]*

*Указание:* В таблицу **МЕНЕДЖЕРЫ** добавьте менеджера Орлову (код М5) и проверьте объединение на примере любого запроса. Удалите внесенную запись.

- **способ 3** – объединение тех записей, в которых связанные поля обеих таблиц совпадают, а также объединение всех записей из второй таблицы, для которых нет связанных в первой, с пустой записью первой таблицы. В этом случае SQL ядра Access объединяет их, используя операцию *RIGHT JOIN* (правое внешнее объединение) как часть параметра *FROM*. Объединение производится при условии равенства полей, приведенных после ключевого слова *ON*, в записях таблиц, указанных в операции *RIGHT JOIN*. Например,

FROM Менеджеры *RIGHT JOIN* [Клиенты-счета]

ON Менеджеры.[№ менеджера] = [Клиенты-счета].[№ менеджера]

*Указание* В таблицу КЛИЕНТЫ-СЧЕТА добавьте запись, касающуюся менеджера с кодом М6, и проверьте объединение на примере любого запроса. При этом в таблицу МЕНЕДЖЕРЫ менеджера с кодом М6 добавлять не нужно. Удалите внесенную запись.

*SQL ANSI* при любом способе объединения таблиц оформляет объединение как часть условия *WHERE*:

WHERE Менеджеры.[№ менеджера]=[Клиенты-счета].[№ менеджера]

**Задание 2. Конструирование запроса на основе нескольких взаимосвязанных таблиц** (используем запросы, разобранные в лабораторной работе №6)

*Указание:* все запросы выполняются по методике, описанной в лабораторной работе №6, но в бланк запроса добавляются поочередно таблицы, нужные для реализации запроса.

**Запрос 1.** Вывести сведения по тем из менеджеров, которые обслужили валютные срочные счета от 10000 у.е. и выше и валютные депозиты от 8000 у.е. и выше. В результате запроса вывести поля *ФИО менеджера, № счета, Код счета, Сумма на счете*. Использовать таблицу КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ.

### Выполнение задания

1. Составьте запрос на языке *SQL ANSI* и сверьте его с текстом запроса, приведенным ниже:

```
SELECT [ФИО менеджера], [№ счета], [Код счета], [Сумма на счете]
FROM Менеджеры, [Клиенты-счета]
WHERE Менеджеры.[№ менеджера] = [Клиенты-счета].[№ менеджера]
      AND(([Код счета] = "BC" AND [Сумма на счете] >= 10000)
      OR ([Код счета] = "ВД" AND [Сумма на счете] >= 8000));
```

2. Создайте запрос в режиме конструктора запросов: в окне базы данных выберите объект *Запросы* ⇒ *Создать*.
3. В окне *Новый запрос* выберите *Конструктор* ⇒ *ОК*. В диалоговом окне *Добавление таблицы* выберите таблицу МЕНЕДЖЕРЫ ⇒ *Добавить*, затем таблицу КЛИЕНТЫ-СЧЕТА ⇒ *Добавить*. Выбранные таблицы будут отображены в области схемы данных запроса. Закройте окно *Добавление таблицы*.
4. В окне конструктора перетащите из списка полей таблицы МЕНЕДЖЕРЫ поле *ФИО менеджера*, из списка полей таблицы КЛИЕНТЫ-СЧЕТА поля *№ счета, Код счета, Сумма на счете*. в столбцы запроса в строку *Поле*.
5. Бланк запроса 1 представлен на рис. 7.1.

Поле:	ФИО менеджера	№ счета	Код счета	Сумма на счете
Имя таблицы:	Менеджеры	Клиенты-счета	Клиенты-счета	Клиенты-счета
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			"ВС"	>=10000
или:			"ВД"	>=8000

Рис. 7.1 - Бланк запроса 1

6. Перейдите в режим SQL и сверьте текст запроса, сгенерированного ACCESS с текстом запроса, разработанного вами. Обратите внимание на организацию объединения записей.
7. Выполните запрос и проанализируйте результаты.
8. Сохраните запрос под именем Запрос 7\_1.

**Запрос 2.** Получить информацию по вкладам, начисления в месяц по которым удовлетворяют определенным условиям.

Вывести список открытых валютных счетов, начисления в месяц по которым составляют 1% и выше и список рублевых счетов, начисления в месяц по которым составляют менее 2%. Кроме того в результате запроса вывести полное название счета, ФИО и адрес владельца счета. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА, СЧЕТА.

### Выполнение задания

1. Текст запроса на языке SQL ANSI:

```
SELECT [№ счета], [Вид счета], [Начисления в месяц], [Сумма на счете],
      [ФИО клиента], Адрес
FROM Счета, Клиенты, [Клиенты-счета]
WHERE Клиенты.[Код клиента] = [Клиенты-счета].[Код клиента]
      AND Счета.[Код счета] = [Клиенты-счета].[Код счета]      AND
      (([Вид счета] Like "B*" AND [Начисления в месяц] >= 0.01*[Сумма на счете]) OR
      ([Вид счета] Like "P*" AND [Начисления в месяц] < 0.02*[Сумма на счете]));
```

2. Фрагмент бланк запроса 2 представлен на рис. 7.2.

Поле:	№ счета	Вид счета	Начисления в месяц
Имя таблицы:	Клиенты-счет	Счета	Клиенты-счета
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Like "B*"	>=0,01*[Сумма на счете]
или:		Like "P*"	<0,02*[Сумма на счете]

Рис. 7.2 – Фрагмент бланка запроса 2

3. Выполните запрос и проанализируйте результат.
4. Перейдите в режим SQL и сверьте текст запроса, сгенерированного ACCESS с текстом запроса, разработанного вами. Обратите внимание на организацию объединения записей.

5. Сохраните запрос под именем Запрос 7\_2.

**Запрос 3.** Найти записи по валютным вкладам, для которых начисления по процентам за год (за 12 месяцев) составят более 1000 у.е. По запросу должны отбираться только те записи, в которых рассчитанное значение выше указанного в условии. Кроме того в результате запроса вывести полное название счета, ФИО и адрес владельца счета. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА, СЧЕТА.

**Выполнение задания**

1. Текст запроса на языке *SQL ANSI*:

```
SELECT [№ счета], [Вид счета], [Начисления в месяц], [Вид валюты],
12*[Начисления в месяц] AS [% за год], [ФИО клиента], Адрес
FROM Счета, Клиенты, [Клиенты-счета]
WHERE Клиенты.[Код клиента] = [Клиенты-счета].[Код клиента]
AND Счета.[Код счета] = [Клиенты-счета].[Код счета]
AND [Вид валюты] = "у. е" AND 12*[Начисления в месяц] > 1000;
```

2. Фрагмент бланка запроса 3 представлен на рис. 7.3.

Поле:	Вид счета	Начисления в месяц	Вид валюты	Выражение1: 12*[Начисления в м
Имя таблицы:	Счета	Клиенты-счета	Клиенты-счет	
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:			"у. е"	>1000
или:				

Рис. 7.3 – Фрагмент бланка запроса 3

3. Выполните запрос и проанализируйте результат.

4. Перейдите в режим SQL и сверьте текст запроса, сгенерированного ACCESS с текстом запроса, разработанного вами. Обратите внимание на организацию объединения записей.

5. Сохраните запрос под именем Запрос 7\_3.

**Задание 3. Групповые операции в многотабличном запросе**

**Запрос 4.** Подвести итоги по каждому менеджеру за 2009 год: количество открытых счетов в валюте и в белорусских рублях, общую сумму средств на счетах, размер комиссионных за год. Указать фамилию менеджера и № филиала. Использовать таблицы КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ.

**Выполнение задания**

1. Текст запроса на языке *SQL ANSI*:

```
SELECT [ФИО менеджера], [№ филиала], Count([Сумма на счете]) AS
[Количество счетов], Sum([Сумма на счете]) AS [Итого],
```

```

[Комиссионные %]/100*Sum([Сумма на счете]) As Комиссионные, [Вид валюты]
FROM Менеджеры, [Клиенты-счета]
WHERE Менеджеры.[№ менеджера] = [Клиенты-счета].[№ менеджера]
AND [Дата открытия счета] like "*2009"
GROUP BY [ФИО менеджера], [№ филиала], [Вид валюты], [Комиссионные %]
ORDER BY [№ филиала], [ФИО менеджера], [Вид валюты] ;

```

2. Бланк запроса 4 представлен на рис. 7.4.

Поле:	ФИО менеджера	№ филиала	Сумма на счете	Сумма на счете	Комиссионные: [Комиссионные %]/100*Sum([Сумма на счете])
Имя таблицы:	Менеджеры	Менеджеры	Клиенты-счета	Клиенты-счета	
Групповая операция:	Группировка	Группировка	Count	Sum	Выражение
Сортировка:	по возрастанию				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					

Рис. 7.4 - Бланк запроса 4

3. Выполните запрос и проанализируйте результат.

4. Перейдите в режим SQL и сверьте текст запроса, сгенерированного ACCESS с текстом запроса, разработанного вами. Обратите внимание на организацию объединения записей.

5. Сохраните запрос под именем Запрос 7\_4.

**Запрос 5. Составить запрос,** позволяющий проанализировать популярность у вкладчиков видов вкладов, предоставляемых банком. Вывести полную информацию по видам вкладов (*Название вклада, Процентную ставку по вкладу, Общую сумму на каждом виде вклада и Количество открытых вкладов каждого вида*). Использовать таблицы КЛИЕНТЫ-СЧЕТА, СЧЕТА.

### Выполнение задания

1. Текст запроса на языке *SQL ANSI*:

```

SELECT [Вид счета], [% ставка], Sum([Сумма на счете]) AS [Итого],
Count([Сумма на счете]) AS [Количество счетов]
FROM Счета, [Клиенты-счета]
WHERE Счета.[Код счета] = [Клиенты-счета].[Код счета]
GROUP BY Счета.[Вид счета], Счета.[% ставка];

```

2. Бланк запроса 5 представлен на рис. 7.5.

Поле:	Вид счета	% ставка	Сумма на счете	Сумма на счете
Имя таблицы:	Счета	Счета	Клиенты-счета	Клиенты-счета
Групповая операция:	Группировка	Группировка	Sum	Count
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				

Рис. 7.5 - Бланк запроса 5

3. Выполните запрос и проанализируйте результат.



4. Перейдите в режим SQL и сверьте текст запроса, сгенерированного ACCESS с текстом запроса, разработанного вами. Обратите внимание на организацию объединения записей.

5. Сохраните запрос под именем Запрос 7\_5.

#### Задание 4. Ввод параметров в запрос

**Запрос 6.** Вывести из базы данных сведения о клиентах, имеющих на счетах сумму, указываемую пользователем в валюте, указанной пользователем. В результате запроса вывести поля *№ филиала*, *ФИО клиента*, *Адрес*, *Вид валюты*, *Сумма на счете*. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ.

#### Выполнение задания

1. Текст запроса на языке SQL ANSI:

```
SELECT [№ филиала], [ФИО клиента], Адрес, [Сумма на счете], [Вид валюты]
FROM Менеджеры, Клиенты, [Клиенты-счета]
WHERE Клиенты.[Код клиента] = [Клиенты-счета].[Код клиента] AND
      Менеджеры.[№ менеджера] = [Клиенты-счета].[№ менеджера]
AND [Сумма на счете] = [Введите сумму]
AND [Вид валюты] Like [Введите валюту]+ "*"
ORDER BY [№ филиала], [ФИО клиента];
```

2. Бланк запроса 6 представлен на рис. 7.6.

Поле:	№ филиала	ФИО клиента	Адрес	Сумма на счете	Вид валюты
Имя таблицы:	Менеджеры	Клиенты	Клиенты	Клиенты-счета	Клиенты-счета
Сортировка:	по возрастанию	по возрастанию			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:				[Введите сумму]	Like [Введите валюту]+ "*"

Рис. 7.6 - Бланк запроса 6

3. Выполните запрос и проанализируйте результат.

4. Перейдите в режим SQL и сверьте текст запроса, сгенерированного ACCESS с текстом запроса, разработанного вами. Обратите внимание на организацию объединения записей.

5. Сохраните запрос под именем Запрос 7\_6.

**Запрос 7.** Узнать, какая сумма по процентам к концу 2010 года накопится у каждого из вкладчиков, открывших вклады в 2009 году. *Вид счета* задать формальным параметром. В результате запроса кроме данных по счету вывести *ФИО клиента*. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА, СЧЕТА (см. запрос 5 из лабораторной работы №6).

#### Выполнение задания

1. Текст запроса на языке SQL ANSI:

```

SELECT [№ счета], [Вид счета], [Дата открытия счета], [Сумма на счете],
12-Month([Дата открытия счета])+12 AS [Срок, мес], (24-Month([Дата открытия
счета]))*[Начисления в месяц] AS [Всего %], [Вид валюты],[ФИО клиента]
FROM Счета, Клиенты, [Клиенты-счета]
WHERE Клиенты.[Код клиента] = [Клиенты-счета].[Код клиента] AND
Счета.[Код счета] = [Клиенты-счета].[Код счета]
AND [Вид счета] = [Вид счета?] AND [Дата открытия счета] <= #1/1/2010#;

```

2. Бланк запроса 7 представлен на рис. 7.7.

Поле:	№ счета	Вид счета	Дата открытия сч	Сумма на счете	Выражение1: 12-Month	Выражение2: (24-
Имя таблицы:	Клиенты-счета	Счета	Клиенты-счета	Клиенты-счета		
Сортировка:						
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		[Вид счета?]	<=#01.01.2010#			
или:						

Рис. 7.7 - Бланк запроса 7

3. Выполните запрос и проанализируйте результат.

4. Перейдите в режим SQL и сверьте текст запроса, сгенерированного ACCESS с текстом запроса, разработанного вами. Обратите внимание на организацию объединения записей.

5. Сохраните запрос под именем Запрос 7\_7.

**Запрос 8.** Подвести итоги по каждому менеджеру. *ФИО менеджера* и *№ филиала* задать формальным параметром. В результате запроса вывести поля *№ филиала, ФИО менеджера, количество открытых этим менеджером счетов, Общую сумму средств на счетах, вид валюты.* При необходимости задать собственные имена выводимым полям. Использовать таблицы КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ.

### Выполнение задания

1. Текст запроса на языке SQL ANSI:

```

SELECT [№ филиала], [ФИО менеджера], Count([Сумма на счете]) AS [Count-
Сумма на счете], Sum([Сумма на счете]) AS [Sum-Сумма на счете],[Вид валюты]
FROM Менеджеры, [Клиенты-счета]
WHERE Менеджеры.[№ менеджера] = [Клиенты-счета].[№ менеджера]
GROUP BY[№ филиала], [ФИО менеджера], [Вид валюты]
HAVING [№ филиала] = [Введите № филиала]
AND [ФИО менеджера] Like [Введите ФИО менеджера ]+"*"
ORDER BY Count([Сумма на счете]);

```

2. Бланк запроса 8 представлен на рис. 7.8.

3. Выполните запрос и проанализируйте результат.

4. Перейдите в режим SQL и сверьте текст запроса, сгенерированного ACCESS с текстом запроса, разработанного вами. Обратите внимание на организацию объединения записей.

## 5. Сохраните запрос под именем Запрос 7\_8.

Поле:	№ филиала	ФИО менеджера	Сумма на счете	Сумма на счете	Вид валюты
Имя таблицы:	Менеджеры	Менеджеры	Клиенты-счета	Клиенты-счета	Клиенты-счета
Групповая операция:	Группировка	Группировка	Count	Sum	Группировка
Сортировка:			по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	[Введите № филиала]	Like [Введите ФИО менеджера ]+"*"			

Рис. 7.8 - Бланк запроса 8

### Задания для самостоятельной работы

**Запрос 1.** Вывести сведения по рублевым счетам (поля ФИО клиента, № счета, Код счета, Сумма на счете), на которых размещены суммы от 10 до 20 млн. бел.руб. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА

**Запрос 2.** Вывести информацию по вкладу конкретного клиента. ФИО клиента задать формальным параметром. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА

**Запрос 3.** Вывести информацию по всем вкладчикам, проживающим в *Витебском районе*. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА

**Запрос 4.** По *ФИО менеджера* вывести информацию об обслуженных им счетах. *ФИО менеджера, вид валюты* задать формальным параметром. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ.

**Запрос 5.** Вывести информацию по счетам заданного вида по каждому филиалу. *№ филиала и вид счета*, задавать формальными параметрами. Использовать таблицы КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ

**Запрос 6.** Преобразуйте Запрос 4 в запрос с параметром.

**Запрос 7.** Вывести информацию о менеджере, обслуживающем самый крупный счет в указанной валюте. *Вид валюты* задать формальным параметром. Использовать таблицы МЕНЕДЖЕРЫ, КЛИЕНТЫ, КЛИЕНТЫ-СЧЕТА.

**Запрос 8.** Подвести итоги по всем видам вкладов за каждый месяц работы банка. В результат работы запроса включить полное название вклада. Использовать таблицы СЧЕТА, КЛИЕНТЫ-СЧЕТА.

**Запрос 9.** Подвести итоги по всем видам вкладов за задаваемый пользователем *месяц* работы банка. В результат работы запроса включить *полное название вклада*. Использовать таблицы СЧЕТА, КЛИЕНТЫ-СЧЕТА.

**Запрос 10.** Подвести *итоги* обслуживания клиентов банка, проживающих в *Витебском районе*. Использовать таблицы КЛИЕНТЫ, СЧЕТА, КЛИЕНТЫ-СЧЕТА.

---

## Лабораторная работа №8

### ПЕРЕКРЕСТНЫЕ ЗАПРОСЫ

---

**Цель работы:** Изучить возможности СУБД ACCESS по формированию запросов и поиску информации на их основе. Получить навыки работы с запросами.

#### Порядок выполнения работы

*Перекрестные запросы* — это запросы, в которых происходит статистическая обработка данных, результаты которой выводятся в виде таблицы, очень похожей на сводную таблицу Excel.

*Достоинства* перекрестных запросов: возможность обработки значительного объема данных и вывода их в формате, который очень хорошо подходит для автоматического создания графиков и диаграмм; простота и скорость разработки сложных запросов с несколькими уровнями детализации.

*Недостатки* перекрестных запросов: невозможность сортировки таблицы результатов по значениям, содержащимся в столбцах; возможность сортировки только для заголовков строк; сложность построения отчетов.

#### Задание 1. Создание перекрестной таблицы

**Запрос 1.** Пусть необходимо определить, количество счетов, открытых по каждому виду вклада.

##### Выполнение задания

1. Для создания запроса в окне базы данных выберите объект *Запросы* ⇒ *Создать*.
2. В окне *Новый запрос* выберите *Перекрестный запрос*.
3. В первом окне мастера *Создание перекрестных таблиц* выберите таблицу *КЛИЕНТЫ-СЧЕТА*, ее поля будут выведены в перекрестном запросе.
4. Во втором окне мастера выберите для заголовков строк поле *Код счета*.
5. В третьем окне мастера выберите для заголовков столбцов поле *Вид валюты*.
6. В следующем окне мастера для подсчета количество счетов открытых на разных видах вкладов и размещения этого значения в ячейке на пересечении строк и столбцов таблицы выберите поле *Сумма на счете* и функцию *Число*.
7. Бланк запроса представлен на рис. 8.1.

Поле:	Код счета	Вид валюты	Сумма на счете
Имя таблицы:	Клиенты-счета	Клиенты-счета	Клиенты-счета
Групповая операция:	Группировка	Группировка	Count
Перекрестная таблица:	Заголовки строк	Заголовки столбц	Значение
Сортировка:			

Рис. 8.1 – Бланк запроса 1

8. Выполните запрос и проанализируйте результат.

9. Просмотрите текст запроса в режиме SQL:

```

TRANSFORM Count([Клиенты-счета].[Сумма на счете]) AS [Count-Сумма на
счете]
SELECT [Клиенты-счета].[Код счета]
FROM [Клиенты-счета]
GROUP BY [Клиенты-счета].[Код счета]
PIVOT [Клиенты-счета].[Вид валюты];

```

10. Сохраните запрос под именем Запрос 8\_1.

## **Задание 2. Создание перекрестной таблицы на основе данных из нескольких таблиц**

**Запрос 2.** Очевидно, в результате выполнения перекрестного запроса, созданного в предыдущем задании, удобно видеть в таблице не коды счетов, а их названия.

**Указание:** названия можно получить из таблицы СЧЕТА, которая находится в отношениях 1:М с таблицей КЛИЕНТЫ-СЧЕТА и связана с ней по полю Код счета. Поскольку мастер не позволяет строить перекрестный запрос на полях нескольких таблиц, перед тем как создавать такой запрос, нужно создать запрос на выборку для таблицы СЧЕТА и КЛИЕНТЫ-СЧЕТА, в котором необходимо выбрать нужные поля.

### **Выполнение задания**

1. Создайте с помощью конструктора *запрос на выбор*. В этом запросе выберите из таблицы СЧЕТА поле *Вид счета*, а из таблицы КЛИЕНТЫ-СЧЕТА – поля *Вид валюты*.

2. Преобразуйте запрос на выбор в перекрестный запрос. Для этого выполните команду *Запрос ⇒ Перекрестный*. В бланке запроса появятся две дополнительные строки *Групповая операция* и *Перекрестная таблица*.

3. В строке *Перекрестная таблица* выберите поле *Вид счета* в качестве поля для заголовков строк, поле *Вид валюты* – для заголовков столбцов.

4. Для включения в перекрестную таблицу итоговых значений для каждой строки дополните бланк запроса полем *Сумма на счете*, для которого в строке *Перекрестная таблица* выберите параметр *Значение*.

5. В строке *Групповая операция* для поля *Сумма на счете* выберите из списка значение *Count*, оставив в остальных строках значение *Группировка*.

6. Бланк запроса 2 должен иметь вид, представленный на рис. 8.2.

Поле:	Вид счета	Вид валюты	Количество счетов
Имя таблицы:	Счета	Клиенты-счета	Клиенты-счета
Групповая операция:	Группировка	Группировка	Count
Перекрестная таблица:	Заголовки строк	Заголовки столбц	Значение
Сортировка:	Заголовки строк		
Условие отбора:	Заголовки столбцов		
или:	Значение		

Рис.8.2 – Бланк запроса 2

7. Просмотрите запрос в режиме SQL (объясните связь таблиц):

```

TRANSFORM Count([Клиенты-счета].[Сумма на счете]) AS [Количество
счетов]
SELECT Счета.[Вид счета]
FROM Счета INNER JOIN [Клиенты-счета] ON Счета.[Код счета] =
[Клиенты-счета].[Код счета]
GROUP BY Счета.[Вид счета]
PIVOT [Клиенты-счета].[Вид валюты];

```

8. Сохраните запрос под именем Запрос 8\_2.

### Задания для самостоятельного выполнения

1. Подведите итоги по открытым счетам по каждому клиенту.
  2. Подведите итоги за каждый день работы каждого менеджера: выведите общие суммы в разных видах валюты.
  3. Подведите итоги по открытым счетам за каждый день работы банка.
  4. Подведите итоги по счетам в валюте и в белорусских рублях по каждому филиалу.
  5. Подведите итоги по открытым счетам за каждый месяц работы банка.
- Указание:* используйте параметр **PIVOT** в формате:  
**PIVOT** Month([Дата открытия счета]);

---

## Лабораторная работа №9

### СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ОДНОТАБЛИЧНОЙ ФОРМЫ

---

**Цель работы:** Изучить возможности СУБД ACCESS по созданию и редактированию объектов типа ФОРМЫ. Получить навыки работы с формами при вводе информации в БД.

#### Порядок выполнения работы

**Форма** – это шаблон на экране, используемый, главным образом, для ввода, просмотра и редактирования записей.

Формы существуют простые и составные. Составной формой называется форма, включающая другие формы. Формы могут содержать различные элементы управления: надписи, поля, списки, раскрывающиеся списки, флажки, переключатели, кнопки, рисунки, вкладки, линии и др. В них возможны вычисления для отдельных записей и их групп, а также наглядное представление данных в виде диаграмм.

Форму можно создать тремя способами: с помощью конструктора форм, с помощью мастера по созданию форм, используя автоформу.

Форма может быть представлена на экране в одном из трех режимов: в режиме конструктора, в режиме формы (рабочем режиме), в режиме таблицы.

Переключение между режимами осуществляется с помощью команд меню *Вид*, либо с помощью кнопки *Вид* на панели инструментов (первая кнопка слева), открывающей список с названиями режимов.

#### Задание 1. Создание однотобличной формы в режиме автоформы

Рассмотрим создание однотобличной формы на примере таблицы МЕНЕДЖЕРЫ базы данных Банкь.

*Указание:* добавьте в таблицу МЕНЕДЖЕРЫ поле *Услуги* типа МЕМО, в котором перечислите круг обязанностей каждого менеджера. Например:

*Запись 1* (менеджер Николаев) — операции с ценными бумагами, операции по дорожным чекам, обмен валюты, ведение счетов;

*Запись 2* (менеджер Полякова) — расчетно-кассовое обслуживание юридических лиц, операции с ценными бумагами, операции по дорожным чекам, ведение счетов

*Запись 3* (менеджер Гусева) — оформление рублёвых пластиковых карточек, выдача наличных по пластиковым карточкам, ведение счетов.

*Запись 4* (менеджер Львова) — расчетно-кассовое обслуживание юридических лиц, операции с ценными бумагами, операции по дорожным чекам, ведение счетов

### **Выполнение задания**

Режим *Автоформа* обеспечит ввод/вывод всех полей выбранной таблицы в однозаписевой форме, которая удобна в данном случае, так как поле Услуги (тип *МЕМО*) требует много места в форме.

1. В окне базы данных *Банкъ* ⇒ область *Объекты* ⇒ выберите *Формы* ⇒ *Создать*.

2. В окне *Новая форма* выберите в качестве источника данных таблицу *МЕНЕДЖЕРЫ* и режим создания — *Автоформа: в столбец*. Работа команды *Автоформа: в столбец* завершается отображением формы. Сохраните форму под именем *МЕНЕДЖЕРЫ*.

3. Просмотрите информацию по менеджерам с помощью формы *МЕНЕДЖЕРЫ*.

### **Задание 2. Редактирование однотобличной формы, созданной мастером, в режиме конструктора**

Рассмотрим технику редактирования на примере ранее созданной однотобличной формы на основе таблицы *МЕНЕДЖЕРЫ* базы данных *Банкъ*. Все поля таблицы *МЕНЕДЖЕРЫ* должны размещаться в области данных формы в соответствии с требованиями пользователя. Приведем форму к виду, представленному на рис. 9.1.

В форму необходимо добавить заголовок, элемент управления — кнопку, которая будет закрывать форму, в области данных переместить поля с их надписями и установить параметры форматирования, кроме того можно ввести вычисляемое поле, в котором будет автоматически подсчитываться количество менеджеров, работающих в банке.

### **Выполнение задания**

1. Откройте форму, *Менеджеры* ранее созданную в режиме автоформы.

2. Перейдите в режим конструктора, нажав кнопку *Вид* на панели инструментов *Режим формы*. Кнопка *Вид* обеспечена списком, развернув который, можно выбрать необходимый режим представления формы. После перехода в режим конструктора форма откроется в окне конструктора форм.

3. Для ввода текста заголовка расширьте область заголовка формы, установив курсор мыши на границу области заголовка и области данных и перетаскивая эту границу на нужное расстояние.

4. Создадим графический элемент *Надпись*. Щелчком на кнопке *Надпись* панели элементов. На панели форматирования выберем



*полужирный* шрифт *Arial Cyr, 10* и выравнивание *по центру*. Переместим курсор мыши на место начала текста. Нажмем кнопку мыши и, не отпуская ее, растянем рамку текста до желаемого размера.

5. Введем в рамку созданного элемента *Надпись* текст заголовка: **МЕНЕДЖЕРЫ**. Завершим создание элемента нажатием клавиши *<Enter>* или щелчком кнопкой мыши вне рамки элемента.

*Рис. 9.1 Форма Менеджеры*

6. Аналогичным образом введем в область заголовка формы надпись *Акционерный банк газовой промышленности* и отформатируем ее.

7. Введем в область заголовка кнопку. На панели элементов выберем элемент *Кнопка* и разместим его в нужном месте (в области заголовка). Автоматически запустится *Мастер создания кнопок*. Следуя его указаниям, создадим кнопку *Закреть форму*.

8. Произведем последовательно перемещение пар связанных элементов *Поле* и *Надпись* в соответствии с размещением элементов, представленном на рис. 9.1. Для перемещения будем устанавливать курсор на границе рамки так, чтобы он отображался в виде раскрытой ладони:

- *Перемещение поля без перемещения его надписи.*

Когда вы передвигаете поле, его надпись обычно перемещается вместе с ним. Чтобы предотвратить перемещение надписи, поставьте указатель на левый верхний угол поля. При этом указатель приобретает вид руки с выставленным указательным пальцем. Теперь перетащите поле.

- *Равномерное расположение элементов по горизонтали (вертикали)*

Выделите элементы управления, которые необходимо выровнять. Перетащите самый левый элемент управления на то место, где он должен находиться. Сделайте то же с самым правым элементом управления. В

меню *Формат* ⇒ подменю *Интервал по горизонтали (Интервал во вертикали)* ⇒ *Сделать равным*.

- **Выравнивание элементов по размеру**

Выделите элементы управления, для которых требуется задать одинаковый размер. В меню *Формат* ⇒ *Размер* ⇒ *По самому .....*, — в зависимости от того, к размеру какого элемента управления вы хотите подогнать размеры элементов управления всей группы.

**9.** В области данных установите одинаковые параметры форматирования надписей полей: *шрифт Arial Cyr, 10, обычный, выравнивание по правому краю*. Для полей выберем *обычный шрифт ArialCyr, 10, выравнивание по левому краю*.

**10.** Добавление фонового рисунка.

- В режиме *Конструктор* ⇒ *Вид* ⇒ *Свойства* ⇒ вкладка *Макет* ⇒ *свойство Рисунок* и щелкните на строке этого свойства.
- Введите имя файла рисунка, который необходимо использовать в качестве фонового, или нажмите кнопку с многоточием в конце строки, чтобы найти его среди файлов.

**11.** Просмотрите свойства элементов формы. Для этого при выделенном элементе откройте диалоговое окно свойств по команде *Вид* ⇒ *Свойства* или с помощью контекстного меню.

**12.** Сохраните форму после редактирования в режиме конструктора (см. рис. 9.1) ⇒ *Файл* ⇒ *Сохранить* или кнопка панели инструментов *Сохранить*. Можно сохранить форму и при ее закрытии командой *Файл* ⇒ *Закрыть* или нажатием кнопки *Закрыть* окна формы. Для сохранения формы под новым именем ⇒ *Файл* ⇒ *Сохранить как*.

**13.** Для включения в форму вычисляемого поля следует на панели элементов нажать кнопку *Поле* и вычертить курсором поле в нужном месте формы. Вычисляемое поле должно размещаться в области примечаний формы.

**14.** Выделите создаваемое поле в форме и вызовите окно его свойств. В этом окне на вкладке *Данные* откройте окно *конструктора выражений*, из категории статистических функций выберите функцию *Count* (для подсчета количества записей), в качестве аргумента функции - поле № менеджера таблицы МЕНЕДЖЕРЫ. Закройте окна.

**15.** Перейдите в *режим формы*, просмотрите результат.

**16.** Введите через форму новые записи в базу данных по собственному усмотрению. Обратите внимание, на пересчет значения поля *Итого менеджеров в банке*.

### **Задания для самостоятельного выполнения**

- 1.** Создайте формы для каждой таблицы базы данных *Банкъ*. Предусмотрите вычисляемые поля и элементы управления.
- 2.** Внесите новые записи в базу данных, пользуясь формами.
- 3.** Проанализируйте результаты работы.
- 4.** Удалите (при необходимости) внесенные записи.

---

## Лабораторная работа №10

### КОНСТРУИРОВАНИЕ ФОРМ ДЛЯ РАБОТЫ С ДАННЫМИ ВЗАИМОСВЯЗАННЫХ ТАБЛИЦ

---

Цель работы: Изучить возможности СУБД ACCESS по созданию и редактированию объектов типа ФОРМЫ. Получить навыки работы с формами при вводе информации в БД.

#### Порядок выполнения работы

#### Задание 1. Создание двухтабличной формы с включением подчиненной формы

##### Выполнение задания

Осуществим средствами Access конструирование формы для одновременной работы с таблицами СЧЕТА и КЛИЕНТЫ-СЧЕТА в базе данных БанкЪ. Документом-источником такой формы является **Список открытых счетов** (см. лабораторную работу №2). С этого документа должны загружаться одновременно таблицы СЧЕТА и КЛИЕНТЫ-СЧЕТА, которые в совокупности образуют объект загрузки.

1. Перейдем в режим создания формы. Для этого в окне *БанкЪ: база данных* выбираем ⇒ *Объекты* ⇒ *Формы* ⇒ *Создать*.

2. В диалоговом окне *Новая форма* выбираем режим создания *Мастер форм* и таблицу *СЧЕТА*, которая будет служить источником записей для основной части создаваемой многотабличной формы. В открывшемся окне *Создание форм* в списке *Таблицы* ⇒ *Запросы* сначала будет отражена выбранная ранее таблица *СЧЕТА* и ее поля.

3. Выберем для таблицы *СЧЕТА* в списке *Доступные поля* все поля, перемещая их в область *Выбранные поля*.

4. Выберем далее из списка *Таблицы* ⇒ *Запросы* таблицу *КЛИЕНТЫ-СЧЕТА* и в списке *Доступные поля* все ее поля. Эта таблица будет *источником записей* подчиненной формы, связанных с записью, отображаемой в основной части формы. Нажмем кнопку *Далее*>.

5. В новом сеансе окна *Создание форм* схематично отобразится макет формы с перечнем полей в основной части формы и в подчиненной форме, если в рамке *Выберите тип представления данных* выделена таблица-источник записей основной части формы (*СЧЕТА*).

6. Для непосредственного включения подчиненной формы выберем вариант *Подчиненные формы* ⇒ *Далее*<sup>4</sup>.

7. В следующем диалоговом окне мастера для получения многозаписевой подчиненной формы и вывода подписей полей (определенных в свойствах таблиц) выберем внешний вид подчиненной формы - *ленточный*.

8. Если для подчиненной формы выбран внешний вид *ленточный*, то каждая запись будет представлена в отдельной области данных, которую можно настраивать. Если выбрать внешний вид подчиненной формы табличный, то данные подчиненной формы выводятся в виде обычной таблицы.

9. После нажатия кнопки *Далее* в очередном сеансе диалогового окна выберем стиль оформления. Если выбрать *Стандартный*, получим форму с утопленными полями и т. д..

10. В последнем окне *Создание форм* введем имена (заголовки) составной формы ИНФОРМАЦИЯ ПО СЧЕТАМ и подчиненной формы КЛИЕНТЫ-СЧЕТА.

11. Выберем в этом же окне дальнейшие действия мастера — *Открытие формы для просмотра и ввода данных*. Это позволит после завершения работы мастера сразу приступить к работе с данными через форму.

12. Чтобы мастер завершил создание формы, следует нажать кнопку *Готово*. Сохранение формы производится автоматически.

13. Соответственно выбранным действиям, мастер выводит форму с данными из таблиц СЧЕТА и КЛИЕНТЫ-СЧЕТА. При этом в основной части формы отображаются данные одной текущей записи таблицы СЧЕТА. В подчиненной форме выводятся те записи таблицы КЛИЕНТЫ-СЧЕТА, которые связаны с текущей записью таблицы СЧЕТА. Дополнительно создается форма КЛИЕНТЫ-СЧЕТА.

## Задание 2. Редактирование формы в режиме Конструктор

### **Выполнение задания**

Произведем следующие действия по доработке формы ИНФОРМАЦИЯ ПО СЧЕТАМ.

1. Введем в область заголовка полное название формы ИНФОРМАЦИЯ ПО СЧЕТАМ, которое будет выводиться при распечатке формы.

2. Разместим поля формы так, как показано на рис. 10.1, выделяя курсором мыши необходимый элемент и перетаскивая его в нужное место.

3. Уточним подписи полей, шрифт и размеры полей и подписей, заголовков формы. Для изменения размера любого элемента надо расширять или сжимать рамочку этого элемента. Можно задать ширину и высоту подписи в соответствии с размером и шрифтом текста, используя кнопку *По размеру данных*.

---

<sup>4</sup> Если выбрать вариант *Связанные формы*, вложенная форма будет открываться из основной нажатием кнопки.

4. Аналогичные действия по доработке выполним для подчиненной формы **КЛИЕНТЫ\_СЧЕТА**. Для редактирования подчиненной формы следует перевести курсор в область подчиненной формы или открыть подчиненную форму **КЛИЕНТЫ\_СЧЕТА** из перечня форм в окне базы данных.

5. Уточним подписи полей-столбцов в заголовке формы, а также шрифт, размеры полей и подписей. Сохраним подчиненную форму после редактирования.

### **Задание 3. Создание кнопок управления с помощью мастера**

#### **Выполнение задания**

Для того чтобы в форме переходить к следующей и предыдущей записи таблицы **СЧЕТА**, создадим соответствующие *кнопки управления* в основной части формы.

1. Нажмем кнопку *Мастер элементов* на панели элементов Access, а затем используем инструмент *Кнопка*. После переноса кнопки и вычерчивания ее рамки в выбранном месте (см. рис. 10.1) запустится мастер кнопок *Создание кнопок*.

2. В окне *Мастера кнопок* выберем действие, которое необходимо выполнять при нажатии кнопки. В области *Категории* выберем *Переходы по записям*, в области *Действия* выберем *Предыдущая запись*.

3. В следующем сеансе диалогового окна можно выбрать вид кнопки - *Текст* или *Рисунок*. Выбираем *Рисунок* ⇒ *Показать все рисунки*. Выбираем подходящий рисунок из списка, например, *Стрелка вверх (синяя)* ⇒ *Готово*, в результате чего кнопка перехода с выбранным рисунком встраивается в форму.

4. Аналогичные действия произведем для встраивания кнопки перехода к последующей записи таблицы, выбирая в области *Действия* соответственно пункт *Следующая запись* и рисунок *Стрелка вниз (синяя)*.

5. Отредактируем размер и надписи кнопок для перехода к записи другой группы, записав *ПРЕДЫДУЩАЯ, СЛЕДУЮЩАЯ*.

6. Для создания кнопки *закрытия* формы в области *Категории* выбираем *Работа с формой*, а в области *Действия* - *Закреть Форму*. После формирования кнопки заменим ее название на *ЗАКРЫТЬ*.

### **Задание 4. Вычисление итоговых значений**

Вычисление итоговых значений для записей формы выполняется с помощью встроенных статистических, функций, записываемых в качестве выражения в вычисляемых элементах управления. Например, можно создать элемент управления для суммирования по полю *Сумма на счете*:

**Sum** ([Сумма на счете]).

Вычисляемый элемент управления, создаваемый для расчета итогового значения нужно размещать в области *примечания* формы,

Пусть в форме ИНФОРМАЦИЯ ПО СЧЕТАМ нужно на основе записей подчиненной формы КЛИЕНТЫ-СЧЕТА подсчитать количество открытых счетов каждого вида (функция Count) и общую сумму средств на счетах каждого вида (функция Sum).

### Выполнение задания

1. В подчиненной форме в области примечаний создадим вычисляемые элементы управления с именами *Поле1*, *Поле2*.

2. В соответствии с методикой, изложенной в лабораторной работе 9 (п.12, 13) для подсчета числа открытых счетов каждого вида через окно свойств поля в *Поле1* введем функцию *Count([№ счета])*, в *Поле2* введем функцию *Sum([Сумма на счете])*. Причем в качестве аргумента функции берем не имя элемента управления в форме, а имя поля, так как источником данных является соответствующая таблица или запрос.

3. Изменим имена элементов управления *Поле1* и *Поле2* на соответствующие по смыслу – *Всего открыто счетов* и *Итого по счетам*.

4. Перейдем из режима конструктора в режим формы и просмотрим результат (см рис. 10.1).

The screenshot shows a form titled "ИНФОРМАЦИЯ ПО СЧЕТАМ". It contains several input fields for account details: "Вид счета" (Ruble account), "% ставка" (25), "Срок мес" (6), and "Код счета" (PC). Below these is a table of accounts with columns for account number, client code, account code, opening date, currency, balance, monthly interest, and manager. Summary statistics show 6 open accounts with a total balance of 5,700,000. Navigation and control buttons are visible at the bottom.

№ счета	Код клиента	Код счета	Дата открытия счета	Вид валюты	Сумма на счете	Начисления в месяц	№ менеджера
1014	K14	PC	05.09.2009	бел. р.	10000000	208333,3	M1
1015	K7	PC	15.09.2009	бел. р.	5000000	104166,7	M3
1019	K12	PC	25.09.2009	бел. р.	7000000	145833,3	M1
1022	K15	PC	10.06.2010	бел. р.	10000000	208333,3	M1
1004	K3	PC	14.07.2009	бел. р.	10000000	208333,3	M2

Всего открыто счетов: 6      Итого по счетам: 57000000

Рис.10.1 Многотабличная форма Информация по счетам

### Задание 5. Разработка одиночной многотабличной формы при использовании мастера

Рассмотрим создание *одиночной* многотабличной формы без подчиненных и связанных форм, предназначенных для работы с данными всех таблиц. В соответствии со схемой базы данных Банкxь таблица КЛИЕНТЫ-

СЧЕТА, содержащая перечень всех операций проводимых банком, подчинена таблицам МЕНЕДЖЕРЫ, КЛИЕНТЫ, СЧЕТА, которые содержат справочные данные. Для отображения в одной записи формы данных об отдельном счете из таблицы КЛИЕНТЫ-СЧЕТА, а также дополнительной справочной информации (о клиенте, виде счета и обслуживающем счет менеджере) можно построить *многотабличную форму без подчиненных форм*, т. е. *одинокую форму*<sup>5</sup>.

Источником записей и дополнительных полей формы должна быть таблица КЛИЕНТЫ-СЧЕТА. Таблицы МЕНЕДЖЕРЫ, КЛИЕНТЫ, СЧЕТА будут источниками для расшифровки кодов значений.

### Выполнение задания

1. В окне *Банк: база данных* ⇒ *Объекты* ⇒ *Формы* ⇒ *Создать*.

2. В окне *Новая форма* в качестве таблицы, которая будет источником записей формы, следует выбрать таблицу КЛИЕНТЫ-СЧЕТА и способ создания ⇒ *Мастер форм*.

3. В окне *Создание форм* выберем поля таблицы КЛИЕНТЫ-СЧЕТА (*№ счета*, *Сумма на счете*, *Вид валюты*, *Дата открытия счета*), включаемые в форму, а затем — поля таблиц СЧЕТА (*Вид\_счета*), КЛИЕНТЫ (*ФИО клиента*), МЕНЕДЖЕРЫ (*ФИО менеджера*).

4. Следуя указанием мастера, построим форму типа *Одинокая форма* с выбранными полями из всех таблиц.

5. Выберем вид формы *В один столбец*, при котором в форме будет отображаться только одна запись, построенная путем добавления полей из всех таблиц.

6. Завершим построение формы, задав ей имя СПИСОК СЧЕТОВ.

7. Перейдя в режим *конструктора Форм*, можно отредактировать внешний вид формы по собственному усмотрению.

Одинокая многотабличная форма, построенная мастером доработанная в режиме конструктора, приведена на рис. 10.2.

СПИСОК СЧЕТОВ	
№ счета	1013
Вид счета	Валютный депозит
ФИО клиента	Смирнова
Дата открытия счета	05.09.2009
Сумма на счете	8000
Вид валюты	у.е.
ФИО менеджера	Николаев

Запись: 1 из 23

Рис. 10.2 - Одинокая многотабличная форма

<sup>5</sup> В процессе создания одиночной формы мастером автоматически формируется запрос на соединение подчиненной таблицы с главной. Этот запрос становится источником записей одиночной формы. Запрос можно просмотреть в общих свойствах формы.

## Задания для самостоятельного выполнения

1. Создайте с помощью мастера двухтабличную форму **МЕНЕДЖЕРЫ** с включением подчиненной формы **СПИСОК СЧЕТОВ**. Предусмотрите элементы управления и вычисляемые поля.
2. Создайте с помощью мастера двухтабличную форму **МЕНЕДЖЕРЫ** с включением подчиненной формы **ИТОГИ РАБОТЫ** (постройте эту форму на основании запроса, подводящего итоги работы каждого менеджера).
3. Создайте с помощью мастера двухтабличную форму **ФИЛИАЛЫ** с включением подчиненной формы **ИТОГИ РАБОТЫ** (постройте эту форму на основании запроса, подводящего итоги работы каждого филиала).
4. Создайте с помощью мастера двухтабличную форму **СЧЕТА** с включением подчиненной формы **КЛИЕНТЫ** и связанной формы **МЕНЕДЖЕРЫ**. Предусмотрите элементы управления в главной форме и вычисляемые поля в подчиненной и связанной формах (В форме **МЕНЕДЖЕРЫ** рассчитайте размер комиссионных менеджера от каждой операции).



---

## Лабораторная работа №11

### КОНСТРУИРОВАНИЕ ОТЧЕТОВ

---

**Цели занятия:** Изучить возможности СУБД ACCESS по созданию и редактированию объектов типа ОТЧЕТ. Получить навыки работы с отчетами.

#### Порядок выполнения работы

##### **Задание 1. Создание однотобличного отчета с помощью конструктора**

Рассмотрим создание однотобличного отчета на примере получения списков счетов по видам валюты.

Определим требования к макету отчета.

- Списки открытых счетов по каждому виду валюты должны выводиться последовательно вместе с заголовками.
- Необходимо рассчитать общую сумму средств, хранящихся на счетах, по каждому виду валюты и отобразить это значение в отчете.
- Записи списка счетов должны выводиться в хронологическом порядке.

##### **Выполнение задания**

**1.** Создадим отчет в режиме мастера. В окне базы данных выберем объект *Отчеты* ⇒ *Создать*. В первом диалоговом окне Мастера *Новый отчет* выберем таблицу **КЛИЕНТЫ-СЧЕТА**, которая будет источником данных для отчета.

**2.** Во втором окне Мастера *Создание отчетов* выберем все поля таблицы **КЛИЕНТЫ-СЧЕТА** для включения в отчет.

**3.** В третьем окне Мастера добавим уровни группировки, поскольку общий список должен быть разбит по группам: первый уровень - *Вид валюты*, второй уровень - *Код клиента*. Уровень группировки, установленный мастером по умолчанию, можно изменять, пользуясь соответствующими элементами управления окна *Создание отчетов*.

**4.** Далее зададим порядок сортировки по полю *№ счета*, и, нажав кнопку *Итоги*, определим, какое из итоговых значений следует рассчитывать для каждой группы. В соответствии с заданием выберем функцию *Sum* для поля *Сумма на счете*.

**5.** В следующем окне мастера посмотрим возможные виды отчетов и выберем *Ступенчатый*.

**6.** Определим стиль оформления отчета и зададим имя *Список счетов*.

**7.** Готовый отчет можно просмотреть из окна базы данных *БанкЪ*.

## Задание 2. Редактирование отчета с помощью конструктора

### Выполнение задания

1. Откройте созданный отчет в режиме конструктора.
2. Очевидно, что вычисляемые итоги по группам *Код клиента* являются избыточными, поэтому их следует просто убрать (*область примечаний группы Код клиента*).
3. Вычисление общей суммы средств по всем счетам (итог по всему отчету) является некорректным, так как нет разбиения по видам валюту, поэтому это поле также следует убрать (*область примечаний отчета*).
4. Изменим заголовок отчета **КЛИЕНТЫ-СЧЕТА**, сформированный мастером по умолчанию, на **СПИСОК СЧЕТОВ** (*область заголовка отчета*).
5. При необходимости можно изменить тип и размер шрифта полей и надписей (по усмотрению), способ выравнивания текста и т.п., аналогично тому, как это делается при работе с формами.
6. Готовый отчет может иметь вид, представленный на рис. 11.1<sup>6</sup>.

<b>СПИСОК СЧЕТОВ</b>							
<i>Вид валюты</i>	<i>Код клиента</i>	<i>№ счета</i>	<i>Код счета</i>	<i>Дата открытия счета</i>	<i>Сумма на счете</i>	<i>Начисления в месяц</i>	<i>№ менеджера</i>
<b>бел. р.</b>							
	<i>K1</i>	1002	РД	01.06.2009	1000000	18333,33	М2
	<i>K12</i>	1019	РС	25.09.2009	7000000	145833,3	М1
	<i>K14</i>	1014	РС	05.09.2009	10000000	208333,3	М1
	<i>K15</i>	1022	РС	10.06.2010	10000000	208333,3	М1
	<i>K2</i>	1003	РД	01.06.2009	5000000	91666,66	М2
	<i>K3</i>	1004	РС	14.07.2009	10000000	208333,3	М2
	<i>K6</i>	1009	РТ	01.06.2009	25000000	208333,3	М1
		1010	РС	10.06.2009	15000000	312500	М1
	<i>K7</i>	1015	РС	15.09.2009	5000000	104166,7	М3
	<i>K8</i>	1016	РД	06.09.2009	8000000	146666,7	М3
	<b>Итого по счетам</b>				<b>96000000</b>		
<i>12 августа 2010 г.</i>							
<i>Страница 1 из 3</i>							

Рис.11.1 – Пример однотобличного отчета

<sup>6</sup> Для более детального редактирования отчета, созданного мастером, следует пользоваться панелью инструментов конструктора отчетов и окном свойств поля.

### **Задание 3. Отчет с подчиненной частью на основе двух взаимосвязанных таблиц**

#### **Выполнение задания**

##### **❖ Создание основной части отчета на базе главной таблицы**

1. В качестве главной таблицы используем таблицу **МЕНЕДЖЕРЫ**.
2. Определим требования к макету отчета со списками счетов, обслуживаемых менеджерами:
  - Будем выводить в отчет данные по каждому менеджеру, группируя их по месту работы (по филиалу), включая *ФИО менеджера, размер комиссионных, перечень оказываемых услуг*. Эти данные содержатся в таблице **МЕНЕДЖЕРЫ**.
  - В табличной части (подчиненном отчете) по каждому менеджеру будем выводить данные об обслуживаемых им счетах. Эти данные содержатся в таблице **КЛИЕНТЫ-СЧЕТА**.
3. В окне базы данных выберем объект *Отчеты* ⇒ кнопку *Создать*. В окне *Новый отчет* ⇒ *Мастер отчетов* для построения отчета в режиме мастера ⇒ таблицу **МЕНЕДЖЕРЫ**, которая будет источником данных для основной части отчета ⇒ *ОК*.
4. Далее, следуя указаниям мастера, построим *ступенчатый отчет*, задав группировку по полю *№ филиала* и сортировку по полю *ФИО менеджера*.
5. Просмотрим готовый отчет и отредактируем его по собственному усмотрению, перейдя в режим конструктора.
6. Сохраним отчет под именем **МЕНЕДЖЕРЫ**.

##### **❖ Создание подчиненного отчета в режиме Автоотчет**

1. Для автоматического создания многозаписевого подчиненного отчета в окне *Новый отчет* выберем режим *Автоотчет: ленточный*,
2. В качестве источника данных этого отчета выберем таблицу **КЛИЕНТЫ-СЧЕТА**.
3. Завершим создание автоотчета выводом на экран отчета, в котором в качестве заголовка взято имя таблицы. В отчет включены все поля таблицы и в качестве заголовков столбцов выбраны подписи этих полей, заданные в свойствах таблицы. Сохраним отчет под именем **КЛИЕНТЫ-СЧЕТА**.
4. Доработаем подчиненный отчет в режиме конструктора для того чтобы его можно было включить в основную часть отчета в соответствии с проектом макета
  - Удалим верхний и нижний колонтитулы, поскольку отчет будет использоваться только как подчиненный.
  - Удалим заголовки полей.
  - Зададим стили оформления текста полей.
  - Изменим заголовок отчета **КЛИЕНТЫ-СЧЕТА** на **СЧЕТА**

- Сохраним отредактированный подчиненный отчет.

### ❖ Включение подчиненного отчета в основной

#### 1-й способ

Включим подчиненный отчет в основной способом перетаскивания.

1. Разместим для этого на экране окно базы данных рядом с окном отчета МЕНЕДЖЕРЫ в режиме конструктора. Перейдем в окно базы данных. Выберем из перечня имен отчетов подчиненный отчет КЛИЕНТЫ-СЧЕТА и перетащим его в область данных отчета МЕНЕДЖЕРЫ.

2. Удалим элемент надписи подчиненного отчета, выделив его и нажав клавишу <Del>.

3. Для того чтобы подчиненный отчет при отображении не был заключен в рамку, выделим подчиненный отчет, откроем его свойства и на вкладке *Макет* в строке *Тип границы* выберем значение *Отсутствует*.

Просмотреть содержимое двухтабличного отчета МЕНЕДЖЕРЫ (рис. 11.2), можно нажав кнопку Предварительный просмотр на панели конструктора отчетов.

<i>Менеджеры</i>					
<i>№ филиала</i>	<i>ФИО менеджера</i>	<i>Комиссионные %</i>	<i>Услуги</i>		
1	Николаев	5	операции с ценными бумагами, операции по дорожным чекам, обмен валюты, ведение счетов		
<i>Счета</i>					
1009	РТ	01.06.2009	25000000	бел. р.	208333,3
1010	РС	10.06.2009	15000000	бел. р.	312500
1001	ВД	01.08.2009	5000	у. е	41,66667
1014	РС	05.09.2009	10000000	бел. р.	208333,3
1013	ВД	05.09.2009	8000	у. е	66,66666
1019	РС	25.09.2009	7000000	бел. р.	145833,3
1022	РС	10.06.2010	10000000	бел. р.	208333,3

Рис.11.2 – Фрагмент отчета с подчиненным отчетом

#### 2-й способ

1. Воспользуемся для включения подчиненного отчета другим способом, кнопкой панели элементов *Подчиненная форма/отчет* при выключенной кнопке панели элементов *Мастер элементов*. При этом создается элемент *Свободный* с надписью *Внедренный*.

2. В свойствах элемента *Свободный* на вкладке *Данные* в строке *Объект-источник* выберем встраиваемый отчет КЛИЕНТЫ-СЧЕТА.

Access автоматически устанавливает связь между основным и подчиненным отчетом по полю № менеджера, т. к. она определена в схеме данных. При этом не имеет значения, включено ли поле связи в подчиненный отчет. Access работает со всеми полями таблицы, на базе которой построен отчет. Увидеть эту связь можно в свойствах подчиненного отчета на вкладке *Данные* в строках *Подчиненные поля* и *Основные поля*.

При необходимости созданный отчет с подчиненной частью опять может быть доработан в режиме конструктора.

#### **Задание 4. Построение мастером многотабличного отчета без подчиненной части**

##### **Выполнение задания**

1. Определим требования к отчету: в отчет должны выводиться в виде списка информация по каждому виду счета, предоставляемому банком, с перечнем списка клиентов с указанием фамилии, имеющих счета соответствующего вида. Строки отчета должны быть упорядочены по полю Вид счета, записи о клиентах — по полю Дата открытия счета. При выводе данных в отчете должны также формироваться расчетные итоги – сумма средств на счете каждого вида.

2. Определим таблицы, необходимые для построения отчета. Основные сведения о видах счетов содержит подчиненная таблица СЧЕТА, из которой можно получить информацию о каждом виде счета. Эта таблица является основным источником записей для создаваемого отчета. Список открытых счетов содержит главная таблица КЛИЕНТЫ-СЧЕТА, а фамилии клиентов — таблица КЛИЕНТЫ.

3. Создадим отчет:

- Выберем в окне *Новый отчет* ⇒ *Мастер отчетов* ⇒ подчиненную таблицу СЧЕТА в качестве записеобразующей таблицы для отчета.
- В окне *Создание отчетов* выберем из таблицы СЧЕТА поля *Вид счета*, *% ставка*, *Срок мес.*
- Из таблицы КЛИЕНТЫ-СЧЕТА выберем поля *Дата открытия счета*, *№ счета*, *Сумма на счете*, *Вид валюты*, *Начисления в мес*
- Из таблицы КЛИЕНТЫ выберем поле *ФИО клиента* ⇒ *Далее >>*.
- В следующем сеансе окна мастера в строке *Выберите тип представления данных* выделяется таблица СЧЕТА, поскольку она была выбрана в качестве источника записей ⇒ *Далее >>*.
- Зададим уровень группировки по полю *Вид счета*, и добавим второй уровень группировки по полю *ФИО клиента*. При этом в окне

мастера отображается общая структура формируемого макета отчета ⇒ *Далее >>*.

- В следующем сеансе диалогового окна мастера *Создание отчета* выберем поле *Дата открытия счета*, по которому нужно произвести сортировку по возрастанию в группе.
- Для подсчета итоговых значений в поле отчета *Итого по группе* в итоговой строке группы, нажмем кнопку *Итоги*. Для числовых полей в открывшемся окне *Итоги* будут выведены строки, в которых выберем функцию *Sum* для поля *Сумма на счете*, чтобы рассчитать суммарное количество средств на счетах каждого вида ⇒ *Далее >>*..
- Если необходимо подсчитать долю суммарных часов группы от общих часов всех групп, нужно отметить *Вычислить проценты*.
- В следующем сеансе окна мастера выберем вид макета отчета блок с ориентацией альбомная. Установим флажок *Настроить ширину полей для размещения на одной странице* ⇒ *Далее >>*.
- В очередном сеансе выберем один из возможных стилей оформления - полужирный. Потом зададим имя отчета *Счета*, которое также отобразится в заголовке отчета. Под этим именем мастер автоматически сохраняет отчет в базе данных ⇒ *Далее >>* ⇒ *Готово*.
- Полученный отчет по основным параметрам соответствует сформулированным требованиям к отчету. В него включены поля из трех взаимосвязанных таблиц. Причем нам не потребовалось задавать связи между таблицами и включать в отчет поля *Код счета* и *Код клиента*, являющиеся полями связи.
- Перейдем в режим конструктора отчетов и, пользуясь приемами редактирования, отредактируем макет отчета по собственному усмотрению. Пример отредактированного отчета, построенного на основании трех таблиц, приведен на рис. 11.3.

### **Задание 5. Просмотр запроса, который формирует мастер для создания многотабличного отчета.**

На основе информации, сохраняемой в схеме данных, и заданных пользователем полей при создании макета отчета мастер строит необходимый запрос. По этому запросу формируются записи из полей нескольких взаимосвязанных таблиц.

1. Откроем окно свойств отчета (рис. 11.4). В строке *Источник записей* отображается созданная мастером инструкция SQL, реализующая запрос и определяющую выборку заданных полей из различных таблиц для формирования отчета.

Счета								
Вид счета	% ставка	Срок мес	ФИО клиента	Дата открытия счета	№ счета	Сумма на счете	Вид валюты	Начисления в месяц
Валютный депозит	10	6	Арт емова	01.08.2009	1001	5000	у.е	41,7
			Мавзышова	04.09.2009	1020	10000	у.е	83,3
			Смирнова	05.09.2009	1013	8000	у.е	66,7
			Шитикова	05.09.2009	1012	5000	у.е	41,7
			Шараб	25.09.2009	1018	7000	у.е	58,3
			Шареев	05.06.2010	1020	6000	у.е	50,0
Итого на счетах вида Валютный депозит				41000				
Валютный срочный	12	12	Григорьева	20.05.2009	1008	5000	у.е	50,0
			Алексеева	22.06.2009	1005	10000	у.е	100,0
				22.06.2009	1006	5000	у.е	50,0
			Сергеева	05.07.2009	1017	5000	у.е	50,0
			Мавзышова	02.09.2009	1011	6000	у.е	60,0
			Савосеева	05.06.2010	1021	10000	у.е	100,0
Итого на счетах вида Валютный срочный				41000				
Валютный текущий	3	12	Григорьева	10.05.2009	1007	10000	у.е	25,0
Итого на счетах вида Валютный текущий				10000				

13 августа 2010 г. Страница 1 из 2

Рис. 11. 3– Пример отредактированного отчета

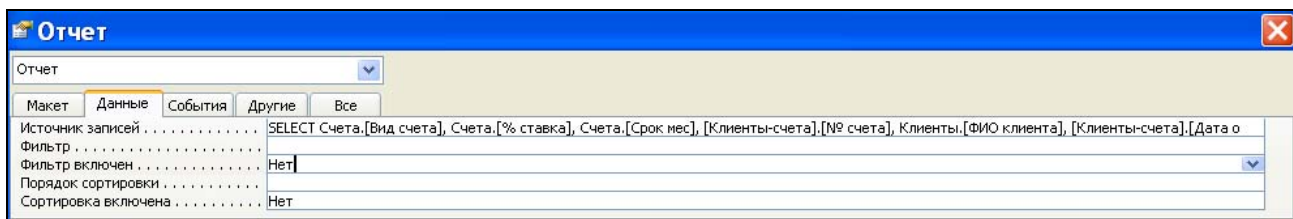


Рис. 11.4 – Окно свойств отчета

2. Просмотрим запрос, созданный мастером в виде запроса SQL. Для этого в диалоговом окне свойств отчета в строке *Источник записей* выберем *Построитель*, которая вызовет построитель запросов, в окне которого отображается схема данных и бланк запроса (рис. 11.5).

3. Перейдя в режим SQL, можно прочитать текст SQL-запроса, сгенерированного Access для формирования отчета:

```
SELECT Счета.[Вид счета], Счета.[% ставка], Счета.[Срок мес], [Клиенты-счета].[№ счета], Клиенты.[ФИО клиента], [Клиенты-счета].[Дата открытия счета], [Клиенты-счета].[Сумма на счете], [Клиенты-счета].[Вид валюты], [Клиенты-счета].[Начисления в месяц]
FROM Счета INNER JOIN (Клиенты INNER JOIN [Клиенты-счета] ON
Клиенты.[Код клиента] = [Клиенты-счета].[Код клиента]) ON Счета.[Код счета] =
[Клиенты-счета].[Код счета];
```

4. Обратите внимание, что в списке полей также записана инструкция SQL. Отобразим список полей, доступных в отчете. Для этого нажмем на панели конструктора отчетов кнопку *Список полей*. В заголовке открывшегося окна можно увидеть инструкцию SQL.

**Инструкция SQL : построитель запросов**

Поле:	Вид счета	% ставка	Срок мес	№ счета	ФИО клиента	Дата открытия сч-
Имя таблицы:	Счета	Счета	Счета	Клиенты-счета	Клиенты	Клиенты-счета
Сортировка:						
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:						

*Рис. 11.5 – Фрагмент бланка запрос, подготовленного Мастером отчетов, при создании отчета*



## Лабораторная работа №12

### РАЗРАБОТКА МАКРОСОВ В БАЗАХ ДАННЫХ

**Цель работы:** изучить возможности Access по созданию и использованию макросов.

#### Теоретические сведения

В Access макрос – это определенная последовательность операций. Макросы хранятся в окне, напоминающем таблицу, в таком порядке, в котором их необходимо выполнять. При разработке макроса необходимо задать действия (ввести макрокоманды), которые он должен выполнить (например, открыть форму, распечатать отчет, выполнить запрос или экспортировать содержимое таблицы в файл электронной таблицы). С полным перечнем стандартных макрокоманд, поддерживаемых СУБД Access, можно ознакомиться в специальной литературе и справочной системе.

При создании макросов для быстрого переключения между панелями описаний и аргументов, так же как при работе с таблицами и запросами в режиме Конструктора, можно использовать клавишу <F6>.

#### Порядок выполнения работы

**Задание 1<sup>7</sup>.** Создать *линейный* макрос для выделения таблицы КЛИЕНТЫ-СЧЕТА. Сохранить созданный макрос под именем *Клиенты-Счета*.

#### Выполнение задания

1. Для создания макроса в первом столбце окна базы данных выберите объект *Макросы* и нажмите кнопку *Создать*. Откроется окно конструктора макроса.
2. В столбце *Макрокоманда* щелкните в первой строке, раскроется список макрокоманд и выберите команду *ВыделитьОбъект*.
3. В качестве ее аргументов задайте следующие:

Таблица 12.1 – Аргументы команды *ВыделитьОбъект*

<b>Тип объекта</b>	Таблица
<b>Имя объекта</b>	Клиенты-счета
<b>В окне базы данных</b>	Да

<sup>7</sup> Пример подготовлен студенткой группы МК-20 УО «ВГТУ» Швындиковой Е.П.

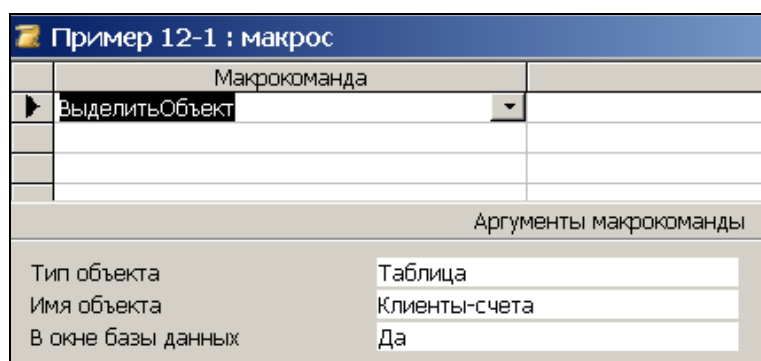


Рис.12.1-Окно конструктора макроса

4. Сохраните созданный макрос, нажав на кнопку *Сохранить*, и запустите его, используя кнопку *Запуск* на панели инструментов. Результатом работы данного макроса будет выделение таблицы КЛИЕНТЫ-СЧЕТА.

Для выполнения следующего задания должна быть создана форма *Счета* по таблице СЧЕТА.

**Задание 2.** Создать макрос, позволяющий автоматически открывать форму *Счета* в режиме добавления записей. Макрос сохранить с именем *Для добавления счетов*.

#### Выполнение задания

1. Нажмите кнопку *Создать* для открытия окна создания макросов.
2. В столбце *Макрокоманда* щелкните в первой строке, раскроется список макрокоманд и выберите команду *ОткрытьФорму*.
3. В качестве ее аргументов задайте следующие (см. таблицу 12.1):

Таблица 12.1 – Аргументы команды *ОткрытьФорму*

<b>Имя формы</b>	Счета
<b>Режим</b>	Форма
<b>Имя фильтра</b>	
<b>Условие отбора</b>	
<b>Режим данных</b>	Добавление
<b>Режим окна</b>	Обычный

4. Сохраните и запустите макрос на выполнение по команде. Результатом работы данного макроса будет отображение формы *Счета* с пустыми полями.

**Задание 3.** Создать макрос, который выбирает из таблицы СЧЕТА рублевые счета с процентной ставкой больше 20%. Сохранить созданный макрос под именем *Рублевые счета*.

#### Выполнение задания

1. Нажмите кнопку *Создать* для открытия окна создания макросов. Для

создания макроса в первом столбце окна базы данных выберите объект *Открыть таблицу*.

2. После выбора станет активной панель *Аргументы макрокоманды*, расположенное внизу экрана. В этой панели выберите таблицу, которую надо открыть, т.е. таблицу СЧЕТА. Остальные параметры оставьте без изменений.

3. Выберите следующую макрокоманду *ПрименитьФильтр*, щелкнув кнопкой мыши в следующей строке столбца *Макрокоманда*. В панели Аргументы макрокоманды укажите условия фильтрации. Для этого в строке *Условие отбора* напишите `[Счета]![% ставка]>20` (см. таблицу 12.2).

Таблица 12.2 – Текст макроса «Рублевые счета»

Макрокоманда	Аргументы макрокоманды	
ОткрытьТаблицу	Имя таблицы	Счета
	Режим	Таблица
	Режим данных	Изменение
ПрименитьФильтр	Имя фильтра	
	Условие отбора	<code>[Счета]![% ставка] &gt; 20</code>

4. Сохраните и запустите макрос на выполнение по команде.

5. После просмотра закройте таблицу СЧЕТА, не сохраняя изменения.

**Задание 4.** Создать макрос с условием, который выбирает из таблицы КЛИЕНТЫ-СЧЕТА клиентов, у которых на рублевых счетах размещена сумма не менее 10млн.руб. Сохранить созданный макрос под именем *Клиенты*.

#### Выполнение задания

1. Для создания макроса необходимо реализовать следующий перечень макрокоманд:

Макрокоманда	Аргументы макрокоманды	
ОткрытьТаблицу	Имя таблицы	Клиенты-счета
	Режим	Таблица
	Режим данных	Изменение
ПрименитьФильтр	Имя фильтра	
	Условие отбора	<code>[Клиенты-счета]![Сумма на счете]&gt;10000000 And [Клиенты-счета]![Код счета] Like "P*"</code>

2. Сохраните макрос и проверьте его работу.

**Задание 5.** Создать групповой макрос, позволяющий выводить диалоговое окно с двумя кнопками *[Да]* и *[Нет]* и с вопросом к пользователю о виде открываемой формы *Клиенты-Счета*. По нажатии кнопки *[Да]* должна открываться эта форма без полей *№ менеджера* и *Начисления в месяц*, а по

нажатии на кнопку *[Нет]* – открываться с указанными полями. Макрос сохранить с именем *Скрытие полей в форме*.

### Выполнение задания

1. Откройте окно *Конструктора макросов*. Добавьте столбец «Условие» при помощи команды: *Вид → Условия*.

2. Для создания макроса необходимо реализовать следующий перечень макрокоманд:

	Условие	Макро-команда	Аргументы макрокоманды	
1	MsgBox("Открыть форму Клиенты-Счета без полей [№ менеджера], [Начисления в месяц]?";4)=6	<b>Открыть Форму</b>	Имя формы	Клиенты-счета
			Режим	Форма
			Имя фильтра	
			Условие отбора	
			Режим данных	
			Режим окна	Обычное
	В качестве первого аргумента функции MsgBox указан текст, который должен выводиться в диалоговом окне; в качестве второго аргумента – число 4, которое указывает, что в окне должны быть две командные кнопки [Да] и [Нет]. Если функция возвращает число 6, то это означает, что должна быть нажата кнопка [Да]; если не число 6, то должна быть нажата кнопка [Нет]. Условие сформируйте с помощью построителя выражений.			
2	...	<b>Задать Значение</b>	Элемент	[Формы]![Клиенты-счета]![№ менеджера].[Visible]
			Выражение	Нет
	Для того, чтобы поле [№ менеджера] не отображалось, его свойству Visible (Отображать) присваивается значение Нет. Выбор этого свойства осуществите в Построителе выражений по схеме: папка Forms→папка Все формы→форма Клиенты-Счета→поле № менеджера→свойство Visible.			
3	...	<b>Задать Значение</b>	Элемент	[Формы]![Клиенты-счета]![Начисления в месяц].[Visible]
			Выражение	Нет
4		<b>Открыть форму</b>	Имя формы	Клиенты-счета
			Режим	Форма
			Имя фильтра	
			Условие отбора	
			Режим данных	
			Режим окна	Обычное
Если функция MsgBox возвращает число 6, то выполняются команды 2,3; в противном случае – команда 4. Обратите внимание, что перед макрокомандами				

2,3, выполняемых в случае истинности условия, в столбец Условие вводится многоточие (...).

**Задание 6.** Создать макрос, позволяющий повысить процентные ставки по валютным счетам на 3 пункта. Макрос должен обеспечивать просмотр процентных ставок до и после увеличения. Сохранить макрос с именем *Повышение % ставок*.

### Выполнение задания

1. Откройте окно *Конструктора макросов*. Добавьте столбец «Имя макроса»: Вид → Имена макросов.

2. Для создания макроса следует реализовать следующий перечень макрокоманд:

	Имя макроса	Макрокоманда	Аргументы макрокоманды	
1		<b>Открыть Таблицу</b>	Имя таблицы	Счета
			Режим	Таблица
			Режим данных	Изменение
2		<b>Применить Фильтр</b>	Имя фильтра	
			Условие отбора	[Счета]![Код счета] Like "B*"
В таблице Счета отбираются только валютные счета. Условия отбора задайте в построителе выражений.				
3		<b>Сообщение</b>	Сообщение	После просмотра таблицы Счета до повышения % ставок нажмите ОК
			Сигнал	Да
			Тип	Информационное
			Заголовок	Повышение % ставок
4		<b>Открыть Форму</b>	Имя формы	Счета
			Режим	Форма
			Имя фильтра	
			Условие отбора	[Счета]![Код счета] Like "B*"
			Режим данных	
			Режим окна	Обычное
Открывается форма Счета для отображения записей таблицы Счета, отобранные по условию.				
5		<b>Запуск Макроса</b>	Имя макроса	Повышение % ставок.Повышение
			Число повторов	=DCount("*";"Счета";"[Код счета] like 'B*'" )
			Условие отбора	

	Запускается на выполнение макрос <b>Повышение</b> , который увеличивает процентные ставки по валютным счетам на 3 пункта. Он входит в состав макроса <b>Повышение % ставок</b> . Поэтому его имя является составным <b>Повышение % ставок.Повышение</b> . Макрос <b>Повышение</b> выполняется столько раз, сколько указано в аргументе «Число повторов». Число повторов подсчитывается с помощью функции DCount и равно количеству записей по валютным счетам в таблице Счета. Выражение для расчета введите в окне Область ввода.			
6		<b>Заккрыть</b>	Тип объекта	Форма
	Имя объекта		Счета	
	Сохранение		Подсказка	
7		<b>Применить Фильтр</b>	Имя фильтра	
	Условие отбора		[Счета]![Код счета] Like "B*"	
В таблице Счета отбираются записи по валютным счетам после повышения процентных ставок.				
8		<b>Сообщение</b>	Сообщение	После просмотра новых % ставок по валютным счетам нажмите ОК
			Сигнал	Да
			Тип	Информационное
			Заголовок	Повышение % ставок
9		<b>Заккрыть</b>	Тип объекта	Форма
			Имя объекта	Счета
			Сохранение	Подсказка
1	<b>Повыше ние</b>	<b>Задать Значение</b>	Элемент	[Формы]![Счета]![% ставка]
			Выражение	[Формы]![Счета]![% ставка]+3
Полю [% ставка] в текущей записи формы Счета присваивается новое значение.				
2		<b>НаЗапись</b>	Тип объекта	
			Имя объекта	
			Запись	Следующая
			Смещение	
Осуществляется переход на следующую запись формы Счета.				
3		<b>Остановить Макрос</b>		
Происходит останов работы макроса <b>Повышение</b> и возврат к макросу <b>Повышение % ставок</b> .				

**Задание 7.** Для открытия нового счета в банке необходимо ввести данные в форму **КЛИЕНТЫ-СЧЕТА**, но информация о клиенте еще отсутствует в таблице **КЛИЕНТЫ** базы данных «Банк». Это может быть выполнено следующим образом: открывается таблица **КЛИЕНТЫ**, добавляется новая запись, таблица закрывается. После этого можно продолжать работу с формой. Создать макрос, позволяющий автоматизировать эту операцию.

### Выполнение задания

1. Нажмите кнопку *Создать* для открытия окна создания макросов.
2. В столбце *Макрокоманда* щелкните в первой строке, раскроется список макрокоманд и выберите команду *ОткрытьТаблицу*.
3. В области *Аргументы макрокоманды*, раскрывая списки выбора значений, задайте аргументы как на рисунке 12.2.
4. Сохраните макрос под именем *Добавление записей в таблицу Клиенты* нажмите кнопку *Запуск*. Результатом работы будет открытие таблицы *Клиенты* в режиме добавления записей, т.е. с единственной пустой строкой.
5. Закройте окно таблицы КЛИЕНТЫ и окно создания макроса.
6. Чтобы макрос можно было запускать из формы *Клиенты-счета*, откройте форму КЛИЕНТЫ-СЧЕТА в режиме конструктора.

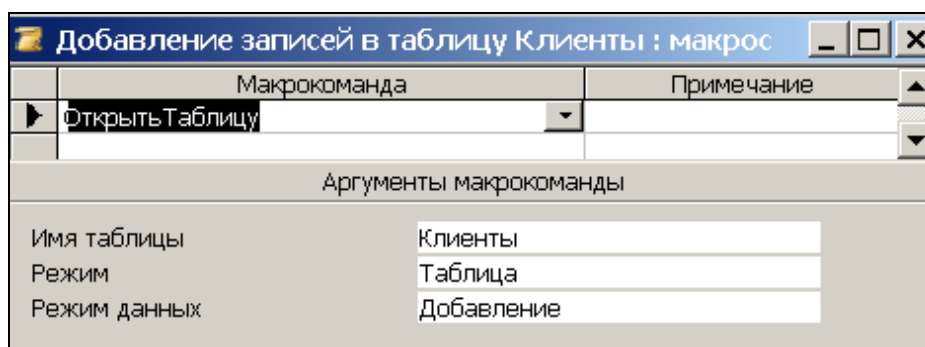


Рис. 12.2 - Окно Конструктора макросов

7. На первом шаге мастера выберите категорию «*Разное*», а действие – «*Выполнить макрос*», после чего нажмите кнопку «*Далее*».
8. На втором шаге будет отображен список макросов базы данных – выберите макрос *Добавление записей в таблицу КЛИЕНТЫ* и нажмите кнопку «*Далее*».
9. На следующем шаге мастера задайте текст, отображаемый на кнопке (например, «*Новый клиент*»), и нажмите кнопку «*Далее*», а на последнем шаге нажать кнопку «*Готово*», оставив все без изменения.
10. Поместите кнопку в удобном месте области данных, сохраните форму (рис.12.3) и перейдите в режим ее просмотра. Проверьте корректность работы кнопки, макроса и закройте форму.

Рис. 12.3 - Окно формы «Клиенты-счета»

### Задания для самостоятельной работы

1. Создать линейный макрос для выделения таблицы КЛИЕНТЫ. Сохранить созданный макрос под именем Клиенты.
2. Создать макрос, позволяющий повысить комиссионные на 2% менеджерам, работающим в филиале № 1. Макрос должен обеспечивать просмотр комиссионных до и после увеличения. Сохранить макрос с именем *Повышение комиссионных*
3. Для открытия нового счета в банке необходимо ввести данные в форму «Клиенты-счета», но информация по счетам и менеджерам пока отсутствует в таблицах МЕНЕДЖЕРЫ и СЧЕТА базы данных. Это может быть выполнено следующим образом: открывается таблица МЕНЕДЖЕРЫ, добавляется новая запись, таблица закрывается. Аналогичным образом добавляются данные по счетам. После этого можно продолжать работу с формой «Клиенты-счета». Создать макрос, позволяющий автоматизировать эти операции.



## Лабораторная работа № 13

### СОЗДАНИЕ ГЛАВНОЙ КНОПОЧНОЙ ФОРМЫ

**Цель работы:** научиться создавать кнопочные формы в СУБД MS ACCESS и формы в клиентском приложении разрабатываемой автоматизированной информационной системы.

#### Порядок выполнения работы

**Задание 1. Создание страниц кнопочной формы с помощью диспетчера.**

Создайте *главную кнопочную форму*, которая будет использоваться как панель управления приложением *БАНКЪ*, структура которого представлена на рис. 13.1.



Рис.13.1 – Пример структуры приложения

Определите в соответствии со структурой приложения иерархическую взаимосвязь главной и подчинённой страниц кнопочной формы приложения и состав кнопок каждой из страниц. В обобщенном виде страницы кнопочной формы *БАНКЪ* приведены на рис. 13.2.

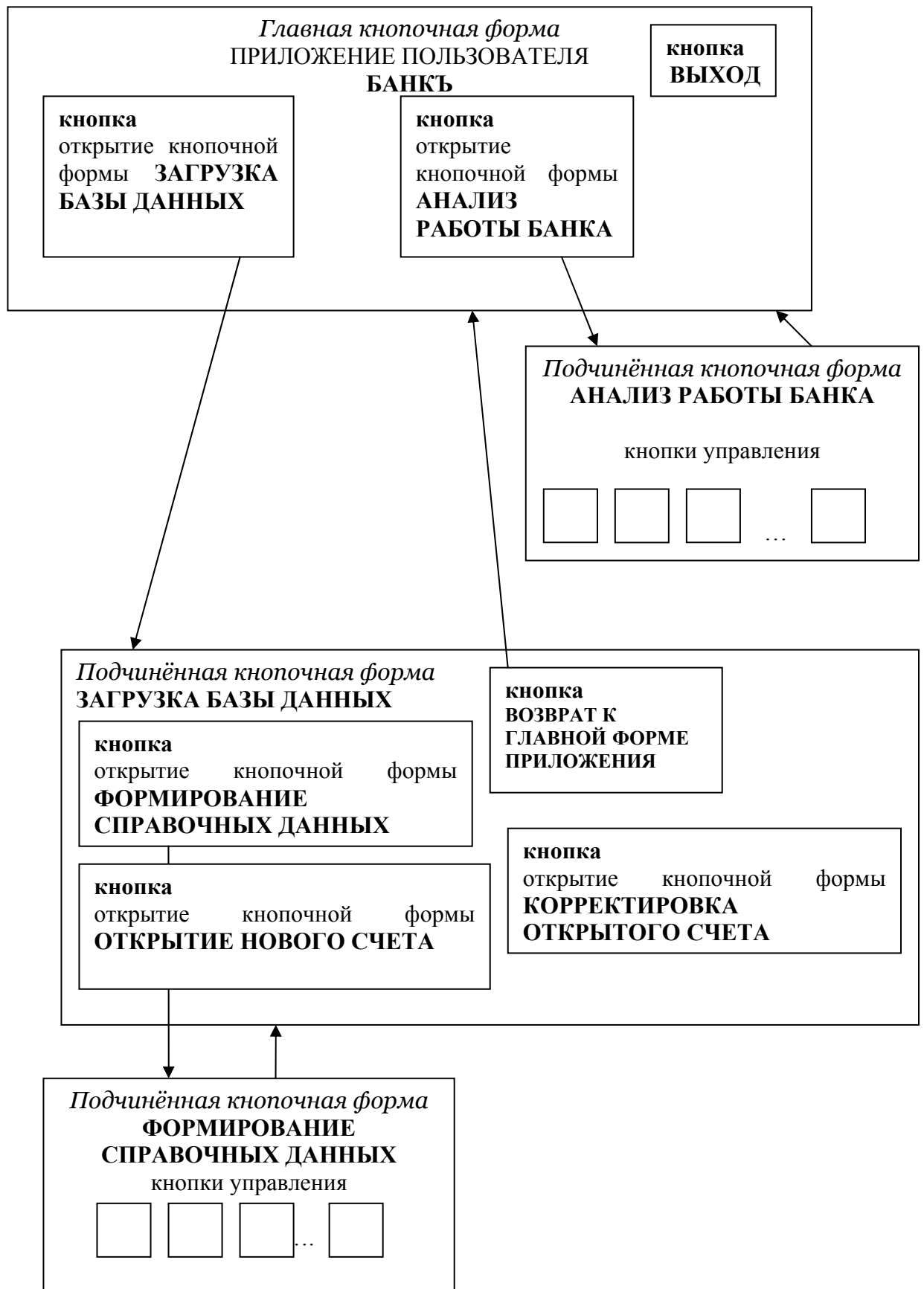


Рис.13.2 – Взаимосвязь главной и подчиненной кнопочной форм

### Выполнение задания

1. Выберите команду *Сервис* → *Служебные программы* → *Диспетчер кнопочных форм*.

2. В окне *Диспетчера кнопочных форм* подтвердите создание кнопочной формы в базе данных.

3. В следующем окне *Диспетчера кнопочных форм* появится перечень страниц кнопочной формы, содержащей пока одну строку *Главная кнопочная форма (По умолчанию)*(рис. 13.3).

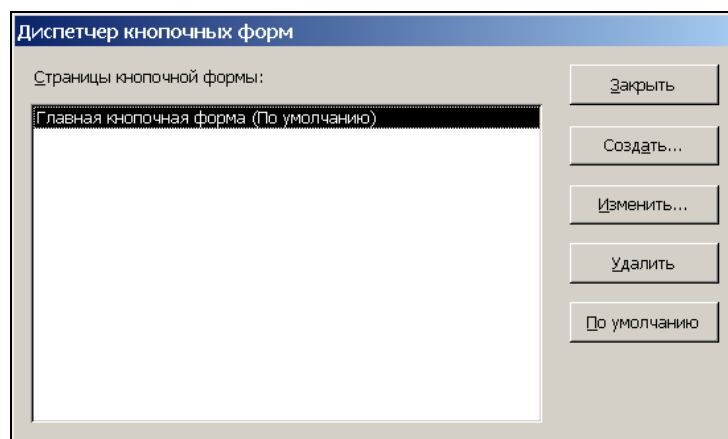


Рис.13.3 – Окно списка страниц кнопочных форм приложения

4. Для создания других страниц кнопочной формы нажмите кнопку *Создать*.

5. В окне *Создание* (рис. 13.4) в поле *Имя страницы кнопочной формы* введите имя новой страницы кнопочной формы *Загрузка базы данных*.

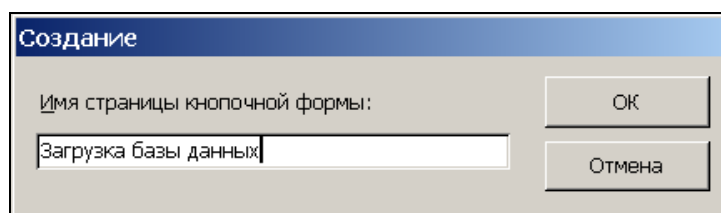


Рис.13.4 – Окно создания пустой кнопочной формы

6. Нажмите кнопку *ОК*, в окне диспетчера кнопочных форм появится строка с заданным именем новой страницы кнопочной формы.

7. Аналогичным образом создайте страницу кнопочной формы *Анализ работы банка*. В результате в окне диспетчера кнопочных форм отобразятся имена созданных страниц (рис. 13.5).

8. Чтобы любую из подготовленных форм сделать стартовой, выделите её и нажмите кнопку *По умолчанию*. Оставим главную кнопочную форму стартовой по умолчанию.

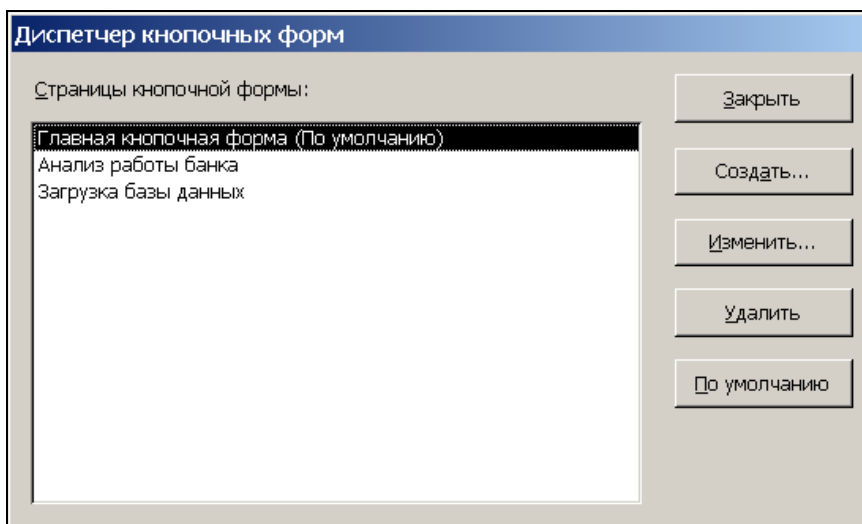


Рис.13.5 –Список страниц кнопочных форм приложения БАНКЪ

**Задание 2. Формирование элементов взаимосвязи страниц кнопочной формы.** Для навигации по страницам кнопочной формы создайте кнопки перехода от главной кнопочной формы к обеим её подчиненным.

#### Выполнение задания

1. В окне *Диспетчера кнопочных форм* выделите нужную строку и нажмите кнопку *Изменить*.

2. В окне диалога *Изменение страниц кнопочной формы* (рис.13.6) переименуйте главную кнопочную форму в *Приложение БАНКЪ*, введя это имя в поле *Название кнопочной формы*.

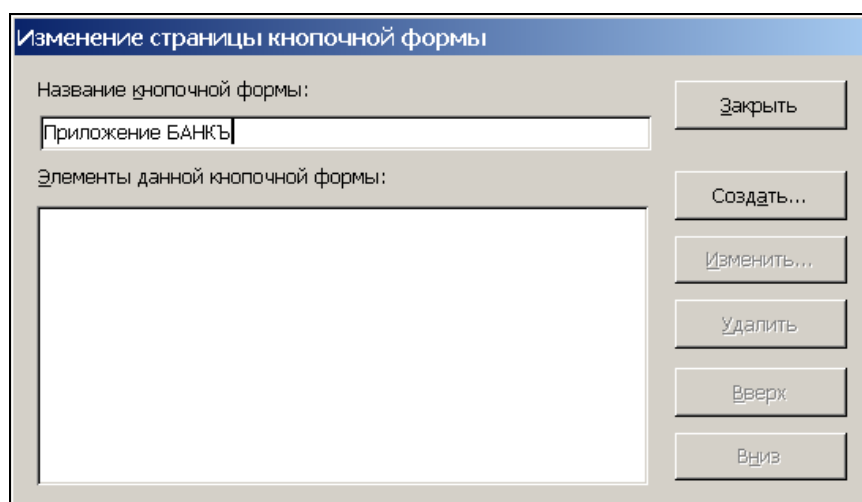


Рис.13.6 –Окно редактирования имени кнопочной формы и создание её кнопок

3. Для формирования в главной кнопочной форме *Приложение БАНКЪ* кнопки вызова подчиненной страницы кнопочной формы нажмите кнопку *Создать*.

4. В окне *Изменение элемента кнопочной формы* в строке *Команда* выберите *Перейти к кнопочной форме* (рис.13.7).

5. В строке *Кнопочная форма* откройте список созданных приложений страниц кнопочной формы и выберите форму – *Загрузка базы данных*, к которой должна обеспечивать переход создаваемая кнопка текущей формы.

6. В строке *Текст* введем подпись для этой кнопки *Загрузка базы данных* (рис.13.7) и нажимаем *ОК* для завершения формирования элемента.

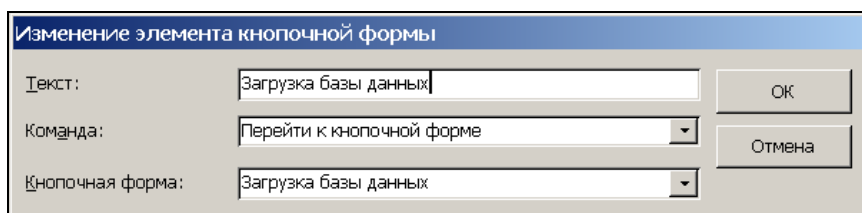


Рис.13.7 – Окно создание элемента кнопочной формы

7. Аналогичным образом создайте кнопку вызова другой подчиненной формы и задайте подпись *Анализ работы банка*.

8. Для обеспечения перехода к редактированию созданных кнопочных форм создадим в главной кнопочной форме кнопку с именем *Изменение главной формы*, выбрав для нее команду *Конструктор приложения* (рис. 13.8). В дальнейшем это позволит выбрать в любой момент *Диспетчер кнопочных форм* и внести необходимые изменения.

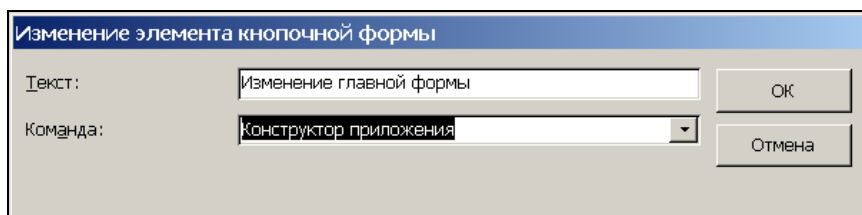


Рис.13.8 – Окно создание элемента кнопочной формы

9. Для завершения работы с приложением создайте кнопку с именем *Выход из приложения*, выбрав для нее команду *Выйти из приложения*.

Перечень созданных элементов главной кнопочной формы *Приложение БАНКЪ* представлен на рис.13.9.

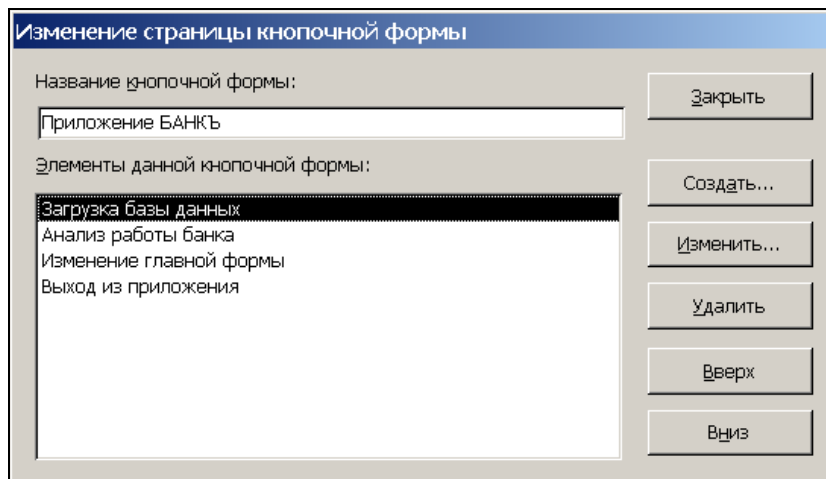


Рис.13.9 – Страница главной кнопочной формы

10. Закончив создание страницы главной кнопочной формы, нажмите кнопку *Закреть*. Это позволит вернуться к списку всех кнопочных форм. Закройте окно диспетчера кнопочных форм. В списке форм окна базы данных появится форма с именем *Кнопочная форма (Switchboard)*. В списке таблиц окна базы данных появится таблица *Switchboard Items (Элементы кнопочной формы)*, содержащая описание кнопок, выводящихся в форме, и выполняемых ими действий.

11. Откройте кнопочную форму. На экране отобразится главная кнопочная форма с созданными элементами (рис.13.10).

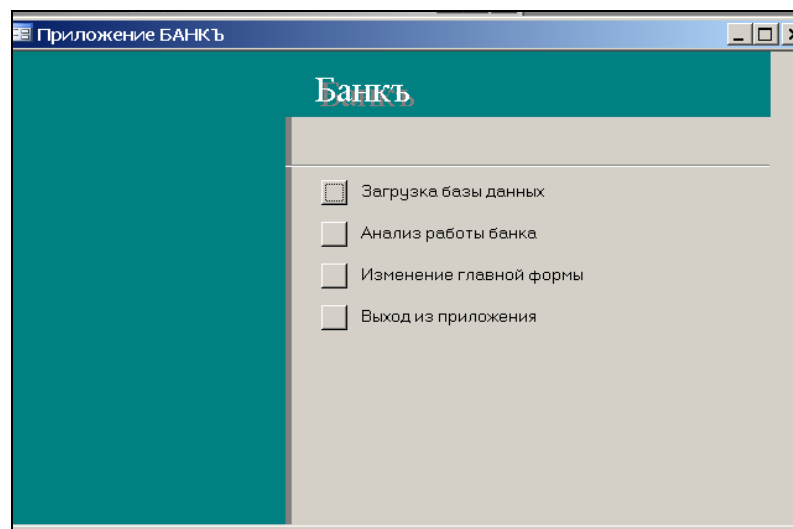


Рис.13.10 – Главная кнопочная форма приложения БАНКЪ

12. Для продолжения разработки кнопочной формы нажмите кнопку *Изменение главной формы*.

13. В подчиненных страницах кнопочной формы создайте кнопку для возврата к главной кнопочной форме, заполнив поля в окне *Изменение элемента кнопочной формы*, как показано на рис.13.11.

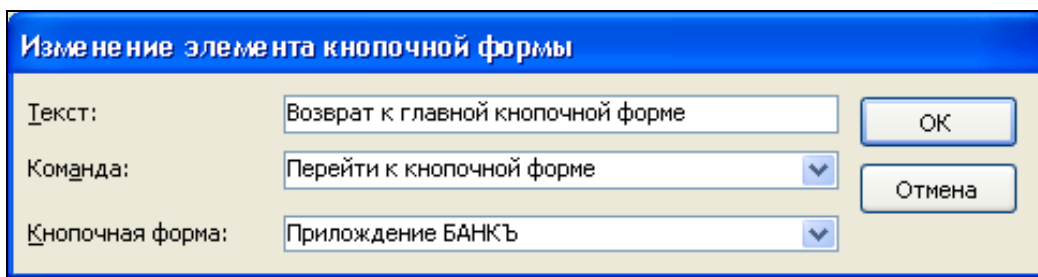


Рис.13.11 – Создание кнопки для возврата в главную форму

Таким образом, могут быть созданы все связи между страницами кнопочной формы, представленными в списке окна диспетчера кнопочных форм.

### **Задание 3. Формирование элементов для открытия объектов приложения.**

Для формирования кнопок, обеспечивающих выполнение пользователем своих задач, свяжите их с одним из объектов приложения, т.е. создайте на странице *Загрузка базы данных* кнопки для работы с формой *Клиенты-счета*, одну для ввода данных из документа открытия счета, а другую для просмотра и редактирования хранящегося в базе договора на открытие счета.

#### **Выполнение задания**

1. Для формирования элементов на странице кнопочной формы в окне диспетчера кнопочных форм в списке страниц выберите *Загрузка базы данных* (рис.13.5).

2. Для создания элементов в этой форме нажмите *Изменить*.

3. В открывшемся окне *Изменение страниц кнопочной формы*, аналогичном представленному на рис.13.6, нажмите кнопку *Создать*.

4. В окне *Изменение элемента кнопочной формы* в раскрывающемся списке *Команда* выберите *Открыть форму для добавления*. Эта команда открывает форму в режиме ввода новых записей, не предоставляя возможности просматривать существующие.

5. В соответствие с выбранной командой, появится третья строка *Форма*. Откройте список доступных форм и выберите форму *Клиенты-счета*. В строку *Текст* введите *Открытие новых счетов*. Подпись кнопки должна по возможности кратко и точно называть действия, которые будут выполняться при нажатии кнопки.

6. Аналогичными действиями создайте кнопку для просмотра и редактирования открытых счетов. Для связи этой кнопки с объектом выберите команду *Открыть форму для изменения*.

7. Создайте на странице *Анализ работы банка* кнопку *Итоги по банку* для открытия запроса *Итоги по банку* в режиме таблицы.

8. Так как в списке команд, которые могут выполняться при нажатии кнопки на странице кнопочной формы, нет команды открытия запроса,

создайте макрос, обеспечивающий это действие. Такой макрос с именем *Итоги по банку* показан на рис.13.12.

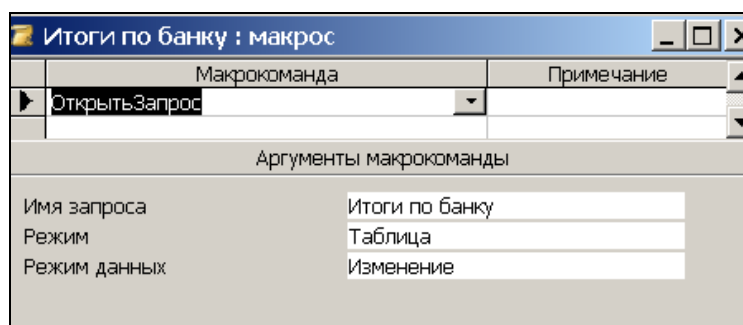


Рис.13.12 –Макрос, обеспечивающий открытие запроса в режиме таблицы

9. При создании кнопки *Итоги по банку* в окне *Изменение элемента кнопочной формы* (рис.13.13) выберите команду *Выполнить макрос* и в строке *Макрос* имя созданного макроса.

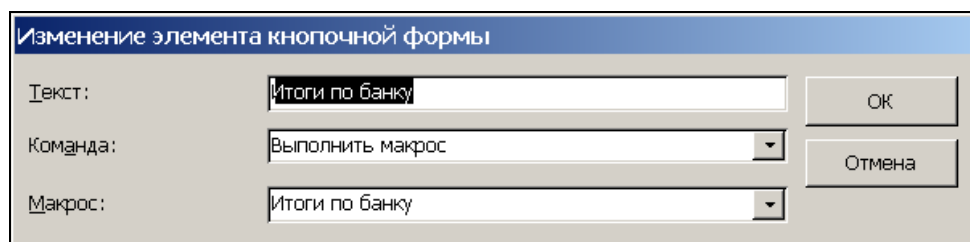


Рис.13.13 –Выбор команды и макроса для открытия таблицы

#### **Задание 4. Добавление рисунка в кнопочную форму**

##### **Выполнение задания**

1. Откройте кнопочную форму в режиме конструктора. В левой части области данных стандартной кнопочной формы уже размещен элемент управления типа *Рисунок*, в котором рисунок отсутствует, но сохранена рамка для его размещения.

2. Щелкните кнопкой мыши в левой части формы, выделится рамка рисунка. Нажмите правую клавишу мыши и откройте свойства рисунка.

3. Путь к рисунку выберите с помощью построителя, который вызывается нажатием кнопки построителя в конце строки *Рисунок*.

4. В открывшемся окне *Выбор рисунка* откройте папку, в которой храниться файл с нужным рисунком.

5. Измените в свойствах рисунка *Внедренный* на *Связанный*. В результате рисунок в кнопочной форме всегда будет соответствовать сохраняемому в связанном файле. Кнопочная форма приложения приведена на рис.13.14.



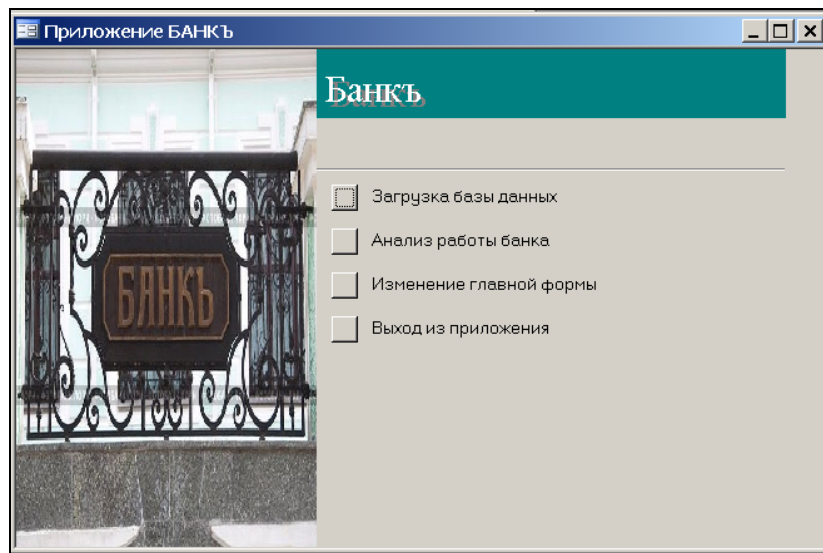


Рис.13.14 – Кнопочная форма приложения с рисунком

## **Задание 5. Выбор параметров запуска приложения**

### **Выполнение задания**

1. Откройте окно параметров запуска приложения, выполните команду *Сервис* → *Параметры запуска*.
2. Для открытия главной кнопочной формы сразу при открытии файла базы данных в окне параметров запуска в строке *Вывод формы/страницы* выберите *кнопочную форму*.
3. Для того, чтобы пользователю запретить просмотр окна базы данных снимите флажок *Окно базы данных* и флажок *Специальные клавиши*.
4. Чтобы пользователь не мог зайти в режим конструктора объекта базы данных Access и изменить его структуру и характеристики через команды меню, надо скрыть помимо окна базы данных, и некоторые меню. Снимите флажок *Полный набор меню*. Теперь для приложения будет выводиться ограниченный набор меню *Файл, Правка, Вставка, Записи, Окно*. Здесь не включают меню *Вид* и *Сервис*. Чтобы вернуть полный набор меню, при открытии приложения Access удерживайте в нажатом положении клавишу *<Shift>*, пока не будут закрыты все окна с предупреждениями о безопасности.

### **Задания для самостоятельного выполнения**

**Задание 1.** На странице кнопочной формы *Загрузка базы данных* создайте кнопку *Формирование справочных данных* для перехода к подчиненной странице *Справочные данные* рис.13.15. На этой подчиненной форме предусмотрите кнопки для открытия форм справочной информации по клиентам, менеджерам и счетам, а также кнопку возврата к странице *Загрузка базы данных*.

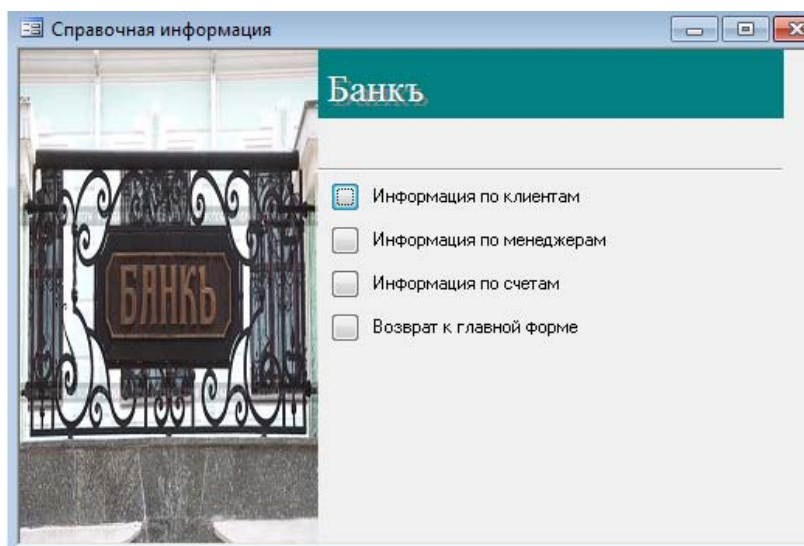


Рис.13.15 – Вид подчиненной страницы *Справочные данные*

**Задание 2.** Создайте на странице кнопочной формы *Анализ работы банка*, 3-4 кнопки, позволяющие выполнить переход к отчетам по результатам анализа деятельности банка (например, итоги по каждому менеджеру, анализ вкладов за истекшие месяцы 2009 года, популярность у вкладчиков разных видов валют и т.д.). Для активизации работы кнопок создать соответствующие макросы.

---

---

## Лабораторная работа № 14

### СОЗДАНИЕ СТРАНИЦ ДОСТУПА К ДАННЫМ

---

---

**Цель работы:** изучить возможности СУБД ACCESS по созданию страниц доступа к данным с помощью мастера и средствами конструктора.

#### Теоретические сведения

В Access возможно создание статических Web-страниц и страниц доступа к данным.

*Статическая Web-страница* – это Web-страница, которая отображает данные из базы в состоянии на момент создания Web-страницы. Ее можно создавать для объектов базы данных – таблицы, запроса, отчета, экспортируя их содержимое в HTML-файл.

*Страница доступа к данным* – это диалоговая Web-страница, которая поддерживает динамическую связь с базой данных и позволяет просматривать, редактировать и вводить данные в базу. *Страница доступа к данным* сохраняется не в базе, а в HTML-файле. В базе данных храниться лишь указатель на этот файл – ярлык. В базе данных Access страница доступа к данным так же, как и форма или отчет, может быть создана с помощью мастера или в режиме конструктора.

#### Порядок выполнения работы

**Задание 1. Создание статической Web-страницы к данным одной таблицы.** Создайте статическую *Web-страницу* для таблицы МЕНЕДЖЕРЫ и сохраните ее в своей папке.

##### Выполнение задания

1. В окне базы данных *Банкь.mdb* выделите таблицу МЕНЕДЖЕРЫ.
2. Выполните команду: *Файл→Экспорт*.
3. В окне *Экспорт объекта* укажите свою папку для сохранения файла HTML. Оставьте заданное имя файла Менеджеры.
4. Выберите из раскрывающегося списка «Тип файла» *Документы HTML (HTML Documents)*, установите флажок «*Сохранить формат*» и нажмите *Сохранить*.
5. В окне *Параметры вывода в формате HTML* нажмите *ОК*.
6. Для загрузки созданной статической Web-страницы в Internet Explorer откройте файл МЕНЕДЖЕРЫ из своей папки.

## **Задание 2. Создание страницы доступа к данным одной таблицы.**

Создайте страницу доступа к данным для работы с таблицей КЛИЕНТЫ с помощью мастера автостраниц.

### **Выполнение задания**

1. Откройте базу данных *БанкЪ*.
2. В окне базы данных с списке *Объекты* выберите объект *Страницы* и нажмите *Создать*.
3. В диалоговом окне *Новая страница доступа к данным* выберите *Автостраница: в столбце*.
4. Выберите таблицу КЛИЕНТЫ, на данных которой должна быть основана страница, и нажмите *ОК*.
5. Перейдите в режим конструктора. Введите название страницы и исправьте надписи полей, как показано на рис. 14.2, воспользовавшись средствами, аналогичными тем, что были в конструкторах форм и отчетов.

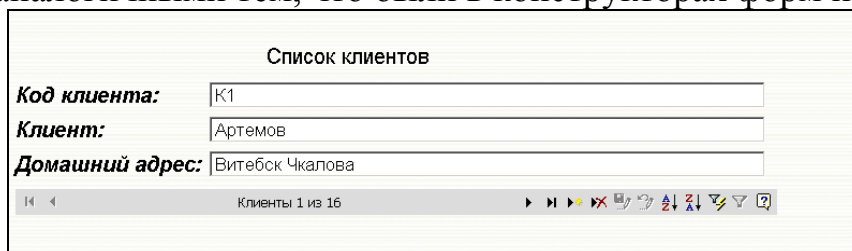


Рис.14.1 – Страница доступа к данным таблицы КЛИЕНТЫ после редактирования в режиме просмотра данных

6. Щелкнув на значке списка в разделе *Заголовок:Клиенты*, отметьте флажок в строке *Подпись*. Для создания в открывшемся разделе новой надписи используйте соответствующую кнопку панели элементов.

7. Сохраните созданную страницу доступа к данным под именем *Клиенты*. Страница сохраняется в виде HTML-файла, имеющего расширение *.htm*. Автоматически Access добавляет ярлык для этой страницы в окно базы данных. Чтобы узнать путь к файлу страницы, наведите указатель на ярлык страницы в окне базы данных и подождите появление всплывающей подсказки.

## **Задание 3. Создание страницы доступа к данным нескольких таблиц.**

Создайте страницу доступа к данным для просмотра открытых счетов, используя таблицы КЛИЕНТЫ-СЧЕТА, КЛИЕНТЫ, МЕНЕДЖЕРЫ, СЧЕТА. Для создания такой страницы воспользуйтесь мастером.

### **Выполнение задания**

1. Начните создание страницы с двойного щелчка мышью на строке *Создание страницы доступа к данным с помощью мастера* в рабочем пространстве объектов *Страницы*.
2. В окне мастера выберите таблицы КЛИЕНТЫ, СЧЕТА, КЛИЕНТЫ-СЧЕТА, МЕНЕДЖЕРЫ и необходимые поля.

3. Для того чтобы спецификация счета открывалась только при щелчке на маркере развертывания, добавьте уровень группировки по полю [№ счета], используя кнопку «>>» (рис. 14.2). Для образования уровня группировки мастер предлагает только выбор поля.

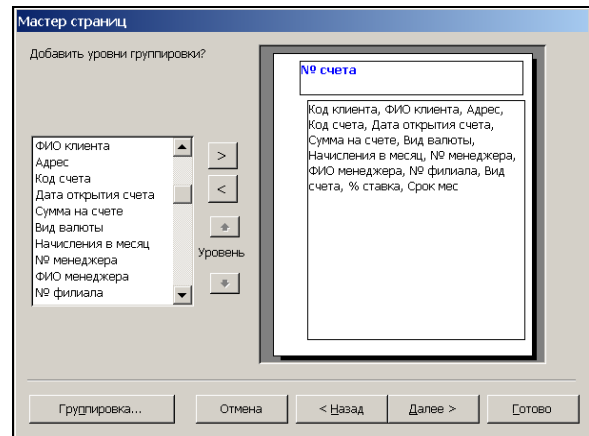


Рис.14.2 – Уровни группировки

4. Далее выберите поле [ФИО клиента], по которому будет производиться сортировка записей по возрастанию, нажмите *Далее* и установите флажок *Применить тему к странице?* Нажмите кнопку *Готово*. Откроются созданная страница в режиме конструктора и окно с большим списком тем. Каждая тема может быть просмотрена и подходящая выбрана для оформления страницы. Созданная мастером страница доступа к данным в режиме конструктора будет иметь вид, показанный на рис. 14.3.

5. Для перехода в режим просмотра страницы щелкните по кнопке панели инструментов *Вид*. Редактировать данные на данной странице невозможно.

6. Access при размещении на странице полей автоматически создает запрос SQL, который и становится источником ее записей. С помощью запрос на страницу из базы доставляются указанные в нем данные. На странице, созданной мастером, отображаются по одной записи запроса.

7. Доработайте страницу в режиме конструктора. Вид отредактированной страницы в режиме конструктора представлен на рис. 14.4.

8. Прежде всего переместите поля таблицы КЛИЕНТ в раздел *Заголовок:Счета-№ счета*. Это позволит сразу отображать информацию о клиенте, открывшем счет в банке и при необходимости с помощью маркера развертывания открывать его спецификацию.

9. Для изменения вида маркера развертывания выберите с помощью правой клавиши мыши *Свойства элемента* и в строке *Src* его свойство вместо значения *Черный плюс/минус* новое значение, например *Папка*.

10. Для добавления строки подписей столбцов в список счетов щелкните на стрелке *Заголовок: КлиентыWiz* и отметьте строку *Подпись*.

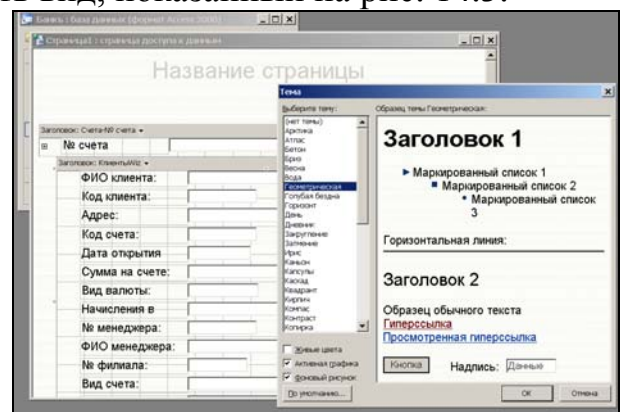


Рис.14.3 – Страница доступа к данным в режиме конструктора и окно с большим списком тем.

Щелкните по кнопке панели инструментов *Вид*. Страница с соответствующими доработками будет иметь вид, представленный на рис. 14.5.

**11.** Если необходимо, чтобы на странице отображался список всех договоров, уберите *Кнопки Перехода: Счета-№ счета*. При этом запись каждого маркера будет иметь маркер разворачивания, позволяющий просмотреть полную информацию по каждому счету.

**12.** Сохраните созданную страницу доступа к данным под именем *Сведения по счетам*.

Рис.14.4 – Доработанная страница в режиме конструктора


Рис.14.5 – Страница, отображающая спецификацию счета

#### **Задание 4. Создание страницы доступа к данным в режиме конструктора.**

Создайте в виде сводной таблицы страницу доступа к данным, позволяющую просматривать в интрасети сведения об ежемесячных начислениях клиентам в виде сводного списка по клиентам и менеджерам.

##### **Выполнение задания**

**1.** В окне базы данных выберите объект *Страницы* и щелкните по кнопке инструментальной панели окна *Создать*. В диалоговом окне *Новая страница ...* выберите вариант создания *с помощью конструктора*, а в качестве источника данных – таблицу *КЛИЕНТЫ-СЧЕТА* ==>*ОК*.

**2.** На расположенной в верхней части окна конструктора панели элементов щелкните по кнопке  *Сводная таблица Office*. Не отпуская левую клавишу мыши, очертите прямоугольный контур будущей сводной таблицы.

**3.** Перетащите мышкой из окна *Список полей* внутрь контура сводной таблицы в область деталей поле *Начисления в месяц*. В результате в области данных появится столбец с суммами начислений в месяц.

**4.** Раскройте в окне *Список полей* папку *Связанные таблицы* для таблицы *КЛИЕНТЫ-СЧЕТА*, а в ней списки полей таблиц *КЛИЕНТЫ*, *МЕНЕДЖЕРЫ*. Перетащите мышкой из окна *Список полей* внутрь контура сводной таблицы поле *ФИО менеджера* таблицы *МЕНЕДЖЕРЫ* в область

полей фильтра, поле *ФИО клиента* таблицы КЛИЕНТЫ в область полей строк. Закройте *окно Список полей*.

5. Подсчитайте общие суммы начислений за месяц у каждого клиента. Для этого выделите сначала поле *Начисления в месяц*, щелкнув по заголовку соответствующего столбца, а затем щелкните по кнопке *Автовычисления* на панели управления сводной таблицей. Выберите в меню пункт *Сумма*. На пересечении строки *Начисления в месяц* и *Общие итоги* появится общая сумма начислений за месяц. При этом для некоторых клиентов сумма начислений будет выводиться в двух строках. Отключите вывод вторых строк. Для этого щелкните по кнопке *Скрыть детали* на панели управления сводной таблицей (рис 14.6).

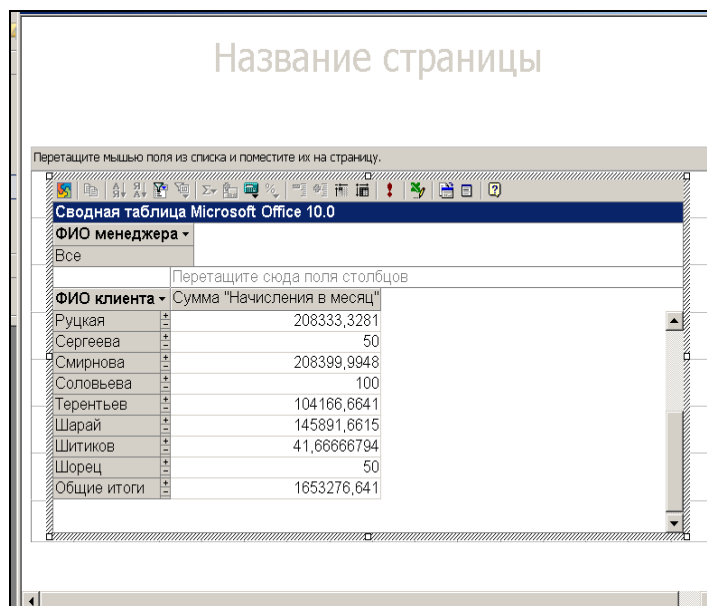


Рис.14.6 – Вид страницы доступа в режиме конструктора

6. Отключите вывод заголовка и областей сводной таблицы ==> кнопка *Команды и параметры* на панели управления сводной таблицей. Установите вкладку *Настройки* и снимите флажки в соответствующих пунктах. Закройте *окно Команды и параметры*.

7. Введите название страницы СВОДНАЯ ВЕДОМОСТЬ НАЧИСЛЕНИЙ ЗА МЕСЯЦ. Просмотрите страницу доступа, щелкнув по кнопке *Вид* инструментальной панели. Просмотрите сведения о начислениях для каждого клиента за месяц. Закройте страницу и сохраните под именем *Сводная ведомость начислений* в своей папке.

8. Просмотрите созданную страницу с помощью обозревателя Internet Explorer (см. рис. 14.7)

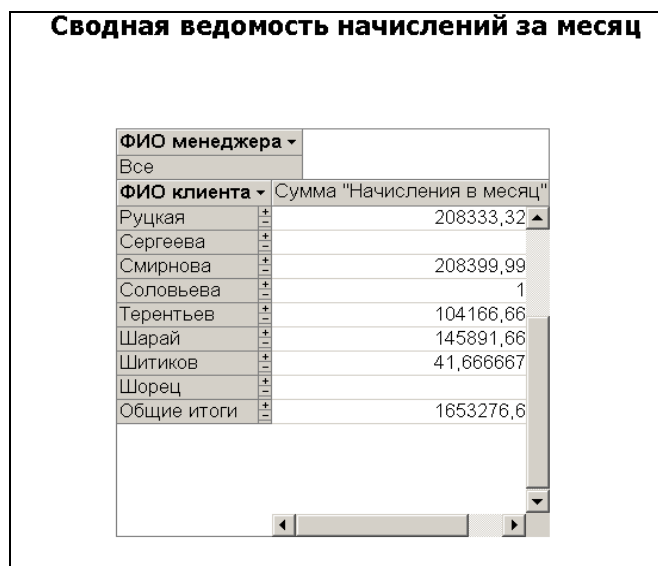


Рис.14.7 – Вид страницы доступа Сводная ведомость начислений

### **Задание 5. Создание гиперссылок на странице доступа к данным.**

По гиперссылкам на страницах доступа к данным осуществляется переход к другим страницам базы данных или проекта Access, к документам, созданным в других приложениях, таких как MS Word, MS Excel, MS Power Point и расположенных на локальных и сетевых дисках, на Web- и FTR-серверах в глобальной сети Интернет или в корпоративной сети Интранет.

#### **Выполнение задания**

1. Откройте страницу доступа к данным *Клиенты* в режиме конструктора.

2. Щелкните на кнопке панели элементов *Гиперссылка* и вставьте ее в нужное место страницы.

3. Создайте гиперссылку на другую страницу, подключенную к той же базе данных. Для этого в диалоговом окне *Добавление гиперссылки* в разделе *Связать с* щелкните на значке *страницей в базе данных*.

4. В поле *Текст* введите текст, который должен отображаться на странице.

5. Нажмите кнопку *Подсказка* и введите текст, который будет появляться при наведении указателя мыши на гиперссылку.

6. Выберите в списке страницу доступа к данным, ссылку на которую нужно создать, например, страницу *Сведения по счетам*. На странице доступа к данным будет создан элемент управления *Гиперссылка*.

7. Создайте на странице *Сведения по счетам* в разделе подписей гиперссылку на страницу *Клиенты*.

8. Закройте базу данных и Access. Чтобы проверить действие этой ссылки, откройте Internet Explorer и в нем страницу *Клиенты*. Щелкнув по гиперссылке *Сведения по счетам*, перейдите на соответствующую страницу. По гиперссылке на этой странице вернитесь на страницу *Клиенты*.

9. Создайте на странице доступа к данным *Клиенты* гиперссылку на текстовый файл. Снова откройте страницу доступа к данным *Клиенты* в режиме конструктора. Выполните команду *Вставка - Гиперссылка*.

10. В диалоговом окне *Добавление гиперссылки* в разделе *Связать с* щелкните на значке *файлом, веб-страницей*.

11. В поле *Текст* введите текст, который должен отображаться на странице. Если нужно введите подсказку.

12. В поле *Адрес* введите путь к файлу.

13. Для добавления гиперссылки на страницу в Интернете щелкните в разделе *Связать с* на значке *файлом, веб-страницей*. Нажмите кнопку *Интернет*, чтобы перейти на нужную страницу. После открытия нужной страницы переключитесь из Internet Explorer в Access. Заголовок и URL-адрес текущей открытой Web-страницы подставились в поля *Текст* и *Адрес*.



## **Задание 6. Добавление гиперссылки для создания сообщений электронной почты**

### **Выполнение задания**

- 1.** Откройте страницу доступа к данным *Клиенты* в режиме конструктора и с помощью кнопки панели элементов *Гиперссылка* начните создание гиперссылки.
- 2.** В диалоговом окне *Добавление гиперссылки* в разделе *Связать с* щелкните на значке *электронной почтой*.
- 3.** В поле *Текст* введите: *Здесь вы можете отправить письмо клиенту*. Если оставит поле пустым, то *Access* в качестве текста выведет адрес электронной почты клиента. Если нужно введите подсказку.
- 4.** В поле *Адрес эл.почты* введите адрес электронной почты клиента, например, *lenovo@mail.ru*.
- 5.** В поле *Тема* введите текст, который должен появляться в почтовом сообщении в поле *Тема*.
- 6.** Переключитесь в режим страницы и щелкните гиперссылку. *Access* откроет установленную почтовую программу и окно нового сообщения с заполненными полями *Кому* и *Тема*.

### **Задания для самостоятельного выполнения**

**Задание 1.** Создайте статическую Web-страницу для таблицы *МЕНЕДЖЕРЫ* и сохраните ее в своей папке.

**Задание 2.** Сконструируйте страницу доступа к данным, на которой можно вводить, редактировать и удалять данные из таблицы *МЕНЕДЖЕРЫ*. Заголовок страницы отобразить бегущей строкой. Страницу сохранить под именем *Сведения о менеджерах* в своей папке.

**Задание 3.** Создайте в виде сводной таблицы страницу доступа к данным, позволяющую просматривать размеры вкладов клиентов по видам счетов и менеджерам, которые их обслуживают. Создание страницы произвести с помощью конструктора.

---

## Лабораторная работа №15

### ВЗАИМОДЕЙСТВИЕ СУБД ACCESS С ПРИЛОЖЕНИЯМИ MS OFFICE

---

**Цель работы:** Изучить механизмы взаимодействия приложений Microsoft Office с СУБД Access; приобрести опыт использования информации из БД при подготовке серийных документов в Microsoft Word; изучить механизмы *импорта и экспорта данных в базу*, приобрести опыт анализа информации из БД средствами Microsoft Excel.

#### Порядок выполнения работы

##### **Задание 1. Подготовить серийный документ в MS Word**

Используя команду *Публикация в MS Word*, (меню *Сервис/Связи с Office*), сохранить каждую из таблиц базы данных *БанкЪ* в MS Word.

##### **Задание 2. Проанализировать информацию из БД средствами MS Excel:**

1. Составите перекрестный запрос, подводящий итоги по открытым счетам за каждый месяц работы банка (см. лабораторную работу №8).
2. Используя команду *Анализ в MS Excel* (из меню *Сервис/Связи с Office*), сохранить этот запрос в виде рабочей книги MS Excel.

##### **Задание 3. Проанализировать информацию из БД средствами MS Excel:**

1. В режиме *Мастера простого запроса*, сформировать запрос на выборку, позволяющий отобразить всю информацию из базовых таблиц. Сохранить запрос под именем *Исходные данные*. Названия полей, включаемых в запрос, приведены в лабораторной работе № 2 (таблица 1).
2. Используя команду *Анализ в MS Excel* (из меню *Сервис/Связи с Office*), сохранить запрос "Исходные данные" в виде рабочей книги *MS Excel*.
3. На основе полученного в *MS Excel* списка сформировать компактную сводную таблицу по усмотрению (пример см. на рис.15.1).

Сумма по полю	Сумма на счете	ФИО менеджера			
Вид счета	Гусева	Львова	Николаев	Полякова	Общий итог
Валютный депозит		23000	13000	5000	41000
Валютный срочный	20000	5000		16000	41000
Валютный текущий		10000			10000
Рублевый депозит	8000000			6000000	14000000
Рублевый срочный	5000000		42000000	10000000	57000000
Рублевый текущий			25000000		25000000
Общий итог	13020000	38000	67013000	16021000	96092000

Рис. 15.1 – Сводная таблица

#### **Задание 4. Подготовить стандартное серийное письмо для массовой рассылки.**

1. Использовать команду *Слияние с MS Word*, меню *Сервис/Связи с Office*.

2. Содержание письма должно соответствовать направлению разработанной базы данных *БанкЪ*: приглашение на презентацию нового программного продукта, информационное письмо о сезонных скидках или распродажах, приглашение на юбилей компании и т.д.

3. Предусмотреть в тексте письма поля *слияния MS Access (адрес, ФИО адресата, личный код адресата)* для подстановки данных из исходной таблицы индивидуальной БД.

#### **Задание 5. Осуществить импорт данных из MS Excel в базу данных**

##### **Выполнение задания:**

1. На рабочем листе MS Excel создать таблицу *Филиалы* с полями [*№ филиала*], [*Название филиала*], *адрес, телефон, УНП, [Описание услуг]* (см. лабораторную работу №5).

2. Сохранить файл Excel под именем *Филиалы*.

3. Открыть базу данных *БанкЪ*.

4. Для выполнения импорта данных с рабочего листа MS Excel выбрать меню *Файл ⇒ Внешние данные ⇒ Импорт*.

5. В открывшемся окне в строке *тип файлов* выбрать Microsoft Excel, а в строке *имя файла* указать имя рабочей книги, откуда импортируются данные, в данном случае *Филиалы.xls*, ⇒ кнопка «*Импорт*».

6. Далее следовать указаниям мастера.

7. Определить имя импортируемой таблицы *Филиалы*.

*Примечание:* При импорте данных Access считывает данные из файла вне текущей базы данных и создает для них новую таблицу. В Access можно импортировать файлы нескольких типов:

- объекты из других баз Access;
- файлы dBASE;
- файлы FoxPro;
- электронные таблицы Excel;
- текстовые файлы (с разделителями или с фиксированной длиной записей);

- документы HTML;
- базы данных ODBC, такие как таблицы Microsoft SQL Server.

Процедура импорта для всех типов файлов единообразна.

1. Активизировать динамическое меню в области окна базы данных (щелчок правой кнопкой мыши) и выбрать в контекстном меню команду *Импорт* или команду *Файл ==> Внешние данные ==> Импорт*. Откроется диалоговое окно *Импорт*.

2. Выбрать требуемый объект для импорта и щелкнуть на кнопке *Импорт*. На экране появится диалоговое окно *Импорт объектов*.

3. Выполнить действия, следуя указаниям *Мастера импорта*.

---

---

## Лабораторная работа №16

### ОБСЛУЖИВАНИЕ БАЗЫ ДАННЫХ

---

---

**Цель работы:** освоить технологию создания и использования MDE-файла для защиты базы данных, приобрести практические навыки оптимизации производительности базы данных, автоматической оптимизации объектов, документирования базы данных.

#### **Задание 1. Использование MDE-файла для защиты базы данных**

Если база данных создается для нескольких пользователей, то путем сохранения базы данных в виде MDE-файла можно защитить ее формы и отчеты, а также любой код Visual Basic. При использовании файла этого типа имеется возможность просмотра и редактирования данных в формах и отчетах, однако открыть эти объекты в режиме конструктора не получится.

#### **Выполнение задания**

##### **❖ Создание MDE-файла из файла базы данных**

Чтобы создать MDE-файл, выполните следующие действия.

**1.** Выберите команду *Сервис ==> Служебные программы ==> Создать MDE-файл*.

**2.** В диалоговом окне *Сохранение файла MDE* под именем введите имя нового файла. Можно использовать имя исходного файла, поскольку новый файл будет иметь расширение *.mde*.

**3.** Щелкните на кнопке *Сохранить*.

##### **❖ Открытие MDE-файла**

Чтобы открыть базу данных, которая была сохранена в виде *MDE-файла*, выполните следующие действия.

**1.** На панели инструментов щелкните на кнопке *Открыть* или выберите команду *Файл ==> Открыть*.

**2.** В списке *Тип файлов* диалогового окна *Открытие файла базы данных* оставьте выбранным пункт *Файлы данных* или выберите пункт *Файлы MDE (\*.mde)*.

**3.** Выберите *MDE-файл* для открытия.

**4.** Щелкните на кнопке *Открыть*.

Окно базы данных *MDE-файла* выглядит почти так же, как и окно исходного файла базы данных, за исключением того момента, что для объектов *Формы*, *Отчеты*, *Страницы* и *Модули* кнопки *Конструктор* и *Создать* недоступны (окрашены в серый цвет). На панели *Модули* также неактивна кнопка *Запуск*.

## **Задание 2. Оптимизация производительности базы данных**

Если выполнение таких задач, как запуск запроса или сортировка записей занимает слишком много времени, следует попробовать оптимизировать работу базы данных. Для этого можно

- сжать базу данных, чтобы ликвидировать неиспользуемое дисковое пространство внутри базы данных;
- выполнить анализ быстродействия, который позволяет протестировать объекты базы данных, предложить варианты изменения их структуры для повышения быстродействия и даже автоматически внести некоторые изменения\*.

### **Выполнение задания**

#### **❖ Сжатие базы данных**

1. Закройте текущую базу данных и все открытые в данный момент базы данных.
2. Для сжатия текущей базы данных выберите команду *Сервис ==> Служебные программы ==> Сжать и восстановить базу данных*.
3. В диалоговом окне *База данных для сжатия* выберите необходимую базу данных.
4. Щелкните на кнопке *Сжать*.
5. В диалоговом окне *Сжатие базы данных под именем* выберите имя исходной базы данных или введите новое имя в поле списка *Имя файла*.
6. Щелкните на кнопке *Сохранить*. Если вы переписываете исходный файл, в ответ на предложение заменить файл щелкните на кнопке *Да*.

#### **❖ Анализ быстродействия базы данных**

Процедура анализа быстродействия позволяет тестировать работу объектов базы данных. Чтобы инициировать эту процедуру, выполните следующие действия:

1. Откройте базу данных.
2. Выберите команду *Сервис ==> Анализ ==> Быстродействие*. Откроется диалоговое окно *Анализ быстродействия*.
3. Установите флажки для тех объектов базы данных, работу которых необходимо проанализировать. Чтобы выбрать все объекты на текущей вкладке, щелкните на кнопке *Выделить все*. Для выделения всех объектов всех вкладок перейдите на вкладку *Все типы объектов*.
4. Чтобы проанализировать существующие в базе данных отношения объектов, щелкните на вкладке *Текущая база данных* и установите флажок для объекта *Схема данных*. (На этой же вкладке находится пункт *VBA Project*.)

---

\* В Справочной системе Access содержится множество полезных советов по улучшению быстродействия базы данных. Откройте окно *Помощника*, введите слово «анализ» и выберите тему «Оптимизация быстродействия».

Установите флажок для этого объекта, если хотите проверить код входящих в базу данных программ.)

**5.** Щелкните на кнопке ОК.

После завершения анализа быстрого действия откроется диалоговое окно с результатами, показанное на рис. 16.1. Каждый пункт списка *Результаты анализа* входит в одну из категорий: *Совет*, *Предложение*, *Мысль* (в категорию *Исправлено* входят исправленные объекты). Среди предложений непосредственно в списке *Результаты анализа* можно увидеть, что программа советует нам предпринять установить связь между таблицами МЕНЕДЖЕРЫ и другими таблицами базы данных. (Эта связь специально была удалена перед выполнением анализа быстрого действия, чтобы посмотреть, как программа справится с работой.) и таблицей ФИЛИАЛЫ и другими таблицами базы данных.

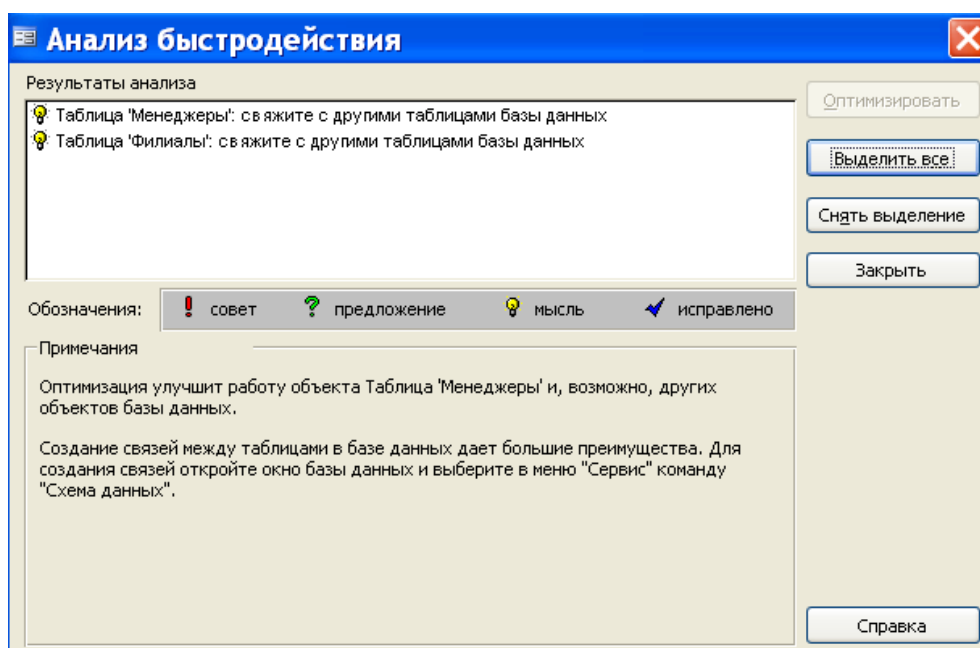


Рис. 16.1 - Средство анализа быстрого действия представляет результаты своей работы в виде списка

### **Задание 3. Автоматическая оптимизация объектов базы данных**

После анализа быстрого действия программа может внести изменения в текущую базу данных по тем пунктам, которые помечены как *Совет* и *Предложение*. Для внесения этих изменений выполните следующие действия.

#### **Выполнение задания**

1. Щелкните на тех пунктах, в которых упоминаются требующие оптимизации объекты. (Чтобы выделить несколько смежных пунктов, удерживайте при выделении клавишу <Shift>. Чтобы выделить несколько несмежных пунктов, удерживайте при выделении клавишу <Ctrl>.)
2. Щелкните на кнопке *Оптимизировать*.
3. По окончании оптимизации щелкните на кнопке *Закреть*.

После выполнения изменений по выделенным пунктам программа помечает их галочкой и относит в категорию *Исправлено*.

*Примечание.* Для пунктов, входящих в категорию *Мысль*, кнопка *Оптимизировать* будет неактивна. Кнопка доступна только для категорий *Совет* и *Предложения*.

#### **Задание 4. Документирование базы данных**

Средства Access позволяют составить *архив* с полной документацией базы данных, в которую входят описания структуры таблиц, свойств их полей, характеристик форм и отчетов, инструкций SQL для запросов и другие сведения.

#### **Выполнение задания**

Чтобы создать архив, выполните следующие действия.

1. Выберите команду *Сервис ==> Анализ ==> Архивариус*. Откроется следующее диалоговое окно (см. рис.16.2):

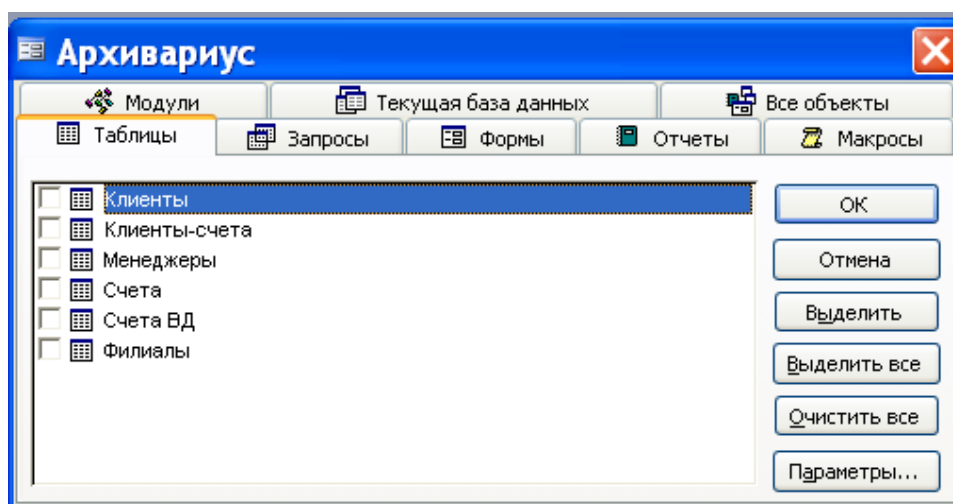


Рис. 16.2 - Окно архивариуса

2. Установите флажки для тех объектов, информация о которых должна войти в документацию. (на вкладке *Все объекты* представлены все объекты базы данных.)

3. Если необходимо задать степень детализации при описании выделенных объектов, щелкните на кнопке *Параметры* и в открывшемся диалоговом окне измените свойства архива.

4. Щелкните на кнопке *ОК*, чтобы вернуться в *окно Архивариус*, сохранив внесенные изменения.

5. Если требуется включить в документацию описание свойств всей базы данных, а также отношения ее объектов, щелкните на вкладке *Текущая база данных* и установите флажок для пунктов *Свойства* и *Связи*.

6. Щелкните на кнопке *ОК*.

В процессе составления архива в строке состояния в нижней части окна Access будут появляться сообщения о текущем обрабатываемом объекте базы



данных. По окончании документирования откроется окно отчета *Описание объектов*, показанное на рис. 16.3. Состав архива будет зависеть от выбранных вами параметров в диалоговом окне *Архивариус*. На последней странице отчета будет представлено описание отношений между объектами базы данных.

<b>Столбцы</b>		
Имя	Тип	Размер
№ счета	Целое	2
Код клиента	Текстовый	5
Код счета	Текстовый	5
Дата открытия счета	Дата/время	8
Сумма на счете	Одинарное с плавающей то	4
Вид валюты	Текстовый	10
Начисления в месяц	Одинарное с плавающей то	4
№ менеджера	Текстовый	5

**Схема данных**

**Клиент\Клиент\счета**

```

    erDiagram
        CLIENTS ||--}| CLIENT_ACCOUNTS : "Код клиента"
    
```

Attributes:                      Обязательность  
 RelationshipType:              один-ко-многим

**Менеджер\Клиент\счета**

```

    erDiagram
        MANAGERS ||--}| CLIENT_ACCOUNTS : "№ менеджера"
    
```

Attributes:                      Обязательность  
 RelationshipType:              один-ко-многим

**Счета\Клиент\счета**

```

    erDiagram
        ACCOUNTS ||--}| CLIENT_ACCOUNTS : "Код счета"
    
```

Attributes:                      Обязательность  
 RelationshipType:              один-ко-многим

Рис. 16.3 – Фрагмент отчета, созданного Архивариусом

## Лабораторная работа № 17

### РАЗРАБОТКА ПРЕЗЕНТАЦИИ БАЗЫ ДАННЫХ СРЕДСТВАМИ MS POWER POINT

**Цель работы:** получить практические навыки разработки презентации информационного продукта (базы данных) средствами MS PowerPoint

#### Порядок выполнения работы

1. Средствами MS PowerPoint подготовить презентацию Базы данных, разработанной в предыдущих работах. Презентация должна включать 10-15 слайдов, содержащих следующую информацию:

- краткое описание информационного продукта, включая назначение, область применения, отличительные особенности (достоинства) разработанной БД, автор-разработчик -2-3 слайда;
- подробное описание Базы данных, включая:
  - структурную схему БД - 1 слайд;
  - описание структуры таблиц (поля и типы данных) – 2-3 слайда;
  - запросы – 3 слайда;
  - формы – 3 слайда;
  - страницы доступа к данным – 1 слайд;
  - отчеты – 1 слайд;
- цена программного продукта (базы данных), сроки поставки, адрес и телефон поставщика, контактная персона – 1 слайд.

Для автоматизации работы при подготовке презентации можно воспользоваться Мастером автосодержания или выбрать один из стандартных шаблонов презентаций MS PowerPoint.

2. Сохранить презентацию на диске (как демонстрацию MS PowerPoint).

3. Продемонстрировать презентацию преподавателю в режиме непрерывного цикла (до нажатия <Esc>).

## ВАРИАНТЫ ЗАДАНИЙ ИНДИВИДУАЛЬНЫХ БАЗ ДАННЫХ

- Разработать структуру базы данных для хранения необходимой информации согласно варианту. При необходимости ввести дополнительные поля.
- Реализовать проект средствами MS Access. Спецификации таблиц получить с использованием команды Сервис/Анализ/Архивариус.
- Сформировать схему данных: установить связи между исходными таблицами, обеспечить условия ссылочной целостности данных.
- Выполнить информационное наполнение базы данных. Значения полей базовых таблиц задать самостоятельно.
- Выполнить индивидуальные задания (по указанию преподавателя) по ведению баз данных.

### **Вариант №1 "Договора подряда"**

База данных "Договора подряда" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество сотрудника.
- 2) Название должности сотрудника.
- 3) Домашний адрес сотрудника.
- 4) Характеристика сотрудника.
- 5) Наименование проекта, в котором сотрудник принимал участие.
- 6) Дата начала/окончания проекта.
- 7) Количество дней, отработанных сотрудником в проекте.
- 8) Должностные оклады.
- 9) Описание проекта.

База данных должна содержать информацию о 10 сотрудниках, 4 должностях, 2 проектах. Предусмотреть, чтобы не менее 5 сотрудников работали в 2 проектах одновременно.

### **Вариант №2 "Телефонные переговоры"**

База данных "Телефонные переговоры" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество абонента.
- 2) Телефонный номер абонента.
- 3) Телефонный код и название города, куда звонил абонент.
- 4) Дата разговора.
- 5) Продолжительность разговора.
- 6). Тариф за 1 минуту разговора с указанным городом.
- 7) Домашний адрес абонента.

База данных должна содержать информацию о 10 абонентах, 5 городах. Предусмотреть, чтобы 5 абонентов сделали не менее 2 телефонных разговоров с различными городами.

### **Вариант №3 "Торговые операции"**

База данных *"Торговые операции"* должна хранить следующую информацию:

- 1) Название фирмы-клиента.
- 2) Фамилия, Имя, Отчество руководителя.
- 3) Юридический адрес фирмы-клиента.
- 4) Контактный телефон.
- 5) Название проданного товара.
- 6) Единица измерения проданного товара.
- 7) Количество проданного товара.
- 8) Дата продажи.
- 9) Цена товара.
- 10) Скидка.
- 11) Описание товара.

База данных должна содержать информацию о 10 фирмах, 5 товарах. Предусмотреть, чтобы 5 фирм сделали не менее 2 покупок различных товаров.

### **Вариант №4 "Преподаватели вуза"**

База данных *"Преподаватели вуза"* должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество преподавателя.
- 2) Телефон преподавателя.
- 3) Место работы преподавателя.
- 4) Название должности преподавателя.
- 5) Предмет, который ведет преподаватель.
- 6) Должностная почасовая ставка преподавателя.
- 7) Количество проведенных часов по данному предмету.
- 8) Домашний адрес преподавателя.
- 9) Характеристика.

База данных должна содержать информацию о 10 преподавателях, 3 должностях, 4 предметах. Предусмотреть, чтобы 5 преподавателей проводили занятия по 2 предметам одновременно.

### **Вариант №5 "Библиотека"**

База данных *"Библиотека"* должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество читателя.
- 2) Домашний адрес читателя.
- 3) Телефон читателя.
- 4) Дата рождения читателя.
- 5) Номер читательского билета.
- 6) Автор книги, которую взял читатель.
- 7) Название книги.
- 8) Дата выдачи книги.
- 9) Дата возврата книги.
- 10) Цена книги.

База данных должна содержать информацию о 7 читателях, 7 книгах. Предусмотреть, чтобы каждый читатель брал не менее 2 книг.

### **Вариант №6 "Банковские вклады"**

База данных "Банковские вклады" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество вкладчика.
- 2) Номер счета в банке.
- 3) Дата посещения.
- 4) Сумма вноса.
- 5) Сумма выдачи.
- 6) Серия и номер паспорта вкладчика.
- 7) Домашний адрес вкладчика.

База данных должна содержать информацию о 10 вкладчиках. Предусмотреть, чтобы каждый вкладчик посещал банк не менее 2 раз.

### **Вариант №7 "Резервирование билетов"**

База данных "Резервирование билетов" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество пассажира.
- 2) Домашний адрес пассажира.
- 3) Телефон пассажира
- 4) Номер поезда.
- 5) Тип поезда (скоростной, скорый, пассажирский).
- 6) Номер вагона.
- 7) Тип вагона (общий, плацкартный, купе, спальный).
- 8) Дата отправления.
- 9) Время отправления/прибытия.
- 10) Пункт назначения.
- 11) Расстояние до пункта назначения.
- 12) Стоимость проезда до пункта назначения.
- 13) Доплата за срочность
- 14) Доплата за тип вагона.

База данных должна содержать информацию о 5 пассажирах, 3 поездах, 4 пунктах назначения. Предусмотреть, чтобы 5 пассажиров пользовались услугами резервирования билетов не менее 2 раз.

### **Вариант №8 "Фотосервис"**

База данных "Фотосервис" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество клиента.
- 2) Домашний адрес клиента.
- 3) Телефон клиента
- 4) Наименование фотоуслуги.
- 5) Описание фотоуслуги (характеристика)
- 6) Количество единиц заказа.
- 7) Цена за единицу.

8) Дата приемки заказа.

9) Дата выдачи заказа.

База данных должна содержать информацию о 10 клиентах, 5 видах услуг. Предусмотреть, чтобы каждый клиент делал заказы не менее, чем на 2 фотоуслуги.

### **Вариант №9 "Коммунальные услуги"**

База данных "*Коммунальные услуги*" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество квартиросъемщика.
- 2) Домашний адрес квартиросъемщика.
- 3) Номер лицевого счета.
- 4) Количество жильцов.
- 5) Площадь квартиры, кв.м
- 6) Вид услуги (название платежа).
- 7) Стоимость услуги на квадратный метр площади.
- 8) Стоимость услуги на 1 жильца.

База данных должна содержать информацию о 10 квартиросъемщиках, 5 видах. услуг. Стоимость одних услуг должна определяться площадью квартиры, других — количеством жильцов. Предусмотреть, чтобы каждый квартиросъемщик пользовался не менее чем 3 коммунальными услугами.

### **Вариант №10 "Прокат товаров"**

База данных "*Прокат товаров*" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество клиента.
- 2) Серия и номер паспорта клиента.
- 3) Домашний адрес клиента.
- 4) Телефон клиента
- 5) Наименование товара.
- 6) Описание товара.
- 7) Стоимость товара.
- 8) Дата выдачи.
- 9) Дата возврата.
- 10) Стоимость проката за сутки.

База данных должна содержать информацию о 10 клиентах, 5 товарах. Предусмотреть, чтобы 5 клиентов пользовались услугами проката не менее 2 раз.

### **Вариант №11 "Успеваемость студентов"**

База данных "*Успеваемость студентов*" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество студента.
- 2) Домашний адрес студента.
- 3) Телефон студента
- 4) Дата рождения.
- 5) Предмет.
- 6) Дата сдачи.

- 7) Оценка,
- 8) Фамилия, имя, отчество преподавателя.
- 9) Размер стипендии.

База данных должна содержать информацию о 10 студентах, 4 предметах, 2 преподавателях. Предусмотреть, чтобы 5 студентов сдавали не менее 2 экзаменов.

### **Вариант №12 "Рейтинг студентов"**

База данных *"Рейтинг студентов"* должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество студента.
- 2) Домашний адрес студента.
- 3) Телефон студента.
- 4) Дата рождения.
- 5) Предмет.
- 6) Месяц, за который определяется рейтинг.
- 7) Текущая рейтинговая оценка.
- 8) Максимальная рейтинговая оценка по предмету
- 9) Фамилия, имя, отчество преподавателя.

База данных должна содержать информацию о 10 студентах, 4 предметах, 2 преподавателях. Предусмотреть, чтобы 5 студентов имели рейтинговую оценку не менее чем по 2 предметам.

### **Вариант № 13 "Доставка пиццы"**

База данных *"Доставка пиццы"* должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество клиента.
- 2) Адрес клиента
- 3) Телефон.
- 4) Дата заказа.
- 5) Название пиццы.
- 6) Количество заказанной пиццы.
- 7) Цена пиццы.
- 8) Описание пиццы.
- 9) Скидка

База данных должна содержать информацию о 10 клиентах, 5 видах пиццы. Предусмотреть, чтобы 5 клиентов сделали не менее 2 заказов.

### **Вариант № 14 "Туристическая фирма"**

База данных *"Туристическая фирма"* должна хранить следующую информацию:

- 1) Фамилия, имя, отчество клиента
- 2) Телефон клиента
- 3) Адрес клиента (город, улица...)
- 4) Дата поездки, длительность поездки
- 5) Название маршрута
- 6) Количество заказанных путевок

- 7) Цена путевки
- 8) Предоставленная скидка
- 9) Описание маршрута

База данных должна содержать информацию о 10 клиентах, 5 маршрутах. Предусмотреть, чтобы 5 клиентов заказали не менее 2 путевок по различным маршрутам.

### **Вариант №15 "Салон видеопроката"**

База данных "*Салон видеопроката*" должна хранить следующую информацию:

- 1) Фамилия, Имя, Отчество клиента.
- 2) Серия и номер паспорта клиента.
- 3) Домашний адрес клиента.
- 4) Телефон клиента
- 5) Наименование видеокассеты.
- 6) Описание видеокассеты.
- 7) Залоговая стоимость видеокассеты.
- 8) Дата выдачи.
- 9) Дата возврата.
- 10) Стоимость проката за сутки.

База данных должна содержать информацию о 7 клиентах, 7 видеокассетах. Предусмотреть, чтобы 5 клиентов пользовались услугами проката не менее 2 раз.



## ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ТЕСТОВОМУ КОНТРОЛЮ ЗНАНИЙ

по дисциплине «Компьютерные информационные технологии» часть 2

1. Основные понятия теории баз данных: "база данных", "система управления базами данных", "запись", "поле".
2. Состав и назначение объектов СУБД MS Access (таблицы, запросы, отчеты, формы, макросы, модули).
3. Управление окнами в СУБД MS Access.
4. Учебная БД "Борей". Назначение, загрузка, схема данных.
5. Основные приемы работы с существующей базой данных: добавление, удаление, модификация записей в таблице.
6. Режимы таблицы и конструктора таблиц.
7. Действия по проектированию базы данных, выполняемые на этапе концептуального проектирования.
8. Форматы представления числовых значений реквизитов документов.
9. Ключевые реквизиты информационных объектов.
10. Определение связи между информационными объектами.
11. Виды связей между информационными объектами.
12. Действия по проектированию базы данных, выполняемые на этапе логического проектирования.
13. Принципы нормализации таблиц базы данных.
14. Требования, предъявляемые к отношениям, находящимся в первой нормальной форме (1НФ).
15. Требования, предъявляемые к отношениям, находящимся в второй нормальной форме (2НФ).
16. Требования, предъявляемые к отношениям, находящимся в третьей нормальной форме (3НФ).
17. Перечислите основные этапы проектирования реляционной БД. Кратко поясните содержание каждого этапа.
18. Какие требования предъявляют к содержанию таблиц реляционной БД?
19. Приведите порядок создания макета таблицы в режиме Конструктора.
20. Понятия "ключевое поле". Какие бывают ключевые поля?
21. Для чего в каждой таблице задается первичный ключ? В чем различие между первичным и внешним ключом?
22. Порядок формирования схемы БД.
23. Нормализация базы данных. Правила нормализации.
24. Типы данных MS Access. Какие типы данных не могут быть использованы при определении первичного ключа?
25. Поля подстановки
26. Свойства полей (краткая характеристика).

27. Для чего применяется индексирование полей?
28. Отношения между таблицами. Установка связей между таблицами при создании схемы данных.
29. Поиск и замена значений в полях таблиц и форм
30. Сортировка записей. Отбор записей по выделенному фрагменту
31. Создание, применение и редактирование обычного фильтра
32. Создание, применение и сохранение расширенного фильтра
33. Можно ли запросы-действия создавать с помощью мастера?
34. С создания какого типа запроса начинается конструирование запроса-действия?
35. Как посмотреть записи, которые составят новую таблицу, создаваемую запросом?
36. Определен ли в структуре таблицы, созданной запросом, ключ?
37. Какой командой осуществляется преобразование запроса на выборку в запрос на обновление?
38. В какой строке запроса на обновление указывается новое значение обновляемого поля?
39. Можно ли для подсчета нового значения обновляемого поля использовать выражение?
40. Можно ли в новую таблицу, создаваемую запросом включать поля нескольких таблиц?
41. Какая инструкция языка SQL соответствует запросу на обновление?
42. Можно ли просмотреть содержимое обновляемых полей до и после обновления, не выходя из режима конструктора запросов?
43. В какой строке запроса на добавление указывается, в какие поля должны попадать добавляемые записи?
44. Какая инструкция языка SQL соответствует запросу на добавление?
45. Вслед за каким словом инструкции INSERT записывается имя таблицы, к которую добавляются записи?
46. С помощью чего формируются добавляемые записи в инструкции INSERT?
47. Могут ли добавляемые записи формироваться на основе нескольких таблиц?
48. Можно ли одним запросом удалить записи из нескольких таблиц?
49. Если в запросе на удаление использованы главная таблица и две подчиненных, находящихся в отношении 1:М, из какой таблицы могут удалиться записи?
50. Какая инструкция языка SQL соответствует запросу на удаление?
51. В предложении FROM инструкции DELETE указана одна таблица, нужно ли использовать в списке полей конструкцию *имя\_таблицы.\**?
52. Какие поля включаются в список полей запроса на удаление?
53. Назначение запросов. Отличие запросов от таблиц БД.
54. На основе, каких объектов БД, может формироваться запрос?
55. Что такое условие отбора? Как задать простое условие для отбора

данных?

**56.** В чем особенности создания запроса на выборку с логическими операциями в условии отбора?

**57.** В чем особенности создания запроса на выборку с именами полей?

**58.** В чем особенности создания запроса на выборку с вычисляемым полем?

**59.** Как задать для поля таблицы пользовательский формат?

**60.** Когда используются и как создаются запросы с параметрами?

**61.** Каковы особенности использования в запросах статистических функций?

**62.** Как просмотреть инструкции SQL для существующего запроса?

**63.** Способы объединения записей в многотабличном запросе.

**64.** Какие существуют операции SQL ядра Access для объединения записей в многотабличном запросе?

**65.** Какие существуют операции SQL ANSI для объединения записей в многотабличном запросе?

**66.** Создание запроса на выборку с логическими операциями в условии отбора в многотабличном запросе.

**67.** Создание запроса на выборку с именами полей в многотабличном запросе.

**68.** Создание запроса с вычисляемым полем в многотабличном запросе.

**69.** Конструирование запроса на основе нескольких взаимосвязанных таблиц. Ввод параметров в запрос.

**70.** Варианты использования оператора подобия при вводе параметров в запрос.

**71.** Групповые операции в многотабличном запросе.

**72.** Ввод параметров в многотабличный запрос с групповыми операциями.

**73.** Этапы создания перекрестной таблицы в режиме мастера запросов.

**74.** Особенности создания перекрестного запроса с помощью мастера.

**75.** Встроенные функции мастера перекрестных запросов для вычисления итоговых значений.

**76.** Создание перекрестной таблицы на основе данных из нескольких таблиц с помощью конструктора запросов.

**77.** Встроенные функции для вычисления итоговых значений при построении перекрестного запроса с помощью конструктора запросов.

**78.** Форматы команд языка SQL для создания перекрестных запросов.

**79.** Создание однотобличной формы в режиме автоформы.

**80.** Конструирование новой формы и включение в нее связанных полей базового источника.

**81.** Ввод и оформление надписей в форме.

**82.** Создание в форме элементов управления.

**83.** Оформление формы (изменение фона, вставка рисунков, объектов).

**84.** Создание в форме вычисляемых полей.

85. Встроенные функции для вычислений итоговых значений в формах.
86. Использование выражений в вычисляемых полях форм.
87. Редактирование однотобличной формы, созданной мастером, в режиме конструктора.
88. Ввод новых записей в таблицу с помощью форм.
89. Создание с помощью мастера двухтабличной формы с включением подчиненной (связанной) формы.
90. Разработка одиночной многотабличной формы при использовании мастера.
91. Приемы редактированием в конструкторе созданной с помощью мастера многотабличной формы.
92. Технологии добавления элементов управления (кнопок) в формы.
93. Что такое макрокоманда?
94. Что такое макрос?
95. Как создать макрос в среде СУБД Access?
96. Какие способы запуска макроса существуют в среде СУБД Access?
97. Какие ошибки могут появиться при выполнении макроса, созданного в среде СУБД Access и как их исправить?
98. Как добавить на панель инструментов кнопку, запускающую макрос?
99. Как в режиме формы создать кнопку для запуска макроса?
100. Какая служебная программа позволяет создать панель управления приложением?
101. Ограничено ли количество страниц в кнопочной форме?
102. Какие команды можно связать с элементами кнопочной формы?
103. Как связать с элементами кнопочной формы открытие запроса?
104. Какая команда позволяет открыть окно, в котором можно указать, что одновременно с запуском Access должна открываться кнопочная форма?
105. Можно ли запретить пользователю работу в окне базы данных, оставив только возможность выполнять задачи, предусмотренные в кнопочной форме?
106. Какой параметр запуска позволяет ограничить набор команд стандартных меню Access?
107. В каком режиме кнопочной формы можно вставить в нее рисунок?
108. Можно ли, не открывая базу данных, открыть в СУБД Access страницу доступа к данным?
109. Можно ли создать страницу доступа к данным, не открывая базу данных в СУБД Access?
110. Где отображается имя базы данных, к которой подключена создаваемая страница?
111. Как определить, к какой базе данных подключена открытая страница?
112. Можно ли изменять данные в базе через страницу доступа к данным, созданную мастером автостраниц?

**113.** Можно ли фильтровать данные в базе данных через страницу доступа к данным, созданную мастером автостраниц?

**114.** Какой раздел следует удалить, чтобы записи группы отображались на странице доступа к данным целиком?

**115.** Возможно ли создание вычисляемых полей на странице доступа к данным?

**116.** Возможно ли добавление новых записей в несколько таблиц базы данных через страницу доступа к данным, созданную конструктором и содержащую несколько уровней группировки?

**117.** В каком разделе добавляются или удаляются *Раздел кнопок перехода, Раздел подписей, Примечание группы* при проектировании страницы доступа к данным?

**118.** Где можно просмотреть все источники записей, которые используются для страницы доступа к данным, и их связи?

**119.** Можно ли через страницу доступа к данным, содержащую поля со списками, изменять данные в базе?

**120.** Может ли гиперссылка на странице доступа к данным открывать документы Word и Excel?

**121.** Какое окно открывается при щелчке на гиперссылке, содержащей адрес электронной почты?

**122.** Основные механизмы взаимодействия приложений Microsoft Office.

**123.** Способы конвертирования данных в форматы Microsoft Office.

**124.** Назначение и особенности команд Публикация в MS Word и Слияние с MS Word (из меню Сервис/Связи с Office).

**125.** Технологии импорта данных в среду СУБД Access.

**126.** Что такое презентация? Способы создания презентации в Microsoft PowerPoint.

**127.** Основные форматы файлов Microsoft PowerPoint и их особенности.

**128.** Назначение и особенности режимов просмотра презентации. Способы переключения между ними.

**129.** Назначение и особенности создания итогового слайда в режиме Структуры.

**130.** Что такое слайд? Виды слайдов. Способы создания (добавления) нового слайда, удаления, перемещения слайда.

**131.** Краткая характеристика атрибутов слайда: макет (разметка) слайда, цветовая схема слайда, фон слайда, шаблон оформления. Способы их изменения.

**132.** Как добавить мультимедийный объект на один слайд презентации? На каждый слайд презентации?

**133.** Как выполняется настройка анимации объектов слайда, порядок анимации, особенности анимации по щелчку и автоматически.

**134.** Как выполнить показ презентации без анимации объектов на слайдах?

**135.** Как настроить смену слайдов для выделенной группы слайдов? для

всех слайдов презентации? Какие параметры при этом задаются?

**136.** Как настроить смену слайдов, чтобы переход к следующему слайду осуществлялся по щелчку мыши и автоматически, по истечении определенного интервала времени?

**137.** Как настроить время показа отдельных слайдов? В каком режиме работы с презентацией можно отследить время показа отдельных слайдов?

**138.** Как добавить на слайд управляющие кнопки и настроить переходы? Как настроить презентацию для управления только кнопками?

**139.** Как скрыть слайд при показе презентации? Как в процессе показа презентации выполнить переход к скрытому слайду?

## ЛИТЕРАТУРА

1. Бекаревич, Ю. Б. Microsoft Access за 21 занятие для студента / Ю. Б. Бекаревич, Н. В. Пушкина. - Санкт-Петербург: БХВ-Петербург, 2005. - 544с
2. Информатика. Базовый курс: учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симоновича. - 2-е изд. - Санкт-Петербург: Питер, 2007. - 640с.
3. Малыхина, М. П. Базы данных: основы, проектирование, использование: учебное пособие для студентов вузов, обучающихся по направлению подготовки "Информатика и вычислительная техника" / М. П. Малыхина. - Санкт-Петербург: БХВ-Петербург, 2006. - 528с.
4. Советов, Б. Я. Базы данных. Теория и практика: учебник для студентов вузов, обучающихся по направлениям "Информатика и вычислительная техника" и "Информационные системы" / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. - 2-е изд., стер. - Москва: Высшая школа, 2007. - 463с.
5. Фрост, Р. Проектирование и разработка баз данных. Визуальный подход = Database Design and Development / Р. Фрост, Д. Дей, Ван Слайк К. - Москва: NT Press, 2007. - 592с.
6. Автоматизированные информационные технологии в экономике / М. И. Семенов, [и др.]; под ред. И. Т. Трубилина. - М. : Финансы и статистика, 2001. – 416 с.
7. Гринберг А.С., Горбачев Н.Н., Бондаренко А.С. Информационные технологии управления: Учеб. пособие для вузов.– М.: ЮНИТИ-ДАНА, 2004.– 479 с.
8. Шарстнев В.Л.Электронный курс лекций «Системы управления базами данных», (<http://test>), 2000 г.
9. Вардомацкая Е.Ю., Шарстнев В.Л. Технологии обработки, хранения и организации баз данных: методические указания для студентов заочного отделения, Витебск, 2006 г. – 60 стр.
10. Вардомацкая Е.Ю., Казаков В.Е., Шарстнев В.Л. и др. Технологии обработки, хранения и организации баз данных: методические указания к выполнению типового расчета для студентов дневного отделения, Витебск, 2006 г. – 65 стр.