

Государственное бюджетное профессиональное образовательное учреждение
«Байконурский электrorадиотехнический техникум имени М.И. Неделина»
(ГБ ПОУ «БЭРТТ»)

Методические рекомендации
по выполнению практических работ

по междисциплинарному курсу
«Технология разработки и защиты баз данных»

для специальности 09.02.03 «Программирование в компьютерных системах»

г. Байконур
2016 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данные методические указания по выполнению лабораторных работ составлены в соответствии с ФГОС по специальности СПО 09.02.03 «Программирование в компьютерных системах» (базовый уровень) по междисциплинарному курсу «Технология разработки и защиты баз данных».

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения междисциплинарного курса должен:

уметь:

- создавать объекты баз данных в современных системах управления базами данных и управлять доступом к этим объектам;
- работать с современными case-средствами проектирования баз данных;
- формировать и настраивать схему базы данных;
- разрабатывать прикладные программы с использованием языка SQL;
- создавать хранимые процедуры и триггеры на базах данных;
- применять стандартные методы для защиты объектов базы данных.

знать:

- основные положения теории баз данных, хранилищ данных, баз знаний;
- основные принципы построения концептуальной, логической и физической моделей данных;
- современные инструментальные средства разработки схемы базы данных;
- методы описания схем баз данных в современных системах управления базами данных (СУБД);
- структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров;
- методы организации целостности данных;
- способы контроля доступа к данным и управления привилегиями;
- основные методы и средства защиты данных в базах данных;
- модели и структуры информационных систем;
- основы разработки приложений баз данных.

иметь практический опыт:

- работы с объектами базы данных в конкретной системе управления базами данных;
- построения концептуальной модели баз данных и разработки структуры баз данных;
- использования средств создания и заполнения базы данных;
- использования стандартных методов защиты объектов базы данных.

Перечень лабораторных работ

№	Тема
1	Проектирование базы данных
2	Построение схемы отношений. Создание и заполнение таблиц базы данных
3	Размещение новых объектов в таблице
4	Создание форм базы данных в режиме конструктора. Создание вычисляемых полей в форме. Создание кнопок на форме
5	Создание подчиненных форм
6	Формирование запросов на модификацию данных. Создание перекрестных запросов
7	Создание отчетов, сводных таблиц и диаграмм
8	Создание макросов
9	Сортировка, поиск и фильтрация данных
10	Создание структуры базы данных с помощью SQL-инструкций
11	Визуальные средства создания запросов (программы SQL-Builder, Database Desktop)
12	Разработка прикладных программ
13	Создание и управление базы данных (программы IBConsole, SQL Explorer)
14	Инструментальные средства для упрощения работы с удаленными базами данных (программы SQL-монитор, DataPump)
15	Управление привилегиями и доступом к данным
16	Резервное копирование и восстановление
17	Создание хранимых процедур и триггеров
18	Организации целостности данных
19	Организация каскадного удаления
20	Организация контроля доступа к данным

Правила выполнения лабораторных работ

Студент перед выполнением лабораторной работы должен строго выполнить весь объем домашней подготовки, указанный в описании соответствующей лабораторной работы. Выполнению каждой работы может предшествовать проверка готовности студента, которая проводится преподавателем.

Отчет о лабораторной работе должен содержать исчерпывающие систематизированные сведения о выполненной работе. Оформление отчета о лабораторной работе должно быть предусмотрено заданием или планом выполнения работы. Отчет составляется исполнителем работы и защищается в индивидуальном порядке в сроки, установленные преподавателем.

Общими требованиями к отчету являются:

- четкость и логическая последовательность изложения материала;
- убедительность аргументации;
- краткость и точность формулировок, исключающих возможность неоднозначного толкования;
- конкретность изложения результатов работы;
- обоснованность выводов.

Лабораторная работа №1
Тема: «Проектирование базы данных»

Цель лабораторной работы:

- освоение технологии построения информационной модели логического и физического уровней в нотации IDEF1X с использованием пакета Microsoft Office Visio 2007;
- развитие познавательного интереса;
- формирование универсальных учебных действий, связанных с поиском информации, необходимой для решения поставленной задачи.

Предварительная подготовка: изучить материал тем «Основные принципы проектирования баз данных» и «Модель сущность-связь» (по конспекту).

Рекомендации по выполнению лабораторных работ: лабораторная работа выполняется группой студентов (2-3 человека) в пакете Microsoft Office Visio 2007. Данная работа может выполняться на основе результатов функционального моделирования предметной области.

Количество часов: 2 часа.

Постановка задачи

Разработать модель «Сущность-связь» по индивидуальному заданию и
предварительно определить структуру таблиц базы данных.

Таблица 1. Варианты заданий

№ вар.	Предметная область
1	База данных «Страховая фирма» Ориентировочные таблицы: «Виды страховок», «Клиенты\объекты», «Страховая деятельность»
2	База данных «Агентство недвижимости» Ориентировочные таблицы: «Объекты недвижимости», «Продажи», «Покупки».
3	База данных ГИБДД (Государственная инспекция безопасности дорожного движения)
4	База данных «Деканат ВУЗа» Ориентировочные таблицы: «Список студентов», «Список предметов», «Сессия»
5	База данных отдела кадров производственного предприятия Ориентировочные таблицы: «Сотрудники», «Штатное расписание», «Отделы», «Цеха»
6	База данных фирмы покупки и продажи автомобилей Ориентировочные таблицы: «Продажи», «Покупки», «Автомобили»
7	База данных «Гостиница» Ориентировочные таблицы: «Номера», «Счета», «Клиенты»
8	База данных «Расчет квартплаты ТСЖ» Ориентировочные таблицы: «Список жильцов», «Оплаты», «Тарифы».
9	База данных «Железнодорожные кассы» Ориентировочные таблицы: «Продажи», «Посадочные места», «Направления»
10	База данных «Авиапассажирские перевозки» Ориентировочные таблицы: «Рейсы», «Самолеты», «Продажи»
11	База данных музея Ориентировочные таблицы: «Экспонаты», «Авторы», «Экспозиции».
12	База данных «Спортивные комплексы района» Ориентировочные таблицы: «Нормативы», «Спортсмены», «Соревнования»
13	База данных «Экзаменационная сессия» Ориентировочные таблицы: «Предметы», «Оценки», «Студенты»
14	База данных «Турагентство» Ориентировочные таблицы: «Туры», «Продажи», «Клиенты»
15	База данных Аптека»

	Ориентировочные таблицы: «Товары», «Поставщики», «Продажи»
16	База данных «Сборка и реализация компьютеров» Ориентировочные таблицы: «Продукция», «Клиенты», «Заказы».
17	База данных Продуктовые магазины района Ориентировочные таблицы: «Продажи», «Отделы», «Товары».
18	База данных больницы (одного отделения) Ориентировочные таблицы: «Больные», «Диагнозы», «Врачи»
19	База данных «Видеотека» Ориентировочные таблицы: «Артисты», «Фильмы», «Продажи»

Краткие теоретические сведения

Методология IDEF1X – язык для семантического моделирования данных, основанный на концепции «сущность-связь».

Различают два уровня информационной модели: логический и физический.

Логическая модель позволяет понять суть проектируемой системы, отражая логические взаимосвязи между сущностями.

Физическая модель отражает физические свойства проектируемой базы данных (типы данных, размер полей, индексы). Параметры физической информационной модели зависят от выбранной системы управления базами данных (СУБД).

Основные элементы информационной модели логического уровня:

Сущность – это множество реальных или абстрактных объектов (людей, предметов, документов и т.п.), обладающих общими атрибутами или характеристиками.

Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована. Именованная сущность осуществляется с помощью существительного в единственном числе. При этом имя сущности должно отражать тип или класс объекта, а не его конкретный экземпляр (например, Студент, а не Петров) (рисунок 1).

Студент



Рисунок 1. Графическое представление сущности «Студент» в MS Office Visio

Любая сущность характеризуется набором атрибутов (свойств). Атрибут сущности – характеристика сущности, то есть свойство реального объекта.

Например, на рисунке 1 атрибутами сущности «Студент» являются: «ID студента», «Фамилия», «Имя», «Отчество», «Дата поступления» и «Номер билета».

Первичный ключ (Primary Key, PK). Каждая сущность должна обладать атрибутом или комбинацией атрибутов, чьи значения однозначно определяют каждый экземпляр сущности. Эти атрибуты образуют первичный ключ сущности.

Внешний ключ (Foreign Key, FK). Если между двумя сущностями имеется специфическое отношение связи или категоризации, то атрибуты, входящие в первичный ключ родительской или общей сущности, наследуются в качестве атрибутов сущностью потомком или категориальной сущностью соответственно. Эти атрибуты и называются внешними ключами.

Пример выполнения работы

Список потенциальных сущностей для рассматриваемого примера:

- 1) задание;
- 2) пояснительная записка;
- 3) курсовая работа;
- 4) положение о курсовом проектировании;
- 5) студент;
- 6) преподаватель;
- 7) график;
- 8) методические указания.

Упражнение 1. Создание логической модели «сущность-связь»


1. Запустите MS Office Visio 2007.

2. На закладке выбора шаблона выберите категорию Программное обеспечение и базы данных и в ней элемент Схема модели базы данных. Нажмите кнопку Создать в правой части экрана.

3. Установите необходимые параметры страницы (масштаб, ориентация страницы).

4. MS Office Visio 2007 поддерживает различные нотации моделей баз данных. Для того чтобы задать нотацию IDEF1X, необходимо выбрать пункты меню База данных → Параметры → Документ. В открывшемся окне на вкладке Общие установить переключатель в меню Набор символов на IDEF1X. Меню Имена, видимые на схеме позволяет указать, какие имена атрибутов сущности будут отображены на диаграмме (концептуальные, физические или оба варианта одновременно). В данном случае для логического представления информационной модели необходимо выбрать пункт Концептуальные имена (рис. 2).

В закладке Отношение окна Параметры документа базы данных в меню Показывать нужно отметить галочкой пункт Мощность, меню Отображение вида выбрать пункт Показывать вербальную фразу, снять галочку в пункте Обратный текст (рис. 3). Данные настройки позволят отобразить имя и мощность связи в модели.

5. Для того чтобы создать сущность, необходимо перетащить элемент  на рабочее поле. Переход в режим редактирования сущности осуществляется двойным щелчком по сущности или по нажатию правой кнопки мыши и выбора пункта меню Свойства базы данных.

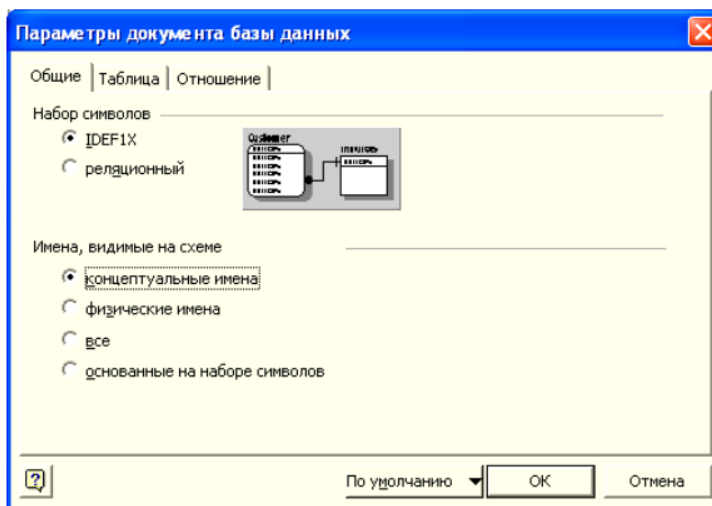


Рисунок 2 – Настройка параметров модели

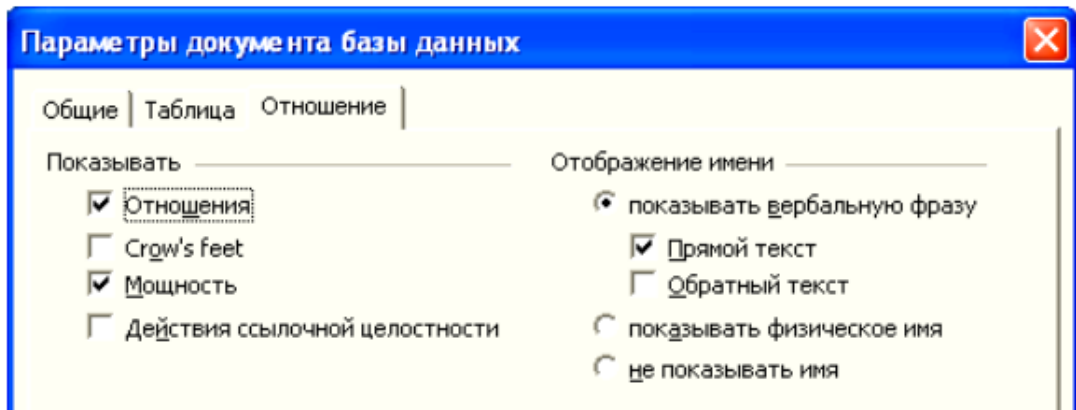


Рисунок 3 – Настройка вида отношений информационной модели

Чтобы задать имя сущности, в окне Свойства базы данных нужно выбрать категорию Определение, снять галочку в пункте Синхронизация имен при вводе (в противном случае, физическое и логическое имя сущности будут совпадать, что по практическим соображениям не всегда удобно) и задать концептуальное имя сущности. Руководствуясь данным алгоритмом, создадим 8 сущностей (рис. 4).



Рисунок 4 – Сущности информационной модели логического уровня

6. Далее необходимо установить связи между сущностями. Сначала составим описание предметной области на естественном языке.

Любой студент должен выполнить одну или несколько курсовых работ.

Каждая курсовая работа должна выполняться одним студентом (в идеале).

Каждая курсовая работа выполняется в соответствии с методическими указаниями и положением о курсовом проектировании.

Курсовая работа сдается по графику.

Курсовая работа оформляется в виде пояснительной записки.

Преподаватель проводит консультации, проверяет и ставит оценку за курсовую работу.

Таким образом, сформулируем имена связей:
 СТУДЕНТ выполняет КУРСОВУЮ РАБОТУ.
 ПРЕПОДАВАТЕЛЬ проверяет КУРСОВУЮ РАБОТУ.
 КУРСОВАЯ РАБОТА выполняется в соответствии с ЗАДАНИЕМ.
 КУРСОВАЯ РАБОТА оформляется в виде ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.
 МЕТОДИЧЕСКИЕ УКАЗАНИЯ определяют требования к КУРСОВОЙ РАБОТЕ.
 КУРСОВАЯ РАБОТА организуется согласно ПОЛОЖЕНИЮ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ.

КУРСОВАЯ РАБОТА сдается по ГРАФИКУ.

Во всех случаях сущность Курсовая работа является дочерней, за исключением связи с сущностью Пояснительная записка. В дальнейшем можно будет подкорректировать связи между сущностями.

Чтобы установить связи между сущностями, необходимо перетащить на рабочую область



элемент, поднести один конец стрелки к родительской сущности, другой – к дочерней.

Примечание. При правильном связывании каждая сущность будет подсвечена красным цветом.

В MS Office Visio 2007 по умолчанию используется не идентифицирующее отношение. Чтобы изменить тип связи, необходимо двойным щелчком по связи открыть окно Свойства базы данных и в категории Прочее указать тип отношения (идентифицирующее, не идентифицирующее). В этой же категории указывается мощность связи (см. рис. 5).



Рисунок 5 – Определение типа связи и мощности

Примечание. Кроме того, при не идентифицирующем отношении нужно указать, является ли наличие родительской сущности обязательным (т.е. может ли существовать экземпляр дочерней сущности, если не существует экземпляра родительской). Если наличие родительского объекта является необязательным, графически это отобразится в виде не закрашенного ромба со стороны родительской сущности.

Следующий шаг – в категории Имя в поле Вербальная фраза нужно указать имя отношения (рис. 6). Также можно указать имя связи в поле Обратная фраза для спецификации отношения потомок родитель (в нашем случае обратная фраза отображаться не будет).

Примечание. Все изменения при закрытии окна свойств сохраняются автоматически.

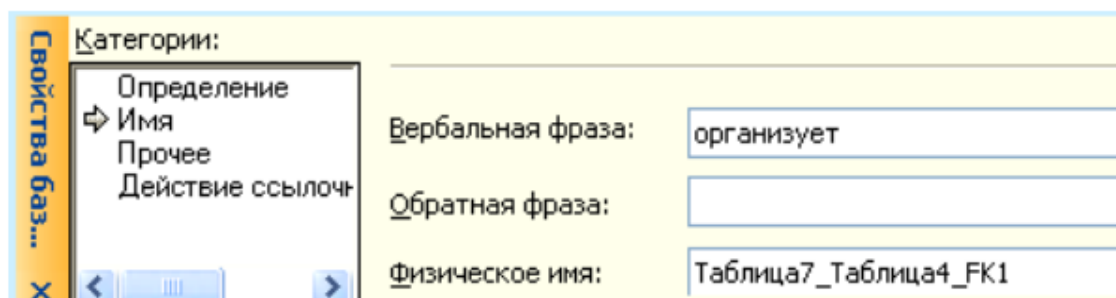


Рисунок 6 – Определение имени отношения

После определения имен, типов связей и задания мощностей получим информационную модель, представленную на рис. 7.

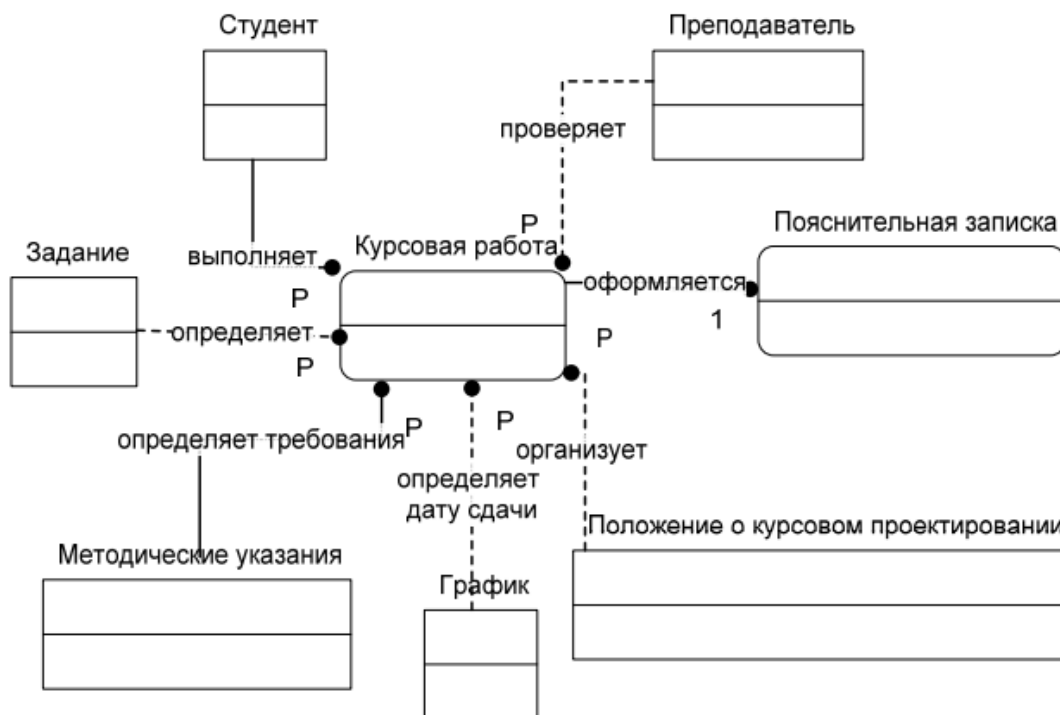


Рисунок 7 – Информационная модель уровня «сущность-связь»

Упражнение 2. Разработка логической модели данных, основанной на ключах

1. Необходимо определить ключевые атрибуты для каждой сущности, обращая внимание на то, что дочерние сущности наследуют ключевые атрибуты от родительских (рис. 9).

Для этого двойным щелчком мыши по сущности откроем окно редактирования ее свойств, перейдем в категорию Столбцы, по нажатию кнопки Добавить введем имя поля (например, для сущности Задание ключевым атрибутом будет являться Вариант задания). Чтобы сделать атрибут ключевым, необходимо отметить галочкой пункт РК (рис. 8). Данное поле становится обязательным автоматически.

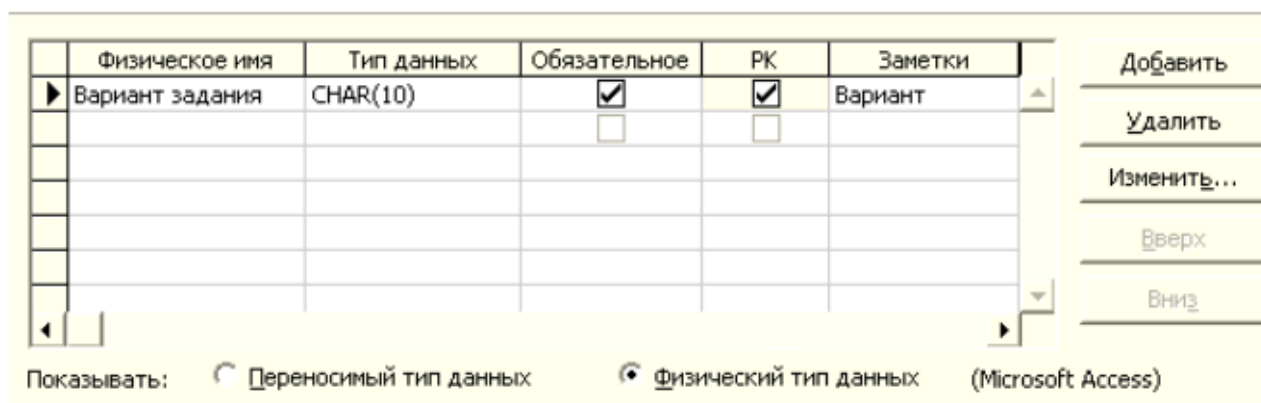


Рисунок 8 – Определение ключевого атрибута

Аналогичным образом зададим ключевые атрибуты для всех сущностей информационной модели. Результат представлен на рис.9.

Как видно из рис. 9 по сравнению с информационной моделью уровня «сущность-связь», был изменен тип связи между сущностями

Методические указания и Курсовая работа, поскольку ключевые атрибуты сущности Методические указания для сущности Курсовая работа будут являться избыточными (зная номер зачетной книжки, можно узнать специальность и курс, на котором учится студент).

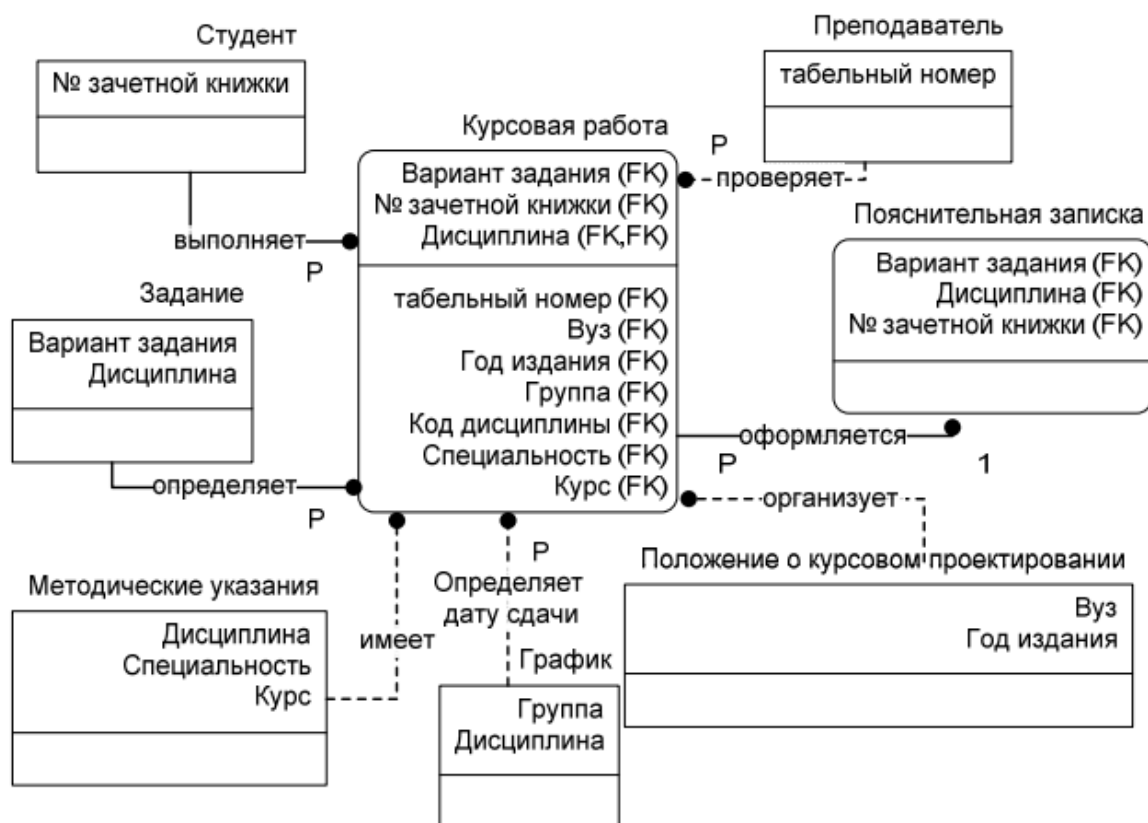


Рисунок 9 – Информационная модель с ключевыми атрибутами

2. Кроме того, отметим, что три сущности (Задание, График, Методические указания) содержат одинаковые атрибуты Дисциплина.

Это является некорректным. Чтобы устранить данную ошибку, выделим одноименную сущность и свяжем ее идентифицирующими связями с вышеуказанными сущностями (рис. 10).

Упражнение 3. Создание полной атрибутивной модели

Для того чтобы получить полную атрибутивную модель, необходимо дополнить сущности не ключевыми атрибутами. Дополненная модель представлена на рис. 11.

Примечание. Если атрибут не является обязательным, нужно убедиться, что в окне Свойства базы данных в категории Столбцы в пункте Обязательное не стоит галочка. Не обязательные к заполнению атрибуты справа от имени имеют пометку (O).



Рисунок 10 – Скорректированная информационная модель, основанная на ключах

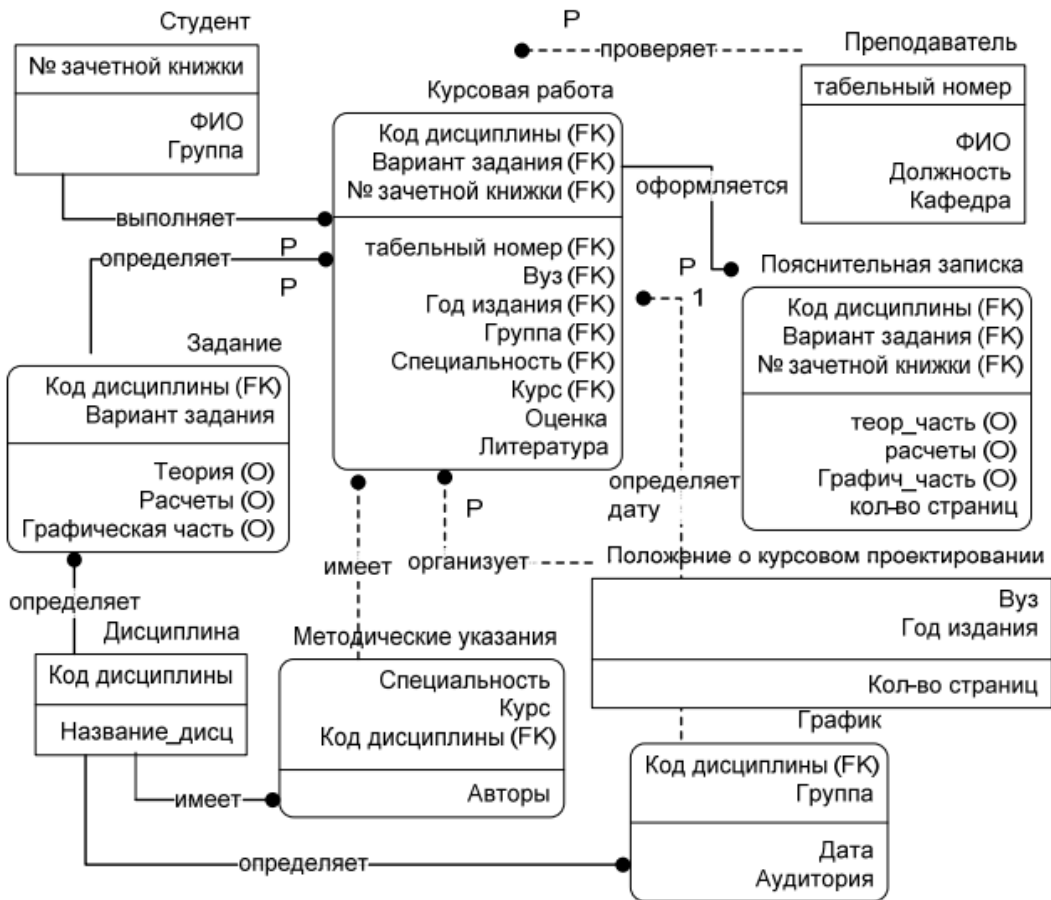


Рисунок 11 – Полная атрибутивная модель

Отчет должен содержать:

1. Название, цель и задание лабораторной работы.
2. Диаграмма «сущность-связь».
3. Распечатка полной атрибутивной модели.
4. Ответы на контрольные вопросы.

Контрольные вопросы:

1. Дайте определение понятиям «Сущность», «Атрибут сущности», «Внешний ключ», «Первичный ключ».
2. Виды связей между сущностями.
3. Какие функции выполняет администратор БД и пользователь БД?
4. Что такое схема данных?
5. Правила построения ER-диаграмм.

Лабораторная работа № 2

Тема: «Построение схемы отношений. Создание и заполнение таблиц базы данных»

Цель лабораторной работы:

- выработка умений и навыков создания и редактирования таблиц, определения и установления связей между ними на схеме данных.
- развитие познавательного интереса;
- изучение новых алгоритмов решения задач;
- формирование универсальных учебных действий, связанных с поиском информации, необходимой для решения поставленной задачи.

Предварительная подготовка: изучить материал тем «Основные принципы проектирования баз данных», «Модель сущность-связь» и «Построение ER – диаграммы» (по конспекту).

Рекомендации по выполнению лабораторных работ: лабораторная работа выполняется группой студентов (2-3человека) средствами СУБД MS Access. Данная работа может выполняться на основе результатов функционального моделирования предметной области.

Количество часов: 2 часа.

Постановка задачи

Создать базу данных по варианту первой лабораторной работы. Данные таблиц должны быть нормализованы. Построить схему отношений. Заполнить таблицы. Создать и применить таблицы-подстановки.

Пример выполнения работы

Для каждого объекта Microsoft Access существуют следующие режимы работы:

- режим таблицы;
- конструктор;
- мастер.

Для перехода с одного режима работы на другой в Microsoft Access есть панель <Режимы>. Режим таблицы При выборе режима таблицы, программа создает абстрактную, но готовую к использованию таблицу. Режим таблицы Совершенство ее, пользователь определяет необходимые свойства и заполняет ее информацией. Например, поля таблицы имеют имена, данные по умолчанию (<Поле1>, <Поле2> и т. д.), но пользователь может присвоить полям новые имена, которые будут нести смысловую нагрузку. Тип полей будет определяться программой автоматически, в зависимости от внесенной в них информации.

Отображение таблицы в окне базы данных Access позволяет менять структуру таблицы и в дальнейшем, на любом этапе работы с базой данных. Для этого предусмотрен режим конструктора. Режим конструктора Этот режим как нельзя лучше подходит для того, чтобы понять, что такое таблица Access. При построении таблицы в режиме конструктора программа никаких полей вначале не создает. Теперь вы сами должны определить структуру таблицы, что называется, «с нуля». В верхней части окна конструктора расположена таблица (бланк), которая должна содержать перечень всех полей, их типов и заголовков. Для создания таблицы необходимо, как минимум, заполнить графы «Имя поля» и «Тип данных».

В Access можно выбрать один из следующих типов данных:

Текстовый. Наиболее чисто используемый в Access тип данных. Этот тип данных подходит для хранения адресов, для полей с кратким описанием, для числовых данных, не требующих расчетов, таких, как телефонные номера и почтовые индексы. Длина – 255 символов.

Поле Мемо. Предназначен для полей, длина которых превосходит 255 символов. Пример: длинное поле описания. Поле Мемо может хранить до 65 535 символов, что приблизительно равно 32 страницам текста.

Числовой. Данные, используемые для математических вычислений, за исключением финансовых расчетов (для них следует использовать тип «Денежный»).

Дата/время. Значения дат и времени. Сохраняет 8 байтов. Можно вводить даты с 1 января 100 года по 31 декабря 9999 года. Access предлагает несколько

Счетчик. Этот тип данных является еще одним специальным числовым полем. Если вы определите, что данное поле будет счетчиком, то Access автоматически присвоит уникальный порядковый номер каждой записи. Первой записи будет присвоено значение 1, девятой — 9 и т. д.

Логический. Требуется ответа «Да» или «Нет», и только. Например: «Просрочен ли платеж этого клиента?»

Поле объекта OLE. Этот тип данных вы позволяет ввести в поле практически все, что пожелаете: рисунок, график, нотную запись, таблицу, подготовленную с помощью одного из приложений Excel или Windows.

Гиперссылка. Текст или комбинация текста или чисел, хранимые как текст и используемые в качестве адреса гиперссылки. Длина каждой из трех частей гиперссылки Текст или комбинация текста или чисел, хранимые как текст и используемые в качестве адреса гиперссылки. Длина каждой из трех частей гиперссылки не должна превышать 2048-ми знаков.

Мастер подстановок. При выборе этого параметра запускается мастер подстановок, создающий поле подстановок. После завершения работы мастера устанавливается тип данных, основанный на значениях, выбранных в мастере.

Установка свойств поля

Размер поля показывает Access, сколько символов (букв, цифр, знаков препинания и им подобных) можно ввести в поле. Текстовое поле может содержать до 255-ти символов. По умолчанию Access устанавливает его длину равную 50-ти (для других типов полей также необходим этот атрибут, однако вам не нужно беспокоиться, Access сделает это за вас). Помните, что размер поля - это не то же самое, что размер столбца, соответствующего данному полю в таблице.

Формат поля позволяет контролировать процесс ввода и вывода данных. Например, можно использовать его для преобразования букв из строчных в прописные. Также можно использовать его для проверки количества цифр после десятичной точки в действительном числе; для указания, каким способом выводить поля «дата» и «да / нет»; для ввода в поля текстов.

Маска ввода позволяет определить для ввода только ограниченную часть поля, и если все значения поля имеют одинаковый формат, то ввести в него форматирование символов.

Условие на значение указывает Access, какие данные допустимы для данного поля, а какие нет. Например, если вы создаете базу данных для книжного магазина, в котором нет книг, дешевле \$10, то можно указать значение \$10 в качестве минимального допустимого в поле «Цена» в таблице заказов. Сообщение об ошибке – это сообщение, которое появляется на экране, если пользователь пытается ввести значение, запрещенное правилом «Условие на значение».

Обязательное поле. Если установить значение этого атрибута «да» для данного поля, то Access Обязательное поле. Если установить значение этого атрибута «да» для данного поля, то Access не разрешит пользователю сохранять запись, пока не будут введены данные в это поле. Допустимость пустых значений применима к текстовым полям. Если установить его значение равным «да», то можно ввести две кавычки для указания того, что поле не используется в текущей записи. Например, пустое поле «Номер факса» может означать, что вы не знаете этого номера данного клиента. Но если введены две кавычки, то это показывает, что данный клиент не имеет факса вообще.

Индексирование поля служит для быстрого поиска записей по содержимому данного поля:

– неиндексированное поле. Если выбран этот режим, Access не создаст индекс для данного поля. Все будет работать нормально, но поиск информации будет несколько за

медлен, т. е. он будет похож на попытку найти описание интересующей темы в книге, которая не имеет предметного указателя;

– индексированное поле (совпадения допускаются). Эта опция указывает Access создать индекс, но также допускает, что может быть более одной записи с одним и тем же значением данного поля. Эту опцию следует использовать, например, в случае индексирования поля «Фамилия», т. к. существует много людей с одинаковой фамилией;

– индексированное поле (совпадения не допускаются). Эта опция указывает Access создать индекс, но не допускает записей с одинаковым значением данного поля. Такая опция подходит в случае индексирования по порядковому номеру клиента, т. к. значение этого поля для каждого клиента уникально.

Создание ключей

Одно из полей таблицы должно быть определено как ключевое. Для этого установите курсор на строку бланка и нажмите кнопку <Ключевое поле> на панели инструментов (или выберите команду <Ключевое поле> из контекстного меню, связанного с данной строкой). Первичный ключ может включать несколько полей. Такая ситуация возникает, когда только комбинация значений не скольких полей может однозначно идентифицировать запись в таблице. Для того чтобы создать такой составной ключ, надо выделить несколько строк в бланке, а затем использовать инструмент <Ключевое поле>. О том, что поле является ключевым, свидетельствует изображение ключика в области выделения строки, содержащей данные о поле. Не может быть более одного ключа, поэтому, если вы, работая с какой-нибудь таблицей в режиме конструктора, увидите, что символом ключика помечены две или более строки то сделайте вывод: эта таблица имеет составной ключ.

Создание таблиц

Для создания таблицы базы данных: в окне базы данных выберите вкладку Создание, а затем щелкните по кнопке <Конструктор таблиц>. Создание таблиц в режиме Конструктора. Для определения полей таблицы: введите в строку столбца Имя поля имя первого поля - ID врача. В строке столбца «Тип данных» щелкните по кнопке списка и выберите тип данных Числовой. Поля вкладки Общие оставьте такими, как предлагает Access. Для определения всех остальных полей таблицы базы данных Врачи выполните действия, аналогичные указанным выше в соответствии с Таблица 1. Описание свойств полей таблицы Врачи

Таблица 1. Врачи

Имя поля	Тип данных	Размер поля
ID врача	Числовой	
ФИО	Текстовый	20
Должность	Текстовый	20
№ кабинета	Числовой	Длинное целое (по умолчанию)
Время приема	Текстовый	12
ВидОказываемыхУслуг	Текстовый	50(Максимальное значение)

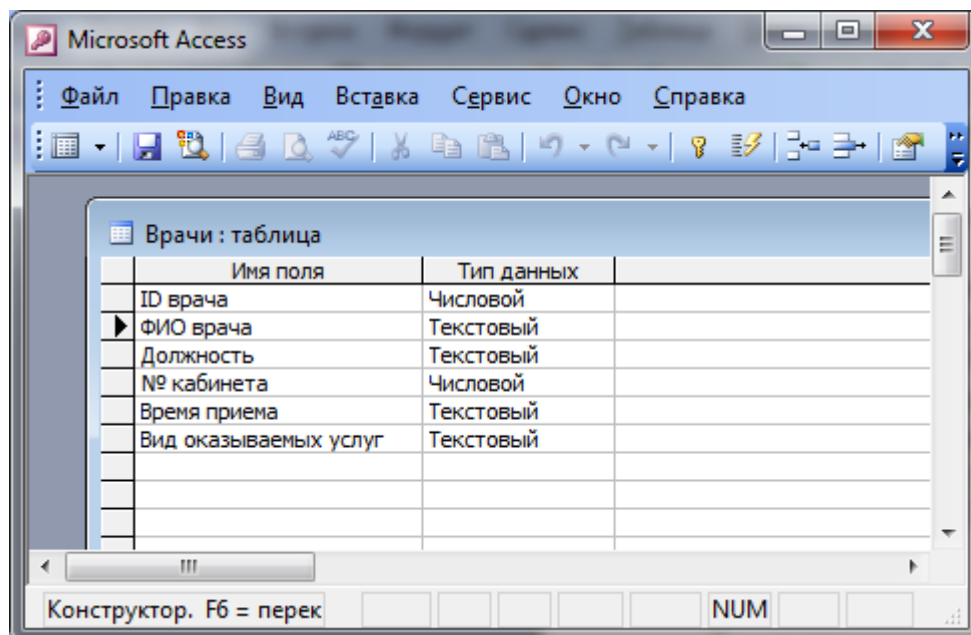


Рисунок 1 - Таблица «Врачи» в режиме конструктора

Использование для поля дополнительных параметров с помощью вкладки «Общие»

Задайте условие для поля «№ кабинета» Пусть по условию в поликлинике 2 этажа и значение номера кабинета находится в пределах от 100 до 130 и от 200 до 230. В строке «Условие на значение» перейдите в режим Построителя выражений (рис. 2) и введите условие: ≥ 100 And ≤ 130 Or ≥ 200 And ≤ 230 (см. рис. 5). В строке «Сообщение об ошибке» введите: «Такого номера кабинета нет».

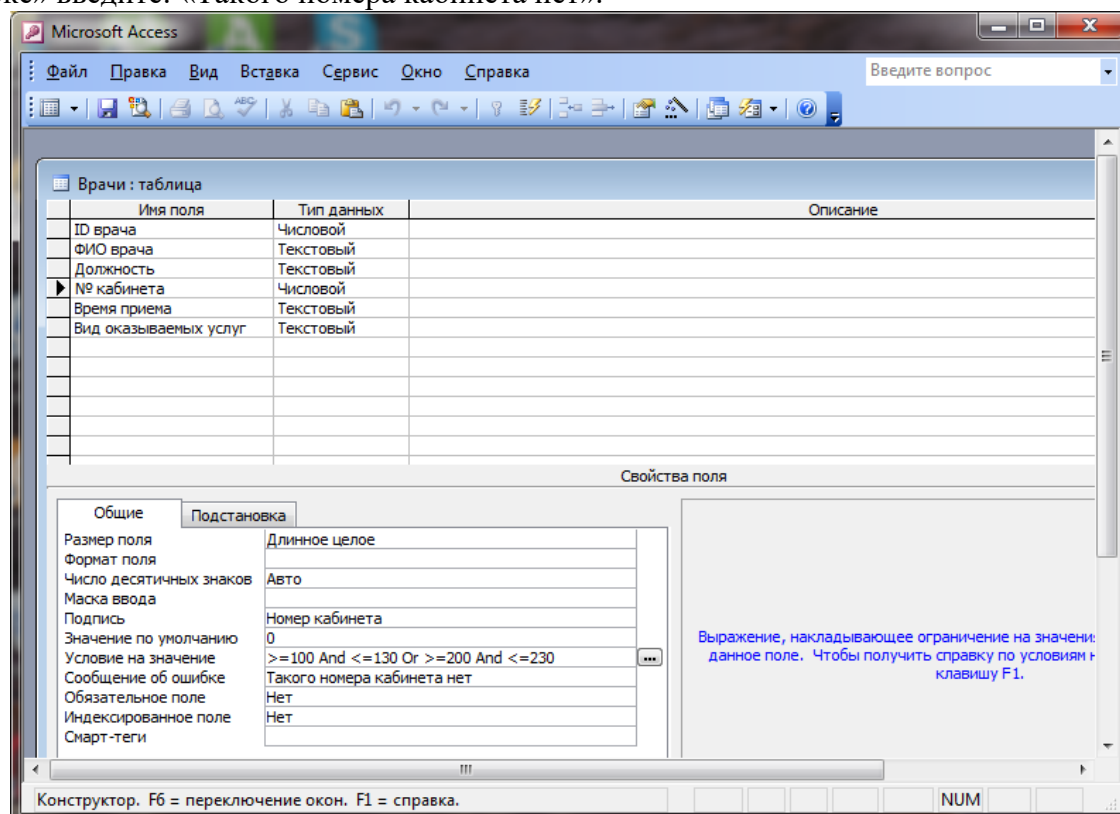


Рисунок 2 - Вкладка «Общие» с заданными условиями для поля «№ кабинета»

Создание ключевых полей

Первичный ключ - это поле (или ряд полей), которое однозначно идентифицирует каждую запись в таблице. Access не допускает повторных значений в поле первичного

ключа. Сделайте поле ID врача ключевым. Для этого откройте таблицу Врачи в режиме Конструктора и, поместив курсор на имя этого поля, щелкните по кнопке - Ключевое поле (рис. 3):

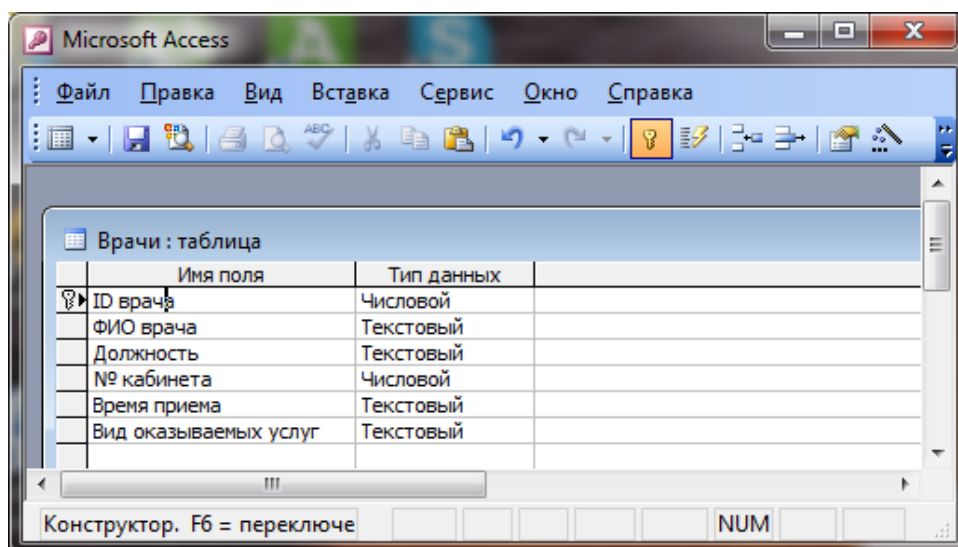


Рисунок 3 - Ключевое поле

Создание таблиц-подстановок

Для упрощения ввода данных в таблицу «Врачи» создайте таблицы-подстановки для полей: «ВремяПриема» и «ВидОказываемыхУслуг». Для поля «Должность» данные будут подставляться из таблицы «Штатное расписание». Подстановка упрощает режим ввода. При этом нет необходимости набирать длинные названия несколько раз. Достаточно будет выбрать необходимую запись из выпадающего списка.

Таблица «Смена-подстановка» (используется для поля «ВремяПриема»). Поля таблицы текстового типа.

Таблица 2. Смена-подстановка

№ смены	Время приема
1	08-12
2	13-19

Описание свойств полей таблицы «Смена_подстановка» В результате щелчка по кнопке <ОК> MS Access предложит вам задать ключевое поле, наличие ключевого поля в данной таблице нет.

Таблицу «НаименованиеУслуги_подстановка» (используется для поля «Вид оказываемых услуг») создайте в Excel, чтобы импортировать её в нашу базу данных.

Таблица 3. Таблица с данными «НаименованиеУслуги_подстановка»

Наименование услуги	направление
Терапевтическая	
Хирургическая	
Ортопедическая	

Импорт как самих таблиц, так и полей таблиц можно производить из разных источников и файлов разных форматов. Определите формат ячеек: для поля НаименованиеУслуги – текстовый. Сохраните таблицу в той же папке, где находится ваша

БД. В базе данных на панели инструментов выберите Внешние данные Excel. В открывшемся диалоговом окне выберите созданную таблицу.

В следующем окне «Импорт электронной таблицы» выберите Лист 1, нажмите <Далее>. установите флажок «Первая строка содержит заголовки столбцов», далее: «Данные необходимо сохранить в новой таблице», далее подтвердите описание и название полей, далее выберите «Не создавать ключ».

Таблица «Штатное расписание» (данные поля «НаименованиеДолжности» будут использоваться для подстановки в таблицу «Врачи»

Таблица 4. Таблица с данными «Штатное расписание»

Должность	Оклад	Надбавка
ортопед	9000	1000
стоматолог	11000	700
хирург	14000	1200

Для поля «Оклад» во вкладке Общие в строке «Формат поля» выбрать «Денежный». Для поля «Должность» задать значение Ключевое поле.

Создание подстановочного поля

Для создания подстановочного поля используется три способа:
режим Конструктора. Этот способ используется в том случае, если поле для подстановки стоит первым в подстановочной таблице,
использование Мастера подстановок,
или использование языка запросов SQL (с помощью построителя запросов).

1) Откройте таблицу «Врачи» в режиме Конструктора. Для поля «Должность» во вкладке Подстановка выберите в строке Тип элемента управления: «Поле со списком»; в строке Тип источника строк: «Таблица или запрос» (задается по умолчанию); в строке Источник строк из выпадающего списка выбрать таблицу «Штатное расписание» - то есть таблицу, из которой данные будут подставляться в таблицу «Врачи».

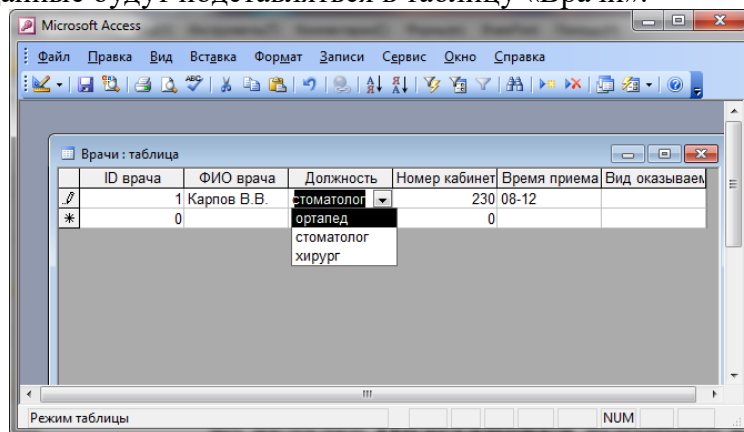


Рисунок 4 - Подстановка данных из таблицы «Штатное расписание» в таблицу «Врачи»

Для создания подстановки для поля «ВремяПриема» в столбце Тип данных выберите из выпадающего списка «Мастер подстановки». В следующем окне нажмите кнопку <Далее>, в открывшемся диалоговом окне выберите «Таблица: Смена_подстановка». Нажмите <Далее>. В следующем окне «Создание подстановки» из списка «Доступные поля» выберите при помощи кнопки поле «ВремяПриема» (данные из этого поля будут подставляться в нашу таблицу).

Построение схемы отношений

Для создания схемы данных выполните действия: вкладка «Работа базами данных» «Схема данных». Появится окно Добавление таблицы. Выберите вкладку Таблицы, выделите таблицы в данном примере это таблицы БД «Библиотека»: «Книги», «Издательство», «Читатель», «Выдача» разместите их с помощью кнопки <Добавить>. В окне Схема данных появится условный вид этих таблиц (см.рис. 5).

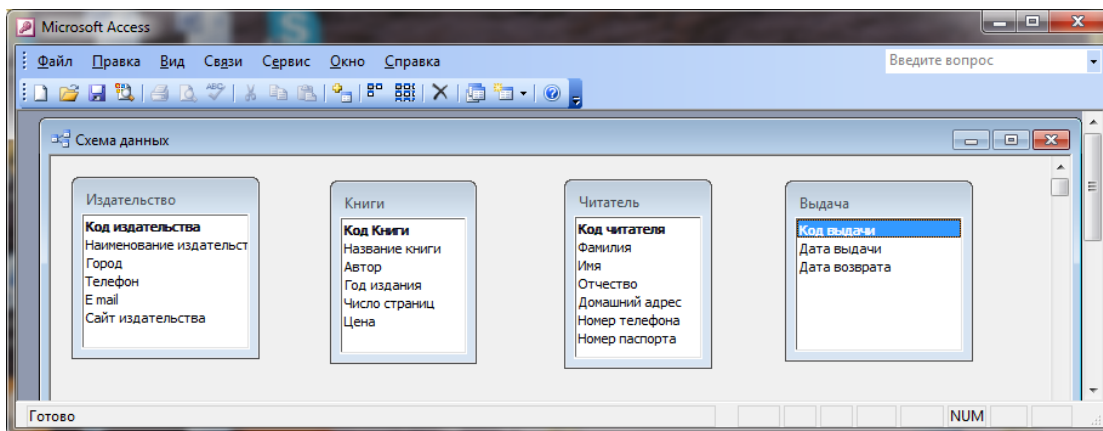


Рисунок 5 - Добавление таблиц в окно Схема данных

Примечание: Жирным текстом в схеме будут выделены поля, имеющие ключевое значение.

Установите связь между таблицами «Книги» и «Издательство». Одно издательство может выпустить множество книг, следовательно в таблицу «Книги» необходимо добавить в режиме конструктора поле «Код издательства» тип поля числовой. Перетащите поле «Код издательства» из таблицы «Издательство» на поле «Код издательства» в таблице «Книги». Появится диалоговое окно Связи рисунок 6.

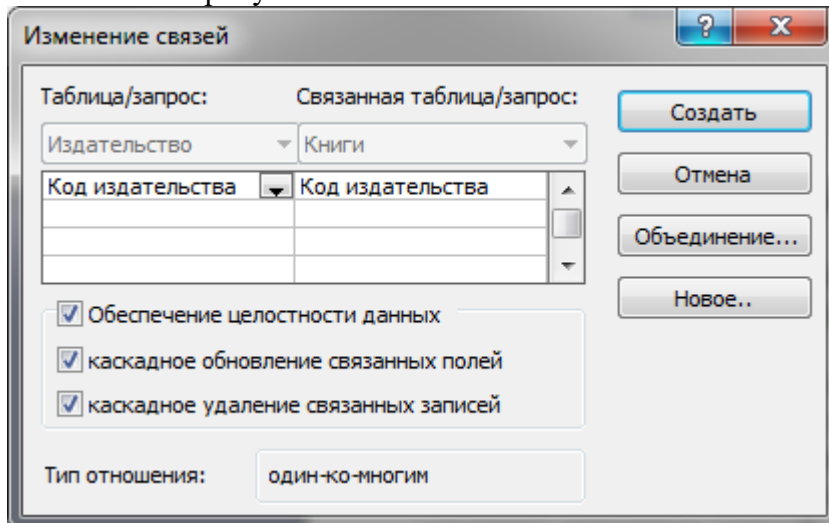


Рисунок 6 - Создание связи между таблицами

Для автоматической поддержки целостности базы данных установите флажок: Обеспечение целостности данных. Кроме этого значка в этом окне представлены другие: Каскадное обновление связанных полей. При включении данного режима изменения, сделанные в связанном поле первой таблицы, автоматически вносятся в поля связанной таблицы, содержащей те же данные. Каскадное удаление связанных записей. При включении данного режима удаление записей в первой таблице приводит к удалению соответствующих записей связанной таблицы.

Примечание: Это возможно сделать только в том случае, если типы и размер полей заданы одинаково. Щелкните по кнопке <Создать>. Появится связь один-ко-многим: Рис. 7.

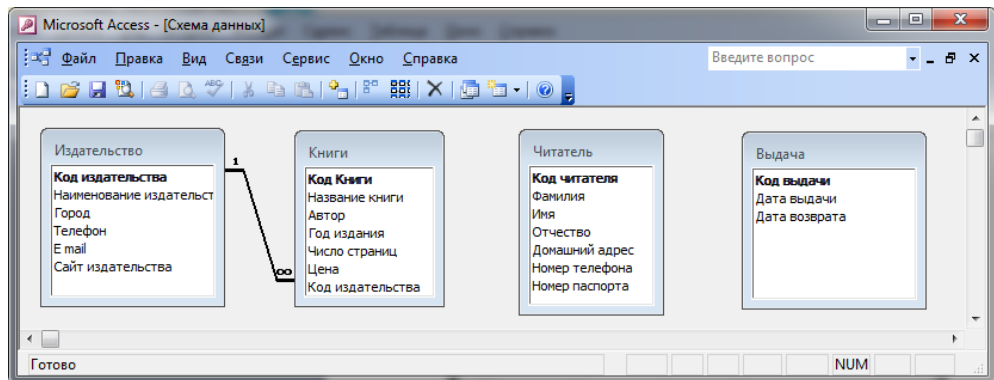


Рисунок 7 - Создание схемы отношений.

Необходимо связать таблицы «Читатель» и «Выдача», проанализировав понятно, что одному читателю могут выдавать несколько книг, следовательно в таблицу «Выдача» необходимо добавить поле «Код читателя». По аналогии связываем таблицы «Книги» и «Выдача» (одна книга может выдаваться множество раз).

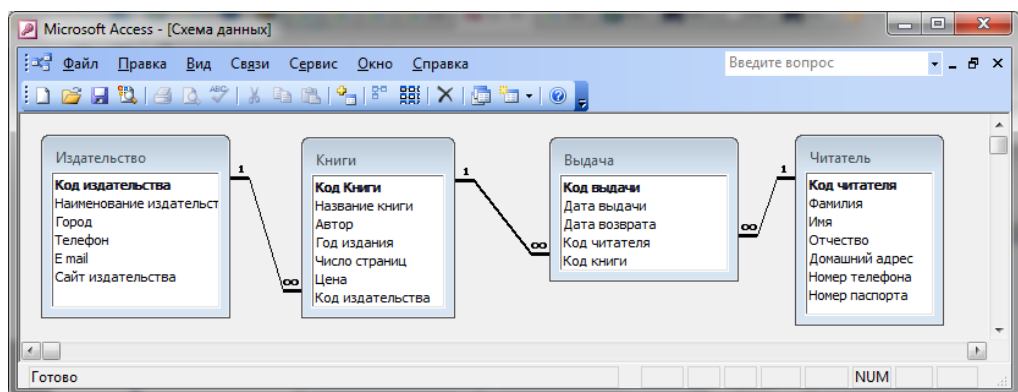


Рисунок 8 - Схема отношений

Отчет должен содержать:

1. Название, цель и задание лабораторной работы.
2. Анализ предметной области.
3. Инфологическую модель предметной области.
4. Ответы на контрольные вопросы.

Контрольные вопросы:

1. Что такое схема отношений?
2. Какие типы данных существуют.
3. Перечислить способы подстановок данных в таблицы.
4. Что такое сообщение на значение?
5. Дайте определение понятиям «Тип сущности», «Экземпляр сущности», «Ключ», «Связь».
6. Какие типы связей существуют?

Лабораторная работа № 3
Тема: «Размещение новых объектов в таблице»

Цель лабораторной работы:

- изучение способов создания масок и форматов полей в таблицах БД, также способов размещения объектов OLE;
- развитие познавательного интереса;
- изучение новых алгоритмов решения задач;
- формирование универсальных учебных действий, связанных с поиском информации, необходимой для решения поставленной задачи.

Рекомендации по выполнению лабораторных работ: лабораторная работа выполняется группой студентов (2-3человека) средствами СУБД MS Access. Данная работа может выполняться на основе результатов функционального моделирования предметной области.

Количество часов: 2 часа.

Постановка задачи

1. Используя свои таблицы, установить маски ввода и форматы для полей этих таблиц в соответствии с пунктами задания.
2. В текстовом формате написать маску для ввода в базу данных значений, содержащих буквенный и цифровые данные (например, номер паспорта А23476-ГА №023896, или регистрационный номер машины КАМ-456, или библиотечный шифр ISBN-56738).
3. В числовом формате задать шаблон ввода:
\$#,##0.00[Синий];(\$#,##0.00)[Красный];"Zero"[Зеленый];"Null"
4. Создать формат текстового поля так, чтобы можно было вводить любое число необязательных строчных символов, а если поле не заполнено, выводить синим цветом слово «Null».
5. Создать формат поля МЕМО так, чтобы можно было вводить любое число обязательных текстовых знаков красного цвета.
6. В формате дата/время установить один из шаблонов ввода данных (можно придумать свой):

№ задания	Значение
1	ddd", "mmm d", "уууу
2	mddd dd", "уууу
3	"Номер текущей недели — "ww
4	"Сегодня "dddd

7. Вставить в поле объект OLE одним из способов: внедрением или связав с файлом.
8. Установить в поле таблицы условие на значение так, чтобы при вводе ошибочных данных выводилось сообщение об ошибке.

Пример выполнения работы
Размещение объекта OLE

Рассмотрим размещение объекта OLE в таблице «Врачи» на примере поля «ФотоВрача». В нашем примере фотографии хранятся в формате графического редактора Paint в файлах с расширением bmp. (Создайте несколько таких объектов и сохраните их в той же папке, где хранится база данных).

Откройте таблицу «Врачи» в режиме Конструктора и добавьте новое поле «ФотоВрача». Тип данных – Поле объекта OLE.

Перейдите в режим Таблицы. В графе «Фото» выполните команду правой кнопкой мыши: Добавить объект.

В окне Вставка объекта (рис. 1) отметьте переключатель «Создать из файла». Окно Вставка объекта позволяет ввести имя файла с фотографией. Для поиска файла воспользуйтесь кнопкой Обзор.

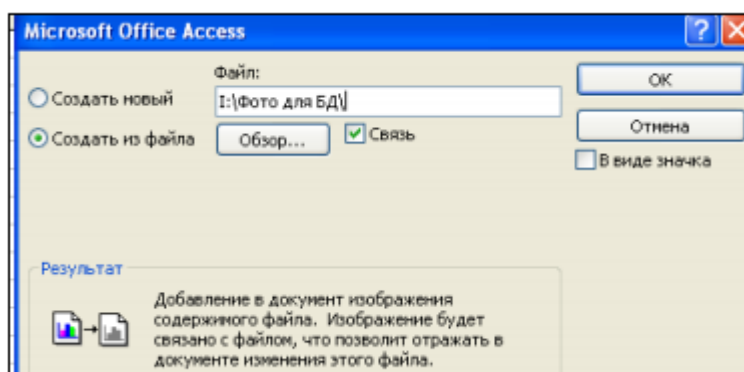


Рисунок 1 – Добавление объекта

Для введения в поле связанного объекта надо установить флажок «Связь», что позволит в дальнейшем отражать в документе изменения этого файла.

Текстовые и МЕМО поля

Для создания специальных форматов текстовых и МЕМО полей используются следующие символы.

Символ	Описание
@	Обязательный текстовый знак или пробел.
&	Необязательный текстовый знак.
<	Преобразует все знаки в строчные.
>	Преобразует все знаки в прописные.

Символы < и > относятся ко всему полю, а @ и & - соответствуют каждому символу поля

Специальные форматы для текстовых полей и полей МЕМО могут включать один или два раздела. Эти разделы содержат спецификации формата различных данных в поле.

Раздел	Описание
Первый	Формат отображения текста.
Второй	Формат отображения пустых строк и пустых (Null) значений.

Например, для помещения в элемент управления поле слова «Пусто», когда в поле отсутствует строка, в качестве значения свойства Формат поля (Format) допустимо применение специального формата @;"Пусто". Символ @ определяет отображение текста, введенного в поле, второй раздел — слово «Пусто», появляющееся, если в поле находится пустая строка или пустое (Null) значение.

Ниже приведены примеры специальных текстовых и МЕМО форматов.

Значение	Данные	Вид на экране
@ @ @ - @ @ - @ @ @ @	465043799	465-04-3799
@ @ @ @ @ @ @ @ @ @	465-04-3799 465043799	465-04-3799 465043799
>	фывапрол ФЫВАПРОЛ Фывапрол	ФЫВАПРОЛ ФЫВАПРОЛ ФЫВАПРОЛ
<	фывапрол ФЫВАПРОЛ Фывапрол	фывапрол фывапрол фывапрол
@;"Неизвестен"	Пустое (Null) значение	Неизвестен

Числовые и денежные поля

В следующей таблице приводятся встроенные значения свойства Формат поля (Format) для числовых полей.

Значение	Описание
Основной	(Значение по умолчанию.) Числа отображаются так, как они были введены.
Денежный	Используется разделитель групп разрядов; используются настройки, заданные в диалоговом окне «Язык и стандарты» (панель управления Microsoft Windows) для отрицательных значений, символа денежной единицы, символов десятичного разделителя и десятичных разрядов.
Евро	Используется денежный формат с символом евро вне зависимости от символа денежной единицы, определенного в диалоговом окне «Язык и стандарты» (панель управления Microsoft Windows).
Фиксированный	Выводится по крайней мере один разряд; используются настройки, заданные в диалоговом окне «Язык и стандарты» (панель управления Microsoft Windows) для отрицательных значений, символа денежной единицы, символов десятичного разделителя и десятичных разрядов.
С разделителями разрядов	Используется разделитель разрядов; используются настройки, заданные в диалоговом окне «Язык и стандарты» (панель управления Microsoft Windows) для отрицательных значений, символов десятичного разделителя и десятичных разрядов.
Процентный	Значение умножается на 100; добавляется знак процента (%); используются настройки, заданные в диалоговом окне «Язык и стандарты» (панель управления Microsoft Windows) для отрицательных значений, символов десятичного разделителя и десятичных разрядов.
Экспоненциальный	Числа выводятся в экспоненциальной нотации.

Специальные форматы

Специальные числовые форматы могут включать в себя от одного до четырех разделов, отделенных друг от друга точкой с запятой (;). Каждый формат содержит спецификацию для различных типов числовых данных.

Раздел	Описание
Первый	Формат положительных чисел.
Второй	Формат отрицательных чисел.
Третий	Формат нулевых значений.
Четвертый	Формат пустых (Null) значений.

Например, возможно использование следующего специального денежного формата:
`##,##0.00[Green];($#,##0.00)[Red];"Zero";"Null"`

Этот числовой формат имеет четыре раздела, разделенные точкой с запятой (;), и в каждом разделе используется различный формат.

Если используется несколько разделов, но формат каждого из них не указан, в результате либо не будет никакого форматирования, либо по умолчанию будет использовано форматирование первого из них.

Специальные числовые форматы создаются с помощью следующих символов.

Символ	Описание
. (точка)	Десятичный разделитель. Разделители выбираются двойным щелчком в диалоговом окне «Язык и стандарты» (панель управления Microsoft Windows).
, (запятая)	Разделитель групп разрядов.
0	Прототип разряда. Выводится цифра или 0.
#	Прототип разряда. Выводится цифра или ничего не выводится.
\$	Выводится текстовая константа «\$».
%	Процентный формат. Число умножается на 100 и к нему добавляется знак процента.
E- или e-	Экспоненциальная нотация (перед отрицательными показателями степени выводится знак минус, а перед положительными ничего). Этот символ используется только вместе с другими символами (например, 0.00E-00 или 0.00E00).
E+ или e+	Экспоненциальная нотация (перед отрицательными показателями степени выводится знак минус, а перед положительными знак плюс). Этот символ используется только вместе с другими символами (например, 0.00E+00).

Ниже приведены примеры специальных числовых форматов.

Значение	Описание
0;(0);;"Пусто"	Положительные значения отображаются стандартно; отрицательные — в скобках; слово «Пусто» выводится для пустых значений (Null).
+0.0;-0.0;0.0	Выводит знак плюс (+) или минус (-) с положительными или отрицательными числами; в случае нулевого значения выводится 0.0.

Поля Дата / время

В следующей таблице приводятся встроенные значения свойства Формат поля (Format) для полей даты/времени.

Значение	Описание
Полный формат даты	(Значение по умолчанию.) Если значение содержит только дату, то время не отображается; если значение содержит только время, то дата не отображается. Данный формат является комбинацией двух: «Краткий формат даты» и «Длинный формат времени». Примеры: 01.11.95 1:07:19 и 23.01.96 23:01:04.
Длинный формат даты	Совпадает с настройкой «Полный формат», задаваемой в диалоговом окне Язык и стандарты (панель управления Microsoft Windows). Пример: 1 Июнь 1995 г.
Средний формат даты	Пример: 03-апр-95.
Краткий формат даты	Совпадает с настройкой «Краткий формат», задаваемой в диалоговом окне Язык и стандарты (панель управления Microsoft Windows). Пример: 11.06.95. Внимание! При использовании значения краткого формата даты предполагается, что даты из диапазона 01.01.00 и 31.12.29 относятся к двадцать первому веку (то есть от 2000 до 2029 года). Также предполагается, что даты из промежутка 01.01.30 и 31.12.99 относятся к двадцатому веку (то есть от 1930 до 1999 года).

Длинный формат времени	Совпадает с форматом времени, заданном на вкладке Время (панель управления Microsoft Windows, диалоговое окно Язык и стандарты). Пример: 20:58:10.
Средний формат времени	Пример: 05:34 PM.
Краткий формат времени	Пример: 17:34.

Специальные форматы

Специальные форматы даты и времени создаются с помощью следующих символов.

Символ	Описание
: (двоеточие)	Разделитель компонентов времени. Знак разделителя задается в диалоговом окне Язык и стандарты (панель управления Microsoft Windows).
/	Разделитель компонентов даты.
c	Задаёт встроенный «Полный формат даты».
d	Номер дня месяца, состоящий из одной или двух цифр (1-31).
dd	Номер дня месяца, состоящий из двух цифр (01-31).
ddd	Сокращенное название дня недели (Пн-Вс).
dddd	Полное название дня недели (понедельник-воскресенье).
ddddd	Задаёт встроенный «Краткий формат даты».
dddddd	Задаёт встроенный «Длинный формат даты».
w	Номер дня недели (1-7).
ww	Номер недели в году (1-53).
m	Номер месяца, состоящий из одной или двух цифр (1-12).
mm	Номер месяца, состоящий из двух цифр (01-12).
mmm	Первые три буквы названия месяца (янв-дек).
mmmm	Полное название месяца (Январь-Декабрь).
q	Номер квартала в году (1-4).
y	Номер дня в году (1-366).
yy	Последние две цифры номера года (01-99).
yyyy	Полный номер года (0100-9999).
h	Число часов, состоящее из одной или двух цифр (0-23).
hh	Число часов, состоящее из двух цифр (00-23).
n	Число минут, состоящее из одной или двух цифр (0-59).
nn	Число минут, состоящее из двух цифр (00-59).
s	Число секунд, состоящее из одной или двух цифр (0-59).
ss	Число секунд, состоящее из двух цифр (00-59).
tttt	Задаёт встроенный «Длинный формат времени».
AM/PM	12-часовой формат времени с добавлением прописных букв «AM» или «PM».
am/pm	12-часовой формат времени с добавлением строчных букв «am» или «pm».
A/P	12-часовой формат времени с добавлением прописных букв «A» или «P».
a/p	12-часовой формат времени с добавлением строчных букв «a» или «p».
AMPM	12-часовой формат времени; используется индикатор «утро/день», выбранный в диалоговом окне Язык и стандарты (панель управления Microsoft Windows).

Специальные форматы выводятся в соответствии со значениями, установленными в диалоговом окне Язык и стандарты (панель управления Microsoft Windows). Специальные форматы, противоречащие настройкам диалогового окна Язык и стандарты, игнорируются.

Примечание. Для добавления в специальный формат запятой или другого знака разделителя следует ввести этот знак в кавычках: mmm d", "уууу.

Ниже приведены примеры специальных форматов даты и времени.

Значение	Вид на экране
ddd", "mmm d", "уууу	Пн, июн 2, 1997
mmmm dd", "уууу	Июнь 02, 1997
"Номер текущей недели — "ww	Номер текущей недели — 22
"Сегодня "dddd	Сегодня четверг

Для отображения индикаторов «A.D.» до или «B.C.» после года, зависящего от знака введенного числа, можно использовать специальный формат. Чтобы увидеть, как работает этот специальный формат, создайте новое поле таблицы с числовым типом данных и введите формат:

"A.D. " #;# " B.C."

Положительные числа отображаются как «A.D.» с последующим годом. Отрицательные — год, а затем «B.C.».

Поля логического типа

Самый скромный выбор встроенных форматов у полей логического типа – их всего три: Истина\Ложь, Да\Нет (по умолчанию), Вкл\Выкл.

Для логического Да программа Access сохраняет значение -1, для Нет – значение 0.

Специальный логический формат включает три раздела, однако первый элемент представлен только символом «;», второй - задает строковое значение, заменяющий Да, Истина, Вкл, а третий – строковое значение, заменяющее Нет, Ложь, Выкл.

Например, формат ;"Имеется"[Синий];"Отсутствует"[Красный] позволит для поля НаличиеТовара выводить синим цветом текст Имеется, а красным – Отсутствует.

Общие символы для форматов любого типа

Символ	Описание
(Пробел)	Выводит пробел как текстовую константу
“ABC”	Все символы внутри кавычек считаются текстовыми константами
!	Выравнивание по левому краю
*	Выравнивает текст по правому краю, заполняя его символами, указанными после *
\	Выводит следующий символ как текстовую константу
[цвет]	Отформатированные данные отображаются с использованием цвета в скобках: Черный, Белый, Синий, Зеленый, Бирюзовый, Красный, Лиловый, Желтый

Другие типы полей

Счетчик	Уникальные последовательно возрастающие (на 1) или случайные числа, автоматически вводящиеся при добавлении каждой новой записи в таблицу. Значения полей типа счетчика обновлять нельзя. Подробнее см.	4 байта (16 байт только если для свойства Размер поля (FieldSize))
---------	---	--

	в разделе справки для свойства Новые значения.	задано значение «Код репликации»).
Поле объекта OLE	Объект (например, электронная таблица Microsoft Excel, документ Microsoft Word, рисунок, звукозапись или другие данные в двоичном формате), связанный или внедренный в таблицу Microsoft Access.	До 1 Гбайт (ограничивается объемом диска).
Гиперссылка	Текст или комбинация текста и чисел, хранимые как текст и используемые в качестве адреса гиперссылки. Адрес гиперссылки может состоять максимум из трех частей: текст — текст, выводимый в поле или в элементе управления; адрес — путь к файлу (в формате пути UNC) или странице (URL); дополнительный адрес — расположение внутри файла или страницы; подсказка — текст, отображаемый в виде всплывающей подсказки; Чтобы вставить адрес гиперссылки в поле или в элемент управления, выберите команду Гиперссылка (меню Вставка).	Длина каждой из трех частей гиперссылки не более 2048 знаков.
Мастер подстановок	Создает поле, позволяющее выбрать значение из другой таблицы или из списка значений с помощью списка или поля со списком. Выбор этого параметра запускает мастер подстановок, создающий Поле подстановок. После завершения работы мастера устанавливается тип данных, основанный на значениях, выбранных в мастере.	Тот же размер, что и у первичного ключа поля, используемого в подстановке, обычно 4 байта.

Использование масок ввода

Маски ввода используются в полях (в таблицах и запросах), в текстовых полях и в полях со списком (в формах) для форматирования данных и управления вводимыми значениями. Значение свойства Маска ввода (InputMask) состоит из текстовых знаков (таких как пробелы, точки, тире, скобки) и специальных знаков, определяющих тип значений, которые могут быть введены в данную позицию. В основном маски ввода используются в текстовых полях и полях даты/времени, а также в числовых и денежных.

В приведенной ниже таблице указано, как Microsoft Access интерпретирует знаки, содержащиеся в свойстве Маска ввода (InputMask). Чтобы включить в маску текстовые константы, отличные от представленных в таблице, в том числе знаки и пробелы, следует просто ввести их в нужную позицию. Чтобы включить один из следующих знаков в качестве текстовой константы, необходимо перед ним ввести знак обратной косой черты (\).

Знак	Описание
0	Цифра (от 0 до 9, ввод обязателен; знаки плюс [+] и минус [-] не допускаются).
9	Цифра или пробел (ввод не обязателен; знаки плюс и минус не допускаются).
#	Цифра или пробел (ввод не обязателен; пустые знаки преобразуются в пробелы, допускаются знаки плюс и минус).
L	Буква (от A до Z или от A до Я, ввод обязателен).
?	Буква (от A до Z или от A до Я, ввод не обязателен).

A	Буква или цифра (ввод обязателен).
a	Буква или цифра (ввод необязателен).
&	Любой знак или пробел (ввод обязателен).
C	Любой знак или пробел (ввод необязателен).
.,:;- /	Десятичный разделитель и разделители тысяч, значений дат и времени. (Отображаемый знак зависит от настроек языка и стандартов на панели управления Microsoft Windows.)
<	Указывает перевод всех следующих знаков на нижний регистр.
>	Указывает перевод всех следующих знаков на верхний регистр.
!	Указывает заполнение маски ввода справа налево, а не слева направо. Заполнение маски знаками всегда происходит слева направо. Восклицательный знак в маске ввода можно помещать в любую позицию.
\	Указывает ввод любого следующего знака в качестве текстовой константы. Используется для отображения всех перечисленных в данной таблице знаков как текстовых констант (например, \A выводится как знак «А»).
Пароль	Значение Пароль, заданное для свойства Маска ввода (InputMask), создает поле для ввода пароля. Любой знак, введенный в поле, сохраняется как знак, но отображается как звездочка (*).

В приведенной ниже таблице указаны описания некоторых масок ввода и примеры значений, которые в них могут быть введены.

Описание маски ввода	Примеры значений
(000) 000-0000	(206) 555-0248
(999) 999-9999!	(206) 555-0248 () 555-0248
(000) AAA-AAAA	(206) 555-TELE
#999	-20 2000
>L????L?000L0	REENGR339M3 МАЙ Р 452Ю7
>L0L 0L0	T2Ф 8M4
00000-9999	98115- 98115-3007
>L<??????????????	Мария Иван
ISBN 0-#####-0	ISBN 1-55615-507-7 ISBN 0-13-964262-5
>LL00000-	DB51392-0493

Проверка вводимых в поле значений

Некоторую часть работы по проверке допустимости значений данных программа Access выполняет автоматически. Однако ограничиваться только этими проверками не стоит. Характер проверки задается в поле Условие на значение области Свойства поля, и если требования условия на значение не соблюдаются, то выдается сообщение об ошибке. Текст такого сообщения можно определить в опции Сообщение об ошибке.

Отчет должен содержать:

1. Название, цель и задание лабораторной работы.
2. Иллюстрации конструктора таблиц с масками или форматами полей и результата выполнения.
3. Ответы на контрольные вопросы.

Контрольные вопросы:

1. Что такое маска ввода?
2. Какие форматы полей можно установить для полей таблицы в Microsoft Access?
3. Как создаются форматы и маски полей средствами Microsoft Access?

Лабораторная работа № 4

Тема: «Создание форм базы данных в режиме конструктора.
Создание вычисляемых полей в форме. Создание кнопок на форме»

Цель лабораторной работы:

- изучение способов создания форм базы данных, элементов управления данными в форме;
- развитие познавательного интереса;
- изучение новых алгоритмов решения задач;
- формирование универсальных учебных действий, связанных с поиском информации, необходимой для решения поставленной задачи.

Рекомендации по выполнению лабораторных работ: лабораторная работа выполняется группой студентов (2-3человека) средствами СУБД MS Access.

Количество часов: 2 часа.

Постановка задачи

1. Для всех таблиц из лабораторной № 3 «Таблицы» создать формы с помощью мастера соответствующие составные формы.
2. Создать раскрывающиеся списки в формах.
3. На формах создать кнопки для закрытия формы, перехода к предыдущей и последующей записям.
4. Удалить с формы например:«Книги» поле <ТипОбложки> и создать группу из двух переключателей «твердая» и «мягкая». Заменить переключатели флажками (выделить, нажать правую кнопку мыши → <Преобразовать элемент в>).
5. В форме создать поле со списком для поиска книги по ее названию. Отсортировать названия «по возрастанию».
6. Создать свой стиль оформления полей, надписей, фона формы и сохранить его под названием «BEST».
7. Выполнить Поиск, сортировку, фильтрацию данных в форме.
8. Согласно выбранному вами варианту базы данных создайте диалоговые и главную кнопочную формы.

Пример выполнения работы

В тех случаях, когда пользователей не удовлетворяет представление данных в виде таблиц, они могут построить форму, которая позволит расположить данные на экране так, как это требуется. Кроме того, форма может использоваться для поиска данных. Если пользовательская задача требует документов определенного вида, то без форм обойтись практически невозможно. Форма позволяет вывести на экран больше данных, нежели таблица. Это достигается при помощи объектов, обеспечивающих «трехмерность» данных, например, раскрывающихся списков. Таким образом, формы используют для ввода, просмотра, редактирования, удаления данных; для создания интерфейса приложения; для поиска и выборки данных, удовлетворяющих определенным условиям, заданным пользователем.

Способы создания форм

Конструктор форм. Создание новой пустой формы в режиме конструктора.
Создание формы «Старинные монеты» в режиме конструктора:



Рисунок 1 - Форма «Старинные монеты» в режиме конструктора

Для этого в диалоговом окне Монеты: база данных перейти на вкладку Формы. Нажать кнопку Создать.

В диалоговом окне Новая форма выбрать вариант Конструктор, а в списке Выберите в качестве источника данных таблицу или запрос выбрать Монеты. Нажать кнопку ОК.

В главном меню выбрать пункт Вид, а в нем команду Заголовки / примечание формы. Нажать кнопку Панель элементов на панели инструментов Конструктор формы.

На панели элементов нажать кнопку Надпись и разместить объект Надпись в области Заголовок формы. Вписать в надпись Старинные монеты.

Для изменения свойств Надписи в главном меню выбрать пункт Вид, в нем команду Свойства, в диалоговом окне Раздел: заголовок формы изменить размер шрифта (14), цвет текста - синий.

Нажать кнопку Поле Аб на панели инструментов Панель элементов и разместить объект Поле в Области данных.

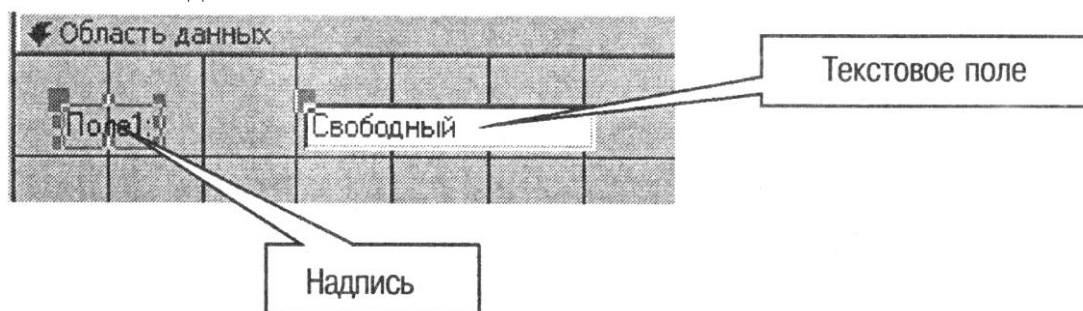


Рисунок 2 - Создание элемента управления «Надпись»

В надпись Поле1 ввести Название. В Текстовое поле указать данные из столбца Название, для этого в свойствах поля выбрать вкладку Данные, в списке Данные выбрать Название.

Аналогичные действия выполнить для оставшихся полей таблицы. Форма в режиме конструктора.

С помощью кнопки Вид-Режим формы перейти в режим Формы.

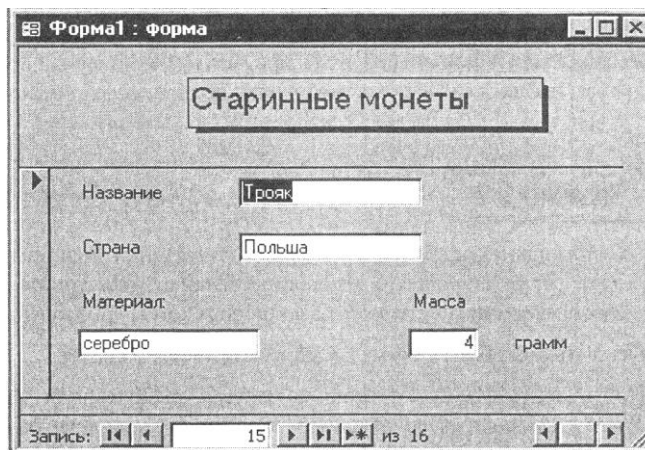


Рисунок 3 - Форма «Старинные монеты» в режиме формы

Мастер форм. Создание формы с расширенными инструментами.

Пустая форма. Создание формы для ввода данных по одной записи за раз.

Разделенная форма. В верхней части отображается таблица, а в нижней – форма для ввода данных в запись, выделенную в таблице. Сообщение об ошибке.

Режим таблицы. Она выводит поля в виде строк и столбцов, подобно таблицам.

Сводная диаграмма. Этот вид форм включает графические представления. Для ее создания нужно выбрать таблицу, содержащую числовые значения, которые можно представить в графическом виде.

Мастер по созданию форм. Стандартная форма создается системой только при наличии открытого окна базы данных. Средства конструирования форм доступны независимо от того, находится система в режиме проектирования или заполнения таблицы. Более того, таблица, для которой строится форма, может вообще быть закрытой. Для получения большей свободы выбора полей, отображаемых на форме, можно воспользоваться мастером форм.

Чтобы создать стандартную форму с помощью мастера форм, необходимо выполнить следующие действия:

В окне базы данных перейти в главное меню на вкладку <Создание> и нажать кнопку <Другие формы> в разделе <Формы> и выбрать <Мастер форм>.

В открывшемся диалоговом окне выбрать таблицу из списка созданных таблиц и поля для проектируемой формы. С помощью кнопки переноса (>) следует перенести нужные поля в список <Выбранные поля> и, нажав кнопку <Далее>, перейти к следующему окну мастера форм

В следующем окне активизировать форму <В один столбец> и нажать кнопку <Далее>.

Назначить имя заголовка и нажать кнопку <Готово>.

Выбрать стиль оформления формы (например, <Стандартная>).

Сохранить форму. Форма необязательно должна включать все поля таблицы: сколько их будет зависит от вашего желания.

Фильтрация информации в форме

При использовании формы пользователь, как и при работе с таблицами и запросами, может применять функцию поиска или замены. В качестве альтернативы запросам можно использовать фильтры для селекции записей, удовлетворяющих сформулированным критериям, т. е. «записей, проходящих через фильтр». В отличие от селекции записей с помощью запросов отдельные поля при селекции с помощью фильтров не подавляются. В них также нельзя выполнять вычисления. Фильтр служит только для селекции или сортировки записей. К формам применимы три способа фильтрации данных:

Фильтр для форм – фильтрация данных в форме с помощью простых условий выборки (на основе логического оператора <Or>). Для изготовления фильтра воспользуйтесь кнопкой <Фильтр> на вкладке <Главная> в разделе <Сортировка и фильтр>.

Фильтр для выделенного – фильтрация данных в форме с помощью выделения данных. Для изготовления фильтра данным способом используйте кнопку <Фильтр для выделенного>, но прежде необходимо выделить данные, которые будут служить критерием фильтрации.

Расширенный фильтр – расширенное построение фильтра для формы. Необходимо выбрать кнопку <Параметры расширенного фильтра> и в раскрывающемся списке выбрать <Расширенный фильтр>.

Конструктор форм Конструктор форм представляет комплекс средств для проектирования формы и включает в себя:

Представление проекта – окно проектирования формы.

Конструктор – панель инструментов конструктора форм.

Сервис – панель, позволяющая открыть окно свойств элементов, окно со списком полей и окно с кодом. Элементы управления – панель элементов управления.

Установка свойств

Внешний вид, структура и режимы работы отдельных управляющих элементов и форм определяются значениями характеристик этих объектов. Характеристики объектов доступны в окне характеристик, которое обычно выводится на экран при запуске конструктора форм. Для установки или изменения характеристик (свойств) элемента его надлежит маркировать и в меню <Инструменты конструктора форм> на вкладке <Конструктор> активизировать опцию <Страница свойств> рисунок 4.

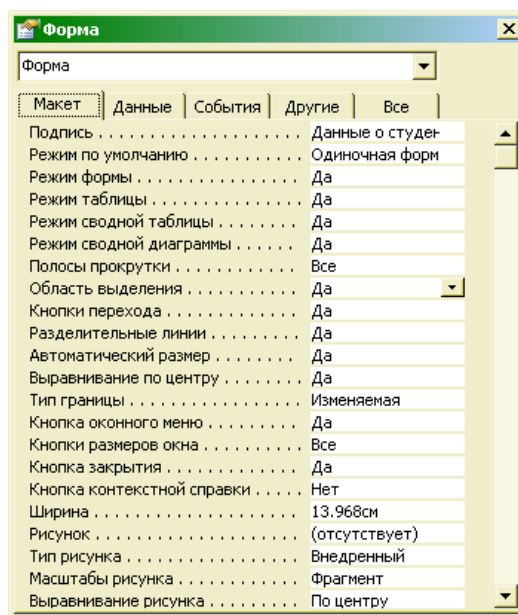


Рисунок 4 - Окно свойства формы

Окно характеристик имеет многостраничную структуру, позволяющую группировать характеристики, имеющие родственные функциональные признаки, по разделам. В окне находятся следующие разделы с характеристиками:

Макет. Характеристики, описывающие формат, в котором выводится тот или иной элемент. Данные. Характеристики, относящиеся к источнику исходных данных.

События. Характеристики, определяющие реакцию элемента на те или иные события.

Другие. Остальные характеристики, не вошедшие в предыдущие.

Все. Характеристики, относящиеся к данному элементу.

Управляющие элементы

Элементы управления – панель элементов управления. При проектировании формы в нее можно вставить управляющие элементы. Каждый управляющий элемент может быть

независимым, т. е. несвязанным, или зависимым – связанным. Элемент может быть связан с некоторым объектом (полем) в таблице (или запросе), на которой базируется форма. Например, связанное с некоторым полем записи таблицы текстовое поле формы будет постоянно отображать содержимое «своего» поля активной записи таблицы. Выбранные и размещенные в окне проектирования формы элементы управления приобретают свой реальный вид только в режиме заполнения или в режиме предварительного просмотра перед печатью.

В режиме же проектирования они обычно представлены в виде условных обозначений, зачастую близких к реальному виду обозначаемых элементов.

Связывание элемента с полем таблицы Существуют два способа связывания элемента с полем таблицы:

Если необходимо изготовить связанный управляющий элемент автоматически, то при создании и размещении элемента, кроме панели элементов, следует использовать еще и окно списка полей. В окне панели элементов с помощью мыши выбирается управляющий элемент, который необходимо вставить в форму, а затем из списка полей выбирается поле, с которым следует связать создаваемый элемент. После этого, не отпуская кнопку мыши, нужно перетащить выбранное поле в то место проекта формы, в котором надлежит разместить управляющий элемент. После того как кнопка мыши будет отпущена, в указанном месте появится элемент выбранного типа, привязанный к указанному полю. Такой управляющий элемент будет иметь те же характеристики, что и поле в таблице, с которым он связан.

Связать с полем таблицы можно и ранее несвязанный элемент. Для этого достаточно вызвать окно свойств двойным щелчком на этом элементе в окне проектирования формы. В окне свойств в разделе <Данные> следует найти характеристику Источник для управляющего элемента и задать в качестве ее значения имя связываемого поля.

Создание диаграмм

Диаграммы можно создать двумя способами:

С помощью <Мастера диаграмм> можно изготовить любые диаграммы, используя числа, полученные, например, путем применения функций вычислений внутри запроса.

Создайте новую форму на основании таблицы, из которой будут использоваться данные.

В режиме конструктора форм на панели <Инструменты конструктора форм> – <Конструктор> в разделе <Элементы управления> нажмите кнопку <Вставить диаграмму>, затем щелкните по любому месту формы для вызова мастера диаграмм. Следуя указаниям мастера диаграмм, создайте свою диаграмму. На вкладке <Создание> в разделе <Формы> выберите кнопку <Сводная диаграмма>. Появляется вкладка для создания диаграммы <Работа со сводными диаграммами> – <Конструктор>.

На данной вкладке можно поместить на диаграмму нужные числовые поля, выбрать вид диаграммы и задать ей заголовок.

Сортировка записей

Пользователь может использовать фильтр, чтобы отсортировать записи таблицы с помощью формы по содержимому некоторого поля в возрастающей (А – Я, 0 – 100) или убывающей (Я – А, 100 – 0) последовательности.

Отчет должен содержать:

1. Название, цель и задание лабораторной работы.
2. Иллюстрации конструктора таблиц с масками или форматами полей и результата выполнения.
3. Ответы на контрольные вопросы.

Контрольные вопросы:

1. Какие виды форм существуют?
2. Какие существуют управляющие элементы и для чего предназначен каждый из них?
3. Что такое составная форма?
4. Перечислите способы создания составных форм.
5. Как создать диалоговую форму.

Лабораторная работа № 5
Тема: «Создание подчиненных форм»

Цель лабораторной работы:

- изучение способов создания подчиненных форм базы данных;
- развитие познавательного интереса;
- изучение новых алгоритмов решения задач;
- формирование универсальных учебных действий, связанных с поиском информации, необходимой для решения поставленной задачи.

Рекомендации по выполнению лабораторных работ: лабораторная работа выполняется группой студентов (2-3 человека) средствами СУБД MS Access.

Количество часов: 2 часа.

Постановка задачи

1. Для таблиц созданных в процессе выполнения предыдущих лабораторных работ создать формы с помощью мастера. Главные формы – в столбец и подчиненные в ленточном виде разными способами.
2. Создать главную кнопочную форму для базы данных.
3. Установить, запуск главной кнопочной формы при открытии файла базы данных.

Пример выполнения работы

1) Главные и подчиненные формы

Работая с реляционных данных (связанные данные, которые хранятся в отдельных таблицах), часто требуется просмотреть данные из нескольких таблиц или запросов в одной форме. Например можно просмотреть запись клиента из одной таблицы и сведения о заказах этого клиента из другой таблицы одновременно. Подчиненные формы - это удобный инструмент для этого и Microsoft Access предоставляет несколько способов, которые помогут быстро создать.

Подчиненной называют форму, вставленную в другую форму. Главная форма называется основной. Комбинацию «форма и подчиненная форма» иногда называют также иерархической формой или комбинацией родительской и дочерней форм.

Подчиненные формы особенно удобны для отображения данных из таблиц или запросов с отношением «один-ко-многим». Отношение «один-ко-многим» представляет собой связь между двумя таблицами, при которой значение первичного ключа каждой записи в главной таблице соответствует значению в соответствующем поле или полях нескольких записей в связанной таблице. Например, можно создать форму, в которой отображаются сведения о сотрудниках и которая содержит подчиненную форму с заказами каждого сотрудника. Данные в таблице «Сотрудники» являются в отношении стороной «один». Данные в «Заказы» являются в отношении стороной «многие»— для каждого сотрудника может быть несколько заказов.

ИД заказ	Клиент	Дата разме	Дата отгрузки
41	Организация Ж	24.03.2006	
42	Организация К	24.03.2006	07.04.2006
43	Организация Л	24.03.2006	
44	Организация А	24.03.2006	

Рисунок 1 - Подчиненная форма «Сотрудники Заказы»

«Заказы» имеют отношение «многие» - для каждого сотрудника может быть несколько заказов.

В основной форме отображаются данные отношения "один".

Основная и подчиненная формы взаимодействуют таким образом, что в подчиненной выводятся только те записи, которые связаны с текущей записью в основной. Например, если в основной форме отображаются данные Юлии Ильиной, то в подчиненной отображаются только ее заказы. Если связь между основной и подчиненной формами была разорвана, то в подчиненной форме будут отображаться заказы всех сотрудников, а не только Юлии.

В таблице ниже даны определения некоторых терминов, связанных с подчиненными формами.

Термин	Определение
Элемент управления подчиненной формы	Элемент управления, внедряющий форму в форму. Является «представлением» другого объекта в базе данных, будь то другая форма, таблица или запрос. Элемент управления подчиненной формы предусматривает свойства, которые позволяют связывать данные, отображаемые в элементе управления, с данными в главной форме.
Свойство «Объект-источник»	Свойство элемента управления подчиненной формы, которое определяет, какой объект отображается в элементе управления.
Таблица	Простое отображение данных в строках и столбцах, как в листе. В элементе управления подчиненной формы отображается таблица, когда объектом-источником является таблица или запрос, либо форма, для свойства «Режим по умолчанию» которой задано значение «Режим таблицы». В этих случаях подчиненную форму иногда называют таблицей или подтаблицей.
Свойство «Подчиненные поля»	Свойство элемента управления подчиненной формы, которое задает поле или поля в подчиненной форме, которые будут связывать ее с основной.
Свойство «Основные поля»	Свойство элемента управления подчиненной формы, которое задает поле или поля в основной форме, которые будут связывать ее с подчиненной.

Для достижения наилучших результатов перед выполнением действий, описанных в задании, рекомендуется установить связи между таблицами. Это позволит приложению Access автоматически создать связи между основными и подчиненными формами. Чтобы просмотреть, добавить или изменить связи между таблицами в базе данных, на вкладке «Работа с базами данных» в группе «Отношения» нажмите кнопку «Схема данных».

Когда элемент управления подчиненной формы имеет в качестве объекта-источника форму, он содержит поля, помещенные в нее и его можно просмотреть как одиночную форму, ленточную форму или таблицу. Одним из преимуществ создания подчиненной формы на основе формы является возможность добавления вычисляемых полей в подчиненную форму, например: ([Количество]) * ([Цена]).

Вы можете вставлять таблицы или представление сводной таблицы в форме путем создания которого исходный объект элемента управления подчиненной формы, таблицы или запроса.

2) Создание или добавление подчиненной формы

В таблице ниже представлены рекомендации по выбору процедуры, которая лучше всего подходит для конкретной ситуации.

Ситуация	Рекомендуемая процедура
Нужно создать в приложении Access основную и подчиненную формы и связать их.	Создание формы, содержащей подчиненную форму, с использованием мастера форм
Нужно использовать существующую форму в качестве основной, но при этом создать в Access новую подчиненную форму и добавить ее в основную.	Добавление одной или нескольких подчиненных форм в существующую форму с помощью мастера подчиненных форм
Нужно использовать существующую форму в качестве основной и добавить в нее одну или несколько существующих форм в качестве подчиненных.	Создание подчиненной формы путем перетаскивания одной формы в другую

3) Создание формы, содержащей подчиненную форму, с использованием мастера форм

Данная процедура позволяет создать новую комбинацию формы и подчиненной формы с помощью мастера форм. Это также самый быстрый способ начать работу, если еще не были созданы формы, которые планируется использовать в качестве основной или подчиненной формы.

На вкладке Создание в группе Формы выберите команду Мастер форм.

На первой странице мастера в раскрывающемся списке Таблицы и запросы выберите таблицу или запрос.

Например, чтобы создать форму «Сотрудники» для отображения в подчиненной форме списка заказов для каждого сотрудника, выберите пункт Таблица: Сотрудники (сторона «один» отношения «один-ко-многим»).

Дважды щелкните поля таблицы или запроса, которые нужно включить в форму.

На той же странице мастера в раскрывающемся списке Таблицы и запросы выберите другую таблицу или запрос. В данном примере выберите таблицу Заказы (сторона «многие» отношения «один-ко-многим»).

Дважды щелкните поля таблицы или запроса, которые нужно включить в форму.

При нажатии кнопки Далее, при условии, что настроенные до связи правильной работы мастера, мастер спрашивает способ просмотра данных? - то есть, выберите таблицу или запрос. Выберите таблицу на стороне «один» связи "один ко многим". В этом примере для создания формы сотрудников, щелкните по сотрудникам. Мастер отображает небольшой схема формы. Страница

должен выглядеть приведенной ниже иллюстрации:

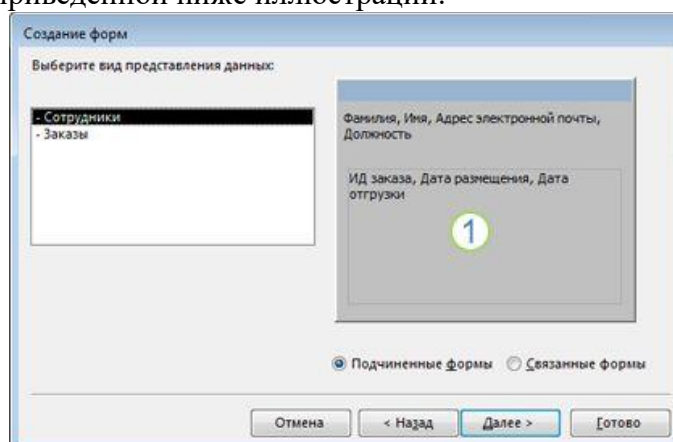


Рисунок 3 - Создание подчинённой формы.

Прямоугольник в нижней части изображения формы обозначает подчиненную форму.

На странице Выберите внешний вид подчиненной формы: выберите нужный вариант макета и нажмите кнопку Далее. В макетах обоих стилей данные подчиненной формы располагаются в виде строк и столбцов, но ленточный макет имеет больше возможностей настройки. В ленточной подчиненной форме можно добавлять цвет, изображения и другие элементы форматирования, однако табличный макет компактнее, как и табличное представление таблицы.

На последней странице мастера введите заголовки форм. Имена автоматически присваиваются формам на основе введенных заголовков, а надписи подчиненных форм — на основе введенных заголовков подчиненных форм.

Укажите режим, в котором нужно открывать форму: режим формы (чтобы можно было просмотреть или ввести данные) или режим конструктора (чтобы можно было изменить ее структуру). Затем нажмите кнопку Готово.

Будут созданы две формы - основная, содержащая подчиненную форму, и сама подчиненная форма.

Добавление одной или нескольких подчиненных форм в существующую форму с помощью мастера подчиненных форм. Процедура для добавления одной или нескольких подчиненных форм в существующую форму. Для каждой подчиненной формы можно создать новую форму или использовать существующую.

В области навигации щелкните существующую форму правой кнопкой мыши и выберите команду Конструктор.

На вкладке Конструктор в группе Элементы управления щелкните стрелку вниз, чтобы открыть коллекцию Элементы управления, и убедитесь в том, что установлен флажок Использовать мастера.

На вкладке Конструктор в коллекции Элементы управления нажмите кнопку Подчиненная форма/отчет.

В форме щелкните место, куда нужно поместить подчиненную форму.

Следуйте указаниям мастера.

После нажатия кнопки «Готово» в форму будет добавлена подчиненная форма. Если для подчиненной формы создается новая форма, а не используется существующая, новый объект формы будет создан и добавлен в область навигации.

Создание подчиненной формы путем перетаскивания одной формы в другую. Процедура, используется для того, чтобы существующая форма была в качестве основной и добавить в нее одну или несколько существующих форм в качестве подчиненных.

В области навигации щелкните правой кнопкой мыши форму, которую нужно использовать в качестве основной, и выберите пункт Конструктор.

Перетащите форму, которую нужно использовать в качестве подчиненной, из области навигации на основную форму. В основную форму будет добавлен элемент управления подчиненной формы, который будет связан с формой, добавленной путем перетаскивания из области навигации. Будет также выполнена попытка связать подчиненную форму с основной на базе отношений, определенных в базе данных.

Чтобы проверить, успешно ли создана связь, на вкладке Главная в группе Режимы нажмите кнопку Режим, выберите команду Режим формы и с помощью селектора записей основной формы просмотрите несколько записей. Если в подчиненной форме правильно отображаются данные для каждого сотрудника, это означает, что процедура завершена успешно.

Если проверка не работает, это означает, что программе Access не удалось создать связь подчиненной формы с основной и свойства Подчиненные поля и Основные поля элемента управления подчиненной формы остались пустыми. Задайте значения для этих свойств вручную, выполнив указанные ниже действия.

В области навигации щелкните основную форму правой кнопкой мыши и выберите команду Режим макета.

Щелкните элемент управления подчиненной формы один раз, чтобы выделить его.

Если область задач «Окно свойств» не отображается, нажмите клавишу F4.

В окне свойств откройте вкладку Данные.

Нажмите кнопку Построить кнопка  рядом с полем свойству Подчиненные поля.

Будет открыто диалоговое окно Связь с полями подчиненной формы.

В раскрывающихся списках Основные поля и Подчиненные поля выберите поля, с помощью которых нужно связать формы, и нажмите кнопку ОК. Если неизвестно, какие поля нужно использовать, нажмите кнопку Предложить. Будет предпринята попытка автоматически определить связывающие поля.

4) Главная кнопочная форма

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

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

Для одной базы данных можно создать несколько кнопочных форм. Кнопки следует группировать на страницах кнопочной формы таким образом, чтобы пользователю было понятно, в каких кнопочных формах можно выполнять определенные команды (запросы, отчеты, ввода и редактирования данных). Необходимо отметить, что на подчиненных кнопочных формах должны быть помещены кнопки возврата в главную кнопочную форму.

Технология создания кнопочных форм следующая:

создать страницу главной кнопочной формы (ГКФ);

создать необходимое количество страниц подчиненных кнопочных форм (например, формы для ввода данных, для отчетов, для запросов и т.д.);

создать элементы главной кнопочной формы;

создать элементы для кнопочных форм отчетов и форм ввода или изменения данных;

создать макросы для запросов или для таблиц с уникальными именами;

создать элементы для кнопочных форм запросов или таблиц.

Структура кнопочных форм может быть представлена в следующем виде.



Рисунок 3

Для создания главной кнопочной формы и ее элементов необходимо открыть базу данных, (например, «Успеваемость_ студентов») и выполнить команду Сервис / Служебные программы / Диспетчер кнопочных форм. Если кнопочная форма ранее не создавалась, то откроется окно диалога «Диспетчер кнопочных форм».

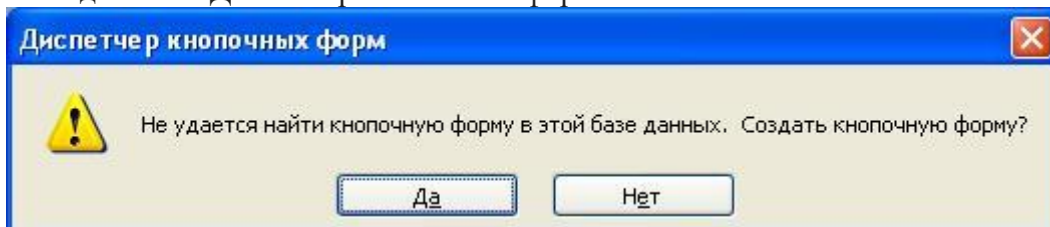


Рисунок 4

В окне диалога надо нажать кнопку «Да», тем самым подтвердить создание кнопочной формы. В результате будет создана страница Главной кнопочной формы.

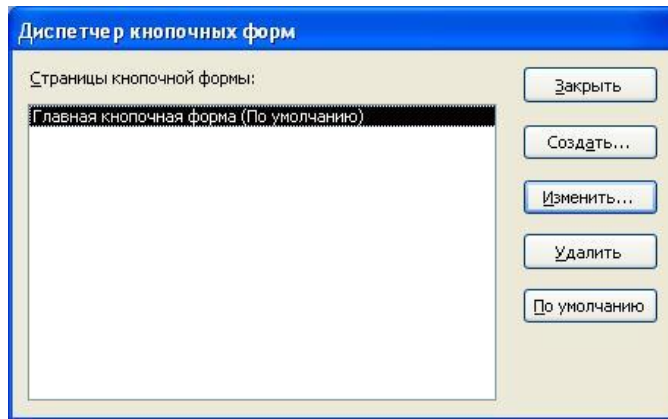


Рисунок 5

Далее можно создать еще три страницы кнопочной формы: Формы ввода данных, Отчеты и Запросы. Для этого следует щелкнуть на кнопке «Создать» и в появившемся окне ввести имя новой страницы «Формы ввода данных» и щелкнуть на кнопке «ОК».

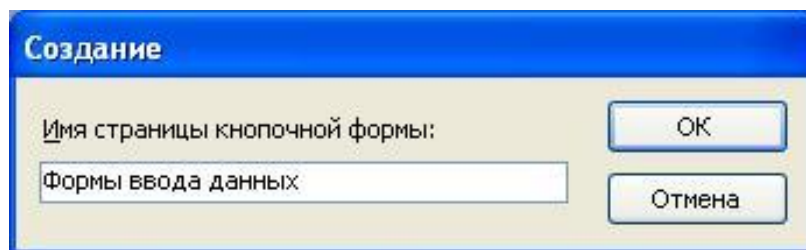


Рисунок 6

Будет создана страница кнопочной формы «Формы ввода данных». Аналогичным образом надо создать еще две страницы, в итоге получим четыре страницы кнопочных форм, которые отображаются в окне «Диспетчер кнопочных форм».

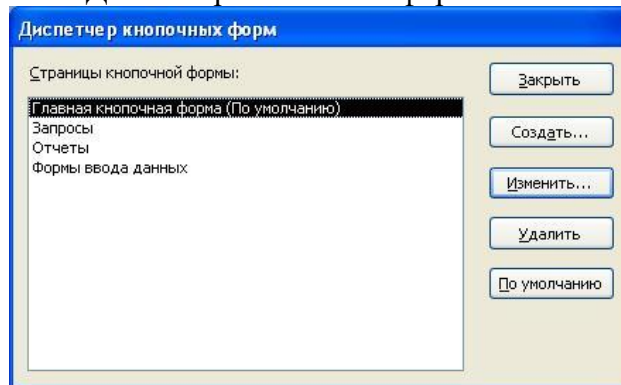


Рисунок 7

После этого создаем элементы ГКФ, для этого в «Окне диспетчер кнопочных форм» выделяем страницу «Главная кнопочная форма» и щелкаем «Изменить», откроется новое окно «Изменение страниц кнопочной формы».

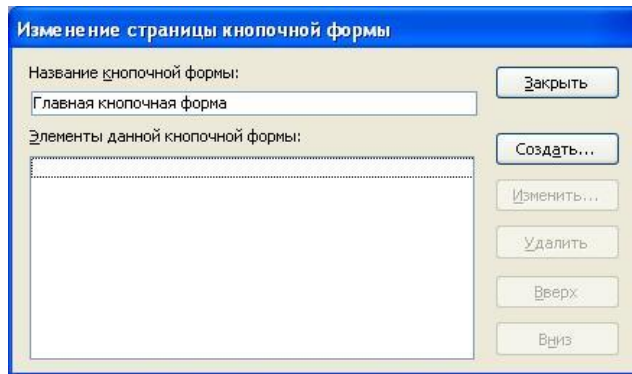


Рисунок 8

В этом окне щелкаем на кнопке «Создать», откроется новое окно «Изменение элемента кнопочной формы».



Рисунок 9

В окне выполняем следующее:
 вводим текст: **Формы для ввода данных**;
 выбираем из раскрывающегося списка команду: **Перейти к кнопочной форме**;
 выбираем из списка кнопочную форму: **Ввод данных в формы**, щелкаем на кнопке «ОК».

В окне «Изменение страницы кнопочной формы» отобразится элемент кнопочной формы «Формы для ввода данных».

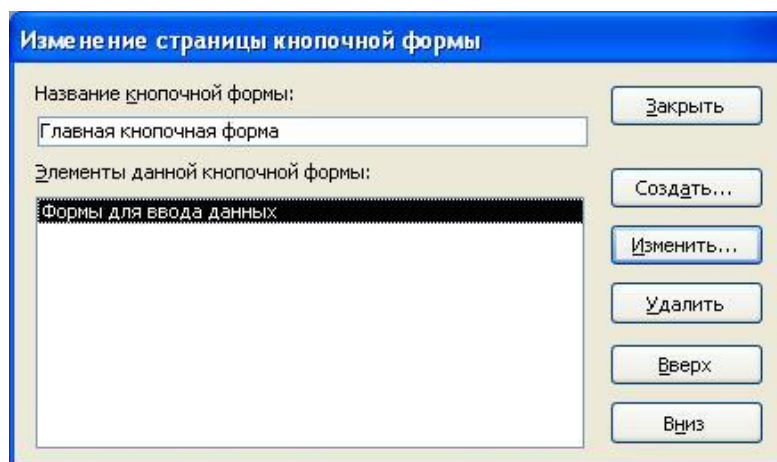


Рисунок 10

Аналогичным методом надо создать элементы: «Запросы» и «Отчеты», а затем элемент (кнопку) "Выход из БД".

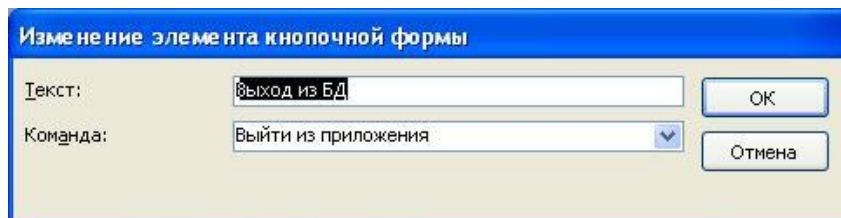


Рисунок 11

В результате в окне «Изменение страницы кнопочной формы» будут отображаться все элементы главной кнопочной формы.

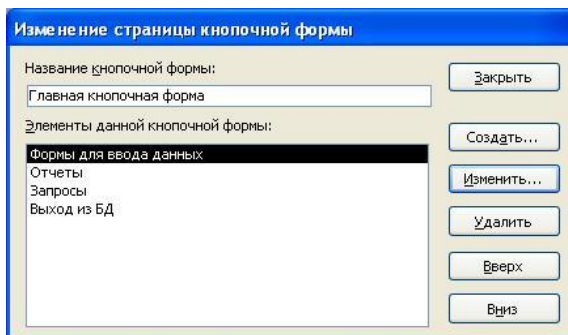


Рисунок 12

Кнопочная форма появится в списке в области окна базы данных на вкладке Формы на панели Объекты, а на вкладке Таблицы в списках появится таблица Switchboard Items. Дважды щелкнув на надписи "Кнопочная форма", откроется Главная кнопочная форма.

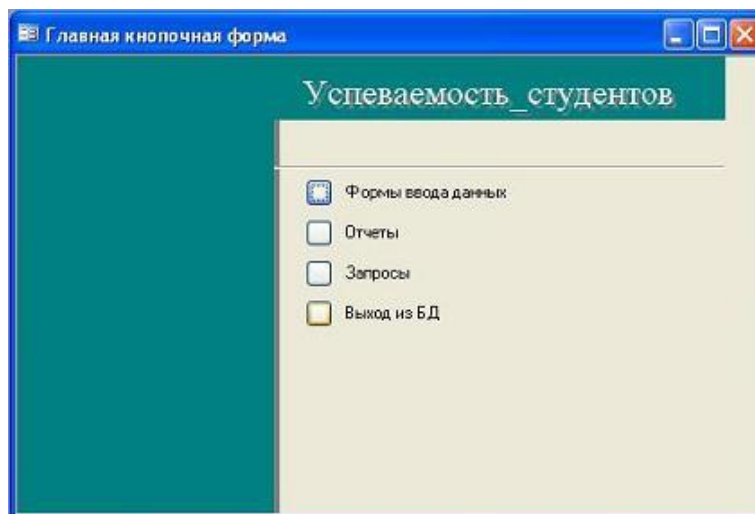


Рисунок 13

Для того чтобы эта форма отображалась при запуске базы данных, необходимо выполнить команду Сервис/Программы запуска, и в открывшемся окне выбрать "Кнопочная форма" из раскрывающегося списка, кроме того, надо снять флажки Окно базы данных и Строка состояния. Можно также ввести заголовок и значок приложения.

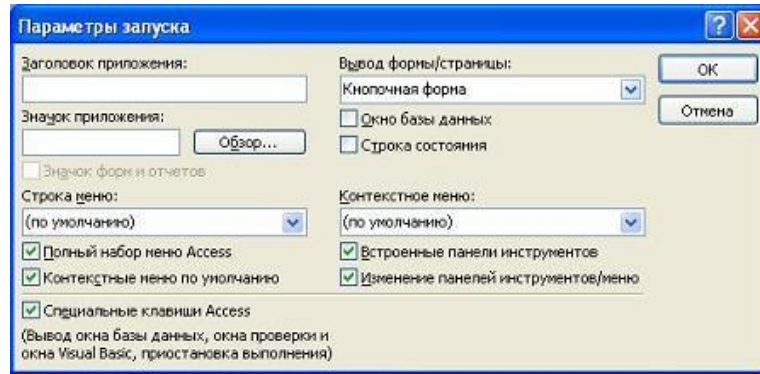


Рисунок 14

Отчет должен содержать:

1. Название, цель и задание лабораторной работы.
2. Иллюстрации конструктора таблиц с масками или форматами полей и результата выполнения.
3. Ответы на контрольные вопросы.

Контрольные вопросы:

1. Что такое подчиненная форма?
2. Какие способы создания подчиненных форм существуют?
3. Для чего предназначена главная кнопочная форма?
4. Автозапуск кнопочной формы при запуске базы данных.

Лабораторная работа № 6
Тема: «Формирование запросов на модификацию данных.
Создание перекрестных запросов»

Цель лабораторной работы:

- освоить приёмы создания запросов, научиться использовать построитель выражений для формирования новых полей в таблице запроса, изучить технологию формирования сложных запросов с использованием Мастера запросов;

- развитие познавательного интереса;

- изучение новых алгоритмов решения задач;

- формирование универсальных учебных действий, связанных с поиском информации, необходимой для решения поставленной задачи.

Рекомендации по выполнению лабораторных работ: лабораторная работа выполняется группой студентов (2-3 человека) средствами СУБД MS Access.

Количество часов: 2 часа.

Постановка задачи

1. Создать запрос на добавление записей в таблицу.
2. Создайте запрос на удаление записей.
3. Создать простой запрос на выборку, используя вариант предыдущий лабораторной работы.
4. Создать параметрический запрос.
5. Формирование запроса по неполному значению поля
6. Создать запрос для проведения вычислений в таблице (использовать возможности Построителя выражений).
7. Провести анализ таблиц базы данных по варианту и создайте перекрестный запрос.

Пример выполнения работы

1) Запросы-действия

Запрос на создание таблицы – запрос, который выполняет отбор записей из базы данных и сохраняет их как отдельную таблицу.

Запрос на добавление – запрос, который добавляет данные в существующую таблицу.

Запрос на обновление – запрос, который обновляет данные в существующей таблице.

Запрос на удаление – запрос, который удаляет данные, соответствующие условиям, из существующей таблицы.

Перекрестный запрос – запрос, который выполняет сведение данных по двум наборам значений, один из которых отображается в левой части таблицы, а другой – в верхней ее части.

Параметрический запрос – запрос, который выполняется пользователем регулярно при неизменных критериях отбора.

Запрос на создание таблицы. Запрос на создание таблицы получает данные из одной или нескольких таблиц, а затем помещает набор результатов в новую таблицу. Эта новая таблица может располагаться в базе данных, открытой в данный момент, или создается в другой базе данных.

Обычно запрос на создание таблицы создается, когда необходимо скопировать данные или поместить их в архив. Предположим, что существует таблица (или несколько таблиц) с данными по продажам за прошедшие периоды, и эти данные нужно использовать в отчетах. Эти данные не изменяются, поскольку все сделки были совершены, по меньшей мере, день назад. Многократное выполнение запроса для извлечения данных требует значительного времени, особенно если запрос сложный, а хранилище данных имеет большой объем. Загрузив данные в отдельную таблицу и используя эту таблицу в качестве источника данных, можно снизить нагрузку и получить удобный архив данных. В процессе работы следует

помнить, что данные в новой таблице являются копией исходных данных в определенный момент времени; они не имеют связи и не подключены к исходной таблице (или таблицам).

Процесс создания запроса на создание таблицы состоит из следующих основных этапов:

Включите содержимое базы данных, если она не подписана или не находится в надежном расположении. Иначе нельзя будет запускать запросы на изменение (запросы на добавление, на обновление или на создание таблицы).

В режиме конструктора запроса создайте запрос на выборку, а затем изменяйте этот запрос, пока он не будет возвращать нужные записи. Можно выбирать данные из нескольких таблиц и, фактически, денормализовать данные. Например, можно поместить данные о заказчиках, грузоотправителях и поставщиках в одну таблицу - это обычно не делается в производственной базе данных с правильно нормализованными таблицами. В запросе можно также использовать условия для более точной настройки или сужения набора результатов.

Преобразуйте запрос на выборку в запрос на создание таблицы, выберите расположение для новой таблицы и затем запустите запрос, чтобы создать таблицу. Не следует путать запрос на создание таблицы с запросами на обновление или добавление.

Запрос на обновление используется, когда необходимо добавить или изменить данные отдельных полей. Запрос на добавление используется, когда требуется добавить записи (строки) в набор записей в существующей таблице.

2) Преобразование запроса на выборку

Откройте запрос на выборку в режиме конструктора (рис. 1).

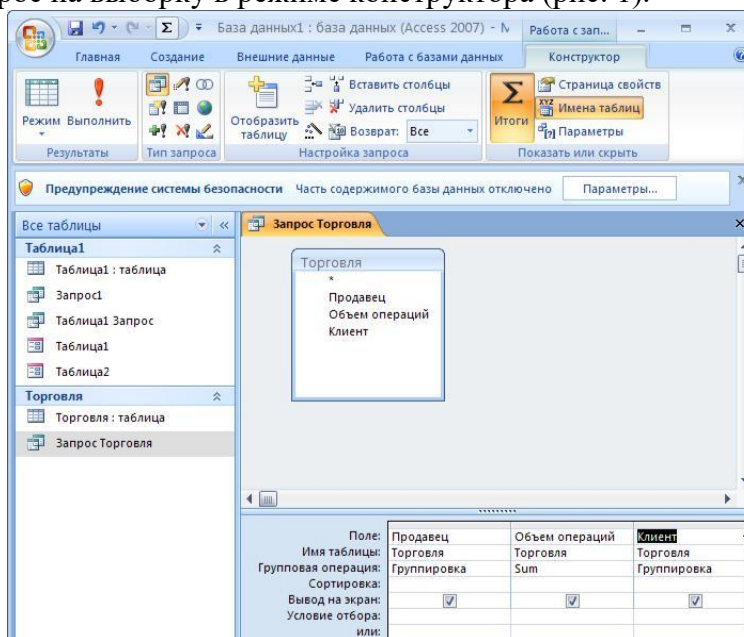


Рисунок 1 - Запрос на выборку в режиме конструктора

На вкладке <Конструктор> в группе <Тип запроса> выберите команду <Создание таблицы>. Откроется диалоговое окно <Создание таблицы> (рис. 2).

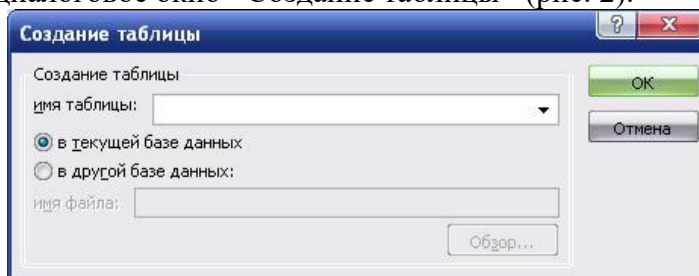


Рисунок 2 - Запрос на создание таблицы

В поле <Имя таблицы> введите имя новой таблицы, или выберите имя существующей таблицы (рис. 3).

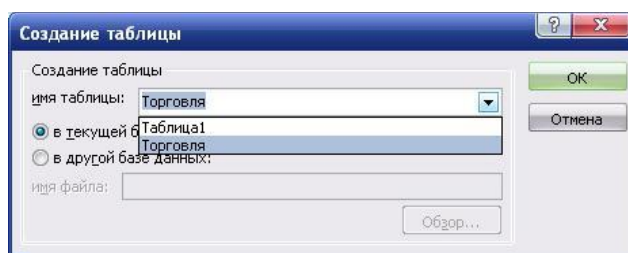



Рисунок 3 - Указание имени создаваемой таблицы

Выполните одно из следующих действий:

Поместите новую таблицу в текущую базу данных. Выберите параметр <Текущая база данных>, если он еще не выбран, и нажмите кнопку <ОК>.

Нажмите кнопку <Выполнить> , а затем – <Да> для подтверждения операции.

При замене существующей таблицы эта таблица сначала удаляется и перед этим запрашивается подтверждение на удаление. Нажмите кнопку <Да>, а затем – <Да> еще раз для создания новой таблицы. Поместите новую таблицу в другую базу данных. Выберите параметр <В другой базе данных>, и в поле <Имя файла> введите расположение и имя файла другой базы данных; или нажмите кнопку <Обзор> и в новом диалоговом окне <Создание таблицы> укажите расположение другой базы данных и нажмите кнопку <ОК>. Нажмите кнопку <ОК>, чтобы закрыть первое диалоговое окно <Создание таблицы>, затем – <Выполнить> , после чего подтвердите операцию нажатием кнопки <Да>.

При замене существующей таблицы эта таблица сначала удаляется и перед этим запрашивается подтверждение на удаление. Нажмите кнопку <Да>, а затем – <Да> еще раз для создания новой таблицы (рис. 4).

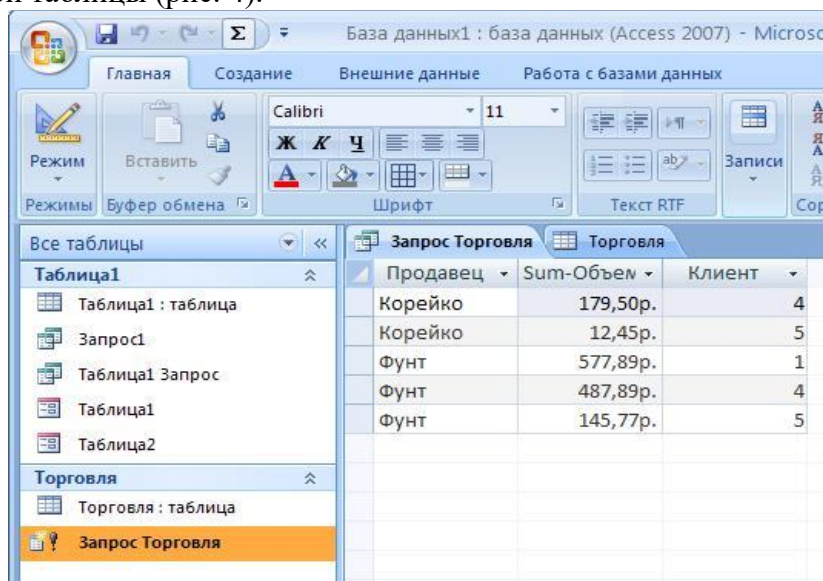


Рисунок 4 - Запрос на создание таблицы Запрос Торговля

3) Запрос на добавление

С помощью запроса на добавление записи одной таблицы (все или отобранные записи) можно присоединить в конец другой таблицы. Эта другая таблица может находиться как в той же базе данных, так и в другой. Например, появилось несколько новых клиентов, а также база данных, содержащая сведения о них. Чтобы не вводить новые данные вручную, их можно добавить в соответствующую таблицу имеющейся базы данных. Можно также использовать запросы на добавление при выполнении следующих действий:

добавление полей на основании условий отбора. Например, необходимо добавить имена и адреса клиентов с очень крупными заказами;

добавление записей, когда некоторых полей одной таблицы не существует в другой. Допустим, в таблице «Заказчики» пользователя имеется 11 полей, тогда как в таблице «Клиенты» другой базы данных из этих 11 полей имеется только 9. Можно использовать запрос на добавление только тех данных, которые находятся в соответствующих друг другу полях, а все остальные игнорировать.

Следует обратить внимание на то, что нельзя изменить данные в отдельных полях имеющихся записей с помощью запросов на добавление. Для этого используются запросы на обновление, а при помощи запросов на добавление можно только добавлять целые строки данных.

Процесс создания запроса на добавление состоит из следующих основных шагов:
создание запроса на выборку;
преобразование запроса на выборку в запрос на добавление;
выбор конечных полей для каждого столбца в запросе на добавление;
выполнение запроса на добавление записей.

Помните, что результаты работы запроса на добавление невозможно отменить, поэтому следует действовать очень аккуратно. Кроме того, следует помнить, что типы данных, заданные для полей в исходной таблице, должны быть совместимыми с типами данных для полей конечной таблицы. Так, например, можно добавить числа в текстовое поле, но нельзя добавить текст в числовое поле, если только не используется выражение вида (Val(ТекстовоеПоле)), чтобы сначала преобразовать данные в тип данных <Числовой>.

Преобразование запроса на выборку в запрос на добавление

На вкладке <Конструктор> в группе <Тип запроса> выберите команду <Добавление>.

Появится диалоговое окно <Добавление> (рис. 5).

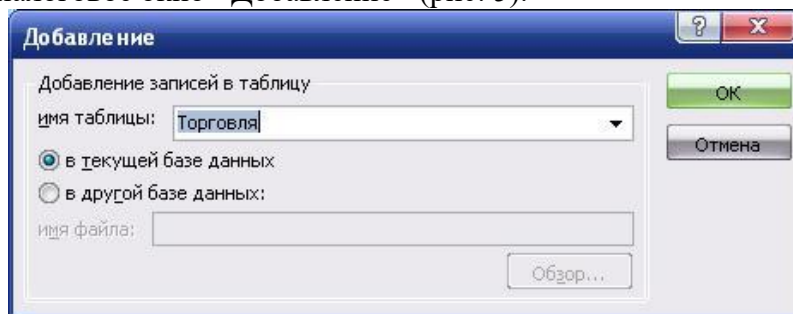


Рисунок 5 - Создание запроса на добавление

Здесь можно выбрать, добавлять ли записи из таблицы в другую таблицу той же базы данных или в таблицу другой базы данных.

Если необходимо скопировать записи в новую таблицу, следует воспользоваться запросом на создание таблицы.

4) Добавление записей в таблицу в той же базе данных

В диалоговом окне <Добавление> выберите пункт <Текущая база данных> (если он еще не выбран), а затем в поле со списком <Имя таблицы> выберите конечную таблицу. Нажмите кнопку <ОК>.

То, что будет отображаться на экране, зависит от способа создания запроса на выборку. Например, если из исходной таблицы или запроса были взяты все поля, эти поля будут добавлены в конечную таблицу к строке <Добавление записей в таблицу> в бланке запроса, или, если в запрос были добавлены отдельные поля, а имена полей в исходной и конечной таблицах совпадают, то имена полей конечной таблицы будут автоматически добавлены в строку запроса <Добавление записей в таблицу>, или, если были добавлены отдельные поля и некоторые или все имена полей в исходной и конечной таблице не совпадают, то поля с

несовпадающими именами будут оставлены пустыми. Можно выбрать ячейку в строке <Добавление записей в таблицу> и выбрать конечное поле (рис. 6).

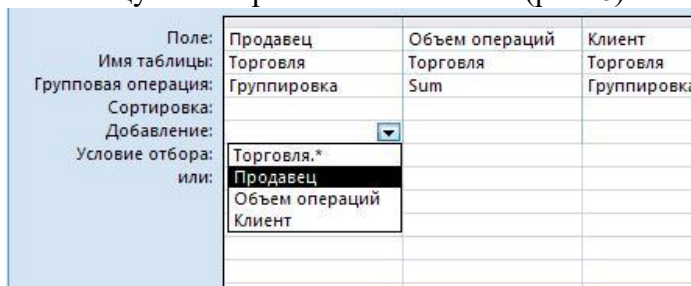


Рисунок 6 - Запрос на добавление в режиме конструктора

Если результирующее поле оставлено пустым, запрос не добавит данных к этому полю.

Для предварительного просмотра внесенных изменений переключитесь в режим таблицы. Для этого щелкните правой кнопкой мыши вкладку документа таблицы и выберите команду <Режим таблицы> (рис. 7).

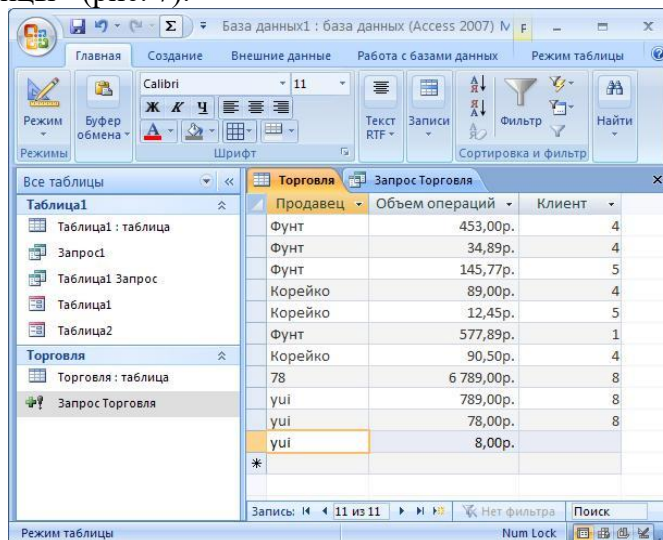


Рисунок 7 - Результат выполнения запроса на добавление «Запрос Торговля»

5) Добавление записей в таблицу другой базы данных

в диалоговом окне <Добавление> выберите параметр в другой базе данных.

поле <Имя файла> введите расположение и имя конечной базы данных.

поле со списком <Имя таблицы> введите имя конечной таблицы, а затем нажмите кнопку <ОК> или нажмите кнопку <Обзор> и с помощью второго диалогового окна <Добавление> найдите конечную базу данных. После определения расположения и выбора конечной базы данных нажмите кнопку <ОК>. Второе диалоговое окно <Добавление> будет закрыто. В первом диалоговом окне <Добавление> в поле <Имя таблицы>, введите имя конечной таблицы, а затем нажмите кнопку <ОК>.

То, что будет отображаться на экране, зависит от способа создания запроса на выборку. Например, если из исходной таблицы или запроса были взяты все поля, эти поля будут добавлены в конечную таблицу к строке <Добавление записей в таблицу> в бланке запроса, или, если в запрос были добавлены отдельные поля, а имена полей в исходной и конечной таблицах совпадают, то имена полей конечной таблицы будут автоматически добавлены в строку запроса <Добавление записей в таблицу>, или если были добавлены отдельные поля и некоторые или все имена полей в исходной и конечной таблице не совпадают, то поля с несовпадающими именами будут оставлены пустыми. Можно щелкнуть ячейку в строке <Добавление записей в таблицу> и выбрать конечное поле. Вернитесь в режим конструктора (щелкните правой кнопкой мыши вкладку документа запроса и выберите команду <Конструктор>) и нажмите кнопку <Выполнить>, чтобы добавить записи.

б) Запросы на удаление данных

Для удаления данных из базы данных Access можно использовать запросы двух типов. Тип используемого запроса зависит от того, что именно требуется удалить.

Если нужно удалить отдельные поля в записях одной или нескольких таблиц, следует использовать запрос на обновление, который заменяет существующие значения либо на пустое значение (то есть на отсутствие данных), либо на пустую строку (пару двойных кавычек без пробела между ними).

Запрос на обновление используется для удаления отдельных полей потому, что запрос на удаление по умолчанию удаляет только строки целиком. Процесс использования запроса на обновление для удаления данных состоит из следующих основных этапов: создание запроса на выборку и проверка, что он возвращает только те данные, которые нужно удалить; преобразование этого запроса в запрос на обновление с использованием значения NULL или пустых строк в качестве нового значения; выполнение этого запроса для обновления существующих данных.

Если требуется удалить записи (строки) целиком, следует использовать запрос на удаление.

По умолчанию запрос на удаление удаляет все данные каждого поля записи вместе со значением ключа, который делает эту запись уникальной. При выполнении такого запроса всегда удаляется строка таблицы целиком. Процесс состоит из следующих основных этапов: создание запроса на выборку и проверка, что он возвращает только те данные, которые нужно удалить; преобразование этого запроса в запрос на удаление и выполнение его для удаления данных.

7) Создание простого запроса на выборку в режиме Конструктор

Предположим, что необходимо сформировать список сотрудников предприятия с указанием их должности, основного оклада и надбавки за стаж работы. Интересующие нас данные находятся в разных таблицах, следовательно, выборку необходимо осуществлять из связанных таблиц. Для выполнения поставленной задачи необходимо выполнить следующие шаги:

Щёлкнуть по пиктограмме «Конструктор запросов». В результате открывается пустое окно Конструктор запросов и диалоговое окно «Добавление таблицы».

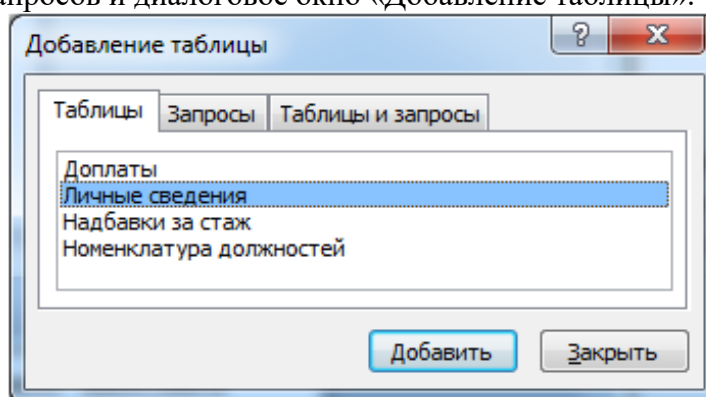


Рисунок 8 - Диалоговое окно для добавления необходимых таблиц на поле запросов

Выберите название необходимой таблицы и нажмите на кнопку **Добавить**, если требуется несколько таблиц для выборки данных, то повторите эту операцию (для выполнения поставленной задачи, понадобятся таблицы «Личные сведения», «Надбавки за стаж» и «Номенклатура должностей»). По окончании переноса таблиц нажмите на кнопку **Закреть**. Обратите внимание, на поле **Запрос1** отобразятся все интересующие нас таблицы, как показано на рисунке 1. В нижней части окна находится бланк запроса, служащий для определения параметров запроса.





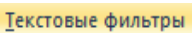
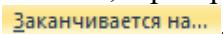
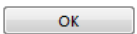
Заполните бланк запроса, для этого в бланке запроса необходимо указать наименование поля таблицы, из которой это поле выбирают. Подведите указатель мыши к необходимому названию поля в таблице, и дважды щёлкните левой кнопкой мыши. Можно выбрать имя необходимого поля из списка, раскрыв его с помощью . На рисунке 1 показан раскрывающийся список доступных полей из всех таблиц. В списке находятся имя таблицы и имя поля, после того, как будет выбрана необходимая строка, в бланке запроса появится в первой строке имя поля, а во второй – имя таблицы. На бланке запроса в строке под наименованием «Вывод на экран» установлен знак , в отдельных случаях можно снять эту пометку, тогда данные из этого поля будут обрабатываться в запросе, но на экране мы их не увидим.

Таблица с результатами по сформированному запросу на выборку данных содержит записи всех сотрудников организации. Для пользователя неудобно просматривать данные, если они никак не упорядочены. Это следует иметь в виду при формировании запроса. В бланке запроса использовать строку с наименованием «Сортировка». Например, можно выстроить список сотрудников по надбавкам за стаж от минимальной величины до максимального значения. Для этого в столбце с полем «Надбавка» раскрыть список символом , а затем выбрать строку с командой  [Сортировка от минимального к максимальному](#). Пользователь, в свою очередь, запустив запрос, может провести упорядочивание данных внутри таблицы с результатами запросов или воспользоваться фильтрацией данных.

Например, пользователя интересуют только женщины, которые получают надбавку за стаж, тогда при использовании фильтра, следует выполнить следующие действия:

Раскрыть в поле «Фамилии» список символом .

Выбрать строку , и раскрыть её.

Выбрать строку с наименованием , нажать на кнопку .

В диалоговое окно «Настраиваемый фильтр» ввести «ова» (Рисунок 9).

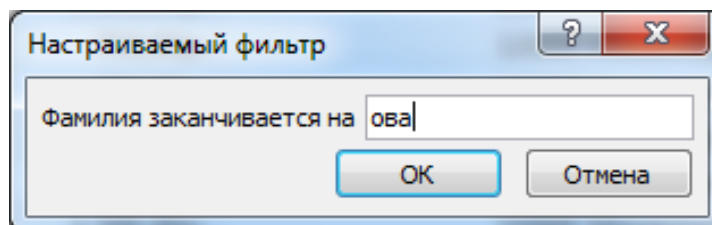



Рисунок 9 - Пример использования текстового фильтра в таблице запроса

Для того, чтобы восстановить таблицу запроса, нажмите на символ  - Удалить фильтр или работайте правой кнопкой мыши в таблице запроса.

8) Создание параметрических запросов

Под параметрическим запросом следует понимать – отбор значений данных из таблиц по заданному параметру. Фактически, вводится условие на поиск информации в однотипных строках таблиц. Например, канцелярия организации должна пригласить на совещание сотрудников определённых должностей и обязательно всех заведующих отделами, для этого необходимо задавать параметры отбора данных. Для формирования запроса, необходимо продумать наименования таблиц, в которых находятся, интересующие нас сведения, а затем запустить Конструктор запросов. В запросе должны быть отражены фамилии сотрудников, место их работы, телефон и фотография. Следовательно, таблицами для отбора данных будут: «Личные сведения» и «Номенклатура должностей», а результатом выполнения запроса будет таблица, например, «Телефоны и должность». На рисунке 10 представлено поле для формирования запроса.

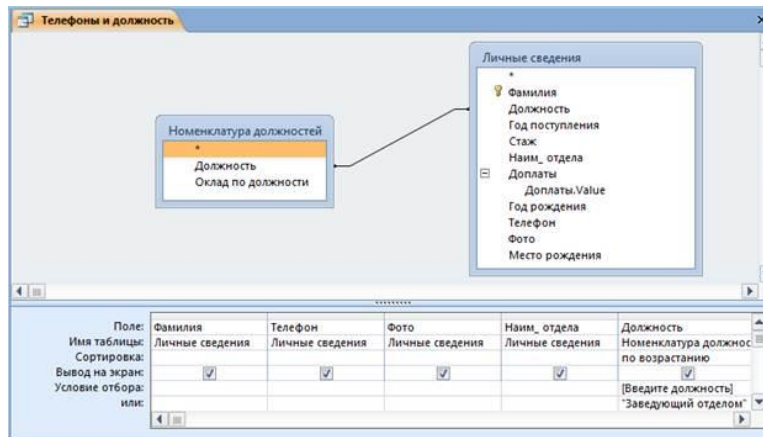


Рисунок 10 - Пример создания параметрического запроса

Поместим таблицы на поле запроса.

На бланк запроса перенесём поля из таблиц (Фамилия, Телефон, Фото, Наим_отдела, Должность).

В строке «Условия отбора» запишем в квадратных скобках текст сообщения [Введите должность], которое будет появляться перед пользователем.

В строке «Или» запишем параметр в двойных кавычках «Заведующий отделом».

Сохраним запрос под именем «Телефоны и должность», а затем запустим этот запрос на исполнение.

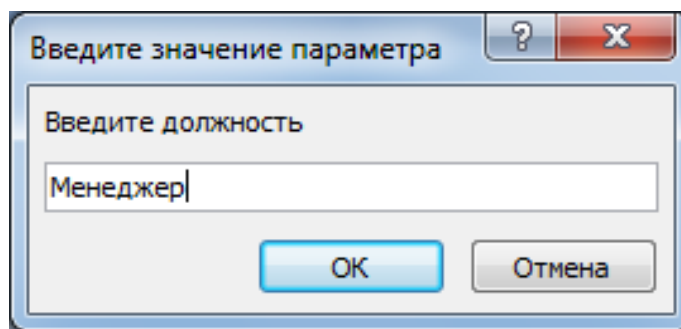


Рисунок 11 - Предложение системы для ввода параметра

В результате будет сформирована таблица с данными (Рисунок 5), отобранными из двух таблиц. Если этот запрос закрыть, то результаты не сохранятся, но при последующих запусках запроса из окна переходов, система будет выдавать диалоговое окно для ввода той должности, которая интересует пользователя.

Фамилия	Телефон	Фотография сотрудника	Наим_отдела	Должность
Иванцов	Код города: (8)-901-34-99	Документ Microsoft Word	Отдел внедрений	Заведующий отделом
Иваницкий	Код города: (79)-035-63-4	Microsoft PowerPoint 97-2003	Отдел внедрений	Менеджер
Шувалова	Код города: (8)-567-23-72	Документ Microsoft Word	Отдел разработок	Менеджер
Петров	Код города: (8)-903-56-28		Отдел внедрений	Менеджер

Рисунок 12 - Результат выполнения отбора данных по заданным параметрам

После выполнения запроса откройте этот запрос в режиме Конструктор, и вы увидите, что в строке с наименованием «Или» бланка запроса ничего нет. Не удивляйтесь, система самостоятельно формирует выражение и записывает его в строку «Условие отбора». Попробуйте мышкой расширить в бланке запроса столбец с наименованием «Должность». В строке появилось логическое выражение, которое показано на рисунке 13.

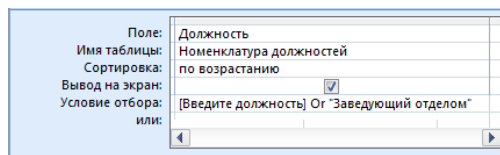


Рисунок 13 - Логическое выражение Or (Или) для заданных параметров отбора данных

Следует отметить, что параметры для отбора данных могут быть заданы в виде текста или числа, с помощью диалогового окна или выражения. При вводе жёсткого параметра в виде текста, например, можно указать должность, тогда текстовый параметр заключается в двойные кавычки («...»). При вводе жёсткого параметра в виде числа, например год рождения, двойные кавычки не используют. При использовании диалогового окна, признаком его открытия являются открытая и закрытая квадратные скобки ([...]), параметр, вводимый в диалоговое окно может быть и текстом и числом. При построении логического выражения в качестве параметра, следует помнить, что сравниваются только данные, принадлежащие одному столбцу бланка запроса.

9) Формирование запроса по неполному значению поля

Обращаясь к базе данных, может предполагать лишь приблизительное написание текстовых значений, например, фамилии, или предполагать диапазон численных данных, которые его интересуют. В этом случае создают запрос на выборку по неполному значению поля. Предположим, что руководитель организации хочет получить сведения о сотрудниках организации, которые поступили на работу между 2012 и 2007 годами, а их фамилии содержат сочетание букв – «ва». В этом случае целесообразно составить поисковое предписание, использующее стандартные функции “Like” – Подобно и “Between” – Между. Для решения поставленной задачи следует выполнить следующие действия:

Создать запрос в режиме Конструктора, используя в качестве источника таблицу «Личные сведения».

Поместить в бланк запроса поля «Фамилия», «Должность», «Год поступления», «Наим_отдела» (можете использовать и другие поля).

В строку «Условия отбора» для поля «Фамилия» ввести условие: Like “*ва*”. Звёздочка слева и справа заменяет неизвестный фрагмент текста (можно поставить звёздочку только с одной стороны).

В строку «Условия отбора» для поля «Год поступления» ввести условие: Between 2007 2012, как показано на рисунке 14.

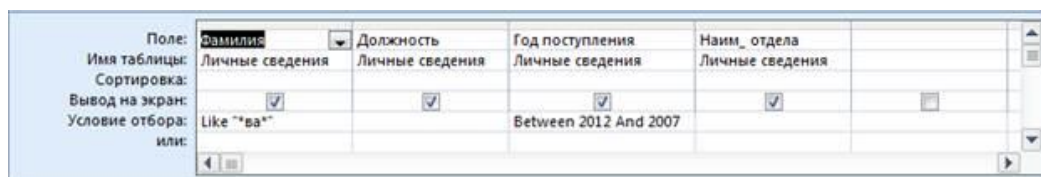


Рисунок 14 - Заполнение бланка запроса для осуществления поиска по неполному значению поля

10) Обработка множественных данных

Предположим, что планово-экономическому отделу необходимо разработать фонд заработной платы сотрудников организации, при этом, требуется знать величину надбавки для каждого сотрудника, его ставку (оклад по должности) и суммарный коэффициент, на который увеличивается оклад сотрудника. Используя возможности Access, эта процедура не вызывает затруднений. Для начала необходимо продумать, в каких исходных таблицах можно найти необходимые значения данных, а затем приступить к формированию запроса.

Создайте новый запрос в режиме Конструктора. Перенесите на поле конструктора таблицы, как показано на рисунке 15.

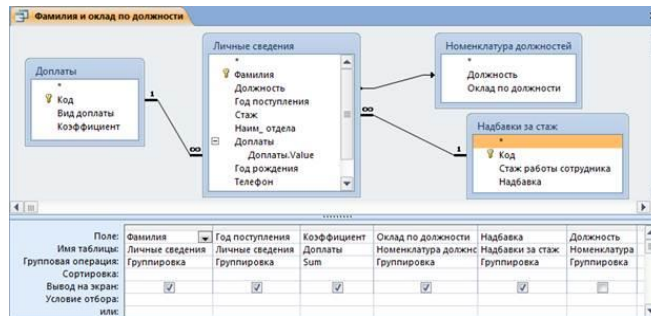


Рисунок 15 - Подготовка запроса для работы с множественным значением данных

В бланке запроса добавьте новую строку с наименованием «Групповая операция». Для этого можно щёлкнуть правой кнопкой мыши по бланку запроса, а затем выбрать строку **Σ Итоги** или на ленте активизировать элемент **Σ** - итоги (на вкладке Работа с запросами). Раскройте список выбора функций при работе с групповыми данными в столбце для таблицы «Доплаты» (Рисунок 16), и выберите функцию Sum.

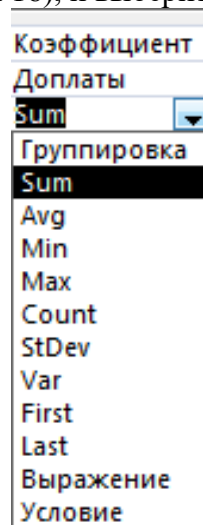


Рисунок 16 - Список для выбора функций при работе с групповыми данными

Сохраните запрос, например под именем «Фамилия и список выплат», запустите запрос на выполнение, результаты показаны на рисунке 17.

Личные сведения	Личные сведения	Sum-Кoeff	Номенклатура	Надбавки за стаж
Аристова	2006	0,15	18000	1 800,00р.
Верховин	2009	0,15	13500	750,00р.
Иваницкий	2012	0	15000	0,00р.
Иванов	1998	0,55	18000	1 800,00р.
Иванцов	2008	0,7	25000	1 200,00р.
Костиков	2010	0,15	13500	1 200,00р.
Новикова	2011	0	14000	750,00р.
Петров	2005	0,25	15000	1 800,00р.
Серафимова	2009	0	13500	750,00р.
Степанов	2012	0,25	14000	0,00р.
Шувалова	2007	0,7	15000	1 200,00р.

Рисунок 17 - Результаты работы запроса с обработкой множественных данных

В бланке запроса строка с наименованием **Групповая операция: Группировка** относится ко всем полям, следовательно, кроме выполнения операции суммирования данных, можно задавать и другие условия отбора данных в запрос, например, выражения.

Технология использования

Построителя выражений для выполнения вычислений

Выполнение вычислений над данными, которые находятся в разных полях таблицы или в различных таблицах и запросах приходится очень часто. Эффективным средством для составления формул по определённым алгоритмам, является надстройка в Access - Построитель выражений, которая подключается в режиме Конструктора с помощью

пиктограммы . В диалоговом окне Построитель выражений содержатся два раздела.

Раздел в верхней части окна содержит поле, в котором создаётся выражение.

Раздел в нижней части окна предназначен для создания элементов выражений и вставки их в поле выражения. Допускается непосредственный ввод выражения с клавиатуры. Этот раздел разделён на три вертикальных поля. В левом поле (Элементы выражений) выводятся папки, содержащие объекты базы данных - , встроенные и определённые пользователем функции - , константы - , операторы - и выражения - . Среднее поле (Категории выражений) служит для выбора элемента или типа элементов из папки, заданной в левом поле. В правом поле (Значения выражений) выводится список значений для элементов, заданных в левом и среднем полях. Например, на рисунке 10 в построителе выражений показано, как отображается информация в полях Построителя выражений.

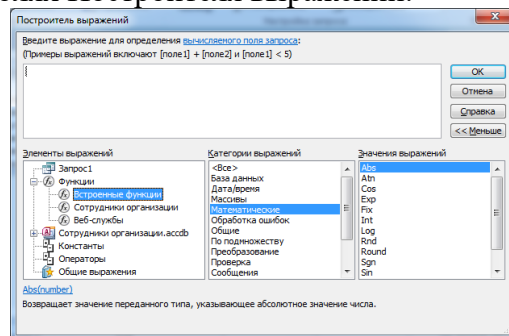


Рисунок 18 - Общий вид построителя выражений

Создание выражения для проведения вычислений в таблице

Предположим, отдел кадров ежегодно готовит приказ об изменении надбавок за стаж сотрудникам организации. Очевидно, чтобы не проводить сложных поисков, требуется всего лишь выяснить, сколько сотрудник проработал в организации. Для создания такого запроса, потребуются сведения из таблицы «Личные сведения».

На поле конструктора поместите таблицу «Личные сведения»

В бланк запроса перенесите последовательно поля: «Фамилия», «Год поступления», «Должность».

В пустом поле щёлкните мышкой, и на ленте щёлкните по пиктограмме , после чего откроется Построитель выражений.

В верхнем поле Построителя введите выражение - Набрал стаж:2013-, в поле «Элементы выражений» раскройте объект «Таблицы» и щёлкните по значку , в поле «Категории выражений» дважды щёлкните по - , после чего, верхнем поле Построителя будет сформирована запись -Набрал стаж:2013- [Личные сведения]![Год поступления].

В бланке запроса отобразится, создаваемое выражение (Рисунке 19), сохраните запрос, например под именем «Фактический стаж работы».

Поле:		Год поступления	Должность	Набрал стаж: 2013-[Личные сведения]![Год поступления]
Имя таблицы:	Личные сведения	Личные сведения	Личные сведения	по возрастанию
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				
или:				

Рисунок 19 - Бланк запроса для вычисления фактического стажа работы сотрудника

Получив результаты выполнения запроса (рисунок 20), можно ввести логические переменные на предмет анализа, кто из сотрудников должен получить в следующем году другую надбавку, в связи с переходом в другую категорию, напомним, что градации надбавок находятся в таблице «Надбавка за стаж». Для удобства проведения визуального анализа результаты, которые отображаются в поле «Набрал стаж» отсортированы по возрастанию.

Фамилия	Год поступления	Должность	Набрал стаж
Иванчицкий	2012	Менеджер	1
Степанов	2012	Бухгалтер	1
Новикова	2011	Бухгалтер	2
Костилов	2010	Инженер проекта	3
Серафимова	2009	Инженер проекта	4
Верховин	2009	Инженер проекта	4
Иванцов	2008	Заведующий отде.	5
Шувалова	2007	Менеджер	6
Аристова	2006	Старший инженер	7
Петров	2005	Менеджер	8
Иванов	1998	Старший инженер	15

Рисунок 20 - Результаты вычислений фактически отработанных лет сотрудниками

11) Перекрестные запросы

Если необходимо объединить данные в формате строк-колонок (двумерная таблица), то следует изготовить перекрестный запрос. При проектировании перекрестного запроса можно указать в качестве заголовков для колонок значения некоторых полей или выражений.

Перекрестные запросы позволяют более компактно, чем обычные запросы, индексировать данные, объединяя однотипную информацию. Рассмотрим создание перекрестного запроса «Продавец-Покупатель» на базе таблицы «Торговля», используя в качестве вспомогательной таблицу «Адреса» для выбора фамилий покупателей. Для создания перекрестного запроса в режиме проектирования следует сформировать запрос, добавив таблицу, которая должна лежать в его основе. В данном случае это таблицы «Торговля» и «Адреса». В меню <Тип запроса> активизируйте команду <Запрос: перекрестный>. Access отобразит в зоне спецификации запроса строки <Групповая операция> и <Перекрестная таблица>. Затем включите в проект запроса необходимые поля <Продавец>, <Объем операций таблицы Торговля> и поле <Покупатель из таблицы Адреса> (рис. 21).

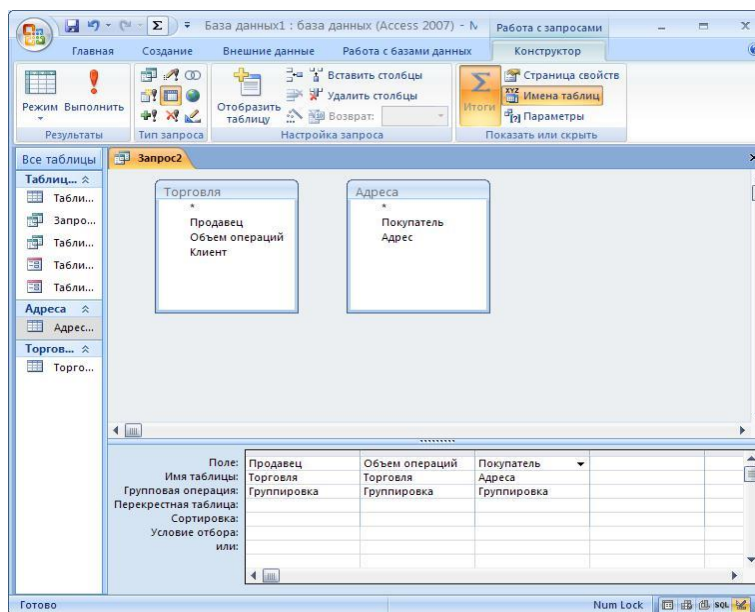


Рисунок 21 - Перекрестный запрос в режиме конструктора

Чтобы значения поля (например, <Продавец>) появились в заголовках строк, выполните щелчок на строке <Перекрестная таблица> и из разворачивающегося списка выберите строку <Заголовок строк>. В строке <Групповая операция> для этого поля должна сохраниться стандартная установка <Группировка>.

Чтобы значения поля появились в качестве заголовка колонки (<Фамилия>), выполните щелчок мышью на строке <Перекрестная таблица> и выберите строку <Заголовок столбцов>. В этом случае в строке <Групповая операция> также должна остаться стандартная установка <Группировка>.

Чтобы содержимое поля индексировалось в кросстаблице в качестве значения (<Объем операций>), щелкните на строке <Перекрестная таблица> и выберите строку <Значение>. В

строке <Групповая операция> для этого поля введите необходимую функцию, например, подсчета суммы (<Sum>), определения среднего значения (<Avg>) или количества (<Count>). В нашем случае для получения объема продаж каждого продавца больше всего подходит функция <Sum>.

При определении критериев отбора в строке <Групповая операция> для желаемого поля следует выбрать значение <Условие> и ввести выражение в строку <Условие отбора>. Для этого поля оставьте пустой строку <Перекрестная таблица>. Критерий отбора мы вводить не будем (рис. 22).

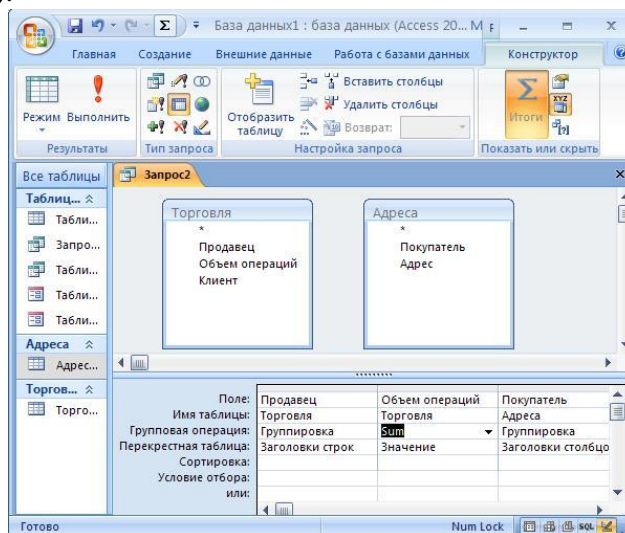


Рисунок 22 - Перекрестный запрос «Запрос2» в режиме конструктора

Создание запроса с помощью Мастера перекрестного запроса

Помимо рассмотренного способа создания перекрестного запроса «вручную», с помощью конструктора запроса в арсенале Access есть и другое средство, позволяющее добиться той же цели – <Мастер запросов> (вкладка <Создание> – панель <Другие> – кнопка <Мастер запросов>) (рис. 23).

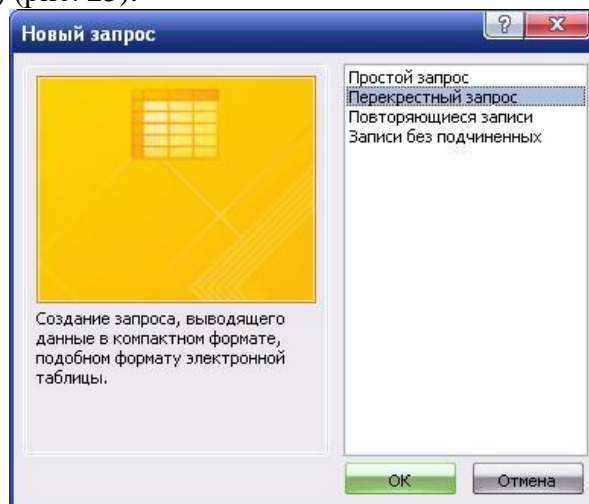


Рисунок 23 - Создание перекрестного запроса с помощью «Мастера запросов»

В первом диалоговом окне мастера для создания перекрестного запроса следует выбрать таблицу или запрос, если для создания запроса используется не одна таблица. Поскольку в данном примере используются как раз две таблицы: «Торговля» и «Адреса», - сначала следует создать запрос, в котором использовались бы поля обеих таблиц, необходимые для перекрестного запроса.

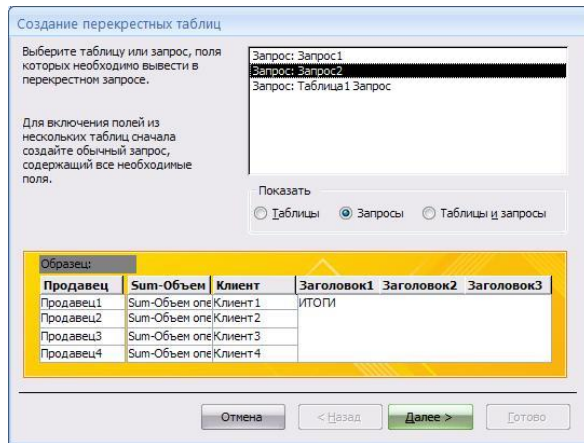


Рисунок 24 - Выбор запроса-источника данных для перекрестного запроса

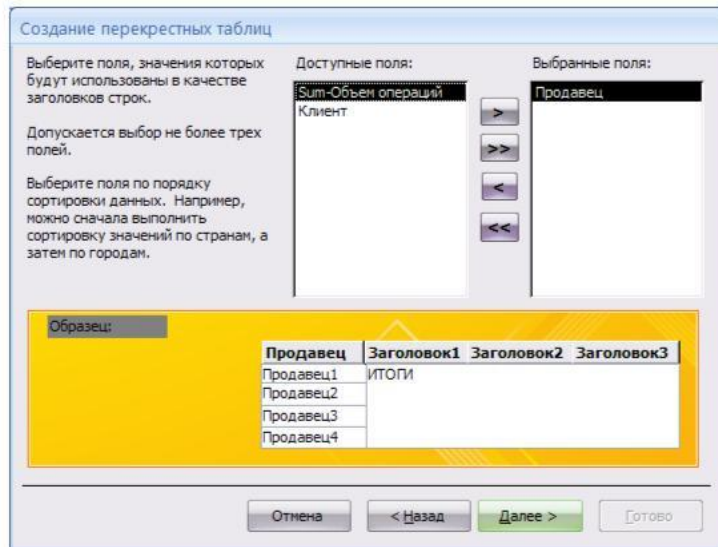


Рисунок 25 - Выбор полей, значения которых будут использованы в качестве заголовков строк

В этом окне следует выбрать поле, значения которого будут фигурировать в качестве заголовков колонок таблицы перекрестного запроса. Выберем поле <Клиент> (рис. 26).

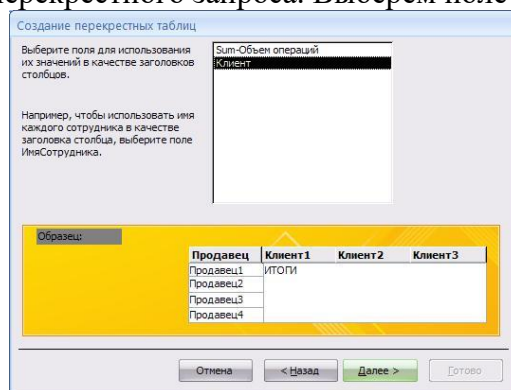


Рисунок 26 - Выбор полей, значения которых будут использованы в качестве заголовков столбцов

В данном окне мастера нам необходимо определить поле, значения которого будут индексироваться в клетках таблицы, и выбрать операцию для обработки этих значений. В нашем случае это операция суммирования (<Sum>). Кроме того, в перекрестной таблице можно произвести суммирование по каждой строке (т. е. по каждому продавцу). Для этого необходимо установить опцию (<Да, вычислить итоговое значение для каждой строки>). В

результате в перекрестный запрос добавится колонка с суммами продаж каждого продавца (рис. 27).

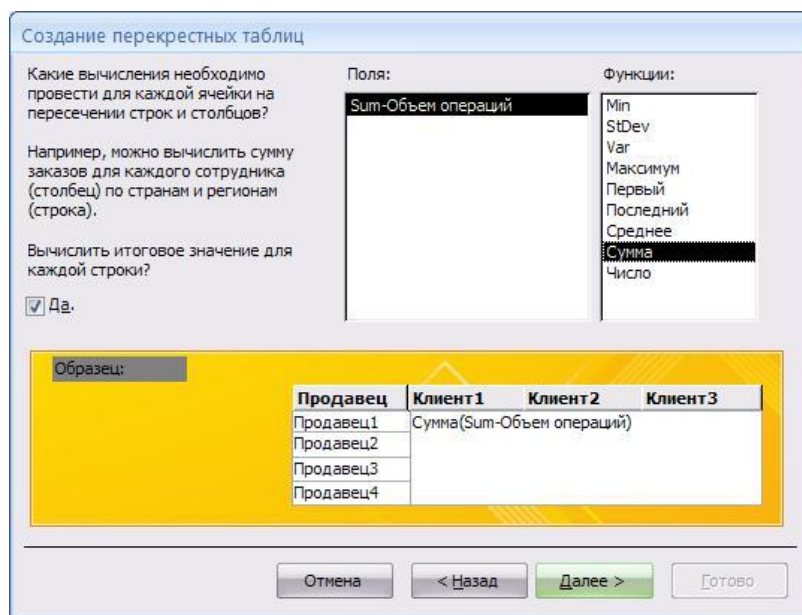


Рисунок 27 - Выбор необходимых функций для полей перекрестного запроса

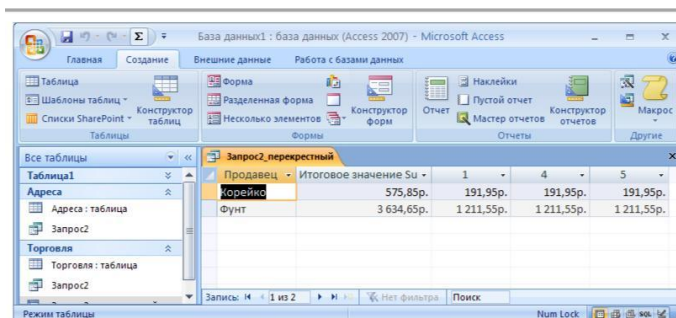


Рисунок 28 - Результат выполнения перекрестного запроса «Запрос2_перекрестный»

Отчет должен содержать:

1. Название, цель и задание лабораторной работы.
2. Иллюстрации конструктора таблиц с масками или форматами полей и результата выполнения;
3. Ответы на контрольные вопросы.

Контрольные вопросы:

1. Как отображаются результаты запроса?
2. Какие основные типы запросов создают в базе данных Access?
3. Для чего предназначены запросы на добавление?
4. В каком порядке следует работать с Конструктором запросов?
5. Какие дополнительные возможности получает пользователь при просмотре запроса на выборку?
6. Какая последовательность запуска Построителя выражений при составлении запросов?
7. Как использовать поле «Элементы выражений» для создания выражения?
8. Как связывается бланк запроса с Построителем выражений?
9. Из каких элементов состоит «Выражение»?

Лабораторная работа № 7
Тема: «Создание отчетов, сводных таблиц и диаграмм»

Цель лабораторной работы:

- освоить приёмы создания и оформления простых, составных отчетов, сводных таблиц и диаграмм;
- развитие познавательного интереса;
- изучение новых алгоритмов решения задач;
- формирование универсальных учебных действий, связанных с поиском информации, необходимой для решения поставленной задачи.

Рекомендации по выполнению лабораторных работ: лабораторная работа выполняется группой студентов (2-3 человека) средствами СУБД MS Access.

Количество часов: 2 часа.

Постановка задачи

1. Создать отчет с сортировкой по фамилии.
2. Создать отчет с сортировкой по нескольким столбцам.
3. Создать отчет с группировкой по определенной группе данных.
4. Создать отчет с использованием в нем агрегатных функций.
5. Создать отчет, выполняющий подсчет общей суммы по определенному столбцу.
6. Создать сводную таблицу данных с вычислением автосуммы.
7. Создать сводную диаграмму.

Пример выполнения работы

1. Отчеты

Программа Microsoft Office Access помогает создавать множество различных отчетов любой степени сложности. Прежде чем приступать к работе, следует выбрать источник, из которого будут извлекаться записи отчета. Отчет может представлять собой как простой список, так и подробную сводку данных о продажах, сгруппированных по районам. Од-нако в любом случае необходимо сначала определить, в каких полях содержатся данные, которые должны войти в отчет, и в каких таблицах или запросах находятся эти поля.

После того как выбран источник записей для создания отчета, проще всего воспользоваться мастером отчетов. Мастер отчетов – это средство Microsoft Access, помогающее создать отчет на основании ответов, полученных на заданные пользователю вопросы.

2. Выбор источника записей

Наряду с данными, извлекаемыми из таблиц и запросов, в отчете содержится информация о макете отчета: подписях, заголовках, рисунках и т. д. Таблицы и запросы, содержащие базовые данные, называются источником записей отчета. Если все поля, которые нужно включить в отчет, находятся в одной таблице, эта таблица и будет источником записей. Если поля находятся в нескольких таблицах, в качестве источника записей придется использовать один или несколько запросов. Иногда эти запросы уже существуют в базе данных, но может по надобиться сформировать новые запросы специально для данного отчета.

Создание отчета с помощью средства «Отчет»

Средство «Отчет» – это самый быстрый способ создания отчета, потому что с его помощью отчет формируется сразу же, без запроса дополнительной информации. В отчете будут представлены все записи базовой таблицы или запроса. Хотя получившийся отчет, скорее всего, будет далек от совершенства, он позволит быстро просмотреть базовые данные. После этого отчет можно сохранить и изменить в режиме макета или конструктора в соответствии с поставленной задачей. В области переходов щелкните таблицу или запрос, на основе которых нужно создать отчет (рис.1)

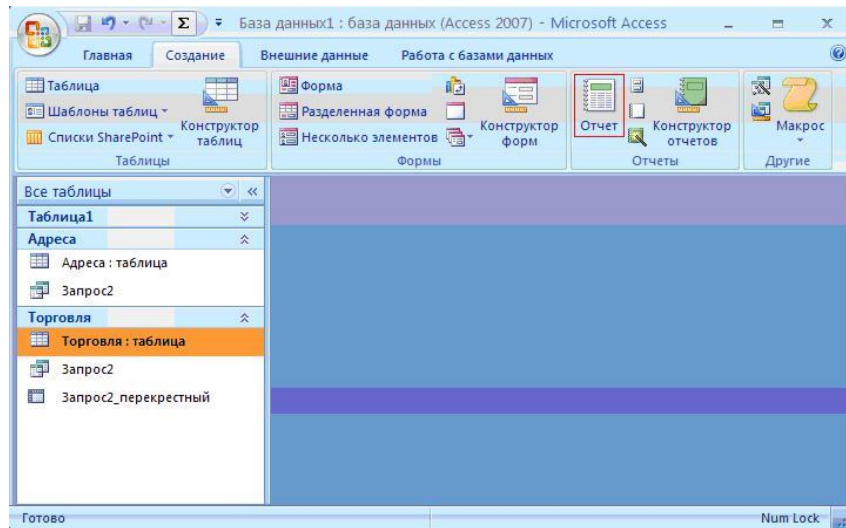



Рисунок 1 - Создание отчета

На вкладке <Создание> в группе <Отчеты> щелкните <Отчет> .

В области переходов выберите таблицу или запрос, на основе которых нужно создать отчет.

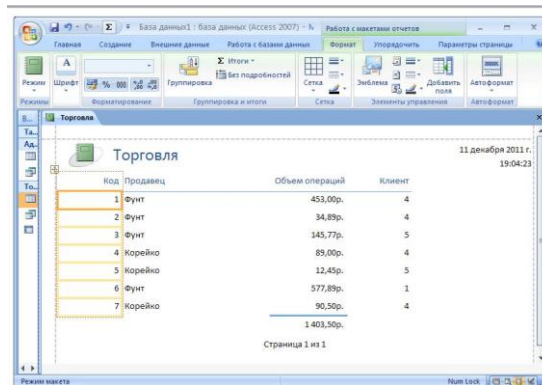


Рисунок 2 - Отчет Торговля в режиме макета

3. Создание отчета с помощью Мастера отчетов

В <Мастере отчетов> предоставляется больше возможностей относительно выбора полей для включения в отчет. При этом можно указать способ группировки и сортировки данных, а также включить в отчет поля из нескольких таблиц или запросов, если отношения между этими таблицами и запросами заданы заранее.

На вкладке <Создание> в группе <Отчеты> щелкните <Мастер отчетов>.

В первом окне предлагается выбрать таблицу или запрос и поля для отчета

В следующем окне можно назначить группировку полей (рис. 3).

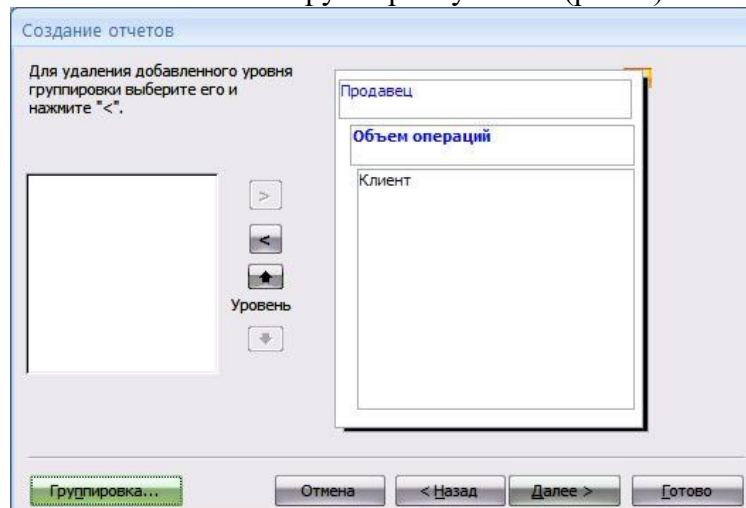


Рисунок 3 - Группировка полей отчета

Также, нажав на кнопку <Группировка>, можно задать необходимые интервалы группировки (рис. 4).

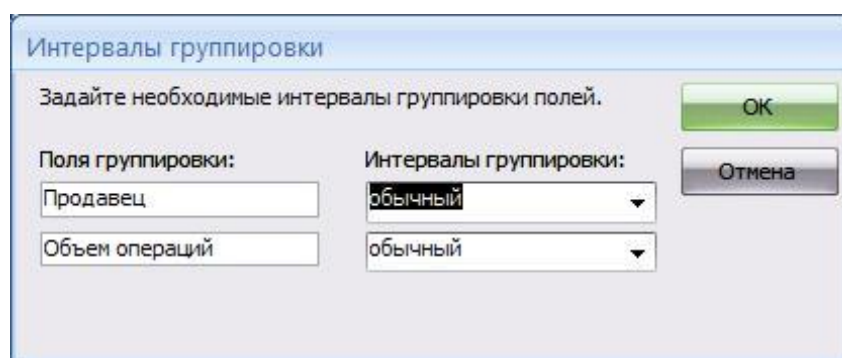


Рисунок 3 - Интервалы группировки

В следующем окне задается порядок сортировки, далее предлагается выбрать вид макета для отчета и установить его положение на странице, в следующем окне выбирается стиль отчета.

Далее предлагается открыть отчет в том виде, какой мы создали, либо изменить макет отчета. В первом варианте появится окно предварительного просмотра, которое позволяет увидеть (рис. 5).

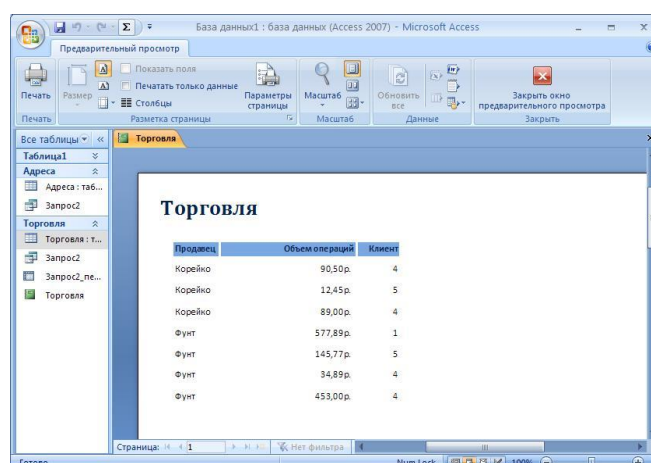



Рисунок 5 - Отчет Торговля

Создание отчета с помощью средства «Пустой отчет»

Если по каким-либо причинам для построения отчета не подходят ни средство <Отчет>, ни <Мастер отчетов>, можно создать отчет с нуля при помощи средства <Пустой отчет>. Это очень быстрый способ, особенно если в отчете должно содержаться небольшое количество полей. На вкладке <Создание> в группе <Отчеты> щелкните <Пустой отчет> .

В режиме макета появится пустой отчет, а в правой части окна Access – область <Список полей> (окно, содержащее перечень всех полей в базовом источнике записей объекта базы данных, за исключением страницы доступа данным в режиме конструктора; в режиме конструктора страницы доступа к данным в списке полей содержится перечень всех источников записей и всех полей в подключенной базе данных) (рис. 6).

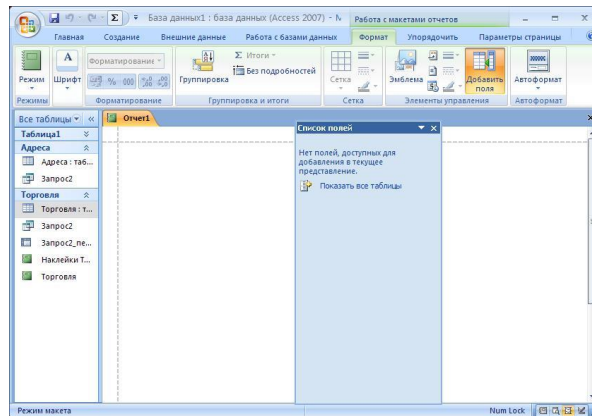


Рисунок 6 - Пустой отчет с областью <Список полей>

В области <Список полей> щелкните знак «+» рядом каждой из таблиц, содержащих поля, которые необходимо включить в отчет.

По очереди перетащите каждое из полей в отчет либо, нажав и удерживая клавишу <CTRL>, выберите несколько полей и перетащите их одновременно (рис. 7).

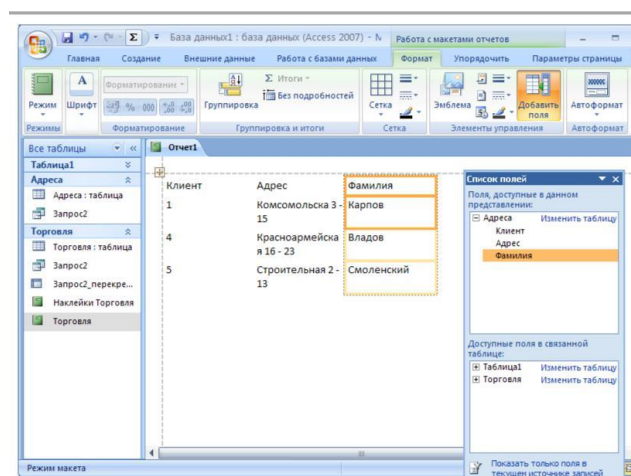


Рисунок 7 - Добавление необходимых полей для отчета перетаскиванием

С помощью инструментов, представленных в группе <Элементы управления> на вкладке <Форматирование>, можно добавить в отчет эмблему компании, заголовок, номера страниц, дату и время (рис. 8).

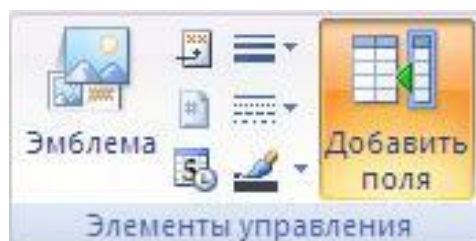


Рисунок 8 - Элементы управления для отчетов

4. Разделы отчета

В программе Access макет отчета разбит на разделы. Разделы можно просматривать в режиме конструктора. Чтобы правильно создавать отчеты, необходимо понимать назначение каждого раздела. Например, от выбора раздела, в который будет помещен вычисляемый элемент управления, зависит способ вычисления результата.

Заголовок отчета печатается только один раз в начале отчета. В заголовок включается информация, обычно помещаемая на обложке, например эмблема компании, название отчета или дата. Если в заголовке отчета помещен вычисляемый элемент управления,

использующий статистическую функцию <Sum>, сумма рассчитывается для всего отчета. Заголовок отчета печатается перед верхним колонтитулом.

Верхний колонтитул печатается вверху каждой страницы. Верхний колонтитул используется, например, в тех случаях, когда нужно, чтобы название отчета повторялось на каждой странице.

Заголовок группы печатается перед каждой новой группой записей. Используется для печати названия группы. Например, если отчет сгруппирован по изделиям, в заголовках групп можно указать их названия. Если поместить в заголовок группы вычисляемый элемент управления, использующий статистическую функцию <Sum>, сумма будет рассчитываться для текущей группы.

Область данных печатается один раз для каждой строки данных из источника записей. В нем размещаются элементы управления, составляющие основное содержание отчета.

Примечание группы печатается в конце каждой группы записей. Примечание группы можно использовать для печати сводной информации по группе.

Нижний колонтитул печатается внизу каждой страницы. Используется для нумерации страниц и для печати постраничной информации.

Примечание отчета печатается один раз в конце отчета. Примечание отчета можно использовать для печати итогов и другой сводной информации по всему отчету.

В конструкторе <Примечание отчета> находится под нижним колонтитулом. Однако при печати и в режиме предварительного просмотра <Примечание отчета> помещается над нижним колонтитулом, сразу после <Примечания группы> для последней группы или строки подробностей последней страницы.

Элементы управления

Элементы управления - это объекты, позволяющие отображать и просматривать данные, производить с ними различные действия, а также работать с дополнительными элементами пользовательского интерфейса, такими как подписи и рисунки. В программе Access поддерживаются элементы управления трех типов: присоединенные, свободные и вычисляемые.

Присоединенные элементы управления - это элементы управления, для которых источником данных служит поле таблицы или запроса. Они используются для отображения значений из полей базы данных. Это могут быть значения даты или времени, текстовые, числовые, логические, а также рисунки и диаграммы. Типичным примером присоединенного элемента управления является текстовое поле. Например, источником информации для текстового поля формы, в котором указывается фамилия сотрудника, может быть поле <Фамилия> таблицы «Сотрудники».

Свободные элементы управления - это элементы управления, не связанные с источником данных (полем или -выражением). Свободные элементы управления используются для вывода на экран текста, линий, прямоугольников и рисунков. Примером свободного элемента является подпись, в которой отображается заголовок отчета.

Вычисляемые элементы управления - это элементы управления, для которых источником данных является выражение, а не поле. Чтобы задать значение, которое должно содержаться в таком элементе управления, необходимо задать выражение, которое станет для этого элемента источником данных.

Выражение - это сочетание операторов (например, «=» и «+»), имен других элементов управления, имен полей, функций, возвращающих единственное значение, и констант.

Например, в следующем выражении цена изделия рассчитывается с 25%-ной скидкой путем умножения значения поля «Цена за единицу» на константу 0,75: [Цена за единицу] * 0,75.

В выражение можно включать данные из базовой таблицы или базового запроса, а также из другого элемента управления в отчете.

При создании отчета, в котором будут присутствовать элементы управления всех трех типов, рекомендуется вначале разместить в отчете все присоединенные элементы, особенно если они будут преобладать в отчете. Затем можно добавить свободные и вычисляемые

элементы управления, воспользовавшись инструментами группы <Элементы управления> на вкладке <Конструктор>.

Элемент управления присоединяется к полю путем указания поля, из которого этот элемент управления получает данные. Чтобы создать элемент управления, присоединенный к выбранному полю, перетащите поле из области <Список полей> в отчет. В области <Список полей> перечисляются поля базовой таблицы или запроса. Чтобы отобразить область <Список полей>, на вкладке <Конструктор> в группе <Элементы управления> нажмите кнопку <Добавить существующие поля> (рис. 9).

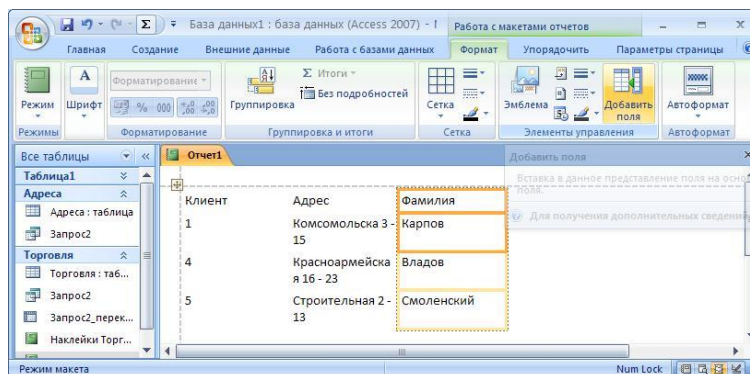


Рисунок 9 - Добавление существующих полей

5. Создание составных отчетов

В Access предусмотрена возможность вставлять в любой отчет другие отчеты, называемые в этом случае подотчетами. С помощью механизма подотчетов пользователь может связать друг с другом два или более отчетов в один главный отчет или вставить итоговые данные подотчетов в главный отчет.

Для связывания подотчета с главным отчетом следует открыть главный отчет в режиме проектирования и перетащить в него из окна базы данных пиктограмму отчета, подлежащего встраиванию в качестве подотчета, разместив ее там в нужной секции. В итоге получим составной отчет.

При установке характеристик связи «вручную» необходимо следить за тем, чтобы связываемые поля имели одинаковый тип или, по крайней мере, одинаковый размер.

6. Создание сводных таблиц и диаграмм

Сводные таблицы и диаграммы предназначены для интерактивного анализа данных различной сложности и объема. Они позволяют произвести сравнительный анализ данных простыми и удобными средствами. Сводные таблицы являются многомерными и это позволяет представлять данные, сгруппированные по различным координатам. Разработка сводной таблицы для таблицы базы данных. Рассмотрим таблицу «Журнал» в режиме сводной таблицы. Чтобы открыть таблицу в этом режиме, выберите её в окне базы данных, нажмите кнопку «Открыть», а

затем для перехода из режима таблицы в режим сводной таблицы выберите на панели Вид Сводная таблица.

При первом открытии в ее рабочую область не включено не одного поля таблицы (рис. 10)

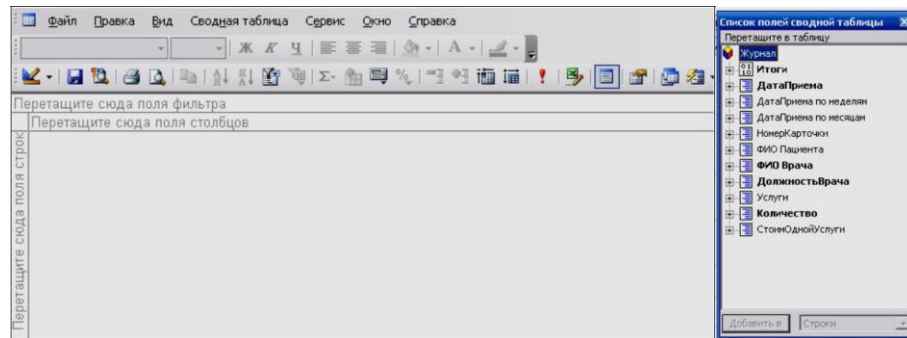


Рисунок 10 - Пустой макет сводной таблицы Журнал и список полей источника записей

Пусть необходимо подсчитать количество выполненных услуг каждым врачом за каждый день (за определенный период). Если список полей этой таблицы не отображается, нажмите на панели инструментов пиктограмму

Уникальные значения фамилии врачей составят заголовки строк. В эту область помещены поля ФИО Врачей. В область полей столбцов поле ДатаПриема. Фильтрация по должностям. В область фильтра необходимо поместить – поле ДолжностьВрача (рис. 11).

ДолжностьВрача	ДатаПриема								Общие итоги
	01.01.2011	02.01.2011	03.01.2011	04.01.2011	06.01.2011	09.01.2011	01.02.2011	09.02.2011	
ФИО Врача	Количество	Количество	Количество	Количество	Количество	Количество	Количество	Количество	Нет итогов
Валиуллин Д.К.	1	2				1			
Галимова О.А.	5							1	
Дюкина Я.В.	4								
Елизарова Е.П.		2				2	2		
Жуков А.Е.			2						
Карбышев Г.О.						3	1		
Леонидова М.Р.			3				2		
Накипов Н.Д.		2							
Никоенко П.С.				2		1			
Раддинова Ф.Я.			1						
Сидорович М.Л.				4			1		
Халимов И.Ш.				1					
Харитонов Н.Н.				2		4			1
Шабалинова Ф.Н.									2
Общие итоги									

Рисунок 11 - Сводная таблица с фамилиями врачей и датами приема

Для подсчета итоговых значений в контекстном меню в области данных необходимо выбрать Автовычисления Сумма. В результате получим таблицу с итоговыми значениями (количество выполненных услуг) и суммарной величиной этих значений. (рис. 12).

ДолжностьВрача	ДатаПриема								Общие итоги
	01.01.2011	02.01.2011	03.01.2011	04.01.2011	06.01.2011	09.01.2011	01.02.2011	09.02.2011	
ФИО Врача	Количество	Количество	Количество	Количество	Количество	Количество	Количество	Количество	Сумма "Количество"
Валиуллин Д.К.	1	2				1			4
Галимова О.А.	5							1	6
Дюкина Я.В.	4								4
Елизарова Е.П.		2				2	2		4
Жуков А.Е.			2						4
Карбышев Г.О.						3	1		6
Леонидова М.Р.			3				2		5
Накипов Н.Д.		2							3
Никоенко П.С.				2		1			3
Раддинова Ф.Я.			1						1
Сидорович М.Л.				4			1		8

Рисунок 12 - Сводная таблица с вычисляемыми полями.

Также можно получить сводную диаграмму по таблице, выбрав на панели инструментов Вид Сводная диаграмма. (рис. 13).

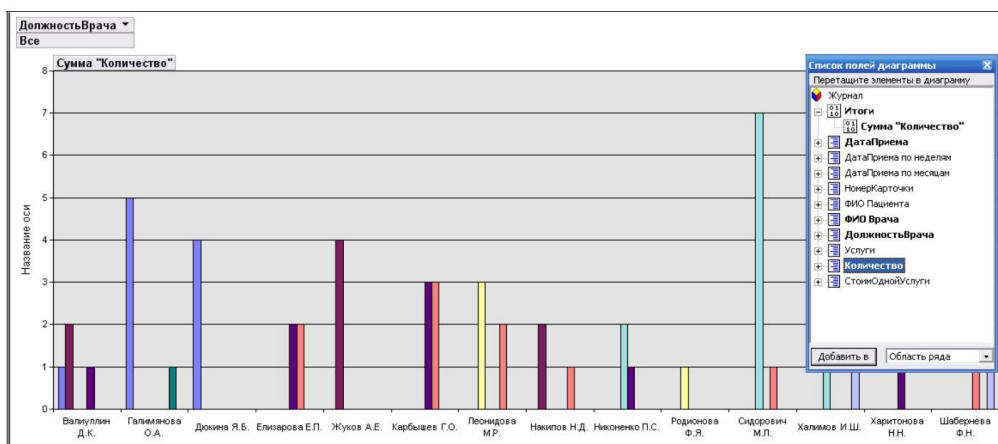


Рисунок 13 - Таблица Журнал в режиме сводной диаграммы.

Отчет должен содержать:

1. Название, цель и задание лабораторной работы.
2. Иллюстрации конструктора таблиц с масками или форматами полей и результата выполнения;
3. Ответы на контрольные вопросы.

Контрольные вопросы:

1. Для чего нужен отчет?
2. Какие способы создания отчетов вы знаете?
3. Виды элементов управления.
4. Каким образом создается составной отчет?
5. Способы создания параметрического отчета?
6. Назначение сводных таблиц.
7. Способ создания сводных диаграмм.

Лабораторная работа № 8
Тема: «Создание отчетов, сводных таблиц и диаграмм»

Цель лабораторной работы:

- изучить принципы использования макрокоманд в MS Access;
- развитие познавательного интереса;
- изучение новых алгоритмов решения задач;
- формирование универсальных учебных действий, связанных с поиском информации, необходимой для решения поставленной задачи.

Рекомендации по выполнению лабораторных работ: лабораторная работа выполняется группой студентов (2-3 человека) средствами СУБД MS Access.

Количество часов: 2 часа.

Постановка задачи

1. Создать макрос для перевода отчета в MS Word.
2. Создать макрос для перевода запроса в MS Excel.
3. Создать макрос для окончания сеанса работы с базой.

Пример выполнения работы

Макросы великолепно подходят для автоматизации повторяющихся действий. С помощью макросов Вы можете решить большинство своих проблем. Если макросов станет недостаточно, нужно обратиться к интегрированному в MS Access языку программирования Visual Basic. Данный язык является наиболее доступным языком программирования в мире, с другой стороны, Visual Basic обеспечивает очень высокую производительность.

Макросы встречаются гораздо реже, чем программы. Как правило, они очень тесно связаны с объектами среды MS Access. Практически каждый управляющий элемент располагает множеством так называемых «реакций на события». События определяют, что должно произойти при выполнении нажатия кнопки, перехода в поле и выходе из него.

Если в окне свойств активизировать одно из событий, MS Access выведет на экран список всех сохраненных в базе данных макросов. Пользователю остается только выбрать один из элементов списка, чтобы связать свойство элемента управления с макросом.

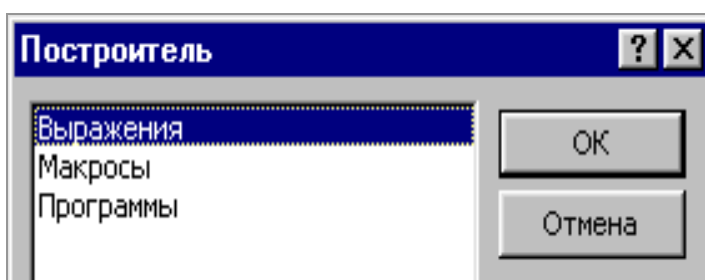


Рисунок 1 - Построитель макросов

Прежде всего, следует открыть в режиме «Конструктора» форму, в которой находится соответствующий объект (например, кнопка). Поместите указатель мыши на объект (в данном случае на кнопку) и нажмите правую кнопку мыши. Объект будет выбран, и откроется контекстное меню, из которого следует выбрать элемент «Свойства». В окне свойств щелкните по вкладке «События». Выберите то событие, в результате которого будет вызываться макрокоманда. В той же строке выполните щелчок мышью на кнопке вызова построителя, которая расположена рядом со строкой ввода значения свойства.

Появляется диалоговое окно «Построитель» (рисунок 1), в котором для запуска построителя макросов необходимо выбрать «Макросы» и нажать «ОК». MS Access создает новый макрос и предлагает пользователю ввести его имя. После этого указанное имя макроса будет автоматически внесено в строку ввода свойства события. Одновременно макрос

открывается в режиме «Конструктора», после чего можно непосредственно приступить к его созданию. Каждый макрос может содержать одну или несколько команд. Макрокоманды могут запускаться с использованием условий или строго последовательно. После завершения работы над макросом его следует сохранить и закрыть окно макроса. Теперь можно снова нажать кнопку вызова построителя. Однако MS Access не создает нового макроса, а открывает заданный, непосредственно готовый к редактированию в режиме «Конструктора».

Автоматический запуск макроса

Как правило, при открытии базы данных постоянно выполняются одни и те же действия. Типичным примером этого является открытие формы, содержащая панель управления прикладной программой. Было бы удобно, если бы MS Access выполнял это действие автоматически. Все, что для этого нужно – это автоматически выполняемый макрос, который называется «AutoExec».

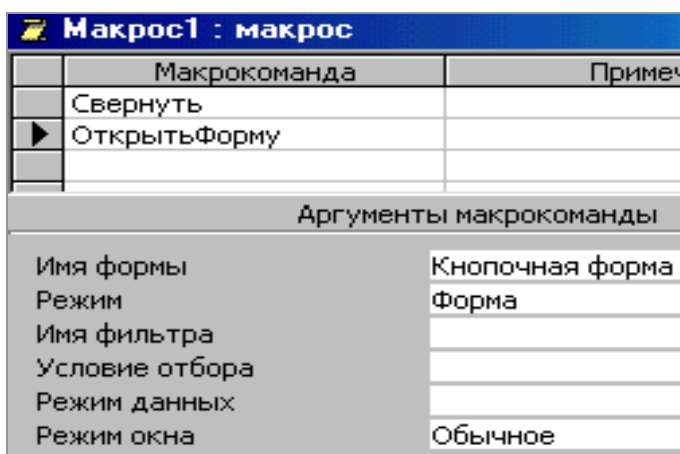


Рисунок 2 – Создание макроса

В окне базы данных выберите вкладку «Макросы». Затем нажмите кнопку «Создать», в результате чего MS Access откроет пустое окно макроса в режиме «Конструктора». Нажмите мышью самую верхнюю строку столбца «Макрокоманда». MS Access отображает на экране список макрокоманд. Осуществите прокрутку списка и выберите элемент «Свернуть», чтобы при запуске макроса окно базы данных автоматически было свернуто и помещено в левый нижний угол экрана.

Далее выберите команду «Открыть Форму». Теперь необходимо определить аргументы макрокоманды. Нажмите мышью строку «Имя форм» в нижней части окна. Здесь MS Access отображает список всех форм базы данных. Выберите ту форму, которая должна открываться автоматически (например, «Кнопочная форма»). Для всех других аргументов можно оставить стандартные установки без изменений (рисунок 2).

Сохраните новый макрос под именем «AutoExec». При следующем открытии базы данных автоматически будет открываться форма «Кнопочная форма».

Если в исключительном случае потребуется подавить выполнение «AutoExec»-макроса, при открытии базы данных следует удерживать нажатой клавишу SHIFT.

В более серьезных случаях в макрос «AutoExec» помещают команды подготовки приложения к работе, например, сюда можно поместить команды подсоединения таблиц из другой БД.

Макросы в качестве заменителей команд меню

Взаимодействие с формой можно упростить, разместив наиболее часто используемые команды меню в виде кнопок. Нажатие кнопки выполняется, как правило, гораздо быстрее, чем выбор команды меню. Это оправдывает себя в том случае, если часто приходится возвращаться в окно базы данных.

MS Access позволяет создавать макрос, в котором можно объединить несколько макрокоманд. Создание такой группы рекомендуется в тех случаях, когда для формы нужно несколько макрокоманд.

При использовании группы макрокоманд можно избежать того, что в окне базы данных появляется каждая отдельная макрокоманда. Это значительно улучшает наглядность.

Определение группы макрокоманд выполняется точно так же, как определение отдельной макрокоманды.

Условие выполнения макроса

Не всегда макрос должен выполняться сразу же, как только вы нажали определенную кнопку. Так, например, можно создать макрос, который последовательно распечатывает несколько отчетов. Прежде чем начнется печать, на экране должно появиться сообщение о том, что печать займет продолжительное время. Процесс печати должен начаться лишь после того, как пользователь согласится с этим.

Создайте новый макрос. Выберите команду «Вид»→«Условия», в результате чего на экране появится столбец «Условия». Прежде всего, следует указать макрокоманды в той последовательности, в которой они должны выполняться. Например, если требуется отпечатать несколько отчетов, следует выбрать макрокоманду «Открыть Отчет» несколько раз.

Затем с помощью аргумента макрокоманды «Режим» можно определить, будет MS Access сначала открывать отчеты в режиме предварительного просмотра или сразу выводить их на печать.

Перейдите в столбец условий, введите там следующее выражение: MsgBox («Печать требует продолжительного времени. Приступить к печати?»; 4+32)=6.

Сначала вызывается встроенная функция MsgBox (Сообщение). Данная функция отображает на экране диалоговое окно с указанным текстом сообщения и двумя кнопками «Да» и «Нет». Функция возвращает значение 6 лишь тогда, когда вы нажмете кнопку «Да».

MS Access выполняет указанную после условия макрокоманду только тогда, когда условие выполнено, а именно, когда пользователь нажимает кнопку «Да».

Теперь, если выбрать пиктограмму «Выполнить» для запуска макроса, MS Access сначала отображает на экране окно диалога. Если нажать кнопку «Нет», то не будет выполнена только первая макрокоманда. Однако второй отчет MS Access распечатает.

Причина этого заключается в том, что по умолчанию условие влияет только на ту макрокоманду, которая расположена в одной строке с условием. Можно повторить условие во всех последующих строках. Однако это приведет к тому, что на экране многократно будет появляться окно диалога.

Решение проблемы заключается во вводе в столбец условий многоточия (...) для всех макрокоманд, которые должны выполняться только при удовлетворении условия. В результате MS Access оценивает условие только один раз, следовательно, и окно диалога появляется только однажды. Если теперь нажать кнопку "Нет", то MS Access не выполняет макрокоманду из первой строки. Кроме этого, он пропускает и все последующие строки, столбец условий которых содержит многоточие.

Следующие друг за другом строки макрокоманд одного условия выполняются только тогда, когда условие удовлетворено.

Если необходимо воспрепятствовать тому, чтобы макрокоманды следующего макроса выполнялись даже тогда, когда условие удовлетворено, в качестве последней макрокоманды макроса следует выбрать «Остановить Макрос». Данная макрокоманда останавливает выполнение макроса.

Упражнение 1. Создание макроса для перевода отчета ПРОДУКТЫ в Ms Word
В окне База Данных выберите Макросы→ Создать.

В первой строке окна Макрос1: макрос в столбце Макрокоманда откройте выпадающий список команд и выберите Открыть отчет. В нижней части бланка макроса выберите:

- Имя отчета – ПРОДУКТЫ;
- Режим – просмотр;
- Режим окна – обычное.

Во второй строке окна Макрос1: макрос в столбце Макрокоманда откройте выпадающий список команд и выберите ВывестиВФормате. В нижней части бланка макроса выберите:

- Тип объекта – отчет;
- Имя объекта – ПРОДУКТЫ;
- Формат вывода – Rich Text Format;
- Имя файла – введите “ДоговорыПродуктWord.doc”;
- Автозагрузка – Да.

В третьей строке окна Макрос1: макрос в столбце Макрокоманда откройте выпадающий список команд и выберите Закрыть. В нижней части бланка макроса выберите:

- Тип объекта – макрос;
- Имя объекта – введите “ДОГОВОРЫ ПО ПРОДУКТАМ В WORD”;
- Сохранение – Подсказка.

Закройте окно с текстом макроса. Подтвердите запрос о сохранении и введите то же имя макроса: “ДОГОВОРЫ ПО ПРОДУКТАМ В WORD”.

Проверьте, как работает макрос, запустив его двойным щелчком или командой Макросы→ Запуск.

Введите какие-либо изменения в Word-файл, например: измените разбивку на страницы, выделите форматированием названия продуктов в заголовках групп, оформите заголовок отчета как титульный лист и т. п. Сохраните этот файл в папке своей группы.

Закройте окно отчета в программе Access.

Упражнение 2. Создание макроса для перевода запроса ИтогиПродукт в Ms Excel

В окне База Данных выберите Макросы→ Создать.

В первой строке окна Макрос1: макрос в столбце Макрокоманда откройте выпадающий список команд и выберите ВывестиВФормате. В нижней части бланка макроса выберите:

- Тип объекта – Запрос;
- Имя объекта – ИтогиПродукт;
- Формат вывода –Microsoft Excel (*.xls);
- Имя файла – введите “ИтогиПродукт в Excel.xls”;
- Автозагрузка – Да.

Во второй строке окна Макрос1: макрос в столбце Макрокоманда откройте выпадающий список команд и выберите Закрыть. В нижней части бланка макроса выберите:

- Тип объекта – макрос;
- Имя объекта – введите “ИТОГИ ПО ПРОДУКТАМ В WORD”;
- Сохранение – Подсказка.

Закройте окно с текстом макроса. Подтвердите запрос о сохранении и введите то же имя макроса: “ИТОГИ ПО ПРОДУКТАМ В WORD”.

Проверьте, как работает макрос, запустив его двойным щелчком или командой Макросы→ Запуск.

Оформите информацию в Excel-файле как ведомость. Сохраните этот файл в папке своей группы.

Упражнение 3. Создание макроса для окончания сеанса работы с базой

В окне База Данных выберите Макросы→ Создать.

В первой строке окна Макрос1: макрос в столбце Макрокоманда откройте выпадающий список команд и выберите Выход. В нижней части бланка макроса выберите Сохранить все.

Во второй строке окна Макрос1: макрос в столбце Макрокоманда откройте выпадающий список команд и выберите Закрывать. В нижней части бланка макроса: выберите:

- Тип объекта – макрос;
- Имя объекта – введите “КОНЕЦ РАБОТЫ”;
- Сохранение – выберите подсказка.

Закройте окно с текстом макроса. Подтвердите запрос о сохранении и введите имя макроса “КОНЕЦ РАБОТЫ”.

Отчет должен содержать:

1. Название, цель и задание лабораторной работы.
2. Результаты выполнения макрокоманд;
3. Ответы на контрольные вопросы.

Контрольные вопросы:

1. Как отображаются результаты макроса?
2. Построитель макросов
3. Автоматический запуск макроса
4. Условие выполнения макроса