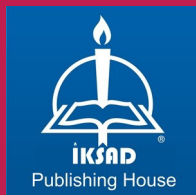


Sevinc Cəbrayılzadə

---

# MS ACCESS БАЗА ДАННЫХ



**Sevinc Cəmil Cahid qızı Cəbrayılzadə**

**МС ACCESS БАЗА ДАННЫХ**

(Учебное пособие)

**ELMİ REDAKTOR.**

**Dos. P.ü.f.d. Tağıyeva Samirə Cəmil- Cahid qızı**

**ADPU İbtidai riyaziyyatın ibtidai**

**kursunun tədrisi texnologiyası kafedrasının dosenti**

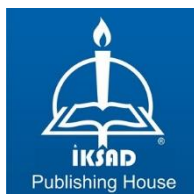
**RƏYÇİ.**

**Prof.p.e.d. Pələngov Əbülfət Qulam oğlu**

**ADPU İbtidai riyaziyyatın ibtidai**

**kursunun tədrisi texnologiyası**

**kafedrasının müdiri**



Copyright © 2023 by iksad publishing house  
All rights reserved. No part of this publication may be reproduced, distributed  
or transmitted in any form or by  
any means, including photocopying, recording or other electronic or  
mechanical methods, without the prior written permission of the publisher,  
except in the case of  
brief quotations embodied in critical reviews and certain other  
noncommercial uses permitted by copyright law. Institution of Economic  
Development and Social  
Researches Publications®  
(The Licence Number of Publicator: 2014/31220)  
TÜRKİYE TR: +90 342 606 06 75  
USA: +1 631 685 0 853  
E mail: iksadyayinevi@gmail.com  
www.iksadyayinevi.com

It is responsibility of the author to abide by the publishing ethics rules.

Iksad Publications – 2023©  
**ISBN: 978-625-367-304-8**  
Cover Design: İbrahim KAYA  
September / 2023  
Ankara / Türkiye  
Size = 14,8 x 21 cm

## ОГЛАВЛЕНИЕ

<b>От автора .....</b>	<b>1</b>
<b>1 . База данных и программа MS ACCESS.....</b>	<b>3</b>
1.1. Информационные системы и технологии. Основные понятия информационных систем и технологий.....	4
1.2.Этапы развития информационных технологий.....	10
1.3. Актуальность применения информационных технологий.....	12
1.4. Виды информационных технологий .....	14
<b>2. Технологии баз данных. СУБД Microsoft Access 2016. Основные понятия баз данных. ....</b>	<b>16</b>
2.1. Таблицы данных в СУБД Microsoft Access 2010. Создание таблицы в режиме Конструктора .....	28
2.2. Создание таблицы в режиме Таблица.....	31
2.3. Построение связей в реляционной СУБД Microsoft Access 2016 .....	32
<b>3. Создание баз данных .....</b>	<b>36</b>
<b>4.Создание таблиц в MS ACCESS .....</b>	<b>52</b>
4.1.Макросы и модули .....	74
4.2. Создание структур в таблицах .....	85
4.3.Этапы проектирования и создания базы данных .....	86
4.4. Имена полей и типы данных .....	90

4.5. Общие свойства поля .....	96
4.6.Схема данных. ....	99
4.7. Создание связей между таблицами схемы данных	100
<b>5.Создание таблиц при помощи ввода данных .....</b>	<b>107</b>
5.1. СВОЙСТВА ТАБЛИЦ И ПОЛЕЙ .....	109
5.2.ТИПЫ ДАННЫХ .....	110
Примеры автоматического определения типа данных.....	112
5.3. ОТНОШЕНИЯ МЕЖДУ ТАБЛИЦАМИ .....	117
<i>Ключи</i> .....	118
<i>Преимущества использования связей</i> .....	120
5.4.Создание новой таблицы. ....	122
Создание таблицы в существующей базе данных .....	123
<b>6. Использование импорта или связи для создания таблицы.....</b>	<b>124</b>
6.1. СОЗДАНИЕ ТАБЛИЦЫ С ПОМОЩЬЮ ИМПОРТА ВНЕШНИХ ДАННЫХ ИЛИ СВЯЗИ С НИМИ.....	125
6.3. ОПРЕДЕЛЕНИЕ ПОЛЕЙ, ИСПОЛЬЗУЕМЫХ В КАЧЕСТВЕ ПЕРВИЧНОГО КЛЮЧА .....	131
<b>7. Вставка и добавление поля в таблицу и создание нового поля.....</b>	<b>141</b>
7.1. Задание свойств полей.....	146
7.2. ИЗМЕНЕНИЕ ТИПА ДАННЫХ ПОЛЯ .....	148
7.3. ИЗМЕНЕНИЕ ФОРМАТА ПОЛЯ.....	150

7.4. ОТКРЫТИЕ ТАБЛИЦЫ В КОНСТРУКТОРЕ.....	152
<b>8. Ввод данных в таблицу и редактирование их .....</b>	<b>156</b>
8.1. РЕДАКТИРОВАНИЕ ДАННЫХ.....	158
<b>9. Модификация структуры таблиц .....</b>	<b>159</b>
9.1. Поиск данных и замена данных.....	165
9.2. Проектирование многотабличной базы данных .....	170
<b>10. Выбор данных из таблиц при помощи запросов.....</b>	<b>176</b>
<b>11. Выполнение вычислительных операций в таблицах.....</b>	<b>184</b>
<b>12. Форматы в таблицах и способ их создания .....</b>	<b>192</b>
<b>13. Создание иерархических форм межтабличных связей.....</b>	<b>195</b>
<b>14. Отчеты и способы их создания.....</b>	<b>198</b>
<b>15. Макросы и их способ создания.....</b>	<b>202</b>
15.1. ФОРМИРОВАНИЕ МАКРОКОМАНД В ОКНЕ МАКРОСА.....	206
15.2. ПРИМЕР РАЗРАБОТКИ МАКРОСА .....	207
15.3. КОНСТРУИРОВАНИЕ МАКРОСА.....	213
15.4. ФОРМИРОВАНИЕ МАКРОКОМАНД В ОКНЕ МАКРОСА.....	214
<b>СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....</b>	<b>219</b>



## **От автора**

Принимая во внимание важность систем управления базами данных на практике, а также принимая во внимание важность получения знаний в этом аспекте как теоретически, так и практически, мы решили написать этот учебник для наших Уважаемых Читателей.

Как известно, программа Microsoft Access пакета офисных программ Microsoft создавалась в основном для создания простой базы данных. становится возможным выполнение практических операций, таких как

По мере обновления программных пакетов от Microsoft пользователям становится проще и интереснее работать с представленными в них программами. Учитывая такие качества, мы также поставили цель теоретического и практического освоения базы данных Ms Access пакета Microsoft Office 2016. В данном учебнике мы посчитали целесообразным представить теоретическую информацию, которую могут усвоить студенты и аспиранты, обучающиеся на учителя информатики, а также практические примеры.



Принимая во внимание, что выполнять практические задания без теоретической информации сложно и утомительно, в связи с этим мы позаботились о том, чтобы материалы были размещены в более простой и понятной форме, а в конце к ним давались простые практические указания. Учебник рассчитан на русскоязычных читателей, и выделена цель обеспечить их необходимыми знаниями для более легкого доступа к материалам преподаваемого курса и в то же время для развития их практических умений. Для студентов бакалавриата, специализирующихся в области компьютерных наук, этот учебник будет выделяться своей удобочитаемостью и простым языком, и я надеюсь, что он мгновенно станет настольной книгой. Хотя в современное время созданы языки для создания большого количества баз данных, их структура и механизм выполнения более сложны, но программа Ms Access более удобна в плане создания и подключения нескольких таблиц.



**Автор.**



## **1 . База данных и программа MS ACCESS**

**Цель.** Знакомство с общей характеристикой дисциплины. Рассмотрение основных понятий информационных систем (ИС) и технологий (ИТ), истории их развития, актуальность применения на транспорте и классификацию. Ознакомить студентов с таблицами данных в СУБД Microsoft Access 2016.

Рассмотреть основные этапы создания таблиц в Access и особенности их применения на транспорте.

1. Информационные системы и технологии
  - 1.1. Основные понятия информационных систем и технологий.
  - 1.2. Этапы развития информационных технологий.
  - 1.3. Актуальность применения информационных технологий на транспорте.

- 1.4. Виды информационных технологий на транспорте.
2. Технологии баз данных. СУБД Microsoft Access 2010.
  - 2.1. Основные понятия баз данных.
  - 2.2. Развитие СУБД.
  - 2.3. Функции современных СУБД.
  - 2.4. СУБД Microsoft Access 2016.
  - 2.5. Интерфейс СУБД Microsoft Access 2016.
  - 2.6. Начало работы в Microsoft Access 2016.
3. Таблицы данных в СУБД Microsoft Access 2016.
  - 3.1. Создание таблицы в режиме Конструктора.
  - 3.2. Создание таблицы в режиме Таблица.
  - 3.3. Построение связей в реляционной СУБД

Microsoft Access 2016

### **1.1. Информационные системы и технологии. Основные понятия информационных систем и технологий**

Курс «Информационные системы и технологии» (ИСИТ) посвящен изучению способов и методов переработки исходной информации для получения информации нового

качества, а также основных принципов организации и обработки больших массивов данных, описывающих 2

Информация в автоматизированной информационной системе (АИС) – совокупность всех данных и программ, которые используются в АИС независимо от способа их физического и логического представления. Информация может отражаться в вещественном виде (документе) и в устных сообщениях. Организация информационных ресурсов на предприятии осуществляется через функции информационного обеспечения систем управления. Данные – это первичные сведения, получаемые в результате прямого наблюдения за совершившимся событием в каком-либо контролируемом объекте или процессе в форме чисел, символов, знаков и слов. Понятие информационной технологии (ИТ) появилось с возникновением информационного общества, основой социальной динамики в котором являются не традиционные, материальные, а информационные ресурсы – знания, наука, организационные вопросы, интеллектуальные способности людей, их инициатива и творчество. Информационная технология – совокупность методов и средств сбора, обработки и передачи данных для получения информации нового качества и ее использования на базе вычислительной и информационной

техники. ИТ представляют собой научно-методическую и технологическую базу информационной индустрии. В этом понятии объединяются методы и средства для производства, передачи, обработки и использования информационных ресурсов, а также для создания инструментов и технологий информационной индустрии. Создание ИТ базируется на использовании многих видов современных отраслей: компьютерной, телекоммуникационной и пр. При решении конкретных проблем обычно ограничиваются той частью реального мира, которая является областью данной деятельности. В этих случаях интерес представляют лишь некоторые его объекты. Совокупность объектов части реального мира, которая является областью рассматриваемой деятельности, называется предметной областью, а сами объекты - объектами предметной области. Объектами могут быть:

- люди, например, перечисленные в какой либо платежной ведомости;
- предметы, например детали, которые производятся;
- построения – воображаемые объекты, например счета, в задаче получения счетов. Очевидно, что одновременно могут существовать несколько

предметных областей, соответствующих разным проблемам. Информационная система - это взаимосвязанная совокупность информационных, технических, программных, технологических и других средств, а также персонала, предназначенная для сбора, обработки, хранения и выдачи информации и принятия управленческих решений. В деятельности организации информационная система рассматривается как программное обеспечение, реализующее деловую стратегию организации.

### **Свойства информационных систем:**

- любая ИС может быть подвергнута анализу, построена и управляема на основе общих принципов построения сложных систем;
- при построении ИС необходимо использовать системный подход;
- ИС является динамичной и развивающейся системой;
- ИС следует воспринимать как систему обработки

информации, состоящую из компьютерных и телекоммуникационных устройств, реализованную на базе современных технологий;

- выходной продукцией ИС является информация, на основе которой принимаются решения или производятся автоматическое выполнение рутинных операций;
- участие человека зависит от сложности системы, типов и наборов данных, степени формализации решаемых задач. Процессы в информационной системе:
  - ввод информации из внешних и внутренних источников;
  - обработка входящей информации;
  - хранение информации для последующего ее использования;
  - вывод информации в удобном для пользователя виде;
  - обратная связь, т.е. представление информации, переработанной в данной организации, для корректировки входящей информации.

Этапы развития информационных систем. Первые ИС появились в 50-х гг. В эти годы они были предназначены для обработки счетов и расчета зарплаты, а реализовывались на электромеханических бухгалтерских счетных машинах. Это приводило к некоторому сокращению затрат и времени на подготовку бумажных документов. 60-е гг. знаменуются изменением отношения к ИС.

Информация, полученная из них, стала применяться для периодической отчетности по многим параметрам. Для этого организациям требовалось компьютерное оборудование широкого назначения, способное выполнять множество функций, а не только обрабатывать счета и считать з/пл. В 70-х - начале 80-х ИС начинают широко использоваться в качестве средства управленческого контроля, поддерживающего и ускоряющего процесс принятия решений. К концу 80-х гг. концепция использования ИС вновь изменяется. Они становятся стратегическим источником информации и используются на всех уровнях организации любого профиля. ИС этого периода, предоставляя вовремя нужную информацию, помогают организации достичь успеха в своей деятельности, создавать новые товары и услуги, находить новые рынки сбыта, обеспечивать себе достойных партнеров,



организовывать выпуск продукции по низкой цене и многое другое.

## **1.2. Этапы развития информационных технологий**

В своем развитии ИТ прошли несколько этапов: а) ручные информационные технологии (до конца XIX в.). Технические средства: перо, чернильница, бумага и другие подручные средства, известные из истории. Коммуникации: почтовая связь, голубиная почта и др.; б) механические информационные технологии (до середины XX в.). Технические средства: пишущая машинка (механическая), арифмометр, авторучка. Коммуникации: почта, телеграф, телефон; в) электрические информационные технологии (до середины XX в.). Технические средства: электрические пишущие машинки, диктофоны, магнитофоны, копировальные машины. Коммуникации: те же; г) электронные информационные технологии (настоящее время). Данный этап обусловлен появлением ЭВМ.

Содержит два подэтапа: 1-й - появление в конце 60-х - начале 70-х гг. ЭВМ. Именно с появлением ЭВМ и возникла необходимость разработки информационных технологий. Появились первоначально крупные ЭВМ, призванные

централизованно обрабатывать большие объемы информации в едином центре. Этот период характеризуется повсеместным внедрением автоматизированных систем управления (АСУ) на предприятиях; 2-й - появление в начале 80-х г. персональных ЭВМ. Происходит принципиальная модернизация идей АСУ: от ВЦ и централизации управления к распределенному вычислительному потенциалу, повышению однородности технологии обработки информации, к децентрализации управления. Современный этап развития ИТ характеризуется следующими особенностями: а) распределенная компьютерная техника - ПК находится на каждом рабочем месте. Каждый участник управленческого процесса использует самостоятельное автоматизированное рабочее место (АРМ); б) "дружественное" программное обеспечение (интуитивный интерфейс) - каждый пользователь, независимо от специализации может свободно решать свои задачи на ПК; в) развитые коммуникации - в настоящее время невозможно представить себе АРМ или ПК, не объединенный с другими пользователями в единую сеть; г) развитие сети Интернет - широкое распространение сети Интернет и "всемирной паутины" World Wide Web; 4 д) беспроводные технологии - беспроводные модемы, позволяющие подключать

переносной компьютер к АСУ АТП в любой момент и в любом месте. е) средства мультимедиа - распознавание речи, трехмерное изображение, объемный звук и др. Создание мультимедиа-документов с помощью перечисленных выше средств, видео и аудио презентаций, рассылка их по электронной почте, проведение видео - и телеконференций - это перспектива новых информационных технологий на базе мультимедиа.

### **1.3. Актуальность применения информационных технологий**

Современная цивилизация является постиндустриальным или информационным обществом, в котором знания, представленные в виде информационных ресурсов, становятся главным достоянием и важнейшим фактором экономического развития, а информационная индустрия – одной из основных отраслей экономики. Процессы информатизации человеческой деятельности, как в производственной, так и в непроизводственной сфере являются столь масштабными и глубокими, что ведут к качественным изменениям самого общества, безгранично расширяя область применения продуктов и сервисов информационной индустрии, неуклонно вовлекая в мир

обработки информации все общество. ИТ качественно изменили процессы управления во всех областях человеческой деятельности, в том числе и на транспорте. Электронная торговля (E-Commerce), интернеттехнологии, автоматизированное управление на базе современных технических и программных средств открыли новые возможности повышения эффективности работы транспорта и экономичности логистических систем. Этому в значительной мере способствовали современные системы телекоммуникаций и в первую очередь мобильная система связи на основе стандарта GSM (Global System for Mobile Communication). Большое значение для автоматизации на всех видах транспорта имеет глобальная система определения местоположения транспортных средств (GPS) на основе спутниковой связи. В значительной мере автоматизации и информатизации на транспорте способствовали успехи в области идентификации грузов и носителей на основе штрихового кода, новые радиочастотные технологии идентификации с применением транспондеров и др. В качестве основного направления для оптимизации использования автомобиль 5 - диспозиция (изменение положения) транспортных средств; - постоянное

перемещение грузов; - переменные спрос и предложение на рынке перевозок и др.

#### **1.4. Виды информационных технологий**

По признаку сферы применения принято различать базовые, прикладные и специальные информационные технологии. Прикладные информационные технологии – технологии, реализующие адаптированные к конкретным областям применения типовые способы работы с информацией.

Примерами прикладных ИТ могут служить:

- ИТ в управлении;
- ИТ в промышленном производстве;
- ИТ в торговле;
- ИТ в образовании;
- ИТ в медицине и др.

В этих сферах информация (данные, информационные сообщения, информационные продукты) выступает в качестве ресурса, средства или промежуточного продукта деятельности, но не является его конечным продуктом. Основная задача прикладных информационных технологий

– рациональная организация того или иного конкретного информационного процесса. Осуществляется это путем адаптации к данному конкретному применению одной или нескольких базовых информационных технологий, позволяющих наилучшим образом реализовать отдельные фрагменты этого процесса. Поэтому основными научными проблемами в области исследования прикладных информационных технологий можно считать следующие:

1. Разработка методов анализа, синтеза и оптимизации прикладных ИТ.
2. Создание теории проектирования ИТ различного вида и практического назначения.
3. Создание методологии сравнительной количественной оценки различных вариантов построения ИТ.
4. Разработка требований к аппаратно-программным средствам автоматизации процессов реализации ИТ.

Например, работа сотрудника кредитного отдела банка с использованием ЭВМ обязательно предполагает применение совокупности банковских технологий оценки кредитоспособности заемщика, формирования кредитного договора и срочных обязательств, расчета графика платежей

и других технологий, реализованных в какой-либо информационной технологии: СУБД, текстовом процессоре и т. д. Трансформация обеспечивающей информационной технологии в чистом виде в функциональную (модификация некоторого общеупотребительного инструментария в специальный) может быть сделана как специалистом проектировщиком, так и самим пользователем. Это зависит от того, насколько сложна такая трансформация, т. е. от того, насколько она доступна самому пользователю. Эти возможности все более и более расширяются, поскольку обеспечивающие технологии год от года становятся дружелюбнее.

## **2. Технологии баз данных. СУБД Microsoft Access**

### **2016. Основные понятия баз данных.**

Данные (в концепции баз данных) – это упорядоченный набор конкретных значений параметров, характеризующих объект, событие или любые другие факторы. База данных (БД) – именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области, предназначенная для обеспечения накопления и многоцелевого использования

данных. Система управления базами данных (СУБД) – совокупность программных средств, предназначенных для создания, ведения и использования БД. СУБД является универсальным программным инструментом создания и обслуживания БД, позволяет не только хранить большие массивы данных в определенном формате, но и обрабатывать их, представляя в удобном для пользователей виде. Приложения - программы, с помощью которых пользователи работают с базой данных.

Развитие СУБД История СУБД как особого вида программного обеспечения неразрывно связана с историей начала использования электронно вычислительных машин для организации хранения и обработки информации. Именно в то время (конец 60х - начало 70-х годов) были разработаны основы программного обеспечения для создания и эксплуатации информационных систем (ИС). В конце 70-х - начале 80-х годов направление программного обеспечения под общим названием «СУБД» превратилось в одну из наиболее бурно развивающихся отраслей программной индустрии. С начала своего возникновения в конце 60-х годов автоматизированные ИС (АИС) ориентировались на хранение и обработку больших объемов данных, которые не



могли быть одновременно и полностью размещены в оперативной памяти ЭВМ. В структуре программного обеспечения ЭВМ, как в то время, так и сейчас за организацию, размещение и оперирование данными в постоянной памяти отвечает операционная система ЭВМ, соответствующий компонент которой чаще всего называется «файловой системой». Данные во внешней памяти компьютера представлены именованными совокупностями, называемыми файлами. В большинстве случаев операционная (файловая) система не «знает» внутренней смысловой логики организации данных в файлах и оперирует с ними как с однородной совокупностью байтов или строк символов. С точки зрения смысла и назначения АИС файлы данных имеют структуру, отражающую информационно логическую схему предметной области АИС. Эта структура данных в файлах должна обязательно учитываться в операциях обработки (собственно, в этом и заключается одна из основных функций АИС). Вместе с тем, в силу невозможности в большинстве случаев размещения файлов баз данных сразу целиком в оперативной памяти компьютера, структуру данных в файлах баз данных приходится учитывать при организации операций обращения к файлам во внешней памяти. Отсюда вытекает

основная особенность СУБД как вида ПО. Будучи по природе прикладным программным обеспечением, т. е. предназначенным для решения конкретных задач, СУБД изначально выполняли и системные функции – расширяли возможности файловых систем системного ПО.

Функции современных СУБД Можно выделить следующие функции, реализуемые современными СУБД: - организация и поддержание логической структуры данных (схемы БД); - организация и поддержание физической структуры данных в постоянной памяти; - организация доступа к данным и их обработка в оперативной и постоянной памяти. Организация и поддержание логической структуры данных (схемы БД) обеспечивается средствами модели организации данных. Модель данных определяется способом организации данных, ограничениями целостности и множеством операций, допустимых над объектами организации данных. Модель данных, реализуемая конкретной СУБД, является одной из основных компонент, определяющих функциональные возможности СУБД по отражению в базах данных информационно-логических схем предметных областей АИС. Модели данных, поддерживаемые СУБД, довольно часто используются в качестве критерия для классификации СУБД. Исходя из

этого, различают иерархические СУБД, сетевые СУБД и реляционные СУБД. Другой важной функцией СУБД является организация и поддержание физической структуры данных во внешней памяти. Эта функция включает организацию и поддержание внутренней структуры файлов базы данных, иногда называемой форматом файлов базы данных, а также создание и поддержание специальных структур (индексы, страницы) для эффективного и упорядоченного доступа к данным. В этом плане эта функция тесно связана с третьей функцией СУБД – организацией доступа к данным. Организация доступа к данным и их обработка в оперативной и внешней памяти осуществляется через реализацию процессов, получивших название транзакций. Транзакцией называют последовательную совокупность операций, имеющую отдельное смысловое значение по отношению к текущему состоянию базы данных. Так, например, транзакция по удалению отдельной записи в базе данных последовательно включает определение страницы файла данных, содержащей указанную запись, считывание и пересылку 7 соответствующей страницы в буфер оперативной памяти, собственно удаление записи в буфере ОЗУ, проверку ограничений целостности по связям и другим параметрам

после удаления и, наконец, «выталкивание» и фиксацию в файле базы данных нового состояния соответствующей страницы данных. СУБД Microsoft Access 2016 Система управления базами данных Microsoft Access 2016 – это одно из приложений, входящих в состав пакета Microsoft Office 2010. Приложение является мощным программным средством обработки БД, которую можно использовать для хранения, сортировки и управления практически любыми типами данных. В качестве примеров можно привести имена и адреса, информацию о продавцах и складах, коллекции фотографий и пр. Microsoft Access является системой управления реляционными БД. Реляционная база данных (relational database – от relation (связь)) – это такая база данных, записи которой можно сравнивать друг с другом и другими источниками данных и анализировать в целях создания динамического источника данных. Содержание такой БД может изменяться на основе результатов сортировки, запросов или других операций с данными. Хотя Microsoft Access является довольно мощной программой, ее очень легко использовать на практике. БД определяется простым указанием полей данных, которые должны содержаться в базе (например, имя, адрес и индекс). Можно легко создавать формы, которые помогут вводить данные в

базу, не задумываясь о работе программы управления этой БД, просматривать введенные данные в удобном виде, создавать запросы для отбора информации по определенным условиям и формировать отчеты. Реляционная БД допускает использование нескольких таблиц с данными, связанными между собой особым образом. Это позволяет существенно уменьшить объем таблиц и увеличить быстродействие СУБД. В Microsoft Access база данных включает в себя все объекты, связанные с данными, в том числе и те, которые предназначены для автоматизации работы с ними, такие, как таблицы, формы, запросы, отчеты, программный код. Средства графического конструирования позволяют пользователю создавать объекты базы данных и объекты приложения с помощью многочисленных графических элементов, не прибегая к программированию. Диалоговые средства представлены разнообразными мастерами, которые в режиме ведения диалога с пользователем позволяют создавать объекты и выполнять разнообразные функции по реорганизации и преобразованию БД. В СУБД поддерживаются различные модели данных. Модель данных – это метод (принцип) логической организации данных, используемый в СУБД. Наиболее известными являются иерархическая, сетевая и реляционная модели. В СУБД для

ПК поддерживается преимущественно реляционная модель, которую отличает простота и единообразие представления данных несколькими простейшими двумерными таблицами. Основной логической структурной единицей манипулирования данными является строка таблицы – запись. Структура записи определяется составом входящих в нее полей.

Совокупность полей записи соответствует логически связанным реквизитам, характеризующим некоторую сущность предметной области. Типовыми функциями СУБД по манипулированию данными являются: выборка, добавление, удаление, изменение данных. Microsoft Access называет объектами все, что может иметь имя, такими объектами являются таблицы, запросы, формы, макросы и модули.

Рассмотрим список объектов Ms Access.

1. Таблица – объект, который используется для хранения информации. Поле – столбец таблицы, содержащий определенное свойство объекта. Свойства полей следующие: - каждое поле имеет имя; - внутри имени поля нельзя использовать точки; - для связки между словами можно ставить знак

подчеркивания; тип поля определяет множество значений, которые может принимать данное поле в различных записях; 8 - в реляционных базах данных используется четыре основных типа полей: числовой, символьный, дата, логический.

2. Запрос – объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц. Для создания запроса можно использовать QBE (запрос по образцу) или инструкции SQL. Можно создавать запросы на выборку, удаление или добавление данных, можно создавать новые таблицы, используя данные одной или нескольких таблиц, которые уже существуют.
3. Форма – объект, предназначенный в основном для ввода данных, отображения их на экране или управления работой приложения.
4. Отчет – объект, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения.
5. Макрос – объект, представляющий собой структурированное описание одного или нескольких действий, которые, по мнению разработчика,

должен выполнить Access в ответ на определенное событие. 6. Модуль – объект, содержащий программы на Microsoft Access Basic, которые позволяют разбить процесс на более мелкие действия и обнаружить ошибки, которые невозможно обнаружить с помощью макросов. Модули могут быть независимыми объектами, содержащими функции, которые вызываются из любого места приложения, также модули могут быть «привязаны» к отдельным формам или отчетам для реакции на происходящие в них изменения.

Интерфейс СУБД Microsoft Access 2016 Access имеет характерный для всех приложений Microsoft Windows удобный графический интерфейс, ориентированный на комфортную работу пользователя. Для работы с таблицами БД и другими объектами Access предоставляет многочисленные команды меню и контекстно зависимые панели инструментов. Пользователь имеет возможность переносить объекты БД и их элементы с помощью мыши. Например, любую таблицу или запрос можно перенести из окна БД в окно схемы данных. Для установления связи между объектами можно в схеме данных переместить поле



из одной таблицы в другую. Для размещения подчиненной формы в главной достаточно перенести в нее ранее созданную форму или даже просто перенести таблицу-источник в конструируемую форму.

В Access предусмотрено широкое использование технологии, которая помогает пользователю ориентироваться в выборе необходимых действий и обеспечивает высокую производительность труда за счет автоматизации выполнения основных функций. Выдача Помощником контекстно-зависимой справочной информации помогает принять решение, как лучше выполнить то или иное действие или найти нужный инструмент в Access. При вызове справки ее окно не заслоняет окна Access, которое автоматически изменяет размер, освобождая пространство для окна справки. В Access обеспечивается более удобный доступ к справочной системе через поле Введите вопрос, расположенное в строке меню. Access располагает разнообразными диалоговыми средствами пользователя, которые позволяют создавать приложения для решения задач, не прибегая к разработке запросов на языке SQL или к программированию макросов или модулей на языке Microsoft Visual Basic for Applications.

Для автоматизации процесса создания объектов базы данных (таблиц, запросов), схем базы данных и объектов приложения (форм, отчетов) используются специализированные диалоговые графические средства, называемые Конструктор (Design). Конструктор предоставляет пользователю набор инструментов, с помощью которых можно быстро создать и модифицировать объект. Предусмотрено также автоматическое конструирование форм, запросов, отчетов, страниц и их элементов с помощью программ-мастеров и команд, начинающихся с приставки «авто». Мастера Access позволяют автоматизировать процесс создания таблиц БД, форм, запросов, отчетов, анализировать таблицы БД и выполнять многие другие работы. Практически для любых работ имеется Мастер (Wizard), который поможет их выполнить. Вот перечень некоторых из них: мастер подстановок, мастер запросов, мастера по созданию форм и отчетов, мастер кнопок, мастер анализа таблиц, мастер сводных таблиц, мастер баз данных и т.д. Начало работы в Microsoft Access 2016 Чтобы начать работу с СУБД Access, необходимо запустить ее после загрузки операционной системы. Это можно сделать, например, так: в меню Пуск выбрать пункт Программы и в появившемся меню щелкнуть

на пункте Microsoft Access. Для быстрого запуска Access удобно иметь ярлык этой программы на рабочем столе Windows. Создать ярлык можно разными способами. Например, выберите программу Microsoft Access, как при ее запуске. Нажмите правую кнопку мыши. В контекстном меню последовательно выберите команды Отправить и Рабочий стол. Ярлык будет создан и отобразится на рабочем столе и запуск можно выполнить двумя нажатиями мыши на ярлыке. Для получения справки о назначении команды можно вызвать справку, выбрав последовательно пункты меню Файл→Справка. Эта команда позволяет получить различные сведения о текущем режиме Access, а также сведения более общего характера о программе. В отличие от многих других приложений (например, Word), пользователю нет необходимости специально сохранять базу данных. Сохранение выполняется автоматически после присвоения названия БД. Access записывает данные на диск всякий раз при вводе записей или внесении изменений в объекты БД.

## **2.1. Таблицы данных в СУБД Microsoft Access 2010. Создание таблицы в режиме Конструктора**

Создание таблиц данных рассмотрим на примере создания базы данных Отдел кадров. Данная база данных

состоит из трех таблиц: Співробітник, Штатний Розклад. Рассмотрим создание в MS Access таблицы данных . Для создания новой базы данных после запуска Access необходимо выбрать пункты меню Файл→Создать и в открывшемся окне выбрать пункт Новая база данных. Затем в появившихся справа двух ячейках ввести название папки, где будет размещена создаваемая БД и ее название . После нажатия кнопки Создать открывается окно новой БД. Вид окна Microsoft Access 2016 при создании новой базы данн 10

При выборе режима Конструктор появляется окно Таблица 1, в котором определяется структура таблицы БД . При отображении таблицы в режиме Конструктор в окне можно выделить три основные области: панель инструментов Конструктор таблиц, панели Бланк и Свойства поля. Панель инструментов используют для модификации содержимого этой области. Основные кнопки панели Следующие: Область панели Бланк – основная часть окна Режим конструктора таблиц. Здесь можно задать имена полей, типы данных и их описание. Область состоит из столбца маркера текущей строки и ключевого поля (крайний слева), столбцов Имя поля, Тип данных и Описание. В столбце Имя поля можно изменять (или назначать) имена полей. Имена должны содержать не более 64 символов, цифр и пробелов. Кроме

того, они должны быть уникальны для данной таблицы. Типы данных, которые можно хранить в данном поле, определяются в данном столбце. Недопустимо заносить в поле данные несовместимого с ним типа. Нельзя, например, размещать текст в числовом поле. Вид окна Microsoft Access 2016 при создании новой таблицы данных в режиме Конструктор Общие свойства для каждого поля задаются в области Свойства полей, эти свойства можно установить, чтобы задать метод сохранения и отображения данных этого поля. Поля, создаваемые в режиме Конструктор, используются другими объектами базы данных, работающими с таблицами, например, формами, отчетами и запросами. Свойства полей изменяют на вкладках Общие и Подстановка. Вид окна Microsoft Access 2016 при создании таблицы данных Співробітник в режиме Конструктор На вкладке Подстановка задается свойство Тип элемента управления, которое определяет, в каком виде будет отображаться поле в таблице и в форме: в виде Поля, Списка или Поля со списком. Если для поля выбран тип элемента управления Список или Поле со списком, на вкладке Подстановка появляются дополнительные свойства. Каждая таблица в реляционной БД должна иметь уникальный ключ (ключевое поле), однозначно определяющий каждую запись.

Это позволяет быстро найти и связать данные из разных таблиц в запросах, формах и отчетах. Ключевое поле должно содержать уникальные значения, такие как коды или инвентарные номера и не может содержать значения Null. Ключ может быть простым или составным, включающим несколько полей (до 10). Для определения ключа выделите соответствующую строку и нажмите на панели инструментов Конструктор таблиц кнопку Ключевое поле или выполните команду меню

Правка→Ключевое поле. Примечание. Режим конструктора позволяет в любой момент изменить структуру таблиці вне зависимости от способа ее создания.

## **2.2. Создание таблицы в режиме Таблица**

Пользователь может создать таблицу, не определяя предварительно ее структуру. Для этого необходимо выбрать на ленте вкладку Создание и нажать кнопку Таблица. После выбора этого режима открывается заготовка для создания новой таблицы. В создаваемой таблице уже присутствует поле Код (для автоматической нумерации записей). Для создания следующего поля необходимо щелкнуть ЛК мыши в графе Щелкнуть для добавления. В появившейся графе

необходимо выбрать тип данных для этого поля. Если необходимы еще поля, то требуется повторить две последних операции. Полям таблицы по умолчанию присваиваются имена Поле1, Поле2 и т.д. Любое поле этой таблицы можно переименовать в соответствии с требованиями пользователя непосредственно редактируя имена в заголовках полей при помощи контекстного меню.

Вид окна Microsoft Access 2016 при создании новой таблицы данных в режиме Создание→ Таблица После заполнения данными первой записи и нажатия клавиши Enter откроются поля следующей записи для ввода данных.

### **2.3. Построение связей в реляционной СУБД Microsoft Access 2016**

Одно из важнейших достоинств реляционных баз данных состоит в возможности хранить логически сгруппированные данные в разных таблицах и задавать связи между ними, объединяя их в единую базу. Связь между таблицами устанавливает отношения между совпадающими значениями в общем для этих таблиц поле. Такая организация данных позволяет уменьшить избыточность хранимых данных, упрощает их ввод и организацию запросов и отчетов, а также снижает затраты времени и

ресурсов при работе с БД. В Access можно задать три вида связи между таблицами: Один-ко-многим, Многие-ко-многим и Один-к-одному. Связь Один-ко-многим – наиболее часто используемый тип связи между таблицами. В такой связи каждой записи таблицы А может соответствовать несколько записей таблицы В (поля с этими записями называют внешними ключами), а запись таблицы В не может иметь более одной соответствующей ей записи в таблице А. При связи Многие-ко-многим одной записи таблицы А может соответствовать несколько записей в таблице В, а одной записи таблицы В – несколько записей таблицы А. При связи Один-к-одному запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот. Это тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением Один-к-одному применяют для разделения очень широких таблиц, для отделения части таблицы в целях ее защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице. Для создания логической модели нужно после создания таблиц выполнить команды Работа с базами данных→Схема данных. После этого откроется окно Схема данных. Для добавления в схему таблиц нужно выполнить



команду Отобразить таблицу. В окне Добавление таблицы 13 выбираем нужную таблицу (делаем текущей) и нажимаем кнопку Добавить. После добавления всех нужных таблиц закрываем окно нажатием кнопки Закреть. После добавления таблиц в схему установим связи между ними. Для этого разместим таблицы в нужном порядке (установим курсор мыши в заголовок таблицы и, удерживая кнопку мыши нажатой, перетаскиваем ее). Далее выделяем в таблице поле для связи и, удерживая кнопку мыши нажатой, тянем ее к соответствующему полю другой таблицы. На экране появляется окно Изменение связей, где определяется тип отношения, указывается, нужно ли обеспечивать целостность данных, а также, нужно ли каскадное обновление связанных полей и каскадное удаление связанных полей. Окно для создания или изменения связей между таблицами После нажатия кнопки Создать это окно закрывается, а в схеме данных появляется «связь».

Окно для отображения схемы данных 14 Созданная схема данных отражает логическую схему БД Отдел кадров, которая будет в дальнейшем использоваться для создания форм, запросов и отчетов.

**Вопросы для самостоятельной работы:**

1. Сформулируйте понятия информационной технологии и информационной системы?
2. Раскройте понятие информационной технологии.
3. Раскройте понятие информационной системы.
4. Перечислите этапы развития информационных технологий и дайте им характеристику.
5. Перечислите этапы развития информационных систем и дайте им характеристику.
6. Обоснуйте актуальность применения информационных технологий на транспорте.
7. Какие виды информационных технологий применяются на транспорте?
8. Какие основные понятия баз данных?
9. Охарактеризуйте основные этапы развития СУБД.
10. Какие функции у современных СУБД?
11. Опишите СУБД Microsoft Access 2016.
12. Как начать работу в Microsoft Access 2016?
13. Как создать таблицу в режиме Конструктора?
14. Как создать таблицу в режиме Таблица?
15. Как построить связи в реляционной СУБД Microsoft Access 2016?

16. Какие существуют способы создания таблиц в СУБД MS Access?
17. Как удалить данные из полей таблиц БД?
18. Как можно ввести данные в таблицу БД?

### **3. Создание баз данных**

Основные понятия баз данных Базой данных (БД) называют множество взаимосвязанных элементарных групп данных, которые могут обрабатываться одной или несколькими прикладными системами. Чаще всего это файлы специального формата, содержащие информацию, структурированную заданным образом. База данных используется различными задачами в рамках единой информационной системы. Электронные базы данных используются в различных видах деятельности: от экономики, медицины, криминалистики до экологии. Система базы данных состоит из хранилища данных и программного обеспечения общего назначения, называемого системой управления базой данных (СУБД). СУБД — это программные средства, с помощью которых можно создавать базу данных, наполнять, осуществлять хранение и быстрый поиск информации, осуществлять

пользовательские запросы. В мире существует немало различных систем управл подлежит изучению с целью организации процесса управления посредством использования информационных технологий. Т.о. предметной областью можно считать предприятие, фирму, отрасли промышленности и т.д. Ее описание может включать в себя сведения о трудовых, материальных, финансовых ресурсах отдельных составных единиц и о других предметах, событиях и явлениях. Фундаментальными понятиями в теории БД являются объект, атрибут и сущность. Объектом или сущностью (entities) называется элемент информационной системы, сведения о котором хранятся в БД. Информационное отображение некоторой сущности, т. е. реального объекта, явления, процесса или события, о которых должна быть представлена информация в базе данных, — информационный объект программного обеспечения. Информационный объект может быть конкретным (человек, товар, фирма, населенный пункт) или абстрактным (событие, явление, философская категория, характеристика и т.д.). Некоторые объекты обладают одними и теми же свойствами. Классом объектов называют их совокупность, обладающих одинаковым набором свойств. Например, классами объектов могут быть фирмы, товары на

складе, люди, работающие на предприятии, и т.п. Атрибут (attributes) — это информационное отображение свойств объекта, данное или показатель, который характеризует объект и принимает для конкретного экземпляра объекта числовое или текстовое значение. Каждый объект характеризуется набором атрибутов. Например, возьмем в качестве класса объектов фирмы, реализующие компьютерную технику. Стоимость реализуемого товара — это данное (атрибут), которое принимает числовое значение (принтер — 2400р, сканер — 3200).

Помимо понятий сущность и атрибут, при работе с базами данных часто используются понятия «поле», «запись», «таблица». Поле (field) — это минимальная неделимая структура организации данных. Для каждого поля базы данных определены следующие значения: - уникальное имя; - тип данных, которые будут храниться в поле (числовой, символьный, логический и т. д.); - максимальная длина (или размер), которую могут иметь данные, хранящиеся в поле; - дополнительные характеристики (например, для числовых данных точность, для символьных — формат, и т. д.). Множество логически связанных полей образуют запись (record). Запись БД является строкой таблицы. Экземпляр записи — это отдельная реализация

записи, содержащая конкретные значения ее полей. Каждое пол' базы данных имеет наименование и тип реквизита. Каждому значению поля соответствует реквизит, это либо реквизит-признак, либо реквизит-основание. Каждая запись базы данных содержит один или несколько показателей, в зависимости от PDF created with pdfFactory Pro trial version [www.pdffactory.com](http://www.pdffactory.com) количества в ней реквизитов-оснований, для которых реквизиты-признаки будут одинаковыми, поэтому дублирование записей в базе данных исключается, а поиск по реквизитам признаков запрашиваемой записи производится однозначно. Множество экземпляров записи одной структуры образуют таблицу (table).

Количество записей в таблице может меняться. Ключевым элементом данных называется такой атрибут (или группа атрибутов), который позволяет определить значения других элементов данных. Первичный ключ — это атрибут (или группа атрибутов), который уникальным образом идентифицирует каждый экземпляр объекта (запись). Вторичным ключом называется атрибут (или группа атрибутов), значение которого может повторяться для нескольких записей (экземпляров объекта). Прежде всего вторичные ключи используются в операциях поиска

записей. Процедуры хранения данных в базе должны подчиняться некоторым общим принципам, среди которых в первую очередь следует выделить: • целостность и непротиворечивость данных, под которыми понимается как физическая сохранность данных, так и предотвращения неверного использования данных, поддержка недопустимых сочетаний их значений, защита от структурных искажений и несанкционированного доступа; • минимальная избыточность данных означает, что любой элемент данных должен храниться в базе в единственном виде, что позволяет избежать необходимости дублирования операций, производимых с ним.

2. Модели организации баз данных Набор принципов, определяющих организацию логической структуры хранения данных в базе, получил название модели данных (data model). Модели баз данных определяются тремя компонентами: • допустимой организацией данных; • ограничениями целостности; • множеством допустимых операций; В теории систем управления БД обычно выделяют три основных модели: иерархическую, сетевую и реляционную. Иерархическая модель данных В иерархической модели все записи, агрегаты или атрибуты базы данных образуют иерархическую структуру или

структуру типа «дерева», в которой все элементы связаны на взаимно подчиненном основании и любой элемент может подчиняться только одному какому-нибудь другому элементу. PDF created with pdfFactory Pro trial version [www.pdffactory.com](http://www.pdffactory.com)

К основным понятиям иерархической модели данных относятся уровень, элемент (или узел) и связь. Узлом называется совокупность атрибутов данных, описывающих некоторый объект. Каждый узел связан с одним узлом более высокого уровня и с любым количеством (в том числе и с нулем) узлов нижнего уровня. Исключением является узел самого высокого уровня, который не связан ни с одним узлом более высокого уровня. Этот узел называется корнем дерева. Подчиненные узлы располагаются на втором, третьем и т. д. уровнях. К каждой записи базы данных существует единственный путь от корневой записи. Примером иерархической модели данных может служить адрес. Действительно, на первом уровне (в корне дерева) лежит страна, на втором уровне — регион, на третьем — район, затем город, улица, дом и квартира. Один и тот же город не может принадлежать двум странам или районам. Другим примером иерархической модели является пространство доменных имен Интернет — система DNS.

Сетевая модель данных В сетевой модели, в отличие от



иерархической, возможно любое число предшественников по иерархии. То есть любой элемент нижних уровней иерархии может иметь несколько предшественников. В основе этой модели лежат те же понятия, что и в основе иерархической модели данных, — узел, уровень и связь. Узел может быть связан с любым другим узлом, в том числе лежащим на уровне, не связанном непосредственно с уровнем исходного узла. Простейшим примером сетевой модели данных служит база данных, в которой хранится информация об участии студентов в олимпиаде. Каждый из студентов может участвовать в олимпиадах по разным предметам. В свою очередь, в одной олимпиаде участвуют студенты из разных групп. PDF created with pdfFactory Pro trial version [www.pdffactory.com](http://www.pdffactory.com) В процессе создания и

использования сетевой БД используют набор записей и набор связей между этими записями. Иерархическая и сетевая модели данных имеют недостатки, затрудняющие обработку и доступ к данным. Реляционная модель данных В основе реляционной модели данных лежит идея о том, что любой набор данных можно представить в виде таблиц. Простейшая реляционная база данных может состоять из единственной таблицы, в которой будут храниться все необходимые данные. Для связи между таблицами

используется первичный ключ (key), который позволяет уникально опознавать любую строку таблицы. Ключом называется поле или несколько полей, задание значений которых позволяет однозначно определить значения остальных атрибутов таблицы. Ключ должен состоять из минимального набора атрибутов, это обеспечивает единственность записей. Для обработки данных в реляционных таблицах используют такие категории теории множеств, как объединение, пересечение, разность, декартово произведение и др. Реляционная модель представляет собой некоторое число таблиц, представляющих собой поля (столбцы таблицы), которые задаются при создании структуры таблицы, и записи (строки таблицы), содержащие информацию об объекте системы.

Доступ к конкретному элементу отношения может быть получен с указанием адреса этого элемента в формате  $A[i, j]$ , где  $A$  — элемент данных,  $i$  — строка отношения,  $j$  — номер атрибута отношения. Количество атрибутов в отношении определяет порядок (или степень) этого отношения. Множество всех значений  $A[i, j]$  при постоянном  $i$  и всех возможных  $j$  образует кортеж, или строку таблицы. Множество всех кортежей отношения образует тело отношения. Количество кортежей отношения задает его

мощность (или кардинальное число). Количество кортежей в теле отношения может со временем меняться, а следовательно, будет изменяться и мощность отношения. Поскольку отношения реляционной теории являются специальными математическими двумерными таблицами, в которых по определению не существует двух одинаковых строк, то в отношении не может одновременно существовать двух одинаковых кортежей. PDF created with pdfFactory Pro trial version [www.pdffactory.com](http://www.pdffactory.com) 3.

Типы связей между таблицами. Для работы с таблицами устанавливают правила взаимоотношений между ними. Реляционная база данных обычно состоит из множества таблиц, связанных по определенным критериям. Одна из таблиц — главная, а остальные — зависимые. Такой подход к построению БД позволяет более эффективно управлять данными, исключить дублирование информации и уменьшить объем, который занимают данные. При связывании строк главной и зависимой таблицы реляционной базы данных возможны самые разные сочетания. Тип связи определяет правила сопоставления строк главной и зависимой таблицы. Существуют следующие виды связей. Один-к-одному (One-to-one). При установке отношения «один-к-одному» каждой строке

главной таблицы соответствует единственная (или ни одной) строка зависимой таблицы. С другой стороны, каждая строка зависимой таблицы должна быть связаны только с одной строкой главной таблицы.

В качестве примера такой связи можно привести связь серийного номера двигателя и регистрационного номера автомобиля. Один-многим (One-to-many). В случае связи «один-многим» каждой строке главной таблицы соответствует ноль, одна или более строк зависимой таблицы. С другой стороны, каждая строка зависимой таблицы должна быть связана только с одной строкой главной таблицы. Примером такой связи может являться связь сотрудника с отделом, в котором он работает. Частным случаем связи «один-ко-многим» является связь «один-ко-многим». Многие-ко-многим (Many-to-many). Этот тип связи предполагает, что любой строке главной таблицы может соответствовать ноль, одна или множество строк зависимой таблицы. При этом каждая строка зависимой таблицы может быть связана с одной или более строк главной таблицы. Примером такой связи может являться совместное использование файлов сотрудниками. С каждым файлом могут работать множество сотрудников, в то же время, каждый сотрудник может работать с множеством

файлов. В реляционных базах данных обычно используются связи «один-многим» или частный случай связи «один-к-одному». Для создания связи «многие-многим» требуется создать дополнительную таблицу, содержащую отношение, которое будет использоваться в качестве первичного ключа.

4. Работа в СУБД MS Access Microsoft Access является в настоящее время одной из самых популярных среди современных СУБД. Последняя версия программы входит в состав Microsoft Office

2003. К преимуществам этой программы, делающим ее наиболее привлекательной для современного пользователя, можно отнести универсальный интерфейс, совместимость и возможности интеграции с другими программными продуктами, богатый набор средств для разработки приложений.

Microsoft Access позволяет активно использовать функции встроенного языка программирования Microsoft Visual Basic for Applications. PDF created with pdfFactory Pro trial version [www.pdffactory.com](http://www.pdffactory.com) Встроенные функции для работы с Интернет делают Microsoft Access незаменимой системой ведения баз данных для небольших предприятий. Все объекты, относящиеся к конкретной базе данных, Access хранит в одном большом файле, который имеет расширение

\*.mdb. Для каждой темы отводится отдельная таблица, что позволяет избежать повторения сохраняемых данных. В начале работы над приложением этот файл надо создать. После создания базы появляется окно. Основные разделы главного окна соответствуют типам объектов, которых может содержать база данных Access. Это Таблицы, Запросы, Отчеты, Макросы и Модули. Все этапы работы с программой осуществляются в окнах. Окна базы данных Microsoft Access 2002 выполнены в едином стиле с окнами всех приложений Microsoft Office 2002, что делает работу с ним проще и удобнее. Заголовок окна содержит имя файла базы данных. По каждому объекту предусматриваются стандартные режимы работы: • создать — предназначен для создания структуры объектов; • конструктор — предназначен для изменения структуры объектов; • открыть — предназначен для работы с объектами базы данных. Важным средством, облегчающим работу с Access для начинающих пользователей, являются мастера — специальные программные надстройки, предназначенные для создания объектов БД в режиме последовательного диалога.

Для опытных пользователей существуют возможности более гибкого управления ресурсами и возможностями объектов СУБД в режиме конструктора. База данных Access состоит из набора таблиц, форм, запросов и отчетов, используемых для обработки и представления данных. Пользователи более высокого класса могут использовать макросы и модули. Работа с этими объектами базы данных происходит в окнах. Таблица представляет собой основную единицу хранения данных в базе. Понятие таблицы в Access полностью соответствует аналогичному понятию реляционной модели данных. В произвольной базе обычно имеется совокупность связанных между собой таблиц.

Между двумя таблицами можно устанавливать связи с помощью окна описания схемы данных. Основными операциями над таблицами являются: просмотр и обновление (ввод, модификация и удаление), сортировка, фильтрация и печать. Запрос представляет собой формализованное требование на отбор данных из таблиц или на выполнение определенных действий с данными. Запрос позволяет создать набор записей из данных, находящихся в разных таблицах и использующих его, как источник данных для формы или отчета. В Access можно создавать и выполнять следующие основные типы запросов: на выборку,

обновление, удаление или добавление данных. С помощью запросов можно также создавать новые таблицы, используя данные из одной или нескольких существующих таблиц. Каждый запрос состоит из одного или нескольких критериев, на основе которых создается шаблон или правило для отбора подходящих записей. Данные в каждой записи сравниваются с критерием запроса, и если информация в записи удовлетворяет критерию, то запись включается в таблицу запроса. Любой запрос может содержать несколько критериев, которые используются совместно или альтернативно.

Формы. После создания таблиц и запросов можно создать формы. Форма позволяет отображать, вводить и редактировать данные в удобном для пользователя виде. В Microsoft Access существует несколько способов создания форм.

- Автоформа — автоматическое создание формы с использованием одного из стандартных шаблонов. Это наиболее простой и быстрый способ создания формы.
- Мастер форм — создание формы с помощью мастера (в зависимости от назначения формы Мастер предлагает на выбор стандартные шаблоны и стили оформления).
- Конструктор — создание формы на основе пустого бланка при помощи инструментальных средств конструктора форм.



- Мастер диаграмм — создание формы с диаграммой на основе выбранных полей таблицы. - Мастер сводных таблиц — создание сводной таблицы Microsoft Excel на основе таблиц или запросов Microsoft Access 2000.

- Для обработки готовых форм предназначен конструктор форм. Формы создаются на основе таблицы и запросов. При каждом открытии сохраненной формы обновляются данные запроса, на основе которого создается форма.

Благодаря этому содержимое формы всегда соответствует информации в таблицах и запросах.

Отчеты — это гибкое и эффективное средство для организации данных при выводе на печать. В отчете отражаются результаты запроса или необходимые сведения из таблиц в том виде, в котором это требуется для решения конкретной задачи.

Конструирование отчета не только позволяет вывести информацию в удобном привлекательном виде, но и (по аналогии с формами) комбинировать данные нескольких таблиц, а также упорядочивать записи таблицы на основании других таблиц. Конструирование отчета происходит в режиме Конструктора отчетов. Страницы доступа к данным

представляют собой специальный тип Webстраниц, предназначенный для просмотра и работы через Интернет или интрасеть с данными, хранящимися в базах данных Microsoft SQL Server. Страница доступа к данным может также включать данные из других источников, таких как Microsoft Excel. Такие страницы разрабатываются в режиме конструктора страниц в Microsoft Access. Страница представляет собой отдельный файл, хранящийся за пределами Microsoft Access; однако при создании этого файла Microsoft Access автоматически добавляет ярлык к нему в окно базы данных. Разработка страниц доступа к данным аналогична разработке форм и отчетов: используются список полей, панель элементов, элементы управления, диалоговое окно Сортировка и группировка и т.д. Макросы и модули служат для разработки программных приложений, ориентированных на конечных пользователей. Эти средства базируются на инструментальных возможностях языка приложений SQL и программных средств Visual Basic.

#### 4.Создание таблиц в MS ACCESS

Основные понятия В широком смысле слова база данных – это совокупность сведений о конкретных объектах реального мира в какой-либо области. Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, только если данные структурированы. Структурирование – это введение соглашений о способах представления данных. Неструктурированными называют данные, записанные, например, в текстовом файле. Итак, база данных (БД) – это поименованная совокупность структурированных данных, относящихся к определенной предметной области. Система управления базами данных (СУБД) – это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации. Модели данных В СУБД происходит процесс логического представления данных, с которыми имеет дело пользователь, в язык ЭВМ и обратно. Общая логическая структура БД называется моделью данных. Различают три основные модели данных:

иерархическую, сетевую и реляционную. Иерархическая модель данных Иерархическая структура представляет совокупность элементов, образующих граф (дерево). К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. Узел – это совокупность атрибутов данных, описывающих некоторые объекты. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне.

Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей. К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рисунка 1, для записи С4 путь проходит через записи А и В3. Рис.1. Иерархическая модель данных Рис. 2. Фрагмент иерархическая базы данных «Структура института» Сетевая модель данных В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим

элементом (рис. 3). Рис. 3. Сетевая модель данных  
Рассмотрим фрагмент сетевой БД «Курсовые работы студентов» (рис. 4). Много студентов, много предметов.

Любой элемент структуры может быть связан с другими.  
Рис. 4. Фрагмент сетевой БД «Курсовые работы студентов»

Реляционная модель данных Реляционные модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением данных и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных. Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами: · Каждый элемент таблицы – один элемент данных; · Все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.); · Каждый столбец имеет уникальное имя. Примером реляционной БД может служить БД «Студент» (табл. 1). Таблица 1 № личного дела  
Фамилия Имя Отчество Дата рождения  
34876 Кондратьев Алексей Иванович 02.12.78  
34877 Лаптев Михаил Петрович 03.09.79  
34878 Миронова Анастасия Григорьевна 08.05.78

## СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ACCESS

В любой области деятельности часто приходится иметь дело с большими объёмами данных. Основными операциями при этом являются сбор информации, её обработка (поиск требуемых данных, сортировка и т.п.), создание форм для просмотра и распечатки данных. Одним из самых популярных программных продуктов, обеспечивающих все эти функции, признана система управления базами данных фирмы Microsoft Access. Access – это система управления базами данных (СУБД). Под системой управления понимается комплекс программ, который позволяет не только хранить большие массивы данных в определённом формате, но и обрабатывать их, представляя в удобном для пользователей виде. Access даёт возможность также автоматизировать часто выполняемые операции (например, расчет заработной платы, учёт материальных ценностей и т.п.). С помощью Access можно не только разрабатывать удобные формы ввода и просмотра данных, но и составлять сложные отчеты. Access – это реляционная

СУБД. Это означает, что с её помощью можно работать одновременно с несколькими таблицами базы данных.

Применение реляционной СУБД помогает упростить структуру данных и таким образом облегчить выполнение работы.

Таблицу Access можно связать с данными, хранящимися на другом компьютере или на сервере, а также использовать таблицу, созданную в СУБД Paradox или Dbase. Данные Access очень просто комбинировать с данными Excel. СУБД Access работает под управлением Windows, таким образом все преимущества Windows доступны в Access

(например, вы можете копировать, вырезать, вставлять данные из любого приложения Windows в приложение Access и наоборот). В СУБД Access предусмотрено много дополнительных сервисных возможностей. Мастера помогут вам создавать таблицы, формы или отчеты из имеющихся заготовок, сделав за вас черновую работу. Выражения используются в Access, например для проверки допустимости введенного значения. Макросы позволяют автоматизировать многие процессы без программирования, так как встроенный в Access язык VBA (Visual Basic for Applications) даёт возможность опытному пользователю программировать сложные процедуры обработки данных.

Просматривая свою форму или отчет, вы сможете представить как они будут выглядеть в распечатанном виде. Используя такие возможности языка программирования C, как функции и обращения к Windows API (Application Programming Interface – интерфейс прикладных программ Windows), можно написать подпрограмму для взаимодействия Access с другими приложениями – источниками данных.

В Microsoft Access добавлено множество новых средств, разработанных для облегчения работы в Интернет и создания приложений для Web. Запуск Microsoft Access осуществляется: · Пуск – Главное меню – Программы – Microsoft Access · С помощью

Проводника · Запуск с помощью ярлыка, находящегося на Рабочем столе. Создание однотабличной базы данных Структура таблицы и типы данных Все составляющие базы данных, такие, как таблицы, отчеты, запросы, формы и объекты, в Access хранятся в едином дисковом файле. Основным структурным компонентом базы данных является таблица. В таблицах хранятся вводимые нами данные. Внешне каждая таблица Access похожа на таблицы, с которыми мы привыкли работать на бумаге, – она состоит из



столбцов, называемых полями, и строк, называемых записями. Каждая запись таблицы содержит всю необходимую информацию об отдельном элементе базы данных. При разработке структуры таблицы прежде всего необходимо определить названия полей, из которых она должна состоять, типы полей и их размеры. Каждому полю таблицы присваивается уникальное имя, которое не может содержать более 64 символов. Имя желательно делать таким, чтобы функция поля узнавалась по его имени. Далее надо решить, данные какого типа будут содержаться в каждом поле.

В Access можно выбирать любые из основных типов данных. Один из этих типов данных должен быть присвоен каждому полю. Значение типа поля может быть задано только в режиме конструктора. Тип данных Описание  
Текстовый (значение по умолчанию) Текст или числа, не требующие проведения расчетов, например номера телефонов (до 255 знаков)  
Числовой Числовые данные различных форматов, используемые для проведения расчетов  
Дата/Время Для хранения информации о дате и времени с 100 по 9999 год включительно  
Денежный Денежные значения и числовые данные, используемые в математических расчетах, проводящихся с точностью до 15

знаков в целой и до 4 знаков в дробной части Поле MEMO Для хранения комментариев; до 65535 символов Счетчик Специальное числовое поле, в котором Access автоматически присваивает уникальный порядковый номер каждой записи. Значения полей типа счетчика обновлять нельзя Логический Может иметь только одно из двух возможных значений (True/False, Да/Нет) Поле объекта OLE Объект (например, электронная таблица Microsoft Excel, документ Microsoft Word, рисунок, звукозапись или другие данные в двоичном формате), связанный или внедрённый в таблицу Access Гиперссылка Строка, состоящая из букв и цифр и представляющая адрес гиперссылки. Адрес гиперссылки может состоять максимум из трёх частей: текст, выводимый в поле или в элементе управления; путь к файлу (в формате пути UNC) или к странице (адрес URL). Чтобы вставить адрес гиперссылки в поле или в элемент управления, выполните команду Вставка, Гиперссылка Мастер подстановок Создаёт поле, в котором предполагается выбор значений из списка или из поля со списком, содержащего набор постоянных значений или значений из другой таблицы.

Это в действительности не тип поля, а способ хранения поля В Access существует четыре способа создания пустой таблицы:

1. Использование мастера баз данных для создания всей базы данных, содержащей все требуемые отчёты, таблицы и формы, за одну операцию. Мастер баз данных создаёт новую базу данных, его нельзя использовать для добавления новых таблиц, форм, отчетов в уже существующую базу данных;
2. Мастер таблиц позволяет выбрать поля для данной таблицы из множества определённых ранее таблиц, таких, как деловые контакты, список личного имущества или рецепты;
3. Ввод данных непосредственно в пустую таблицу в режиме таблицы. При сохранении новой таблицы в Access данные анализируются, и каждому полю присваивается необходимый тип данных и формат;
4. Определение всех параметров макета таблицы в режиме конструктора. Независимо от метода, применённого для создания таблицы, всегда имеется возможность использовать режим конструктора для дальнейшего изменения макета таблицы, например

для добавления новых полей, установки значений по умолчанию или для создания масок ввода. Однако только четвертый метод позволяет сразу задать ту структуру таблицы, которая вам нужна. Ввод и редактирование данных в ячейках таблицы Ввод данных в ячейки таблицы производится обычным способом – вы переводите курсор в нужную ячейку, т.е. делаете её активной, и вводите в неё данные.

Особенности ввода следующие:

1. При нажатии клавиши [Del] ячейка очищается;
2. Если ввод данных в ячейку прервать, нажав клавишу [Esc], то восстанавливается старое значение;
3. Если нажать клавиши [Enter] или [Tab], то в ячейку заносится новое значение.

Можно редактировать текущее значение, меняя в нём отдельные символы. Для этого надо в ячейке таблицы дважды щелкнуть мышью или нажать клавишу [F2]. Редактирование данных осуществляется следующим образом: курсор переводится в нужную ячейку, старые

данные удаляются ([Del], [Backspace]) и вводятся новые данные. Удалять можно не только данные в ячейках, но и целиком строки, предварительно их выделив. Но если таблица большая, то редактируемые данные надо сначала найти. Это делается следующим образом: курсор переводят в любую ячейку поля, начиная с которой будет вестись поиск, затем открывается меню Правка – Найти. В появившемся окне вводят образец искомых данных и щелкают по кнопке Найти. Если значение найдено, курсор перейдет в эту ячейку. Иногда требуется большое количество одинаковых данных заменить на другое значение. Для этого надо открыть меню Правка – Заменить. В появившемся диалоговом окне ввести образцы того, что надо найти и на что заменить. Затем надо решить, будете ли вы просматривать каждое заменяемое значение или нет. Если будете, то надо щелкнуть по кнопке Заменить, а если нет – один раз щелкнуть по кнопке Заменить всё. Для всех типов полей (кроме типов Счётчик и поле объекта OLE) можно самостоятельно задать ограничения для вводимых данных. Для этого в режиме конструктора надо выбрать вкладку Общие, перевести курсор в поле с именем «Условия на значение» и ввести ограничение на данные. Например, стаж работы вряд ли может быть более 70 лет, и ограничение

будет следующее: <70. Можно использовать ещё один удобный инструмент при вводе данных – параметр Значение по умолчанию (находится также на вкладке Общие). Здесь можно задать данные, которые Access будет вводить по умолчанию при заполнении таблицы. Это удобно использовать, когда большинство значений данного поля одинаковы и лишь некоторые отличаются.

Например, должность большинства преподавателей – доцент. Если ввести это слово в строку параметра Значение по умолчанию, то все ячейки поля «Должность» примут значение Доцент и надо будет лишь изменить значения ячеек для преподавателей с другими должностями. Данные можно вводить и копированием их из одной ячейки в другую стандартными средствами Windows. Для каждого типа данных (кроме типа данных Счётчик) предусмотрено пустое (нулевое) значение. Различают два типа пустых значений: пустые (Null) значения и пустые строки. Если пустое поле имеет пустое (Null) значение, то это означает, что данные для него существуют, но пока не известны. Если же введена пустая строка (два знака прямых кавычек (“”)), то это означает, что данные не существуют вовсе. Access позволяет обрабатывать такие пустые значения. Сортировка данных Для удобства просмотра можно сортировать записи в

таблице в определенной последовательности. Кнопки сортировки на панели инструментов (лил команды меню Записи – Сортировка – Сортировка по возрастанию (сортировка по убыванию)) позволяют сортировать столбцы по возрастанию или по убыванию. Если вы хотите восстановить порядок отображения записей, используйте команду Записи – Удалить фильтр.

### **Отбор данных с помощью фильтра**

Фильтр – это набор условий, применяемых для отбора подмножества записей. В Access 97 существуют фильтры четырёх типов: · Фильтр по выделенному фрагменту · Обычный фильтр · Расширенный фильтр · Фильтр по вводу. Фильтр по выделенному фрагменту – это способ быстрого отбора записей по выделенному образцу. Выделяете нужную запись, щелкаете по кнопке Фильтр по выделенному (или Записи – Фильтр – Фильтр по выделенному). Если вы хотите просмотреть значения, которые не удовлетворяют заданному критерию, щелкните правой кнопкой мыши (курсор мыши должен находиться внутри таблицы) и выберите команду Исключить выделенное. Будут выбраны все записи, кроме недавно выбранных (инверсный выбор). Фильтрация

данных в Access производится с помощью кнопок **Фильтр по выделенному** или **Изменить фильтр** (меню **Записи – Фильтр – Изменить фильтр**). После нажатия второй кнопки от таблицы остаётся одна запись. Каждое поле становится полем со списком (когда в нем находится курсор), в котором можно выбрать из списка все значения для данного поля. После щелчка по кнопке **Применить фильтр** будут выбраны записи, соответствующие измененному фильтру.

Ещё более сложные условия фильтрации можно задать командой меню **Записи – Фильтр – Расширенный фильтр...**

**Работа с формами** **Форма – объект**, в основном предназначенный для удобного ввода данных. **Форма – это формат (бланк, шаблон) показа данных на экране компьютера.** В форму могут быть внедрены рисунки, диаграммы, аудио- и видеоизображения. **Режимы работы с формой:**

1. **Режим форм.** Используется для просмотра и редактирования данных. Режим форм предоставляет дружественную среду для работы с данными и удобный дизайн их представления на экране.
2. **Режим конструктора форм.** Используется для изменения структуры или шаблона формы.
3. **Режим таблицы.** Режим позволяет увидеть таблицу, включающую все поля форм; чтобы переключиться



в этот режим при работе с формой, надо нажать кнопку таблицы на панели инструментов.

Наиболее удобным и гибким способом создания форм является Мастер форм. В этом режиме вы можете выбрать поля таблицы для отображения в форме, стиль и цвет оформления фона и ячеек, а также вид формы. Мастер форм предлагает четыре вида формы представления данных: · в один столбец, · ленточная, · табличная, · выровненная. Настраивать внешний вид и возможности ввода, обработки и просмотра данных можно в режиме конструктора. Вы можете использовать огромное количество

Инструментов и Свойств формы.

Для манипулирования записями (операции поиска, замены, сортировки и фильтрации данных) в режиме формы используются те же способы, что и в режиме таблицы. Запросы Запросы служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразование данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными,

импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое. Запросы можно создать самостоятельно и с помощью мастеров. Мастера запросов автоматически выполняют основные действия в зависимости от ответов пользователя на поставленные вопросы.

Самостоятельно разработать запросы можно в режиме конструктора.

Можно создать следующие типы запросов:

- 1.запрос на выборку;
- 2.запрос с параметрами;
- 3.перекрёстный запрос;
- 4.запрос на изменение (запрос на удаление, обновление и добавление записей на создание таблицы);
- 5.запросы SQL (запросы на объединение, запросы к серверу, управляющие запросы, подчиненные запросы). Запрос на выборку используется наиболее часто. При его выполнении данные, удовлетворяющие условиям отбора, выбираются из одной или нескольких таблиц и выводятся в определенном порядке.

Можно также использовать запрос на выборку, чтобы сгруппировать записи для вычисления сумм, средних значений, пересчета и других действий. Запрос с параметрами – это запрос, при выполнении которого в его диалоговом окне пользователю выдаётся приглашение ввести данные, на основе которых будет выполняться запрос.

Итоговые запросы, назначение которых отдалённо напоминает итоговые функции электронных таблиц (производят математические вычисления по заданному полю и выдают результат). Запросы на изменение позволяют автоматизировать заполнение полей таблиц. Перекрёстные запросы, позволяющие создавать результирующие таблицы на основе результатов расчетов, полученных при анализе группы таблиц. Специфические запросы SQL – запросы к серверу базы данных, написанные на языке запросов SQL. Упорядочение записей в результирующей таблице. Если необходимо, чтобы данные отобранные в результате работы запроса на выборку, были упорядочены по какому-либо полю, применяют сортировку. В нижней части бланка имеется специальная строка Сортировка. При щелчке на этой строке открывается кнопка раскрывающегося списка, в

котором можно выбрать метод сортировки: по возрастанию или по убыванию.

В результирующей таблице данные будут отсортированы по тому полю, для которого задан порядок сортировки. Управление отображением данных в результирующей таблице. В нижней части бланка запроса по образцу (Запроса на выборку) имеется строка Вывод на экран. По умолчанию предполагается, что все поля включенные в запрос, должны выводиться на экран, что не всегда целесообразно. Тогда отображение содержимого на экране подавляют сбросом флажка Вывод на экран. Использование условия отбора.

Дополнительным средством, обеспечивающим отбор данных по заданному критерию, является так называемое Условие отбора. Соответствующая строка имеется в нижней части бланка запроса по образцу. Для каждого поля в этой строке можно задать индивидуальное условие. При сохранении запроса остаётся только структура запроса – перечень таблиц, список полей, порядок сортировки, ограничения на записи, тип запроса и т.д.

При сохранении в базе данных запрос, по сравнению с результирующим набором данных, имеет ряд преимуществ:

1. на физическом носителе информации (обычно это жесткий диск) требуется меньший объём пространства;
2. запрос может использовать обновлённые версии любых записей, изменённых со времени последнего запуска запроса.

При каждом выполнении запрос обращается к базовым та блицам и снова создаёт результирующий набор данных. Поскольку сам по себе результирующий набор данных не сохраняется, запрос автоматически отображает любые изменения, произошедшие в базовых таблицах с момента последнего запуска этого запроса (даже в реальном времени в многопользовательской среде). Для сохранения запроса следует выполнить следующее: Файл – Сохранить. Если вы впервые сохраняете запрос, введите новое имя запроса в диалоговом окне Сохранение. Работа с отчётами Отчёты служат для форматированного вывода данных на печатающие устройства и, соответственно, при этом должны учитывать параметры принтера и параметры используемой бумаги. В отчете можно получить результаты сложных расчетов, статистических сравнений, а также поместить в него рисунки и диаграммы. Access 2016 Для создания отчета

надо открыть вкладку Отчеты и щелкнуть по кнопке Создать. Откроется окно Новый отчет, в котором приведены шесть пунктов меню, т.е. шесть способов создания отчета:

1. Конструктор
2. Мастер отчетов
3. Автоотчёт в столбец
4. Автоотчёт ленточный
5. Мастер диаграмм
6. Почтовые наклейки

Конструктор позволит вам самостоятельно создать отчет. Мастер отчетов автоматически создаст отчет на основе выбранных вами полей таблиц (запросов) и макетов отчетов. Этот способ является наиболее удобным как для начинающих, так и для опытных пользователей. Автоотчёт в столбец и Автоотчёт ленточный – самые простейшие способы создания отчетов: достаточно указать только имя таблицы (запроса), на основе которого будет создан отчёт, а остальное сделает мастер отчётов. Мастер диаграмм поможет создать отчет в виде диаграммы, а Почтовые наклейки создадут отчет, отформатированный для печати почтовых наклеек.

Access 2016 Средства автоматического проектирования реализованы автоотчётами (База данных – Создать – Новый отчет – Автоотчёт в столбец). Кроме автоотчётов «в столбец» существуют «ленточные» автоотчёты. Средством автоматизированного создания отчётов является Мастер отчетов. Он запускается двойным щелчком на значке Создание отчета с помощью мастера в окне База данных. Мастер отчетов работает в шесть этапов. При его работе выполняется выбор базовых таблиц или запросов, на которых отчет базируется, выбор полей, отображаемых в отчете, выбор полей группировки, выбор полей и методов сортировки, выбор формы печатного макета и стиля оформления. Структура готового отчета отличается от структуры формы только увеличенным количеством разделов. Кроме разделов заголовка, примечания и данных, отчет может содержать разделы верхнего и нижнего колонтитулов. Редактирование структуры отчета выполняют в режиме Конструктора (режим запускается кнопкой Конструктор в окне База данных). Приёмы редактирования те же, что и для форм. Элементы управления в данном случае выполняют функции элементов оформления, поскольку печатный отчёт не интерактивный объект, в отличие от электронных форм и Web-страниц.

Размещение элементов управления выполняют с помощью Вид – Панели элементов, которая по составу практически не отличается от Панели элементов формы. Важной особенностью отчетов является наличие средства для вставки в область верхнего и нижнего колонтитула текущего номера страницы и полного количества страниц. Эту операцию выполняют с помощью диалогового окна Номера страниц (Вставка – Номера страниц). Страницы доступа к данным Это специальные объекты баз данных, реализованные в последней версии СУБД Access 2016. Физически это особый объект, выполненный в коде HTML, размещаемый на Web-странице и передаваемый клиенту вместе с ней. Сам по себе этот объект не является базой данных, но содержит компоненты, через которые осуществляется связь переданной Web-страницы с базой данных, остающейся на сервере. Пользуясь этими компонентами, посетитель Web-узла может просматривать записи базы в полях страницы доступа. Таким образом, страницы доступа к данным осуществляют интерфейс между клиентом, сервером и базой данных, размещенной на сервере. Эта база данных не обязательно должна быть базой данных Microsoft Access.



Страницы доступа, созданные средствами Microsoft Access, позволяют работать также с базами данных Microsoft SQL Server.

#### **4.1.Макросы и модули**

Эти категории объектов предназначены как для автоматизации повторяющихся операций при работе с системой управления базами данных, так и для создания новых функций путём программирования. В СУБД Microsoft Access макросы состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой. Макрос – это объект, представляющий структурированное описание одного или нескольких действий, которые должен выполнить Access в ответ на определённое событие. Например, можно определить макрос, который в ответ на выбор некоторого элемента в основной форме открывает другую форму. В макросы включаются макрокоманды. В Microsoft Access имеется свыше 40 макрокоманд. Макрокоманды выполняют такие действия, как открытие таблиц и форм, выполнение запросов, запуск других макросов, выбор опций из меню, изменение размеров открытых окон и т.д.

Модули создаются средствами внешнего языка программирования, в данном случае языка Visual Basic for Applications. Это одно из средств, с помощью которых разработчик базы может заложить в неё нестандартные функциональные возможности, удовлетворить специфические требования заказчика, повысить быстродействие системы управления, а также уровень её защищённости. Подготовка к печати Записи таблицы можно распечатать. Простейший способ сделать это – выбрать команду Файл – Печать или щелкнуть по кнопке Печать на панели инструментов. Для распечатки потребуется столько страниц, сколько необходимо для размещения всех данных. Access автоматически подгоняет распечатку на каждой странице. Управлять распечатыванием можно из диалогового окна Печать, выбирая представленные ниже опции: Опция Назначение Печатать Копии Разобрать Распечатывает либо всю таблицу, либо только выбранные страницы Определяет число копий Определяет, будет ли распечатка разобрана по копиям С помощью кнопки Свойства можно вывести диалоговое окно настройки принтера. Кнопка Настройка позволит определить поля и заголовки. Перед печатью всегда необходимо просмотреть

данные для того, чтобы определить, как будет выглядеть таблица на распечатанной странице.

Для этого щелкните по кнопке Предварительный просмотр. При этом появится окно предварительного просмотра, в котором можно управлять параметрами просмотра. Разработка инфологической модели и создание структуры реляционной базы данных Слово «реляционная» происходит от английского relation – отношение. Отношение – математическое понятие, но в терминологии моделей данных отношения удобно изображать в виде таблицы. При этом строки таблицы соответствуют кортежам отношения, а столбцы – атрибутам. Ключом называют любую функцию от атрибутов кортежа, которая может быть использована для идентификации кортежа. Такая функция может быть значением одного из атрибутов (простой ключ), задаваться алгебраическим выражением, включающим значения нескольких атрибутов (составной ключ). Это означает, что данные в строках каждого из столбцов составного ключа могут повторяться, но комбинация данных каждой строки этих столбцов является уникальной. В локализованной (русифицированной) версии Access вводится термин ключевое поле, которое можно трактовать как первичный ключ. Одно из важнейших достоинств реляционных баз

данных состоит в том, что вы можете хранить логически сгруппированные данные в разных таблицах и задавать связи между ними, объединяя их в единую базу. Для задания связи таблицы должны иметь поля с одинаковыми именами или хотя бы с одинаковыми форматами данных. Связь между таблицами устанавливает отношения между совпадающими значениями в этих полях. Такая организация данных позволяет уменьшить избыточность хранимых данных, упрощает их ввод и организацию запросов и отчетов. В Access используются три типа межтабличных связей:

1. Связь Один-ко-многим –наиболее часто используемый тип связи между таблицами. В такой связи каждой записи в таблице А может соответствовать несколько записей в таблице В (поля с этими записями называют внешними ключами), а запись в таблице В не может иметь более одной соответствующей ей записи в таблице А.
2. При связи Многие-ко-многим одной записи в таблице А может соответствовать несколько записей в таблице В, в одной записи в таблице В – несколько записей в таблице А. Такая схема реализуется только

с помощью третьей (связующей) таблицы, ключ которой состоит по крайней мере из двух полей, одно из которых является общим с таблицей А, а другое – общим с таблицей В.

3. При связи Один-к-одному запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением Одинк-одному применяют для разделения очень широких таблиц, для отделения части таблицы в целях её защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

Тип создаваемой связи зависит от полей, для которых определяется связь:

1. Связь Один-ко-многим создаётся в том случае, когда только одно из полей является ключевым или имеет уникальный индекс, т.е. значения в нём не повторяются;

2. Связь Один-к-одному создаётся в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы;
3. Связь Многие-ко-многим фактически представляет две связи типа Один-ко-многим через третью таблицу, ключ которой состоит по крайней мере из двух полей, общих для двух других таблиц.
4. Целостность данных Целостность данных означает систему правил, используемых в СУБД Access для поддержания связей между записями в связанных таблицах, а также обеспечивает защиту от случайного удаления или изменения данных связанных данных.

Контролировать целостность данных можно, если выполнены следующие условия:

1. Связанное поле (поле, посредством которого осуществляется связь) одной таблицы является ключевым полем или имеет уникальный индекс;
2. Связанные поля имеют один тип данных. Здесь существует исключение. Поле счётчика может быть связано с числовым полем, если оно имеет тип Длинное целое;

3. Обе таблицы принадлежат одной базе данных Access. Если таблицы являются связанными, то они должны быть таблицами Access. Для установки целостности данных база данных, в которой находятся таблицы, должна быть открыта. Связь между таблицами позволяет:
  1. Либо исключит возможность удаления или изменения данных в ключевом поле главной таблицы, если с этим полем связаны какие-либо поля других таблиц;
  2. Либо сделать так, что при удалении (или изменении) данных в ключевом поле главной таблицы автоматически (и абсолютно корректно) произойдёт удаление или изменение соответствующих данных в полях связанных таблиц.

Для настройки свойств связи надо в окне Схема данных выделить линию, соединяющую поля двух таблиц, щелкнуть на ней правой кнопкой мыши и открыть контекстное меню связи, после чего выбрать в нем пункт Изменит связь – откроется диалоговое окно Изменение связи. В нем показаны названия связанных таблиц и имена полей, участвующих в связи (здесь их можно изменить), а также приведены

элементы управления для обеспечения условий целостности данных. Если установлен только флажок Обеспечение целостности данных, то удалять данные из ключевого поля главной таблицы нельзя.

### **Если вместе с ним включены флажки**

Каскадное обновление связанных полей и Каскадное удаление связанных записей, то, соответственно, операции редактирования и удаления данных в ключевом поле главной таблицы разрешены, но сопровождаются автоматическими изменениями в связанной таблице. Таким образом, смысл создания реляционных связей между таблицами состоит, с одной стороны, в защите данных, а с другой стороны – в автоматизации внесения изменений сразу в несколько таблиц при изменениях в одной таблице.

Завершение работы с Access Завершив работу с Access (или с её приложением), надо корректно закончить сеанс. Безопасно выйти из Access можно несколькими способами:

1. Двойным щелчком мыши на пиктограмме системного меню в строке заголовка главного окна Access;



2. Из меню Access выбором пункта Файл – Выход; §  
Нажатием комбинации клавиш [ALT+F4].

### **Контрольные вопросы.**

1. Что такое СУБД?
2. Запуск СУБД Access.
3. Назовите основные элементы окна СУБД Access.
4. Перечислить и охарактеризовать основные объекты окна базы данных.
5. Структура таблицы и типы данных.
6. Ввод и редактирование данных в таблице.
7. Сортировка данных в Access.
8. Фильтрация данных в Access.
9. Какие режимы работы используются для работы с таблицей, формой, отчетом?
10. Для чего нужен запрос?
11. Перечислить и охарактеризовать типы межтабличных связей в СУБД Access?
12. Завершение работы с Access.

### **Задание на самостоятельную работу.**

**Вариант 1** Создайте базу данных «Музыкальный альбом» из двух таблиц. Самостоятельно придумайте поля для этих двух таблиц, выделите ключевые поля, установите связи между таблицами. Заполните данными и проведите редактирование записей таблиц.

Выполнить 2-3 запроса, сделать отчет, формы по таблицам.

**Вариант 2.** Создайте базу данных «Химическая лаборатория». Самостоятельно придумайте поля для таблиц, выделите ключевые поля, установите связи между таблицами. Заполните данными и проведите редактирование записей таблиц.

Выполнить 2-3 запроса, сделать отчет, формы по таблицам.

**Вариант 3.** Создайте базу данных «Компьютерный салон». Самостоятельно придумайте поля для таблиц, выделите ключевые поля, установите связи между таблицами. Заполните данными и проведите редактирование

записей таблиц. Выполнить 2-3 запроса, сделать отчет, формы по таблицам.

**Вариант 4.** Создайте базу данных «Кулинарные рецепты». Самостоятельно придумайте поля для таблиц, выделите ключевые поля, установите связи между таблицами. Заполните данными и проведите редактирование записей таблиц. Выполнить 2-3 запроса, сделать отчет, формы по таблицам.

**Вариант 5.** Создайте базу данных «Салон красоты». Самостоятельно придумайте поля для таблиц, выделите ключевые поля, установите связи между таблицами. Заполните данными и проведите редактирование записей таблиц. Выполнить 2-3 запроса, сделать отчет, формы по таблицам.

**Вариант 6.** Создайте базу данных «Автосервис». Самостоятельно придумайте поля для таблиц, выделите ключевые поля, установите связи между таблицами. Заполните данными и проведите редактирование записей таблиц. Выполнить 2-3 запроса, сделать отчет, формы по таблицам.

**Вариант 7.** Создайте базу данных «Книжный магазин». Самостоятельно придумайте поля для таблиц, выделите ключевые поля, установите связи между таблицами. Заполните данными и проведите редактирование записей таблиц. Выполнить 2-3 запроса, сделать отчет, формы по таблицам.

**Вариант 8.** Создайте базу данных «Химический склад». Самостоятельно придумайте поля для таблиц, выделите ключевые поля, установите связи между таблицами. Заполните данными и проведите редактирование записей таблиц. Выполнить 2-3 запроса, сделать отчет, формы по таблицам.

## **4.2. Создание структур в таблицах**

СУБД Access оперирует реляционными базами данных. На этапе проектирования базы данных должна быть определена логическая структура базы данных для выбранной предметной области. Проект логической структуры БД устанавливает состав реляционных таблиц, их структуру и логические связи между таблицами. При формировании структуры каждой таблицы определяется совокупность атрибутов (полей, столбцов), для каждого из

которых даются описание типа, размера данных и других свойств. Кроме того, должен быть указан уникальный ключ таблицы, который может состоять из одного или нескольких полей.

При проектировании базы данных, отвечающей требованиям нормализации, между таблицами определяются логические связи типа 1 : М. Такие связи позволят осуществлять в Access автоматическое

поддержание связной целостности и непротиворечивости данных в базе.

### **4.3. Этапы проектирования и создания базы данных**

Для проектирования базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, определять все необходимые источники информации для обеспечения предполагаемых запросов пользователя и решаемых в приложении задач.

Определение состава и структуры данных, которые должны быть загружены в базу данных, осуществляется на основе анализа предметной области. Структура данных

предметной области может отображаться информационно логической моделью (ИЛМ). Если при построении такой модели обеспечены требования нормализации данных и она, соответственно, представлена в каноническом виде, далее легко определяется проект логической структуры нормализованной базы данных. На основе канонической модели можно создать реляционную базу без дублирования данных.

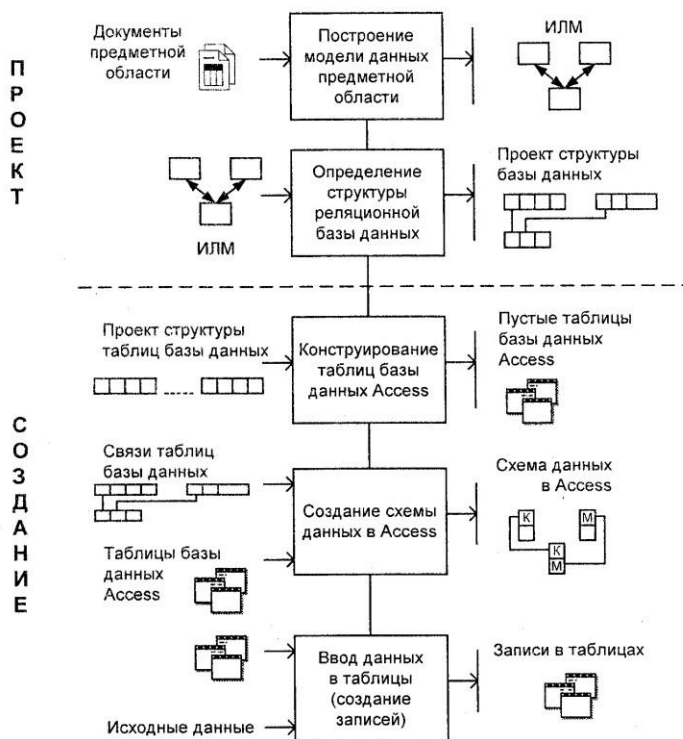
При разработке модели данных предметной области могут использоваться два подхода. В первом подходе (аналитическом или процессном) сначала формулируются основные задачи, для решения которых строится база, выявляются информационные потребности задач приложения пользователя, и, соответственно, определяются состав и структура информационных объектов модели, а также связи между ними. При втором подходе (интуитивном) сразу устанавливаются типовые объекты предметной области и их взаимосвязи. Наиболее рационально сочетание обоих подходов. Это связано с тем, что на начальном этапе, как правило, нет исчерпывающих сведений обо всех задачах. Использование такой технологии тем более оправдано, что гибкие средства создания

реляционной базы данных в Access позволяют на любом этапе разработки внести изменения в базу данных и модифицировать ее структуру без ущерба для введенных ранее данных.

При определении проекта логической структуры реляционной базы данных каждый информационный объект канонической модели предметной области адекватно отображается реляционной таблицей, а связям между двумя информационными объектами соответствуют логические связи между парой соответствующих таблиц. Такие связи устанавливаются по уникальному ключу одной из этих таблиц, которая является главной в связи. Во второй таблице, которая является подчиненной, поле связи может быть либо частью ее уникального ключа, либо неключевым.

В процессе создания базы данных на компьютере сначала осуществляется конструирование ее таблиц средствами Access. Далее создается схема данных, в которой устанавливаются логические связи таблиц. В схеме данных базы могут быть заданы параметры поддержания связной целостности данных, если модель данных была разработана в соответствии с требованиями нормализации.

Информационно-логическая модель (ИЛМ) отображает данные предметной области в виде совокупности информационных объектов (ИО) и связей между ними. Эта модель представляет данные, подлежащие хранению в базе данных. Каждый информационный объект в модели данных должен иметь уникальное имя.



Для определения поля таблицы задаются **Имя поля** (Field Name), **Тип данных** (Data Type), **Описание**



(Description) — краткий комментарий, а также свойства поля в разделе **Свойства поля** (Field Properties), представленные на двух вкладках: **Общие** (General) и **Подстановка** (Lookup). К общим относятся такие свойства поля, как максимальный размер, формат, подпись, которая выводится в заголовке столбца таблицы, значение по умолчанию и др. На вкладке **Подстановка** (Lookup) выбирается Тип элемента управления (Display Control): поле, список фиксированных значений или поле со списком.

#### 4.4. Имена полей и типы данных

**Имя поля** (Field Name). Каждое поле в таблице должно иметь уникальное имя, удовлетворяющее соглашениям об именах объектов в Access. Оно является комбинацией из букв, цифр, пробелов и специальных символов, за исключением точки (.), восклицательного знака (!), надстрочного знака (~) и квадратных скобок ([ ]).

Имя не может начинаться с пробела и содержать управляющие символы с кодами ASCII от 0 до 31.

Максимальная длина имени 64 символа.

**Тип данных** (Data Type), Тип данных определяется значениями, которые предполагается хранить в поле, и

операциями, которые будут выполняться с этими значениями. В Access допускается использование десяти типов данных и мастера подстановок. Список возможных типов данных вызывается нажатием кнопки списка при выборе типа данных каждого поля. Рассмотрим вкратце назначение и допустимые размеры всех типов данных, которые могут назначаться полям таблицы в Access.

**Текстовый (Text)** — используется для хранения текста или комбинаций алфавитно-цифровых знаков, не применяемых в расчетах (например, код товара). Максимальная длина поля 255 знаков.

**Поле МЕМО (Memo)** — используется для хранения обычного текста или комбинаций алфавитно-цифровых знаков длиной более 255 знаков. Поля с этим типом данных в базах данных формата Access 2016 поддерживают также форматирование текста. Это единственный в Access тип данных, обеспечивающий встроенную поддержку отображения и хранения форматированного текста. Максимальный размер поля 1 Гбайт знаков или 2 Гбайт памяти (2 байта на знак) при программном заполнении полей, и 65 535 знаков при вводе данных вручную в поле и в любой элемент управления, связанный с этим полем.

**Числовой** (Number) — служит для хранения числовых значений (целых или дробных), предназначенных для вычислений, исключением являются денежные значения, для которых используется тип данных **Денежный**. Размер поля 1, 2, 4 и 8 байтов, или 16 байтов (если используется для кода репликации) зависит от типа чисел, вводимых в поле.

**Дата/время** (Date/Time) — используется для хранения значений даты и времени в виде 8-байтовых чисел двойной точности с плавающей запятой. Целая часть значения, расположенная слева от десятичной запятой, представляет собой дату. Дробная часть, расположенная справа от десятичной запятой,— это время. Хранение значений даты и времени в числовом формате позволяет выполнять различные вычисления с этими данными.

**Денежный** (Currency) — используется для хранения денежных значений в виде 8-байтовых чисел с точностью до четырех знаков после запятой. Этот тип данных применяется для хранения финансовых данных и, в тех случаях, когда значения не должны округляться.

**Счетчик** (AutoNumber) — используется для уникальных числовых 4-байтовых значений, которые автоматически вводит Access при добавлении записи.

Вводимые числа могут последовательно увеличиваться на указанное приращение или выбираться случайно. Обычно используются в первичных ключах.

**Логический (Yes/No)**— применяется для хранения логических значений, которые могут содержать одно из двух значений: Да/Нет, Истина/Ложь или Вкл/Выкл. (8 битов = 1 байт). Используется 1 для значений Да и 0 для значений Нет. Размер равен 1 биту.

**Поле объекта OLE (OLE Object)** — используется для хранения изображений, документов, диаграмм и других объектов из приложений Ms Office и других программ

Windows в виде растровых изображений, которые затем отображаются в элементах управления форм или отчетов, связанных с этим полем таблицы.

Чтобы в Access просмотреть эти изображения, необходимо, чтобы на компьютере, использующем базу данных, был зарегистрирован OLE-сервер (программа, поддерживающая этот тип файлов). Если для данного типа файлов OLE-сервер не зарегистрирован, отображается значок поврежденного изображения.

В accdb-файлах вместо типа данных **Поле объекта OLE** используется тип **Гиперссылка** (Hyperlink).

**Гиперссылка** (Hyperlink) — применяется для хранения ссылок на узлы (URL-адреса), на узлы или файлы интрасети или локальной сети (UNC-адреса — стандартного формата записи пути), а также на узлы или файлы локального компьютера. Кроме того, можно использовать ссылку на объекты Access, хранящиеся в базе данных. Может хранить до 1 Гбайт данных.

**Вложение** (Attachment) — используется для вложения в поле записи изображений, файлов электронных таблиц, документов, диаграмм и других файлов поддерживаемых типов точно так же, как в сообщения электронной почты. Вложенные файлы можно просматривать и редактировать в соответствии с заданными для поля параметрами. Эти поля не имеют ограничений, связанных с отсутствием зарегистрированных OLE - серверов. Более рационально используют место для хранения, чем поля с типом данных **Поле объекта OLE** (OLE Object), поскольку не создают растровые изображения исходного файла. Максимальная длина поля для сжатых

вложений — 2 Гбайт, для несжатых — примерно 700 Кбайт в зависимости от степени возможного сжатия вложения.

**Мастер подстановок** (Lookup Wizard) — появляется в списке типов данных только в режиме конструктора. Фактически типом данных не является, а вызывает мастер подстановок, с помощью которого можно создать поле, позволяющее выбрать значения из списка, построенного на основе значений поля другой таблицы, запроса или фиксированного набора значений. Такое поле отображается как поле со списком. Если список построен на основе поля таблицы или запроса, тип данных и размер создаваемого поля определяется типом данных и размером привязанного столбца, если на основе набора значений — размером текстового поля, содержащего значение.

**Замечание. Код репликации** (Replication ID). Глобальный уникальный идентификатор (Globally unique identifier, GUID), занимает 16 байтов. Эти длинные генерируемые случайным образом значения обеспечивают малую вероятность их совпадения. Поля такого типа используются Access для создания системных уникальных идентификаторов реплик, наборов реплик, таблиц, записей и других объектов при репликации баз данных. Могут быть

использованы в приложениях пользователя, например, для идентификации товаров.

#### 4.5. Общие свойства поля

Основные свойства задаются для каждого поля на вкладке **Общие (General)** и зависят от выбранного типа данных. Для отображения свойств поля необходимо установить курсор на строке соответствующего поля. Приведем свойства полей, наиболее важные на первом этапе изучения баз данных.

**-Размер поля (Field Size)** позволяет для текстового и числового поля уточнить тип данных или размер, задает максимальный размер данных, сохраняемых в поле. Для поля с типом данных **Текстовый (Text)** задается размер от 1 до 255 знаков.

Изменения в данных, которые происходят вследствие изменения свойства **Размер поля (Field Size)**, нельзя отменить после выполнения сохранения инструкторе таблиц.

**-Формат поля (Format)** является форматом отображения выбранного типа данных при выводе их на экран или печать в режиме таблицы, в форме или отчете. В

Access определены встроенные стандартные форматы отображения для полей с такими типами данных, как

**Числовой** (Number), **Дата/время** (Date/Time), **Логический** (Yes/No) и **Денежный** (Currency). Ряд этих форматов совпадает с настройкой региональных форматов, определяемых в окне **Язык и региональные стандарты** в Панели управления Windows. Пользователь может создать собственный формат для всех типов данных, кроме **Поле объекта OLE** (OLE Object), с помощью символов форматирования. Для указания конкретного формата отображения необходимо выбрать в раскрывающемся списке одно из значений свойства **Формат поля** (Format).

**-Число десятичных знаков** (DecimalPlaces) задает для числового и денежного типов данных количество знаков после запятой. Можно задать число от 0 до 15. По умолчанию (значение **Авто** (Auto)) это число определяется установкой в свойстве **Формат поля** (Format). Следует иметь в виду, что установка этого свойства не действует, если свойство **Формат поля** (Format) не установлено или выбрано значение **Основной** (General Number). Свойство **Число десятичных знаков** (DecimalPlaces) влияет только на количество десятичных знаков, отображаемых на экране, и не влияет на



число сохраняемых десятичных знаков. Для изменения числа сохраняемых знаков нужно изменить свойство **Размер поля** (Field Size).

**-Подпись** (Caption) поля задает текст, который выводится в таблицах, формах, отчетах.

**-Значение по умолчанию** (DefaultValue) определяет текст или выражение, значение которого автоматически вводится в поле при создании новой записи. Например, если задана функция =Now (), то в поле введется текущая дата и время. При добавлении записи в таблицу можно оставить значение, введенное по умолчанию, или ввести другое. Свойство **Значение по умолчанию** (DefaultValue)

используется только при создании новой записи. Максимальная длина значения свойства составляет 255 знаков. Свойство не определено для полей с типом данных

**Счетчик** (AutoNumber) или **Поле объекта OLE** (OLE Object).

**-Условие на значение** (ValidationRule) позволяет осуществлять контроль ввода, задает ограничения на вводимые значения, при нарушении условий запрещает ввод

и выводит текст, заданный свойством **Сообщение об ошибке** (ValidationText).

#### **4.6.Схема данных.**

Реляционная база данных, созданная в соответствии с проектом канонической модели данных предметной области, состоит из нормализованных таблиц, связанных отношениями 1-М. В такой базе данных обеспечивается отсутствие дублирования описательных данных, их однократный ввод, поддержание целостности данных средствами системы. Связи между таблицами позволяют выполнить объединение данных различных таблиц, необходимое для решения большинства задач ввода, просмотра и корректировки данных, получения информации по запросам и вывода отчетов.

Связи между таблицами устанавливаются в соответствии с проектом логической структуры базы данных и запоминаются в схеме данных Access. Схема данных является не только средством графического отображения логической структуры базы данных, она активно используется системой в процессе обработки данных. Система при необходимости обработки данных

взаимосвязанных таблиц автоматически использует связи, определенные в схеме данных. Таким образом, разработчику нет необходимости специально сообщать системе о наличии той или иной связи. Однажды указанные в схеме данных связи используются системой автоматически.

Создание схемы данных позволяет упростить конструирование многотабличных форм, запросов, отчетов, а также обеспечить поддержание целостности взаимосвязанных данных при вводе и корректировке данных в таблицах.

#### **4.7. Создание связей между таблицами схемы данных**

При создании связей в схеме данных используется проект логической структуры реляционной базы данных, в котором показаны все однозначные связи таблиц. Реализуются связи с помощью добавления в связанные таблицы общих полей, называемых ключом связи. При 1М отношениях между таблицами ключом связи является ключ главной таблицы (простой или составной). В подчиненной таблице он может быть частью уникального ключа или вовсе не входить в состав ключа таблицы. 1-М связи являются основными в реляционных базах данных. 1-М связи

используются, как правило, при необходимости распределять большое количество полей, определяемых одним и тем же ключом, по разным таблицам, имеющим разный регламент обслуживания.

### **Обеспечение целостности данных**

Если связываемые в схеме данных таблицы находятся в отношении 1 : 1 и 1 : М, для связи можно задать параметр обеспечения связной Целостности данных. Обеспечение связной целостности данных означает, что Access при корректировке базы данных реализует для связанных таблиц контроль соблюдения следующих условий:

- в подчиненную таблицу не может быть добавлена запись с несуществующим в записях главной таблицы значением ключа связи;
- в главной таблице нельзя удалить запись, если не удалены связанные с пей записи в подчиненной таблице;
- изменение значений ключа связи в записи главной таблицы невозможно, если подчиненная таблица

имеет связанные с ней записи, в которых соответственно не меняется ключ связи.

При попытке пользователя нарушить эти условия в операциях добавления и удаления записей или обновления ключевых данных в связанных таблицах Access не допускает выполнения операции и выводит соответствующее сообщение. Access не позволяет установить параметр целостности для связи таблиц, если ранее введенные в таблицы данные не отвечают требованиям целостности.

Установление между двумя таблицами связи и задание для нее параметров целостности данных возможно только при следующих условиях:

- связываемые поля имеют одинаковый тип данных, причем имена полей могут быть различными;
- обе таблицы сохраняются в одной базе данных Access;
- главная таблица связывается с подчиненной по первичному простому или составному ключу или уникальному индексу главной таблицы.

## **Каскадное обновление и удаление связанных записей**

Если для выбранной связи обеспечивается поддержание целостности, можно задать режим каскадного удаления связанных записей и каскадного обновления связанных полей. Такие параметры делают возможным в главной таблице, соответственно, удаление записей и изменение значения в ключевом поле, т. к. при этих параметрах система автоматически выполнит необходимые изменения в подчиненных таблицах, обеспечив сохранение свойств целостности базы данных.

В режиме **каскадного удаления связанных записей** при удалении записи из главной таблицы будут автоматически удаляться все связанные записи в подчиненных таблицах. При удалении записи из главной таблицы выполняется каскадное удаление подчиненных записей на всех уровнях, если этот режим задан на каждом уровне.

В режиме **каскадного обновления связанных полей** при изменении значения ключевого поля в записи главной таблицы Access автоматически изменит значения в

соответствующем поле в подчиненных записях.

## **Модификация структуры базы данных**

Понятие "модификация структуры базы данных" включает изменение структуры отдельных таблиц, добавление и удаление таблиц, а также изменение схемы данных.

### **Изменение структуры таблиц**

Для изменения структуры таблиц основным является режим конструктор таблиц, хотя некоторые изменения можно произвести в режиме таблицы. Такие простые операции, как изменение имени, некоторых типов данных, добавление, удаление не ключевых полей, добавление полей со списком, могут быть произведены в обоих режимах. Однако в режиме таблицы и эти операции выполняются с различными ограничениями. Например, можно в режиме таблицы изменить имя поля, дважды щелкнув в его заголовке и введя новое значение. Это значение становится именем поля, но при этом теряется значение подписи поля.

### **Изменение полей, которые не являются ключами или полями связи**

Состав и последовательность, а также тип данных, свойства или имена таких полей можно изменять независимо от наличия связей таблицы с другими таблицами базы данных. Для заполненной таблицы при изменении типа данных, размера и других свойств, производится преобразование данных. Однако, если преобразования недопустимы, попытка изменения типа данных может привести к потере данных, поэтому система запрашивает у пользователя подтверждение его действий.

### **Изменение или удаление ключевого поля**

Для загруженной и несвязанной таблицы при попытке изменить свойства или удалить ключевое поле система предупреждает о возможной потере данных удалении ключа. Если, например, удаляется поле в составном ключе, с других полей этого ключа будет снят признак ключа. При попытке назначить ключом другое поле, имеющее повторяющиеся значения в загруженной таблице, операция не будет завершена и будет выдано сообщение о невозможности.



Если надо изменить ключ таблицы, которая имеет связи с другими таблицами в схеме данных, необходимо предварительно разорвать связи.

### **Изменение схемы данных**

При модификации схемы данных может производиться изменение состава ее таблиц — удаление, добавление таблиц, изменение их связей, способа объединения записей связанных таблиц.

Необходимость в изменении связей возникает, в частности, при изменении ключей в таблицах. Изменение ключа по составу, по типу и размеру его полей не может производиться до тех пор, пока не удалены связи таблицы и схеме данных.

При изменении типа данных для неключевых полей, задействованных в связях таблицы, также предварительно необходимо удалить эти связи в схеме данных.

Для внесения изменений в схему данных перед ее открытием надо закрыть все таблицы. Команда **Схема данных** на вкладке **Работа с базами данных** в группе **Показать или скрыть** откроет схему данных. При этом

открывается вкладка ленты **Работа со связями** | **Конструктор**.

## 5.Создание таблиц при помощи ввода данных

В базе данных можно хранить данные в виде таблиц — тематических списков строк и столбцов. Например, вы можете создать таблицу "Контакты" для хранения имен, адресов и телефонных номеров или таблицу "Товары" для хранения сведений о товарах.

В этой статье описано, как создать таблицу, добавить в нее поля и настроить первичный ключ. Кроме того, в ней приведены инструкции по заданию свойств таблиц и полей.

Так как другие объекты базы данных в значительной степени зависят от таблиц, всегда начинайте разработку базы данных с создания всех таблиц, а уже затем создавайте другие объекты. Перед созданием таблиц необходимо тщательно проанализировать свои требования и выяснить, какие таблицы нужны. Общие сведения о планировании и разработке базы данных см. в статье Основные сведения о создании баз данных.

## Обзор

Таблицы — это объект базы данных, который содержит данные по определенной теме, например сведения о сотрудниках или товарах. Таблица состоит из полей и записей.

Каждая запись включает данные об одном элементе, например о конкретном сотруднике. Записи также часто называют строками или экземплярами.

Каждое поле включает данные об одном аспекте объекта таблицы, например имя или адрес электронной почты. Поля также часто называют столбцами или атрибутами.

Запись состоит из значений полей, таких как Contoso, Ltd. или proverka@example.com. Значение поля также называют фактом.

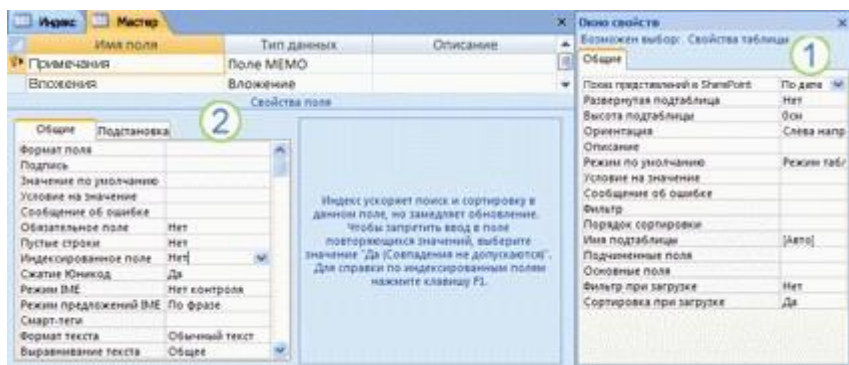
Код	Организация	Имя	Фамилия
1	Организация А	Ольга	Костерина
2	Организация В	Григорий	Верный
3	Организация С	Владимир	Егоров

1. Запись
2. Поле
3. Значение поля

База данных Access может включать много таблиц, в каждой из которых хранятся данные по отдельным объектам. Каждая таблица может содержать много полей с данными различного типа, включая текст, числа, даты и гиперссылки.

## 5.1. СВОЙСТВА ТАБЛИЦ И ПОЛЕЙ

У таблиц и полей также есть свойства, которые позволяют управлять их характеристиками и работой.



Таблица, открытая в Конструкторе.

1. Свойства таблицы
2. Свойства поля

В базе данных Access свойствами таблицы называются атрибуты, определяющие ее внешний вид и работу. Свойства таблицы задаются на странице свойств таблицы в Конструкторе. Например, вы можете задать для таблицы свойство **Режим по умолчанию**, чтобы указать, как она должна отображаться по умолчанию.

Свойство поля применяется к определенному полю в таблице и определяет его характеристики или определенный аспект поведения. Некоторые свойства поля можно задать в режим таблицы. Вы также можете настраивать любые свойства в Конструкторе с помощью области **Свойства поля**.

## 5.2.ТИПЫ ДАННЫХ

У каждого поля есть тип данных. Тип данных поля определяет данные, которые могут в нем храниться (например, большие объемы текста или вложенные файлы).

Имя поля	Тип данных	Описание
Примечания	Поле МЕМО	
Вложения	Вложение	
Свойства поля		

Тип данных является свойством поля, однако он отличается от других свойств:

- Тип данных поля задается на бланке таблицы, а не в области **Свойства поля**.
- Тип данных определяет, какие другие свойства есть у этого поля.
- Тип данных необходимо указывать при создании поля.

Чтобы создать новое поле в Access, введите данные в новый столбец в режиме таблицы. В таком случае Access автоматически определяет тип данных для поля в зависимости от введенного значения. Если оно не относится к определенному типу, Access выбирает текстовый тип. При необходимости его можно изменить с помощью ленты.

## **Примеры автоматического определения типа данных**

Ниже показано, как выполняется автоматическое определение типа данных в режиме таблицы.

Вводимые данные	Тип данных для назначаемый Access поля,
Иван	Текст
<p data-bbox="120 600 389 632"><a href="http://www.contoso.com">http://www.contoso.com</a></p> <p data-bbox="104 699 468 831">Вы можете использовать любой допустимый префикс протокола</p> <p data-bbox="120 903 404 999">IP. Например, являются допустимыми префиксы</p> <p data-bbox="120 1031 393 1062"><a href="http://">http://</a>, <a href="https://">https://</a> и <a href="mailto:">mailto:.</a></p>	Гиперссылка
1	Число, длинное целое
50 000	Число, длинное целое
50 000,99	Число, double



50000,389

Число, double

12.67

Дата и время

Распознаваемые форматы  
даты и времени зависят от  
языкового стандарта.

Дата и время

31 декабря 2016 г.

10:50:23

Дата и время

10:50

Дата и время

17:50

Дата и время

Денежный

12,50 Р

Распознаваемое  
обозначение денежной  
единицы зависит  
от языкового стандарта.

21,75

Число, double

123,00%

Число, double

3,46E+03

Число, double

### 5.3. ОТНОШЕНИЯ МЕЖДУ ТАБЛИЦАМИ

Хотя в каждой из таблиц хранятся данные по отдельному объекту, в базе данных Access все они обычно связаны между собой. Ниже приведены примеры таблиц в базе данных.

- Таблица клиентов, содержащая сведения о клиентах компании и их адреса.
- Таблица продаваемых товаров, включающая цены и изображения каждого из них.
- Таблица заказов, служащая для отслеживания заказов клиентов.

Так как данные по разным темам хранятся в отдельных таблицах, их необходимо как-то связать, чтобы можно было легко комбинировать данные из разных таблиц. Для этого используются связи. Связь — это логическое отношение между двумя таблицами, основанное на их общих полях.

## **Ключи**

Поля, формирующие связь между таблицами, называются ключами. Ключ обычно состоит из одного поля, однако может включать и несколько.

### **Есть два вида ключей.**

- **Первичный ключ.** В таблице может быть только один первичный ключ. Он состоит из одного или нескольких полей, однозначно определяющих каждую запись в этой таблице. Часто в качестве первичного ключа используется уникальный идентификатор, порядковый номер или код. Например, в таблице "Клиенты" каждому клиенту может быть назначен уникальный код клиента. Поле кода клиента является первичным ключом этой таблицы. Если первичный ключ состоит из нескольких полей, он обычно включает уже существующие поля, формирующие в сочетании друг с другом уникальные значения. Например, в таблице с данными о людях в качестве первичного ключа можно использовать сочетание фамилии, имени и даты рождения.

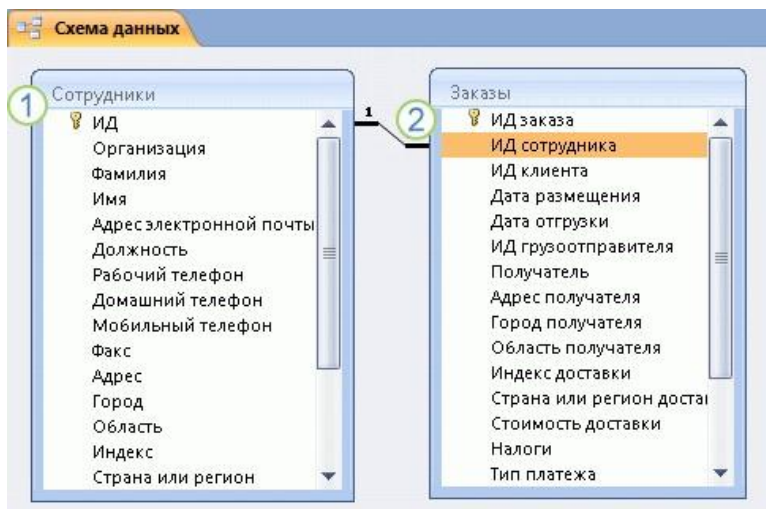
- **Внешний ключ.** В таблице также может быть один или несколько внешних ключей. Внешний ключ содержит

значения, соответствующие значениям первичного ключа другой таблицы. Например, в таблице "Заказы" каждый заказ может включать код клиента, соответствующий определенной записи в таблице "Клиенты". Поле "Код клиента" является внешним ключом таблицы "Заказы".

Соответствие значений между полями ключей является основой связи между таблицами. С помощью связи между таблицами можно комбинировать данные из связанных таблиц. Предположим, есть таблицы "Заказчики" и "Заказы".

В таблице "Заказчики" каждая запись идентифицируется полем первичного ключа — "Код".

Чтобы связать каждый заказ с клиентом, вы можете добавить в таблицу "Заказы" поле внешнего ключа, соответствующее полю "Код" в таблице "Заказчики", а затем создать связь между этими двумя ключами. При добавлении записи в таблицу "Заказы" можно было бы использовать значение кода клиента из таблицы "Заказчики". При просмотре каких-либо данных о клиенте, сделавшем заказ, связь позволяла бы определить, какие данные из таблицы "Заказчики" соответствуют тем или иным записям в таблице "Заказы".



Связь между таблицами в окне "Схема данных".

1. Первичный ключ, который определяется по значку ключа рядом с именем поля.
2. Внешний ключ (определяется по отсутствию значка ключа)

### ***Преимущества использования связей***

Раздельное хранение данных в связанных таблицах обеспечивает указанные ниже преимущества.

- **Согласованность** . Поскольку каждый элемент данных заносится только один раз в одну таблицу, вероятность появления неоднозначных или несогласованных данных снижается. Например, имя клиента будет храниться только в таблице клиентов, а не в нескольких записях в таблице заказов, которые могут стать несогласованными.

- **Эффективность** . Хранение данных в одном месте позволяет сэкономить место на диске. Кроме того, данные из небольших таблиц извлекаются быстрее, чем из больших. Наконец, если не хранить данные по различным темам в разных таблицах, возникают пустые значения, указывающие на отсутствие данных, или избыточные данные, что может привести к неэффективному использованию места и снижению производительности.

- **Простота** . Структуру базы данных легче понять, если данные по различным темам находятся в разных таблицах.

**Совет:** Связи между таблицами необходимо иметь в виду еще на этапе планирования таблиц. С помощью мастера подстановок можно создать поле внешнего ключа, если таблица с соответствующим первичным ключом уже существует. Мастер подстановок помогает создать связь.



#### **5.4.Создание новой таблицы.**

В простой базе данных, такой как список контактов, может быть всего одна таблица. Однако во многих базах данных используется несколько таблиц. При создании базы данных на компьютере создается файл, который используется как контейнер для всех ее объектов, включая таблицы.

Есть несколько способов создать таблицу: вы можете создать новую базу данных, вставить таблицу в существующую базу данных или импортировать таблицу из другого источника данных, например книги Microsoft Office Excel, документа Microsoft Office Word, текстового файла или другой базы данных, либо связать таблицу с этим источником. Когда вы создаете новую базу данных, в нее автоматически вставляется новая пустая таблица. Затем вы можете ввести в нее данные, чтобы начать определение полей.

### **Создание таблицы в новой базе данных**

1. Откройте меню **Файл** и выберите команду **Создать**.  
Если вы используете Access 2016, нажмите **кнопку Microsoft Office** и выберите команду **Создать**.
2. В поле **Файл** введите имя файла новой базы данных.
3. Чтобы сохранить базу данных в другом месте, щелкните значок папки.
4. Нажмите кнопку **Создать**.

Откроется новая база данных, в которой будет создана и открыта в режиме таблицы новая таблица с именем "Таблица1".

### **Создание таблицы в существующей базе данных**

1. Откройте меню **Файл** и выберите команду **Открыть**. Если вы используете Access 2016, нажмите **кнопку Microsoft Office** и выберите команду **Открыть**.

2. В диалоговом окне **Открытие файла базы данных** найдите базу данных, которую вы хотите открыть, и нажмите кнопку **Открыть**.
3. на вкладке **Создание** в группе **Таблицы** нажмите кнопку **Таблица**.

В базу данных будет вставлена новая таблица, которая откроется в режиме таблицы.

## **6. Использование импорта или связи для создания таблицы**

Чтобы создать таблицу, вы можете импортировать данные из другого файла (например, из листа Excel, списка SharePoint, XML-файла, другой базы данных Access, папки Microsoft Outlook и т. д.) либо связать таблицу с ним.

При импорте данных создается их копия в новой таблице текущей базы данных. Последующие изменения, вносимые в исходные данные, не будут влиять на импортированные данные, и наоборот. После подключения к источнику и импорта данных можно использовать импортированные данные без подключения к источнику. В

структуру импортированной таблицы можно вносить изменения.

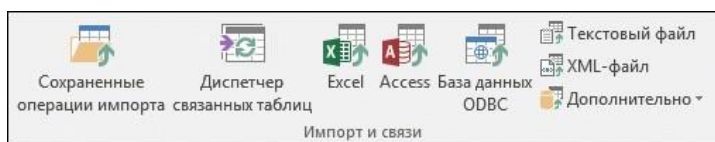
Когда осуществляется связывание с данными, в текущей базе данных создается связанная таблица, обеспечивающая динамическое подключение к данным, хранящимся в другом месте. Изменения данных в связанной таблице отражаются в источнике, а изменения в источнике — в связанной таблице. Для работы со связанной таблицей необходимо подключение к источнику данных. Изменить структуру связанной таблицы нельзя.

**Примечание:** Данные на листе Excel невозможно изменять с помощью связанной таблицы. Чтобы решить задачу обходным путем, импортируйте исходные данные в базу данных Access, а затем свяжите ее с Excel.

## **6.1. СОЗДАНИЕ ТАБЛИЦЫ С ПОМОЩЬЮ ИМПОРТА ВНЕШНИХ ДАННЫХ ИЛИ СВЯЗИ С НИМИ**

1. Откройте меню **Файл** и выберите команду **Открыть**. Если вы используете Access 2016, нажмите кнопку **Microsoft Office** и выберите команду **Открыть**.

2. В диалоговом окне **Открытие файла базы данных** выберите и откройте базу данных, в которой вы хотите создать таблицу.
3. На вкладке **Внешние данные** в группе **Импорт и связи** выберите один из доступных источников данных. Если вы используете Access 2016, источники данных вы найдете в группе **Импорт**.



4. Следуйте инструкциям в появляющихся диалоговых окнах.

Будет создана таблица, имя которой появится в области навигации.

Кроме того, вы можете импортировать список SharePoint или связать с ним таблицу, выбрав соответствующую команду на вкладке **Создание**.

## **Создание таблицы с использованием сайта SharePoint**

Для создания таблицы базы данных можно использовать импортировать список SharePoint или создать связь с ним. Кроме того, вы можете создать новый список SharePoint на основе готового шаблона.

**В Access доступны такие шаблоны, как "Контакты", "Задачи", "Вопросы" и "События".**

2. Откройте меню **Файл** и выберите команду **Открыть**. Если вы используете Access 2016, нажмите кнопку **Microsoft Office** и выберите команду **Открыть**.
3. В диалоговом окне **Открытие файла базы данных** выберите базу данных, в которой вы хотите создать таблицу, и нажмите кнопку **Открыть**.
4. На вкладке **Создание** в группе **Таблицы** нажмите кнопку **Списки SharePoint**.
5. Выполните одно из указанных ниже действий.

○ **Создание списка SharePoint на основе шаблона**

1. Выберите пункт **Контакты, Задачи, Вопросы** и **События**.
2. В диалоговом окне **Создание нового списка** введите URL-адрес сайта SharePoint, на котором вы хотите создать список.
3. Введите имя и описание для нового списка в полях **Укажите имя нового списка** и **Описание**.
4. Чтобы открыть связанную таблицу после ее создания, установите флажок **Открыть список по окончании экспорта** (он установлен по умолчанию).

○ **Создание настраиваемого списка**

1. Выберите пункт **Другой**.
2. В диалоговом окне **Создание нового списка** введите URL-адрес сайта SharePoint, на котором вы хотите создать список.
3. Введите имя и описание для нового списка в полях

**Укажите имя нового списка и Описание.**

4. Чтобы открыть связанную таблицу после ее создания, установите флажок **Открыть список по окончании экспорта** (он установлен по умолчанию). о **6.2.Импорт данных из существующего списка**

1. Выберите пункт **Существующий список SharePoint.**

2. В диалоговом окне **Внешние данные** введите URL-адрес сайта SharePoint, содержащего данные, которые нужно импортировать.

3. Выберите пункт **Импортировать данные источника в новую таблицу в текущей базе данных** и нажмите кнопку **Далее.**

4. Установите флажки всех списков SharePoint, которые нужно импортировать.

о **Связь с существующим списком**

1. Выберите пункт **Существующий список SharePoint.**



2. В диалоговом окне **Внешние данные — сайт SharePoint** введите URL-адрес сайта SharePoint, содержащего список, связь с которым нужно создать.
3. Выберите пункт **Создать связанную таблицу для связи с источником данных** и нажмите кнопку **Далее**.
4. Установите флажки всех списков SharePoint, связи с которыми нужно создать.

### **Настройка первичного ключа таблицы**

Для таблицы следует задать первичный ключ, если только у вас нет веских оснований не делать этого. Access автоматически создает индекс для первичного ключа, что повышает производительность базы данных. Кроме того, Access гарантирует, что каждая запись имеет значение в поле первичного ключа и это значение всегда уникально. Это крайне важно, поскольку в противном случае нельзя надежно отличить одну строку от другой.

При создании таблицы в режиме таблицы Access автоматически создает первичный ключ с именем "Код" и присваивает ему тип данных

"Счетчик".

Изменить или удалить первичный ключ, а также задать первичный ключ для таблицы, в которой его еще нет, можно в Конструкторе.

### **6.3. ОПРЕДЕЛЕНИЕ ПОЛЕЙ, ИСПОЛЬЗУЕМЫХ В КАЧЕСТВЕ ПЕРВИЧНОГО КЛЮЧА**

Иногда данные, которые можно использовать в качестве первичного ключа, уже есть. Например, у сотрудников уже могут быть идентификационные номера. Если вы создаете таблицу для отслеживания данных о сотрудниках, в качестве первичного ключа можно использовать их идентификаторы. Иногда идентификаторы сотрудников уникальны только в сочетании с кодами отделов; в этом случае в качестве первичного ключа необходимо использовать сочетание этих полей. Первичный ключ должен обладать следующими характеристиками:

- Значение данного поля или сочетания полей должно быть уникальным для каждой записи.
- Поле или сочетание полей не должно быть пустым (у них всегда должно быть значение).
- Значения не должны изменяться.

Если данных, подходящих на роль первичного ключа, нет, для этого можно создать новое поле. При создании поля для использования в качестве первичного ключа выберите для него тип "Счетчик" — это обеспечит соответствие трем характеристикам, указанным выше.

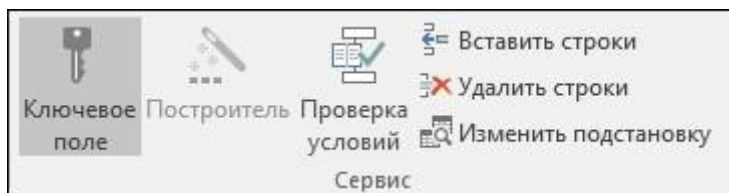
## **НАСТРОЙКА ИЛИ ИЗМЕНЕНИЕ ПЕРВИЧНОГО КЛЮЧА**

1. Выберите таблицу, для которой вы хотите задать или изменить первичный ключ.
2. На вкладке **Главная** в группе **Представления** нажмите кнопку **Представление** и выберите пункт **Конструктор**.
3. На бланке таблицы выберите поле или поля, которые вы хотите использовать в качестве первичного ключа.

Чтобы выделить одно поле, щелкните область выделения строки для него.

Чтобы выделить несколько полей, щелкните область выделения для каждого поля, удерживая нажатой клавишу **CTRL**.

4. На вкладке **Конструктор** в группе **Элементы** нажмите кнопку **Ключевое поле**.

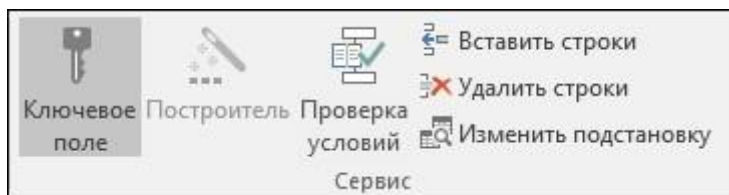


Индикатор ключа будет добавлен слева от поля или полей, определенных как первичный ключ.

## УДАЛЕНИЕ ПЕРВИЧНОГО КЛЮЧА

1. Выберите таблицу, первичный ключ которой вы хотите удалить.
2. На вкладке **Главная** в группе **Представления** нажмите кнопку **Представление** и выберите пункт **Конструктор**.
3. Щелкните область выделения строки для текущего первичного ключа. Если первичный ключ состоит из нескольких полей, щелкните область выделения строки для каждого из них, удерживая нажатой клавишу CTRL.

4. На вкладке **Конструктор** в группе **Элементы** нажмите кнопку **Ключевое поле**.



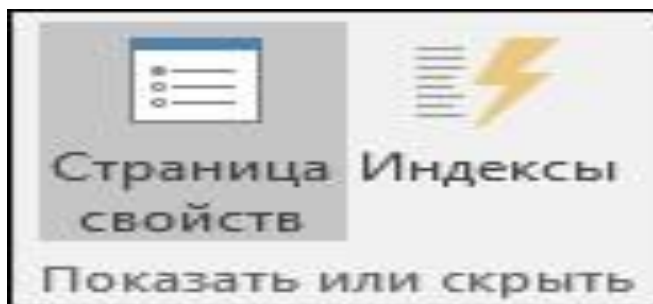
Индикатор ключа будет удален из поля или полей, ранее определенных в качестве первичного ключа.

**Примечание:** Если попытаться сохранить новую таблицу без первичного ключа, Access предложит создать поле для него. Если нажать кнопку **Да**, Access создаст поле "Код" с типом данных "Счетчик", в котором будет храниться уникальное значение для каждой записи. Если в таблице уже есть поле с типом "Счетчик", оно будет использовано в качестве первичного ключа. Если нажать кнопку **Нет**, Access не добавит новое поле и не задаст первичный ключ.

## Настройка свойств таблицы

Кроме настройки свойств полей, можно задать свойства, которые применяются ко всей таблице или ко всем записям.

1. Выберите таблицу, для которой требуется задать свойства.
2. На вкладке **Главная** в группе **Представления** нажмите кнопку **Представление** и выберите пункт **Конструктор**.
3. На вкладке **Конструктор** в группе **Показать или скрыть** выберите пункт **Страница свойств**.



Свойство	Способ использования
таблицы	
<b>Представления на вебсайте SharePoint</b>	Укажите, должны ли представления, основанные на таблице, отображаться на сайте SharePoint. <b>Примечание:</b> Действие этого параметра зависит от значения свойства базы данных <b>Отображать все представления на сайте SharePoint.</b>
<b>Развернутая подтаблица</b>	Укажите, следует ли развертывать все подтаблицы при открытии таблицы.
<b>Высота подтаблицы</b>	Выполните одно из указанных ниже действий Чтобы в окне подтаблицы отображались

все строки, оставьте значение **0**.

Чтобы задать высоту подтаблицы, введите нужное значение.

**Ориентация** Укажите направление отображения данных **ия** в соответствии с направлением письма

(слева направо или справа налево).

<b>Описание</b>	Укажите описание таблицы, которое будет появляться в качестве всплывающей подсказки для этой таблицы.
-----------------	---

**Режим по умолчанию** Установка **таблицы, сводной таблицы** или **сводной диаграммы** в качестве начального режима по умолчанию.



Обратите внимание на то, что режимы сводной таблицы и сводной диаграммы в Access недоступны, начиная с Access

2016.

**Условие на значение** Введите выражение, которое должно

быть истинным при добавлении или

изменении любой записи.

**Сообщение об ошибке** Введите сообщение, отображаемое в том

случае, если запись не соответствует

выражению в своем **Правило проверки**

**Фильтр** Определите условия отображения строк в

Режиме таблицы.

**Порядок сортировки** Выберите одно или несколько полей, чтобы

---

задать порядок сортировки по умолчанию для строк в режиме таблицы.

<b>Имя подтаблицы</b>	Укажите, должна ли подтаблица отображаться в режиме таблицы. Если да, укажите таблицу или запрос, из которых подтаблица должна получать строки.
-----------------------	---

**Подчиненные поля** Укажите поля в таблице или запросе, используемые в подтаблице и соответствующие свойству **Основные поля**, заданному для таблицы.

<b>Основные поля</b>	Укажите поля таблицы, соответствующие свойству <b>Подчиненные поля</b> , заданному для таблицы.
----------------------	---

**Фильтр** Укажите, необходимо ли автоматически  
**при** применять услов фильтра,  
**загрузке**

заданные с помощью свойства **Фильтр** (если

присвоено значение **Да**), при открытии таблицы в режиме  
таблицы.

**Сортировк** Укажите, необходимо ли автоматически  
приме-

**а при**

**загрузке** нять условия сортировки, заданные с помощью

свойства **Порядок сортировки** (если присво-

ено значение **Да**), при открытии таблицы в режиме  
таблицы.


Откроется страница свойств таблицы.

4. На странице свойств откройте вкладку **Общие**.
5. Щелкните поле слева от свойства, которое вы хотите задать, и введите значение.

### Доступные свойства таблицы

**Совет:** Если в поле свойства недостаточно места для ввода или изменения значения, нажмите сочетание клавиш SHIFT+F2, чтобы открыть поле **Область**

**ввода**. Если при назначении выражению свойства

**Правило проверки** требуется помощь, нажмите кнопку  рядом с полем свойства **Правило проверки**, чтобы открыть построитель выражений.

6. Чтобы сохранить изменения, нажмите клавиши CTRL+S.

### 7. Вставка и добавление поля в таблицу и создание нового поля

Все элементы данных, которые необходимо отслеживать, хранятся в отдельных полях. Например, в таблице контактов можно создать поля "Имя", "Фамилия",

"Телефон" и "Адрес", а в таблице товаров — поля "Название товара", "Код товара" и "Цена".

Прежде чем создавать поля, разделите данные на минимальные полезные элементы. Потом вам будет намного проще объединить данные, чем разделить их. Например, вместо поля "Полное имя" лучше создать отдельные поля "Имя" и "Фамилия". Благодаря этому вам будет проще искать и сортировать данные по имени, фамилии или их сочетанию. Если вы собираетесь создавать отчет, выполнять сортировку, поиск или вычисления по элементу данных, выделите его в отдельное поле.

Для поля можно задать свойства, определяющие его вид и поведение.

### **Добавление поля путем ввода данных**

Чтобы при создании новой или открытии существующей таблицы в режиме таблицы добавить в нее поле, введите данные в столбец таблицы **Добавить поле**.

Состояние заказа			
	Код состояние ▾	Название состояния ▾	Добавить поле
+	0	Создать	1
+	1	Выписан счет	
+	2	Отгружен	
+	3	Закрыт	

Введите данные в столбец **Добавить поле**.

1. Чтобы создать или открыть таблицу в режиме таблицы, в области навигации щелкните нужную таблицу правой кнопкой мыши и затем в контекстном меню выберите пункт **Режим таблицы**.
2. Введите имя создаваемого поля в столбец **Добавить поле**.
3. Используйте описательное имя, помогающее идентифицировать поле.
4. Введите данные в новое поле.

### **Добавление поля с помощью шаблона**

Иногда проще выбрать поле из готового списка, чем создавать его вручную. Чтобы выбрать поле из списка

шаблонов, воспользуйтесь списком **Другие поля** в Access 2016, 2013 или 2016, либо областью задач **Шаблоны полей** в Access 2016. Шаблон поля – это заданный набор характеристик и свойств, описывающих его. Определение шаблона поля включает имя поля, тип данных, значение свойства **Формат** и ряд других свойств.

Если вы используете Access 2016, 2013 или 2016 выполните действия ниже.

1. На вкладке **Главная** в группе **Представления** нажмите кнопку **Представление** и выберите пункт **Режим таблицы**.
2. На вкладке **Поля** в группе **Добавление и удаление** нажмите кнопку **Другие поля**.

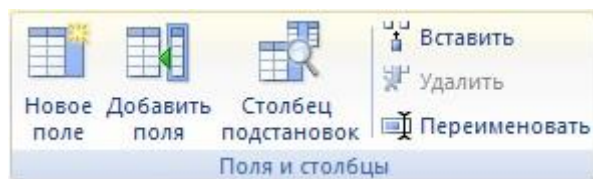


3. Чтобы вставить новый столбец, выберите поле в списке **Другие поля**. Access помещает поле справа от столбца, в котором находится курсор. Если вы

выберете один из заголовков Quick Start, например "Адрес", Access создаст множество полей в таблице для разных составляющих частей адреса.

Если вы используете Access 2016, выполните действия ниже.

1. На вкладке **Главная** в группе **Представления** нажмите кнопку **Представление** и выберите пункт **Режим таблицы**.
2. На вкладке **Таблица** в группе **Поля и столбцы** щелкните **Новое поле**.



3. Выберите одно или несколько полей в области **Шаблоны полей** и перетащите их в то место таблицы, куда вы хотите вставить новый столбец.



## 7.1. Задание свойств полей

Для поля можно задать свойства, определяющие его вид и поведение.

Например, с помощью свойств поля можно:

- изменить вид данных в поле;
- предотвратить ввод неправильных данных в поле;
- задать для поля значение по умолчанию;
- ускорить поиск и сортировку по полю.

Некоторые свойства поля можно задать в режиме таблицы, однако для доступа ко всем свойствам и их настройки необходимо использовать конструктор.

### Задание свойств поля в режиме таблицы

В режиме таблицы можно переименовать поле, изменить его тип, свойство **Формат** и некоторые другие свойства.

## ОТКРЫТИЕ ТАБЛИЦЫ В РЕЖИМЕ ТАБЛИЦЫ

1. В области навигации щелкните правой кнопкой мыши таблицу, которую вы хотите открыть.
2. В контекстном меню выберите пункт **Режим таблицы.**

## ПЕРЕИМЕНОВАНИЕ ПОЛЯ

Когда вы добавляете поле путем ввода данных в режиме таблицы, Access автоматически присваивает ему универсальное имя. Первому полю назначается имя "Поле1", второму — "Поле2" и т. д. По умолчанию имя поля используется в качестве его метки везде, где поле отображается (например, в заголовке столбца таблицы). Если вы присвоите полям описательные имена, вам будет легче просматривать и изменять записи.

1. Щелкните правой кнопкой мыши заголовок поля, которое требуется переименовать (например, "Поле1").
2. В контекстном меню выберите пункт **Переименовать поле.** Если вы используете Access

2016, в контекстном меню выберите пункт **Переименовать столбец**.

3. Введите новое имя в заголовок поля.

Имена полей могут содержать до 64 символов (цифры или букв), включая пробелы.

## 7.2. ИЗМЕНЕНИЕ ТИПА ДАННЫХ ПОЛЯ

Когда вы создаете поле путем ввода данных в режиме таблицы, приложение Access анализирует их, чтобы определить тип данных для поля. Например, если ввести значение **01.01.2017**, Access распознает его как дату и назначит полю тип "Дата и время". Если Access не может точно определить тип данных, по умолчанию полю назначается текстовый тип. ("Краткий текст", если вы используете Access 2016).

Тип данных поля определяет, какие еще свойства вы можете для него задать. Например, свойство **Только добавление** можно задать только для поля с типом данных "Гиперссылка" или "Поле МЕМО" (или "Длинный текст" в Access 2016).

Бывают ситуации, когда нужно изменить тип данных поля вручную. Представьте, что вам нужно ввести номера комнат, напоминающие даты (например, 10.2017). Если ввести значение **10.2017** в новое поле в режиме таблицы, функция автоматического определения типа данных выберет для поля тип данных "Дата и время". Поскольку номера комнат являются метками, а не датами, для них должен быть установлен тип данных "Текст". Чтобы изменить тип данных поля, выполните указанные ниже действия.

1. На ленте откройте вкладку **Поля**. Если вы используете Access 2016, откройте вкладку **Режим таблицы**.
2. В группе **Форматирование** в списке **Тип данных** выберите нужный тип данных. Если вы используете Access 2016, в группе **Форматирование и тип данных** в списке "Тип данных" выберите нужный.

### **Доступные типы данных**

Полный список доступных типов данных для баз данных Access вы найдете в статье [Типы данных для баз данных Access рабочего стола](#).

**Примечание:** Максимальный размер файла базы данных Access составляет 2 ГБ.

### Советы по типам данных

○ Чтобы оптимизировать производительность, при создании текстовых и числовых полей укажите оптимальный **Размер поля**. Например, если вы собираетесь хранить почтовые индексы известной длины, задайте соответствующие размеры поля. Для этого введите нужное значение в поле **Размер поля**. Дополнительные сведения см. в разделе Задание других свойств поля.

○ Для телефонных номеров, артикулов и других номеров, которые не предполагается использовать в математических вычислениях, вместо числового необходимо выбрать текстовый тип данных. Числовые значения, которые хранятся как текст, проще сортировать и фильтровать.

### 7.3. ИЗМЕНЕНИЕ ФОРМАТА ПОЛЯ

Кроме определения типа данных нового поля Access может задать для него значение свойства **Формат**, зависящее от введенных данных.

Например, если ввести значение 10:50, Access выберет тип данных "Дата и время" и присвоит свойству **Формат** значение "Средний формат времени". Чтобы вручную изменить значение свойства **Формат**, сделайте следующее.

1. На ленте откройте вкладку **Поля**. Если вы используете Access 2016, откройте вкладку **Режим таблицы**.
2. В группе **Форматирование** в поле **Формат** введите нужный формат. Если вы используете Access 2016, в группе **Форматирование и тип данных** в списке **Формат** выберите нужный.

**Примечание:** Для полей некоторых типов (например, текстовых) список **Формат** может быть недоступен.

## ЗАДАНИЕ ДРУГИХ СВОЙСТВ ПОЛЯ

1. В режиме таблицы щелкните поле, для которого нужно задать свойство.
2. Если вы используете Access 2016, 2013 или 2016, на вкладке **Поля** в группах **Свойства**,

**Форматирование** или **Проверка поля** выберите нужные свойства.

3. Если вы используете Access 2016, на вкладке **Режим таблицы** в группе **Форматирование и тип данных** выберите нужные свойства.

### **Задание свойств поля в Конструкторе**

При работе с таблицей в Конструкторе можно настраивать любые свойства полей. Тип данных поля задается на бланке таблицы, а другие свойства — в области **Свойства поля**.

#### **7.4. ОТКРЫТИЕ ТАБЛИЦЫ В КОНСТРУКТОРЕ**

1. В области навигации щелкните таблицу правой кнопкой мыши.
2. В контекстном меню выберите пункт **Конструктор**.

#### **ИЗМЕНЕНИЕ ТИПА ДАННЫХ ПОЛЯ**

1. На бланке таблицы найдите поле, для которого вы хотите задать тип данных.

2. В столбце **Тип данных** выберите значение из списка. **Доступные типы данных** Полный список доступных типов данных для баз данных Access вы найдете в статье [Типы данных для баз данных Access рабочего стола.](#)

**Примечание:** Максимальный размер файла базы данных Access составляет 2 ГБ.

#### Советы по типам данных

- Для телефонных номеров, артикулов и других номеров, которые не предполагается использовать в математических вычислениях, вместо числового необходимо выбрать текстовый тип данных. Числовые значения, которые хранятся как текст, проще сортировать и фильтровать, но их сложно использовать в вычислениях.

- Для текстового и числового типа данных вы можете более точно задать размер поля или тип данных с помощью свойства **Размер поля**.

-



## ЗАДАНИЕ ДРУГИХ СВОЙСТВ ПОЛЯ

1. На бланке таблицы выберите поле, для которого вы хотите задать свойства. Свойства поля выводятся в области **Свойства поля**.

Свойства, которые можно задать, зависят от типа данных этого поля.


2. В области **Свойства поля** введите нужные параметры для каждого свойства или нажмите клавишу F6 и выберите свойство с помощью клавиш со стрелками.

### Какие свойства полей доступны?

Полный список свойств полей, доступных для каждого типа данных в базах данных приложения Access см. в статье [Введение в использование типов данных и свойств полей](#).

**Примечание:** Не все свойства доступны для каждого поля. Свойства поля определяются его типом данных.

3. Если требуется больше места для ввода или изменения значений в поле свойства, нажмите клавиши SHIFT+F2 для отображения поля **Область ввода**.

**Совет:** Если вам требуется помощь в создании маски ввода или выражения условия на значение, нажмите кнопку  рядом с полем свойства для отображения соответствующего построителя.

4. Чтобы сохранить изменения, нажмите клавиши CTRL+S.

### **Сохранение таблицы**

После создания или изменения таблицы следует сохранить ее структуру. При первом сохранении таблице необходимо присвоить имя, описывающее содержащиеся в ней данные. Можно использовать до 64 знаков (букв или цифр), включая пробелы. Например, вы можете назвать таблицу "Клиенты", "Перечень запасных частей" или "Товары".

## **8. Ввод данных в таблицу и редактирование их**

В созданную таблицу данные могут быть введены как непосредственно в табличной форме «по умолчанию», так и с использованием специально разработанной пользовательской экранной формы. Редактирование записей и исправление ошибок в данных таблицы возможно также в каждом из двух указанных режимов.

При любом из указанных способов ввода и корректировки данных таблицы Access сохраняет введенную или исправленную запись на диске (том, на котором создана таблица БД).

### **ВВОД ДАННЫХ**

В режиме таблицы показ записей в формате строк и столбцов обеспечивает возможность одновременного просмотра нескольких записей. Допускается также добавление и изменение данных в режиме таблицы.

Вдоль верхнего края окна расположены имена полей таблицы. Каждое поле соответствует определенному столбцу в таблице. Каждая запись занимает одну строку таблицы. Ввод в определенную ячейку таблицы (выделенную курсором) осуществляется путем набора

информации на клавиатуре и последующим нажатием клавиши **Enter** или **Tab**. При окончании ввода данных в последнее поле записи Access сам переходит на первое поле новой записи и ожидает ввода данных.

**Внимание!** Заполнения с клавиатуры требуют все поля, кроме тех, тип которых определен как **Счетчик**.

## ПЕРЕМЕЩЕНИЕ В ТАБЛИЦЕ

Для быстрого просмотра данных, введенных в таблицу, а также необходимого позиционирования в таблице нужно, обратить внимание на возможности быстрого перемещения в таблице.

Первая запись — щелчок мышью по кнопке **Первая запись**;

Последняя запись — щелчок мышью по кнопке **Последняя запись**;

Первый столбец таблицы — клавиша **Home**;

Последний столбец таблицы — клавиша **End**;

Следующий столбец справа — одна из клавиш **Right**, **Enter** или **Tab**;

Следующий столбец слева — клавиша **Right**, или **Shift+Tab**;

На строку вверх – **Up** ;

На строку вниз — **Down**;

Вверх на 26 строк — **PgUp**;

Вниз на 26 строк — **PgDn**;

В левый верхний угол таблицы — **Ctrl+Home**; В  
правый нижний угол таблицы — **Ctrl+End**.

## **8.1. РЕДАКТИРОВАНИЕ ДАННЫХ**

Редактировать данные ячейки таблицы можно как с полной, так и с частичной их заменой.

Для полной замены данных необходимо подвести курсор к редактируемой ячейке так, чтобы все ее содержимое было высвечено в реверсивном виде, а затем набрать (ввести) заменяемую информацию. Частичную замену данных можно осуществить двумя способами:

1. во-первых, щелкнуть в нужной ячейке, и она автоматически откроется для редактирования;
2. во-вторых, используя клавиши, переместиться в нужную ячейку, а затем нажать функциональную клавишу **F2**.

## УДАЛЕНИЕ ЗАПИСИ


Для удаления записи ее необходимо выделить (щелкнуть по области маркировки записи) и либо нажать клавишу **Del**, либо выполнить команду меню **Правка, Удалить**. В выводимом на экран запросе подтвердить удаление.

## 9. Модификация структуры таблиц

После создания базы данных и в процессе работы с ней проектировщик базы данных может обнаружить, что структура таблиц требует некоторых изменений. В этом случае ему необходимо выполнить модификацию ее структуры.



Под *модификацией структуры* таблицы будем понимать удаление существующих полей из таблицы, добавление новых полей, изменение имени, типа, размера, формата данных поля.

В Access модификация структуры таблицы осуществляется в режиме **Конструктор**. Для модификации структуры таблицы необходимо:

1. Открыть базу данных, содержащую таблицу, структуру которой необходимо изменить.
2. Щелкнуть мышью по нужной таблице в окне **База данных**.
3. Щелкнуть мышью по кнопке **Конструктор**  **Конструктор**, чтобы открыть описание структуры таблицы в режиме Конструктора.


**Пример.** Предположим, что в режиме

Конструктор необходимо модифицировать структуру таблицы «Страны», представленную на рисунке 4.15, в структуру, показанную на рисунке 4.16.

	
Рис. 4.15	Рис. 4.16

Проведя анализ двух структур, заметим, что нам требуется удалить из старой структуры поле **Часть света**, а в модифицированной структуре описать новое числовое поле **Плотность населения**.

Для удаления поля в режиме Конструктора следует:

1. Установить текстовый курсор в удаляемую строку **Часть света**.
2. Щелкнуть мышью по кнопке **Удалить строки**  на панели Стандартная или выполнить цепочку команд меню: **Правка Удалить строки**.
3. Щелкнуть мышью по кнопке **Да** в открывшемся окне (Рис. 4.17).

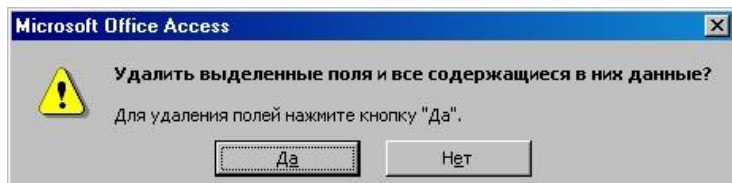



Рис. 4.17

Необходимо помнить, что при удалении поля, все уже содержащиеся в нем данные удаляются.



Для вставки поля в режиме Конструктора требуется:

1. Установить текстовый курсор в поле строки, над которой будет расположено новое поле. Для нашего примера это будет поле **Язык**.
2. Щелкнуть мышью по кнопке **Добавить строки**  на панели Стандартная или выполнить цепочку команд меню: **Вставка Строки**.
3. После вставки пустой строки она заполняется аналогично тому, как это делалось ранее при описании структуры таблицы.

Изменение имени поля осуществляется при редактировании столбца **Имя поля** (Рис. 4.16). Изменение типа данных в режиме Конструктора выполняется с помощью выпадающих меню столбца **Тип данных**, а изменение подтипов, формата и размера данных производится в окне **Свойства поля**.

1. Что понимается под модификацией структуры таблицы?
2. Как выполняется удаление полей из структуры таблицы в Access?

3. Как в Access добавляются новые поля в структуру таблицы?
4. Как изменяются имя, тип, размер, и формат данных поля в Access?

### Упражнения

Откройте готовые базы данных и внесите изменения в структуру таблиц.

- а) база данных «Страны» (файл Country.mdb), содержащая таблицу «Страны».

	Имя поля		Имя поля
	Страна		Страна
	Столица		Столица
	Население		Население
	Площадь (кв км)		Площадь (кв км)
	Язык		Язык
	Часть света		Флаг государства
Старая структура таблиц		Модифицированная структура таблицы	

- б) база данных «Магазин» (файл Магазин.mdb), содержащий таблицу «Товар».

товар : таблица	
Имя поля	Тип данных
Группа товара	Текстовый
Наименование товара	Текстовый
Количество	Числовой
Цена	Денежный
Дата поступления	Дата/время
Остаток товара	Логический
Веб-сайт описания	Гиперссылка

Измените формат данных поля **Дата поступления** с Краткого формата даты на Длинный формат даты.

в) база данных «Озера Беларуси» (файл Lake.mdb), содержащий таблицу «Озера».

<table border="1"> <thead> <tr> <th colspan="2">Озера : таблица</th> </tr> <tr> <th>Имя поля</th> <th></th> </tr> </thead> <tbody> <tr> <td>Название</td> <td></td> </tr> <tr> <td>Площадь (кв км)</td> <td></td> </tr> <tr> <td>Макс глубина ( м)</td> <td></td> </tr> <tr> <td>Объем воды (млн куб м)</td> <td></td> </tr> </tbody> </table>	Озера : таблица		Имя поля		Название		Площадь (кв км)		Макс глубина ( м)		Объем воды (млн куб м)		<table border="1"> <thead> <tr> <th colspan="3">Озера : таблица</th> </tr> <tr> <th>Имя поля</th> <th></th> <th>Тп</th> </tr> </thead> <tbody> <tr> <td>Название озера</td> <td></td> <td>Текс</td> </tr> <tr> <td>Площадь</td> <td></td> <td>Числ</td> </tr> <tr> <td>Длина</td> <td></td> <td>Текс</td> </tr> <tr> <td>Ширина</td> <td></td> <td>Текс</td> </tr> <tr> <td>Макс глубина</td> <td></td> <td>Числ</td> </tr> <tr> <td>Объем воды</td> <td></td> <td>Числ</td> </tr> </tbody> </table>	Озера : таблица			Имя поля		Тп	Название озера		Текс	Площадь		Числ	Длина		Текс	Ширина		Текс	Макс глубина		Числ	Объем воды		Числ
Озера : таблица																																					
Имя поля																																					
Название																																					
Площадь (кв км)																																					
Макс глубина ( м)																																					
Объем воды (млн куб м)																																					
Озера : таблица																																					
Имя поля		Тп																																			
Название озера		Текс																																			
Площадь		Числ																																			
Длина		Текс																																			
Ширина		Текс																																			
Макс глубина		Числ																																			
Объем воды		Числ																																			
<p>Старая таблицы</p>	<p>Модифицированная структура таблицы</p>																																				

В специальном столбце **Описание** укажите единицы измерения.

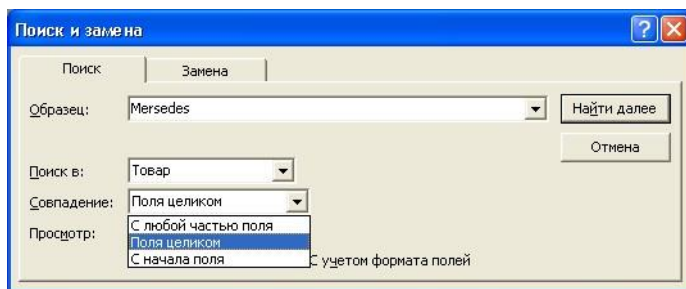
### 9.1. Поиск данных и замена данных

В режиме Таблицы предусмотрены возможности поиска данных по образцу либо в конкретном поле, либо во всей таблице.

Чтобы найти необходимые данные, нужно:

1. Открыть таблицу в режиме Таблицы.
2. Если известно, в каком столбце нужно вести поиск, выделить этот столбец (достаточно поместить курсор в любое поле этого столбца).
3. Нажать кнопку **Найти** (Find) на панели инструментов **Режим таблицы** (Table Datasheet) и открыть вкладку **Поиск** (Find) или выполнить команду меню **Правка,**

**Найти** (Edit, Find). Появится диалоговое окно **Поиск и замена** (Find and Replace) (рис. 2.54).



**Рис. 2.54.** Окно поиска данных

4. Ввести значение, которое требуется найти, в поле **Образец** (Find What). Если точное значение неизвестно, можно использовать подстановочные знаки.
5. Остальные параметры в окне можно изменить или оставить так, как они установлены по умолчанию:
  - значение в поле со списком **Поиск в** (Look In), определяющее место поиска, по умолчанию содержит название выбранного столбца;
  - значение в поле со списком **Совпадение** (Match) определяет один из трех вариантов совпадения образца со значением поля — **С любой частью поля** (Any Part of Field), **Поля целиком** (Whole Field), **С начала поля** (Start of Field);

- поле со списком **Просмотр** (Search), в котором можно задать направление поиска: **Все** (All), **Вверх** (Up), **Вниз** (Down);
- флажок **С учетом регистра** (Match Case) позволяет при поиске учитывать начертание букв — прописные или строчные;
- флажок **С учетом формата полей** (Search Fields As Formatted) позволяет выполнять поиск данных в указанном формате отображения.

Чтобы найти первое вхождение указанного образца, нажмите кнопку **Найти следующий** (Find Next).

Для поиска следующих вхождений этого значения нажимайте кнопку **Найти далее** (Find Next) до тех пор, пока не будет найдено нужное вхождение.

Для доступа к окну **Поиск и замена** (Find and Replace) можно использовать также команду меню **Правка, Найти** (Edit, Find).

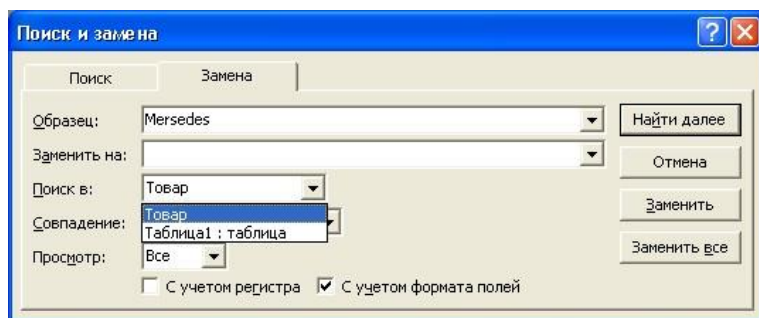
### **Замечание**

Из всех видов поиска самым медленным будет поиск по совпадению с любой частью поля, при этом не используется

индекс, даже если поле, по которому выполняется поиск, проиндексировано.

Чтобы выполнить поиск по всем столбцам таблицы, нужно из списка **Поиск в** (Look In) выбрать значение **<Имя таблицы>**: **таблица**.

Чтобы выполнить не только поиск, но и изменение данных в найденном поле, необходимо в том же диалоговом окне **Поиск и замена** (Find and Replace) раскрыть вкладку **Замена** (Replace) (рис. 2.55) и в поле **Заменить на** (Replace With) указать значение, на которое нужно заменить искомое значение. После этого нажмите кнопку **Заменить** (Replace) или **Заменить все** (Replace All).



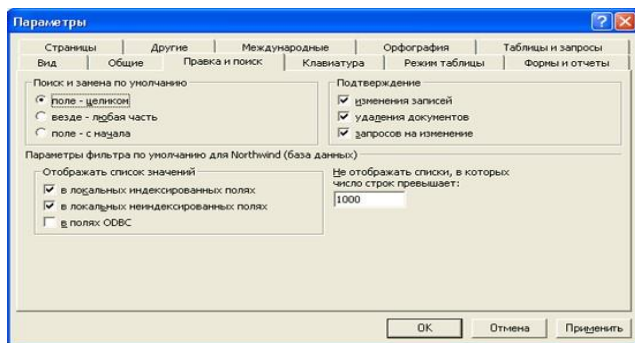
**Рис. 2.55.** Диалоговое окно **Поиск и замена**

Все остальные поля имеют тот же смысл, что и на вкладке **Поиск** (Find). Для замены одного значения необходимо нажать кнопку **Заменить** (Replace).

Чтобы заменить все вхождения, нажмите кнопку **Заменить все** (Replace All).

Есть возможность установить параметры поиска/замены по умолчанию. Для этого необходимо:

1. В меню **Сервис** (Tools) выбрать команду **Параметры** (Options).
2. В окне **Параметры** (Options) открыть вкладку **Правка и поиск** (Edit/Find) (рис.2.56).



**Рис. 2.56.** Настройка параметров поиска по умолчанию

3. В группе **Поиск и замена по умолчанию** (Default find/replace behavior) выбрать нужный тип поиска:



**поле - целиком** (Fast search), **езде - любая часть** (General search), **поле - с начала**(Start of field search).

4. Нажать кнопку **ОК**.

## **9.2. Проектирование многотабличной базы данных**

**Цель работы:** Получить навыки создания многотабличных баз данных в Microsoft Access **Задачи работы:**

1. Освоить приемы создания многотабличных баз данных.
2. Освоить приемы создания запросов.

**Теоретическое введение:** Реляционная база данных представляет собой совокупность связанных таблиц. Связи между таблицами устанавливаются через ключевые поля. В предыдущих работах рассмотрены приемы создания однотоабличных баз данных. В данной работе рассматриваются методы создания многотабличных баз данных

**Задание: создать базу данных «Учебный процесс» (рис.1).**

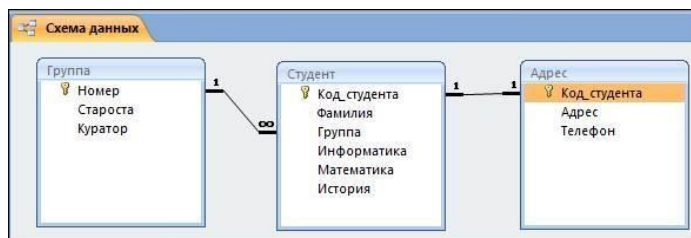


Рис. 1. Схема базы данных «Учебный процесс»

**Порядок выполнения работы.** Создать базу данных из трех таблиц: «Группа», «Студенты», «Адрес». Таблицы создаются в режиме «Конструктор». Поля таблиц указаны на рис.1. Технология создания таблицы рассматривалась в работе СУБД-1. Новые таблицы базы данных создаются командой «Таблица» на вкладке «Создание» (рис. 2).

Таблица «Группа» (рис. 1) состоит из полей: «Номер» - числовой тип; «Староста» - текстовый тип; «Куратор» - текстовый тип. Поле «Номер» является ключевым. Таблица «Студент» создавалась ранее. Таблица «Адрес» состоит из полей: «Код\_студента» - тип счетчик; «Адрес» - текстовый тип; «Телефон» - текстовый тип.

Первоначально необходимо создать пустые таблицы. При сохранении таблиц указывайте заданные имена. Ключевые поля таблиц задаются в соответствии с рис. 1, 3.

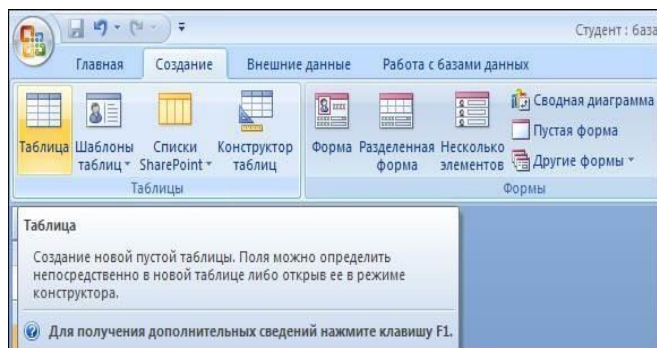


Рис. 2. Создание новой таблицы

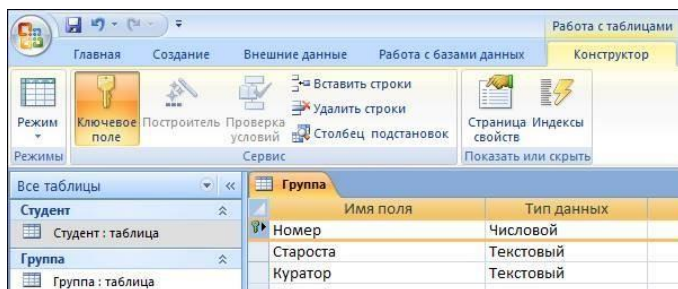


Рис. 3. Конструирование таблицы «Группа»

Для создания многотабличной базы данных необходимо выполнить связывание таблиц путем создания

схемы данных. Связывание таблиц производится средствами команды «Схема данных» на вкладке «Работа с базами данных» (рис. 4).

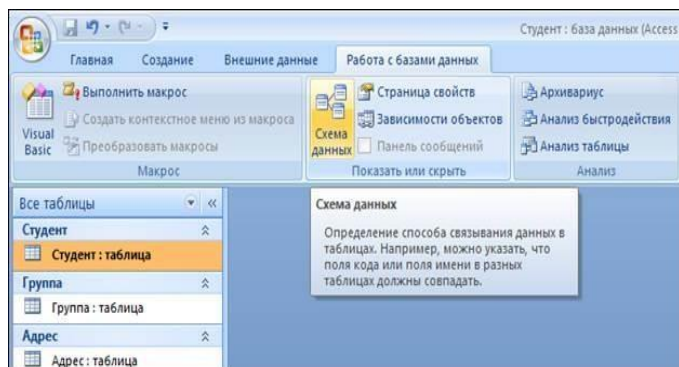


Рис. 4. Команда «Схема данных»

Таблицы последовательно добавляются в схему данных (рис. 5.). До создания схемы данных таблицы должны быть закрыты.

Теперь необходимо создать связи между таблицами (рис. 5.) и установить свойства связей. Связи создаются путем перетаскивания полей.

Для организации связи между полями «Номер» - «Группа» таблиц «Группа» и «Студент» (1:∞) необходимо задать режимы: «Обеспечение целостности данных», «Каскадное обновление» и «Каскадное удаление». Для связи

между полями «Код\_студента» таблиц «Студент» - «Адрес» (1:1) необходимо задать режимы: «Обеспечение целостности данных» и «Каскадное удаление» (рис. 6).

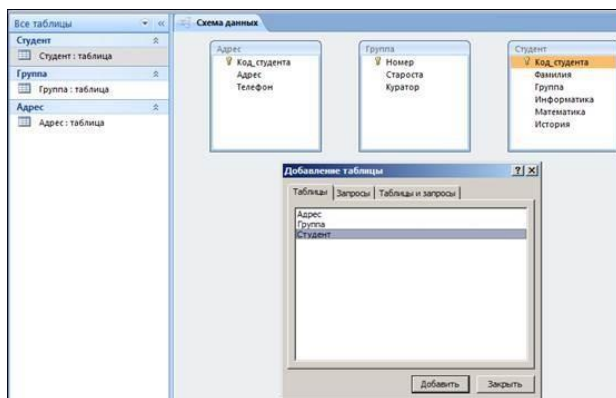


Рис. 5. Добавление таблиц в схему данных.

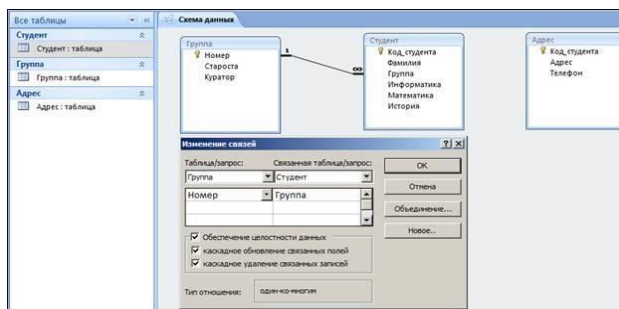


Рис. 6. Создание схемы данных

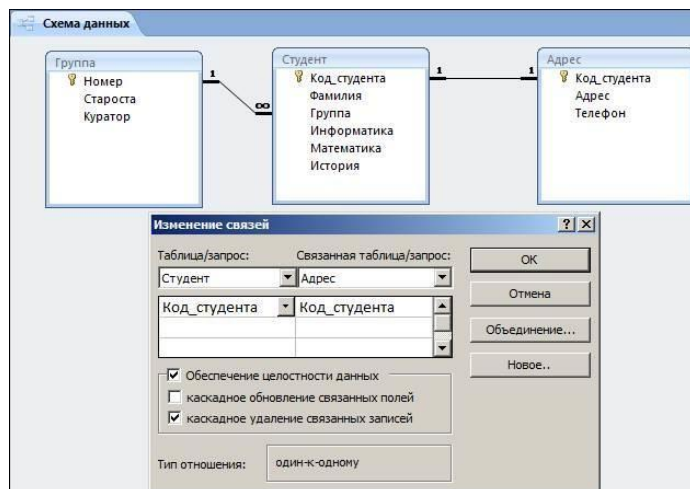


Рис. 7. Связывание таблиц «Студент»-«Адрес»

Режим «Обеспечение целостности данных» исключает в таблице «Студент» ввод номера, который не указан в таблице «Группа». Остальные режимы обеспечивают каскадное обновление и удаление записей через связанные поля.

Подведем итог: создана база данных из трех связанных пустых таблиц.

Следующим шагом является заполнение таблиц.

### **Контрольные вопросы:**

1. Назначение связей.
2. Что такое схема данных.
3. Способ связывания таблиц.
4. Эффект обеспечения целостности данных.
5. Что такое каскадное обновление и удаление связанных полей.

### **10. Выбор данных из таблиц при помощи запросов**

Рассмотрим запросы на выборку в Access на примере получения информации из таблицы ТОВАР базы данных *Поставка товаров*.

**Задача 1.** Пусть необходимо выбрать ряд характеристик товара по его наименованию.

1. Для создания запроса в окне базы данных выберите вкладку ленты — **Создание** (Create) и в группе **Запросы** (Queries) нажмите кнопку **Конструктор запросов** (Query Design). Откроется пустое окно запроса на выборку в режиме конструктора — **ЗапросN** (QueryN) и диалоговое окно **Добавление таблицы** (Show Table) (рис. 4.2).

2. В окне **Добавление таблицы** (Show Table) выберите таблицу ТОВАР и нажмите кнопку **Добавить**(Add). Выбранная таблица будет отображена в области схемы данных запроса. Закройте окно **Добавление таблицы** (Show Table), нажав кнопку **Заккрыть** (Close).

В результате выполненных действий в окне конструктора запросов (рис. 4.1) в верхней панели появится схема данных запроса, которая включает выбранные для данного запроса таблицы. В данном случае одну таблицу ТОВАР. Таблица представлена списком полей. Первая строка в списке полей таблицы, отмеченная звездочкой (\*), обозначает все множество полей таблицы. Нижняя панель является бланком запроса, который нужно заполнить.



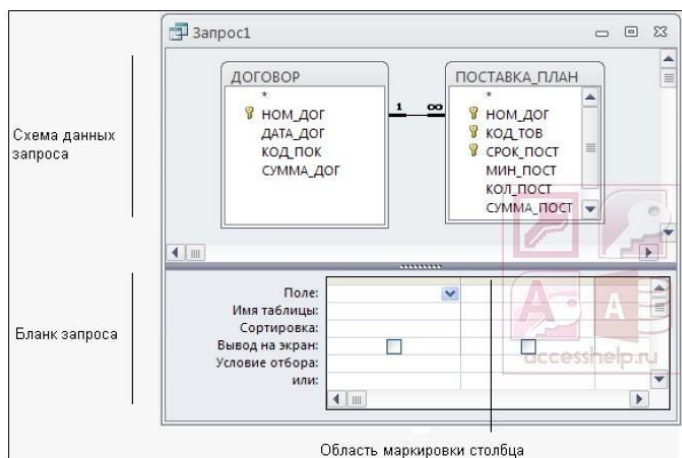


Рис. 4.1. Окно конструктора запросов

Кроме того, на ленте появляется и автоматически активизируется новая вкладка **Работа с запросами | Конструктор** (Query Tools | Design) (на рис. 4.3 представлена на часть этой вкладки), на которой цветом выделен тип созданного запроса — **Выборка** (Select). Таким образом, по умолчанию всегда создается запрос на выборку. Команды этой вкладки представляют инструментарий для выполнения необходимых действий при создании запроса. Эта вкладка открывается, когда в режиме конструктора создается новый запрос или редактируется существующий.

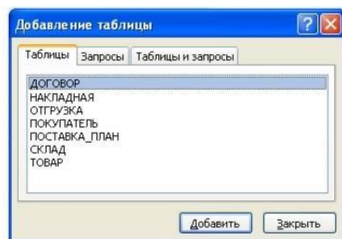


Рис. 4.2. Окно выбора таблиц и запросов для схемы данных запроса

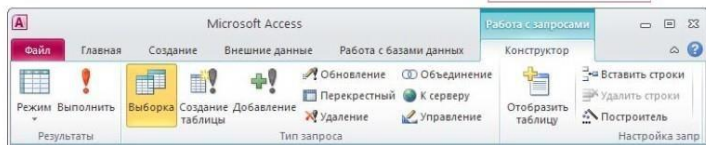


Рис. 4.3. Лента инструментов конструктора запросов

1. Для удаления любой таблицы из схемы данных запроса установите на нее курсор мыши и нажмите клавишу <Delete>. Для добавления — нажмите кнопку **Отобразить таблицу** ( Show Table) в группе **Настройка запроса** (Query Setup) на вкладке **Работа с запросами | Конструктор** (Query Tools | Design) или выполните команду **Добавить таблицу** (Show Table) в контекстном меню, вызываемом на схеме данных запроса.
2. В окне конструктора (рис. 4.4) последовательно перетащите из списка полей таблицы ТОВАР поля **НАИМ\_ТОВ**, **ЦЕНА**, **НАЛИЧИЕ\_ТОВ** в столбцы бланка запроса в строку **Поле** (Field).

3. Для включения нужных полей из таблицы в соответствующие столбцы запроса можно воспользоваться следующими приемами:
- в первой строке бланка запроса **Поле** (Field) щелчком мыши вызвать появление кнопки списка и выбрать из списка нужное поле. Список содержит поля таблиц, представленных в схеме данных запроса;
  - дважды щелкнуть на имени поля таблицы в схеме данных запроса;
  - для включения всех полей таблицы можно перетащить или дважды щелкнуть на символе \* (звездочка) в списке полей таблицы в схеме данных запроса.
4. Если вы по ошибке перетащили в бланке запроса ненужное поле, удалите его. Для этого переместите курсор в область маркировки столбца сверху, где он примет вид черной стрелки, направленной вниз, и щелкните кнопкой мыши. Столбец выделится. Нажмите клавишу <Delete> или выполните команду **Удалить столбцы** (Delete Columns) в группе **Настройка запроса** (Query Setup).

5. В строке **Вывод на экран** (Show) отметьте поля, иначе они не будут включены в таблицу запроса.

6. Запишите в строке **Условия отбора** (Criteria) наименование товара, как показа-но в бланке запроса на рис. 4.4. Так как выражение в условии отбора не содержит оператора, то по умолчанию используется оператор =. Используемое в выражении текстовое значение вводится в двойных кавычках, которые добавляются автоматически.

7. Выполните запрос, щелкнув на кнопке **Выполнить** (Run) или на кнопке **Режим** (View) в группе **Результаты** (Results). На экране появится окно запроса в режиме таблицы с записью из таблицы **ТОВАР**, отвечающей заданным условиям отбора.

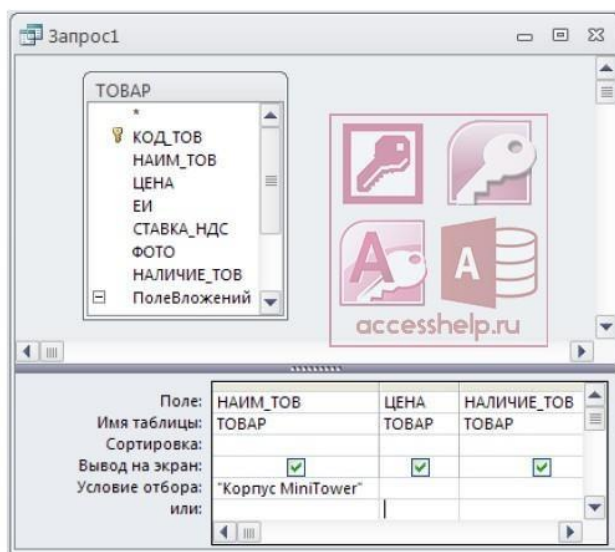


Рис. 4.4. Окно конструктора запроса на выборку

### ***ЗАМЕЧАНИЕ***

Окно запроса в режиме таблицы аналогично окну просмотра таблицы базы данных. Через некоторые таблицы запроса может производиться изменение данных базовой таблицы, лежащей в основе запроса. Запрос, просматриваемый в режиме таблицы, в отличие от таблицы базы данных Access 2016, не имеет столбца **Щелкнуть для добавления** (Click to Add), предназначенного для изменения структуры таблицы. В этом режиме на вкладке ленты

**Главная** (Home) доступны те же кнопки, что и при открытии таблицы базы данных.

Если при вводе сложного наименования товара вы допустили неточность, товар не будет найден в таблице. Использование операторов шаблона — звездочка (\*) и вопросительный знак (?) (стандарт ANSI-89, используемый для запросов по умолчанию) или знак процента (%) и подчеркивания (\_) (ANSI-92, рекомендуемый как стандарт для SQL Server), упрощает поиск нужных строк и позволяет избежать многих ошибок. Введите вместо полного имени товара Корпус\* или Корпус%. Выполните запрос. Если в поле наименования товара одно значение начинается со слова «Корпус», результат выполнения запроса будет таким же, как в предыдущем случае. После выполнения запроса введенное выражение будет дополнено оператором Like «Корпус\*». Этот оператор позволяет использовать символы шаблона при поиске в текстовых полях.

Если необходимо найти несколько товаров, используйте оператор In. Он позволяет выполнить проверку на равенство любому значению из списка, который задается в круглых скобках. Запишите в строке условий отбора In («Корпус MiniTower»;»HDD Maxtor 20GB»;»FDD 3,5"). В

таблице запроса будет выведено три строки. В операторе In не допускается использование символов шаблона.

Сохраните запрос, щелкнув на вкладке **Файл** (File) и выполнив команду **Сохранить** (Save). В окне **Сохранение** (Save As) введите имя запроса **Пример1**. Заметим, что имя запроса не должно совпадать не только с именами имеющихся запросов, но и с именами таблиц в базе данных.

1. Закройте текущий запрос по команде контекстного меню **Заккрыть** (Close) или нажав кнопку окна запроса **Заккрыть** (Close).
2. Выполните сохраненный запрос, выделив запрос в области навигации и выбрав в контекстном меню команду **Открыть** (Open).
3. Для редактирования запроса выделите его в области навигации и выполните в контекстном меню команду **Конструктор** (Design View).

## **11.Выполнение вычислительных операций в таблицах**

Определение связей между таблицами После того как в БД созданы основные таблицы, следует указать, как они

связаны друг с другом. Эти связи Access будет использовать в запросах, формах и отчетах при отборе информации из нескольких таблиц. Задание связей между таблицами позволяет также обеспечить защиту целостности данных в БД. Связь между двумя таблицами, одна из которых является главной, а 30 Работа в Microsoft Access поля (группы полей) связи выбирается ключевое поле таблицы, но достаточно, чтобы оно имело уникальный индекс. В соответствии ему ставится поле (группа полей) подчиненной таблицы, называемое внешним ключом.

Связь между записями устанавливается по совпадению значений в полях связи. Access анализирует значение поля связи (ключа) любой записи главной таблицы и связывает ее со всеми записями подчиненной таблицы, имеющими такое же значение внешнего ключа. Поля связи могут иметь разные имена, но они должны иметь один тип данных и иметь однотипное содержимое. Исключение из этого правила: поле типа Счетчик можно связывать с числовым полем, имеющим в свойстве Размер поля значение «Длинное целое».

Кроме того, связываемые поля числового типа должны иметь одинаковые значения свойства Размер поля. Например, для получения информации о студентах и полученных ими оценках следует определить связь по полям



Код Студента в таблицах Студенты и Сессия. Главной в этой связи будет таблица Студенты, а подчиненной — таблица Сессия. Каждой записи в главной таблице — данным о студенте — соответствует несколько (или ни одной) записей из подчиненной таблицы — информация об его оценках.

Поле Код Студента в таблице Сессия является внешним ключом.

### Типы связей между таблицами

1. Связь типа «один-ко-многим» Описанная выше связь между таблицами Студенты и Сессия — пример связи типа «один-ко-многим». Это наиболее распространенный тип связи. При таком типе связи каждой записи в главной таблице могут соответствовать одна, несколько или ни одной записи в подчиненной таблице, а каждая запись в подчиненной таблице не может иметь более одной соответствующей ей записи в главной таблице. Если запись в подчиненной таблице не имеет соответствующей ей записи в главной таблице, то значение поля связи в такой записи должно быть пусто.

2. Связь типа «один-к-одному» При связи типа «один-к-одному» каждой записи в главной таблице может соответствовать не более одной записи в подчиненной таблице, и наоборот, каждая запись в подчиненной таблице не может иметь более одной соответствующей ей записи в главной таблице.

Как и в предыдущем случае, если запись в подчиненной таблице не имеет соответствующей ей записи в главной таблице, то значение поля связи в такой записи должно быть пусто.

Связь типа «один-к-одному» обычно спользуют для разделения таблиц, имеющих много полей, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице. Например, такой тип связи использован при установлении связей между таблицами Студенты и Общежитие.

3. Связь типа «многие-ко-многим» При установлении связи между таблицами возможна ситуация, когда между ними нельзя установить отношение «главная-подчиненная» из-за того, что любой записи в одной таблице может соответствовать несколько записей из другой таблицы.

Примером могут служить таблицы Студенты и Преподаватели, так как каждый студент сдавал экзамены нескольким преподавателям, а каждый преподаватель принимал экзамен у нескольких студентов. Поэтому между этими таблицами нельзя установить ни одну из описанных выше связей. Это пример связи типа «многие-ко-многим». Access непосредственно не поддерживает такой тип связи. Но ее можно реализовать в виде двух связей типа «один-ко-многим» с помощью третьей (связующей) таблицы. В качестве такой связующей таблицы в БД Деканат используется таблица Сессия. Она связана как с таблицей Студенты по полю Код студента, так и с таблицей Преподаватели по полю Код преподавателя. После того как эти связи установлены, можно легко определить с помощью. Связи между таблицами в БД Деканат 32 Работа в Microsoft Access соответствующих запросов, у каких студентов принимал экзамены данный преподаватель или кому сдавал экзамены данный студент.

**Создание связей.** Для создания связей между таблицами нужно вернуться в окно БД, закрыть все открытые таблицы и щелкнуть по кнопке Схема данных панели

инструментов либо вызвать щелчком правой кнопки мыши контекстное меню и выбрать в нем пункт Схема данных.

Для изменения поля следует открыть список полей справа от его имени. Если связь производится по нескольким полям, то их имена можно добавить, используя пустые строки. Обычно Access сам определяет тип создаваемой связи, проводя анализ полей, для которых определяется связь. Если только одно из полей является ключевым или имеет уникальный индекс, создается связь «одинкомногим». Связь «один-к-одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.

Создавая связь, нужно настроить режим обеспечения целостности данных. Обеспечение целостности позволяет избежать ситуации, когда в подчиненной таблице имеются записи, не связанные с записями главной таблицы. Если этот режим включен, то Access не разрешит добавить в подчиненную таблицу запись, для которой не найдется связанной с ней записи из главной таблицы. Нельзя будет также удалить из главной таблицы запись, имеющую связанные с ней записи в подчиненной таблице. Например, нельзя будет добавить в таблицу Сессия запись с кодом студента, отсутствующим в таблице Студенты.

Соответственно, из таблицы Окно Связи 34 Работа в Microsoft Access Студенты нельзя удалить запись о студенте, пока в таблице Сессия содержатся сведения о его оценках.

Режим обеспечения целостности данных этой связи можно включить, если выполнены следующие условия:

- поле связи главной таблицы является первичным ключом или имеет уникальный индекс;
- связанные поля имеют один и тот же тип данных;
- обе связанные таблицы принадлежат одной базе данных Access.

Если для связи включен этот режим, то можно дополнительно указать, следует ли автоматически выполнять для связанных записей операции каскадного обновления и каскадного удаления. Если включить режим Каскадное обновление связанных полей, то при изменении значения ключа в главной таблице будут автоматически обновлены соответствующие значения в связанных записях подчиненной таблицы. При включении режима Каскадное удаление связанных записей при удалении записи из главной таблицы будут автоматически удалены связанные с ней записи в подчиненной таблице. В том случае, когда эти

режимы не включены, а режим обеспечения целостности данных включен, Access не позволит изменить значение в ключевом поле главной таблицы, а также удалить запись в главной таблице, если в подчиненной таблице имеются данные, связанные с этой записью. После завершения операции создания связи нужно нажать кнопку ОК.

Связь отображается в виде линии, соединяющей две таблицы. Если включен режим обеспечения целостности данных, то Access изобразит на конце линии, соответствующей главной таблице, цифру 1. На другом конце линии, соответствующем подчиненной таблице, будет изображен символ бесконечности  $\infty$  для связи типа «один-ко-многим» и цифра 1 для связи типа «один-к одному». Если перенести с помощью мыши поле, не являющееся ключевым или не имеющее уникального индекса, на другое поле, которое также не является ключевым или не имеет уникального индекса, будет создана связь неопределенного типа. Режим обеспечения целостности данных в этом случае включить нельзя. Любую связь можно изменить или удалить. Для изменения связи нужно сделать двойной щелчок по линии связи, и отредактировать ее в открывшемся окне Связи. Чтобы удалить связь, следует щелкнуть по ней и нажать клавишу Delete.

## **12. Форматы в таблицах и способ их создания**

Операции над таблицами Мы рассмотрим следующие операции над таблицей: копирование, переименование, удаление и экспорт. Все эти операции производятся в окне БД. Копирование таблицы Чтобы создать копию таблицы в БД, нужно вызвать ее контекстное меню и выбрать пункт Копировать (другой способ — выделить щелчком мыши таблицу и нажать клавиши Ctrl+C или щелкнуть по кнопке Копировать панели инструментов). Затем нужно нажать клавиши Ctrl+V или щелкнуть по кнопке Вставить панели инструментов. Access откроет диалоговое окно Вставка таблицы и предложит ввести имя новой таблицы; выбрать режим копирования.

В зависимости от того, какой режим выбран, Access создаст пустую таблицу, имеющую такую же структуру, что и исходная таблица; создаст точную копию исходной таблицы (этот режим задан по умолчанию) или добавит данные из исходной таблицы в указанную таблицу. Переименование таблицы Для переименования таблицы нужно вызвать ее контекстное меню и выбрать пункт Переименовать. Затем нужно ввести новое имя таблицы и щелкнуть по кнопке ОК. Удаление таблицы Чтобы удалить

ставшую ненужной таблицу в БД, нужно вызвать ее контекстное меню и выбрать пункт Удалить (другой способ — выделить щелчком мыши таблицу и нажать клавишу Delete). Access предложит подтвердить ваше желание удалить таблицу и в случае получения положительного ответа удалит ее.

Работа в Microsoft Access. **Экспорт таблицы.** Для экспорта таблицы Access в формат Excel или Word проще всего выполнить такие действия:

1. Щелчком мыши выделить нужную таблицу.
2. Выбрать пункт меню Сервис, а затем — Связи с Office или щелкнуть мышью по правому краю кнопки Связи с Office .
3. Выбрать команду Анализ в MS Excel или соответственно команду Публикация в MS Word. Access скопирует таблицу в электронную таблицу Excel (таблицу Word в формате RTF) с таким же именем. Ее первая строка будет содержать названия (подписи) полей экспортируемой таблицы.

Если файл с таким именем уже существует, то Access предложит заменить этот файл или изменить имя нового



файла. Можно использовать более универсальный способ экспорта, позволяющий экспортировать таблицу Access в другую БД (Access, dBase, FoxPro, Paradox) или файл другого формата (текстовый, документ Word, таблица Excel, документ HTML). Для этого нужно выполнить следующие действия:

1. Вызвать контекстное меню таблицы или открыть меню Файл и выбрать пункт Сохранить как/экспорт...
2. В диалоговом окне Сохранение объекта следует выбрать параметр Во внешнем файле или базе данных и нажать кнопку ОК.
3. В поле со списком Тип файла выбрать формат БД или файла.
4. В поле со списком Папка выбрать диск и папку, в которую следует поместить экспортируемую таблицу.
5. В поле Имя файла нужно указать имя, которое получит экспортируемая таблица, и нажать кнопку Экспорт.

Экспорт в другую БД или в формат HTML Access осуществляет автоматически, создавая файл в заданном

формате. Если этот формат не поддерживает длинные имена таблиц или кириллицу в названиях полей (FoxPro), то перед осуществлением операции следует внести изменения в структуру исходной таблицы. Если задать экспорт таблицы в текстовый файл, то будет вызван мастер Экспорт текста, который позволит выбрать формат текста (Windows или DOS), настроить формат вывода дат и чисел и указать, какие поля нужно экспортировать. Мастер дает возможность создать текстовый файл с полями фиксированной ширины или с разделителями полей.

### **13. Создание иерархических форм межтабличных связей**

Модификация структуры таблицы Все описанные ниже операции по модификации структуры таблицы производятся в режиме конструктора таблицы. Вставка полей Чтобы вставить в таблицу новое поле, нужно установить курсор на строку определения поля, перед которым вы хотите вставить новое поле. Затем следует щелкнуть по кнопке Добавить строки или выбрать команды Вставка/Строки. Access вставит пустую строку, которую можно использовать для определения нового поля. Введите его имя, укажите тип и

задайте нужные свойства. Удаление полей Для удаления поля нужно выделить строку его определения, щелкнув по области маркировки строки.

Область выделения можно расширить, используя клавиши Ctrl и Shift. Затем нужно нажать клавишу Delete или щелкнуть по кнопке Удалить строки . Если таблица содержит данные в удаляемом поле (полях), то Access и их потеряю. Поэтому перед такой операцией имеет смысл на всякий случай создать копию таблицы. Перемещение полей Чтобы переместить поле, нужно выделить строку его определения и, держа нажатой мышью, переместить эту строку в другое место области определения полей. Можно поступить иначе: с помощью клавиш Ctrl+F8 включить режим Сдвиг (в строке состояния появится индикатор СДВ) и затем для перемещения поля использовать клавиши ↓ и ↑. Для выключения этого режима нужно нажать на клавишу Esc.

Аналогично можно переместить выделенную группу полей. Копирование полей Если несколько полей таблицы имеют близкие определения, то можно создать одно поле, а затем скопировать его нужное число раз. Для выполнения операции копирования поля нужно выделить целиком строку определения поля и скопировать в буфер обмена, используя

клавиши Ctrl+C или кнопку Копировать панели инструментов. Затем вставить пустую строку (см. пункт Вставка полей) в то место определения таблицы, где должна находиться копия поля, и выполнить операцию вставки из буфера обмена, используя клавиши Ctrl+V или кнопку Вставить панели инструментов.

После этого вносятся необходимые изменения в название и свойства нового поля.

**Работа в Microsoft Access Изменение типа данных и размера поля.** Для изменения типа данных нужно щелкнуть по ячейке типа данных изменяемого поля. Затем щелкнуть по кнопке справа, открывающей список возможных типов данных, и выбрать новый тип данных. С некоторыми ограничениями Access может успешно преобразовать любой тип данных (кроме типа поле объекта OLE). Например, при преобразовании из текстового в числовой формат поле должно содержать только цифры и допустимые разделители. Для текстового и числового поля задается его размер. Установленный размер поля можно затем изменить.

Если он будет уменьшен, то Access выдаст предупреждающее сообщение о возможной потере некоторых данных. Если вы подтвердите свое желание

сохранить сделанные изменения в определении таблицы, то операция будет выполнена, но это может повлечь за собой ошибки в преобразовании данных и их потерю. Поэтому перед такой операцией имеет смысл на всякий случай создать копию таблицы.

#### **14. Отчеты и способы их создания**

Отчеты — форма представления информации для использования и распространения. Отчеты — итоговые документы для лиц, которым была предназначена создаваемая БД. Если форма — документ разработчика и лиц, работающих с информацией в БД, то отчеты — инструмент «хозяев» БД, позволяющий им в нужный момент иметь информацию для личного использования или передачи другим лицам. Отчет нужен для подведения итогов деятельности за период подсчетов итоговых сумм и т.д. Структура отчета (см. рис. 7.1) напоминает структуру формы за одним существенным исключением — возможностью добавления нескольких пар новых разделов, если возникает потребность группировки данных по каким-либо признакам.

Заголовок отчета Верхний колонтитул Область данных  
Нижний колонтитул Примечание отчета Рис. 7.1. Структура

отчета Возможность группирования данных, - это, пожалуй, главное отличие создания отчета от создания формы. Как реализовать группирование, мы подробно рассмотрим ниже, а пока кратко рассмотрим саму идею группирования и ее влияние на структуру разделов. Предположим, что мы строим отчет по таблице, описывающей реализацию товаров различным клиентам. Можно составить отчет по разным «срезам» такой таблицы. Например, составить отчет, характеризующий заказы каждого клиента. В этом случае основа 184.

**Работа в Microsoft Access группирования данных** — клиенты. Все заказы разбиваются на группы, относящиеся к одному клиенту. Другой вариант — рассмотреть, как реализовывались товары различных видов. В этом случае основа группирования — отдельный товар, а все данные о заказах можно разбить на группы, относящиеся к конкретному товару. Группы могут быть вложенными. Например, в первом случае, внутри группирования по клиентам, можно сгруппировать заказы каждого клиента по отдельным товарам. Каждое группирование сопровождается, как правило, появлением двух разделов — заголовка группы и примечания группы.

В заголовке обычно указывается информация о поле — источнике группирования (клиенты, товары) и др. В области примечания можно подвести итоги группирования — число заказов каждого клиента, сумму стоимости всех заказов каждого клиента, число заказов каждого товара и т. д. Если группы вложенные, можно подвести промежуточные итоги, т. е. число заказов конкретного товара, сделанных клиентом. Общее число вложений групп — до 10. Свойства разделов отчета совпадают со свойствами разделов формы, хотя у форм и отчетов существуют различные режимы. Напомним, что форма могла находиться в трех режимах: конструктора, работы (формы) и предварительного просмотра. У отчета тоже может быть три режима, причем два из них такие же: режим предварительного просмотра и режим конструктора.

Кроме того, отчет может находиться в режиме образца. Этот режим похож на режим предварительного просмотра, но отличается от него тем, что в режиме образца показывается не весь отчет, а только его часть, с целью оценки того, как будет выглядеть весь отчет. Режим, подобный режиму формы, естественно отсутствует, так как с отчетом не работают как с формой или запросом. Некоторые разделы отчета, такие как колонтитулы и примечания, имеют

дополнительные свойства, более тонкие, чем у аналогичных разделов форм. Так, заголовок отчета может быть напечатан отдельно на первой странице, без колонтитулов. Надо отметить, что в отчете дублируются и многие другие свойства различных объектов, изучавшихся нами в формах.

В отчетах, как и в формах, размещаются элементы управления: поля, надписи и т. д. Как и в формах, они могут быть связанными, свободными и вычисляемыми в зависимости от источника информации, реализуемой в элементе управления. Отчеты 185 7.2. Создание отчета

Отчеты, как и формы, можно создавать с помощью мастеров или самостоятельно. Источником данных для отчета также являются таблицы и запросы. Если в отчете требуется представить данные из разных таблиц, имеет смысл предварительно создать многотабличный запрос, а затем строить отчет на его основе. Чтобы создать отчет, нужно в окне БД перейти на вкладку Отчеты и нажать кнопку Создать.

Появится окно диалога Новый отчет. Оно аналогично окну создания формы. Ваши действия начинаются с выбора таблицы/запроса.



Предусмотрено автоматическое создание двух типов отчетов: ленточного и в один столбец. В этих отчетах имеется заголовок, содержащий имя отчета, совпадающее с названием таблицы или запроса, и дата создания отчета. Нижний колонтитул содержит номер страницы. В области данных содержатся все поля таблицы (запроса), расположенные в один столбец или в табличном виде (в ленточном отчете).

На одной странице реализуется Рис. 7.1. Окно диалога Новый отчет 186 Работа в Microsoft Access несколько записей таблицы (запроса).

## **15. Макросы и их способ создания**

Язык макросов является языком программирования, который позволяет реализовать задачи пользователя, выполняя необходимые действия над объектами Access и их элементами. Макрос — программа, состоящая из последовательности макрокоманд. Макрокоманда — это инструкция, ориентированная на выполнение определенного действия.

Например, макрокомандой можно открыть форму, отчет, напечатать отчет, запустить на выполнение запрос,

применить фильтр, присвоить значение, создать свое меню для формы или отчета. Макрокоманда Задать Команду Меню позволяет выполнить любую заданную команду меню. Имеющийся в Access набор макрокоманд реализует практически любые действия, которые необходимы для решения задачи.

Язык макросов обеспечивает возможность выполнения большинства задач, не прибегая к программированию на Visual Basic. Макросы, являясь надстройкой над Visual Basic, обеспечивают пользователя средствами решения задач, не требующими знания детального программирования.

Язык макросов является языком более высокого уровня, чем Visual Basic.

Макросы определяют набор методов обработки объектов. В Access имеются средства, обеспечивающие взаимодействие макросов с объектами на основе событий. События наступают при выполнении определенных действий, к которым относятся прежде всего действия пользователя. Последнее обстоятельство позволяет управлять выполнением программы извне. Пользователь, выполняя различные действия в формах, инициирует выполнение макросов, автоматизирующих решение связанных с действиями пользователя подзадач.

Такой подход существенно отличает программирование на языке макросов от программирования в традиционном понимании, при котором только программа управляет процессом.

Заметим, что наличие аппарата запуска программ через события, вызываемые действиями пользователя, не исключает возможности написания программ на языке макросов, которые без вмешательства пользователя могут решать нужные задачи. В этом случае задача решается рядом взаимосвязанных макросов.

Для организации ветвлений в программе в макросе определяются условия выполнения макрокоманд. Пользователь запускает главную программу на выполнение и далее все управление ее выполнением осуществляется изнутри. Программа сама открывает необходимые объекты, выбирает и обрабатывает данные, вызывает другие макросы, следуя алгоритму, приводящему к решению задачи.

При необходимости программа инициирует диалог с пользователем.

## КОНСТРУИРОВАНИЕ МАКРОСА

Создание макросов осуществляется в диалоговом режиме и сводится к записи в окне макроса последовательности макрокоманд, для которых задаются параметры. Каждому макросу присваивается имя. При выполнении макроса макрокоманды выполняются последовательно в порядке их расположения. При этом используются объекты или данные, указанные в аргументах макрокоманд. Имеется возможность определить условия выполнения макрокоманды.

Выполнение макросов инициируется простой операцией и может сводиться к его открытию, как это делается и для других объектов базы данных. Помимо этого, Access предоставляет возможность автоматически инициировать выполнение макроса при наступлении некоторого события. Для связи макроса с событием достаточно в бланке свойств объекта или его элемента управления внести в строку этого события имя макроса. События, с которыми можно связать макрос, определяются в свойствах форм и отчетов и их элементах управления.

Создание макроса начинается в окне базы данных, где надо выбрать закладку Макросы (Macros) и

нажать кнопку Создать (New). После этого открывается окно макроса.

### **15.1. ФОРМИРОВАНИЕ МАКРОКОМАНД В ОКНЕ МАКРОСА**

Макрокоманды, составляющие макрос, вводятся в столбец Макрокоманда. Для ввода макрокоманды можно нажать кнопку раскрытия списка макрокоманд в этом столбце и выбрать нужную. Можно ввести имя макрокоманды с клавиатуры. В нижней части окна выводятся аргументы макрокоманды. Значения аргументов задаются путем выбора их из списка, открывающегося в строке аргумента. В поле справа от строк аргументов выводится сообщение с пояснениями для выбранного аргумента.

В строки столбца Примечание вводят необязательный комментарий, описывающий результат выполнения макрокоманды. Каждая новая макрокоманда макроса добавляется в следующей строке бланка. Порядок размещения макрокоманд в бланке определяет последовательность их выполнения.

После ввода всех макрокоманд в макрос его надо сохранить, воспользовавшись командой меню Файл/Сохранить или кнопкой панели инструментов макроса.

Для выполнения макроса можно нажать кнопку панели инструментов Запуск. Если макрос уже закрыт, то для его выполнения надо в окне базы данных выбрать макрос и нажать кнопку Запуск в этом окне. Таким образом, по инициативе пользователя задача может решаться многократно.

Для просмотра и редактирования существующего макроса надо выбрать его в окне базы данных и нажать кнопку Конструктор.

## **15.2. ПРИМЕР РАЗРАБОТКИ МАКРОСА**

Пусть необходимо рассчитать количество студентов в группах и внести эти данные в поле КОЛ таблицы ГРУППА. Подсчет количества студентов реализован запросом Число студентов в группе. Обновление поля КОЛ на основе полученных в этом запросе данных выполняет запрос Обновление ГРУППА\_КОЛ. Последовательное выполнение этих запросов решает поставленную задачу.

1. Сначала создадим запрос на обновление. Оформим окно конструктора запроса как показано на рисунке:



В схему данных запроса добавлены таблица Группа и таблица Число студентов, полученная после выполнения запроса на создание Число студентов в группе.

2. Преобразуем запрос на выборку в запрос на обновление. Для этого выполним команду меню Запрос/Обновление. В строке Обновление введем имя поля [CountИС] таблицы Число студентов, из которой выбираются значения для обновления.

Сохраним запрос на обновление под именем Обновление ГРУППА\_КОЛ.

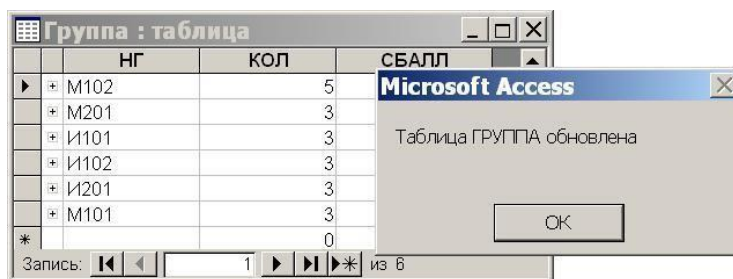
3. Создадим макрос, состоящий из макрокоманд, показанных в окне конструктора:



В столбце Примечания записаны комментарии к макрокомандам. В зависимости от выбранной макрокоманды меняется нижняя часть окна конструктора — Аргументы макрокоманды. На рисунке активна макрокоманда Сдвиг Размера, которая устанавливает положение и размеры окна таблицы Группа, демонстрируемого после ее обновления.



Сохраним макрос под именем Расчет количества студентов в группах. После нажатия кнопки Запуск получим результаты выполнения макроса, показанные на рисунке.



Поверх окна Группа выводится сообщение, в соответствии с последней макрокомандой.

Язык макросов является языком программирования, который позволяет реализовать задачи пользователя, выполняя необходимые действия над объектами Access и их элементами.

Макрос — программа, состоящая из последовательности макрокоманд. Макрокоманда — это инструкция, ориентированная на выполнение определенного действия.

Например, макрокомандой можно открыть форму, отчет, напечатать отчет, запустить на выполнение запрос,

применить фильтр, присвоить значение, создать свое меню для формы или отчета. Макрокоманда Задать Команду Меню позволяет выполнить любую заданную команду меню.

Имеющийся в Access набор макрокоманд реализует практически любые действия, которые необходимы для решения задачи.

Язык макросов обеспечивает возможность выполнения большинства задач, не прибегая к программированию на Visual Basic. Макросы, являясь надстройкой над Visual Basic, обеспечивают пользователя средствами решения задач, не требующими знания детального программирования.

Язык макросов является языком более высокого уровня, чем Visual Basic.

Макросы определяют набор методов обработки объектов. В Access имеются средства, обеспечивающие взаимодействие макросов с объектами на основе событий. События наступают при выполнении определенных действий, к которым относятся прежде всего действия пользователя. Последнее обстоятельство позволяет управлять выполнением программы извне. Пользователь, выполняя различные действия в формах, инициирует выполнение макросов, автоматизирующих решение связанных с действиями пользователя подзадач.

Такой подход существенно отличает программирование на языке макросов от программирования в традиционном понимании, при котором только программа управляет процессом.

Заметим, что наличие аппарата запуска программ через события, вызываемые действиями пользователя, не исключает возможности написания программ на языке макросов, которые без вмешательства пользователя могут решать нужные задачи. В этом случае задача решается рядом взаимосвязанных макросов.

Для организации ветвлений в программе в макросе определяются условия выполнения макрокоманд. Пользователь запускает главную программу на выполнение и далее все управление ее выполнением осуществляется изнутри. Программа сама открывает необходимые объекты, выбирает и обрабатывает данные, вызывает другие макросы, следуя алгоритму, приводящему к решению задачи. При необходимости программа инициирует диалог с пользователем.

### 15.3. КОНСТРУИРОВАНИЕ МАКРОСА

Создание макросов осуществляется в диалоговом режиме и сводится к записи в окне макроса последовательности макрокоманд, для которых задаются параметры. Каждому макросу присваивается имя. При выполнении макроса макрокоманды выполняются последовательно в порядке их расположения. При этом используются объекты или данные, указанные в аргументах макрокоманд. Имеется возможность определить условия выполнения макрокоманды.

Выполнение макросов инициируется простой операцией и может сводиться к его открытию, как это делается и для других объектов базы данных. Помимо этого, Access предоставляет возможность автоматически инициировать выполнение макроса при наступлении некоторого события.

Для связи макроса с событием достаточно в бланке свойств объекта или его элемента управления внести в строку этого события имя макроса. События, с которыми можно связать макрос, определяются в свойствах форм и отчетов и их элементах управления. Создание макроса начинается в окне базы данных, где надо выбрать закладку

Макросы (Macros) и нажать кнопку Создать (New). После этого открывается окно макроса.

#### **15.4. ФОРМИРОВАНИЕ МАКРОКОМАНД В ОКНЕ МАКРОСА**

Макрокоманды, составляющие макрос, вводятся в столбец Макрокоманда. Для ввода макрокоманды можно нажать кнопку раскрытия списка макрокоманд в этом столбце и выбрать нужную. Можно ввести имя макрокоманды с клавиатуры. В нижней части окна выводятся аргументы макрокоманды. Значения аргументов задаются путем выбора их из списка, открывающегося в строке аргумента. В поле справа от строк аргументов выводится сообщение с пояснениями для выбранного аргумента.

В строки столбца Примечание вводят необязательный комментарий, описывающий результат выполнения макрокоманды. Каждая новая макрокоманда макроса добавляется в следующей строке бланка. Порядок размещения макрокоманд в бланке определяет последовательность их выполнения.

После ввода всех макрокоманд в макрос его надо сохранить, воспользовавшись командой меню Файл/Сохранить или кнопкой панели инструментов макроса.

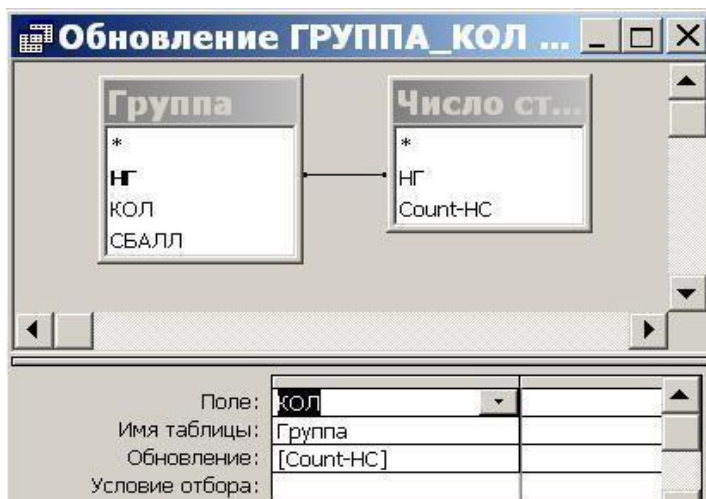
Для выполнения макроса можно нажать кнопку панели инструментов Запуск. Если макрос уже закрыт, то для его выполнения надо в окне базы данных выбрать макрос и нажать кнопку Запуск в этом окне. Таким образом, по инициативе пользователя задача может решаться многократно.

Для просмотра и редактирования существующего макроса надо выбрать его в окне базы данных и нажать кнопку Конструктор.

## **ПРИМЕР РАЗРАБОТКИ МАКРОСА**

Пусть необходимо рассчитать количество студентов в группах и внести эти данные в поле КОЛ таблицы ГРУППА. Подсчет количества студентов реализован запросом Число студентов в группе. Обновление поля КОЛ на основе полученных в этом запросе данных выполняет запрос Обновление ГРУППА\_КОЛ. Последовательное выполнение этих запросов решает поставленную задачу.

1. Сначала создадим запрос на обновление. Оформим окно конструктора запроса как показано на рисунке:



В схему данных запроса добавлены таблица Группа и таблица Число студентов, полученная после выполнения запроса на создание Число студентов в группе.

2. Преобразуем запрос на выборку в запрос на обновление. Для этого выполним команду меню Запрос/Обновление. В строке Обновление введем имя поля [CountНС] таблицы Число студентов, из которой выбираются значения для обновления.

Сохраним запрос на обновление под именем Обновление ГРУППА\_КОЛ.

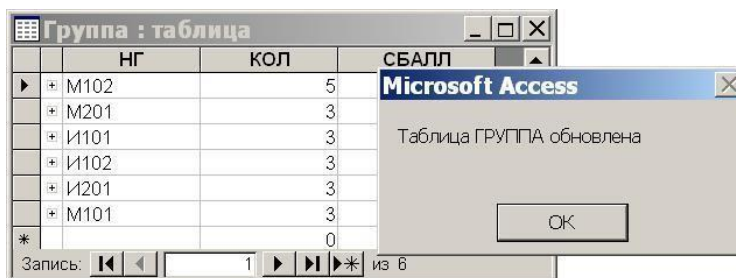
3. Создадим макрос, состоящий из макрокоманд, показанных в окне конструктора:



В столбце Примечания записаны комментарии к макрокомандам. В зависимости от выбранной макрокоманды меняется нижняя часть окна конструктора — Аргументы макрокоманды. На рисунке активна макрокоманда Сдвиг Размера, которая устанавливает положение и размеры окна таблицы Группа, демонстрируемого после ее обновления.



Сохраним макрос под именем Расчет количества студентов в группах. После нажатия кнопки Запуск получим результаты выполнения макроса, показанные на рисунке.



Поверх окна Группа выводится сообщение, в соответствии с последней макрокомандой.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Бекаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2009. – СПб.: БХВ – Петербург, 2009. – 720 с.
2. Дейт К. Дж. Введение в системы баз данных, 6-е издание. – К.; М.; СПб.: Издательский дом "Вильямс", 2008. – 848 с.
3. Калянов Г.Н. CASE - технологии: Консалтинг в автоматизации бизнес-процессов. - 3-е издание. - М.: Горячая линия-Телеком, 2008. - 320 с
4. Маклаков С.В. BPWin, ERWin. CASE – средства разработки информационных систем. – М.: Диалог-МИФИ, 2007.
5. Мандрыкин А.В. Информационные технологии в экономике: учеб пособие / А.В. Мандрыкин, А.В. Непышневский. Воронеж: ВГТУ, 2008, 235 с.
6. Михеева В.Д., Харитоновна И.А. Microsoft Access 2002. – СПб.: БХВ – Петербург, 2007. – 1040 с.

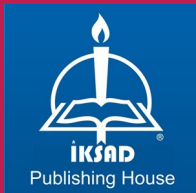
7. Литература: Информатика: Базовый курс/ С. В. Симонович и др. – СПб.: Питер, 2001. – 640 с.: ил.

8. Информатика: Практикум по технологии работы на компьютере/ Под ред. Н.В. Макаровой. – 3-е изд., перераб. – М.: Финансы и статистика, 2001. – 256 с.: ил.

9. Практикум по информатике: Учеб. Пособие для студ. высш. учеб. заведений/ А.в. Могилев, Н.И. Пак, Е.К. Хеннер; Под ред. Е.В. Хеннера. – М.: Издательский центр «Академия», 2001. – 608 с.







**ISBN: 978-625-367-304-8**