



НАРОДНАЯ УКРАИНСКАЯ АКАДЕМИЯ

В. А. Кирвас

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.
СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ
MS ACCESS**

Издательство НУА

НАРОДНАЯ УКРАИНСКАЯ АКАДЕМИЯ

В. А. Кирвас

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.
СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ
MS ACCESS**

УЧЕБНОЕ ПОСОБИЕ

для студентов II курса факультета «Референт-переводчик»,
обучающихся по специальности – Филология

Харьков
Издательство НУА
2017

УДК 004.383.1 (072+078.5)

ББК 32.973.26–018.2 р30–2

К43

*Утверждено на заседании кафедры
информационных технологий и математики
Народной украинской академии.
Протокол № 9 от 3.04.2017*

Рецензент канд. техн. наук, доцент *К. С. Барашев*,
Харьковский гуманитарный университет «Народная украинская академия»

Навчальний посібник містить теоретичні та практичні відомості що до проектування і створення таблиць реляційної бази даних. Розглядаються основи конструювання форм, запитів і звітів, а також викладено принципи форматування даних. Посібник побудований на прикладах, що демонструють основні інформаційні технології при роботі з інструментарієм системи управління базами даних MS Access 2013. Посібник містить «гарячі клавіші», словник основних термінів, список рекомендованої літератури.

Призначений для самостійної роботи студентів.

К43 Кирвас, Виктор Андреевич.

Информационные технологии. Система управления базами данных MS Access : учебное пособие для студентов фак. «Референт–переводчик», обучающихся по специальности – Филология / В. А. Кирвас ; Нар. укр. акад., [каф. информ. технологий и математики]. – Харьков : Изд-во НУА, 2017. – 196 с.

Учебное пособие содержит теоретические и практические сведения о проектировании и создании таблиц реляционной базы данных. Рассматриваются основы конструирования форм, запросов и отчётов, а также изложены принципы форматирования данных. Пособие построено на примерах, демонстрирующих основные информационные технологии при работе с инструментарием системы управления базами данных MS Access 2013. Содержит «горячие клавиши», словарь основных терминов, список рекомендуемой литературы.

Предназначено для самостоятельной работы студентов.

УДК 004.383.1 (072+078.5)

ББК 32.973.26–018.2 р30–2

© Народная украинская академия, 2017

ВВЕДЕНИЕ

Современный мир информационных технологий трудно представить себе без использования баз данных – совокупности сведений (о реальных объектах, процессах, событиях или явлениях), относящихся к определенной теме или задаче, организованные таким образом, чтобы обеспечить удобное представление этих совокупностей как в целом, так и любой их части. Практически все системы в той или иной степени связаны с функциями долговременного хранения и обработки информации. Фактически информация становится фактором, определяющим эффективность любой сферы деятельности. Увеличились информационные потоки и повысились требования к скорости обработки данных, и теперь уже большинство операций не может быть выполнено вручную, они требуют применения наиболее перспективных компьютерных технологий. Любые административные решения требуют четкой и точной оценки текущей ситуации и возможных перспектив ее изменения. Важнейшая задача компьютерных систем – хранение и обработка данных. Для ее решения были предприняты усилия, которые привели к появлению в конце 60–х – начале 70–х годов специализированного программного обеспечения – систем управления базами данных (СУБД). Сегодня если нужно упорядочить заказы от клиентов или отследить поставки товара, потребуется программа управления базами данных. Большинство деловых операций сегодня записывается, отслеживается и анализируется в виде данных, хранящихся в СУБД. Невозможно представить себе деятельность любого современного предприятия или организации без использования профессиональных СУБД. Несомненно, они составляют фундамент информационной деятельности во всех сферах – начиная с производства и заканчивая финансами и телекоммуникациями.

В этой связи, культура работы с большими объемами информации стала базовой для учебного процесса в двадцать первом веке. СУБД позволяют структурировать, систематизировать и организовать данные для их компьютерного хранения и обработки. Сегодня существуют программные продукты, с помощью которых рядовой пользователь (не программист) очень быстро решает прикладные задачи. Одной из программ, завоевавших репутацию надежного инструмента для аналитического труда и занимающих ведущее положение на мировом рынке, является, так называемая, реляционная универсальная СУБД *Microsoft Access*. Данная программа подходит для решения многих задач, имеет довольно простую структуру и привычный интерфейс (входит в знаменитый профессиональный интегрированный пакет *Microsoft Office*) и используется в нашей стране в большинстве организаций.

Улучшенный интерфейс пользователя и интерактивные средства разработки в составе MS Access делают разработку приложений в среде этого программного продукта доступной для начинающих пользователей. Любой

сотрудник, не имея опыта программирования и обладая ограниченными знаниями в области баз данных, может, используя MS Access, самостоятельно решать задачи по обработке данных. В то же время MS Access удовлетворяет требованиям профессиональных разработчиков и позволяет за незначительное время разрабатывать сложные бизнес-системы. СУБД Access обеспечивает объединение больших объемов информации в понятные отчёты, которые позволяют принимать обоснованные решения. Access обладает повышенной степенью безопасности и гибкими функциональными возможностями, позволяет легко собрать информацию и сделать ее доступной для других. Данная СУБД облегчает процесс управления информацией пользователя, обеспечивая высокую прозрачность данных в рабочей среде.

В Украине *MS Access* является, по существу, единственной, реально широко используемой программой СУБД под управлением операционной системы *Windows*. Ею пользуются деловые люди и ученые, бухгалтеры и журналисты. Референт-переводчики работают в различных отраслях в офисах, фирмах, компаниях, в государственных учреждениях и частных организациях, ориентированных в основном на развитие международных контактов, в международных компаниях и предприятиях любого профиля и форм собственности. Сегодня функциональные обязанности референт-переводчика обширны, в том числе – вести базы данных, составлять отчёты, аналитические обзоры по актуальным социальным, экономическим, финансовым вопросам, представлять ту информацию, которая интересует руководство предприятия. Поэтому сегодня профессиональная деятельность референта-переводчика просто немыслима без глубокого знания СУБД *MS Access*.

СУБД *MS Access* изучается обычно после освоения технологий работы с операционной системой *MS Windows* и другими приложениями пакета *Microsoft Office: Word, Excel, Power Point*, что позволяет уделить основное внимание главным особенностям изучаемой программы. В данном пособии рассмотрены особенности построения реляционной базы данных на примере СУБД *Microsoft Access 2013*. Важнейшей целью пособия является демонстрация функционирования рассматриваемых средств СУБД на конкретных примерах. В качестве сквозного примера рассматривается база данных, содержащая сведения гипотетического автосалона о продаже различных моделей автомобилей, а также об учёте обслуживания и ремонта их в конкретных мастерских. Выполнение на ПК в процессе изучения *Microsoft Access*, приведенных в пособии примеров, позволит студентам факультета «Референт-переводчик» успешно освоить информационные технологии работы с базой данных, приобрести навыки применения инструментария СУБД и использовать их в любой дальнейшей профессиональной предметной области.

При разработке пособия были использованы материалы литературных и интернет источников [1 – 6].

1. БАЗЫ ДАННЫХ И СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Основные понятия и термины баз данных

Фрагмент реального мира, сведения о котором необходимо хранить и использовать в решаемой задаче, в том или ином виде деятельности называют *предметной областью*. *База данных (БД)* — это организованная структура, предназначенная для хранения информации. В широком смысле «база данных» — это совокупность описаний объекта предметной области и связей между ними, правил хранения данных и способов их обработки и представления.

Базами данных являются библиотечные каталоги, папки отдела кадров учреждения со сведениями о сотрудниках, личные записные книжки, телефонные справочники, сборники кулинарных рецептов, разные справочники, словари, каталоги товаров и и т. п.

В современных БД информация накапливается не на бумажном носителе, а в устройствах компьютеров — жестких и оптических дисках и т. д. Использование электронных БД позволяет существенно сократить время поиска информации.

Отличительные особенности электронных БД:

- создаются обычно многоцелевого использования, а не для решения какой-либо одной задачи для одного пользователя;
- отражают определённую часть реального мира. Вся информация, описывающая предметную область, должна быть фиксирована в БД однократно, накапливалась и поддерживалась в актуальном состоянии централизованно;
- специальным образом организованные данные (взаимосвязанные данные, единство и целостность которых поддерживается специальными программными средствами);
- для функционирования БД необходимо наличие специальных языковых и программных средств (систем управления базами данных — СУБД), облегчающих для пользователей выполнение всех операций по организации хранения данных, их корректировки и доступа к ним.

Пользователи БД

В процессе создания и эксплуатации БД с ними взаимодействуют пользователи разных категорий.

БД создаются для удовлетворения *потребностей конечных пользователей*. Чаще всего — это специалисты конкретных предметных областей, использующие БД для выполнения своих профессиональных обязанностей. Категория «Конечные пользователи» неоднородна: они различаются широтой

информационных потребностей, квалификацией, режимами взаимодействия с БД и др. Это могут быть *случайные пользователи*, обращающиеся к БД время от времени, а могут быть и *регулярные пользователи*. Конечные пользователи отличаются друг от друга и степенью владения ЭВМ. От конечных пользователей не должно требоваться каких-то специальных знаний в области вычислительной техники и языковых средств.

Особую роль играют *руководители организации*. Они должны обеспечить проведение единой информационной политики и организацию взаимодействия различных подразделений через общую БД. Они должны создать подразделения, отвечающие за создание и функционирование БД, определять функциональные обязанности сотрудников. Руководители выступают как конечные пользователи с наивысшим приоритетом.

Пользователи-параметристы могут менять содержание БД.

Пользователи, *работающие в справочном режиме*, могут только использовать хранящуюся в БД информацию.

Терминальные пользователи взаимодействуют с БД непосредственно, а *нетерминальные* – через посредников.

Администраторы БД – лица, ответственные за создание БД и его надежное функционирование, за соблюдение регламента доступа к данным, за развитие БД.

Классификация БД

По форме представления информации различают *визуальные, аудио и мультимедиа системы*. Эта классификация показывает, в каком виде информация хранится в БД и выдается пользователям: в виде изображения (символьный текст, рисунки, чертежи, фотографии и т.д.), звука или дается возможность использования разных форм отображения информации.

По характеру организации данных БД могут быть разделены на *неструктурированные* (в виде семантических сетей), *частично структурированные* (виде обычного текста или гипертекстовые системы) и *структурированные* (требуют предварительного проектирования и описания структуры БД).

По типу используемой модели данных структурированные БД делятся на *иерархические, сетевые, реляционные, смешанные и мультимодельные*.

По типу хранимой информации БД делятся на *документальные* и *лексикографические*. Среди документальных баз различают *библиографические, реферативные и полнотекстовые*.

К *лексикографическим* БД относятся различные словари, классификаторы, рубрикаторы и т. д. Они обычно используются в качестве справочных совместно с документальными или фактографическими БД.

В *документальных* БД единицей хранения является документ (например, текст закона или статьи). Поиск и выдача документов происходит по их содержанию. В ответ на запрос пользователя выдаётся либо ссылка на доку-

мент, либо сам документ, в котором он может найти интересующую его информацию.

В системах *фактографического* типа в БД хранится информация об интересующих пользователя объектах предметной области в виде «фактов» (например, биографические данные о сотрудниках, данные о выпуске продукции производителями и т. п.).

По характеру организации хранения данных и обращения к ним различают *локальные* и *распределенные* БД.

Локальная (персональная) БД – это БД, предназначенная для использования одним пользователем. Локальные БД могут создаваться каждым пользователем самостоятельно, а могут извлекаться из общей БД.

Распределенные (многопользовательские) БД предполагают возможность одновременного обращения нескольких пользователей к одной и той же информации (многопользовательский, параллельный режим доступа). Физически разные части БД могут быть расположены на разных ЭВМ, а логически, с точки зрения пользователя, они должны представлять собой единое целое.

Следует отличать базу данных, представляющую собой хранимую информацию и систему управления базой данных (СУБД), которая представляет собой программную систему, обеспечивающую все необходимые для хранения и поиска действия.

Классификация СУБД

По языкам общения СУБД делятся на *открытые* (используют универсальные языки программирования), *замкнутые* (собственные языки общения с пользователями) и *смешанные*.

По выполняемым функциям СУБД делятся на *информационные* и *операционные*. *Информационные СУБД* позволяют организовать хранение информации и доступ к ней. *Операционные СУБД* выполняют достаточно сложную обработку, например, автоматически позволяют получать агрегированные показатели, не хранящиеся непосредственно в БД, и т.д.

По сфере возможного применения различают *универсальные* и *специализированные, проблемно-ориентированные СУБД*.

По «мощности» СУБД делятся на *настольные* и *корпоративные*. Характерными чертами *настольных СУБД* являются сравнительно невысокие требования к техническим средствам, ориентация на конечного пользователя, низкая стоимость.

Корпоративные СУБД обеспечивают работу в распределенной среде, высокую производительность, поддержку коллективной работы при проектировании систем, имеют развитые средства администрирования и более широкие возможности поддержания целостности. Эти системы сложны, дороги, требуют значительных вычислительных ресурсов.

Наиболее популярными *настольными* СУБД являются: Visual dBase, Paradox, **Microsoft Access**, Microsoft FoxPro, Microsoft Data Engine.

Корпоративными (серверными) СУБД являются: Oracle, Microsoft SQL Server, Informix, Sybase, DB2.

По ориентации на преобладающую категорию пользователей выделяют СУБД для разработчиков и для конечных пользователей.

Системы, ориентированные на разработчиков, должны: иметь качественные компиляторы; позволять создавать «отчуждаемые» программные продукты; обладать развитыми средствами отладки; включать средства документирования проекта; обладать возможностями, позволяющими создавать эффективные сложные системы.

Основными требованиями, предъявляемыми к системам, *ориентированным на конечного пользователя*, являются: удобство интерфейса; высокий уровень языковых средств; наличие интеллектуальных модулей подсказок; повышенная защита от непреднамеренных ошибок и т. д.

Типы моделей данных

Ядром любой БД является модель данных. *Модель данных* – интегрированный набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные.

Классические модели: иерархическая, сетевая, реляционная. *Современные* модели: постреляционная, многомерная, объектно-ориентированная.

1. Иерархическая модель данных

Дерева. Дерево представляет собой иерархию элементов, называемых *узлами*. На самом верхнем уровне иерархии имеется только один узел – *корень*, являющийся входом в структуру. Каждый узел, кроме корня, связан с одним узлом на более высоком уровне, называемым *исходным узлом* для данного узла. Каждый элемент имеет только один исходный. Каждый элемент может быть связан с одним или несколькими элементами на более низком уровне, которые называются *подчинёнными*. Между исходным узлом и подчинёнными узлами имеется отношение «один – ко – многим». Между двумя узлами может быть только одна связь.

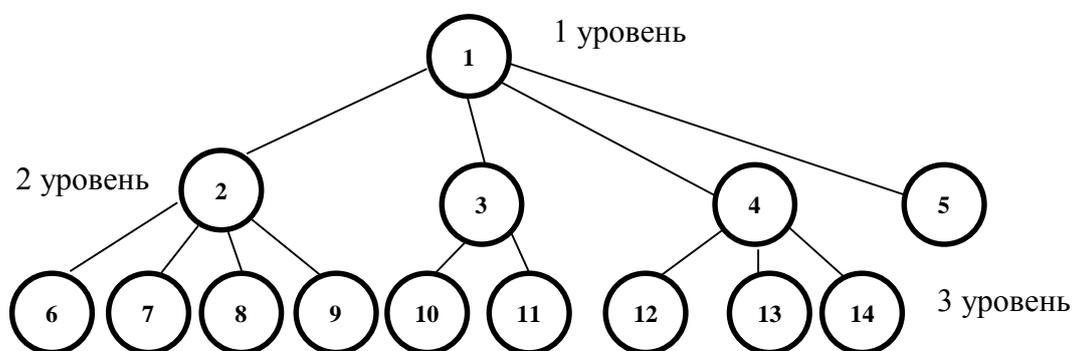


Рис. 1.1. Схема иерархической модели данных.

Иерархические БД (рис. 1.1) используют древовидную структуру для работы с данными. Важной особенностью иерархической структуры является то, что в ней подчинённые экземпляры не могут существовать без наличия исходных экземпляров. Для доступа к отдельным данным в иерархической системе необходимо пройти последовательный перебор элементов, начиная с корневого узла.

Достоинством иерархической модели данных является то, что целостность на уровне ссылок обеспечивается автоматически, потому что узлы полностью зависят от других узлов.

Недостатком данной модели данных является, то что из нижних уровней иерархии нельзя направить информационный поиск по вышележащим узлам.

На иерархической модели данных основано сравнительно ограниченное количество СУБД.

2. Сетевая модель данных

В сетевой структуре (рис. 1.2) любой элемент может быть связан с любым другим элементом, и каждый из элементов может являться входом в структуру. Данные в сетевой модели представлены в виде совокупностей записей, а связи – в виде наборов. Сетевая модель является обобщением иерархической модели.

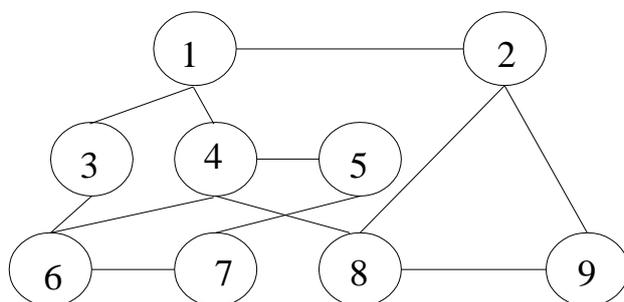


Рис. 1.2. Сетевая модель представления данных

В сетевой модели данных понятия главного и подчинённых объектов несколько расширены. Любой объект может быть и главным и подчинённым (в сетевой модели главный объект обозначается термином «владелец набора», а подчинённый — термином «член набора»). Один и тот же объект может одновременно выступать и в роли владельца, и в роли члена набора. Это означает, что каждый объект может участвовать в любом числе взаимосвязей.

В простых сетевых структурах между парой элементов поддерживается отношение «один – ко – многим». В сетевых БД все данные считаются потенциально взаимосвязанными.

Недостатки сетевой модели представления данных: обладает ограниченной гибкостью по отношению к изменению требований к данным и методам доступа; доступ к данным осуществляется путём перемещения (навигации) по структуре; «сетевая БД – это самый верный способ потерять данные».

Системы на основе сетевой модели не получили широкого распространения на практике.

Иерархическая и сетевая модели считаются моделями БД *первого поколения*. Этим двум моделям присущи ещё и общие недостатки: даже для выполнения простых запросов с использованием переходов и доступом к определённым записям необходимо создавать достаточно сложные программы; независимость от данных существует лишь в минимальной степени, и отсутствуют общепризнанные теоретические основы. Современные СУБД, ориентированные на иерархические и сетевые модели, имеют ограниченные средства для поддержания целостности и защиты данных.

Иерархическая и сетевая модели данных стали применяться в системах управления базами данных в начале 60-х годов. В начале 70-х годов прошлого века сотрудник фирмы IBM Э. Ф. Кодд ввёл реляционную модель данных.

3. Реляционная модель данных

В математических дисциплинах существует понятие «*отношение*» (relation), физическим представлением которого является *таблица*. Отсюда и произошло название модели – *реляционная*.

Применительно к БД понятия «реляционная БД» и «табличная БД» являются синонимами. Реляционные базы получили наибольшее распространение в мире. Почти все БД, созданные с конца 70-х годов, являются реляционными.

Реляционные СУБД относятся к СУБД *второго поколения*.

В реляционной модели объекты и взаимосвязи между ними представляются с помощью таблиц, как это показано на рис. 3. Взаимосвязи также рассматриваются в качестве объектов.

Создание реляционной модели данных преследует следующие цели: обеспечение более высокой степени независимости от данных; создание прочного фундамента для решения проблем непротиворечивости и избыточности данных; расширение языков управления данными.

Реляционная модель (рис.1.3) является удобной и наиболее привычной формой представления данных в виде *таблицы (отношения)*. Каждая таблица представляет один объект, имеет *имя* и состоит из поименованных *атрибутов* (столбцов) данных и строк (записей). Одним из основных преимуществ реляционной модели является её однородность. Все данные хранятся в таблицах, в которых каждая строка имеет один и тот же формат. Каждая строка в таблице представляет некоторый объект реального мира или соотношение между объектами.

В реляционной базе данных каждая таблица должна иметь первичный ключ (ключевой элемент) – поле или комбинацию полей, которые единственным образом идентифицируют каждую строку в таблице. Благодаря своей простоте и естественности представления реляционная модель получила наибольшее распространение в СУБД для персональных компьютеров.

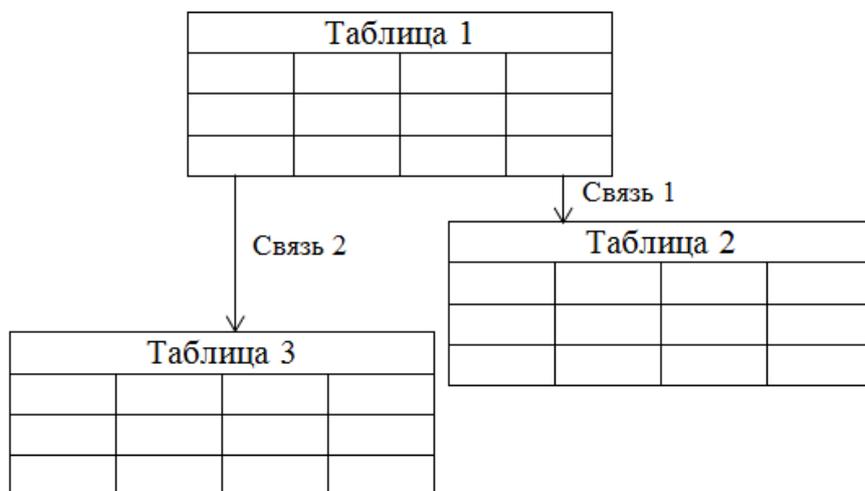


Рис. 1.3. Реляционная модель представления данных

Признаки, позволяющие считать таблицу отношением:

- в таблице нет строк с совпадающими ключами (строки уникальны);
- в каждой строке содержатся значения одного и того же набора атрибутов;
- отношения неразложимы (не могут быть элементами другого отношения).

Основные понятия реляционной БД:

- 1) *реляционная БД* – набор нормализованных отношений;
- 2) *отношение* – файл, плоская таблица, состоящая из столбцов и строк;
- 3) *домен* – совокупность допустимых значений, из которой берутся значения соответствующих атрибутов определённого отношения. С точки зрения программирования, *домен* – это тип данных;

4) *кортеж* – запись, строка таблицы;

5) *кардинальность* – количество строк в таблице;

6) *атрибуты* – поименованные поля, столбцы таблицы;

7) *степень отношения* – количество полей (столбцов);

8) *первичный ключ* – уникальный идентификатор с неповторяющимися записями – столбец или некоторое подмножество столбцов, которые единственным образом определяют строки.

Первичный ключ, который включает более одного столбца, называется *множественным*, или *комбинированным*, или *составным*, или *суперключом*.

Правило целостности объектов утверждает, что первичный ключ не может быть полностью или частично пустым.

9) *внешний ключ* – это столбец или подмножество столбцов одной таблицы, которые могут служить в качестве первичного ключа для другой таблицы. Внешний ключ таблицы является ссылкой на первичный ключ другой таблицы. Поскольку целью построения БД является хранение всех данных, по возможности, в одном экземпляре, то если некий атрибут (столбец) присутствует в нескольких отношениях (таблицах), то его наличие обычно отражает определённую связь между строками этих таблиц.

Правило ссылочной целостности гласит: внешний ключ может быть либо пустым, либо соответствовать значению первичного ключа, на который он ссылается. Внешние ключи реализуют связи между таблицами БД.

Внешний ключ, как и первичный ключ, может представлять собой комбинацию столбцов. На практике внешний ключ всегда будет составным, если он ссылается на составной первичный ключ другой таблицы. Количество столбцов и их типы данных в первичном и внешнем ключах должны совпадать.

Достоинства реляционных моделей данных:

- простая и удобная для пользователя схема данных, представляемая в виде таблиц;
- улучшенная логическая и физическая независимость;
- обеспечение пользователя языками высокого уровня;
- оптимизация доступа к БД;
- улучшение целостности и защиты данных;
- возможности различных применений;
- обеспечение методологического подхода.

Таким образом, достоинство реляционной модели данных заключается в простоте, понятности и удобстве физической реализации на ЭВМ. Именно простота и понятность для пользователя явились основной причиной их широкого использования.

Основными недостатками реляционной модели являются следующие: отсутствие стандартных средств идентификации отдельных записей и сложность описания иерархических и сетевых связей.

При использовании иерархической и сетевой моделей от пользователя требуется знание физической организации БД, к которой он должен осуществлять доступ, в то время как при работе с реляционной моделью независимость от данных обеспечивается в значительно большей степени. Следовательно, если в реляционных системах для обработки информации в БД принят декларативный подход (т. е. они указывают, какие данные следует извлечь), то в сетевых и иерархических системах – навигационный подход (т. е. они указывают, как их следует извлечь).

4. Постреляционная модель данных

Классическая реляционная модель предполагает неделимость данных, хранящихся в полях таблиц. Постреляционная модель – это расширенная реляционная модель, снимающая ограничения неделимости данных. Эта модель допускает многозначные поля – поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную таблицу. В такой модели данные хранятся более эффективно, а при обработке не требуется выполнять операцию соединения данных из двух таблиц.

Достоинства: возможность представления совокупности связанных реляционных таблиц одной постреляционной таблицей, высокая наглядность представления информации и повышение эффективности её обработки.

Недостатки: проблема обеспечения целостности и непротиворечивости хранимых данных.

5. Многомерные СУБД являются узкоспециализированными СУБД, предназначенными для интерактивной аналитической обработки информации. По сравнению с реляционной моделью многомерная модель обладает более высокой наглядностью и информативностью. *Достоинства:* удобство и эффективность аналитической обработки больших объёмов данных, связанных со временем. *Недостатки:* громоздкость для решения простейших задач обычной оперативной обработки данных.

Жизненный цикл БД

БД является фундаментальным компонентом более широкого понятия – информационной системы (ИС). Этапы жизненного цикла БД показаны на рис. 1.4.

Планирование разработки БД. Разработка начинается с маркетинга, т. е. исследования рынка программного обеспечения, поиска аналогов, выяснения потребности в БД. Планирование разработки БД состоит в определении трёх основных компонентов: требуемого объёма работы, необходимых ресурсов и общей стоимости проекта.

Определение требований к системе. Определение диапазона действия и границ приложения БД, состав его пользователей и областей применения.

Сбор и анализ требований пользователей. Необходимая для проектирования БД информация может быть собрана следующими способами: посредством опроса отдельных пользователей предметной области; с помощью наблюдений за деятельностью предметной области; посредством изучения документов, которые используются для сбора или представления информации; с помощью анкет, предназначенных для сбора информации у широкого круга пользователей; за счёт использования опыта проектирования других подобных систем.

Проектирование БД. Полный цикл разработки включает концептуальное, логическое и физическое проектирование БД.

Выбор целевой СУБД. На этом этапе выполняется выбор наиболее подходящей СУБД для приложения БД.

Разработка приложений. Определение пользовательского интерфейса и прикладных программ, которые используют и обрабатывают БД.

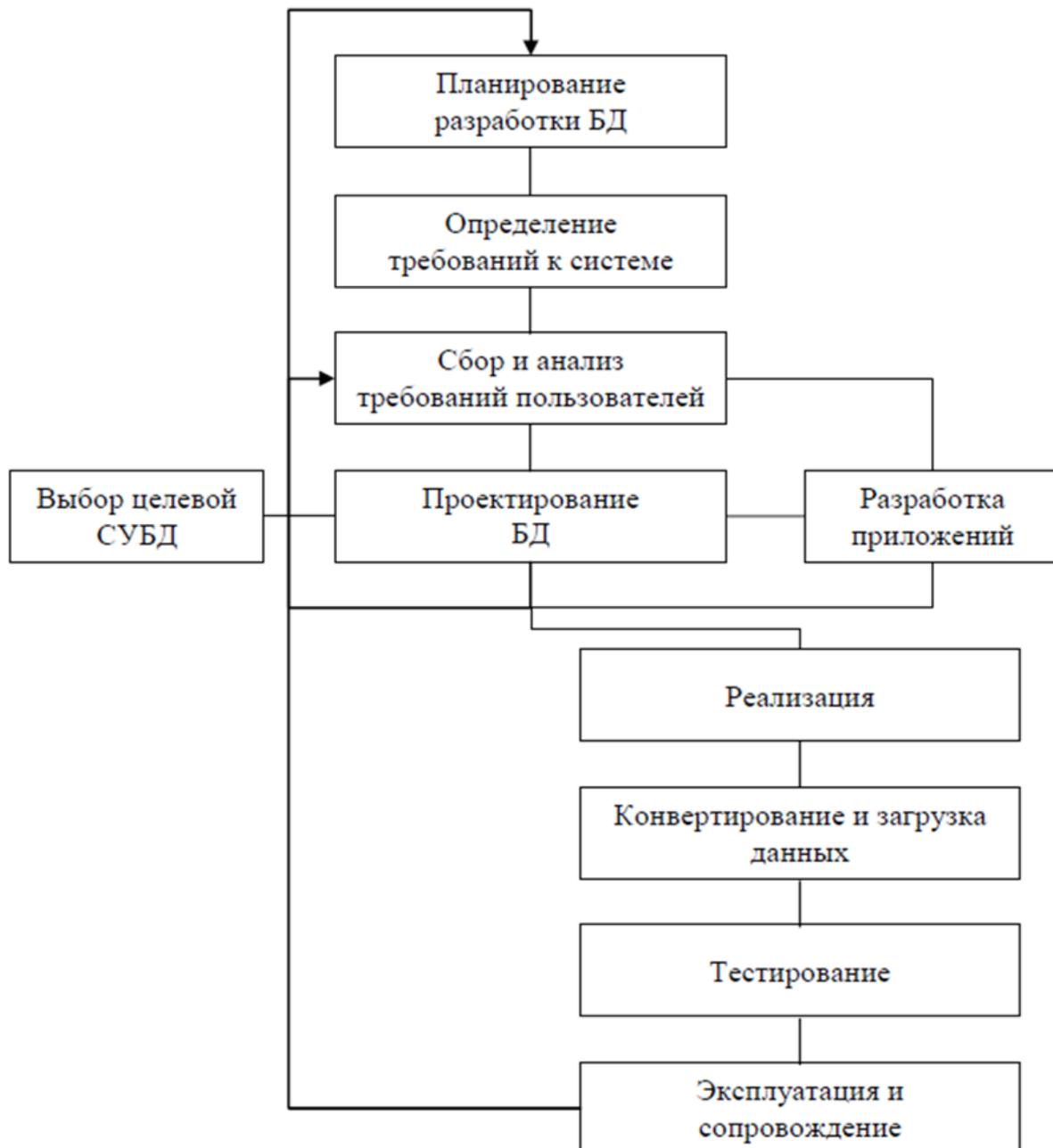


Рис. 1.4. Жизненный цикл БД

Реализация. Реализация БД начинается с её распространения для коммерческих и свободных продуктов или внедрения для собственных разработок и заказных программ. В обоих случаях различают БД, которые не требуют вмешательства специалиста для начала эксплуатации, и БД, нуждающиеся в начальной настройке. Внедрение БД обычно производится с участием разработчиков.

Конвертирование и загрузка данных. Перенос любых существующих данных в новую БД и модификация любых существующих приложений с целью организации совместной работы с новой БД.

Тестирование. Приложение БД тестируется с целью обнаружения ошибок, а также его проверки на соответствие всем требованиям, выдвинутым пользователями. Пользователи должны быть вовлечены в процесс тестирования. По завершении тестирования процесс создания прикладной системы считается законченным, и она может быть передана пользователям в эксплуатацию.

Эксплуатация и сопровождение. На этом этапе приложение БД считается полностью разработанным и реализованным. Впредь вся система будет находиться под постоянным наблюдением и соответствующим образом поддерживаться. В случае необходимости в функционирующее приложение могут вноситься изменения, отвечающие новым требованиям. Реализация этих изменений проводится посредством повторного выполнения некоторых из перечисленных выше этапов жизненного цикла.

Жизненный цикл БД заканчивается вместе с прекращением её распространения и сопровождения. Обычно это происходит при моральном устаревании БД.

1. ОСОБЕННОСТИ ПОСТРОЕНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ В MICROSOFT ACCESS

Общие понятия и требования СУБД MS Access

Реляционные базы данных фактически стали промышленным стандартом. Подавляющее большинство деловых операций сегодня записывается, отслеживается и анализируется в виде данных, хранящихся в реляционных системах управления базами данных. К ним относится СУБД MS Access, которая удовлетворяет потребности самых разных групп пользователей. С помощью мастеров и графических инструментов Access даже пользователи, не владеющие специальными навыками, могут весьма успешно разрабатывать полезные приложения баз данных.

Функциональное назначение реляционной СУБД MS Access:

- хранить и систематизировать данные, что позволяет легко ориентироваться и быстро находить нужные сведения;
- обновлять данные, автоматически или с использованием удобных форм данных;
- провести количественный и качественный анализ хода какого-то процесса в течение определённого отрезка времени или пространства;
- формировать отчёты, собирающие данные из различных таблиц в удобной форме и по заданным критериям.

СУБД MS Access предназначена, в основном, для создания *настольных* и *офисных* БД, для которых характерны: относительно несложная структура данных (десятки таблиц); ограниченное число пользователей в локальной сети (единицы, десятки рабочих мест) или на отдельном компьютере.

Требования к операционной системе для Access 2013

Access 2013 входит в состав MS Office 2013 (в наборы приложений выпусков Профессиональный, Профессиональный Плюс). Кроме того, можно оформить подписку на «облачные» версии Office 365 – Access входит в состав следующих выпусков Office 365: для дома расширенный, для малого бизнеса расширенный, для среднего бизнеса расширенный, профессиональный плюс, корпоративный. Прежде чем устанавливать на ПК программу MS Access 2013, рекомендуется убедиться в том, что компьютер удовлетворяет минимальным требованиям к системе.

Для работы Access 2013 рекомендуется использовать ПК и процессор 86 или 64-разрядный процессор с тактовой частотой от 1 ГГц и поддержкой набора инструкций SSE2. Должно быть не менее 1 ГБ оперативной памяти (для 32-разрядной версии) и 2 ГБ оперативной памяти (для 64-разрядной версии). На жёстком диске должно быть не менее 3,0 ГБ свободного места. В процессе установки на диске необходимо иметь дополнительное простран-

ство примерно такого же объема. Рекомендуется монитор с разрешением экрана не менее 1024 x 768. Для аппаратного ускорения обработки изображения необходима видеокарта, которая поддерживает технологию DirectX 10. Компьютер должен работать под управлением одной из операционных систем: Windows 7, Windows 8, Windows 10, Windows Server 2008 R2 или Windows Server 2012. Для функций Access 2013, работающих через Интернет, необходимо использовать браузеры, поддерживаемые Office 2013 и Office 365 (Internet Explorer 8, 9, 10 или 11; Mozilla Firefox 10.x или более поздней версии; Apple Safari 5 или Google Chrome 17.x.). Версия .Net: 3.5, 4.0 или 4.5. Для использования функции быстрого поиска требуется Windows Search 4.0.

Объекты СУБД MS Access

СУБД MS Access включает следующие сохраняемые в одном файле **объекты** для ввода, хранения и управления информацией:

- *таблицы, запросы, схемы данных*, непосредственно имеющие отношение к БД;
- *формы, отчёты, макросы и модули*, называемые объектами приложения.

Ниже приведена характеристика этих основных объектов базы данных.

Таблица является основой базы данных. Вся информация хранится именно в таблицах. Таблица содержит данные об одной сущности — одном информационном объекте модели данных предметной области, т. е. по определённой теме, например, сведения о сотрудниках или товарах. В реляционной СУБД MS Access все хранимые данные сгруппированы в виде плоских двумерных взаимосвязанных исходных таблиц, иногда называемыми первичными. Каждый столбец в таблице называется **полем**. Он имеет **имя** и содержит один определённый тип информации (одну характеристику информационного объекта предметной области). Различные таблицы могут иметь столбцы с одинаковыми именами. Каждая строка содержит информацию об определённом экземпляре объекта, описываемого в данной таблице, и называется **записью**. Ячейки таблицы, составляющие запись, по терминологии баз данных также называются **полями**. Когда говорят о поле записи, речь идёт о ячейке таблицы. Набор данных состоит из записей, а каждая запись данных — из отдельных полей. Структура всех записей базы одинакова: все они имеют одну и ту же последовательность полей, но содержимое полей у каждой записи своё. База данных может включать множество таблиц, в которых хранятся данные по различным темам. База данных Access может включать до 32 768 объектов (в том числе формы, отчёты и т. д.). Одновременно может открываться до 2048 таблиц.

Схема данных определяет, с помощью каких полей таблицы связываются между собой, как будет выполняться объединение данных этих таблиц, нужно ли проверять связную целостность при добавлении и удалении записей, изменении ключей таблиц.

Форма позволяет более наглядно отобразить информацию, содержащуюся в таблицах базы данных. Кроме того, форма позволяет упростить процесс заполнения базы данных, благодаря чему появляется возможность поручить ввод информации персоналу невысокой квалификации. Формы иногда называются окнами ввода данных. Также форма позволяет ограничить объем информации, доступной пользователю, обращающемуся к базе. Благодаря этому существует возможность блокирования доступа к конфиденциальной информации в базе данных.

Запросы являются основным рабочим инструментом базы данных и могут выполнять множество различных функций по отбору и обработке необходимых данных. Данные, которые необходимо просмотреть, как правило, находятся в нескольких таблицах. Запросы позволяют представить их в одной таблице, выбрать из базы только необходимую информацию, т.е. ту, которая соответствует определённому критерию (условию) и нужна для решения конкретной задачи.

Отчёты используются, как правило, для печати информации. Отчёты служат для сбора и представления данных, содержащихся в таблицах. Можно придать отчёту более привлекательный вид (по сравнению с таблицами и формами). Доступные способы оформления информации в отчётах MS Access достаточно разнообразны. Обычно отчёты форматируют для печати, но их можно также просматривать на экране, экспортировать в другую программу или отправлять в виде сообщений электронной почты.

Макросы используются в тех случаях, когда при обработке больших объёмов данных часто приходится выполнять длинные последовательности действий. Любые действия могут быть оформлены как макросы. Вызов макроса приводит к выполнению соответствующей данному макросу последовательности действий. Макросы в приложении Access можно рассматривать как упрощённый язык программирования, который позволяет добавлять функциональные возможности в базу данных. Большинство операций с базой данных, выполняемых вручную, можно автоматизировать с помощью макросов, которые позволяют существенно экономить время. Макросы позволяют обеспечить создание удобного интерфейса приложения: чтобы формы и отчёты открывались при нажатии кнопок; чтобы при открытии приложения пользователь видел на экране не окно БД с множеством таблиц, запросов, форм и отчётов, а удобную и понятную форму в виде команд меню и т. д.

Модули используются при решении достаточно сложных задач, когда недостаточны возможности построения командных макросов. В то время как макросы создаются в приложении Access путём выбора макрокоманд из списка, для преодоления возникших трудностей можно попробовать написать собственную процедуру обработки информации на языке программирования Visual Basic для приложений (VBA). Такая процедура оформляется как модуль.

Все данные одной БД Microsoft Access хранятся в одном файле форма-

та (расширения) «.mdb» – в Access 2003 или с расширением «.accdb» – в Access 2007, 2010 и 2013.

Каждая последующая версия программы поддерживает ряд усовершенствований. В частности, Access 2007 поддерживает новые свойства, такие как многозначные поля и вложения, Access 2010 – позволяет преобразовать БД на компьютере в Web-базу данных с использованием служб Access. С помощью Access 2013 можно создавать не только классические БД, но и удобные веб-приложения для работы с БД, которые значительно облегчают ведение бизнеса. Данные автоматически сохраняются в БД SQL, поэтому они надёжно защищены, а приложения можно с лёгкостью использовать совместно с коллегами. Веб-приложение Access представляет собой БД нового типа, которая создаётся в Access, а затем используется и публикуется как приложение SharePoint для общего доступа в веб-браузере.

Файлы нового формата нельзя открыть или преобразовать в более ранних версиях Access или связать с ними. Базу данных, созданную в Access 2003 можно открыть в Access 2007, 2010 и 2013, но, чтобы применить новые возможности Access старую БД необходимо преобразовать в формат новой версии Access. И наоборот, если необходимо работать с базой данных в более ранних версиях Access, необходимо применять прежний формат файла, сохраняя новый формат данных в формате прежней версии. Некоторые функции, естественно, при этом теряются.

Разработка структуры базы данных

Если для начала работы с таблицами Excel достаточно ввести данные, то для работы с Access необходимо предварительно спроектировать базу данных. БД с правильной структурой обеспечивает доступ к обновлённым и точным сведениям. Поскольку правильная структура важна для выполнения поставленных задач при работе с базой данных, целесообразно изучить принципы создания баз данных. Это позволит создать БД, отвечающую потребностям пользователя и позволяющую быстро вносить в неё изменения.

Проектирование базы данных состоит из двух основных фаз: *логического* и *физического* проектирования.

Во время фазы *логического* проектирования собираются требования и разрабатывается модель, не зависящая от конкретной СУБД.

Во время фазы *физического* проектирования создаётся модель, оптимизированная для конкретного приложения СУБД. Данная модель реализуется на практике.

Процесс проектирования БД состоит из следующих этапов:

- 1 - сбор информации;
- 2 - идентификация объектов;
- 3 - моделирование объектов;
- 4 - идентификация типов информации для каждого объекта;

- 5 - идентификация отношений (таблиц);
- 6 - нормализация;
- 7 - преобразование к физической модели;
- 8 - создание базы данных.

Этапы 1-6 образуют фазу логического проектирования. Этапы 7 и 8 представляют собой фазу физического проектирования.

На первом этапе проектирования, до начала создания структуры БД, необходимо определить цели создания, назначение базы данных, задачи, режимы её использования и список пользователей, основные алгоритмы, реализующие реальные процессы – т.е. изучить предметную область её использования. При этом анализ поставленной задачи должен учитывать требования заказчика к разрабатываемой системе и опыт разработчика. Основная задача – составить подробное описание цели создания базы данных, чтобы иметь возможность обращаться к нему в процессе проектирования. Наличие описания позволит следовать поставленным целям в процессе принятия решений. Целесообразно составить список вопросов, ответы на которые требуется получать с помощью базы данных. Определение структуры отчётов и рассылок позволит выявить элементы, которые требуется включить в базу данных. Удобно создать прототип каждого отчёта для печати и определить, какие элементы данных потребуются для их создания. Данные необходимо разбивать на минимальные элементы. Например, имя следует разделять на две части: имя и фамилию, чтобы фамилию было легче использовать.

Идентификации подлежат все сущности, относящиеся к поставленной задаче (в данном случае – данные и объекты базы), а также связи между данными. В ходе идентификации определяются атрибуты (свойства) сущностей, и для этого необходимо принять решения по следующим вопросам: какие значения должны содержаться в поле; сколько места необходимо для хранения значений в поле; какие операции должны производиться со значениями в поле; нужна ли сортировка данных поля; необходимо ли группировать данные.

Распределение данных по таблицам. Чтобы распределить данные по таблицам, надо выделить основные группы или темы. Следует разбить информацию на минимальные логические компоненты. При разработке БД надо стремиться к однократному сохранению каждого элемента данных. Если данные повторяются, следует поместить их в отдельную таблицу. После выбора темы для таблицы необходимо отслеживать, чтобы столбцы этой таблицы содержали данные только по этой теме.

Преобразование элементов данных в столбцы. Чтобы определить столбцы таблицы, следует решить, какие сведения требуется отслеживать о предмете, записи о котором хранятся в таблице. Каждый элемент данных должен занимать отдельное поле и отображаться в виде столбца таблицы. Первоначальный набор столбцов для каждой таблицы можно дополнять новыми столбцами. Не следует включать в таблицу вычисляемые данные.

Правильная структура базы данных подразумевает:

- распределение данных по тематическим таблицам в целях сокращения объёма повторяющихся данных;
- добавление в Access данных, необходимых для объединения сведений, которые содержатся в таблицах;
- возможность поддержания и отслеживания точности и целостности данных;
- соответствие требованиям к обработке данных и созданию отчётов.

При создании таблиц в Access необходимо выбирать **тип данных** для каждого поля (столбца) данных. В Таблице 1 приведён список, предусмотренных в приложении MS Access 2013 для настольных ПК, типов данных, инструкции по их применению и сведения о размере места, необходимого для хранения данных каждого типа. Тип данных «Короткий текст» является распространённым выбором, поскольку позволяет вводить практически любой знак (букву, символ или цифру). Тем не менее, внимательный выбор типов данных поможет воспользоваться дополнительными возможностями Access (такими как проверка вводимых значений и функции), а также повысить точность хранимой информации.

Таблица 1.

Типы данных MS Access 2013

Тип данных	Применение	Размер
<u>Короткий текст</u> (ранее известный как «Текстовый»)	Алфавитно-цифровые данные (имена, названия и т.д.), включая текст, а также числа, не применяемые в вычислениях, таких как номера телефонов	До 255 символов
<u>Длинный текст</u> (ранее известный как Мемо)	Большие объёмы алфавитно-цифровых данных: предложения и абзацы, размер которых превышает 255 символов	До 1 гигабайта (ГБ), но контроль отображения полного текста ограничен первыми 64 000 знаков
Числовой	Числовые данные (целые или дробные), которые используются в вычислениях (за исключением денежных сумм)	1, 2, 4, 8 или 16 байтов.
Дата и время	Значения даты и времени за годы с 100 до 9999	8 байтов
Денежный	Денежные данные, хранящиеся с точностью до 4 десятичных знаков после запятой. Ис-	8 байтов

Тип данных	Применение	Размер
	пользуется для предотвращения округлений во время вычислений	
Счётчик	Уникальные значения, сгенерированные Access для каждой новой записи, которые могут применяться в качестве первичного ключа. Эти значения автоматически вставляются в поле при добавлении записи. Поля с данным типом данных могут формироваться добавлением единицы, добавлением заданного значения или с помощью случайных чисел	4 байта (16 байтов для кода репликации).
Логический	Логические данные (Да/Нет, Истина/Ложь или Вкл/Выкл.). Access хранит числовое значение нуль (0) для лжи и -1 для истины.	1 бит (0,125 байта)
Поле объекта OLE	Изображения, графики или другие объекты ActiveX из другого приложения MS Windows. Объект (например, таблицы MS Excel, документ MS Word, звуки или другие двоичные данные), связанный или встроенный в таблице MS Access.	До 2 ГБ.
Гиперссылка	Адрес ссылки на документ или файл в Интернете, интрасети, локальной сети или на локальном компьютере	До 8192 (каждая часть типа данных гиперссылки может содержать до 2048 знаков).
Вложение	Можно вкладывать изображения, документы, электронные таблицы или диаграммы, которые нельзя прочесть с помощью текстового редактора. Каждое поле вложений может содержать неограниченное ко-	До 2 ГБ.

Тип данных	Применение	Размер
	личество вложений на одну запись, вплоть до допустимого размера файла базы данных.	
Вычисляемый	Можно создать выражение, использующее данные из одного или более полей. Из выражения можно назначить различные типы данных результатов.	Зависит от типа данных свойства «Тип результата». Результат типа данных «Короткий текст» может содержать до 243 знаков. Данные типа «Длинный текст», «Числовой», «Да/нет» и «Дата/время» должны соответствовать своим типам данных.
Мастер подстановок	При выборе этой записи запускается мастер, чтобы помочь определить простое или сложное поле подстановки. Простое поле подстановки использует содержимое другой таблицы или списка значений, чтобы проверить правильность содержимого единственного значения в ряду. Сложное поле подстановки позволяет хранить множественные значения одного типа данных в каждом ряду. Запись «Мастер подстановок» в столбце «Тип данных» в «Конструкторе» фактически не является типом данных.	Зависит от типа данных поля подстановки.

В Таблице 2 перечислены *свойства полей* и описано влияние этих свойств на поля в зависимости от их типа данных.

Свойства полей MS Access 2013

Свойства поля	Описание
Размер поля	Определяет максимальную длину текстового или числового поля, так как если размер подобран не оптимально, расходуется, лишняя память
Формат поля	Устанавливает формат отображения данных в форме, запросе, отчёте
Число десятичных знаков	Количество знаков после запятой в десятичном числе
Маска ввода	Задаёт маску (шаблон), при вводе данных в таблицу или форму
Значение по умолчанию	Содержит значение, установленное по умолчанию, для всех новых записей таблицы
Подпись	Задаёт подпись поля, которое выводится в формах, отчётах, заголовках столбцов таблиц (не путать с именем поля). Если подпись не указана, то в качестве заголовка столбца используется свойство Имя поля
Условие на значение	Позволяет задать то условие, которое проверяется при вводе данных в поле
Сообщение об ошибке	Задаётся текст, сообщение выводится в диалоговом окне, если вводимые данные не соответствуют, заданному условию на значение
Обязательное поле	Определяет, может ли поле быть пустым или нет
Пустые строки	Определяет возможность ввода в поля пустых строк с пробелами
Индексированное поле	Задаёт индексы, для ускоренного поиска информации в

Задание первичных ключей. Таблицы баз данных представляют собой изначально неупорядоченный набор записей. Единственный способ идентифицировать определённую запись в таблице — это указать набор атрибутов, который был бы уникальным для этой записи.

Ключом называется набор атрибутов (полей), однозначно определяющий запись (строку). Существуют следующие виды ключей: *первичный ключ* и *внешний ключ*.

Первичный ключ — представляет собой одно или несколько полей (столбцов), значения которых однозначно определяют каждую запись в таблице. Первичный ключ не допускает значений Null, т. е. должен всегда иметь значение. Первичный ключ всегда должен иметь уникальный индекс, т. е. не

должен содержать повторяющихся значений. Например, не следует использовать в качестве первичного ключа имена людей, т. к. они не являются уникальными. В одной таблице могут существовать две записи с одинаковыми именами.

Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах. Первичный ключ может быть *естественным* или *искусственным*. Ключ, состоящий из информационных полей таблицы (т. е. полей, содержащих полезную информацию об описываемых объектах) называется *естественным ключом*. Теоретически, естественный ключ всегда можно сформировать, в этом случае он носит название «*интеллектуальный ключ*». *Искусственный ключ* – это дополнительное служебное поле, единственное предназначение которого – служить первичным ключом. Значения этого поля не образуются на основе каких-либо других данных из БД, а генерируются искусственно. Как правило, это просто числовое поле, в которое заносятся значения из возрастающей числовой последовательности (счётчик).

В некоторых случаях в качестве первичного ключа таблицы требуется использовать два и более поля. Первичный ключ из нескольких столбцов называется *составным*.

Внешний ключ - представляет собой одно или несколько полей (столбцов), содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ связи таблиц.

Индекс – это средство, ускоряющее поиск и сортировку данных в таблице. Существенное повышение скорости выполнения запросов приносит индексирование полей, расположенных в разных таблицах, или создание связи между этими полями, а также индексирование всех полей, используемых для задания условий отбора в запросе.

БД представляет собой множество взаимосвязанных двумерных таблиц, в каждой из которых содержатся сведения об одной сущности автоматизируемой предметной области.

В таблицах базы должны сохраняться все данные, необходимые для решения задач предметной области. Причём каждый элемент данных должен храниться в базе только в одном экземпляре. Для создания таблиц используется процесс, называемый нормализацией данных. *Нормализация* – это удаление из таблиц повторяющихся данных путём их переноса в новые таблицы, записи которых не содержат повторяющихся значений.

Исключение дублирования данных в реляционной базе обеспечивает высокую эффективность поддержания БД в актуальном и непротиворечивом состоянии, однократный ввод и корректировку данных.

Создание связей между таблицами

СУБД Access — это система управления реляционными базами данных. В реляционной базе данных сведения распределяются по отдельным тематически организованным таблицам. Для объединения данных используются связи между таблицами.

После распределения данных по таблицам необходимо получить возможность объединять их. Логические связи между таблицами дают возможность объединять данные из разных таблиц. Связь каждой пары таблиц задается одинаковыми полями в них – ключом связи. Так обеспечивается рациональное хранение недублированных данных и их объединение в соответствии с требованиями решаемых задач.

В СУБД Access процесс создания реляционной базы данных включает создание *схемы данных*. Схема данных наглядно отображает логическую структуру базы данных: таблицы и связи между ними.

Связи, определённые в схеме данных, автоматически используются для объединения таблиц при разработке многотабличных форм, запросов, отчётов, существенно упрощая процесс их конструирования.

В схеме данных связи могут устанавливаться для любой пары таблиц, имеющих одинаковое поле, позволяющее объединять эти таблицы.

В нормализованной реляционной базе данных связь двух таблиц характеризуется отношениями записей типа «*один-к-одному*», «*один-ко-многим*» или «*многие-ко-многим*».

Отношение «*один-к-одному*» предполагает, что каждой записи одной таблицы соответствует одна запись в другой, а каждой записи второй таблицы – только одна запись первой таблицы. Отношение «*один-к-одному*» обычно используется при разбиении таблицы с большим числом полей на несколько таблиц.

Отношение «*один-ко-многим*» предполагает, что каждой записи первой таблицы соответствует много записей во второй, но каждой записи второй таблицы соответствует только одна запись в первой.

Для двух таблиц, находящихся в отношении типа «*один-ко-многим*», связь устанавливается по уникальному первичному ключу таблицы, представляющей в отношении сторону «*один*», – *главной* таблицы в связи. Во второй таблице, представляющей в отношении сторону «*многие*» и называемой *подчинённой*, этот ключ связи называется *внешним* ключом.

Связанные поля (первичный ключ главной таблицы и внешний ключ подчинённой таблицы) имеют один тип данных. Здесь существует исключение: поле счётчика может быть связано с числовым полем.

Отношение «*многие-ко-многим*» предполагает, что каждой записи первой таблицы могут соответствовать несколько записей второй таблицы, а каждой записи второй таблицы соответствуют несколько записей первой таблицы.

Связь «многие-ко-многим» в явном виде в реляционных БД не поддерживается. Однако имеется ряд способов косвенной реализации такой связи, которые с успехом возмещают её отсутствие. Один из наиболее распространённых способов заключается во введении дополнительной таблицы, строки которой состоят из внешних ключей, ссылающихся на первичные ключи двух таблиц.

Таким образом, определение связей между таблицами позволяет обеспечить правильность таблиц и столбцов. При наличии отношения «один-к-одному» или «один-ко-многим» таблицы должны содержать общие столбцы. При наличии отношения «многие-ко-многим» требуется третья таблица.

Далее необходимо определить условия целостности данных. *Условиями целостности данных* называется набор правил для поддержания связей между записями в связанных таблицах. Эти правила делают невозможным случайное удаление или изменение связанных данных.

При установке целостности данных, необходимо помнить следующие правила:

- Невозможно ввести в поле внешнего ключа связанной таблицы значение, не содержащееся в ключевом поле главной таблицы.
- Не допускается удаление записи из главной таблицы, если существуют связанные с ней записи в подчинённой таблице.
- Невозможно изменить значение первичного ключа в главной таблице, если существуют записи, связанные с данной записью.

Нормализация. Следующим этапом создания БД может стать применение правил нормализации данных. Эти правила позволяют проверить правильность структуры таблиц. Процесс применения этих правил к структуре базы данных называется нормализацией базы данных или просто нормализацией.

Нормализацию рекомендуется выполнять после внесения в базу данных всех элементов данных и создания предварительной структуры. Цель этого процесса – проверить правильность распределения элементов данных по таблицам. Однако нормализация не позволяет проверить правильность самих элементов данных.

Применение правил нормализации выполняется последовательно; на каждом этапе проверяется соответствие БД одной из т. н. «нормальных форм». Существует пять нормальных форм. Рассмотрим первые три формы, поскольку их достаточно для большинства баз данных.

Первая нормальная форма содержит правило о том, что на пересечении строки и столбца в таблице должно содержаться одно значение, а не список значений.

Вторая нормальная форма содержит требование о том, что каждый столбец, не входящий в ключ, должен находиться в зависимости от всего

ключевого столбца, а не от его части. Это правило применимо в том случае, если первичный ключ состоит из нескольких столбцов.

Третья нормальная форма содержит требование о том, что столбцы, не являющиеся ключевыми, должны не только зависеть от всего первичного ключа, но и быть независимыми друг от друга.

Таким образом, процесс разработки базы данных включает следующие действия:

- Определение цели создания базы данных.
- Поиск и организация необходимых данных.
- Распределение данных по таблицам.
- Преобразование элементов данных в столбцы. Определение, какие данные требуется хранить в каждой таблице. Каждый элемент данных занимает отдельное поле и отображается в виде столбца таблицы.
- Задание первичных ключей.
- Создание связей между таблицами.
- Применение правил нормализации.
- Усовершенствование структуры. Проверка структуру базы данных на наличие ошибок. Добавка несколько записей с образцами данных и анализ полученные результаты. Внесение в структуру необходимых изменений.

Проектирование структуры реляционной базы данных «Автомобили»

Полный цикл разработки включает концептуальное, логическое и физическое проектирование БД. Таким образом, процесс разработки базы данных включает следующие действия:

- Определение цели создания базы данных.
- Поиск и организация необходимых данных.
- Распределение данных по таблицам.
- Преобразование элементов данных в столбцы. Определение, какие данные требуется хранить в каждой таблице. Каждый элемент данных занимает отдельное поле и отображается в виде столбца таблицы.
- Задание первичных ключей.
- Создание связей между таблицами.
- Применение правил нормализации.

Создание БД начнём с фазы *логического* проектирования. Вначале определим предметную область использования учебной БД *Автомобили*.

Необходимо создать БД *Автомобили*, которая обеспечивала бы работу салона по продаже и обслуживанию автомобилей. Такая БД должна хранить информацию о характеристиках различных моделей автомобилей, о данных покупателей и обеспечивать просмотр этих данных.

БД должна решать задачи: упорядочения заказов от клиентов на продажу им автомобилей; хранения информации об автомастерских и отслеживания работ по обслуживанию и ремонту автомобилей, а также по анализу выполнения договорных обязательств по срокам и объёмам работ. БД должна обеспечивать информацией как сотрудников автосалона и потенциальных покупателей, так и клиентов, обслуживающих свои автомобили в конкретных мастерских.

Информационное обеспечение такой БД включает:

- справочную информацию о характеристиках моделей продаваемых автомобилей (содержится, например, в сервисных книжках автомобилей);
- справочную информацию о покупателях (содержится в договорах);
- справочную информацию о мастерских, где обслуживаются и ремонтируются различные модели автомобилей (содержится в справочниках различных фирм);
- учётные данные о работах по обслуживанию и ремонту автомобилей (информация с данными по фактическим расходам и работам содержится в расходных накладных со складов фирм и в договорах с клиентами на обслуживание их автомобилей).
- учётные данные о купле автомобилей (информация содержится в договорах о купле-продаже между клиентом и автосалоном).

Следующим этапом проектирования БД является распределение данных по информационным объектам (таблицам) и преобразование элементов данных в столбцы (поля). Помним, что объекты каждого класса должны храниться в отдельной таблице, и тем самым получим список таблиц будущей базы данных. Необходимо учитывать, что в каждой таблице должен быть первичный ключ, и надо добавить в каждую таблицу дополнительное поле – уникальный номер записи. В базе данных, таким образом, будет пять таблиц.

Информация о характеристиках моделей продаваемых автомобилей будет содержаться в таблице *Модели* с полями: *Код модели; Модель; Мощность двигателя; Цвет; Количество дверей; Коробка передач; Обивка; Другое оснащение; Заводская цена; Транспортные издержки; Предпродажная подготовка; Специальная модель;*

Информация о покупателях будет заноситься в таблицу *Клиенты* с полями: *Номер клиента; Фамилия; Имя; Населённый пункт; Почтовый адрес; Почтовый индекс; Телефон.*

Учётные данные о купле автомобилей будут аккумулироваться в таблице *Купля автомобилей* с полями: *Номер заказа; Номер клиента; Код модели; Скидка.*

Информация о мастерских, где обслуживаются и ремонтируются автомобили будет содержаться в таблице *Мастерские* с полями: *№ мастерской; Адрес; Категория; Телефон; Директор; Средняя цена; Наценка.*

Данные о проводимых работах по обслуживанию и ремонту автомобилей будут учитываться в таблице *Записи работ* с полями: *Код записи*; *Дата обслуживания*; *Модель*; *Описание*; *Обещанная дата*; *Дата выдачи*; *Затраченные часы*; *Оценочная цена*; *Замена оборудования*; *Закрыто*.

Далее определим функциональные зависимости между реквизитами объектов (таблиц).

Из анализа полей таблицы *Модели* очевидно, что все реквизиты являются описательными. Каждый из них функционально полно зависит только от уникального идентификатора автомобиля – *Код модели*. Таким образом, поле *Код модели* является первичным ключом. Аналогично определяем ключевое поле для таблицы *Клиенты* – *Номер клиента*, для таблицы *Купля автомобилей* – *Номер заказа*, для таблицы *Мастерские* – *№ мастерской* и для таблицы *Записи* – *Код записи*.

Очередным шагом проектирования после обозначения информационных объектов (таблиц) является определение связей между ними.

Связи между объектами *Клиенты* и *Купля автомобилей* характеризуются отношениями как *один-ко-многим*, т. к. с одним клиентом может быть заключено несколько заказов на покупку автомобилей, а один заказ всегда заключается с конкретным покупателем.

Поскольку заказ на покупку автомобиля строго привязан к конкретной модели, а по одной модели автомобиля может быть оформлено несколько заказов на покупку, между объектами *Модели* и *Купля автомобилей* имеет место связь типа *один-ко-многим*.

Очевидна связь типа *один-ко-многим* между объектами *Модели* и *Записи работ*, поскольку одна модель автомобиля может несколько раз записываться на обслуживание, а каждая запись обслуживания принадлежит только конкретной модели автомобиля. Между объектами *Мастерские* и *Модели* также существует связь типа *один-ко-многим*, поскольку каждая мастерская может обслуживать несколько моделей автомобилей, но конкретная модель приписана к определённой мастерской. Следует отметить, что объект *Купля автомобилей* фактически играет роль связного объекта в отношениях *многие-ко-многим* объектов *Клиенты* и *Модели*.

Таким образом, были определены таблицы и установлены связи между ними. Например, если в таблице *Купля автомобилей* есть поля *Номер клиента* и *Код модели*, то таблица *Купля автомобилей* ссылается и на таблицу *Клиенты*, и на таблицу *Модели*.

В соответствие с выявленными информационными объектами и связями между ними может быть построена информационно-логическая модель предметной области – БД *Автомобили* (рис. 2.1). Объекты модели размещены по уровням. На верхнем уровне размещаются объекты, не подчинённые никаким другим объектам. Уровень остальных объектов определяется наиболее длинным путём к объекту от верхнего уровня.

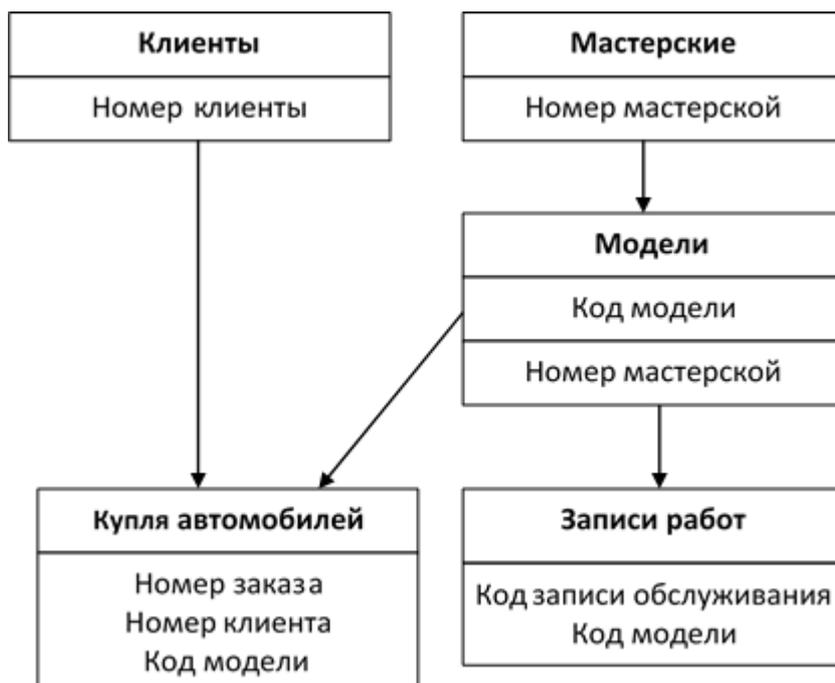


Рис. 2.1. Информационно-логическая модель БД *Автомобили*

Далее на базе полученной информационно-логической модели можно построить структуру базы данных *Автомобили* (рис.2.2). Каждый объект модели базы данных отображается соответствующей таблицей. Структура таблиц определяется составом полей соответствующих информационных объектов. Прямоугольники в схеме отображают таблицы базы данных со списком их полей. Имена ключевых полей для наглядности выделены курсивом и находятся в верхней части полного списка полей каждой таблицы. Определение связей в структуре базы данных осуществляется в соответствии с построенной моделью.

Связи показывают по каким полям осуществляется взаимосвязь таблиц. Между объектами связи реализуются с помощью одинаковых ключей связи в соответствующих таблицах. При этом между таблицами ключом связи типа *один-ко-многим* всегда является первичный ключ главной таблицы. При этом, ключ связи в подчинённой таблице называется внешним ключом. В информационно-логической модели БД *Автомобили* все связи характеризуются отношением типа *один-ко-многим*.

Связь между таблицами *Клиенты* и *Купля автомобилей* осуществляется по *Номеру клиента*, который является первичным ключом главной таблицы *Клиенты* и вторичным (не ключевым) в таблице *Купля автомобилей*. Связь между таблицами *Мастерские* и *Модели* осуществляется по первичному ключу *№ мастерской* главной таблицы *Мастерские*, который в подчинённой таблице *Модели* является вторичным ключом. Связь между таблицами *Мастерские* и *Модели* осуществляется по первичному ключу *№ мастер-*

ской главной таблицы *Мастерские*, который в подчинённой таблице *Модели* является вторичным ключом.

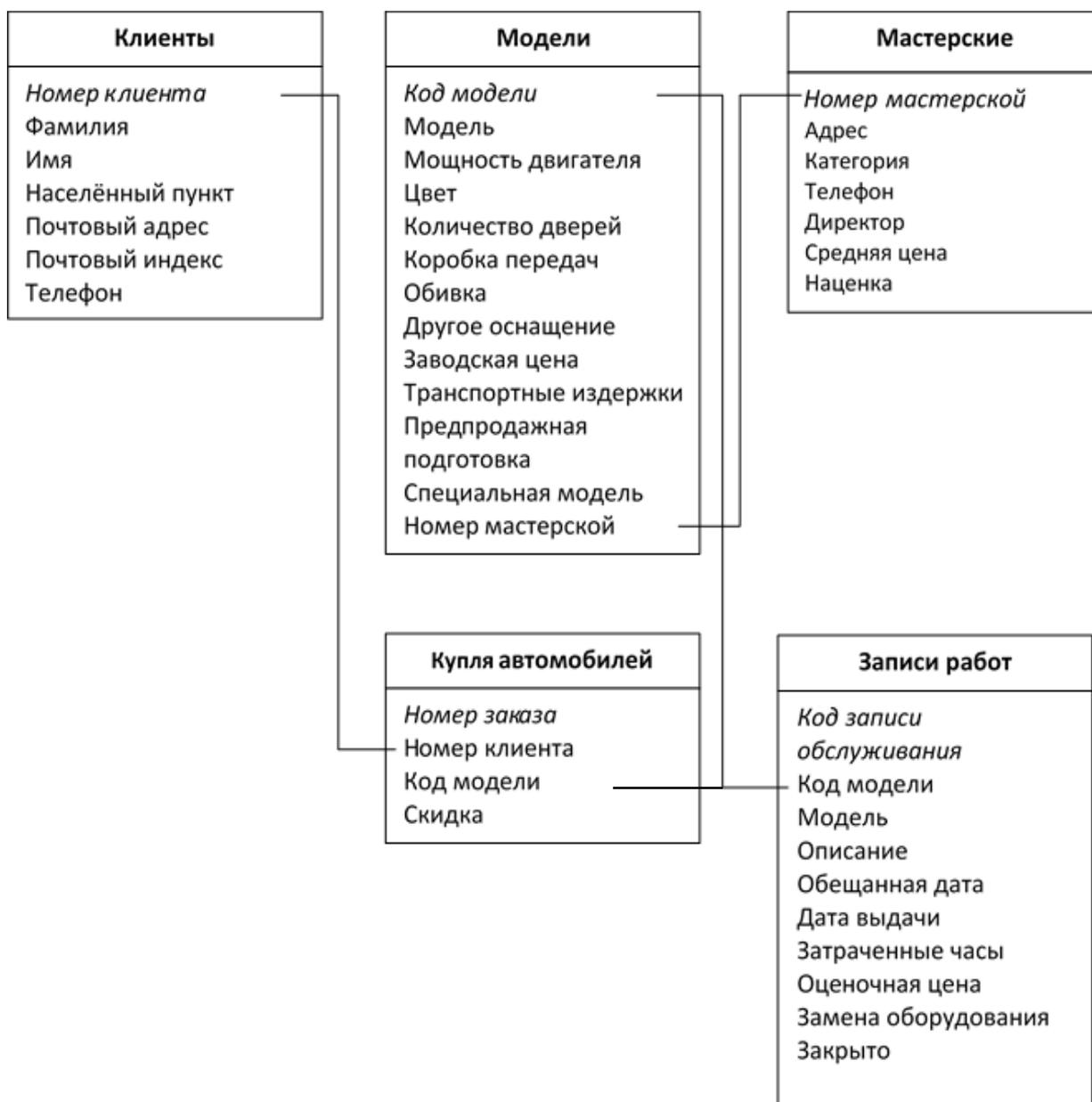


Рис. 2.2. Структура базы данных *Автомобили*

Связь между таблицами *Мастерские* и *Модели* осуществляется по первичному ключу *№ мастерской* главной таблицы *Мастерские*, который в подчинённой таблице *Модели* является вторичным ключом. Связь между таблицами *Модели* и *Записи работ*, а также *Модели* и *Купля автомобилей* осуществляется по первичному ключу главной таблицы *Модели* – *Коду модели*.

Таким образом, нами разработана предварительная структура и можно приступить к непосредственному построению БД в программе Microsoft Access.

2. ОСНОВЫ РАБОТЫ В ПРОГРАММЕ MS ACCESS 2013

Запуск Access

В нижней части рабочего стола ОС Windows на панели задач нажать кнопку *Пуск*, в открывшемся меню выбрать *Все программы*. Из списка программ, установленных на ПК, выбрать *Microsoft Office / Access 2013* и запустить СУБД. Access может быть запущен из списка недавно использовавшихся программ, представленного также в меню *Пуск* или, щёлкнув в соответствующей папке на файле ранее созданной базы данных Access. При этом сразу будет открыта эта БД. Если предварительно создать ярлык программы на рабочем столе (рис. 3.1.), можно быстро запустить Access двойным щелчком мыши на этом ярлыке.



Рис. 3.1. Ярлык для запуска программы MS Access 2013

Сразу после запуска программы отображается стартовое окно Access 2013 (рис. 3.2).



Рис. 3.2. Стартовое окно программы MS Access 2013

Здесь представлен список последних использованных баз данных, команда *Открыть другие Файлы* баз данных, в области создания баз данных предоставлена возможность создать новую базу: *Пустая база данных рабочего стола* и *Пользовательское веб-приложение*. Кроме того, если ПК подключён к Интернету, то отображаются шаблоны для создания различных типовых баз данных. Шаблоны типовых баз данных включают все необходимые объекты для предметных областей различных сфер деловой и личной жизни. База данных готова к использованию и содержит все необходимые для работы таблицы, формы, отчёты, запросы, макросы и связи. Эти стандартные приложения можно использовать без какой-либо модификации и настройки, либо взять их за основу и адаптировать в соответствии с характером информации, которую требуется сохранять и обрабатывать. Пользователь может только ввести необходимые данные. Типовые БД, созданные с помощью шаблонов, позволяют пользователю познакомиться с основными принципами построения базы данных и получить определённые навыки практической работы в среде MS Access.

Существуют шаблоны для создания локальных баз данных (например, рис. 3.3. а) – рисунок не включает значок земного шара) и для создания веб-приложений, доступных в Интернете (например, рис. 3.3. б) – рисунок включает значок земного шара). При этом открываются различные окна (соответственно рис. 3.4. и рис. 3.5.), где нужно определить имя и место расположения базы данных. Для размещения веб-приложения необходимо иметь доступ к сайту SharePoint.

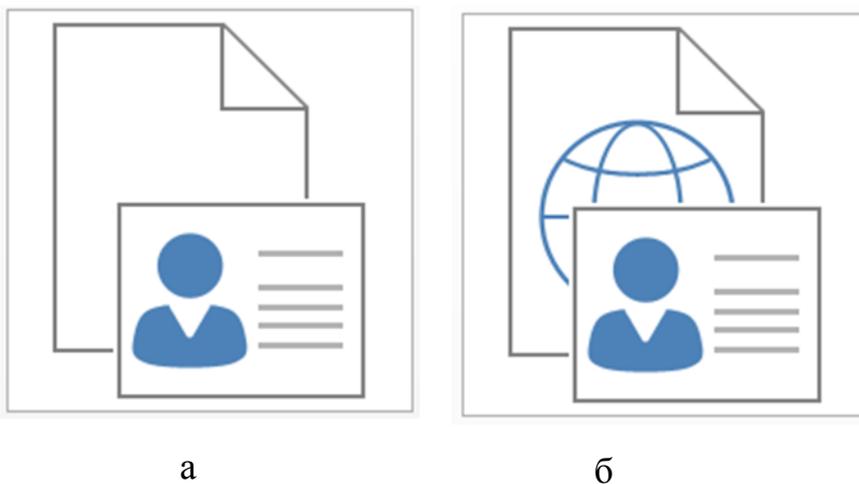


Рис. 3.3. Шаблоны для создания баз данных
а) – создаются локальные базы данных,
б) – создаются веб-приложения, доступные в Интернете.

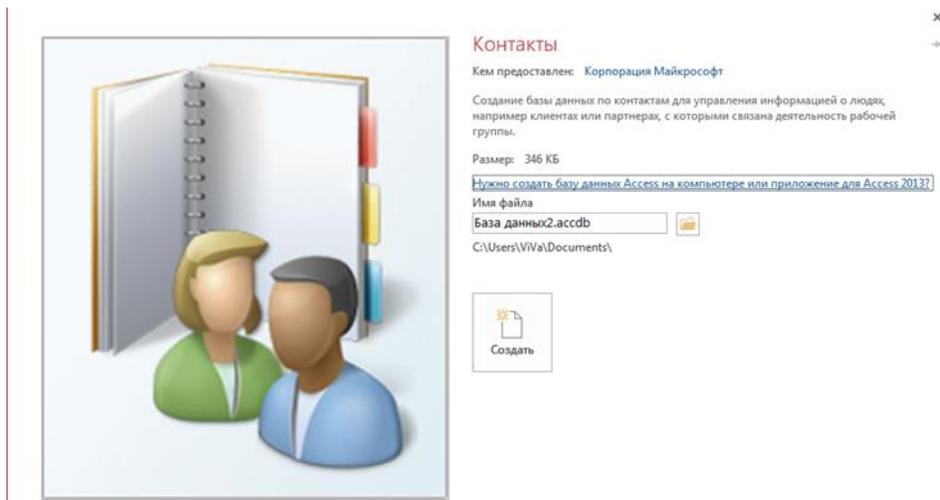


Рис. 3.4. Окно определения имени и места расположения базы данных на локальном ПК, создаваемого на основе шаблона *Контакты*

Технологии Access 2013 позволяют создавать специальные веб-приложения, опубликованные на сайтах Microsoft SharePoint Server 2013. Они предназначены для организации совместной работы с данными и взаимодействия пользователей. Доступ пользователей к содержимому веб-сайтов осуществляется через веб-браузер или Access 2013 из любой точки корпоративной сети или Интернета. Веб-приложение можно разместить как на сайтах корпоративного сервера SharePoint, так и на сайтах Office 365 в «облаках». Для доступа к сайтам необходимо использовать выпуски Office 365, обеспечивающие среду SharePoint 2013. С помощью SharePoint организации создают веб-сайты, которые можно использовать как безопасное пространство для хранения, систематизации, совместного использования данных и организации доступа к ним почти с любого устройства. Понадобится только веб-браузер, например Internet Explorer, Chrome или Firefox.

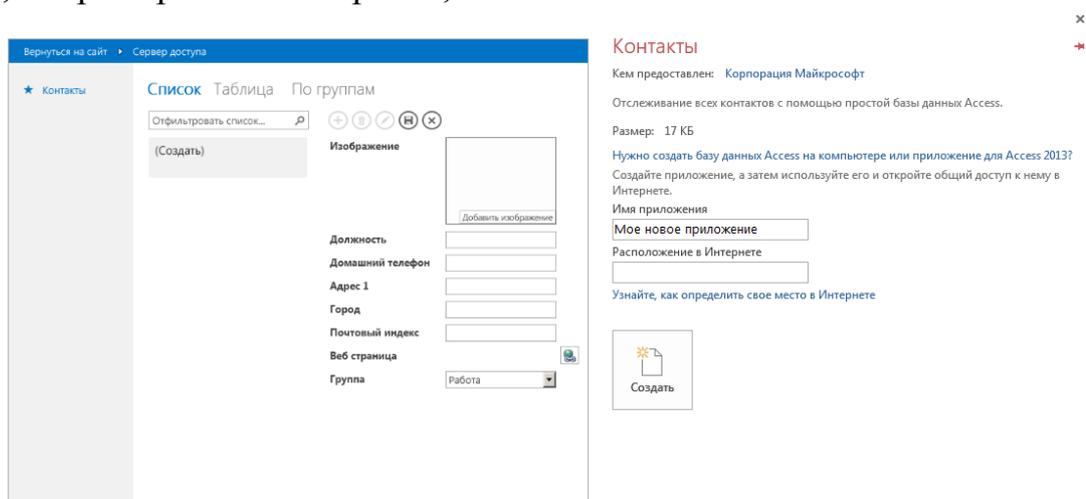


Рис. 3.5. Окно определения имени и места расположения веб-приложения, создаваемого на основе шаблона *Контакты*

Пользователи могут совместно работать с веб-приложением с помощью стандартного браузера, не устанавливая приложение Access на компьютере. В то же время сохраняется возможность изменять, дорабатывать веб-приложение в Access.

Пример. Создадим и сохраним пустую базу данных с именем *Автомобили*.

Для создания новой базы необходимо выбрать шаблон: *Пустая база данных рабочего стола*. При этом открывается окно (рис. 3.6.), где надо указать имя и папку расположения создаваемой БД, затем нажать кнопку *Создать*.

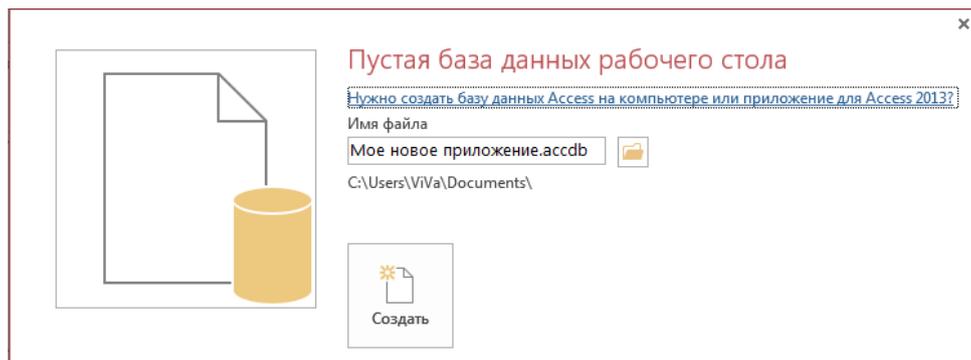


Рис. 3.6. Окно определения имени и места расположения БД на локальном ПК, создаваемого на основе шаблона *Пустая база данных рабочего стола*

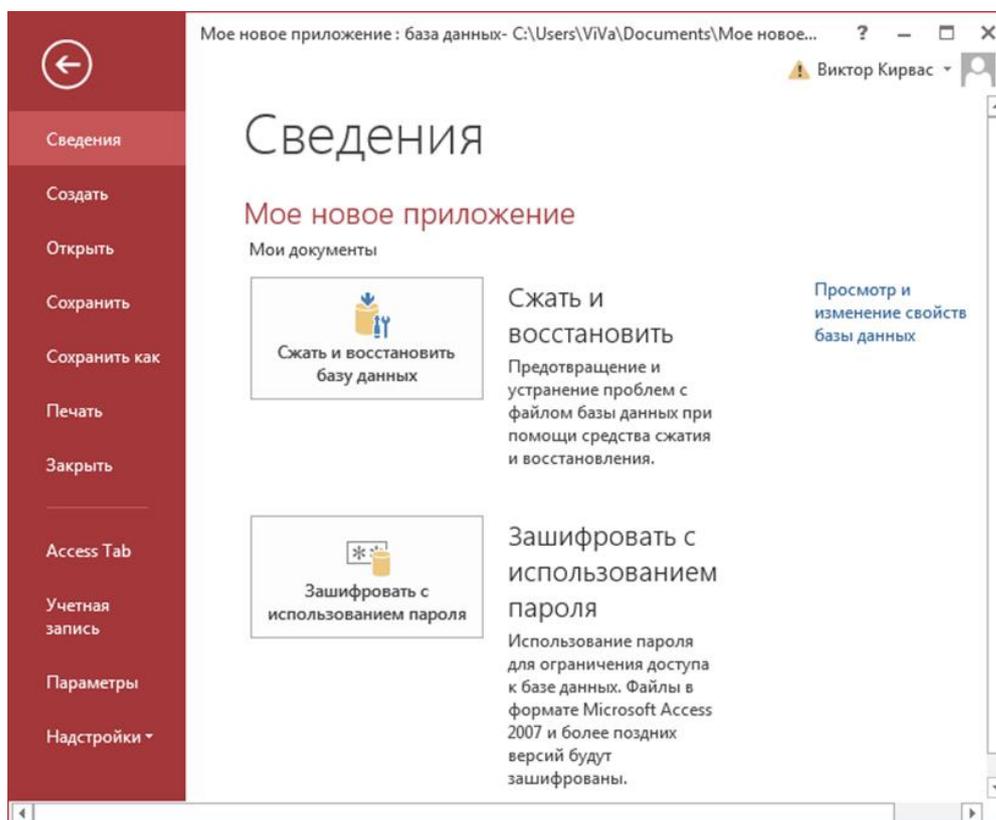


Рис. 3.7. Окно, открываемое значком *ФАЙЛ*

После сохранения новой или при открытии ранее сохранённой БД на экране появляется главное окно Access (рис. 3.9). Щелчок на цветном значке ФАЙЛ открывает окно (рис. 3.7.) с командами, относящимися, прежде всего, к базе данных в целом.

Здесь находятся команды *Сохранить*, *Сохранить как*, *Закреть* базу данных. В разделе *Сведения* можно *Сжать и восстановить базу данных*, *Зашифровать с использованием пароля*, выполнить *Печать* текущего объекта БД.

На странице *Параметры Access* (рис. 3.8.) выполняется настройка основных параметров Access, параметров текущей базы данных, а также безопасности документов и компьютера. Можно открыть *Справку Access*, щёлкнув по значку вопроса в заголовке окна.

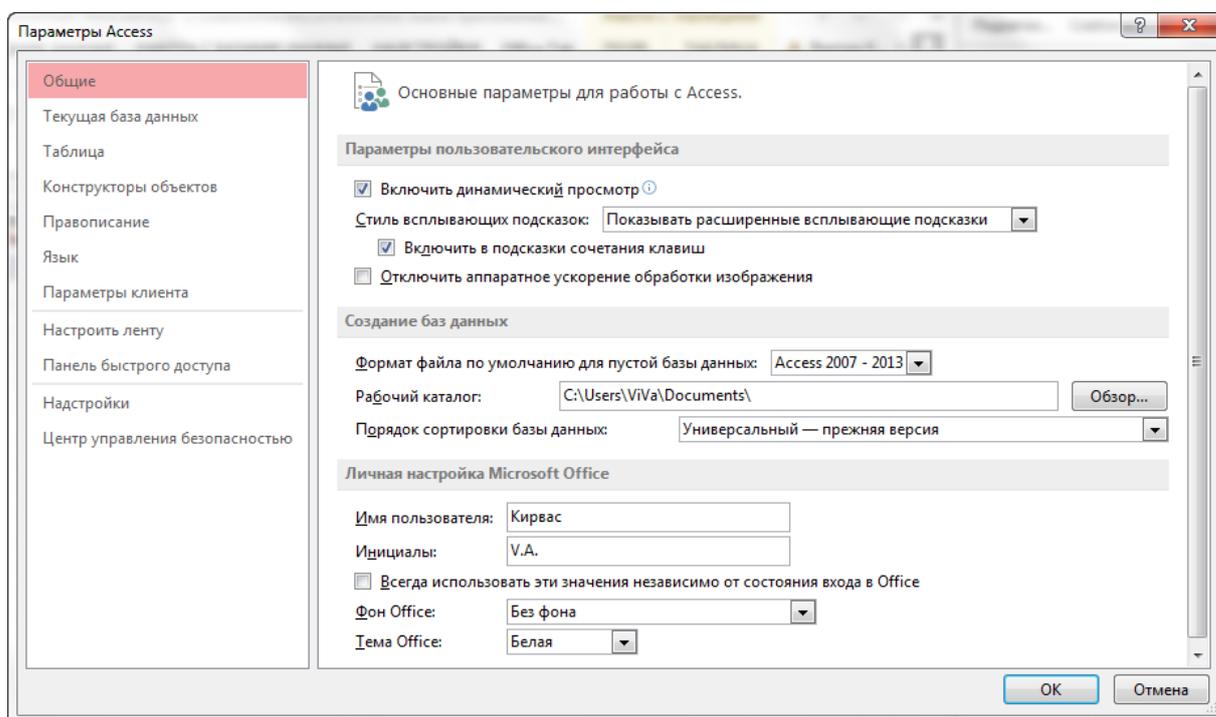


Рис. 3.8. Окно *Параметры Access*

Интерфейс пользователя Access

Основу интерфейса пользователя Access 2013 составляют ленты и область навигации. Собранные на одной ленте команды чётко соответствуют задачам, выполняемым в Access, что позволяет легко находить нужную команду. На экране отображаются только те команды, которые имеют отношение к задаче, выполняемой в данный момент.

Основные элементы главного окна Access 2013 приведены на рис. 3.9.

Вкладка *ФАЙЛ*, выделенная цветом открывает набор команд, отображаемых на отдельной странице (рис. 3.7.)

Панель быстрого доступа – на ней целесообразно располагать значки наиболее часто используемых команд, чтобы вызывать их одним щелчком кнопки мыши и тем самым значительно ускорять работу. Она может размещаться под лентой или над ней. Первоначально панель содержит лишь команды *Сохранить*, *Отменить*, *Вернуть* и *Настроить панель быстрого доступа*, однако её легко дополнить любыми командами и даже целыми группами.

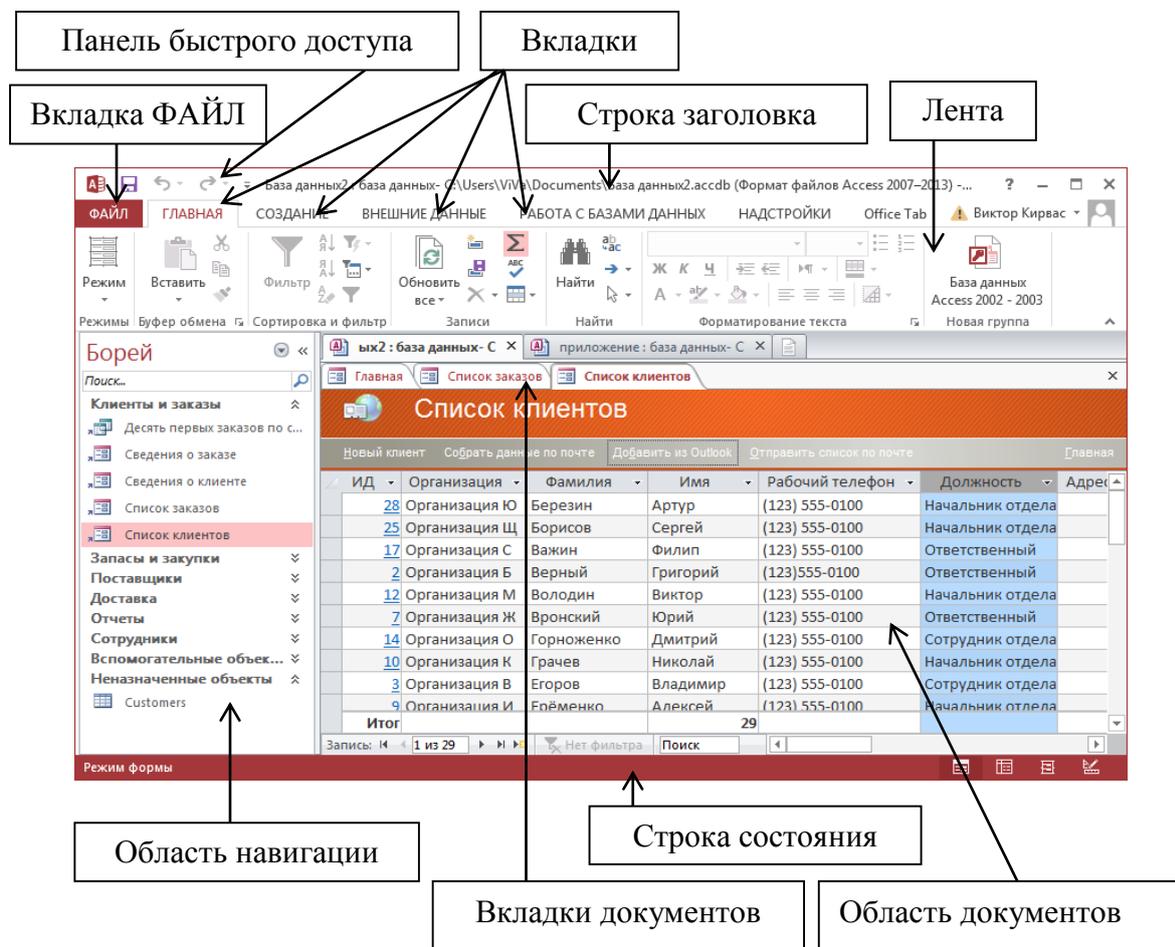


Рис. 3.9. Главное окно Access

Лента – ключевой элемент интерфейса, объединяющий все команды работы с базой данных. Лента состоит из *стандартных вкладок* (*Главная*, *Создание*, *Внешние данные* и *Работа с базами данных*), на каждой из которых расположены значки нескольких групп команд и *контекстные вкладки*, которые появляются только тогда, когда их использование допустимо. Контекстные вкладки отображаются рядом со стандартными вкладками в зависимости от того, с каким объектом работает пользователь, и какие действия он выполняет. В каждом режиме Access отображает свои вкладки и, соответственно, свой набор команд, представляющий всю его функциональность.

Область навигации отображает список документов базы данных (таблиц, запросов, форм, отчётов, макросов, модулей) и позволяет переключаться между ними. Она обеспечивает быстрый доступ к нужным объектам, которые для удобства пользователя могут быть объединены не только в стандартных категориях и группах, но и в созданных пользователем в соответствии с функциональностью приложений. Для представления объектов по группам надо щёлкнуть на кнопке списка в заголовке области навигации, выбрать категорию *Тип объекта* и фильтр *Все объекты Access* (рис. 3.10.).

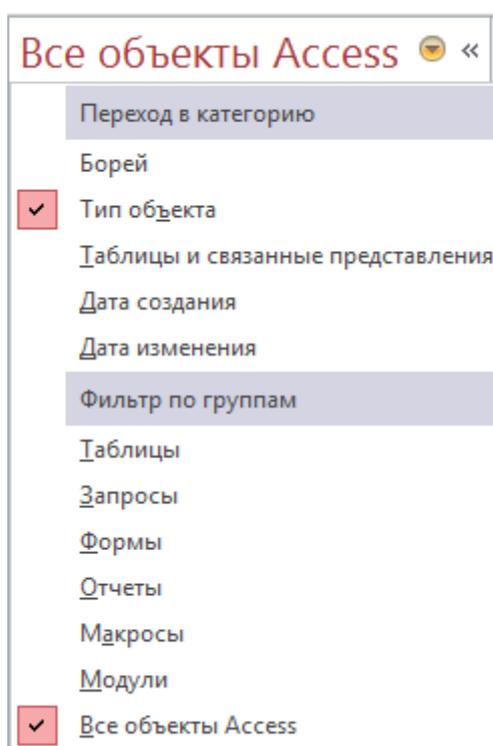


Рис. 3.10. Панель области навигации для выбора категории и отображения группы объектов

Чтобы открыть документ базы данных, достаточно дважды щёлкнуть кнопкой мыши на его названии в области навигации. Для скрытия и открытия области навигации служит кнопка в правом верхнем углу области.

Вкладки документов – это таблицы, запросы, формы, отчёты и макросы, которые отображаются на вкладках в рабочем пространстве окна Access – в *области документов*. Этот режим работы по умолчанию используется для всех баз данных, созданных с помощью Access 2013.

Строка состояния отображает подсказки и сообщения о состоянии программы, а также содержит ряд кнопок, позволяющих быстро переключаться между различными режимами работы с текущим документом.

Панель быстрого доступа первоначально включает лишь кнопки *Сохранить*, *Отменить*, *Вернуть* и кнопку *Настройка панели быстрого доступа*, открывающую список команд, которыми может быть дополнена панель. Поскольку работа с данной панелью в программах Access, Word и Excel ничем не отличается, далее в пособии рассматриваться не будет.

Работа с таблицами базы данных

Таблица содержит данные об одном информационном объекте или сущности, например, сведения о клиентах или моделях автомобиля. Каждая запись (строка) в таблице включает данные об одном экземпляре сущности, например, о конкретном клиенте. Запись состоит из полей (столбцов), каждое из которых включает один из атрибутов сущности, таких как *номер клиента, фамилия, имя, населённый пункт, почтовый адрес, почтовый индекс, телефон*. Для однозначной идентификации записи в таблице определяется *ключевое поле*, например, *номер клиента*. База данных чаще всего содержит много таблиц. Разработка БД начинается с создания этих таблиц.

Создание таблицы БД состоит из двух этапов. На первом этапе определяется её структура: состав полей, их имена, тип данных каждого поля, размер поля, ключи, индексы таблицы и другие свойства полей. На втором этапе производится заполнение таблицы данными.

В Access существуют два режима для создания таблицы: *режим конструктора* и *режим таблицы*. *Режим конструктора* позволяет максимально полно определить структуру таблицы, а *режим таблицы* предназначен, прежде всего, для создания, просмотра, поиска, корректировки её записей.

Создание таблицы в режиме конструктора

В режиме конструктора создаются новые объекты базы данных или изменяются макеты существующие. Для создания таблицы в режиме конструктора необходимо на вкладке *Создание* в группе *Таблицы* выбрать команду *Конструктор таблиц* (рис. 3.11). В режиме конструктора таблиц открывается окно *Таблица1*, в котором определяется структура таблицы базы данных (рис. 3.12).

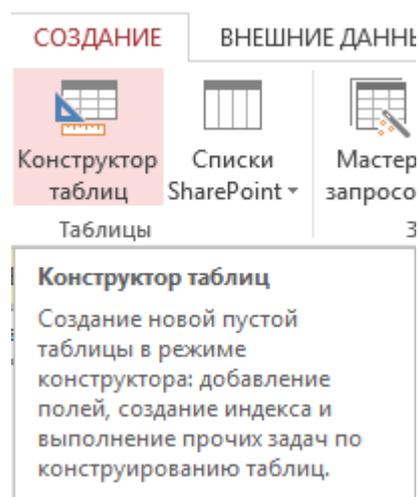


Рис. 3.11. Переход в режим конструктора для определения структуры таблицы

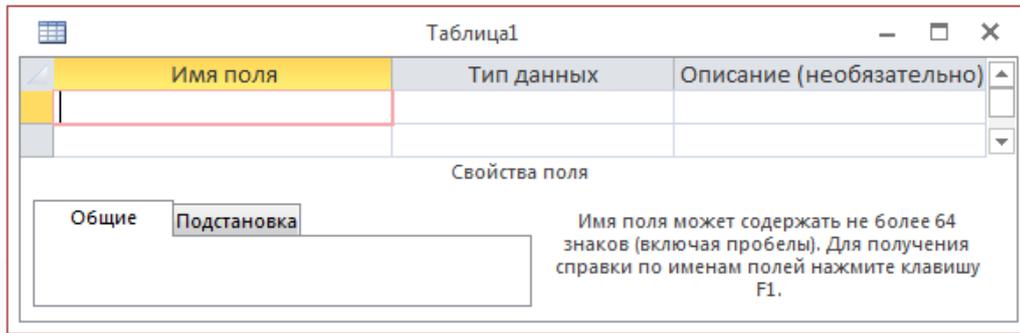


Рис. 3.12. Окно *Таблица1* в режиме конструктора таблиц

Переход из *режима конструктора* в *режим таблицы* и наоборот осуществляется через кнопку *Режим* (рис. 3.13, а), доступной на вкладках *Работа с таблицами / Поля* и *Главная*. Кроме того, перейти из одного режима в другой можно щёлкнув, в строке состояния, на соответствующей кнопке *Конструктор* или *Режим таблицы* (рис. 3.13, б), а также можно щёлкнуть правой кнопкой мыши на заголовке таблицы и в контекстном меню выбрать нужную команду (рис. 3.13, в).

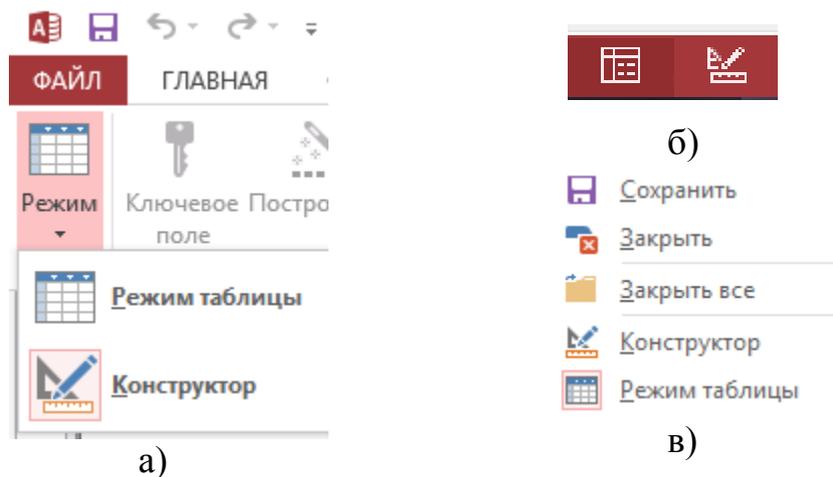


Рис. 3.13. Кнопки перехода в различные режимы создания таблиц
 а) кнопка *Режим*; б) кнопки строки состояния;
 в) кнопки контекстного меню заголовка таблицы.

При переходе в *режим конструктора таблиц* активной становится вкладка команд *Работа с таблицами / Конструктор* (рис. 3.14).

При наведении курсора мыши на любую кнопку ленты всплывает назначение кнопки (например, кнопка *Схема данных* на рис. 3.14).



Рис. 3.14. Вкладка команд *Работа с таблицами / Конструктор*

В окне конструктора таблицы (см. рис. 3.12) определяются поля: задаются имена полей, выбираются типы данных и вводится описание полей. Чтобы вставить в таблицу поле надо щёлкнуть строку, над которой его нужно поместить, и нажать кнопку *Вставить строки* в группе *Сервис* на вкладке *Работа с таблицами / Конструктор* (см. рис. 3.14). Имена полей могут содержать до 64 знаков (цифр или букв), включая пробелы.

В разделе *Свойства поля* на двух вкладках (*Общие* и *Подстановка*) задаются свойства (рис. 3.15), которые определяют функциональные характеристики и внешний вид этого поля. К общим свойствам относятся: размер поля, формат поля, число десятичных знаков, маска ввода, подпись (она выводится в заголовке столбца таблицы), значение по умолчанию и другие.

Общие	Подстановка
Размер поля	Длинное целое
Формат поля	Основной
Число десятичных знаков	0
Маска ввода	
Подпись	
Значение по умолчанию	0
Правило проверки	
Сообщение об ошибке	
Обязательное поле	Да
Индексированное поле	Да (Совпадения не допускаются)
Выравнивание текста	Общее

Рис. 3.15. Пример вкладки *Общие* раздела *Свойства поля*

На вкладке *Подстановка* (рис. 3.16) выбирается *Тип элемента управления* (поле, список или поле со списком) и другие свойства поля. Если для поля выбран тип элемента управления *Список* или *Поле со списком*, на вкладке *Подстановка* появляются дополнительные свойства, которые определяют источник строк списка и ряд других характеристик списка. Однако не все типы данных позволяют использовать поле со списком.

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Список значений
Источник строк	"Красный";"Черный";"Зеленый";"Вишневый";"Белый";"Голуб
Присоединенный столбец	1
Число столбцов	1
Заголовки столбцов	Нет
Ширина столбцов	2,54см
Число строк списка	8
Ширина списка	3см
Ограничиться списком	Нет
Разрешить несколько значений	Нет
Разрешить изменение списка значен	Да
Форма изменения элементов списка	
Только значения источника строк	Нет

Рис. 3.16. Пример вкладки *Подстановка* раздела *Свойства* поля

Основные параметры полей таблицы (типы данных, а также основные свойства полей), которые могут быть заданы в Access, приведены в Табл. 1 и Табл. 2. (стр.22-25). Свойства поля зависят от выбранного типа данных и отображаются при установке курсора на строке соответствующего поля.

В столбце *Тип данных* (рис. 3.17) можно оставить настройку по умолчанию (*Короткий текст*) или выбрать из раскрывающегося списка другой тип данных.

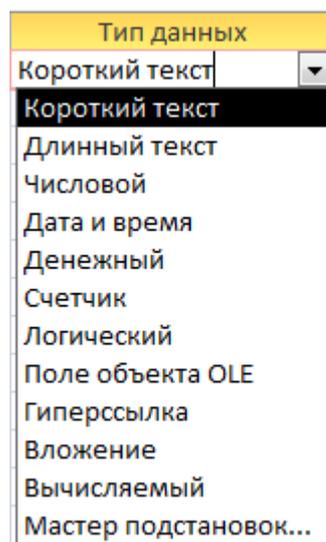


Рис. 3.17. Список столбца *Тип данных*

Определение первичного ключа

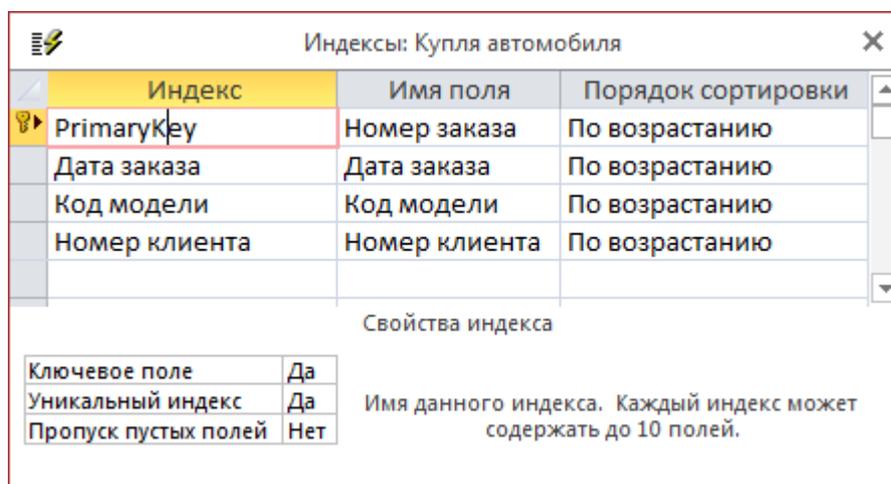
В каждой таблице реляционной БД устанавливается уникальный (первичный) ключ, однозначно определяющий каждую запись (строку) в таблице, что позволяет быстро найти нужную запись. Это поле служит также для связи данных из разных таблиц в запросах, формах и отчётах. Ключевое поле

должно содержать уникальные значения, такие как коды, номера индивидуальных планов студентов, инвентарные номера и т.п., и не допускает пустых полей в записях. В столбце ключевого поля обязательно должны быть данные, и они не должны быть одинаковыми в его полях. Ключ может быть простым или составным, включающим несколько полей.

Назначение первичного ключа таблицы, его изменение или удаление осуществляется в режиме конструктора. Для определения ключа нужно выделить поле (или несколько полей), которые будут составлять ключ, и на вкладке ленты *Работа с таблицами / Конструктор* в группе *Сервис* следует нажать кнопку *Ключевое поле* (рис. 3.14.) или в контекстном меню поля выбрать команду *Ключевое поле*.

Для ключевого поля автоматически строится уникальный индекс с именем PrimaryKey. Кроме того, автоматически строятся неуникальные индексы по вторичным полям таблицы, определяющим связь с главной таблицей. Просмотр и редактирование индексов таблицы можно осуществлять в окне *Индексы*, которое вызывается щелчком на соответствующей кнопке вкладки ленты *Работа с таблицами / Конструктор* (рис. 3.14).

На рис. 3.18 приведено окно *Индексы* таблицы *Купля автомобиля*. Первым представлен уникальный индекс с именем PrimaryKey, построенный по ключевому полю *Номер заказа*. В трёх следующих строках представлены не-уникальные индексы, построенные по вторичным ключам таблицы: *Дата заказа*, *Код модели* и *Номер клиента*.



Индекс	Имя поля	Порядок сортировки
PrimaryKey	Номер заказа	По возрастанию
Дата заказа	Дата заказа	По возрастанию
Код модели	Код модели	По возрастанию
Номер клиента	Номер клиента	По возрастанию

Свойства индекса

Ключевое поле	Да
Уникальный индекс	Да
Пропуск пустых полей	Нет

Имя данного индекса. Каждый индекс может содержать до 10 полей.

Рис. 3.18. *Индексы* таблицы *Купля автомобиля*

В качестве первичного ключа в таблице может быть задано одно поле с типом данных *Счетчик*. В этом случае при добавлении каждой новой записи в таблицу в это поле автоматически по умолчанию вводятся уникальные целые, последовательно возрастающие числа. Можно в свойствах поля выбрать опцию *случайные* новые значения. Указание такого поля является наиболее простым способом создания первичного ключа. Значения этого поля невоз-

можно изменить или удалить. Если до сохранения вновь созданной таблицы первичный ключ не установлен, Access задаст вопрос «Создать ключевое поле сейчас?». При ответе «Да» создаётся первичный ключ с типом данных *Счетчик*.

Пример. Создание в БД *Автомобили* таблицы *Модели*, структура которой была определена в главе 2.

Создание таблицы *Модели* начнём с формирования её структуры в режиме конструктора таблиц. На вкладке ленты *Создание* в группе *Таблицы* необходимо выполнить команду *Конструктор таблиц* (рис. 3.11). В окне конструктора (рис. 3.12) в соответствии с данными, приведёнными в табл. П1 Приложения 1, следует определить все характеристики полей таблицы *Модели*: задать *Имя поля*, выбрать *Тип данных* (см. рис. 3.17) и ввести *Описание*. Затем, последовательно устанавливая курсор на имя поля, в разделе *Свойства поля* на вкладке *Общие* (см. рис. 3.15) нужно задать свойства соответствующих полей, согласно данным приведённым в табл. П2 Приложения 1.

Совет: Изменяйте только свойства, приведённые в таблице, остальные оставьте по умолчанию. Выбирайте характеристики из открывающегося списка или, если его нет, задавайте самостоятельно.

Заметим, что свойство поля *Правило проверки* является выражением, и оно может быть сформировано с помощью *Построителя выражений* (рис. 3.19). Он вызывается в окне конструктора таблиц (см. рис. 3.12). Для этого выделив поле, в которую должно быть введено выражение, в разделе *Свойства поля* на вкладке *Общие* (см. рис. 3.15) в строке *Правило проверки*, надо нажать на кнопку (три точки), расположенной справа. Построитель выражений можно вызвать и командой *Построитель* на вкладке *Работа с таблицами / Конструктор* в группе *Сервис* (см. рис. 3.14), а также запустить его с помощью сочетания клавиш *Ctrl+F2*.

Окно свойств таблицы в режиме конструктора открывается кнопкой *Страница свойств*. Для таблицы может быть введено только одно выражение для проверки данных в таблице.

В нашем примере (таблица *Модели*), для поля *Количество дверей*, правило проверки необходимо установить «2 Or 4», а для поля *Коробка передач* – «Ручная Or Автоматика».

Построитель выражений можно вызывать и для оформления свойства поля *Значение по умолчанию*.

После того как было введено выражение в окно построителя, нажата кнопка *ОК* и выполнен синтаксический анализ выражения, в строке *Правило проверки* отобразится это выражение.

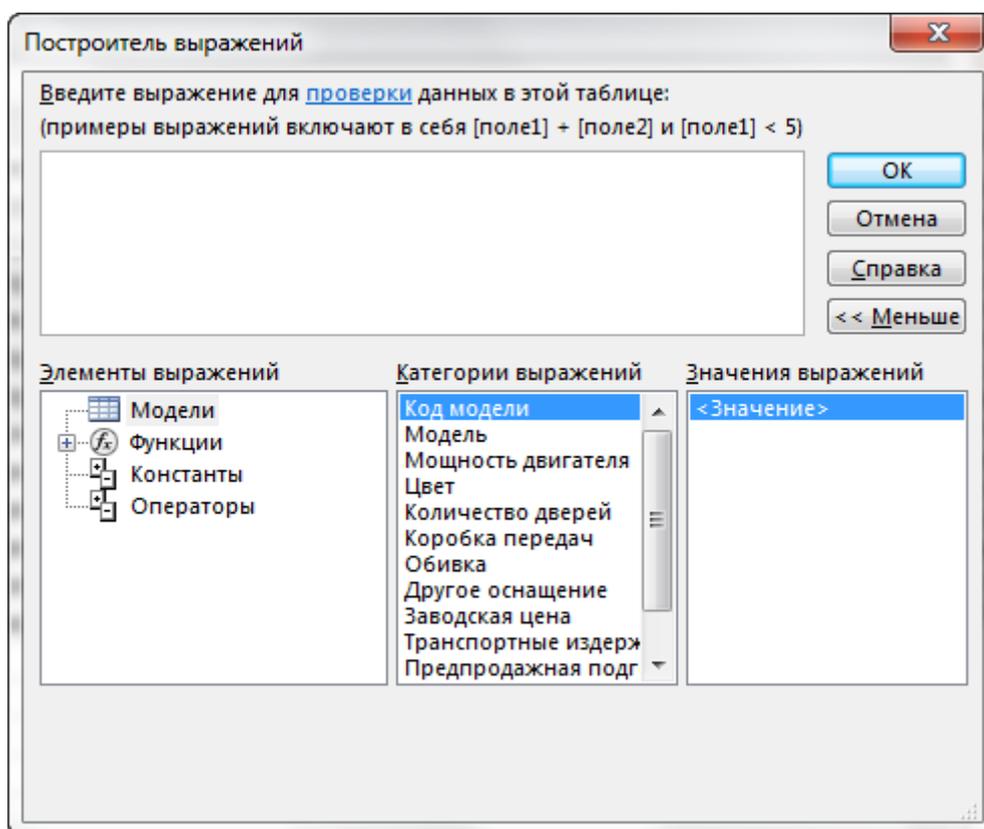


Рис. 3.19. Окно построителя выражений

Создание поля с фиксированным списком

Список возможных значений поля называется *подстановкой*. Если для поля создана подстановка, то значение поля в таблице можно выбрать из раскрывающегося списка, вместо того чтобы вводить вручную с клавиатуры. При этом упрощается и ускоряется процесс ввода данных в созданную таблицу и исключаются ошибки. Процедура создания подстановки зависит от типа источника данных. Источником значений для подстановки может быть другая таблица, запрос или созданный пользователем фиксированный список.

Для создания фиксированного списка необходимо выполнить следующие действия:

- Перейти в режим конструктора таблицы.
- Установить курсор в столбец *Тип данных* того поля, для которого предполагается создать фиксированный список.
- Из выпадающего списка типов данных (см. рис. 3.17) выбрать последний вариант – *Мастер подстановок*.
- Открывается окно первого шага мастера (рис. 3.20), в котором надо назначить способ формирования списка. Выбрать вариант «Будет введен фиксированный набор значений»;

- Завершить выполнение данного шага, как и всех последующих, нажатием кнопки *Далее*.

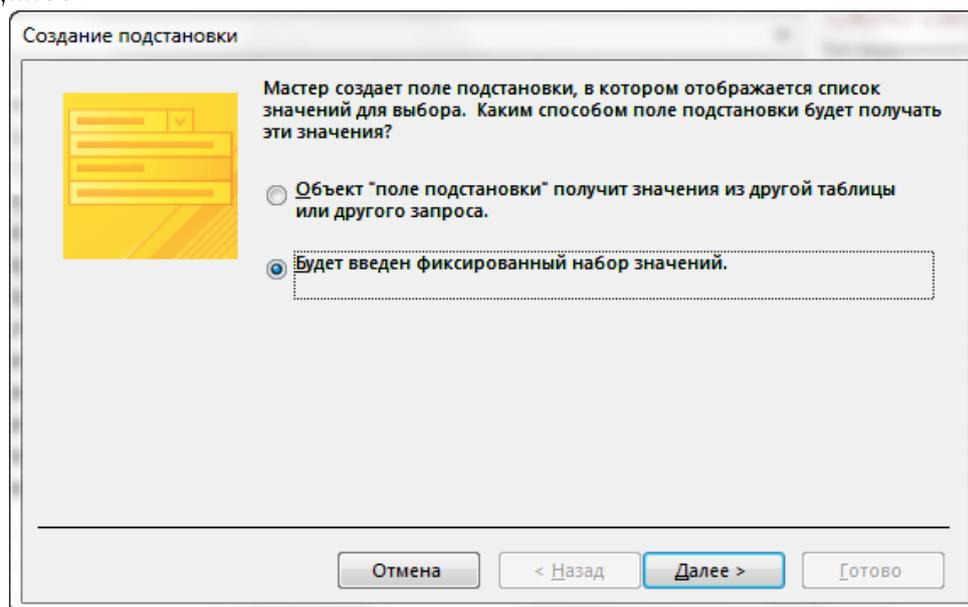


Рис. 3.20. Диалоговое окно «Создание подстановки»
(шаг 1 – выбор способа формирования списка)

- На втором шаге работы мастера в появившемся окне (рис. 3.21) задать число столбцов списка, а в столбце подстановки ввести в каждую строчку по одному элементу фиксированного списка;

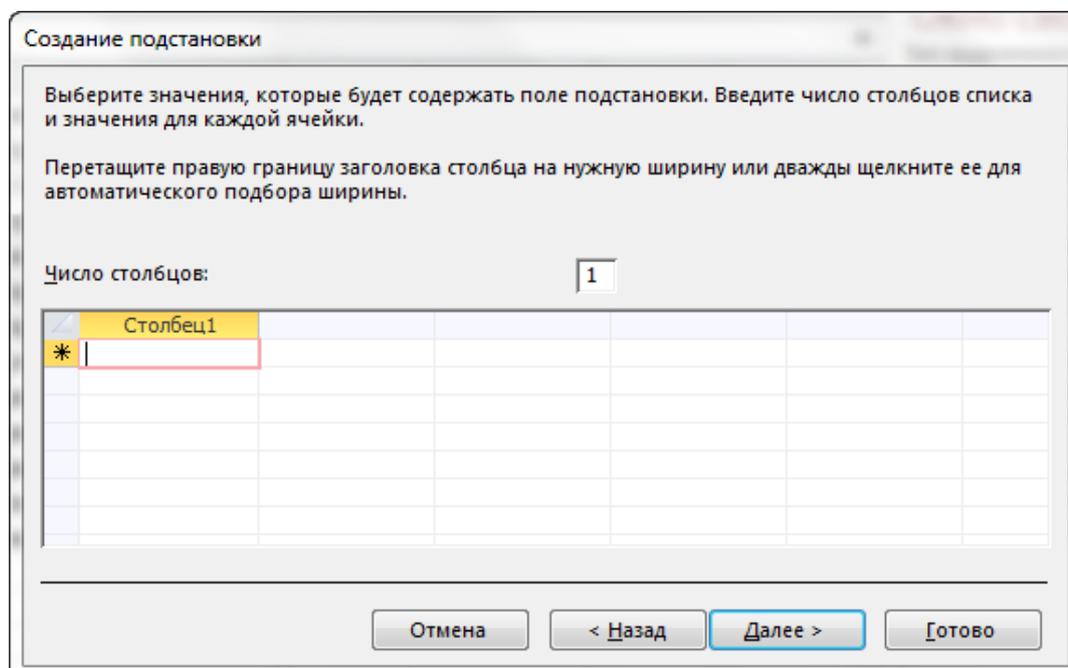


Рис. 3.21. Диалоговое окно «Создание подстановки»
(шаг 2 – ввод значений списка)

- На завершающем шаге работы мастера (рис. 3.22) задать имя списка подстановки, с которым будет связан созданный список. В поле со списком по умолчанию можно вводить значения, не предусмотренные в списке. Чтобы пользователи могли вводить только те значения, которые содержатся в списке нужно установить флажок *Ограничиться списком*. Чтобы пользователи могли выбрать более одного значения для поля установите флажок *Разрешить несколько значений*. Для завершения работы *Мастера подстановки* нажать кнопку *Готово*.

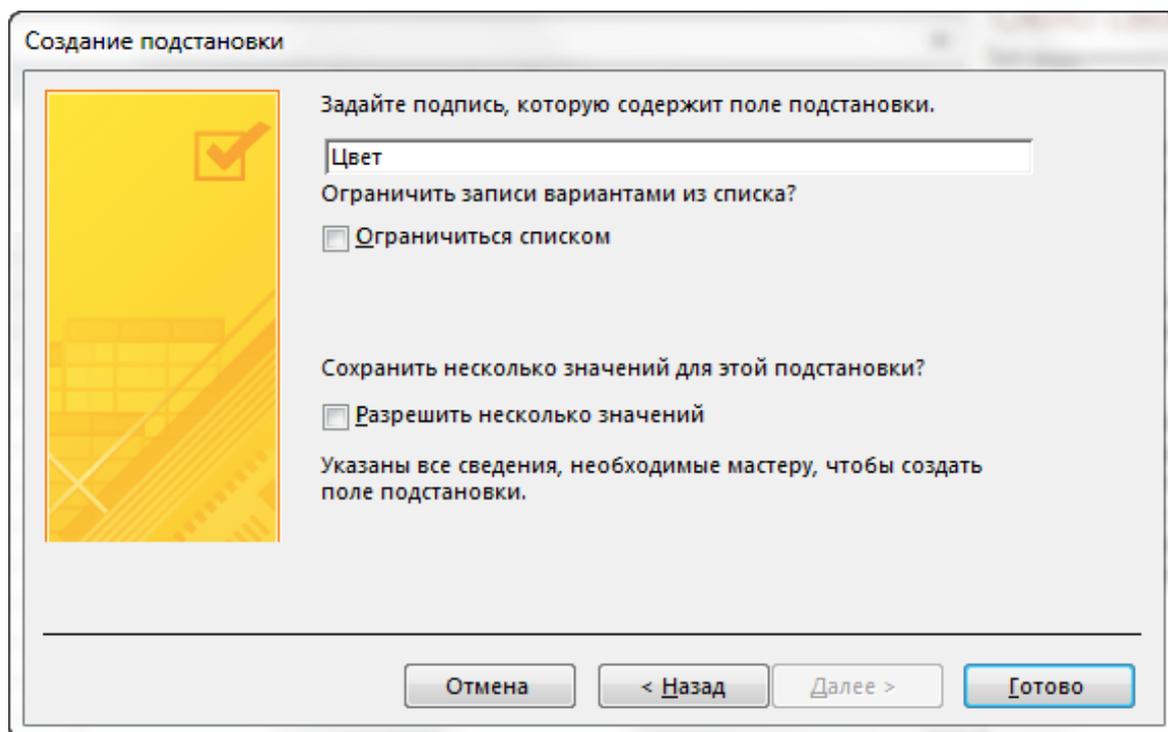


Рис. 3.22. Диалоговое окно «Создание подстановки»
(шаг 3 – ввод имени списка подстановки и режима ограничения списком.)

Теперь для ввода значений в данное поле таблицы достаточно будет выбрать название из выпадающего списка (рис. 3.23).

Параметры списка, подготовленные мастером, отображаются в свойствах поля (например, поле *Цвет* таблицы «Модели») на вкладке *Подстановка* раздела *Свойства поля* (см. рис. 3.16). Здесь можно изменить свойства поля, например, выбрав опцию *Да* или *Нет* в строчках: *Ограничиться списком*, *Разрешить несколько значений* или *Разрешить изменение списка значений*.

Мощность двигателя ▾	Цвет ▾	Количество дверей ▾
100/130	Красный	4
80/100	Черный	4
60/90	Белый	4
53/75	Красный	4
69/90	Зеленый	4
90/135	Голубой ▾	2
100/139	Красный	4
	Черный	4
	Зеленый	
	Вишневый	
	Белый	
	Голубой	
	Синий	

Рис. 3.23. Ввод значения в поле выбором из списка

Далее определим в БД *Автомобили* ключевое поле таблицы *Модели*.

Для этого нужно выделить поле *Код модели* и нажать кнопку *Ключевое поле* в группе *Сервис* на вкладке ленты *Работа с таблицами / Конструктор* (см. рис. 3.14). При этом слева от имени поля появляется изображение ключа. Следует заметить, что часто вместо числового устанавливают текстовый тип данных кодов, индивидуальных и других номеров, если они выбраны в качестве ключевых полей и не используются в математических вычислениях.

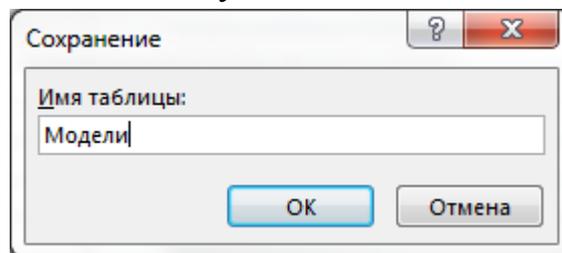


Рис. 3.24. Окно *Сохранение* для ввода имени таблицы

При сохранении новой таблицы (командой *Сохранить* на *Панели быстрого доступа* или на вкладке *ФАЙЛ*, а также можно клавишами *Shift + F12*) в окне нужно задать имя таблицы – *Модели* (рис. 3.24) и нажать *ОК*. При этом в базу данных *Автомобили* помещается созданная таблица, а в области навигации в списке объектов *Таблицы* появляется надпись *Модели*. Далее откройте таблицу *Модели* в режиме таблицы и, используя данные Табл. ПЗ приложения 1, заполните таблицу *Модели*.

Аналогично, используя спроектированные параметры (см. Приложение 1, Табл. П4 и Табл. П5), следует создать структуру таблиц *Клиенты* и *Купля автомобилей*.

Создание подстановки (поля со списком) на основе таблицы или запроса

Для создания подстановки, источником значений для которой служит таблица или запрос, следует выполнить следующие действия.

- В режиме конструктора выбрать поле, для которого нужно создать список значений.

В нашем примере в таблице *Купля автомобилей* для поля *Номер клиента* нужно создать список, берущим свои значения из таблицы *Клиенты*.

- Щёлкнуть кнопкой мыши в ячейке с названием типа данных поля.
- Из выпадающего списка типов (см. рис. 3.17) выбрать вариант *Мастер подстановок*.
- Открывается окно (см. рис. 3.20) *Создание подстановки* (шаг 1), в котором надо установить переключатель в положение *Объект “поле подстановки” получит значения из другой таблицы или другого запроса* и нажать кнопку *Далее*.
- В появившемся окне (рис. 3.25) на втором шаге работы мастера выбрать источник, на основе которого будет создана подстановка. Если источником должна быть таблица, установите переключатель *Показать* в положение *Таблицы* и щёлкните кнопкой мыши на названии нужной таблицы. Если источником должен быть запрос, то установите переключатель в положение *Запросы* и щёлкните кнопкой мыши на названии нужного запроса.

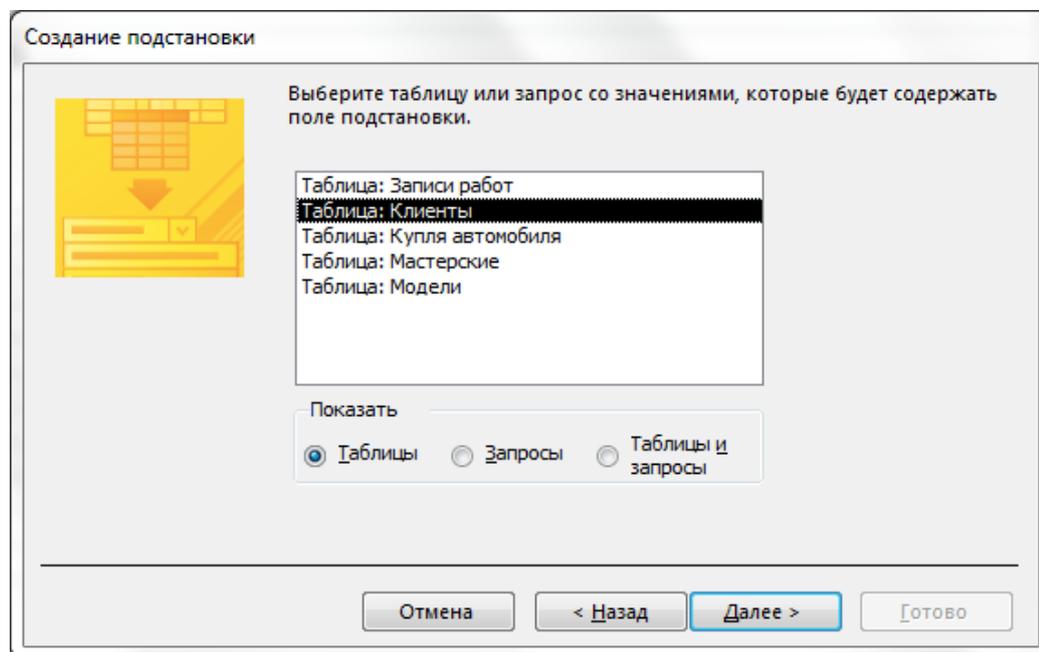


Рис. 3.25. Диалоговое окно «Создание подстановки»
(шаг 2 – выбор источника)

В нашем примере выбираем таблицу *Клиенты*, на значениях которой будет строиться список поля. Нажмите кнопку *Далее*.

- На третьем шаге *Мастер подстановок* (рис. 3.26) выберите в списке *Доступные поля* те поля таблицы или запроса, значения которых необходимо отображать в поле подстановки.

В нашем примере выбираем поля, содержимое которых, должно быть видимо при заполнении таблицы *Купля автомобилей* (ключевое поле *Номер клиента* должно быть выбрано обязательно, остальные поля – на Ваше усмотрение, например, выберем ещё поле *Фамилия*). Чтобы добавить поле в подстановку, щёлкните кнопкой мыши на названии поля в списке *Доступные поля* и нажмите кнопку со стрелкой вправо.

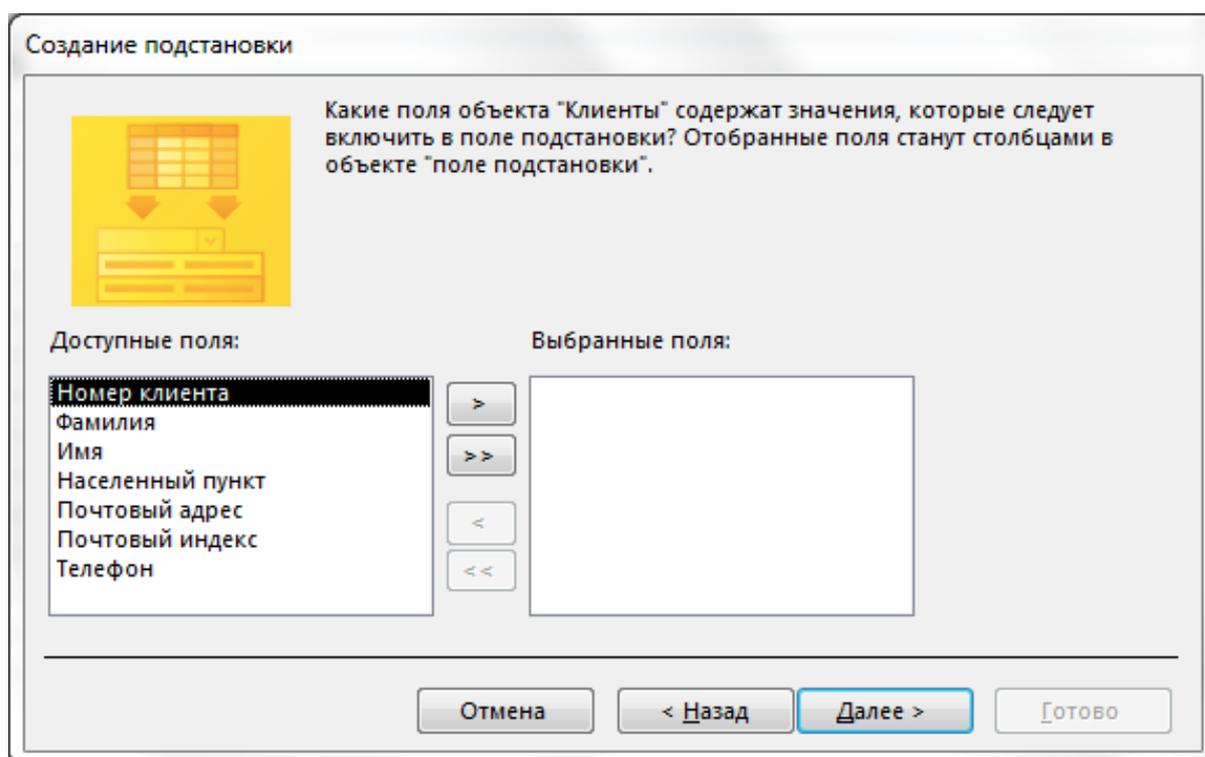


Рис. 3.26. Диалоговое окно «Создание подстановки»
(шаг 3 – выбор полей для подстановки)

- Выбранные поля отображаются в правом списке диалогового окна «Создание подстановки». После того как набор полей будет сформирован, нажмите кнопку *Далее*.

Отметим, что если в главной таблице есть первичный ключ, но он не был добавлен в подстановку, то Access добавит его автоматически после нажатия кнопки *Далее*.

- В следующих окнах мастера выберите порядок сортировки элементов списка (рис. 3.27) и нужную ширину столбца списка (рис. 3.28). Там же есть возможность «скрыть ключевой столбец», чтобы не отображался в списке подстановки.

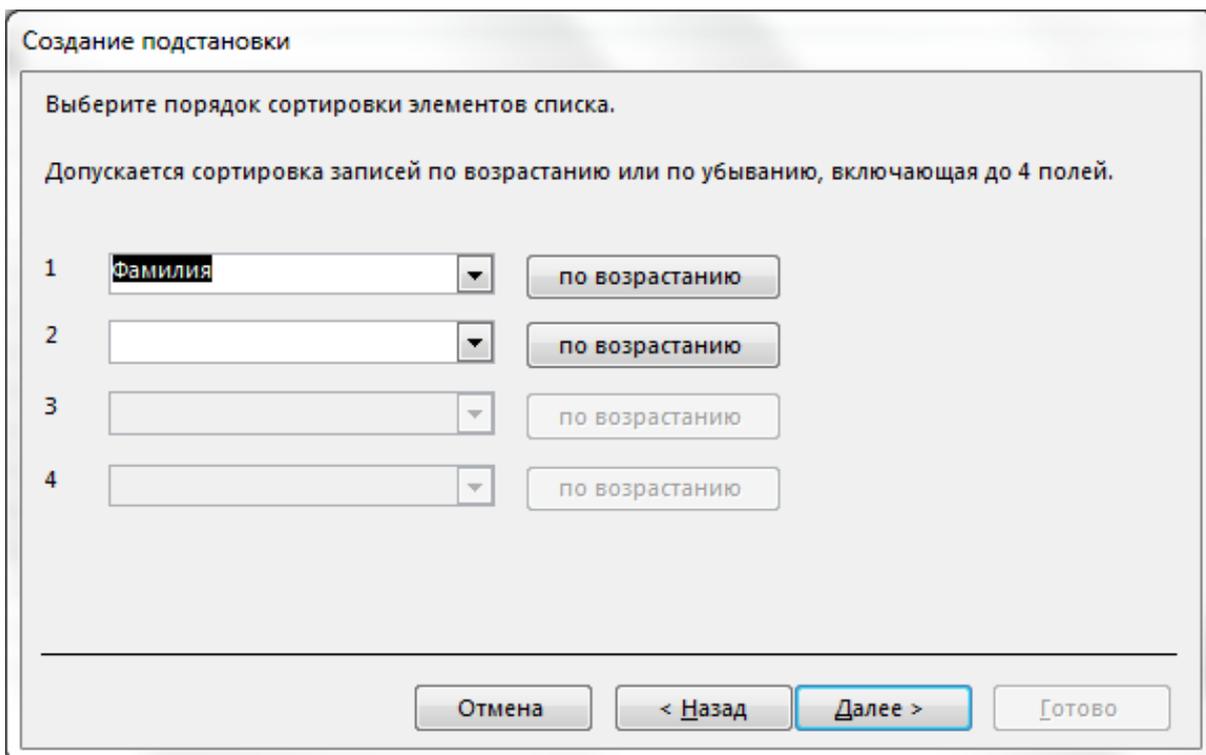


Рис. 3.27. Диалоговое окно «Создание подстановки»
(шаг 4 – выбор режима сортировки)

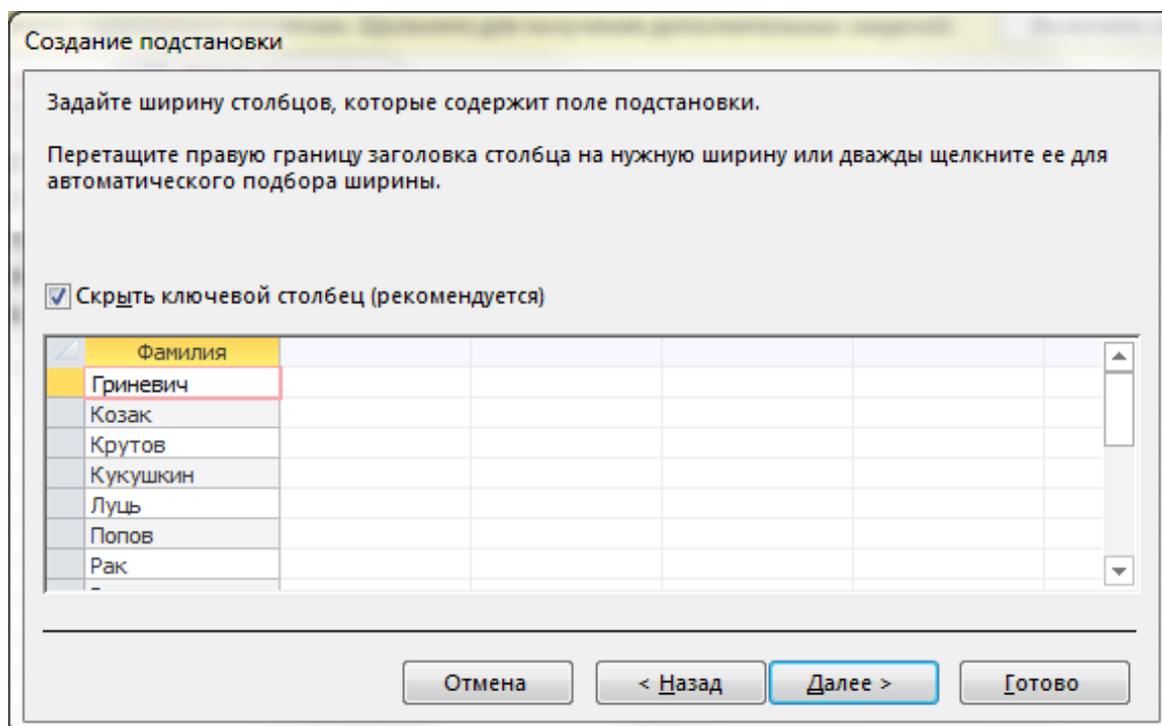


Рис. 3.28. Диалоговое окно «Создание подстановки»
(шаг 5 – настройка отображения столбцов в подстановке)

В нашем примере выбираем сортировку по алфавиту (по возрастанию) по полю *Фамилия* и устанавливаем флажок «Скрыть ключевой столбец», поскольку поле *Номер клиента* при заполнении таблицы *Купля автомобилей* нам не нужно.

- Далее мастер предлагает задать подпись поля со списком и параметры поддержания целостности данных связанных таблиц (рис. 3.29). После нажатия кнопки *Готово* создание мастером нового поля со списком завершается.

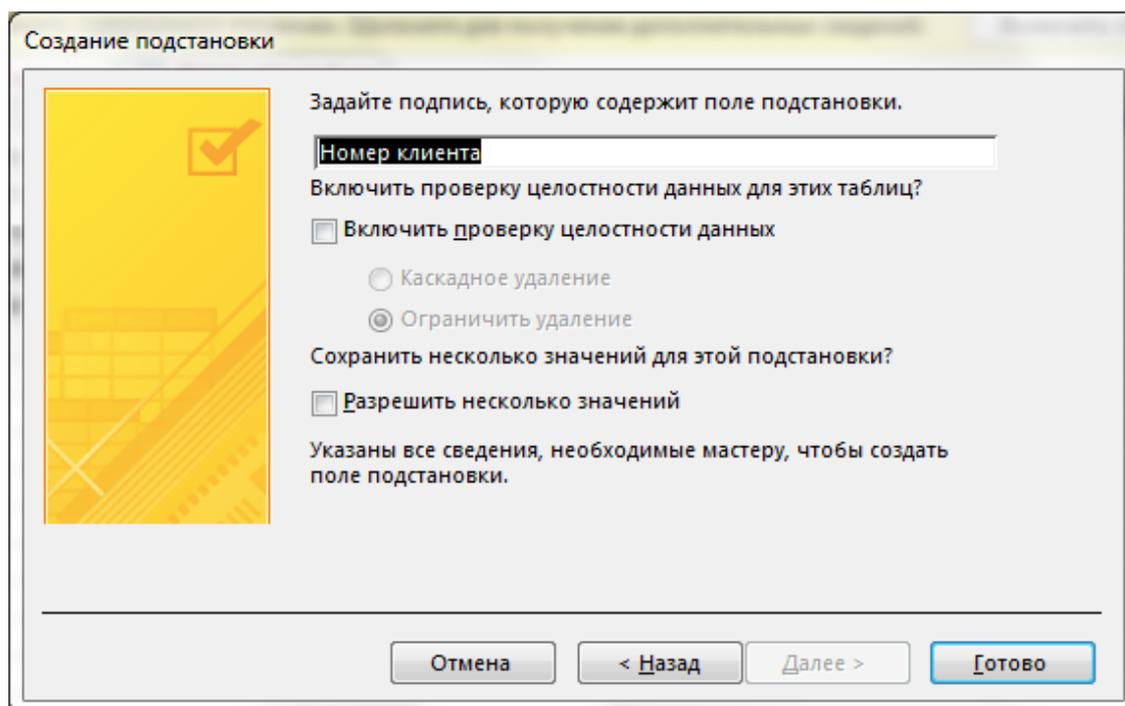


Рис. 3.29. Диалоговое окно «Создание подстановки»
(шаг 6 – завершение создания подстановки)

В нашем примере можно оставить подпись для поля, которую предлагает Access по умолчанию, а параметры поддержания целостности данных будем устанавливать позже.

Чтобы протестировать созданное списочное поле, нужно перейти в режим таблицы (на вкладке *Главная* кнопка *Режим*), щёлкнуть кнопкой мыши в любой ячейке столбца, для которого создавалась подстановка, и нажать кнопку в правой части ячейки для открытия списка значений.

Выполнив аналогичную последовательность действий, организуйте в таблице *Купля автомобилей* поле *Код модели* в виде списка, берущего свои значения из двух первых полей таблицы *Модели*, при этом ключевое поле *Код модели* в подстановке не скрывать.

После определения структуры таблиц можно приступить к формированию записей таблиц, т.е. заполнению их данными. На вкладке ленты *Главная*

(см. рис. 3.9) находятся команды, предназначенные для работы с записями, для оформления внешнего вида таблицы, а также для форматирования текста.

В БД «Автомобили» в режиме таблицы введите в таблицы *Клиенты* и *Купля автомобилей* по десять произвольных записей (некоторые клиенты должны заказать больше одного автомобиля). Создайте предварительно в таблице *Купля автомобилей* маску ввода типа: 23.10.17. (процедура создания маски ввода рассматривается ниже). Заполненные данными таблицы сохраните.

Создание маски ввода

Маски ввода позволяют предотвратить ввод заведомо некорректных данных, задав определённые условия для вводимых символов в значения определённых полей. Маска отображается в поле, когда вводится значение, тем самым она выполняет и функцию подсказки. Следует заметить, что в отличие от формата, который влияет на отображение значений, хранящихся в базе данных, маска влияет только на процесс ввода данных.

Создать маску можно вручную, введя для соответствующего поля значение свойства *Маска ввода*. Кроме того, можно использовать *мастер* масок, чтобы создать маску для ввода пароля или маску для ввода даты/времени.

Рассмотрим создание маски с помощью мастера. Чтобы создать маску ввода с помощью мастера необходимо выполнить следующие действия. Щёлкните на названии свойства *Маска ввода*, а затем на кнопке (три точки) справа от значения свойства.

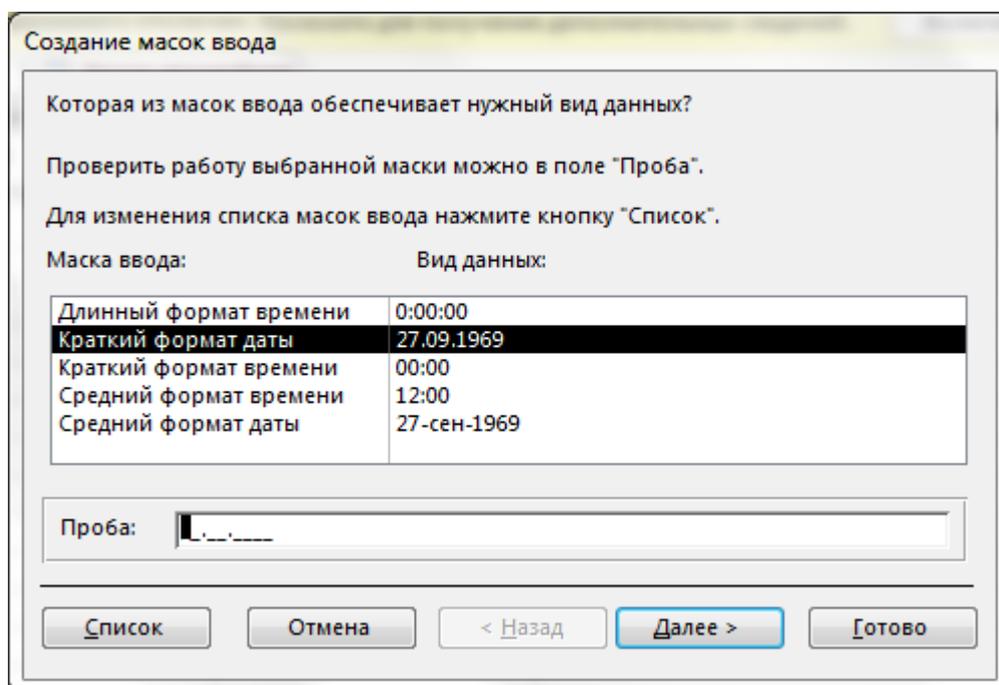


Рис. 3.30. Диалоговое окно мастера масок ввода (выбор маски ввода)

Появится диалоговое окно *Создание масок ввода* (рис. 3.30), в котором можно выбрать нужную маску (даты, времени или пароля). Маска пароля позволяет создать поле для ввода пароля (вводимые данные в такое поле являются скрытыми, а вместо реальных символов отображаются звёздочки).

Тестирование ввода значения с помощью выбранной маски можно вводом в поле *Проба* пример значения поля. Если выбрана маска даты/времени, которую не требуется редактировать можно нажать кнопку *Готово*. Если нужно изменить выбранную маску даты или времени, следует нажать кнопку *Далее*.

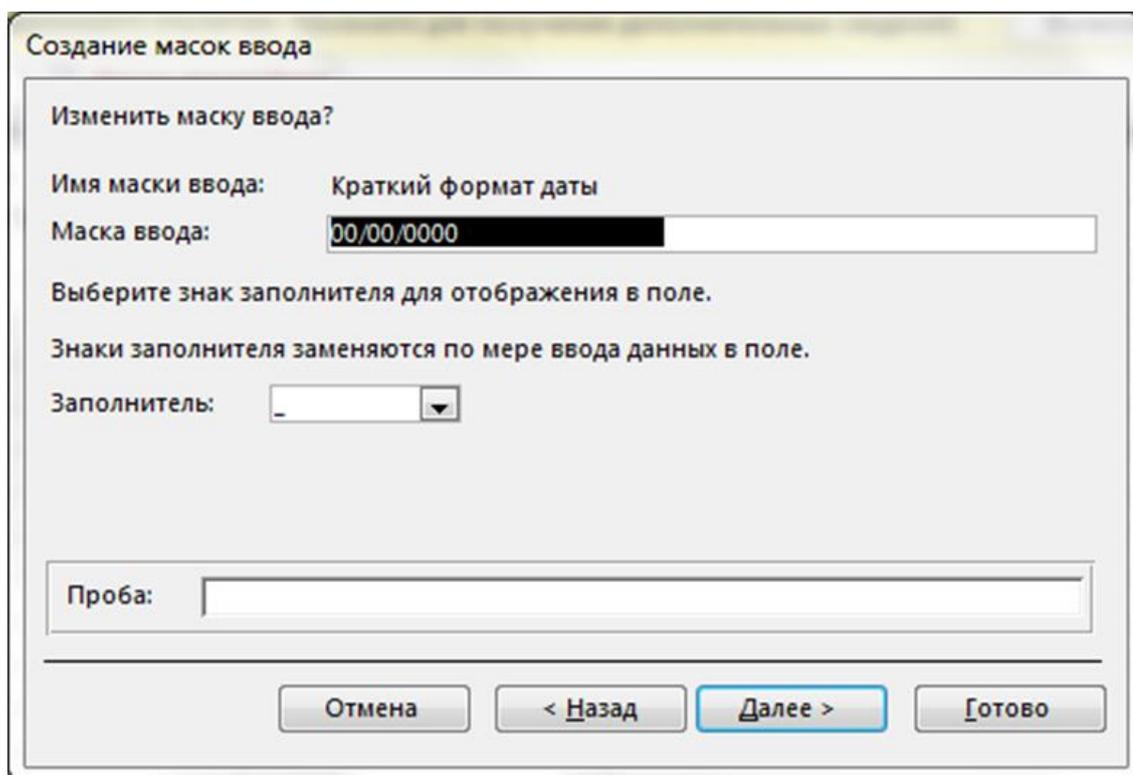


Рис. 3.31. Диалоговое окно мастера масок ввода (изменение маски ввода)

В появившемся диалоговом окне (рис. 3.31) можно изменить вид маски ввода. Дополнительно можно выбрать знак заполнителя (вместо символа подчёркивания, используемого по умолчанию), который будет отображаться в поле таблицы при вводе значения. Также можно протестировать ввод значений в поле *Проба*. Для завершения создания маски следует нажать кнопку *Далее*, а затем *Готово*.

Создание таблицы в режиме таблицы

Для начинающего пользователя создание таблицы в *режиме таблицы* значительно проще. В процессе создания новой БД в ней автоматически создаётся новая пустая таблица с именем *Таблица1*, которая открывается в области документов в режиме таблицы. На рис. 3.32 показана таблица с откры-

тым списком поля *Щелкните для добавления*, обеспечивающим выбор типа данных для поля таблицы.

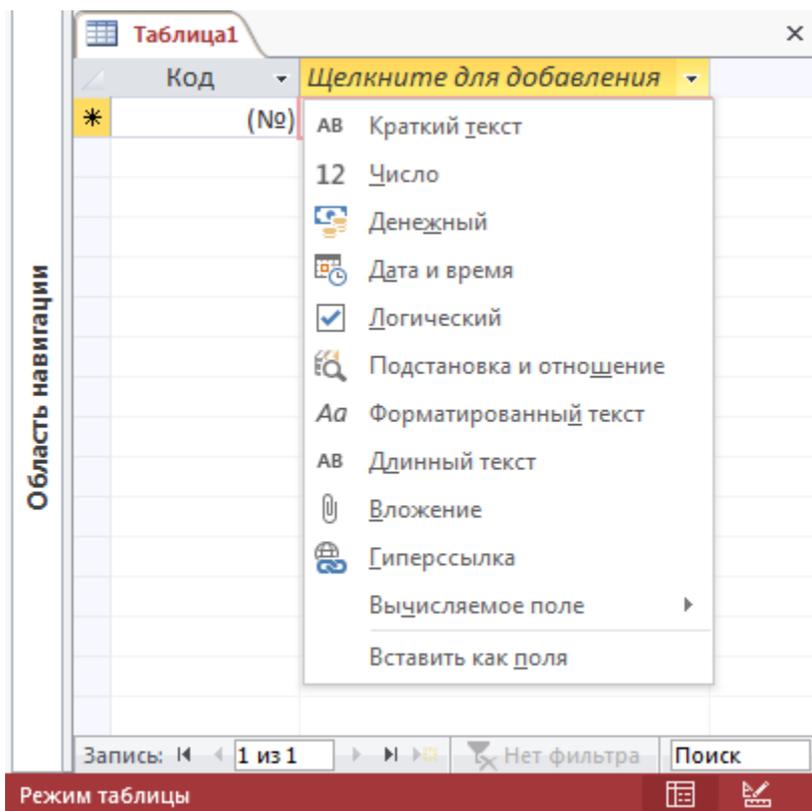


Рис. 3.32. Таблица с открытым списком поля *Щелкните для добавления*

В создаваемой таблице устанавливается ключевое поле с типом данных *Счетчик*, а далее в режиме таблицы можно добавлять новые поля с требуемыми характеристиками.

При создании таблицы в существующей БД на вкладке *Создание* следует нажать кнопку *Таблица*. Будет добавлена новая таблица, которая откроется в режиме таблицы. При этом открывается вкладка ленты *Работа с таблицами / Поля* (рис. 3.33). Здесь можно определить поля таблицы с различными свойствами.

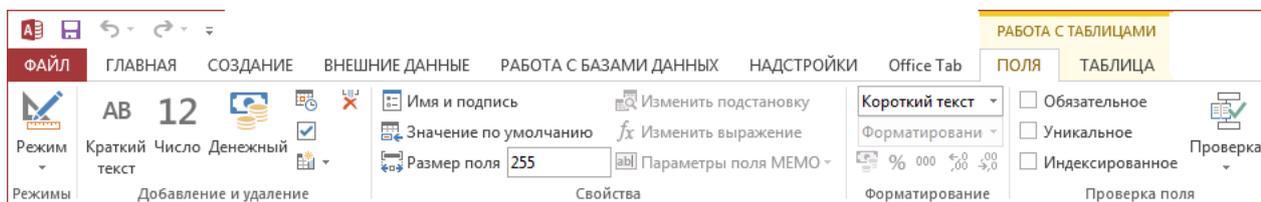


Рис. 3.33. Вкладка ленты *Работа с таблицами / Поля*

Для однозначного определения каждого поля в таблице обязательно ему задаётся имя и тип данных, который должен соответствовать сохраняе-

мым в этом поле данным. Кроме того, в режиме таблицы в Access 2013 каждому полю можно задать большинство свойств. Определение полей таблицы выполняется различными путями: выбором необходимых команд на вкладке *Работа с таблицами / Поля*; используя список столбца *Щёлкните для добавления*; или непосредственным вводом данных в столбец *Щёлкните для добавления*.

На рис. 3.34 приведён открытый расширенный список *Другие поля* вкладки *Работа с таблицами / Поля*, с помощью которого можно добавлять и удалять поля таблицы с различными типами данных. При открытии данного списка в поле предоставляется возможность изменить его элементы. Таким образом, можно сразу приступить к разработке таблицы БД в режиме таблицы.

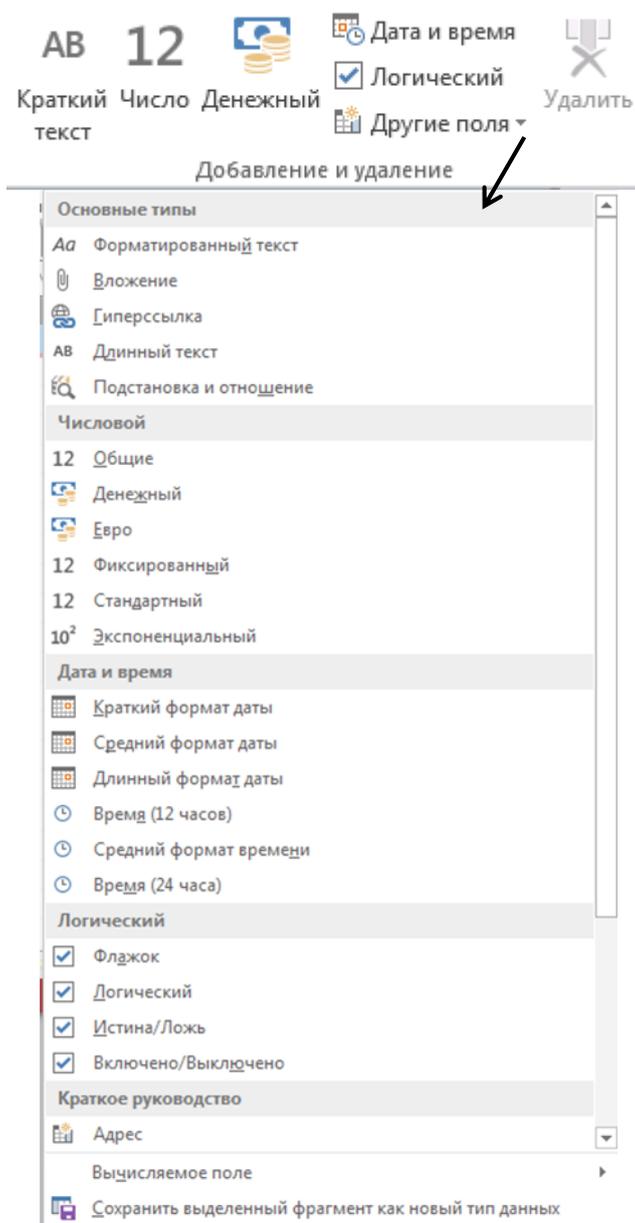


Рис. 3.34. Открытый список *Другие поля* группы *Добавление и удаление* вкладки *Работа с таблицами / Поля*

При добавлении полей Access автоматически назначает им имена *Поле1*, *Поле2* и т. д. Для изменения имени поля можно выполнить двойной щелчок на заголовке поля и ввести новое имя или в контекстном меню выбрать команду *Переименовать поле*.

При вводе первого значения в столбец *Щёлкните для добавления* автоматически определяется тип данных (например, текст, числовой, дата и время или денежный) и добавляется новое поле в таблицу. Если на основании введённых данных невозможно точно определить тип данных, Access задаёт тип данных *Короткий текст*. В зависимости от вида вводимых данных автоматически задаются и свойства поля.

Создав структуру таблицы, описанным выше способом, можно доработать её с помощью команд в группах *Свойства*, *Форматирование* и *Проверка поля* вкладки *Работа с таблицами / Поля* (см. рис. 3.33).

Пример. Создание таблицы *Мастерские* в режиме таблицы.

Структура таблицы *Мастерские* была определена в главе 2 (см. рис. 2.22). При создании её в режиме таблицы в новой пустой таблице по умолчанию определяется ключевое поле с именем *Код* и типом данных *Счетчик*. В режиме таблицы нельзя удалить ключ и нельзя определить новый ключ. Для выполнения таких операций необходимо использовать режим конструктора. В нашем примере при создании таблицы *Мастерские* не может быть использован предлагаемый системой ключ. Таблица *Мастерские* содержит уникальное поле *№ мастерской*, поэтому предлагаемое имя ключевого поля нужно заменить. Для этого можно выполнить двойной щелчок на заголовке поля и исправить имя *Код* на *№ мастерской*. Тип данных, сохраняемых в поле *№ мастерской* необходимо изменить на вкладке ленты *Работа с таблицами / Поля*, в группе *Форматирование*. Для этого нужно открыть список с типами данных (рис. 3.35.) и выбрать *Числовой*.

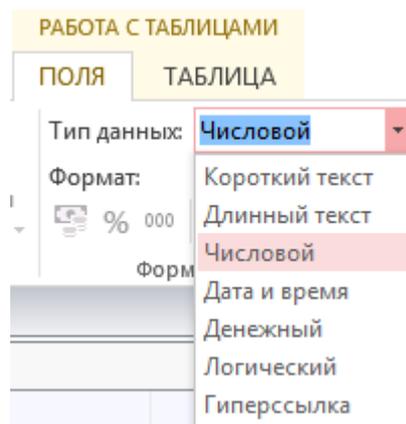


Рис. 3.35. Открытый список *Тип данных* вкладки *Работа с таблицами / Поля*

Добавление следующего поля в таблицу *Мастерские*, например, *Адрес*, можно осуществить следующим образом. В ячейку под заголовком *Щёлкните для добавления* (см. рис. 3.32) можно ввести любое текстовое значение. При вводе такого значения Access для этого поля автоматически распознаёт и назначает тип данных *Короткий текст* с размером поля равным 255. Для нашей таблицы *Мастерские* такой размер поля *Адрес* чрезмерно большой. Изменить размер поля можно в группе *Свойства* (рис. 3.33) на вкладке *Работа с таблицами / Поля*. Здесь же (кнопка *Имя и подпись*) можно изменить имя поля, указать, если необходимо, его *подпись*, отображаемая в заголовке и *описание* поля (рис. 3.36). Если свойство *Подпись* поля не определено, в заголовке столбца таблицы будет отображаться его имя. Можно также указать с помощью *построителя выражений* (см. рис. 3.19) значение по умолчанию, которое будет сохраняться в поле, если пользователь не ввёл никакого значения.

Для добавления следующего поля можно щёлкнуть в столбце *Щёлкните для добавления* и самостоятельно выбрать тип данных нового поля (см. рис. 3.32), а затем изменить имя и задать свойства поля.

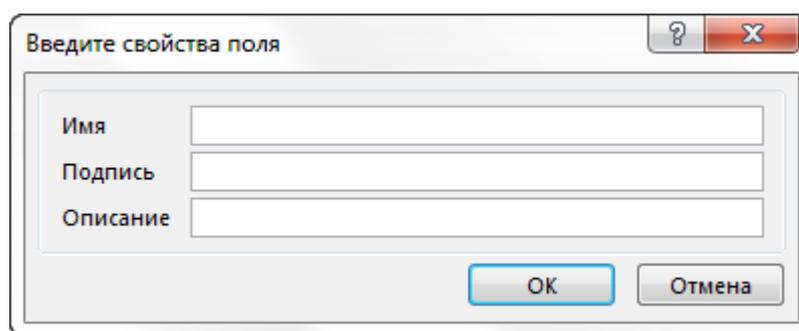


Рис. 3.36. Окно кнопки *Имя и подпись*.

Для добавления в таблицу нового поля, скопированного из другой таблицы нужно подвести курсор к нужному значению поля таблицы, из которой будет выполняться копирование и, при появлении на нем знака плюс, выделить и скопировать его (копируются конкретное значение поля, его имя и все свойства). Затем следует щёлкнуть в столбце *Щёлкните для добавления* (см. рис. 3.32) и выбрать команду *Вставить как поля* для добавления поля в создаваемую таблицу.

Самым простым способом добавления полей в таблицу является использование команд группы *Добавление и удаление* вкладки *Работа с таблицами / Поля* (см. рис. 3.33), в которой отдельными кнопками представлены основные типы данных и в раскрывающемся списке *Другие поля* представлены недостающие типы данных и форматы отображения поля.

Для открытия в режиме таблицы закрытой таблицы нужно в области навигации выбрать и дважды щёлкнуть имя данной таблицы или выполнить в контекстном меню команду *Открыть*.

В режиме таблицы, удобно задавать следующие свойства: *Правило проверки поля*, *Сообщение об ошибке*, а также создавать выражение, ограничивающего значения, которые могут быть введены в поле. Для этого, когда отображены значения в полях таблицы, следует выбрать поле и щёлкнуть на вкладке *Работа с таблицами / Поля* в группе *Проверка поля* кнопку *Проверка* (рис. 3.37).

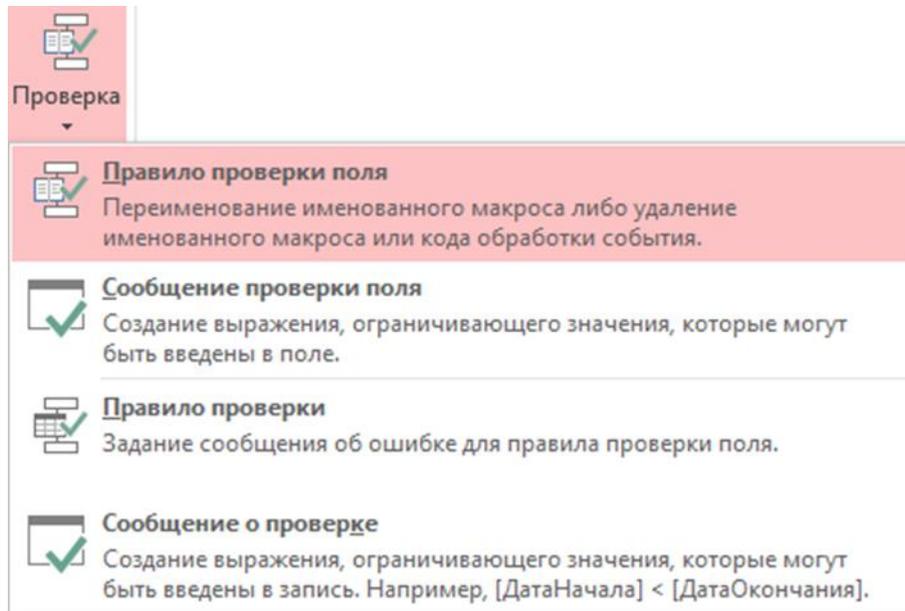


Рис. 3.37. Список кнопки *Проверка* на вкладке *Работа с таблицами / Поля*

Если в открывшемся списке (см. рис. 3.37) выбрать *Правило проверки поля*, откроется *Построитель выражений* (см. рис. 3.19), в котором можно ввести выражение для проверки вводимых в поле значений. Это выражение сохраняется в свойстве таблицы *Правило проверки*. Выбрав *Сообщение проверки поля*, можно ввести текст, который будет выводиться при отклонении вводимых значений от допустимых (т. е. сообщение об ошибке). Команды списка кнопки *Проверка* позволяют ещё задавать выражение для сравнительной проверки значений в полях записи и подготовить сообщение при нарушении условий, заданных выражением.

После добавления всех необходимых полей таблицы можно перейти в режим в режим *Конструктора*, просмотреть и уточнить, если необходимо, ключевое поле, тип данных и свойства полей, предложенные по умолчанию, организовать поля с необходимыми списками.

Например, в БД «*Автомобили*» в таблице «*Мастерские*», внесите следующие описания полей:

- для поля «Средняя цена» – «*Средняя цена капитального ремонта (грн)*»;
- для поля «Наценка» – «*Наценка за срочность*»).

Поле *№ Мастерской* сделайте ключевым.

Поле *Категория* организуйте в виде фиксированного списка (*первая, вторая, высшая*).

Установите тип данных в поле *Средняя цена* – денежный; в поле *Наценка* – числовой, размер поля – Одинарное с плавающей точкой, формат поля – процентный.

После определения структуры таблицы можно приступить к формированию записей таблицы, т.е. заполнению её данными (см. Приложение 1, табл. П6).

Далее дополните таблицу *Модели*. Для этого откройте её в режиме *Конструктора*. Создайте и заполните дополнительное поле *Мастерская*, оформив его в виде поля со списком, берущим свои значения из трёх первых полей таблицы *Мастерские*.

Создание таблицы путём импорта

При работе не только с одним приложением для хранения и обработки данных целесообразно использование импорта и экспорта данных. Эти операции позволяют обмениваться данными между разными приложениями, что значительно повышает эффективность работы.

Создать таблицу можно путём импорта данных, хранящихся в другом месте (на листе Microsoft Excel, в списке Windows SharePoint Services, файле XML, другой базе данных Access, Oracle или Microsoft SQL Server, папке Microsoft Outlook, текстовый файл и т. д.), или создания связи с ними.

При импорте данных создаётся 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.

Можно связать базы данных. Под связыванием нескольких баз данных Access понимают связь с данными другой базы без применения импорта. Когда осуществляется связывание с данными, в текущей базе данных создаётся 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 (добавление, удаление полей, изменение типа данных поля) нельзя.

Однако, следует иметь в виду, что нельзя изменять данные на листе Excel с помощью связанной таблицы. В качестве обходного способа можно импортировать исходные данные в базу данных Access, а затем создать связь с базой данных из Excel.

Чаще всего импортируют данные из одной базы Access в другую, создавая в конечной базе копию данных или объекта, не изменяя исходный материал. Все это осуществляется автоматически. Во время импорта имеются следующие возможности: выбирать данные и объекты для копирования; управлять параметрами импорта таблиц и запросов; управлять импортированием запросов и связей между таблицами.

Импорт данных из книги Excel

В качестве примера можно создать и оформить таблицу *Записи работ*.

Откройте базу данных *Автомобили* для импорта в неё данных из книги Excel.

Для импорта данных и связей из книги Microsoft Excel в базу данных Access 2013 нужно выполнить следующее.

1. Открыть БД для импорта в неё данных из книги Excel.
2. Выбрать вкладку *Внешние данные* (рис.3.38).

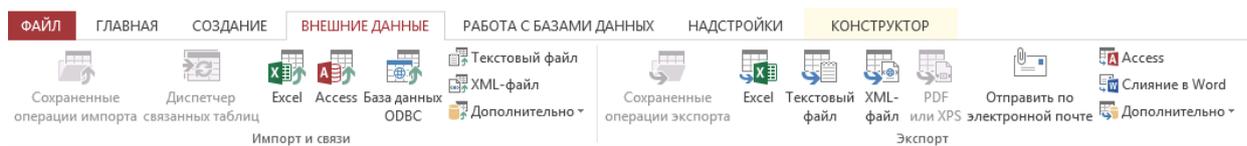


Рис. 3.38. Вкладка *Внешние данные*

3. На вкладке показаны 2 группы: *Импорт и связи* и *Экспорт*. Если нажать указателем мыши на кнопку *Excel* в группе *Импорт и связи* появится окно мастера импорта *Внешние данные – Электронная таблица Excel* (рис.3.39). Здесь надо указать в текстовом поле *Имя файла* название файла Excel и его путь или можно воспользоваться кнопкой *Обзор*, чтобы найти необходимый файл Excel с помощью проводника. Выбрав нужный файл, нужно нажать кнопку *Открыть*. Далее необходимо указать, когда и где сохранять данные в текущей базе.
4. Если установить переключатель в положение *Создать связанную таблицу для связи с источником данных* будет осуществлено связывание с выбранной книгой Excel.
5. Для импорта данных из книги Excel в существующую таблицу нужно выбрать положение переключателя *Добавить копию записей в конец таблицы*.

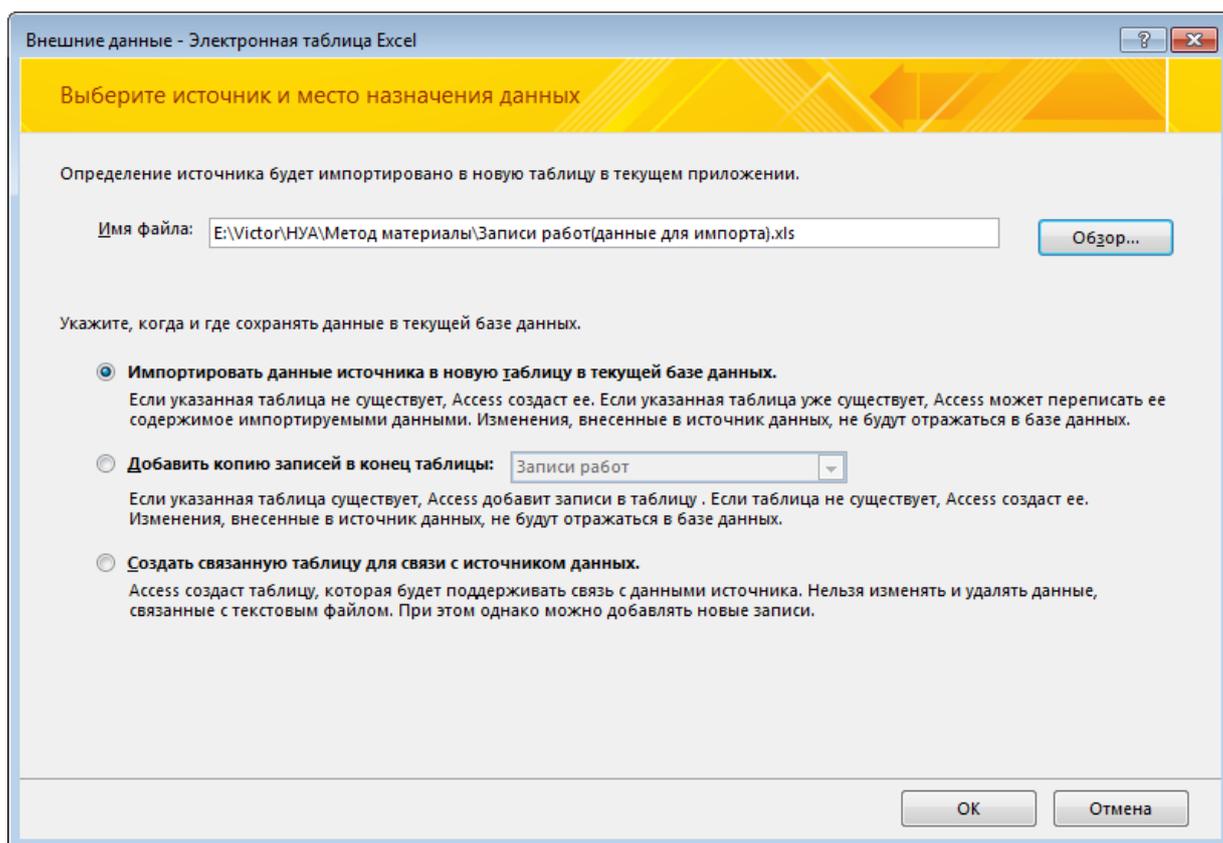


Рис. 3.39. Окно *Внешние данные* – Электронная таблица Excel

6. Для создания новой таблицы с импортированными данными из книги Excel необходимо установить переключатель в положение *Импортировать данные источника в новую таблицу в текущей базе данных* и нажать кнопку *OK*. В данном случае на экране появится окно мастера импорта (рис. 3.40), где предлагается выбрать лист из импортируемой книги Excel. Для выбора листа нужно щёлкнуть на его названии кнопкой мыши.
7. В поле *Образцы данных для листа* показаны данные, которые можно импортировать только с выбранного листа рабочей книги. Для импорта других листов операцию следует повторить.
8. Если надо импортировать данные из именованного диапазона, следует установить переключатель в положение «*именованные диапазоны*» и выбрать название нужного диапазона.
9. Для перехода на следующий шаг мастера импорта необходимо нажать кнопку *Далее*.

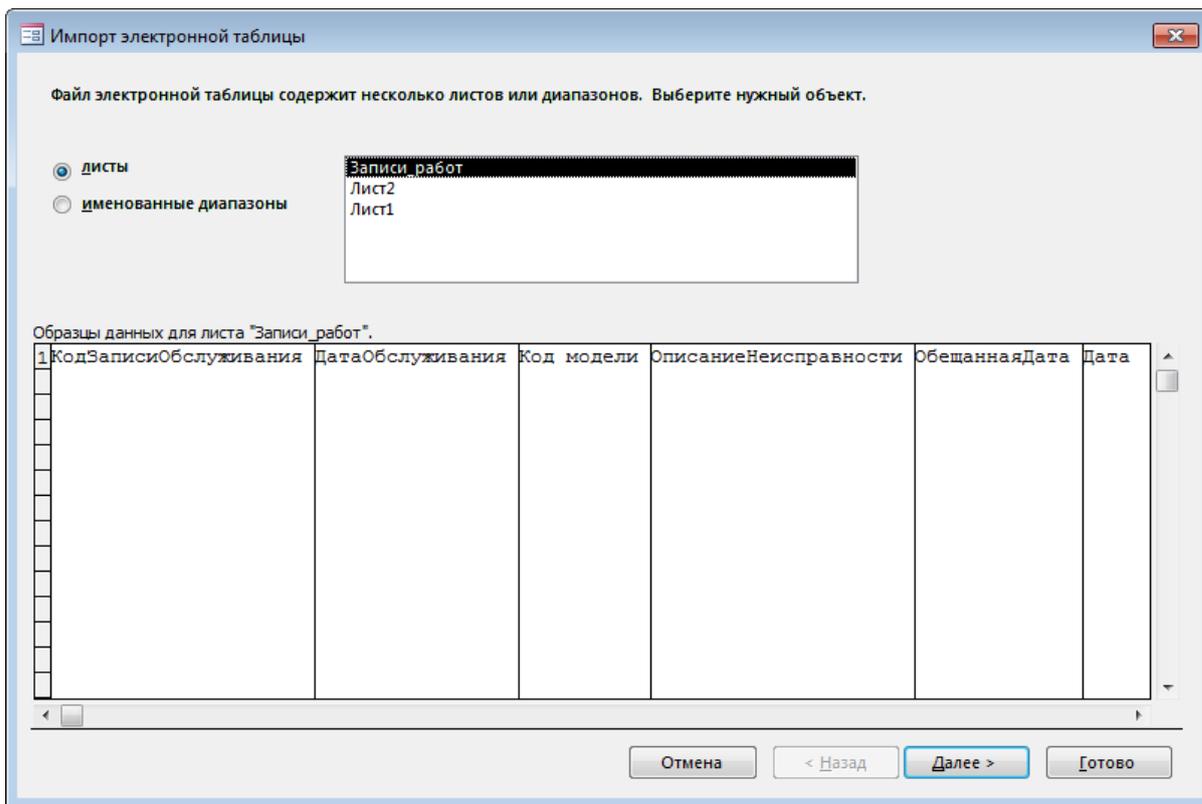


Рис. 3.40. Окно выбора листов для импорта

10. На данном шаге (рис. 3.41), чтобы использовать заголовки из первой строки таблицы Excel (если они есть) в качестве имён полей таблицы, следует установить флажок «Первая строка содержит заголовки столбцов» и нажать кнопку *Далее*.

В группе *Описание поля* можно настроить параметры каждого столбца, последовательно выбирая столбец в таблице, расположенной в нижней части окна мастера:

- *имя поля* – будет введено уже или следует ввести название выбранного столбца, если на предыдущем шаге не был установлен флажок «*Первая строка содержит заголовки столбцов*»;
- *тип данных* – выбрать тип данных для выделенного столбца, раскрыв список;
- *индекс* – выбрать следующие параметры индексации поля: *Нет* – когда поле не индексируется; *Да (Допускаются совпадения)* – когда поле индексируется с допуском на совпадения; *Да (Совпадения не допускаются)* – когда поле индексируется без допуска на совпадения;
- *не импортировать (пропустить) поле* – установить флажок для пропуска выделенного поля.

11. Следует повторить операцию для всех полей, которые необходимо импортировать. Если не надо импортировать какое-либо поле в базу данных устанавливаются флажок *не импортировать (пропустить) поле*.

Пример

В таблице *Записи работ* для поля *Код записи обслуживания* установите: *Да (Совпадения не допускаются)*.

Для поля *Код модели* установите *индексированное поле, допускаются совпадения*.

Для поля *закрито* выберите тип данных *логический*.

Остальные поля настройте самостоятельно.

12. При нажатии кнопки *Далее* в следующем окне мастера импорта (рис. 3.43) предлагается выбрать варианты задания ключа в новой таблице:

- *Автоматически создать ключ* – ключ таблицы, создаётся автоматически.
- *Определить ключ* – ключом будет поле таблицы указанное пользователем, то есть выделенное в данный момент. Для выбора другого поля следует открыть список и выбрать нужный вариант.
- *Не создавать ключ* – ключевое поле не будет создаваться в импортируемой таблице.

Для таблицы *Записи работ* установите переключатель в положение *определить ключ* и укажите поле *Код записи обслуживания* в качестве ключевого, воспользовавшись раскрывающимся списком.

13. Нажмите кнопку *Далее*.

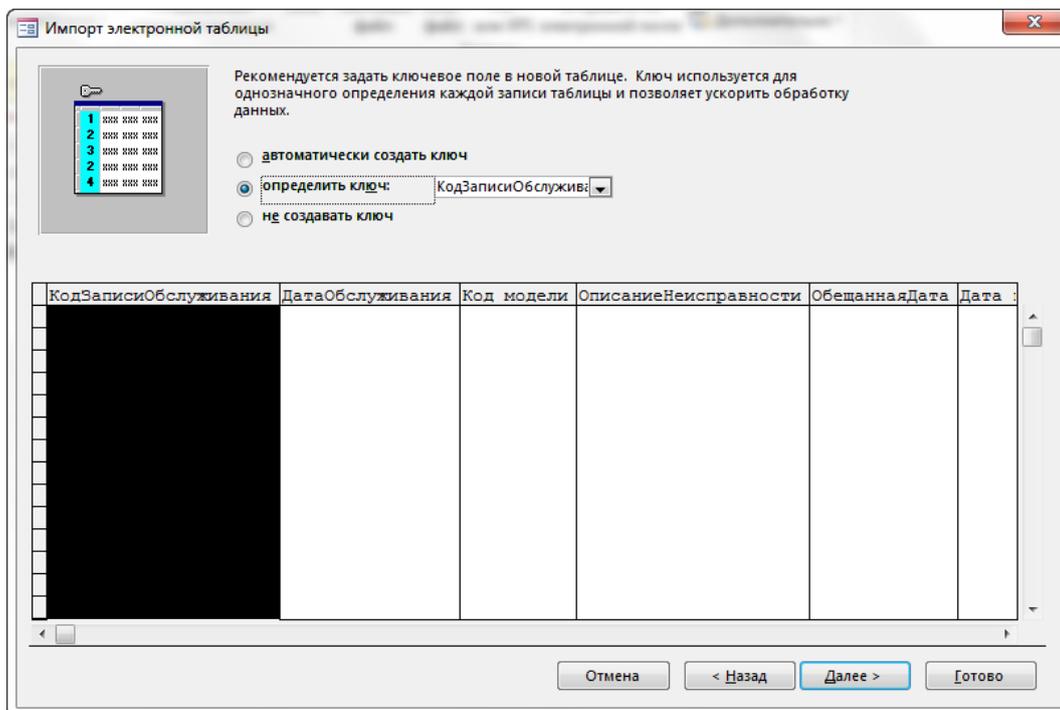


Рис. 3.43. Окно мастера импорта, позволяющее указать ключ

14. На экране появится следующий шаг мастера (рис. 3.44), где необходимо указать имя таблицы для импорта. По умолчанию используется имя листа электронной таблицы.

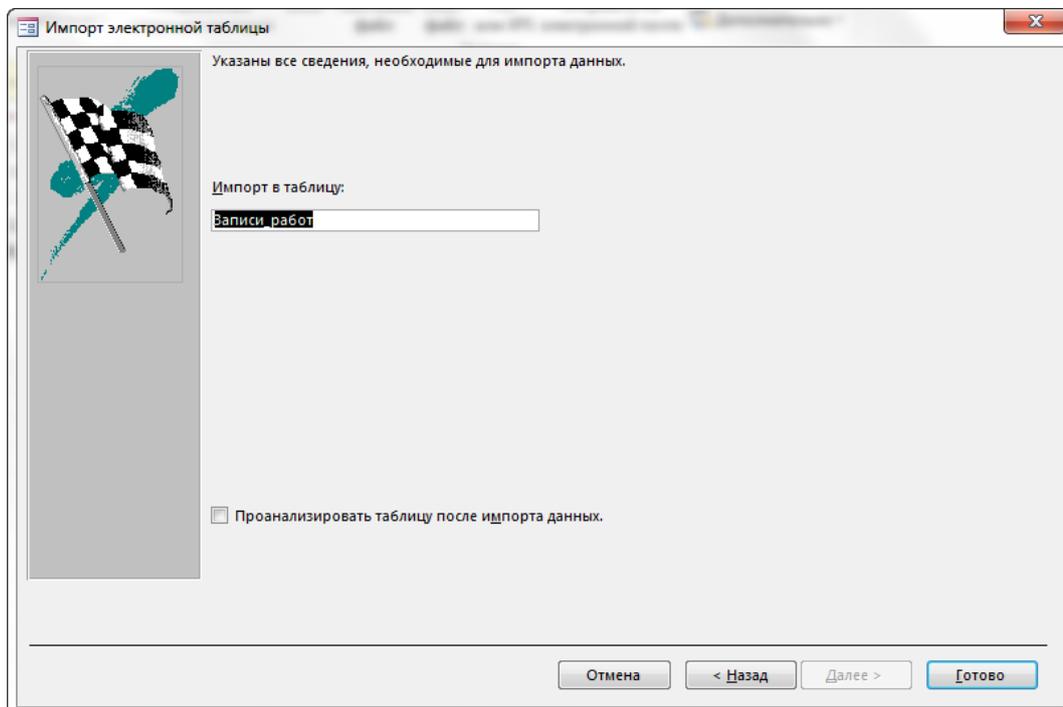


Рис. 3.44. Окно мастера импорта, позволяющее указать имя таблицы и проанализировать её

Можно также выполнить анализ импортируемой таблицы после установки флажка «*Проанализировать таблицу после импорта данных*»

15. Нужно нажать кнопку *Готово* для импортирования данных из книги Excel в базу данных Access. После процесса импорта появится последний шаг мастера импорта, в окне которого отобразятся сообщения о результатах импорта и сообщения о имеющихся ошибках. Если в дальнейшем нужно повторить операцию без использования мастера импорта, следует установить флажок *Сохранить шаги импорта*.
16. Для завершения работы мастера импорта необходимо нажать кнопку *Заккрыть*.

После завершения работы мастера можно в режиме конструктора добавить в импортированную таблицу описание тех полей, для которых из названия однозначно не вытекает предназначение поля, а также можно внести необходимые изменения в свойства полей.

Оформление внешнего вида таблиц

Для изменения внешнего вида таблиц следует воспользоваться командами на вкладке *Главная* в группе *Форматирование текста* (рис. 3.45).

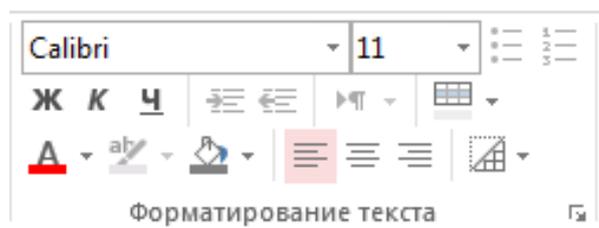


Рис. 3.45. Команды для форматирования таблиц

Здесь можно изменить для таблицы: размер и гарнитуру шрифта, задать его начертание (полужирный шрифт, курсив) и подчеркивание, а также тип выравнивания значения поля в определённом столбце (по левому краю, по центру или по правому краю ячеек). Кроме того можно выбрать цвет шрифта, цвет фона для таблицы; задать отображение или скрытие линий сетки; выбрать дополнительный цвет фона для визуального разделения соседних (чередующихся) строк таблицы.

Нажав в группе *Форматирование текста* (рис. 3.45) внизу справа кнопку-стрелку можно открыть диалоговое окно *Формат таблицы* (рис. 3.46). Здесь задаются дополнительные параметры: включение/выключение эффекта «утопленных» и «приподнятых» ячеек; задание цвета и формы линий сетки; выбор обратного порядка столбцов.

При оформлении внешнего вида таблиц можно также изменить размер ячеек таблицы. Для изменения высоты строк таблицы следует подвести указатель мыши к нижней границе любой строки и перетащить границу вверх или вниз на требуемый уровень. Можно также выбрать в контекстном меню начала строк команду *Высота строки* и в открывшемся окне задать необходимые параметры. Изменится высота всех строк таблицы, поскольку в Access они имеют одинаковую высоту. Аналогично изменяется ширина отдельного столбца: нужно перетащить с помощью кнопки мыши правую границу столбца или в контекстном меню имени столбца выбрать команду *Ширина поля* и в открывшемся окне задать необходимые параметры. При этом, следует заметить, что ширина остальных столбцов не меняется.

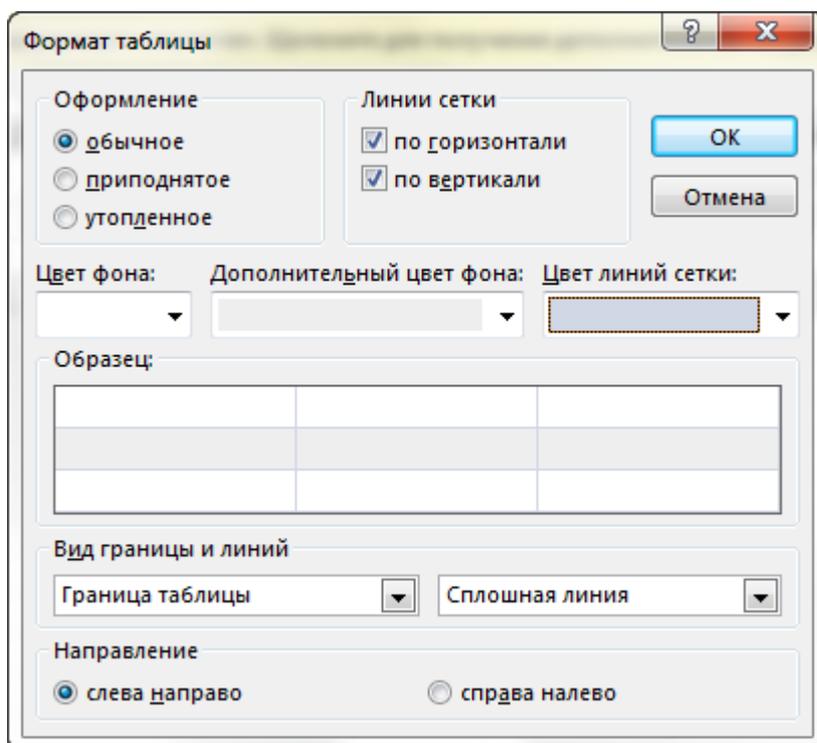


Рис. 3.46. Диалоговое окно *Формат таблицы*

После внесения всех требуемых изменений необходимо сохранить макет таблицы.

Просматривая или заполняя таблицы данными иногда удобно скрыть часть столбцов, с возможностью затем восстановить отображение ранее скрытых столбцов. Для этого надо выделить столбец или диапазон столбцов, удерживая нажатой клавишу клавиатуры *Shift*, и контекстном меню заголовка любого столбца выбрать пункт *Скрыть поля*. Можно выбрать пункт *Отобразить поля* и в появившемся окне *Отображение столбцов* снять или установить флажки возле названий тех столбцов, которые необходимо соответственно скрыть или отобразить. Затем можно закрыть окно.

Когда много полей (столбцов) в таблице, и не все они видны на экране монитора, для просмотра или внесения данных в крайне правых столбцах удобно закрепить в левой стороне окна определённые столбцы, и затем прокрутить экран вправо к необходимым столбцам. Для этого следует выделить необходимые столбцы и в контекстном меню выбрать команду *Закрепить поля*. Выделенные столбцы переместятся к левому краю таблицы и закрепятся там, т.е. при прокрутке экрана вправо данные столбы останутся видны на экране. Для отмены закрепления столбцов надо в контекстном меню выбрать команду *Отменить закрепление всех полей* и далее, выделив закреплённые ранее столбцы, перетащить их на исходное место в таблице

Пример

В БД «Автомобили» форматируйте таблицу *Модели* следующим образом:

- шрифт – *Courier New, 12 пт, цвет синий, курсив*;
- цвет фона выделенных элементов (кнопка цвет фона) – *Синий, Акцент 1, более светлый оттенок 60 %*;
- цвет строки (дополнительный цвет фона)– *Золотистый, Акцент 4, более светлый оттенок 80 %*;
- оформление – *утопленное*;
- цвет линий сетки – *красный*;
- ширина столбца – *по ширине данных* (выделите все имена полей и в контекстном меню выберите команду *Ширина поля*, далее в открывшемся окне нажмите кнопку *По ширине данных*);
- высота строки – *25 пт*;

Скройте любые 4 столбца таблицы *Модели*, затем восстановите их.

Закрепите поле (столбец) *Модель* и просмотрите прокруткой экрана вправо остальные поля.

Освободите столбец *Модель* и переместите его на прежнее место.

Сохраните все изменения, выполненные в таблице *Модели*.

Для изменения параметров внешнего вида всех таблиц (как новых, так и ранее созданных) можно отредактировать настройки по умолчанию. Нажав в левом верхнем углу ленты на кнопку *Файл*, а затем кнопку *Параметры*, и в открывшемся окне *Параметры* (рис. 3.47) Access кнопку *Таблица*, можно внести необходимые изменения в настройки по умолчанию и закончить процедуру, нажав кнопку *ОК*.

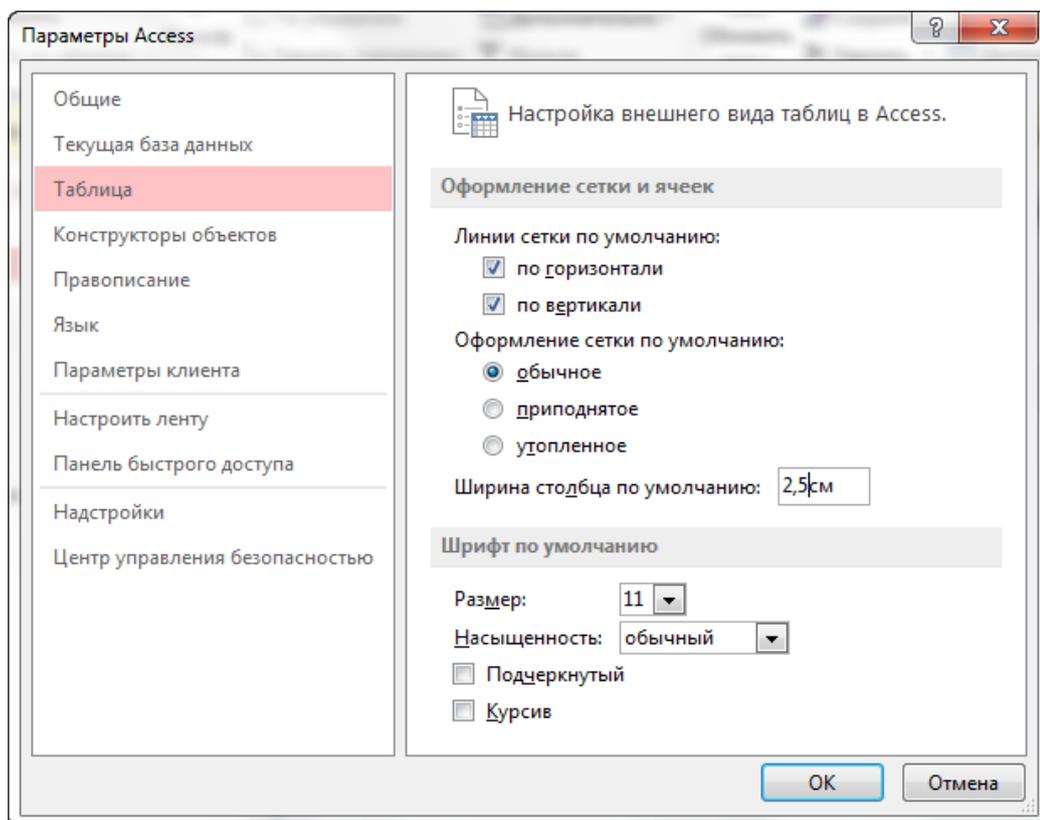


Рис. 3.47. Диалоговое окно *Параметры Access*

Создание схемы данных

После создания таблиц, содержащих данные, относящиеся к различным аспектам БД, нужно создать схему данных, позволяющую объединить эти данные при их извлечении из БД. Реляционная БД состоит из нормализованных таблиц, связанных отношениями, в которой обеспечивается однократный ввод, отсутствие дублирования данных и поддержание целостности данных. Связи между таблицами позволяют выполнить объединение данных различных таблиц, необходимое для решения большинства задач ввода, просмотра и корректировки данных, получения информации по запросам, формам и создания отчётов, а также обеспечить поддержание целостности взаимосвязанных данных в таблицах.

Связи между таблицами устанавливаются в соответствии с предварительно разработанной структурой базы данных (см. рис. 2.2) и запоминаются эти связи в схеме данных Access. Таким образом, схема данных используется системой в процессе обработки данных и является средством графического отображения логической структуры БД. Связи, указанные в схеме данных, используются системой обработки автоматически.

Пример. Рассмотрим процесс создания схемы данных, соответствующей логической структуре БД «Автомобили», разработка которой рассмотрена в главе 2.

Чтобы увидеть все связи в базе данных на вкладке *Работа с базами данных* в группе *Отношения* следует нажать кнопку *Схема данных* (рис. 3.48), после чего открывается окно *Схема данных*.

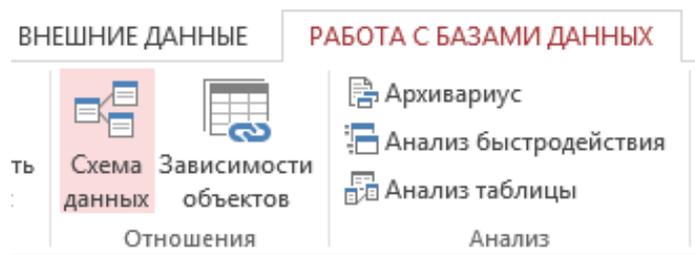


Рис. 3.48. Кнопка *Схема данных* на вкладке *Работа с базами данных*

Для добавления таблиц в схему данных на вкладке *Работа со связями | Конструктор* в группе *Связи* следует нажать кнопку *Отобразить таблицу* (рис. 3.49).

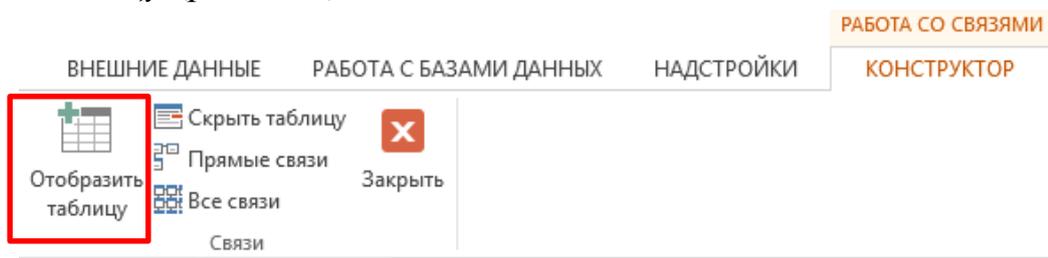


Рис. 3.49. Кнопка *Отобразить таблицу* на вкладке *Работа со связями | Конструктор*

Выбор таблиц, включаемых в схему данных, осуществляется в открываемом диалоговом окне *Добавление таблицы* (рис. 3.50).

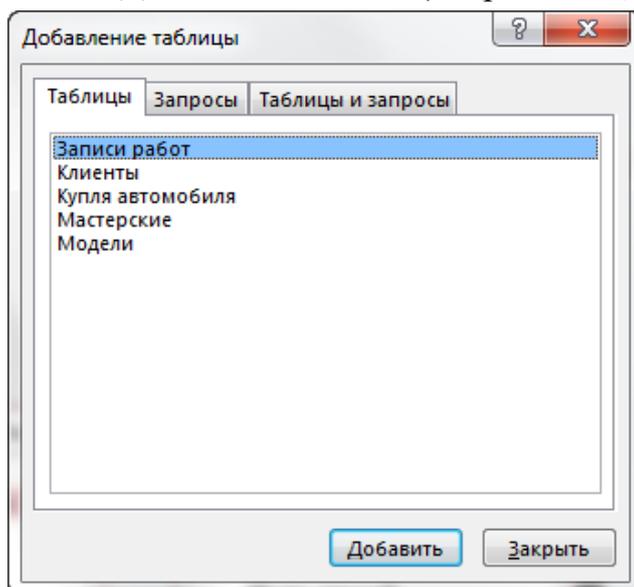


Рис. 3.50. Диалоговое окно *Добавление таблицы*

В окне *Добавление таблицы* отображены все таблицы и запросы, содержащиеся в базе данных. Выбрав вкладку *Таблицы* и выделяя в окне имя таблиц, с помощью кнопки *Добавить* можно разместить в окне *Схема данных* все созданные таблицы БД «Автомобили». Закрыв диалоговое окно *Добавление таблицы*, в окне *Схема данных* представляются таблицы БД со списками своих полей и выделенными ключами (рис. 3.51).

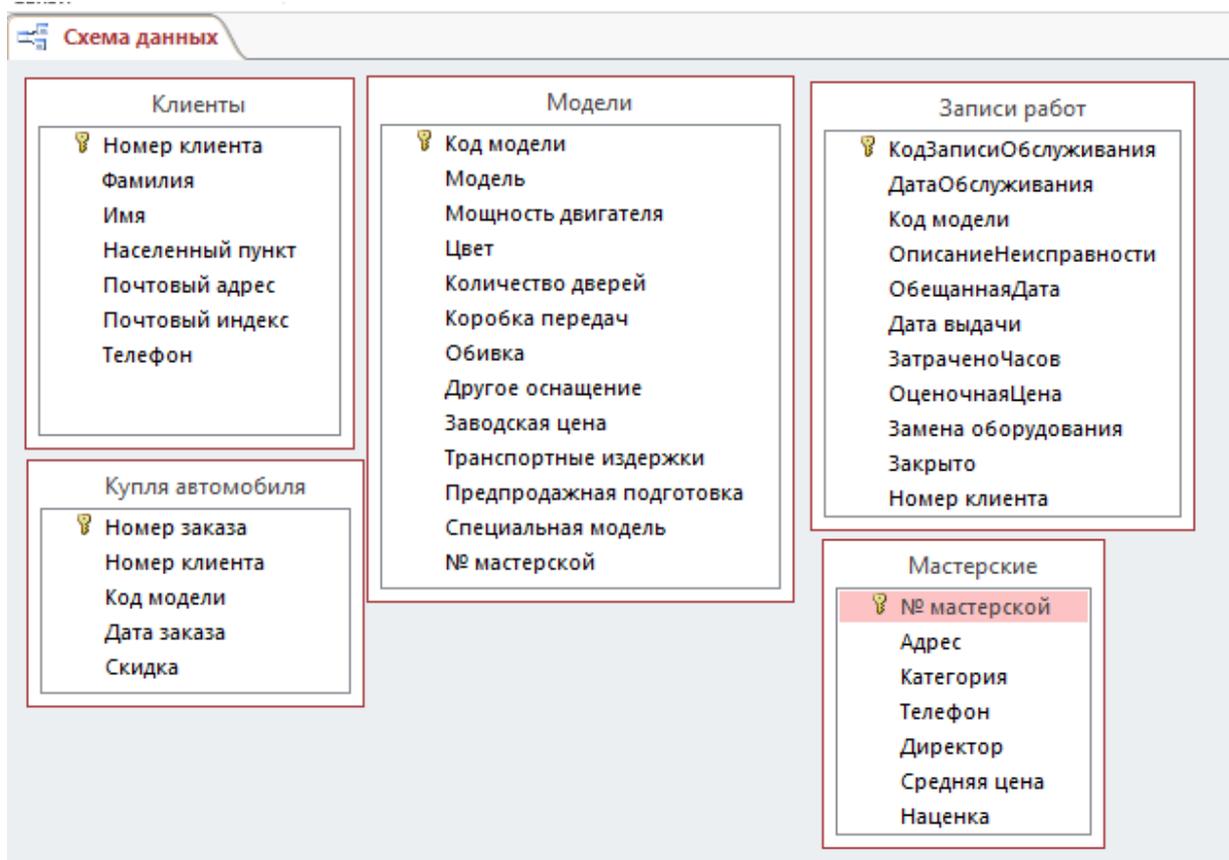


Рис. 3.51. Диалоговое окно *Схема данных БД «Автомобили»* (без созданных связей между таблицами)

Создание связей между таблицами БД

Используя логическую структуру БД, нужно создать в схеме данных связи между таблицами. Как было отмечено ранее, связи по типу бывают «*один-ко-многим*», «*многие-ко-многим*» и «*один-к-одному*».

Связь с отношением «**один-ко-многим**» является наиболее часто используемым типом связи между таблицами. В такой связи каждой записи в определённой таблице могут соответствовать несколько записей в другой таблице, а запись в этой таблице не может иметь более одной соответствующей ей записи в первой таблице. Таблицы связываются по общим полям, называемым ключом связи. При отношении между таблицами типа «один-ко-многим» ключом связи является ключ главной таблицы (простой или состав-

ной) – *первичный* ключ. В подчинённой таблице, представляющей в отношении сторону «многие», этот ключ связи называется *внешним* ключом, он может не входить в состав ключа этой второй таблицы. Ещё раз отметим, что первичный ключ главной таблицы и внешний ключ подчинённой таблицы должны иметь одинаковый тип данных. Отметим ещё раз, что такой тип связи «один-ко-многим» является основным в реляционных БД.

При отношении «**многие-ко-многим**» одной записи в таблице могут соответствовать несколько записей в другой таблице, а одной записи в этой второй таблице могут соответствовать несколько записей в первой таблице. Такая схема реализуется только с помощью третьей (связующей) таблицы, ключ которой состоит из по крайней мере двух полей, которые являются полями внешнего ключа в первой и второй таблицах.

При отношении «**один-к-одному**» запись в одной таблице может иметь не более одной связанной записи в другой таблице и наоборот. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением «один-к-одному» используют для разделения очень широких таблиц, для отделения части таблицы по соображениям защиты и др.

Примечание. Связанные поля не обязательно должны иметь одинаковые имена, но они должны иметь одинаковые типы данных и иметь содержимое одного типа. Кроме того, связываемые поля числового типа должны иметь одинаковые значения свойства *Размер поля*. Существуют следующие два исключения из правила: поле счётчика можно связывать с числовым полем, если в последнем в свойстве *Размер поля* задано значение «*Длинное целое*»; а также поле счётчика можно связать с числовым, если для обоих полей в свойстве *Размер поля* задано значение «*Код репликации*».

Пример

Внимание! Закройте все открытые таблицы. Создавать или изменять связи между открытыми таблицами нельзя.

В БД «*Автомобили*» таблицы «*Мастерские*» и «*Модели*», а также «*Модели*» и «*Записи работ*» находятся в отношении «один-ко-многим». Например, со стороны таблицы «*Мастерские*» тип связи – «один», а со стороны таблицы «*Модели*» – «многие», т.е. любая мастерская может обслуживать несколько моделей автомобилей, а конкретная модель должна обслуживаться только в одной конкретной мастерской. Для установки связи между таблицами необходимо выделить в главной таблице, например, «*Мастерские*» ключевое поле «*Номер мастерской*», далее при нажатой левой кнопке мыши надо перетащить его в соответствующее внешнее поле «*Номер мастерской*» подчинённой таблицы «*Модели*». Поскольку поле «*Номер мастерской*» является уникальным ключом в главной таблице, а в подчинённой таблице не является ключевым, Access определяет отношение «один-ко-многим» между

данными таблицами. Тип отношения «один-ко-многим» отображается в нижней строке диалогового окна «Изменение связей» (рис. 3.52).

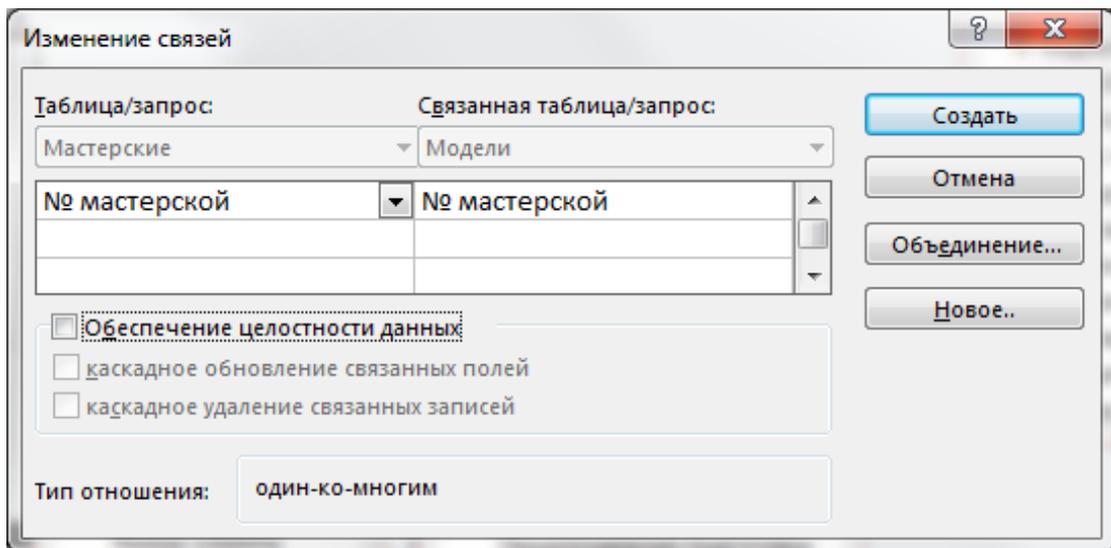
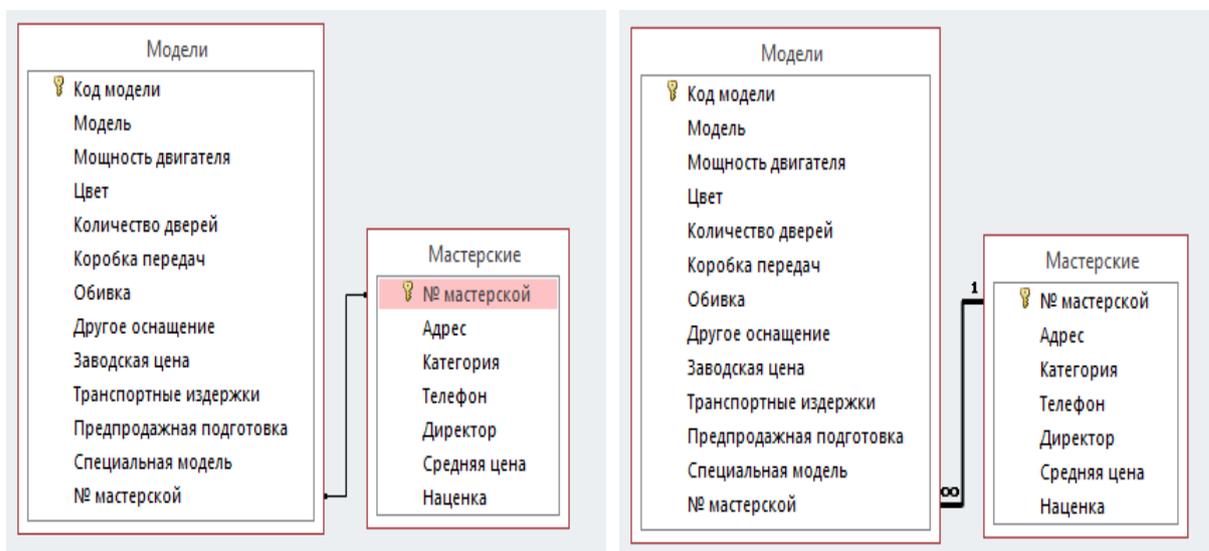


Рис. 3.52. Диалоговое окно «Изменение связей»

Теперь если флажок «Обеспечение целостности данных» не установлен, то после нажатия кнопки «Создать» на схеме данных появится тонкая соединительная линия между ключевым и внешнем полями таблиц и не будет показано с какой стороны тип связи – «один», а с какой стороны – «мно-го» (рис. 3.53 а).



а)

б)

Рис. 3.53. Связи таблиц «Мастерские» и «Модели»

а) – флажок «Обеспечение целостности данных» не установлен;

б) – флажок «Обеспечение целостности данных» установлен.

Если же включено обеспечение целостности данных, то будет толстая соединительная линия (рис. 3.53 б), означающая, что данные будут синхронизироваться (об этом подробнее будет ниже рассмотрено). Кроме того, на изображении появляются цифра **1** и символ ∞ . Цифра **1** означает, что в таблице «*Мастерские*» может быть только одна связанная запись. Значок ∞ означает, что в нескольких записях таблицы «*Модели*» может быть указан одинаковый номер мастерской, поскольку мастерская может обслуживать несколько моделей автомобилей.

Заметим, что при создании полей со списком мастер подстановки автоматически формирует связь между таблицей, в которой создаётся поле со списком, и таблицей, являющейся источником значений списка. Эта связь (тонкая соединительная линия между ключевым и внешним полями таблиц) отображается сразу в *Схеме данных*, как например, на рис. 3.53 а.

Аналогично устанавливается связь между главной таблицей «*Модели*» и подчинённой таблицей «*Записи работ*» с помощью ключа связи «*Код модели*».

Следует отметить, что нажав кнопку «*Новое*» в диалоговом окне «*Изменение связей*» (рис. 3.52), можно создать связь между двумя любыми таблицами базы в появляющемся окне «*Создание*» (см. рис. 3.54), не выходя в окно схемы данных.

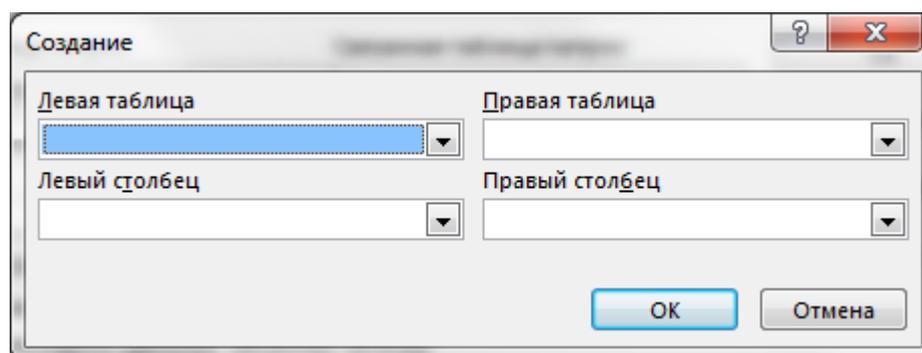


Рис. 3.54. Диалоговое окно «Создание»

Рассмотрим теперь связи между остальными таблицами БД «Автомобили». Таблицы «*Клиенты*» и «*Модели*» связаны отношением «многие-ко-многим». А связующей между ними является третья таблица – «*Купля автомобилей*», ключами, которой являются «*Номер клиента*» и «*Код модели*», являющимися внешними ключами соответственно таблиц «*Клиенты*» и «*Модели*». При этом, тип связи между таблицами «*Номер клиента*» (главная таблица) и «*Купля автомобилей*» (подчинённая таблица), а также между таблицами «*Модели*» (главная таблица) и «*Купля автомобилей*» (подчинённая таблица) – «один-ко-многим». Способ создания такой связи был рассмотрен выше. Создайте связи между всеми таблицами БД. Установите везде флажки

(см. рис. 3.52) *Обеспечение целостности данных, каскадное обновление связанных полей и каскадное удаление связанных записей.*

В результате создания связей между таблицами схема данных БД «Автомобили» приобретёт вид представленный на рис. 3.55.

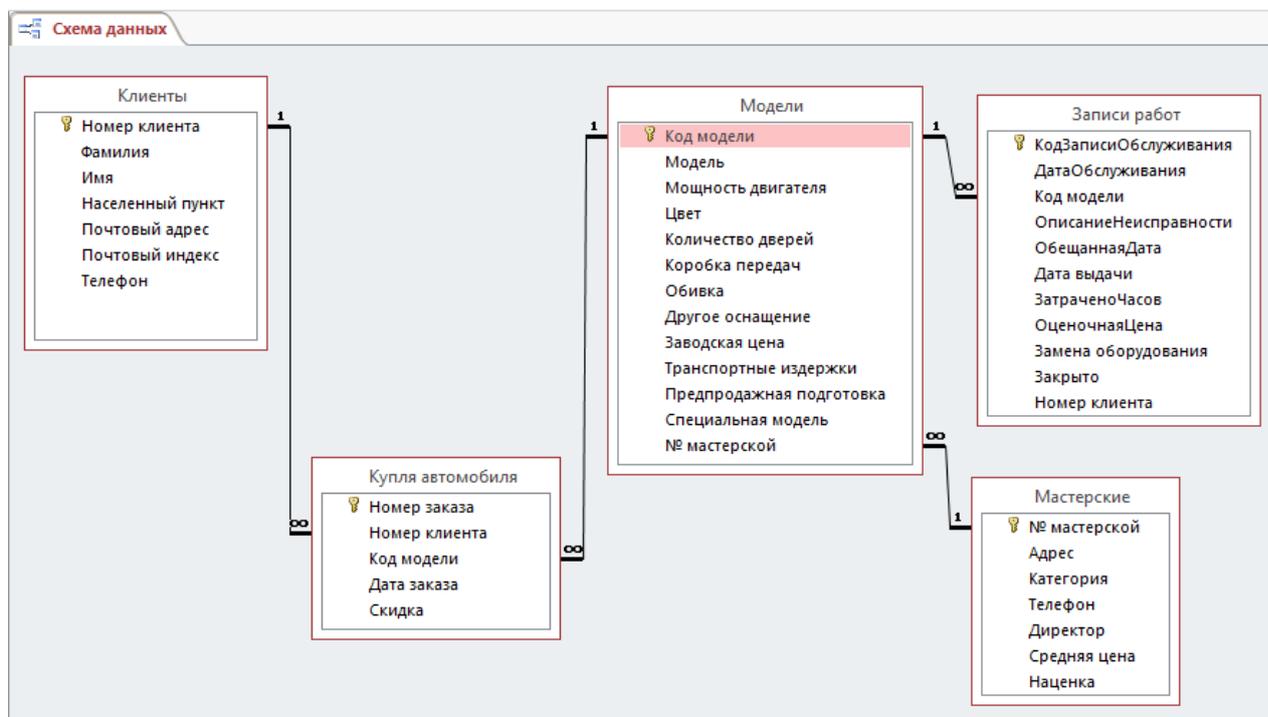


Рис. 3.55. Схема данных БД «Автомобили» со всеми созданными связями между таблицами

Обеспечение целостности данных

Целостность данных означает систему правил, используемых для поддержания связей между записями в связанных таблицах, а также обеспечивающих защиту от случайного удаления или изменения связанных данных. Так в подчинённую таблицу не может быть добавлена запись с несуществующим в записях главной таблицы значением ключа связи. Если не удалены в подчинённой таблице записи, связанные с главной таблицей, в последней удалить запись нельзя. Если подчинённая таблица имеет записи связанные с главной таблицей, в которых ключ связи не меняется, то невозможно в записи главной таблицы изменить значения ключа связи. Установить параметр целостности для связи таблиц невозможно, если ранее введённые в таблицы данные не отвечают требованиям целостности.

Каскадное обновление и удаление связанных записей – это средства поддержания целостности данных в связанных таблицах, обеспечивающие обновление и удаление всех связанных записей в подчинённой таблице или таблицах при изменении записи в главной таблице. Для того чтобы преодолеть ограничения на удаление или изменение связанных записей, сохраняя

при этом целостность данных, следует включить режимы *каскадного обновления* и *каскадного удаления* (рис. 3.52). При установленном флажке *каскадное обновление связанных полей* изменение значения в ключевом поле главной таблицы приводит к автоматическому обновлению соответствующих значений во всех связанных записях в подчинённой таблице. При установленном флажке *каскадное удаление связанных полей* удаление записи в главной таблице приводит к автоматическому удалению связанных записей в подчинённой таблице.

Заметим, что установить в диалоговом окне *Изменение связей* (см. рис. 3.52) флажки *каскадное обновление* и *каскадное удаление связанных записей* можно только после установки флажка *Обеспечение целостности данных*.

Пример

Проверка поддержания целостности в БД «Автомобили»

1. Проверьте возможность добавления записей в подчинённую таблицу «Записи работ» со значением ключа связи «Код модели», не представленным в главной таблице «Модели». Для этого измените значение ключа связи «Код модели» в подчинённой таблице «Записи работ» на значение, не существующее в записях таблицы «Код модели», и убедитесь, что такое изменение запрещено, т. к. при поддержании целостности не может существовать запись подчинённой таблицы с ключом, которого нет в главной таблице.

2. Проверьте автоматическую поддержку целостности при изменении значений ключей связи в таблицах.

Откройте в *режиме таблицы* главную таблицу «Модели». Измените значение ключевого поля «Код модели» в одной из записей и убедитесь, что во всех записях подчинённых таблиц «Записи работ» и «Купля автомобилей» автоматически также изменится значение поля «Код модели». Изменение происходит везде, поскольку был установлен флажок *каскадное обновление связанных полей*.

3. Проверьте действие режима удаления записей в главной таблице. Для этого удалите одну из записей из главной таблицы «Мастерские». Убедитесь в действие режима каскадного удаления записей в подчинённой таблице «Модели», т. к. был установлен флажок *каскадное удаление связанных записей*. Заметим, что если флажок *каскадное удаление связанных записей* не был бы установлен, то невозможно было бы удалить запись в главной таблице «Мастерские», если имеются связанные с ней записи в подчинённой таблице «Модели».

4. Проверьте действие режима каскадного обновления записей в подчинённой таблице «Купля автомобилей», изменив одну из записей в поле «Фамилия» главной таблицы «Клиенты».

Работа с формами в базе данных

В предыдущем разделе была рассмотрена работа с таблицами базы данных. Недостаток режима таблицы проявляется, когда необходимо просмотреть все поля записи или видеть группу связанных записей в таблице с большим количеством полей. В таких случаях следует использовать *формы*. Кроме этого, форма позволяет: автоматизировать процесс ввода данных, путём добавления в диалоговое окно функциональных элементов; ввести проверку данных; оформить данные для просмотра. Формы позволяют добавлять, удалять или изменять записи в таблицах, а также выводить на экран определённые расчётные данные.

Таким образом форма является объектом БД, предназначенным для ввода, просмотра или изменения данных. Формы ускоряют и делают работу с данными удобной.

Формы могут быть представлены в трёх режимах: формы, макета и конструктора.

Режим формы позволяет в таблицах, на которых основана форма, ввести, просмотреть и осуществить необходимые изменения данных.

Режим макета позволяет настраивать внешний вид и обеспечивает возможность внесения определённых структурных изменений формы.

Конструктор предназначен для разработки формы «с нуля». В этом режиме разрабатывают детальную структуру формы с помощью всех инструментов и с использованием всего набора элементов управления. В этом режиме дорабатывают формы после их создания *мастером* и если невозможно некоторые задачи выполнить в режиме макета. В режиме *конструктора* не предусматривается просмотр данных и внесение каких-либо их изменение.

Формы разрабатываются *однотабличные* и *многотабличные*. Ниже отдельно рассмотрены способы создания таких форм.

Разработка однотабличной формы

Однотабличная форма создаётся на базе единственной таблицы. Такая форма предназначена для просмотра, заполнения и изменения данных одной таблицы.

Access 2013 позволяет создавать формы разнообразными способами. Существует возможность разрабатывать следующие виды форм: *обычная форма* (командой *Форма*); *разделённая форма* (командой *Разделённая форма*); *обычная форма* (командой *Несколько элементов*); *обычная форма* (командой *Мастер форм*) и *Пустая форма*. Данные инструменты размещены в группе *Формы* на вкладке ленты *Создание* (рис. 3.56).

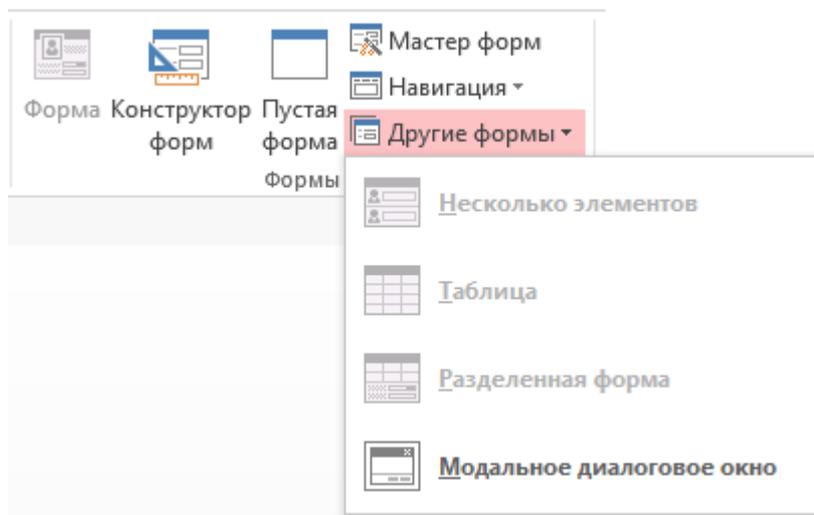


Рис. 3.56. Команды группы *Формы* на вкладке *Создание*

Созданная форма может быть доработана в режимах *конструктора* или *макета*, где можно удалить из неё ненужные поля, добавить новые элементы управления, изменить их размеры и расположение, организовать производство вычислений, задать различные свойства элементов и формы в целом.

Далее будут рассмотрены различные способы проектирования форм, используя в качестве примеров учебную БД «Автомобили».

Вначале изучим наиболее простой способ создания формы, когда большую часть работы автоматически выполняет программа Access.

Пример

Проектирование формы на основе таблицы Клиенты БД «Автомобили».

Самым простым способом автоматического создания формы является использование команды *Форма*.

1. Откройте БД «Автомобили».
2. В области переходов щёлкните таблицу с данными, которые должны отображаться в форме (в качестве источника данных выберите таблицу *Клиенты*).
3. На вкладке *Создание* и в группе *Формы* щёлкните на команде *Форма* (см. рис. 3.56).

Access автоматически создаст новую форму, содержащую все поля таблицы *Клиенты*, расположит эти поля вертикально в один столбец. Заголовок формы соответствует имени таблицы источника, т.е. *Клиенты*.

Примечание: Если в исходной таблице скрыть некоторые поля, например, командой контекстного меню *Скрыть поля*, они не будут включены в форму.

Созданная форма будет отображена в режиме макета (рис. 3.57). В данном режиме можно внести изменения в структуру формы при одновременном отображении данных. Например, при необходимости можно подобрать размер полей в соответствии с данными.

Рис. 3.57. Автоматически созданная форма по команде *Форма* (для работы с таблицей *Клиенты*)

Форма содержит все поля из таблицы *Клиенты* и отображает первую запись из таблицы. Кроме того, так как существует таблица *Купля автомобилей*, связанная с таблицей *Клиенты* отношением «один ко многим» (см. рис. 3.55), в форму добавлена та часть таблицы *Купля автомобилей*, все записи которой относятся к выбранному клиенту.

Примечание: Если существуют нескольких таблиц, связанных с выбранной отношением «один ко многим», чтобы не загромождать форму, они не добавляются.

В нашей форме данные таблицы *Купля автомобилей* не нужны, поэтому эту таблицу можно выделить и удалить.

Для того чтобы сохранить проект формы, выберите из меню *Файл* команду *Сохранить* и в появившемся окне *Сохранение* укажите в качестве имени формы *Потенциальные покупатели*, далее нажмите кнопку *ОК*. Можно просто закрыть форму, но перед закрытием появится окно запроса о сохранении формы и после нажатия кнопки *Да* также появится окно *Сохранение*.

Для каждой формы существуют три режима, в которых она может быть представлена: *Режим формы* – в этом виде форма используется для работы с данными; *Режим макета* – это упрощённый интерфейс создания и редактирования формы и *Конструктор* – является самым сложным, но наиболее

гибким и точным интерфейсом для редактирования формы. На вкладке ленты *Работа с макетами форм / Конструктор* (рис. 3.58) или на вкладке *Инструменты конструктора форм / Конструктор* (рис. 3.59) по левому краю есть команда *Режимы* (рис. 3.60), с помощью которой можно переключиться в требуемый режим представления формы.

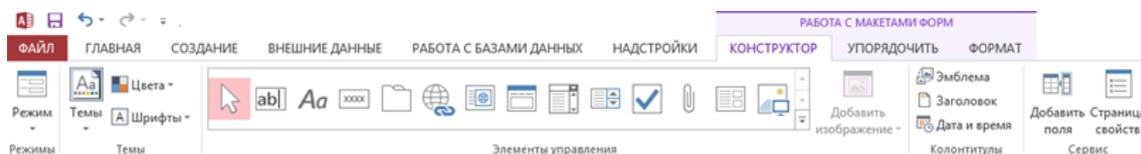


Рис. 3.58. Вкладка *Работа с макетами форм / Конструктор*

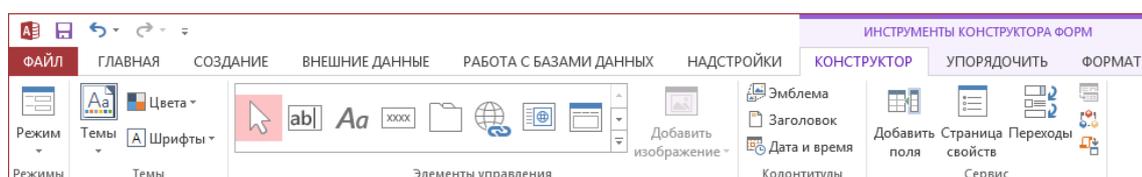


Рис. 3.59. Вкладка *Инструменты конструктора форм / Конструктор*

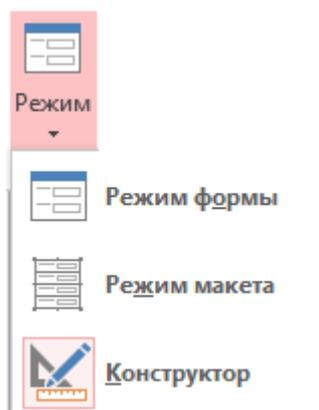


Рис. 3.60. Режимы представления формы

Кроме того, для открытия конкретной формы в том или ином режиме, можно в области переходов, в контекстном меню названия данной формы выбрать нужный режим представления.

Созданная нами таким простым путём форма *Потенциальные покупатели*, уже обладает основными функциями формы, позволяющими вводить и редактировать данные (рис. 3.61).

Рис. 3.61. Форма *Потенциальные покупатели*, созданная по таблице *Клиенты*, в режиме формы

Создание разделённой формы

Разделённая форма содержит две части: данные представленные в виде формы и те же данные представленные в виде таблицы. Такие формы появились впервые в версии Access 2007. Как и простая форма *Разделённая форма* создаётся легко.

Пример

1. В БД «*Автомобили*» в области переходов выделите таблицу с данными *Клиенты*.
2. На вкладке *Создание*, в группе *Формы* щёлкните на команде *Другие формы* и далее нажмите кнопку *Разделённая форма* (см. рис. 3.56).

Access автоматически создаст *разделённую* форму на основе только одной выбранной таблицы, которая отобразится в режиме макета (рис. 3.62).

В любой части разделённой формы можно производить необходимые действия с записями (изменение, добавление или удаление данных) – они всегда синхронизированы. При выделении поля в одной части формы будет выделяться то же поле в другой части. Разделённая форма позволяет использовать преимущества обоих типов представления данных. Например, поиск записей можно быстро осуществить в табличном представлении, а просмотреть или изменить их в другой части формы.

Следует отметить, что если в базе данных уже создана простая форма, основанная на одной таблице, то нет необходимости создавать разделённую форму по той же таблице данных заново. Можно просто изменить свойство этой формы *Режим по умолчанию* для того, чтобы форма с простого приняла разделённый вид.

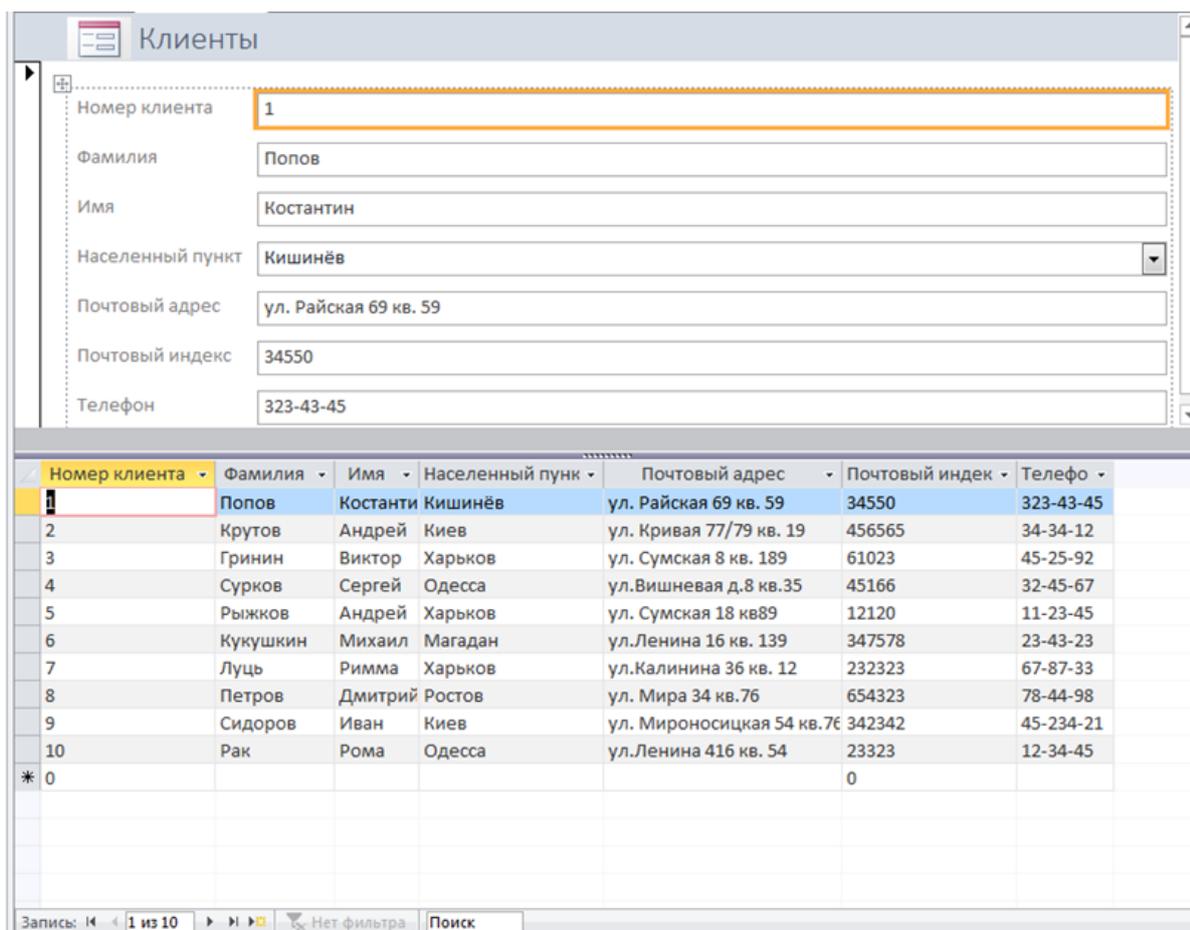


Рис. 3.62. Разделенная форма в режиме макета, созданная автоматически по таблице *Клиенты*

В БД «Автомобили» откройте в режиме конструктора простую форму *Потенциальные покупатели*, созданную по таблице *Клиенты*. Далее, на вкладке ленты *Конструктор* в группе *Сервис* выберите команду *Свойства страницы* и в открывшемся окне свойств, на вкладке *Макет* измените *Режим по умолчанию* с *Простая форма* на *Разделенная форма*.

В том же окне свойств можно выбрать расположение таблицы (свойство *Ориентация разделённой формы*). Свойство может принимать следующие значения: *Таблица сверху*, *Таблица снизу*, *Таблица слева* или *Таблица справа*.

Редактирование формы в режиме макета

Для открытия формы в режиме макета следует в области навигации выделить её и в контекстном меню (рис. 3.63, а) выбрать *Режим макета*.

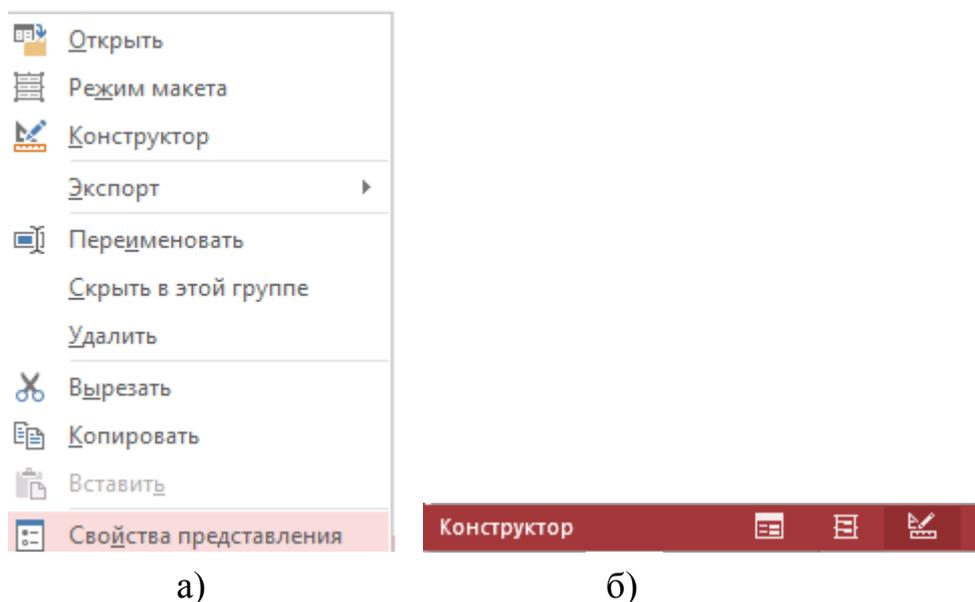


Рис. 3.63. Выбор режима:

а) – в контекстном меню; б) – в строке состояния

Когда форма открыта можно переключаться с одного режима в другой (в том числе в *Режим макета*), щёлкнув на соответствующем значке строки состояния ((рис. 3.63, б) или, выполнив команду *Режим макета* из списка кнопки *Режим* (рис. 3.60) вкладок ленты *Конструктор* (см. рис. 3.58 и рис.3.59) или *Главная*.

При отображении формы в режиме макета появляются вкладки ленты *Работа с макетами форм / Конструктор*, *Упорядочить* и *Формат* (см. рис. 3.58 и рис. 3.64).

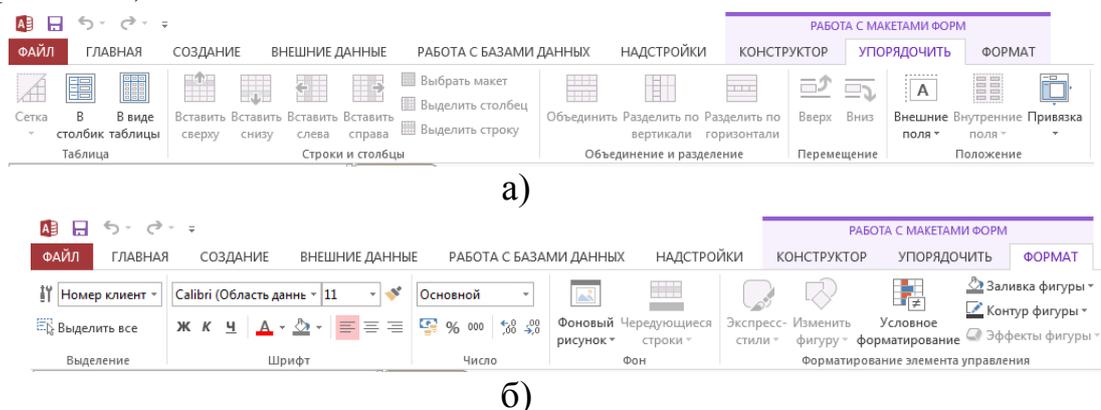


Рис. 3.64. Вкладки ленты *Работа с макетами форм*:
а) – *Упорядочить*; б) – *Формат*.

В режиме макета можно осуществлять определённое редактирование проекта формы, а также просматривать данные почти как в режиме формы. При изменении формы уточняется местоположение, размер, шрифт элементов управления, текста надписей и выполняются другие задачи, связанные с внешним видом формы.

Для вставки в форму даты необходимо на вкладке ленты *Конструктор* в группе *Колонтитулы* щёлкнуть на кнопке *Дата и время* и в открывшемся окне выбрать формат даты, а далее нажать кнопку *ОК*.

Для придания форме нужного стиля в режиме макета можно на вкладке *Конструктор* выбрать понравившуюся тему из имеющейся коллекции.

Редактирование формы в режиме конструктора

После создания нужно отредактировать проект формы: например, изменить расположение и размер отдельных полей, ввести дополнительные элементы оформления, добавить рисунок и др.

Каждый раздел формы характеризуется местом расположения и допустимым набором элементов управления. Любая форма состоит из следующих структурных частей.

Заголовок формы – находится в верхней части, добавляется одновременно с областью примечаний формы и может включать как текст, так и графику. При печати выводится только на первой странице.

Верхний и Нижний колонтитулы – добавляются одновременно, отображаются в режиме предварительного просмотра и используются только при печати формы. Выводятся соответственно вверху и внизу каждой страницы.

Область данных – основная часть формы, где располагаются присоединённые элементы управления и содержатся данные. При печати количество областей данных соответствует количеству записей в источнике данных.

Примечания – добавляется одновременно с областью заголовка и используется для пояснений и примечаний, а также итоговых полей. Печатается только на последней странице.

Пример

На рис. 3.65 представлен пример оформления простой формы *Потенциальные покупатели*. Фон, цвет и размер элементов формы и шрифтов можно выбрать самостоятельно.

ПОТЕНЦИАЛЬНЫЕ ПОКУПАТЕЛИ

Номер клиента:

Фамилия:

Имя:

Населенный пункт:

Почтовый адрес:

Почтовый индекс:

Телефон:

Запись: 1 из 11
Режим формы

Рис. 3.65. Простая формы *Потенциальные покупатели*

Ниже приведены рекомендации по редактированию формы.

1. В БД «Автомобили» откройте в режиме конструктора простую форму *Потенциальные покупатели*, созданную по таблице *Клиенты*.
2. Сверните *Область перехода (навигации)*.
3. Увеличьте ширину и высоту проекта формы: перетащите с помощью мыши правый и нижний край *Области данных* на весь экран.
4. Необходимо расположить поля формы в два столбца так, как это предложено в образце. Для этого необходимо предварительно разгруппировать элементы, удалив макет (в контекстном меню: *Макет / Удалить макет*)
5. Измените шрифт, цвет и рельефность полей формы. Вначале выделите требуемые поля (например, методом *Лассо* или щёлкните по каждому заголовку, удерживая нажатой клавишу **Shift**, либо проведите по горизонтальной линейке напротив столбца с заголовками полей). Выведите диалоговое окно *Свойства*. Установите для полей следующее оформление:
 - для заголовков левого столбца – шрифт *Arial Cyr*, размер шрифта – *выберите самостоятельно* (учитывайте, что на экране будут только те поля, которые представлены в форме и размер шрифта нужно выбрать достаточный для комфортного чтения), насыщенность – *Плотный*, оформление – *Приподнятое*, выравнивание текста – *По левому краю*;
 - для полей заголовков правого столбца – гарнитуру и размер шрифта – *выберите самостоятельно*, насыщенность – *Плотный*, оформление – *Приподнятое*, выравнивание текста – *По левому краю*;

- для полей с данными – шрифт *Arial Cyr*, размер шрифта – *выберите самостоятельно*, насыщенность *Плотный*, оформление *Вдавленное*.

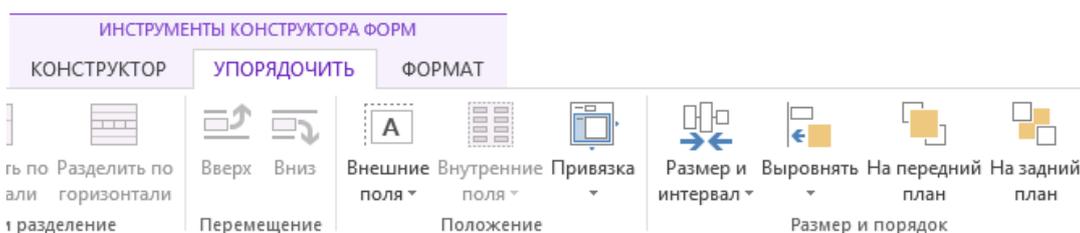


Рис. 3.66. Фрагмент вкладки ленты
Инструменты конструктора форм / Упорядочить

6. Для того, чтобы надписи и данные помещались в рамки полей, надо выделить их, и в контекстном меню в списке *Размер*, или на вкладке ленты *Инструменты конструктора форм / Упорядочить*, в группе *Размер и порядок* нажать кнопку *Размер и интервал* (рис. 3.66).

Далее в открывшемся списке *Размер* (рис. 3.67), выбрать команду *по размеру данных*.

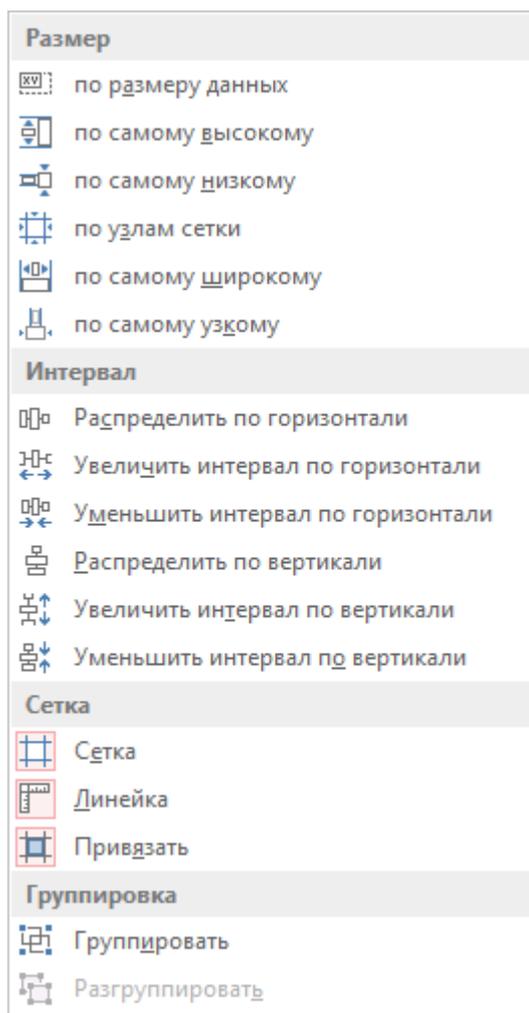


Рис. 3.67. Список команд кнопки
Размер и интервал

7. После данной процедуры поля надписей и данных могут перекрываться. Произведите выравнивание полей так, как это указано в образце.
8. Для распределения элементов в области данных, например по вертикали, можно разместить мышкой в необходимые места верхние и нижние элементы. Затем выделить все элементы и в контекстном меню или нажав кнопку *Размер и интервал*, выбрать в группе *Интервал* команду *Распределить по вертикали* (см. рис. 3.67)
9. Для выравнивания элементов, нужно после их выделения, в контекстном меню или нажав кнопку *Выровнять*, выбрать необходимую команду (*Слева, Справа, Сверху, Снизу, По сетке*).
10. Для установки одинаковой высоты всех полей формы выделите все элементы любым способом, далее выберите команду *Свойства* в контекстном меню и на вкладке *Макет* в строке *Высота* введите требуемое значение.
11. Теперь необходимо изменить ширину полей. Следующие поля должны иметь одинаковую ширину:
 - названия полей *Номер клиента, Фамилия и Имя* и соответствующие области данных;
 - названия полей *Населённый пункт, Почтовый адрес, Почтовый индекс, Телефон* и соответствующие области данных.
12. Для этого, выделите поля, размер которых необходимо изменить. Выведите диалоговое окно *Свойства* и установите нужную ширину полей. Для уравнивания размеров нескольких полей выделите их и выберите из контекстного меню *Размер* команду *По самому широкому* либо *По самому узкому*.
13. Отделите прямоугольником часть формы, содержащую номер, имя и фамилию клиента. С помощью кнопки *Прямоугольник* (она на вкладке *Конструктор* в группе *Элементы управления*) начертите прямоугольник вокруг группируемых полей. Затем щёлкните дважды по прямоугольнику для вывода диалогового окна *Свойства* и установите ширину линии 4 пункта, цвет выберите самостоятельно.
14. Оформите заголовок формы. Расширьте область заголовка. В увеличенную область заголовка поместите надпись *Потенциальные покупатели* (используйте кнопку *Надпись* в группе *Элементы управления*). Выберите и установите самостоятельно гарнитуру и размер шрифта, насыщенность – *жирный*, тип границы – *Сплошная*, выравнивание текста – *По центру*, оформление *Вдавленное*.
15. В правую часть *Заголовка формы* вставьте рисунок. Для этого щёлкните по кнопке *Добавить изображение*, откройте требуемую папку и выберите необходимый файл. После вставки рисунка уменьшите его до нужных размеров. Вы увидите, что часть рисунка стала невидимой. Чтобы исправить это, выведите окно свойств рисунка и выберите необходимое значение свойства *Установка размеров*:

- *Фрагмент* – рисунок выводится на экран в натуральную величину. Если размеры рисунка превышают размеры рамки, рисунок обрезается.
 - *Вписать в рамку* – размеры рисунка изменяются так, чтобы он полностью заполнял рамку. Это может привести к искажению пропорций рисунка.
 - *По размеру рамки* – размеры рисунка изменяются таким образом, чтобы он заполнял рамку по ширине или по высоте без изменения пропорций и без обрезания рисунка.
16. Последнее, чего надо добиться при оформлении формы – выбор оптимальных её размеров, так чтобы она не была излишне большой и чтобы на ней отсутствовали полосы прокрутки. Для этого перейдите в режим *Конструктора* формы и установите мышью с помощью растягивания границ окна размер формы приблизительно так, как это предложено в образце. Затем переключитесь в режим *Формы* и просмотрите, есть ли на форме полосы прокрутки и достаточно ли компактно на ней размещены данные. Если внешний вид формы не совпадает с образцом, снова переключитесь в режим *Конструктора* и изменяйте размеры формы, пока не добьётесь соответствия с образцом.
 17. Используя разработанную форму, добавьте в таблицу *Клиенты* ещё несколько произвольных записей.

Примечание. Основное назначение формы – создание удобного интерфейса для редактирования данных. Возможность добавления, удаления и изменения записей в связанную с формой таблицу регулируется свойствами формы: *Разрешить добавление*, *Разрешить удаление* и *Разрешить изменения* (см. в режиме конструктора формы *Окно свойств*).

Последовательное перемещение между записями можно осуществлять, используя кнопки перехода *панели навигации* (внизу экрана) формы (рис. 3.68).



Рис. 3.68. Панель навигации формы

Для просмотра записей удобно также пользоваться фильтрацией. Для этого надо установить курсор в фильтруемое поле, например, *Имя*, и нажать в группе *Сортировка и фильтр* на вкладке *Главная* кнопку *Фильтр*. Откроется меню фильтра (рис. 3.69), где можно выбрать и сортировку, которая действует одновременно с фильтрацией.

Когда фильтр назначен, кнопка на панели навигации со значком фильтра активна. Она также включает и выключает фильтр.

Поле *Поиск* панели навигации (см рис.3.68) позволяет осуществить поиск искомого значения в любом поле формы.

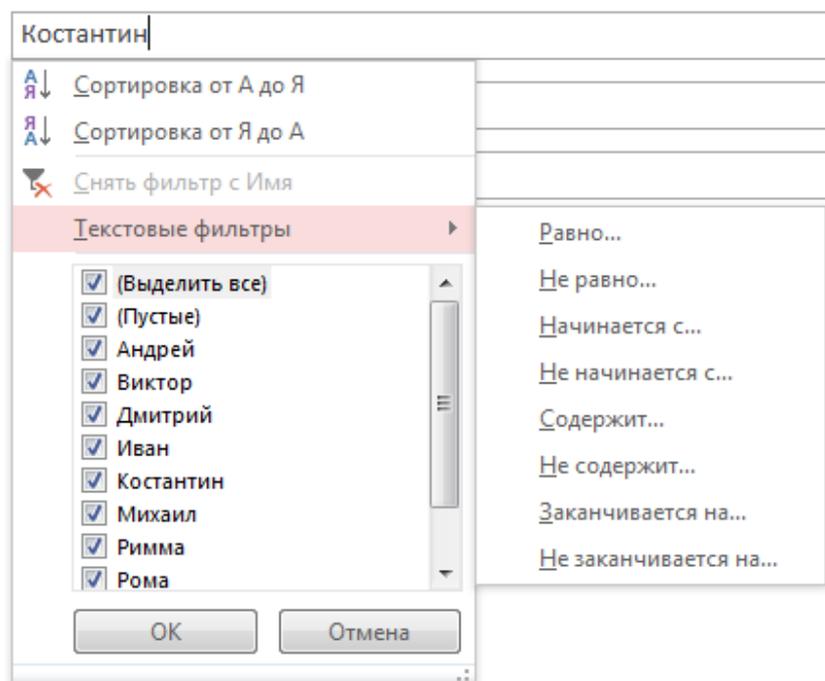


Рис. 3.69. Меню фильтра

Создание новой формы в режиме конструктора

Для создания новой формы необходимо на вкладке ленты *Создать* в группе *Формы* выполнить команду *Конструктор форм*. Последует открытие в режиме конструктора пустой формы, содержащей только раздел *Область данных*.

Вначале следует определить источник данных формы. Для этого необходимо на вкладке ленты *Конструктор* в группе *Сервис* щёлкнуть на кнопке *Добавить поля* (см. рис. 3.59) и из появившейся области *Список полей* перетащить нужные поля в форму. Добавить *Заголовок* и *Примечание формы* можно кнопкой *Заголовок* (см. рис. 3.59) из группы *Колонтитулы* ленты *Конструктор*. Далее необходимо редактировать и форматировать макет формы.

В качестве примера создадим в режиме конструктора диаграммную форму.

Пример

1. В окне базы данных *Автомобили* на вкладке ленты *Создание* в группе *Формы* выберите *Конструктор форм*.
2. В появившейся вкладке *Конструктор* выберите элемент управления *Диаграмма* и щёлкните в области данных конструктора формы, далее в диалоговом окне мастера *Создание диаграмм* укажите таблицу, на которой будет базироваться диаграммная форма (таблица *Модели*).

3. В следующем окне мастера диаграмм нужно указать те поля таблицы, которые будут включены в форму. При этом необходимо соблюдать порядок маркировки, он должен соответствовать требуемому порядку включения полей в форму. Для данной формы целесообразно выбрать поля: *Заводская цена*, *Транспортные издержки* и *Модель*.
4. Во втором окне мастера диаграмм необходимо выбрать *тип диаграммы*. Можно выбрать *Гистограмма* или *Линейчатая*, но в данном случае лучше всего подойдут трёхмерные варианты *Объёмная гистограмма* или *Цилиндрическая*. Выберите *Объёмная гистограмма*.
5. В следующем окне необходимо указать элементы диаграммы. Чтобы указать *легенду диаграммы* (описание условных обозначений) необходимо перетащить поле *Модель* в область *Ряды*.
6. В области *Данные* должны располагаться данные, представляемые (в данном примере) в виде столбцов диаграммы. Сюда необходимо перенести поля *Заводская цена* и *Транспортные издержки*.
7. После переноса полей обнаружится, что поля *Заводская цена* и *Транспортные издержки* в области *Данные* изменили свои названия на следующие: *СуммаЗаводская цена* и *СуммаТранспортные издержки*. Дело в том, что по умолчанию над этими полями для каждой категории легенды производится операция суммирования. В данном примере эта операция не является необходимой, так как каждая модель представлена в таблице одной записью. Чтобы отменить операцию суммирования, следует дважды щёлкнуть на поле *СуммаЗаводская цена*, а затем на поле *СуммаТранспортные издержки*. В появившемся диалоговом окне *Операции суммирования* следует выбрать *Отсутствует*.
8. В последнем окне мастера диаграмм необходимо подобрать диаграмме подходящий заголовок. По умолчанию в качестве заголовка диаграммы используется имя базовой таблицы. Однако в данном случае более подходящим будет заголовок «*Структура цен автомобилей*».
9. Далее, чтобы отобразить на экране готовую форму, необходимо нажать кнопку *Готово*. Сохраните полученную диаграмму под именем *Структура цен*.

Для изменения диаграммы необходимо перейти в режим конструктора формы. Произведите настройку диаграммы, увеличив её до полного экрана и придав ей вид, например, как представленный на рис. 3.70.



Рис. 3.70. Диаграммная форма

Проектирование составной формы с использованием мастера

Составная форма содержит *главную* форму и одну или несколько *подчинённых* форм. Подчинённая форма строится на основе таблицы, подчинённой таблице главной формы, т. е. находится с ней в отношении *один ко многим*. В подчинённой форме отображаются данные из всех записей подчинённой таблицы, которые связаны с записью, отображаемой в главной форме.

Составная форма создаются мастером форм, который запрашивает какие поля следует включать в форму из нескольких взаимосвязанных таблиц и запросов. Форма, полученная с помощью мастера, редактируется в режиме макета или конструктора.

Пример

Спроектируем составную форму в БД «Автомобили», в которой главная форма, построенная на основе таблицы *Модели* связана с подчинённой формой, созданной на основе таблицы *Клиенты*. При помощи такой формы можно будет получить сведения об отдельных моделях автомобилей и о заказавших их клиентах.

1. В окне базы данных на вкладке ленты *Создание* в группе *Формы* выберите *Мастер форм*.
2. В первом диалоговом окне мастера необходимо выбрать таблицы и поля для проектируемой формы. В списке *Таблицы/Запросы* выберите таблицу

Модели, а из списка *Доступные поля* – поля, участвующие в форме (выберите все поля, кроме поля *Специальная модель*). Из таблицы *Модели* в принципе обязательно необходимо выбрать ключевое поле *Код модели* и все другие обязательные поля, иначе форма будет непригодна для ввода сведений о новых моделях. Затем из того же списка *Таблицы/Запросы* выберите таблицу *Клиенты*, которая является таблицей для подчинённой формы. Внесите в проект формы следующие поля в указанной очередности: *Фамилия*, *Имя*, *Населённый пункт*, *Почтовый адрес*, *Телефон*. Нажмите кнопку *Далее*.

3. В следующем окне Access предлагает построить проектируемую форму в виде главной (родительской) и подчинённой (дочерней), которые соединены в одной составной форме (должна быть установлена опция *Подчинённые формы*). В качестве главной формы Access определит форму на основе таблицы *Модели*, проанализировав связи между таблицами. Нажмите кнопку *Далее*.
4. В следующем окне выбирается вид подчинённой формы – *Ленточный* или *Табличный*. Выберите *Табличный* и нажмите кнопку *Далее*.
5. В последнем окне мастера можно присвоить имена главной и подчинённой формам. Укажите для главной формы заголовок – *Автомобили и их покупатели*. Для подчинённой формы оставьте имя, выбранное Access. Установите опцию *Изменить макет формы*. Нажмите кнопку *Готово*. На экране появится в режиме конструктора готовый проект формы, который необходимо доработать.
6. Для того чтобы сохранить проект формы, выберите из меню *Файл* команду *Сохранить* и укажите имя *Заказы* для данной формы.

Оформление составной формы

Далее необходимо отредактировать проект формы, чтобы сделать его более выразительным: добавить заголовок, графический файл и изменить расположение отдельных полей. Пример оформления формы *Заказы* приведён на рис.3.71.

1. Увеличьте ширину и высоту проекта формы: перетащите с помощью мыши правый и нижний край *Области данных*.
2. Перетащите подчинённую форму вниз с таким расчётом, чтобы в освободившемся промежутке опустить поля главной формы. Маркируйте подчинённую форму и поместите курсор мыши на край этой формы, чтобы он приобрёл вид крестика. Теперь, удерживая нажатой левую кнопку мыши, перетащите подчинённую форму в нижнюю часть области данных.
3. Аналогично п.2 переместите поля главной формы вниз, освободив верхнюю часть окна.

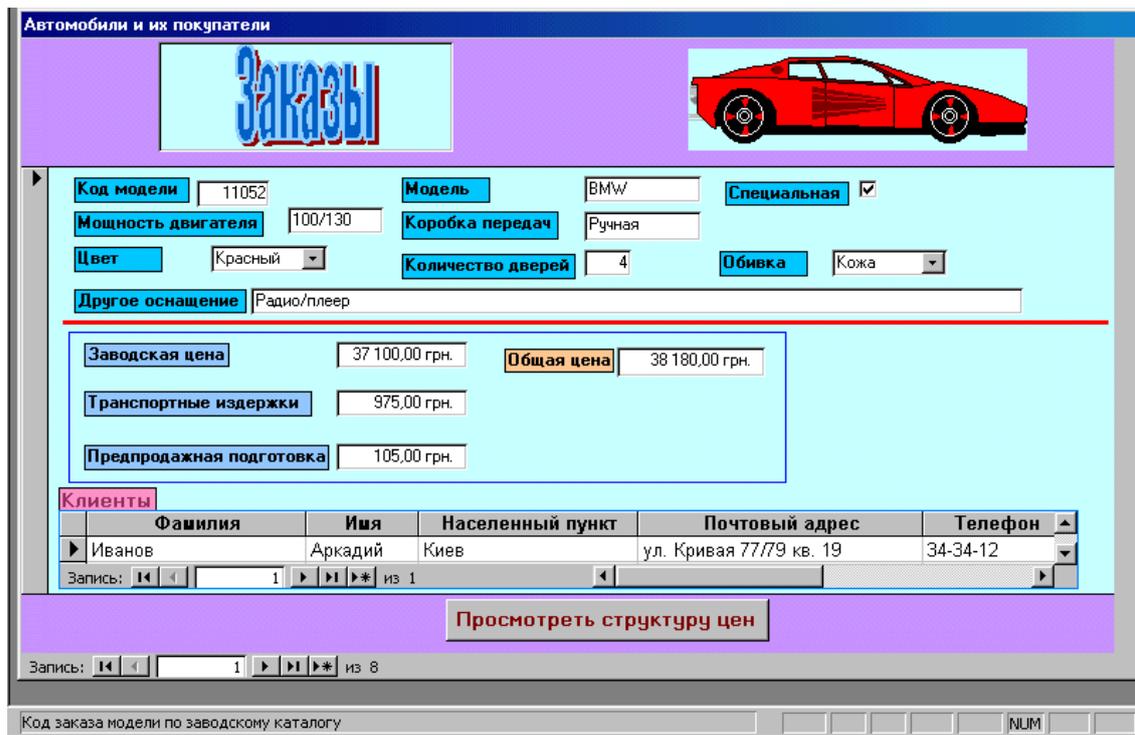


Рис. 3.71. Окно формы *Заказы*

4. Необходимо расположить поля главной формы более компактно. Поле *Модель* расположите в первой строке справа от поля *Код модели*. Расположите во второй строке *Мощность двигателя* и *Коробка передач*, в третьей – *Цвет*, *Количество дверей*, *Обивка*, в четвертой - *Другое оснащение*. Затем через небольшой интервал поместите поля (каждое на своей строке) *Заводская цена*, *Транспортные издержки*, *Предпродажная подготовка*. Пододвиньте прямоугольник, представляющий подчинённую форму, поближе к последнему полю главной формы.
5. Теперь необходимо изменить размеры полей. Установите курсор мыши на линейке, расположенной по левому краю макета формы, напротив той строки, к которой располагаются элементы, размер которых необходимо изменить. Курсор примет вид стрелки, направленной вправо. Нажмите кнопку мыши. Все элементы, расположенные в данной строке, будут маркированы. Выберите из меню *Формат* подменю *Размер* команду *По размеру данных*, с помощью которой подбирается оптимальный размер выделенных элементов.
6. Далее необходимо подогнать размеры полей. Поле *Другое оснащение* может содержать довольно длинный текст. Выполните щелчок на элементе *Другое оснащение*. За исключением левого верхнего угла, занятого маркером перемещения, каждый угол и середина каждой стороны имеет собственный размерный маркер. С их помощью увеличьте размер элемента.
7. Рассмотрим поле *Специальный элемент*, которое в проекте таблицы *Модели* было определено как логическое поле и не было вставлено в форму.

Поместим в форму информацию, о какой модели идёт речь (специальной или нет). Щёлкните на вкладке *Конструктор* в группе *Элементы управления* на кнопку *Флажок*, а затем в проекте формы щёлкните справа от поля *Модель*. Access разместит управляющий элемент типа *Флажок* вместе с соответствующим полем наименования. Переместите надпись слева от флажка.

8. С помощью двойного щелчка на поле надписи откройте окно свойств поля *Надпись*, в котором следует установить параметры данного поля (окно свойств можно открыть, нажав кнопку *Страница свойств* из группы *Сервис* вкладки *Конструктор*, либо в контекстном меню). Большинство параметров можно оставить без внимания, сохранив установки, используемые по умолчанию. В поле *Подпись* вкладки *Макет* введите название поля *Специальная модель*.
9. Откройте окно *Флажок*, в котором следует установить параметры данного элемента. В поле *Данные* вкладки «*Данные*» нужно указать имя поля таблицы, содержимое которого должен отражать этот элемент. Выберите из раскрывающегося списка поле *Специальная модель*.
10. Чтобы отделить группу полей, описывающих технические характеристики машины, от полей цен, проведите разделительную линию между полями *Другое оснащение* и *Заводская цена*. В группе *Элементы управления* нажмите кнопку *Линия* и проведите линию по всей ширине формы. Параметры линии установите самостоятельно.
11. Под разделительной линией в области данных находятся три поля денежного типа, используя которые необходимо вычислить общую цену и показать её в форме. Следовательно, необходимо добавить в форму вычисляемое поле. На вкладке ленты *Конструктор* в группе *Элементы управления* нажмите кнопку *Поле* и разместите это поле рядом с полем *Заводская цена*. Поле наименования будет содержать просто номер поля, а текстовое поле – ссылку *Свободный*. Вызовите окно свойств для наименования и введите в поле *Подпись* значение *Общая цена*. Вызовите окно свойств текстового поля. В поле *Данные* вкладки «*Данные*» нужно ввести следующее выражение $=[\text{Заводская цена}]+[\text{Транспортные издержки}]+[\text{Предпродажная подготовка}]$ (для облегчения ввода можно использовать *Построитель выражения*). В поле *Формат поля* вкладки *Макет* нужно ввести значение *Денежный*.
12. Чтобы оформить часть формы, содержащую данные о цене, эти данные необходимо сгруппировать и выделить. С помощью кнопки *Прямоугольник* группы *Элементы управления* начертите прямоугольник вокруг группируемых полей.
13. Отформатируйте по своему усмотрению текст в полях: шрифт, его размер, начертание, выравнивание. Для ускорения процедуры единообразного оформления нескольких полей, маркируйте их, а затем все вместе оформите.

14. Оформите *Заголовок формы*. Расширьте область заголовка. В увеличенную область заголовка поместите надпись *Заказы* (используйте кнопку *Надпись* группы *Элементы управления* вкладки *Конструктор* либо создайте надпись WordArt в документе Word и скопируйте её в заголовок формы).
15. Вставьте иллюстрацию автомобиля в форму: в группе *Элементы управления* выберите команду *Добавить изображение*, найдите и вставьте картинку с автомобилем.

Встраивание в форму кнопок

Далее встроим в форму кнопку, с помощью которой можно будет переключиться на диаграмму *Структура цен*. При создании кнопки нужно задать, какую функцию будет выполнять данная кнопка. Для этого следует написать *макрос*, реализующий эту функцию.

1. Закройте форму и вернитесь в окно базы данных. На вкладке *Создание* в группе *Макросы и код* нажмите кнопку *Макрос*. Появится окно создания макроса.
2. Щёлкните в любом месте окна и затем откройте список *Добавить новую макрокоманду*. Выберите макрокоманду *Открыть форму*. В поле *Имя формы* выберите из списка форму *Структура цен*. В поле *Режим* установите опцию *Форма*. В поле *Режим данных* установите опцию *Только чтение*. В поле *Режим окна* установите опцию *Обычное*. Сохраните макрос под именем *Открытие диаграммы*, закройте окно макроса и откройте форму *Заказы* в режиме конструктора.
3. Теперь в форму можно вставить кнопку. Её удобно поместить в область примечаний формы. Выполните щелчок на значке кнопки группы *Элементы управления* и разместите указатель мыши в области примечаний формы приблизительно посередине экрана. Отпустите кнопку мыши и на экране появится пронумерованная кнопка. Теперь этой кнопке необходимо присвоить имя и связать с ней некоторое действие. (Если после вставки кнопки открылось окно мастера *Создание кнопок*, то закройте его, нажав кнопку *Отмена*).
4. Выполните двойной щелчок на созданной кнопке для открытия окна свойств. Установите следующие характеристики кнопки: Имя – *Взгляд на диаграмму*, Подпись – *Просмотреть структуру цен*, Нажатие кнопки – *Открытие диаграммы*. Закройте окно свойств. Перейдите в режим формы и проведите испытания кнопки. Должна открыться диаграммная форма *Структура цен*.
5. Теперь необходимо организовать возврат к исходной форме *Заказы*. Один из способов – переключение на форму *Заказы* через закрытие окна формы *Структура цен*. Второй способ – создание дополнительной кнопки в форме *Структура цен*, посредством которой диаграммная форма будет

закрыта, и произойдёт возврат к исходной форме. Следовательно, необходимо создать соответствующий макрос, но можно воспользоваться мастером создания кнопок.

6. Перейдите в режим конструктора для диаграммной формы. Выполните щелчок на значке *Кнопка* группы *Элементы управления* и переместите указатель мыши в области данных. На экране появится пронумерованная кнопка и откроется окно мастера создания кнопок. (Будем создавать кнопку с помощью мастера без использования макроса). Кнопка *Мастера* при этом должна быть включена (*Окно свойств – Данные – Включена – Да*). В первом окне мастера в поле *Категория* выберите *Работа с формой*, в поле *Действия* – *Закрыть форму*. Во втором окне – указывается, что необходимо разместить на кнопке – текст или рисунок (выберем рисунок). В третьем окне задайте название для кнопки. Нажмите кнопку *Готово*. Перейдите в режим формы и проведите испытания кнопки.

Изменение свойств формы

1. Находясь в режиме конструктора формы *Заказы* из контекстного меню выберите команду «Свойства формы».
2. Откроется диалоговое окно «Форма» (обратите внимание на то, что заголовок окна должен называться именно «Форма». В противном случае будут изменены свойства каких-либо других объектов.
3. Установите *Тип границы – Тонкая*, Кнопки размеров окна – *Отсутствуют*. Перейдите в режим формы и посмотрите, каким образом повлияло изменение свойств на возможность минимизации, максимизации и изменения размеров формы.
4. Выведите диалоговое окно изменения свойств подчинённой формы и на вкладке *Все* в строке *Блокировка* установите опцию *Да*. После установки этого свойства поля доступны только для чтения. При этом станет невозможным изменение, добавление и удаление данных в полях подчинённой формы. Перейдите в режим формы и убедитесь в этом.

Создание запросов

Запрос является одним из основных объектов база данных Access и представляет собой обращение к данным для получения информации из базы данных или выполнения действий над ними. Запросы упрощают просмотр, добавление, удаление или изменение данных в базе данных. Кроме того, запросы используются: для быстрого поиска определённых данных путём фильтрации с применением определённых критериев (условий); для вычисления или сведения данных; для автоматизированного управления данными, например регулярного просмотра актуальных данных.

В нормально структурированной базе данных все сведения хранятся в разных таблицах. Запрос может извлечь информацию из разных таблиц и собрать её для отображения в виде формы или отчёта.

Существуют следующие основные типы запросов:

- *на выборку* – получение данных из одной или нескольких таблиц и выполнение вычислений в соответствии с заданными условиями отбора, а результатом является виртуальная таблица, которая существует только до закрытия запроса.
- *на изменение* – добавление, изменение или удаление данных. (для каждой задачи существует специальный тип запроса на изменение);
- *на создание таблицы* – создание новой таблицы на основе данных одной или нескольких существующих таблиц, результат сохраняется в новой постоянной таблице;
- *на автоподстановку* – позволяет автоматически заполнить поля для новой записи;

Далее рассмотрим запрос на выборку, который является наиболее распространённым типом запроса.

Запросы на выборку

В таком запросе извлекаются данные из одной или нескольких таблиц, и результаты отображаются в режиме таблицы, в которой допускается изменение записей (при некоторых ограничениях). Кроме того, запросы на выборку используются для группировки записей, а также для вычисления сумм, средних значений, подсчёта количества записей и расчёта итоговых значений других типов.

Результат выполнения запроса представляет собой таблицу и называется динамическим или временным набором данных. В эту таблицу включаются выбранные из исходной таблицы записи, которые удовлетворяют определённым критериям запроса. Как только запрос будет закрыт, динамический набор данных ликвидируется, хотя данные, которые были представлены пользователю, останутся в исходной таблице. Таким образом, при каждом выполнении запроса он строится на основе современных (существующих в данный момент) табличных данных. Таблица запроса не хранится в базе данных – хранится запрос, который может быть использован пользователями для выполнения задач.

Существует четыре пути создания запросов:

- с помощью мастера запросов;
- с использованием конструктора запросов;
- в режиме SQL-редактора;
- на основе существующего фильтра.

Запрос можно создавать, воспользовавшись мастером запросов или же самостоятельно, с использованием конструктора. Мастер запросов ускоряет процесс создания запроса, автоматически выполняя все основные операции. Вызванный мастер запросов запрашивает сведения и создаёт запрос на основе ответов пользователя. Затем можно перейти в режим конструктора и доработать запрос. При самостоятельной разработке нового запроса необходимо в режиме конструктора выбрать таблицы, содержащие нужные данные и заполнить бланк запроса нужными полями (перетащить).

Рассмотрим построение запросов в режиме конструктора на примере базы данных «Автомобили».

Построение простых запросов с помощью конструктора по одной таблице

Пример

Требуется вывести список всех мастерских высшей категории, отсортированный в порядке убывания цены ремонта.

1. Откройте базу данных *Автомобили* и выберите вкладку *Создание*.
2. Щёлкните в группе *Запросы* по кнопке *Конструктор* запросов. Появится вкладка ленты *Работа с запросами / Конструктор* (рис. 3.72) и откроется пустое окно запроса на выборку в режиме конструктора, а также диалоговое окно *Добавление таблицы* (рис. 3.73).
3. В этом окне нужно выбрать таблицу *Мастерские* и нажать кнопку *Добавить*. Выбранная таблица будет отображена в области схемы данных запроса. Закройте окно *Добавление таблицы*, нажав кнопку *Закреть*.

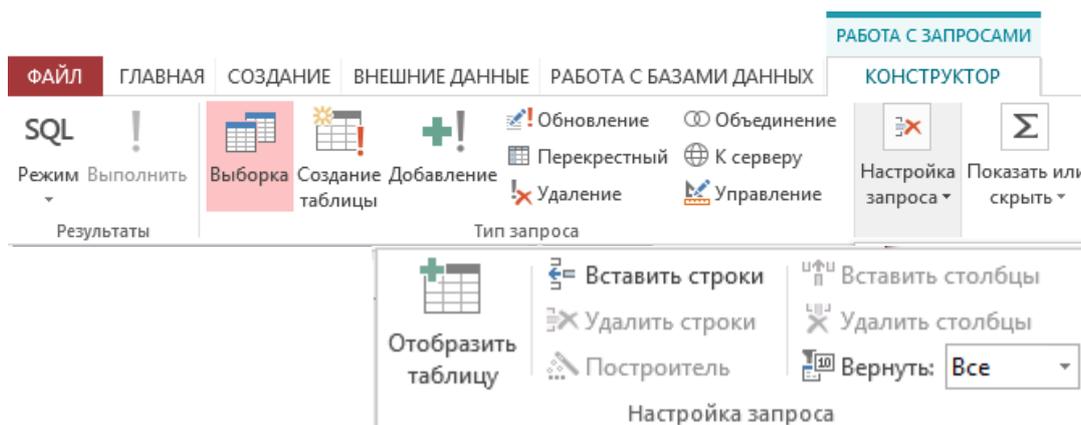


Рис 3.72. Инструменты вкладки ленты
Работа с запросами / Конструктор

Примечание. Когда необходимо добавить любую таблицу в схему данных запроса, надо открыть окно *Добавление таблицы* (см. рис. 3.73) нажав

кнопку *Отобразить таблицу* в группе *Настройка запроса* (см. рис. 3.72) либо выполнив команду *Добавить таблицу* в контекстном меню. Для удаления любой таблицы из схемы данных запроса надо установить курсор мыши на удаляемую таблицу и в контекстном меню выбрать *Удалить таблицу* или нажать клавишу *Delete*.

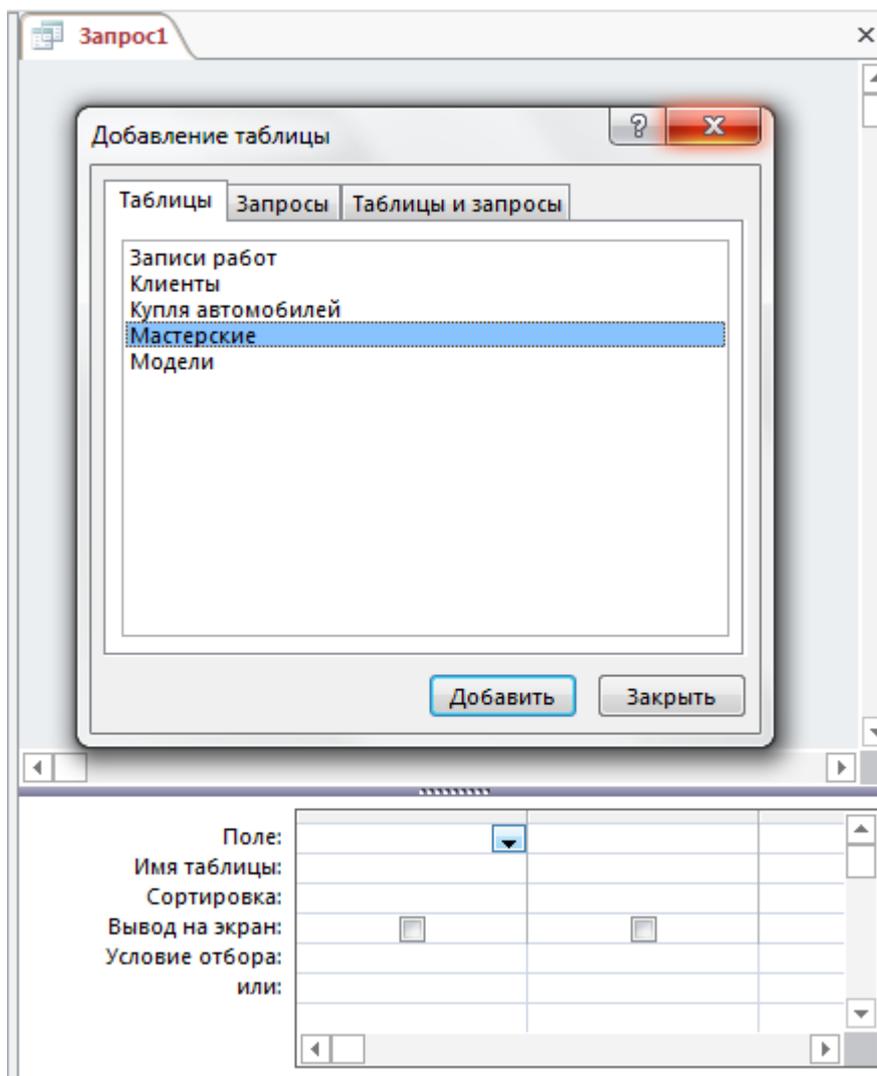


Рис 3.73. Окно запроса и окно выбора и добавления таблиц и запросов

4. Далее необходимо поместить в запрос поля: *Номер мастерской, Телефон, Директор, Средняя цена, Категория*.

Вывод нужных полей из исходной таблицы в соответствующие столбцы запроса можно осуществить любым из следующих приёмов: дважды щёлкнуть по каждому из них; перетащить их мышью в поле запроса; в первой строке бланка запроса *Поле* щелчком мыши вызвать появление кнопки списка и выбрать из списка нужное поле (список содержит поля таблиц, представленных в схеме данных запроса).

Для удаления по ошибке установленное поле в бланке запроса надо выделить соответствующий столбец и нажать клавишу *Delete* или выполнить команду *Удалить столбцы* в группе *Настройка запроса*.

Чтобы требуемые поля были включены в таблицу запроса необходимо в строке *Вывод на экран* отметить их флажками.

5. Для просмотра результатов выполнения запроса надо в группе *Результаты* щёлкнуть на кнопке *Выполнить* или на кнопке *Режим* и выбрать команду *Режим таблицы*. Выполнив запрос, на экране откроется окно запроса в режиме таблицы с записями из таблицы *Мастерские* с отображенными полями.
6. Имя запроса не должно совпадать не только с именами имеющихся запросов, но и с именами таблиц в базе данных. Сохраните запрос с именем *Мастерские высшей категории*.
7. В созданном запросе фигурируют все мастерские. Теперь следует ввести фильтр с тем, чтобы выводились только мастерские высшей категории. Перейдите в режим конструктора, в поле *Категория* установите курсор в строку *Условие отбора* (см. рис. 3.74) и напечатайте критерий "высшая". Обратите внимание, что Access сам заключает напечатанный критерий в кавычки. Перейдите в режим таблицы и снова просмотрите результаты запроса. Все мастерские не высшей категории должны исчезнуть, т.е. отфильтроваться.

Поле:	№ мастерской	Телефон	Директор	Средняя цена	Категория
Имя таблицы:	Мастерские	Мастерские	Мастерские	Мастерские	Мастерские
Сортировка:				по убыванию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:					"высшая"
или:					

Рис 3.74. Заполненный бланк запроса

8. Теперь поле *Категория* в таблице показывать бессмысленно, т.к. оно содержит одинаковое значение для всех записей. Перейдите в режим *Конструктора* и в строке *Вывод на экран* в поле *Категория* снимите флажок – вывод на экран данного поля будет отменен. Категория мастерских уже определена в имени запроса
9. Список результата запроса выводится в порядке следования записей в таблице, т.е. в общем случае он никак не упорядочен. Для выполнения сортировки перейдите в режим конструктора и в строке *Сортировка* (см. рис. 3.73) поля *Средняя цена* из выпадающего списка выберите опцию *По убыванию*. Перейдите в режим таблицы и убедитесь в том, что полученный список результата запроса отсортирован в порядке убывания средней цены ремонта.

10. Сохраните последний вариант запроса. Сохраняется сам запрос, а не его результаты. В случае изменения данных в таблице список результата запроса будет также изменён.

Для выполнения любого созданного запроса нужно в базе данных в области переходов выбрать желаемый запрос и дважды щёлкнуть мышью на его имени. Будет запущен выбранный запрос и на экране представится таблица с результирующим набором данных, которые были выбраны из базы данных для этого запроса. Для редактирования запроса надо выделить его в области переходов и выполнить в контекстном меню команду *Конструктор*.

Таким образом, создание простых запросов с помощью конструктора по одной таблице в Microsoft Access не вызывает сложностей.

Использование логических функций при построении запросов

Все необходимые критерии для фильтрации данных следует заносить в строку *Условия отбора*. Для реализации функции **НЕ** следует использовать знак <> или ключевое слово *Not*, для реализации функции **ИЛИ** – ключевое слово *or*, для реализации функции **И** – ключевое слово *and*. Между условиями, записанными в одной строке и разных столбцах, выполняется логическая операция *And*. Между условиями, записанными в разных строках одного и того же столбца, выполняется логическая операция *Or*. Кроме этого, в условиях отбора используются математические операторы: > – больше, < – меньше, >= – больше или равно, <= – меньше или равно. Оператор *Between* позволяет задать интервалы дат и числовых значений.

После записи каждого условия отбора целесообразно переключаться в режим таблицы и просматривать результат. Обращайте внимание на то, как сокращается количество записей.

Особенности построения запросов по нескольким таблицам

Запросы по нескольким таблицам (многотабличные запросы) или, так называемые сложные запросы, позволяют выполнять следующее: представлять данные из нескольких связанных таблиц в виде одной; выбирать данные по заданным пользователем условиям; создавать вычисляемые поля, значения которых будут основываться на нескольких полях таблиц.

При создании многотабличного запроса надо добавить в схему данных запроса не только те таблицы, поля которых будут задействованы, но и все таблицы, которые связывают требуемые нам таблицы. Если между таблицами в запросе не установлена связь, то в запросе будут формироваться все комбинации записей таблиц, т. е. объединяются все со всеми, что является не достоверным.

Пример

Задание: вывести в базе данных *Автомобили* упорядоченный в алфавитном порядке список всех клиентов вместе с маркой и цветом купленных автомобилей.

1. Откройте базу данных *Автомобили* и на вкладке ленты *Создание* выберите режим *Конструктор запросов*.
2. Нужные сведения можно получить, только лишь используя две таблицы: *Клиенты* и *Модели*. Выполнив команду *Добавить таблицу* в контекстном меню, выберите последовательно эти таблицы (см. рис. 3.73). Закрыв окно добавления таблиц можно заметить, что данные таблицы не связаны. Из схемы данных (см. рис. 3.55) видно, что для их связывания нужна таблица *Купля автомобилей*, поэтому добавим её в схему данных запроса. Обратите внимание на то, что таблицы автоматически связались (так как связывание этих таблиц было предварительно осуществлено).
3. Далее необходимо поместить в бланк запроса поля *Фамилия*, *Имя* из таблицы *Клиенты* и *Код модели*, *Модель*, *Цвет* из таблицы *Модели*. Заметим ещё раз, что из таблицы *Купля автомобилей* никаких полей нам не нужно, однако её необходимо добавить в запрос для связывания таблиц *Модели* и *Клиенты*.
4. Для упорядочения списка в алфавитном порядке фамилий клиентов необходимо в строке *Сортировка* поля *Фамилия* из выпадающего списка выбрать опцию *По возрастанию*.
5. Для отображения результата запроса щёлкните на кнопке *Выполнить* в группе *Результаты* и просмотрите результаты выполнения запроса, а также убедитесь в том, что полученный список упорядочен.

Добавление вычисляемых полей в запросы

При создании запроса есть возможность создавать дополнительное поле, значение которого для каждой записи будет основываться на вычислениях значений из других полей одной или нескольких таблиц. Такое поле называется *вычисляемое*. Для создания такого поля необходимо ввести в строку *Поле* выражение, которое будет вычислять значение данного поля. При составлении выражения используются ссылки на поля, имена которых заключаются в квадратные скобки. Следует заметить, что в исходных таблицах базы данных новые поля не создаются. При каждом выполнении запроса производятся вычисления на основе текущих значений полей таблиц. В выражениях вычисляемых полей помимо имён полей могут использоваться функции и константы.

Пример

Задание: В базе данных *Автомобили* посчитать продажную цену автомобилей (с учётом заводской цены, транспортных расходов, предпродажной подготовки и скидки) и вывести упорядоченный в порядке убывания итоговой

цены список клиентов с указанием фамилии, имени клиента, марки купленного автомобиля и его продажной цены.

1. Чтобы не создавать запрос заново, можно воспользоваться результатами предыдущего запроса. Для этого нужно открыть его и сохранить под новым именем.
2. Далее оставьте в бланке запроса только поля *Фамилия*, *Имя* и *Код модели*, *Модель*, а не нужное в данном случае поле *Цвет* удалите.
3. Для организации вычисляемого поля щёлкните правой кнопкой мыши правее самого последнего поля и из контекстного меню выберите команду *Построить*.

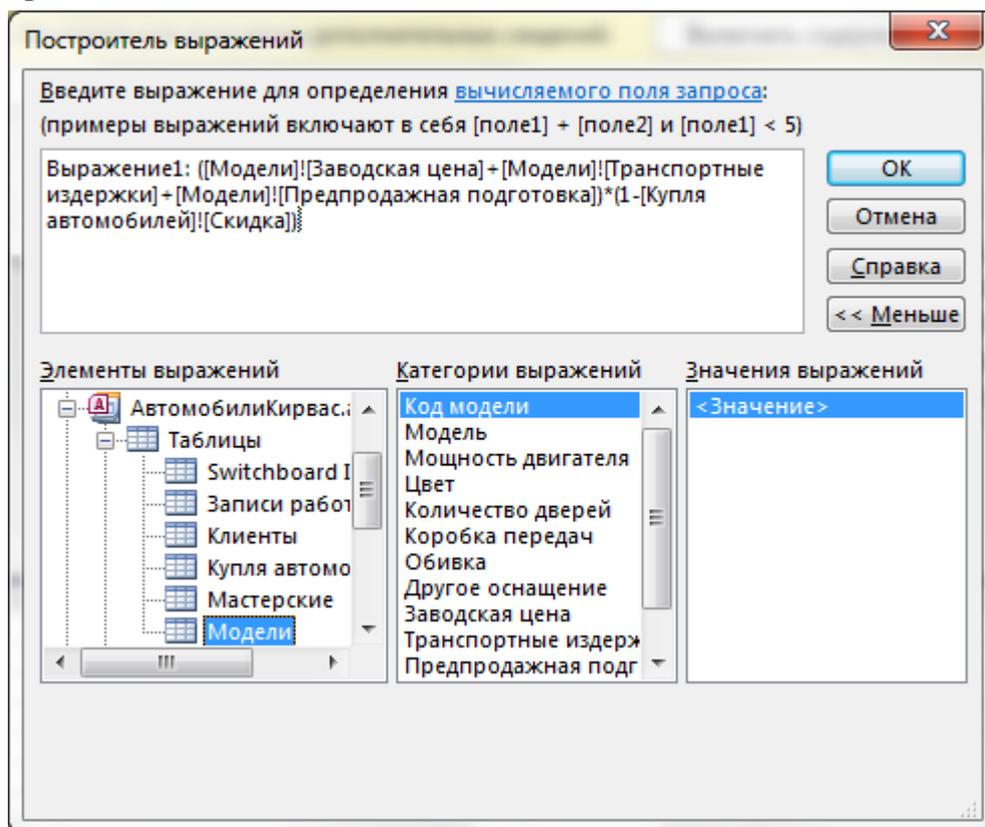


Рис. 3.75. Окно построителя выражений.

В открывшемся диалоговом окне (Рис. 3.75) построителя выражений следует организовать следующее выражение:

$$([Модели]![Заводская\ цена]+[Модели]![Транспортные\ издержки]+[Модели]![Предпродажная\ подготовка])*(1-[Купля\ автомобиля]![Скидка])$$

4. Для этого в окне построителя выражений (см. рис. 3.75) в группе *Элементы выражений* щёлкните значок + левее имени базы данных и далее откройте список таблиц. В этом списке выберите последовательно таблицы *Модели* и *Купля автомобиля*, а в группе *Категории выражений*, соответственно для таблицы *Модели* – поля *Заводская цена*, *Транспортные издержки*, *Предпродажная подготовка* и для таблицы *Купля автомобиля*

– поле *Скидка*. После ввода выражения система по умолчанию формирует имя вычисляемого поля *Выражение1*, которое становится заголовком столбца в таблице с результатами выполнения запроса. Обратите внимание, что имя вычисляемого поля отделяется от выражения двоеточием.

5. Нажмите *ОК* для выхода из диалогового окна и, переключившись в режим таблицы, просмотрите результат выполнения запроса.
6. Чтобы дать вычисляемому полю понятное название следует вместо заголовка поля *Выражение1* написать требуемое имя вычисляемого поля, например, *Продажная цена*. Иначе можно в режиме конструктора выделить поле, выбрать в контекстном меню пункт «Свойства» и в появившемся диалоговом окне в строке *подпись* написать требуемое имя. Здесь же можно установить формат поля (в данном случае *денежный*) и требуемое число десятичных знаков.
7. Отсортируйте полученную таблицу в порядке убывания продажной итоговой цены автомобилей. Для этого в строке *Сортировка* вычисляемого поля *Продажная цена* из выпадающего списка выберите опцию *По убыванию*. Перейдите в режим таблицы и убедитесь в том, что полученный список теперь упорядочен.

Импорт таблиц

Построенная база данных «Автомобили» содержит мало данных. Для построения в учебных целях более разнообразных запросов целесообразно создать новую базу данных, в которую следует импортировать все таблицы из учебной базы данных *Борей* (из приложения Access 2003) корпорации Microsoft. Ниже перечислены действия, которые необходимо выполнить.

1. Создайте новую пустую базу данных любым известным способом (*Файл – Создать* либо стартовое окно (рис. 3.2.), появляющееся автоматически при запуске программы MS Access). Сохраните созданную пустую базу данных с именем: *Ваша фамилия_Борей*.
2. Созданную при этом автоматически таблицу в БД можно удалить.
3. На вкладке ленты *Внешние данные* нажмите кнопку *Access* и далее кнопку *Обзор*, затем откройте ту папку, в которой находится учебная база *Борей*.
4. Выделите файл *Борей.mdb* и нажмите кнопку *Открыть* и далее *ОК*.
5. В открывшемся окне *Импорт объектов* (рис. 3.76), используя кнопку *Выделить все*, выделите все таблицы. Важно импортировать сразу все таблицы, а не по частям, чтобы сохранились межтабличные связи.
6. Нажмите кнопку *ОК*, после чего все выделенные таблицы импортируются в вашу базу данных. Убедитесь, что в вашей БД присутствуют все восемь таблиц импортированные из базы данных *Борей*.

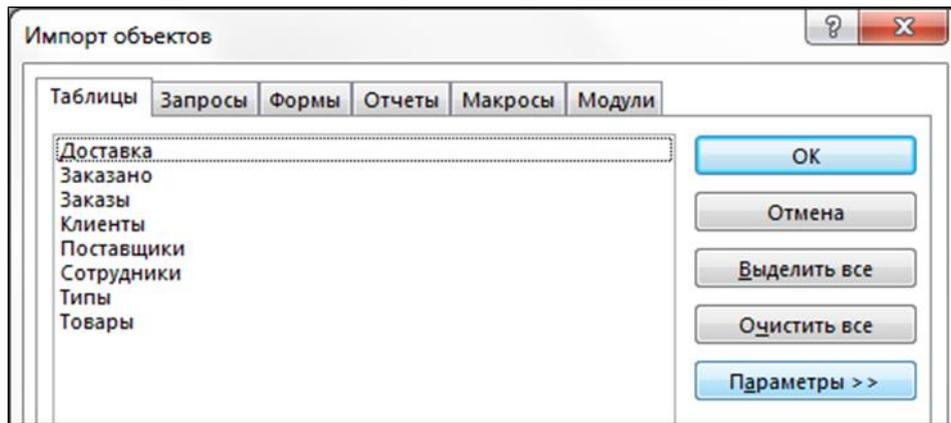


Рис. 3.76. Диалоговое окно *Импорт объектов*

- Оформите схему данных базы данных Борей. Для этого на вкладке *Работа с базами данных* нажмите кнопку *Схема данных*. В открывшемся окне с помощью мыши расположите таблицы так, как это указано на рис. 3.77. Добейтесь, чтобы были видны все поля в таблицах и наглядно отображались связи между всеми таблицами.

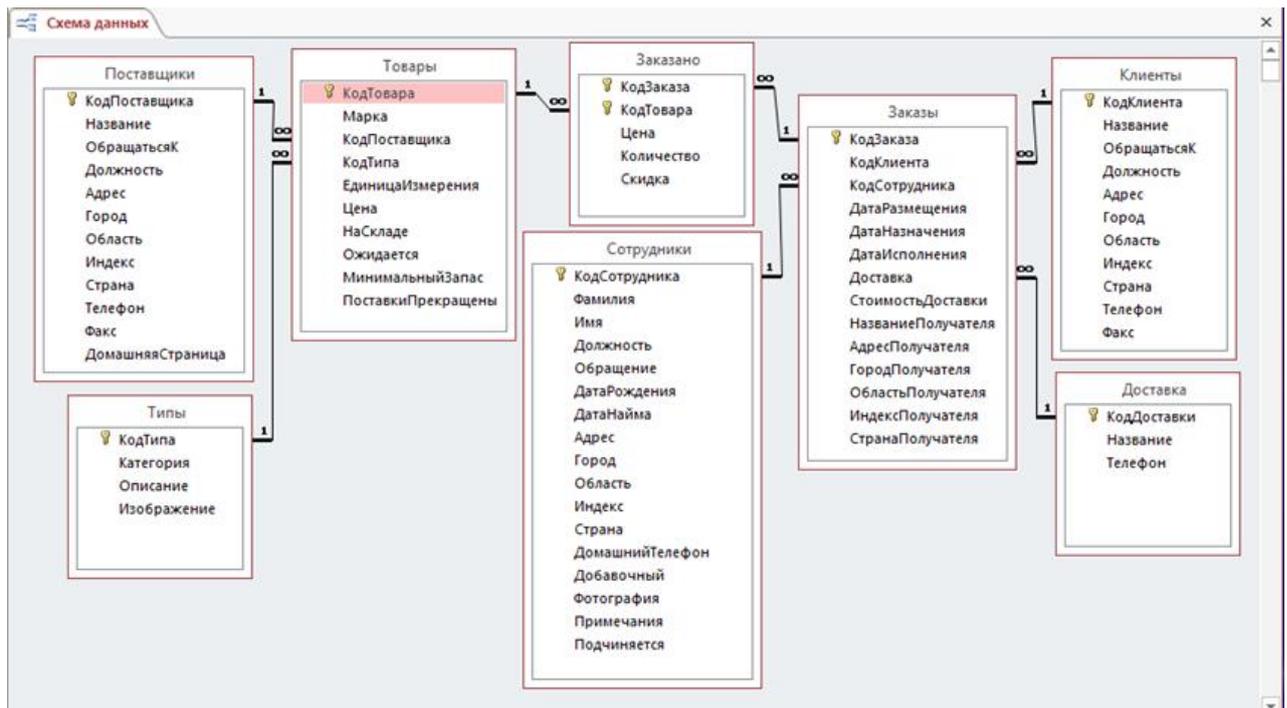


Рис. 3.77. Схема данных БД Борей

Использование шаблонов при организации запросов

При построении запросов в базах данных *Access* используются те же шаблоны, что и при поиске файлов, а также некоторые другие:

- * — любое количество символов;

? – один любой символ;

– любая одиночная цифра от 0 до 9;

[список] – любой одиночный символ, входящий в список (например, [А-Д] – это буква А, Б, В, Г или Д); используется в сочетании с шаблонами * или ?;

[!список] – любой одиночный символ, не входящий в список (например [!А-Б] – это любая буква, только не А и не Б). Данный шаблон также используется в сочетании с шаблонами * или ?.

Пример

Рассмотрим создание запросов с использованием шаблонов в базе данных Борей.

Запрос 1 Вывести список всех поставщиков, страна которых начинается на букву **Н**, а минимальный запас поставляемых ими товаров больше нуля.

Чтобы определить какие таблицы и поля нужны для создания запроса необходимо открыть *Схему данных*. Из схемы видно, что нужно добавить в запрос таблицы *Поставщики* и *Товары*, откуда следует выбрать соответственно поля *Название*, *Страна* и *Марка*, *МинимальныйЗапас*. В строке запроса *Условие отбора* в поле *Страна* следует набрать **Н***, а в поле *МинимальныйЗапас* – ввести **>0**. После перехода в режим таблицы обратите внимание, что в строке *Условие отбора* перед построенным шаблоном появился оператор *Like*.

Запрос 2 Вывести список марок товаров, количество которых на складе выражается трёхзначным числом.

В условиях запроса в поле *НаСкладе* введите *Like "###"*.

Запрос 3 Вывести список клиентов, название которых начинается буквами А, В, С, D, Е.

Данный запрос можно выполнить двумя способами: используя логическую функцию *Or* и шаблон *Like "[А-Е]*"*.

Запрос 4 Вывести список клиентов, в телефоне которых отсутствует код (т. е. номер телефона не начинается открывающей скобкой).

Используйте для поля *Телефон* таблицы *Клиенты* шаблон *Like "[!()]"*.

Запрос 5 Вывести список клиентов, которые разместили свои заказы в декабре 1996 года.

Используйте для поля *ДатаРазмещения* таблицы *Заказы* шаблон *Like "##.12.1996"*

Запрос 6 Вывести список заказов, оформленных каждым сотрудником, и город клиента, который не начинается на букву **Б** и отсортирован по возрастанию.

Используйте для поля *Город* таблицы *Клиенты* шаблон *Like "[!Б]*"*.

Анализ данных с помощью запросов (итоговые запросы)

Запросы не ограничиваются только выборкой. Можно осуществлять группировку данных в запросах при этом выделить группы записей с одинаковыми значениями в указанных полях и вычислить итоговые данные для каждой из групп по другим полям. С помощью таких запросов, называемых итоговыми, можно получить определённые статистические данные из таблиц.

Для анализа данных используются следующие статистические функции.

Sum – суммирует значения элементов поля (в столбце). Работает только с числовыми или денежными данными.

Avg – вычисляет среднее арифметическое набора значений, содержащихся в определённом поле запроса. Столбец должен содержать числовые или денежные величины или значения даты или времени. Функция игнорирует пустые значения.

Count – подсчитывает количество значения элементов в столбце. С помощью данной функции можно подсчитать, например, количество заказов, отправленных в определённую страну.

Max и **Min** – возвращает элемент, имеющий наибольшее или наименьшее значение в поле (столбце). Для текстовых данных наибольшим будет последнее по алфавиту значение, а наименьшим будет первое по алфавиту значение, причём регистр не учитывается. Функция игнорирует пустые значения.

StDev – показывает, насколько значения отклоняются от среднего значения поля в столбце (среднеквадратичное отклонение).

Var – вычисляет статистическую дисперсию для всех значений в столбце. Эта функция работает только с числовыми и денежными данными. Если таблица содержит менее двух строк, то возвращается пустое значение.

First и **Last** – значение поля из первой или последней записи в группе.

Итоговый запрос применяется также, когда требуется использовать значение результата в другом объекте базы данных, например, в отчёте.

Создание итоговых запросов с группировкой данных рассмотрим на примере в базе данных *Автомобили*.

Пример

Задание 1. Построить запрос, позволяющий для каждой модели автомобилей посчитать число проводимых ремонтов, среднюю цену ремонта и общую сумму, уплаченную за все ремонты.

1. Откройте базу данных *Автомобили*, на вкладке ленты *Создание* запустите конструктор построения запросов и добавьте таблицы «*Модели*» и «*Записи работ*».
2. Поместите в бланк запроса поля *Модель*, *Код записи обслуживания* и дважды *Оценочная цена*.

3. Переключитесь в режим таблицы и просмотрите результат выполнения запроса. В таблице представлены все модели, находившиеся в ремонте, причём каждому автомобилю соответствует столько записей, сколько раз он находился в ремонте.
4. Для ответа на поставленные в задании вопросы следует произвести группировку записей. В режиме конструктора в контекстном меню зоны бланка запроса выберите команду «Итоги». Или на вкладке ленты *Конструктор* в группе *Показать или скрыть* можно нажать кнопку *Итоги*. Появится дополнительная строка и в каждом поле запись *Группировка*.
5. Переключитесь в режим таблицы для просмотра результата и убедитесь, что результирующий список запроса пока не изменился. В режиме конструктора в контекстном меню строки «Групповые операции» замените слово *Группировка* в поле *Код записи обслуживания* на функцию **Count** (для подсчёта числа проводимых ремонтов), в первом поле *Цена ремонта* – на функцию **Avg** (для подсчёта средней цены ремонта), во втором поле *Цена ремонта* – на функцию **Sum** (для подсчёта общей суммы за все ремонты). Бланк запроса примет вид, показанный на рис. 3.78.

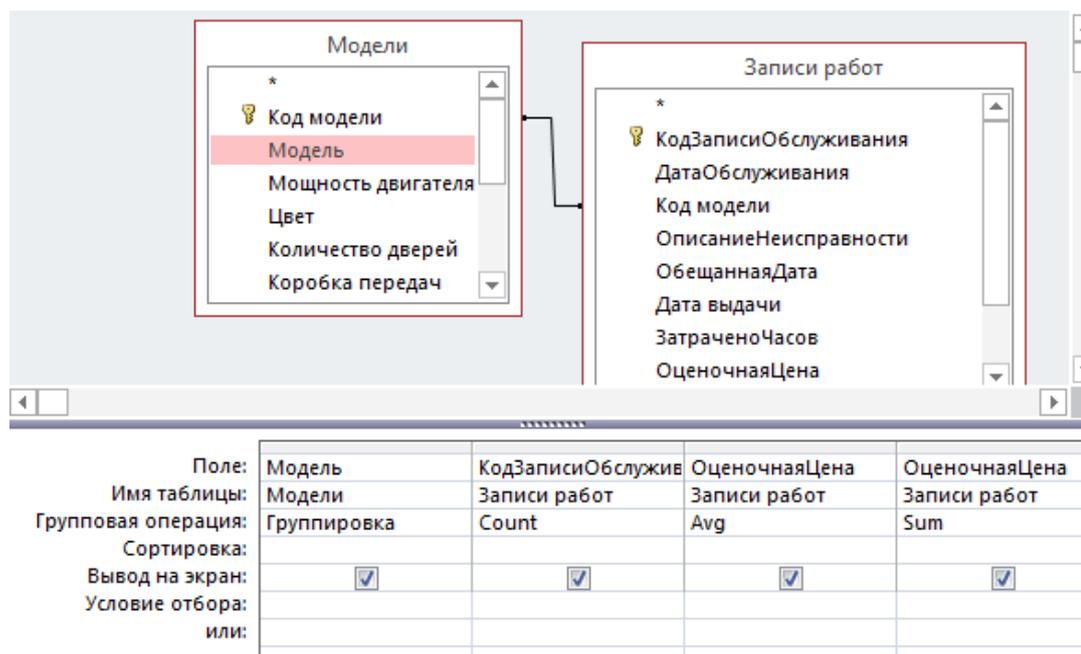


Рис. 3.78. Запрос с группировкой по модели автомобиля и подсчётом числа проводимых ремонтов, средней цены ремонта и общей суммы, уплаченной за все ремонты

6. Переименуйте поля (*Свойства – Подпись*) в соответствии с их назначением и установите соответствующий формат данных.
7. Выполните запрос и просмотрите результат.

Задание 2. Определить 5 клиентов, затративших наибольшую общую сумму денег на покупку автомобилей.

1. Постройте запрос с полем *Фамилия* таблицы *Клиенты* и со следующим вычисляемым полем:
$$([Заводская\ цена] + [Транспортные\ издержки] + [Предпродажная\ подготовка]) * (1 - [Скидка]).$$
2. Осуществите группировку по полю *Фамилия* и суммирование по вычисляемому полю.
3. Произведите сортировку в порядке убывания значений в вычисляемом поле, чтобы наибольшие суммы размещались впереди.
4. Необходимо ограничить список выводом только первых пяти записей. Для этого нужно вывести окно свойств запроса щелчком правой кнопкой мыши по верхней серой части конструктора запросов, а затем установить в свойстве *Набор значений* число 5. А можно поступить даже проще: на вкладке *Конструктор*, в группе *Настройка запросов* открыть список *Вернуть*, в котором по умолчанию установлена опция *Все*, и выбрать число 5 или ввести нужное число.
5. Выполните запрос и просмотрите результат.

Условия отбора в запросах с групповыми операциями

В запрос с групповыми операциями (итоговый запрос) помимо полей, по которым производится группировка и выполнение статистических функций, можно из таблиц включать поля для задания условий отбора записей.

Создание итоговых запросов с использованием условий рассмотрим на примере в базе данных *Борей*.

Пример

Задание. Для каждого клиента посчитать количество оформленных им заказов, исполненных не ранее 1 июня 1996 года. Список упорядочить по убыванию числа заказов. Вывести список клиентов, которые оформили более 10 заказов.

1. Откройте базу данных *Борей*, на вкладке ленты *Создание* запустите конструктор построения запросов и добавьте таблицы *Клиенты* и *Заказы*, в бланк запроса добавьте поле *Название* из первой таблицы и поле *Код Заказа* – из второй таблицы. Добавьте строку *Групповые операции*, в первом поле оставьте значение «группировка», а во втором установите значение *Count*. В свойство поля *Подпись* наберите: *Количество оформленных заказов*. Выполните запрос и просмотрите результат.

Далее, чтобы отобразить только нужные группы записей в режиме конструктора, надо ввести условие отбора в поле, по которому производится группировка, или в поле, где записана функция.

Данный запрос необходимо доработать следующим образом.

2. В поле *Код заказа* в строке *Условие отбора* добавьте условие: **>10**.
3. Помимо двух имеющихся полей, добавьте поле *Дата Исполнения* из таблицы *Заказы*. В строке *Групповая операция* в данном поле появится значение «группировка». Переключитесь в режим таблицы и убедитесь, что ограничение на дату не выполняется.
4. Далее в строке *Групповая операция* измените значение «группировка» на значение «условие». В строке *Условие отбора* впишите условие: **>=#1/06/1996#**. Переключитесь опять в режим таблицы и убедитесь в правильности выполнения запроса.

Обратите внимание, что одновременно с выбором условия отключается режим вывода данного поля на экран. Включите режим вывода поля на экран и после попытки переключения в режим таблицы можно прочитать сообщение, выдаваемое Access в ответ на данное действие:

Приложению «Microsoft Access» не удаётся вывести поле, для которого задано условие в строке групповых операций. Снимите для этого поля флажок «Вывод на экран».

Для отображения поля в результатах запроса добавьте его в бланк запроса дважды. Для отображаемого поля не задавайте значение условия и установите флажок «Вывод на экран».

Отображение строки итогов по столбцу

В режиме таблицы используется *строка итогов* для быстрого расчёта и отображения в столбце таблицы или запроса таких значений, как *итоговая сумма, среднее, количество значений, минимальное и максимальное значение, стандартное отклонение, дисперсия*.

Для добавления строки итогов в таблицу запроса откройте предыдущий запрос в режиме таблицы. На вкладке ленты *Главная* в группе *Записи* нажмите кнопку *Итоги*. В таблице отобразится строка *Итог*, где нажмите кнопку со стрелкой вниз в столбце, для которого требуется выполнить расчёт (*Количество оформленных заказов*), и выберите в списке, например, *Количество значений*. Результат расчёта количества значений в этом столбце показан на рис. 3.79. Для того чтобы скрыть строку итогов, надо повторно на вкладке ленты *Главная* нажать кнопку *Итоги*.

Название	Количество оформленных заказов
Save-a-lot Markets	26
Ernst Handel	28
QUICK-Stop	28
Folk och fa HB	19
Hungry Owl All-Night Grocers	19
HILARION-Abastos	18
Итого	26

Итого	Нет
	Сумма
	Среднее
	Количество значений
	Максимальное значение
	Минимальное значение
	Стандартное отклонение
	Дисперсия

Рис. 3.79. Отображение строки итогов с расчётом количества значений по столбцу

Параметрические запросы

В примерах, рассмотренных ранее условие отбора вводилось в режиме конструктора в бланке запроса. Для того, чтобы задать новое значение в условие отбора, нужно повторно открывать запрос и вводить его. Намного удобнее вводить условие отбора в процессе выполнения запроса в диалоге с пользователем, не переходя в режим конструктора. Такой диалог обеспечивается с помощью, так называемого, параметра запроса. В режиме конструктора в бланке запроса в строке *Условия отбора* задаётся имя параметра запроса в квадратных скобках. При выполнении запроса это имя появится в диалоговом окне *Введите значение параметра* при выполнении запроса. Таким образом, это даёт возможность использовать запрос многократно, не открывая его в конструкторе для изменения условий.

Создание параметрических запросов рассмотрим на примере в базе данных *Борей*.

Пример

Задание 1. Построить запрос, позволяющий вывести список клиентов с указанием *названия, города, страны, номера телефона*, а также *должности*, которая будет указываться пользователем перед выполнением запроса.

1. В режиме конструктора в бланк запроса добавьте таблицу *Клиенты* и вызовите необходимые поля. Для поля *Должность* в строке *Условие отбора*

напечатайте фразу в квадратных скобках [*Укажите должность клиента*]. Таким образом задаётся параметр для поля *Должность*.

2. Выполните запрос и в появляющемся при переходе в режим таблицы диалоговом окне (рис.3.80) напечатайте нужную должность, например, *Совладелец* или *Продавец*.

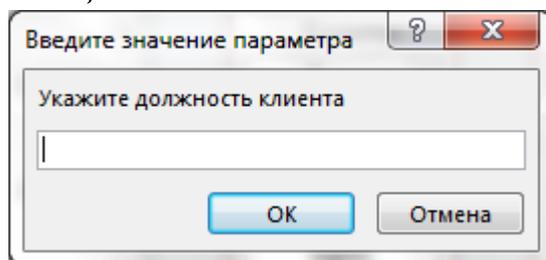


Рис. 3.80. Диалоговое окно ввода значения параметра

3. Пронаблюдайте, каким образом используется заданный параметр и убедитесь в правильности выполнения запроса.
4. Иногда требуется ввести не точное значение параметра, а интервал значений. Пусть, например, следует для каждого клиента вывести *коды заказов*, которые были размещены позже вводимой пользователем даты. После добавления в запрос необходимой таблицы *Заказы* и полей *КодЗаказа*, *ДатаРазмещения* в строке *Условие отбора* для последнего поля следует напечатать фразу: *>[Позже какой даты был размещён заказ?]*.
5. Посмотрите в таблице *Заказы* диапазон дат заказа, затем выполните запрос и просмотрите результат.

В запросе могут быть определены несколько параметров, которые связаны определёнными логическими функциями *And*, *Or* и др.

6. Пусть, например, следует вывести список клиентов, проживающих в любой из двух стран, вводимых пользователем. Тогда в строке *Условие отбора* поля *Страна* следует напечатать фразу:

[Введите первую страну] or [Введите вторую страну]

7. Регистр логической функции *Or* значения не имеет. Выполните запрос и убедитесь в правильности выбора записей.

Задание 2. Выведите список *кодов заказов*, исполненных сотрудниками с фамилиями *Акбаев* и *Кротов*, при этом пользователь должен задавать промежуток времени: *после и до какой даты исполнен заказ*.

1. В бланк запроса добавьте таблицы *Заказы*, *Сотрудники* и вызовите из них, соответственно, необходимые поля: *КодЗаказа*, *ДатаИсполнения* и *Фамилия*.
2. Для поля *Сотрудники* в строке *Условие отбора* следует задать: *"Акбаев" Or "Кротов"*.

3. Для определения числового интервала можно использовать параметры с операторами: >, <, а также параметры в операторе *Between*. Например, для поля *ДатаИсполнения* можно задать следующее выражение:

Between [Введите после какой даты исполнен заказ] And [Введите до какой даты исполнен заказ].

При создании параметрических запросов Access по умолчанию присваивает полю с параметром *текстовый* тип. Если организовывать параметр, например, для *числового* поля, запрос выполняется неправильно. Поэтому важно указать, данные какого типа разрешается вводить в качестве значения параметра. Тип данных можно настроить для любого параметра, но особенно важно сделать это для числовых и денежных данных, а также данных о дате и времени. Когда для параметра указан тип данных, пользователи получают более понятные сообщения об ошибках в случае ввода данных неправильного типа, например ввода текста, когда ожидаются денежные данные. Если параметр настроен таким образом, чтобы принимать текстовые данные, любое введённое значение интерпретируется как текст и сообщение об ошибке не отображается.

Чтобы указать тип данных для параметра в запросе: в режиме конструктора на вкладке *Конструктор* в группе *Показать или скрыть* нажмите кнопку *Параметры* или в контекстном меню области запроса и выберите команду *Параметры*. В появившемся диалоговом окне в левом столбце впишите название параметра без квадратных скобок. В правом столбце выберите тип данных для каждого параметра, например, один из числовых типов данных.

Чтобы избежать появления ошибок, можно также воспользоваться следующим методом: вставьте параметр в функцию *Val([...])*.

Задание 3. Для каждого сотрудника выведите список тех заказов, которые были просрочены, т.е. их *дата исполнения находится позже даты назначения*. Организуйте параметр, позволяющий пользователю указывать *нижний предел числа просроченных дней*.

1. В бланке запроса постройте выражение:

[Заказы]![ДатаИсполнения]-[Заказы]![ДатаНазначения].

2. Для поля, где было создано выражение, в строке *Условие отбора* задайте необходимый параметр и выполните запрос, сначала без учёта приведённых выше рекомендаций, затем организуйте параметр, например, так:

>=Val([Введите минимальное число просроченных дней]).

При выполнении запросов обратите внимание на ошибки, появляющиеся в числовых полях до выполнения указанных рекомендаций.

Отчёты в базе данных Access

Общие сведения об отчётах

В базе данных Microsoft Access с помощью отчётов можно получить наиболее полное представление о данных, содержащихся в таблицах и запросах, можно более удобно использовать и наглядно отображать данные, а также есть возможность лучше просматривать, форматировать и группировать информацию.

Отчёт – это объект базы данных, гибкое средство удобное для организации данных и представления информации при выводе её на печать с одной из следующих целей:

- отображение или распространение сводки данных;
- архивирование моментальных снимков данных;
- предоставление сведений об отдельных записях;
- создание почтовых наклеек.

Средства Access позволяют создавать отчёты любой сложности, обеспечивающие вывод взаимосвязанных данных из многих таблиц, их группировку, вычисление итоговых значений, фильтрацию и анализ данных.

Создание отчётов можно выполнять следующими способами:

- автоматически на основе данных таблицы или запроса;
- с помощью мастера;
- в режиме конструктора.

Во многих случаях удобно использовать мастер отчётов. Мастер выполняет всю рутинную работу и позволяет быстро разработать отчёт. После вызова мастера выводятся окна диалога с приглашением ввести необходимые данные, и отчёт создаётся на основании ответов пользователя. Мастер окажется полезным даже для опытных пользователей, так как позволяет быстро разработать макет, служащий основой создаваемого отчёта. Созданный мастером отчёт можно доработать в режиме конструктора.

При необходимости вывода в отчёте результатов решения задачи в качестве основы для отчёта может быть использован многотабличный запрос, представляющий эти результаты. Запрос может выполнять наиболее сложные виды выборки и предварительной обработки данных. Возможности конструктора отчётов позволяют легко структурировать и оформить данные, полученные в запросе. Access позволяет создать профессионально оформленные отчёты не только с помощью мастера или конструктора, но и в режиме макета.

Отчёты могут быть открыты в четырёх режимах: *Конструктор*, *Предварительный просмотр*, *Представление отчёта* и *Режим макета*. Переключение между режимами отчёта производится аналогично переключению между режимами любого другого объекта баз данных: выбором соответствующей команды в контекстном меню открытого отчёта или с использова-

нием кнопки *Режим*, расположенной слева на вкладке ленты *Главная* (рис 3.81).

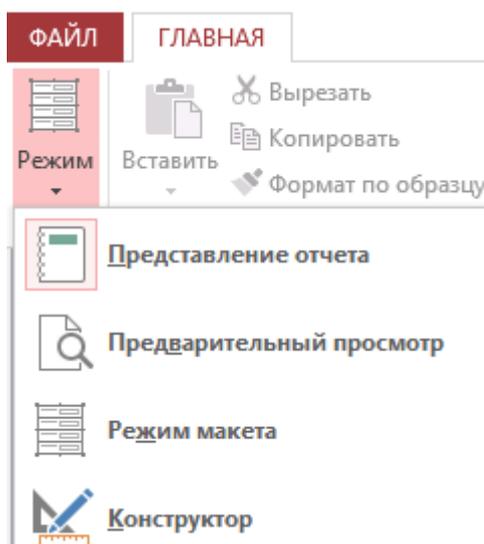


Рис. 3.81. Команды кнопки *Режим* для переключения между режимами отчёта

Режим *конструктора* используют для создания отчёта или изменения структуры существующего отчёта.

Режим макета позволяет, просматривая данные отчёта, изменять его макет.

Режим *Предварительный просмотр* позволяет просматривать данные в том виде, в котором они будут размещаться на печатных страницах. В этом режиме можно увеличивать масштаб для просмотра деталей или уменьшать его для проверки размещения данных на странице, изменять параметры страницы.

В режиме *Представление отчёта* выводятся основные элементы макета отчёта с образцами данных, демонстрирующих представление данных в отчёте. В данном режиме можно фильтровать данные для отображения только заданных строк, найти нужные данные, скопировать в буфер обмена часть или весь текст отчёта. При открытии отчёта программа Access отображает в нем самые последние данные из источника записей.

Перед началом конструирования пользователь должен спроектировать макет отчёта. Макет отчёта разбит на разделы, которые можно просмотреть в режиме конструктора. Понимание принципов работы каждого раздела помогает создавать хорошие отчёты.

Ниже приведены типы разделов отчёта и указано назначение каждого из них (рис.3.82).

Заголовок отчёта отображается при печати в начале отчёта. В заголовке включается информация, обычно помещаемая на обложке, например эмблема компании, название отчёта или дата. Если в заголовке отчёта помещён

вычисляемый элемент управления, использующий статистическую функцию Sum, сумма рассчитывается для всего отчёта. Заголовок печатается перед верхним колонтитулом только один раз в начале отчёта.

↙ Заголовок отчета					
↙ Верхний колонтитул					
↙ Заголовок группы 'Фамилия'					
↙ Область данных					
↙ Примечание группы 'Фамилия'					
↙ Нижний колонтитул					
↙ Примечание отчета					

Рис. 3.82. Разделы отчёта в режиме конструктора

Верхний колонтитул размещается вверху каждой страницы и используется в тех случаях, когда нужно, чтобы название отчёта повторялось на каждой странице.

Заголовок группы отображается в начале каждой новой группы записей. Используется для печати названия группы при группировке записей отчёта. Если поместить в заголовок группы вычисляемый элемент управления, использующий статистическую функцию *Sum*, сумма будет рассчитываться для текущей группы. Заголовок группы может состоять из нескольких разделов в зависимости от добавленных уровней группирования.

Область данных отображается один раз для каждой строки в источнике записей. В нем размещаются элементы управления, составляющие основное содержание отчёта.

Примечание группы размещается в конце каждой группы записей и можно использовать для печати сводной информации по группе. Нижний колонтитул группы может состоять из нескольких разделов в зависимости от добавленных уровней группирования.

Нижний колонтитул отображается внизу каждой страницы и используется для нумерации страниц, а также для печати постраничной информации.

Примечание отчёта размещается в конце отчёта и можно использовать для печати итогов и другой сводной информации по всему отчёту.

Следует заметить, что в режиме конструктора *примечание отчёта* отображается под нижним колонтитулом. Однако во всех остальных режимах (например, в режиме макета, при печати или предварительном просмотре) этот раздел помещается над нижним колонтитулом, сразу после примечания последней группы или строки области данных на последней странице.

Перед созданием отчёта пользователь должен определить состав и содержание разделов отчёта, размещение в нём значений, выводимых из полей таблиц или запросов базы данных, и вычисляемых реквизитов, определить поля, по которым нужно группировать данные. При этом для каждого уровня группировки нужно определить заголовки и примечания, вычисляемые итоговые значения, а также порядок вывода данных в отчёте.

Создание отчётов

Для создания отчёта следует выполнить три действия: выбрать источник записей, выбрать инструмент создания отчёта и непосредственно создать отчёт.

Источник записей должен содержать все строки и столбцы данных, которые требуется отобразить в отчёте. Источником записей для отчёта может быть таблица или запрос. Если нужные данные содержатся в существующей таблице или запросе, следует выделить эту таблицу или запрос в области навигации и перейти к выбору инструмента отчёта.

Инструменты отчёта расположены на вкладке *Создать* в группе *Отчёты* (рис.3.83). Ниже описаны эти инструменты отчёта.

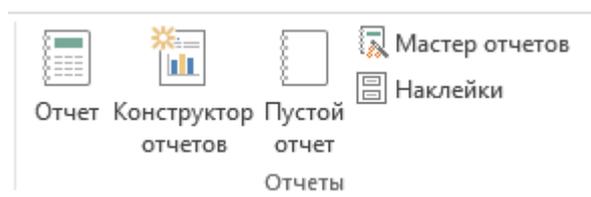


Рис. 3.83. Группа *Отчёты* вкладки *Создать*

Отчёт позволяет создать простой табличный отчёт, содержащий все поля из источника записей, который выбран в области навигации.

Конструктор отчётов открывает в режиме конструктора пустой отчёт, в который можно добавить необходимые поля и элементы управления.

Пустой отчёт позволяет открыть пустой отчёт в режиме макета и отобразить область задач «Список полей», из которой можно добавить поля в отчёт.

Мастер отчётов служит для вызова пошагового мастера, с помощью которого можно задать поля, уровни группировки и сортировки и параметры макета.

Наклейки вызывает мастер, в котором можно выбрать стандартный или настраиваемый размер подписей, набор отображаемых полей и порядок их сортировки.

Для непосредственного создания отчёта следует выбрать требуемый инструмент нажатием соответствующей кнопки в группе *Отчёты*. После появления мастера надо следовать всем его командам и на последнем шаге нажать кнопку *Готово*.

Access отображает отчёт в режиме макета. Далее, чтобы добиться желаемого внешнего вида, можно отформатировать отчёт, выполняя следующие операции: изменять размер полей и подписей, выделяя их и перетаскивая края; располагать поля в нужном порядке, выделяя и перетаскивая их в нужное место; объединять или разбивать ячейки, удалять и выделять поля и выполнять другие задачи форматирования.

Кроме того, отчёт можно сделать более привлекательным и удобным с помощью различных функций: добавлять в отчёт группировку, сортировку или итоги; выделять данные с помощью условного форматирования; настраивать цвета и шрифты; добавлять в отчёт логотип или фоновый рисунок; применять фильтры для выборки необходимых данных; настраивать параметры печати отчётов.

Создание отчётов рассмотрено на примерах, приведённых ниже.

Пример

Задание. В учебной базе данных *Борей* построить отчёт, позволяющий вывести список десяти самых дорогих товаров, а также их среднюю и суммарную цену.

1. Импортируйте из учебной базы данных *Борей* запрос «Товары с ценой выше средней» (*Внешние данные – Access – Обзор – из соответствующей папки, указанной преподавателем – ... Борей – Импорт... – Запросы*).
2. В качестве источника данных в области переходов (навигации) выделите запрос «Товары с ценой выше средней» и щёлкните по кнопке «Отчёты» на вкладке *Создать* в группе *Отчёты* (см. рис. 3.83). Заметим, что это наиболее простой способ создания отчёта на основе таблицы или запроса, потому что при его использовании отчёт формируется сразу же, без запроса дополнительной информации.
3. Без диалога с пользователем создаётся простой табличный отчёт, содержащий все поля из выбранного источника записей (запроса), который отображается в режиме макета (рис. 3.84). Хотя получившийся отчёт далёк от совершенства, он позволяет быстро просмотреть базовые данные. После этого отчёт можно сохранить и изменить в режиме макета или конструктора в соответствии с поставленной задачей.

Разделы отчёта и размещение полей таблицы – источника записей (запрос *Товары с ценой выше средней*) в режиме конструктора представлены на рис. 3.85.

Товары с ценой выше средней	
Марка	Цена
Alice Mutton	1 755,00€
Carnarvon Tigers	2 812,50€
Sir Rodney's Marmalade	3 645,00€
Gumbar Gummibarchen	1 405,35€
Schoggi Schokolade	1 975,50€
Uncle Bob's Organic Dried Pears	1 350,00€
Northwoods Cranberry Sauce	1 800,00€
Mishi Kobe Niku	4 365,00€
Ikura	1 395,00€
Queso Manchego La Pastora	1 710,00€
	63 665,65€

Страница 1 из 1

Рис. 3.84. Простой отчёт, созданный командой *Отчёт*, в режиме макета

Товары с ценой выше средней	
Марка	Цена
Марка	Цена
Марка	Цена
:"Страница " & [Page] & " из " & [Pages]	
=Sum([Цена])	

Рис. 3.85. Отчёт, созданный на базе запроса *Товары с ценой выше средней* в режиме конструктора

В заголовке отчёта размещены: эмблема, название отчёта, которое совпадает с именем источника данных (*Товары с ценой выше средней*), встроенные функции: =Date() – текущая дата и =Time() – текущее время.

В верхнем колонтитуле размещены заголовки полей (столбцов) *Марка* и *Цена*, что обеспечивает вывод их на каждой странице отчёта.

В области данных отчёта размещены все поля таблицы источника данных. Это обеспечивает вывод в отчёте в табличном виде всех записей.

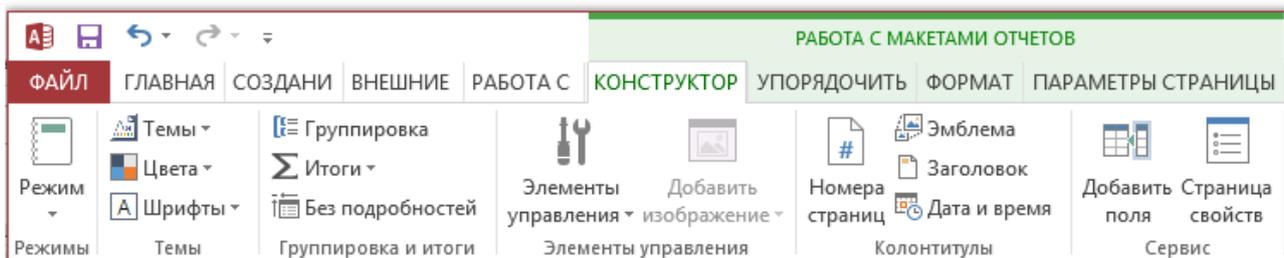
В нижнем колонтитуле для добавления номера страницы сформировано выражение: ="Страница " & [Page] & " из " & [Pages].

В *примечании отчёта* размещено выражение, в котором используется статистическая функция =Sum([Цена]), рассчитывающая общую стоимость всех товаров, представленных в отчёте.

Доработка отчёта

Доработка отчёта в режиме макета

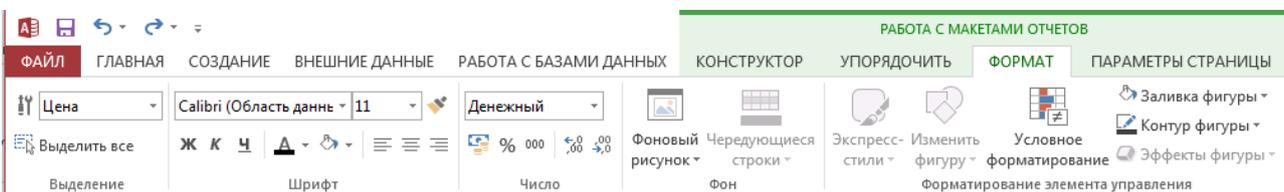
Доработку отчёта можно проводить с помощью инструментов вкладок *Работа с макетами отчётов / Конструктор (Упорядочить, Формат, Параметры страницы)* – рис 3.86.



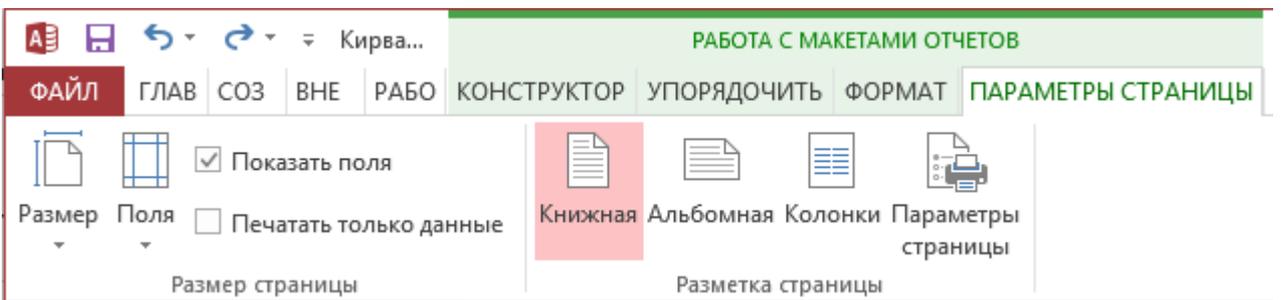
а) вкладка *Конструктор*



б) вкладка *Упорядочить*



в) вкладка *Формат*



в) вкладка *Параметры страницы*

Рис. 3.86. Вкладки *Работа с макетами отчётов*

В режиме макета в отчёте, можно отрегулировать ширину столбцов, высоту строк, изменить размер и цвет шрифта. Можно добавлять в отчёт но-

вые поля, изменять их порядок, а также изменять свойства отчёта и входящих в него элементов управления.

Задание 1. Исправить названия полей, ввести элементы оформления, добавить поля расчёта среднего и суммарного значений.

1. Измените название отчёта. Для этого выполните двойной щелчок на нем и введите новое название – *Товары с высокой ценой*.
2. Замените эмблему. Для этого щёлкнув на старую *Авто_Эмблема* и далее, выбрав на вкладке ленты *Конструктор* команду *Эмблема* откройте нужный файл или выбрав команду *Добавить изображение / Обзор* выберите на компьютере нужный рисунок (можно предварительно найти подходящую картинку на просторах интернета). Рисунок будет добавлена в то же место заголовка отчёта. Эмблему можно перетащить в другое место и изменить её размер. Затем откройте *Окно свойств* выделенного элемента *Рисунок* (в контекстном меню выбрав команду *Свойства* или нажав кнопку *Страница свойств*) и параметре *Установка размеров* выберите опцию *По размеру рамки*.
3. Перетащите поле для вывода страниц отчёта вправо, установите *ширину поля – 3,5 см, задайте зелёный цвет шрифта на жёлтом фоне, тип границы – сплошная, цвет красный, ширина границы – 1 пункт*.
4. Поля для вывода текущих даты и времени перетащите в нижний колонтитул левее поля для вывода страниц и установите *средний формат даты и времени, шрифт Arial, 12 pt, не курсив, насыщенность – плотный, цвет красный*.
5. Выберите тему отчёта *Грань* (на вкладке *Работа с макетами отчётов / Конструктор* группа *Тема*).

Доработка отчёта в конструкторе

1. Перейдите в режим *Конструктор*.
2. Установите ширину и высоту полей текущих даты и времени оптимальной: не больше и не меньше выводимых данных. Переключайтесь в режим предварительного просмотра и просматривайте результаты выполнения задания.
3. Выровняйте поля в нижнем колонтитуле по нижнему краю, выделив их (держат клавишу *Shift*) и применив на вкладке *Упорядочить* в группе *Размер и порядок* команду *Выровнять / Снизу*.
4. Установите в нижнем колонтитуле по верхнему краю *штриховую* линию длиной 17 см, ширину границы 2 пункта и цвет – *синий*.
5. Установите книжную ориентацию бумаги. Формат А4. Поля обычные.
6. Расширьте область *Примечание отчёта*, щёлкните кнопку *Поле* (пиктограмма с буквами *ab*) в группе *Элементы управления* на вкладке *Инструменты конструктора отчётов / Конструктор*, затем щёлкните в разделе *Примечание отчёта*. Двойным щелчком по полю выведите диалоговое

окно свойств. В параметре *Данные* постройте выражение =Avg([Цена]) для вычисления *среднего значения цен товаров*. Установите *денежный формат* с двумя знаками после запятой. Наберите подпись имени поля *Среднее значение цен* и установите приемлемую ширину поля.

7. Оформите поле для вычисления *суммарного значения цен* только десяти приведённых товаров (в источнике данных – запросе, установите *Набор значений – 10*). Добавьте надпись данного поля (кнопка *Aa* в группе *Элементы управления*).
8. Заключите поля со средним значением и суммарным значением цен в прямоугольник (тип границы – *точечная*, ширина границы – *4 пункта*, цвет границы – *тёмно-бордовый*).
9. Последовательно перейдите сначала в режим *Предварительный просмотр*, а затем в режим *Представление отчёта*. Обратите внимание на отличие в представлении данных.
10. В режиме конструктора переместите рамку со всем содержимым из области примечания отчёта в область данных. Перейдите в режим предварительного просмотра и обратите внимание на изменения в отчёте. Перетащите рамку с содержимым обратно в область примечания отчёта.
11. Добавьте слева столбец и пронумеруйте в нем имеющиеся в отчёте товары. Для этого правее поля *Цена* добавьте вычисляемое поле и в окне свойств на вкладке *Данные* в одноимённом параметре постройте выражение: =1, далее установите для параметра *Сумма с накоплением* опцию *Для всего*. *Тип границы данного поля* выберите *Отсутствует*. Поле *Надпись* удалите. Сдвиньте поля *Марка* и *Цена* вправо так, чтобы поле с нумерацией установить по левому краю. В режиме *Представление отчёта* просмотрите результаты организованной нумерации товаров.

Работа с элементами управления при модернизации отчётов

Задание 2. Модернизировать отчёт *Товары с ценой выше средней* следующим образом:

- вставить столбец с *изображением* каждой марки товаров;
- вставить поле, сообщающее, прекращены или нет поставки данного товара;
- расположить каждую марку товара на отдельной странице;
- расположить заголовок отчёта на каждой его странице.

1. Добавьте в бланк запроса *Товары с ценой выше средней* поле *Изображение* из таблицы *Типы* и сохраните запрос.
2. Для вставки поля рисунка в отчёт следует выбирать не элемент *Поле* из группы *элементы управления* вкладки *Конструктор*, а элемент *Присоединённая рамка объекта*. Щёлкните в группе *элементы управления* по кнопке *Присоединённая рамка объекта* и поместите её в область данных. Рам-

- ку для рисунка можно пока не изменять. Надпись данного поля именуется *Изображение* и переместите в область верхнего колонтитула (если не получается выполнить перетаскивание, воспользуйтесь буфером обмена. В крайнем случае, можно удалить надпись поля и вставить в верхний колонтитул надпись из элементов управления).
3. Щёлкните дважды по вставленному в область данных объекту для вывода диалогового окна свойств. В строке *Данные* вкладки *Данные* щёлкните по выпадающему списку полей и выберите поле *Изображение*. Если данного поля в списке нет, см. п.1.
 4. Перейдите в режим предварительного просмотра и убедитесь в том, что для каждой модели вставляется её изображение. Если изображение обрезаётся, т.к. рамка слишком мала, вызовите диалоговое окно свойств поля *Изображение* и в строке *Установка размеров* вкладки *Макет* выберите поочерёдно опции *По размеру рамки* и *Вписать в рамку*. Проанализируйте эти два варианта представления изображений переходом в режим предварительного просмотра. Остановитесь на понравившемся варианте. Установите рамку рисунка шириной 4 см и выберите приподнятое оформление.
 5. Далее выполните вставку поля, в котором выводится информация о том, прекращены или нет поставки данного товара. Для этого добавьте в запрос соответствующее поле и вставьте в отчёт *Флажок* из элементов управления. В окне свойств данного элемента в строке *Данные* из выпадающего списка выберите необходимое поле. Надпись поля *Флажок* именуется *Поставки прекращены* и переместите в область верхнего колонтитула. Вставьте горизонтальную линию (сплошную, темно-синюю, толщиной в 2 пункта), разделяющую заголовки столбцов от данных отчёта. Не забывайте устанавливать заголовки полей в области верхнего колонтитула, соответствующие полям для *данных* и корректировать, при необходимости, длину разделительных линий отчёта.
 6. Вставьте вертикальные линии, разделяющие данные отчёта. Сделайте их *синими, штриховыми, толщиной в 2 пункта*.
 7. Просмотрите, сколько страниц занимает отчёт при выбранном оформлении. Выполните действия, позволяющие расположить каждую модель на отдельной странице. Для этого вставьте из элементов управления элемент *Разрыв страницы* в нижний край области данных. Перейдите в режим предварительного просмотра и убедитесь в правильности установки элемента *Разрыв страницы*.
 8. Выполните действия, позволяющие расположить заголовок отчёта на каждой его странице. Для этого переместите заголовок в область *верхнего колонтитула*. Задайте для него цвет фона *Система – градиент активного заголовка*.

Отчёт *Товары с ценой выше средней* в режиме конструктора представлен на рис. 3.87.

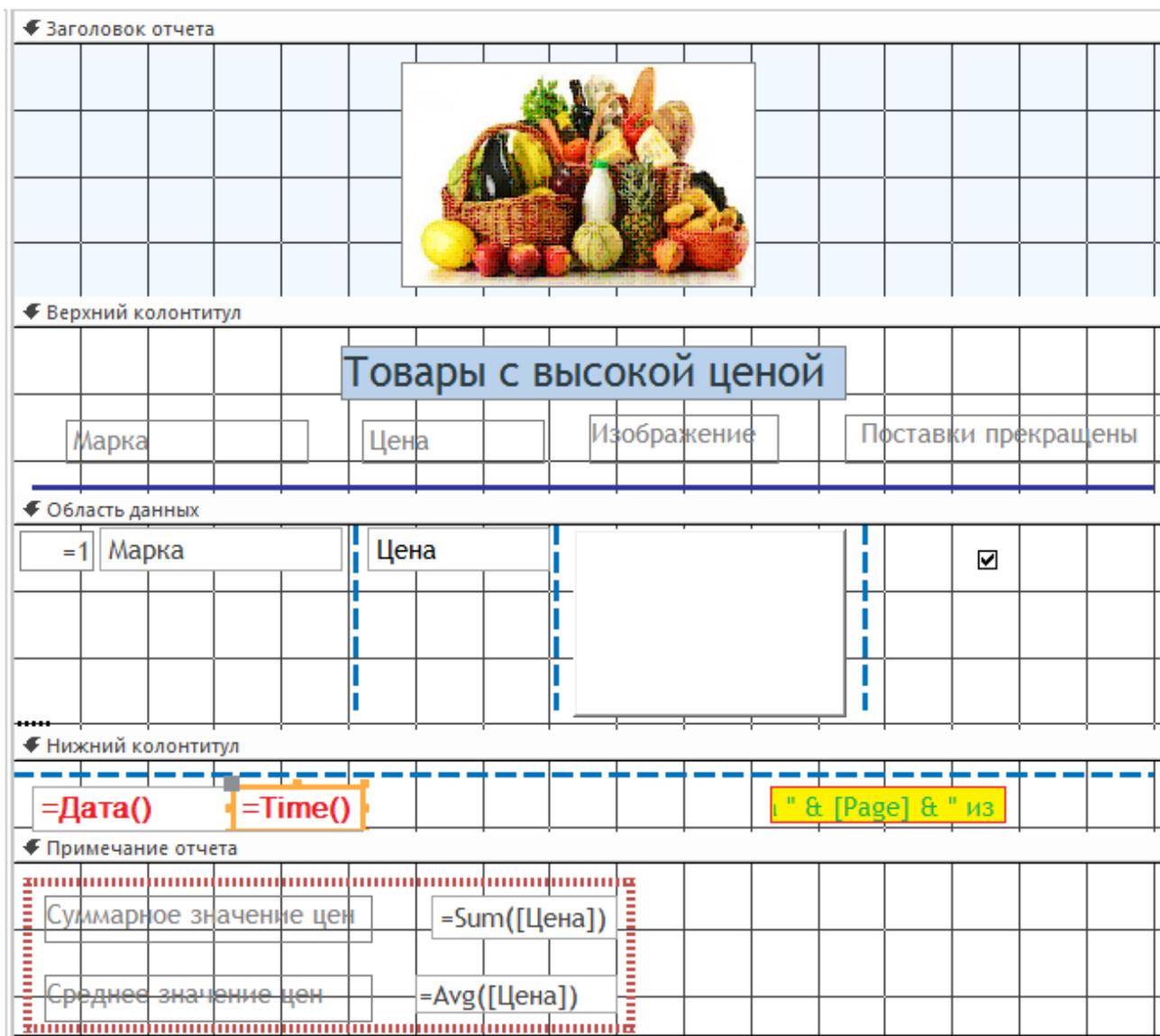


Рис. 3.87. Отчёт *Товары с ценой выше средней* в режиме конструктора

Группировка и сортировка данных в отчётах

Данные проще анализировать, если они разделены на группы. Поместив итоговые значения (например, суммы или средние значения) в конце каждой группы своего отчёта, можно автоматизировать много операций.

Задание. В базе данных Борей: 1) – посчитать для каждого сотрудника суммарную стоимость заказов, размещённых каждым клиентом; 2) – рассчитать для каждого сотрудника итоговые суммы заказов всех клиентов.

1. Первую задачу можно решить построением итогового запроса. Из схемы данных базы *Борей* (см. рис. 3. 77) можно понять, что в бланк запроса необходимо добавить таблицы *Сотрудники*, *Клиенты* и *Заказано*, а также связующую их таблицу *Заказы*. Из них следует включить в запрос поля *Фамилия* сотрудника и *Название* клиента. Затем надо построить выражение, рассчитывающее стоимость заказов: $Цена * Количество * (1 - Скидка)$. Далее включить операцию *Итоги* и выбрать функцию *Sum* для построенного выражения. Выполните запрос и убедитесь, что производится вывод информации о суммарной стоимости заказов, размещённых каждым клиентом для каждого сотрудника.
2. Построив этот запрос, сохраните его с именем *Запрос для отчёта с группировкой*.
3. Технология запросов Access не даёт возможности решения следующей задачи, так как нельзя сделать одновременный вывод нужных данных в одном запросе. Проблема состоит в том, что помимо указанных сумм в первой задаче, надо рассчитать итоговые суммы заказов для каждого сотрудника. Поэтому следует далее задействовать технологию отчётов.
4. На основе сохранённого запроса создайте простой отчёт. Посмотрите, как называется в отчёте вычисляемое поле, и какой оно имеет формат. Если в этом поле неадекватно отражены денежные суммы или надпись поля называется *Выражение 1*, измените формат и название вычисляемого поля. А лучше закройте построенный отчёт без сохранения результатов, откройте запрос и исправьте ошибки, а затем вновь постройте отчёт по данному запросу.
5. Откройте отчёт в режиме конструктора. Заголовок отчёта измените на *Стоимость заказов*. В верхнем колонтитуле измените поле *Фамилия* на *Фамилия сотрудника* и поле *Название* на *Название клиента*. Установите для всех полей, кроме заголовка *Стиль линий сетки справа* – *Сплошная*, для остальных линий сетки – *Отсутствует*.
Далее необходимо осуществить группировку и сортировку данных, отображаемых в отчёте.
6. Для этого предназначена область *Группировка, сортировка и итоги* (рис. 3.88), которая отображается если на вкладке *Инструменты конструктора отчётов / Конструктор* в группе *Группировка и итоги* нажать кнопку *Группировка*. Заметим, что в созданном отчёте нет пока никакой группировки и сортировки. Отображаемая область содержит только кнопки *Добавить группировку* и *Добавить сортировку*.

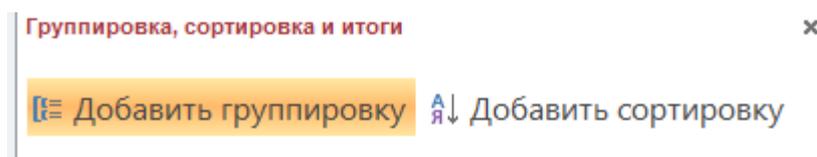


Рис. 3.88. Область отчёта *Группировка, сортировка и итоги*

7. Далее щёлкните на кнопке *Добавить группировку* и появившейся строке *Группировка* (рис. 3.89) выберите из списка поле *Фамилия*.

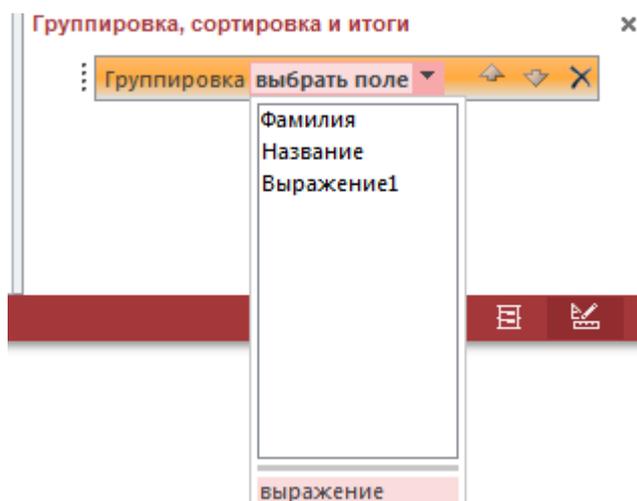


Рис. 3.89. Окно отчёта для выбора поля группировки

В отчёте добавится область *Заголовок группы* 'Фамилия', а в области *Группировка, сортировка и итоги* – строка первого (из десяти возможных) уровня группировки и сортировки по полю *Фамилия* (рис. 3.90).

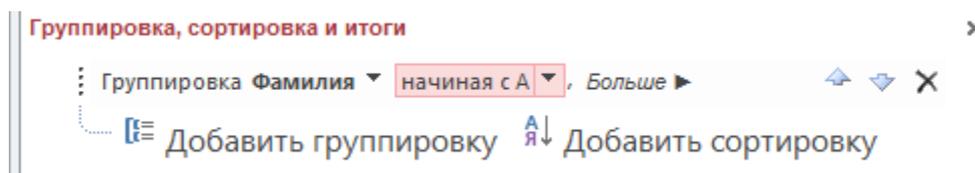


Рис. 3.90. Строка *Группировка* по полю *Фамилия*

8. В этой строке нажмите кнопку *Большее* для установки дополнительных параметров. Установите опцию *с разделом примечания* (обратите внимание, что появился новый раздел *Примечание группы* 'Фамилия'). Чтобы на странице размещались данные об одном сотруднике установите опцию *удерживать группу на одной странице*.

Затем в поле списка *Итог по полю* выберите поле *Выражение 1* (Суммарная стоимость заказов) по которому нужно рассчитать итоговые значения (итоговая денежная сумма для каждого сотрудника), а из списка *Тип* выберите *Сумма*. Для отражения общего итога денежных сумм всех сотрудников отметьте флажок *Показать общий итог*, а для определения места отображения итоговых значений по каждому сотруднику выберите флажок *Показать промежуточный итог в примечании группы* и для отображения доли итоговых денежных сумм сотрудника от общего итога всех сотрудников выберите флажок *Показать промежуточный итог по группе как % от общего итога* (рис. 3.91).

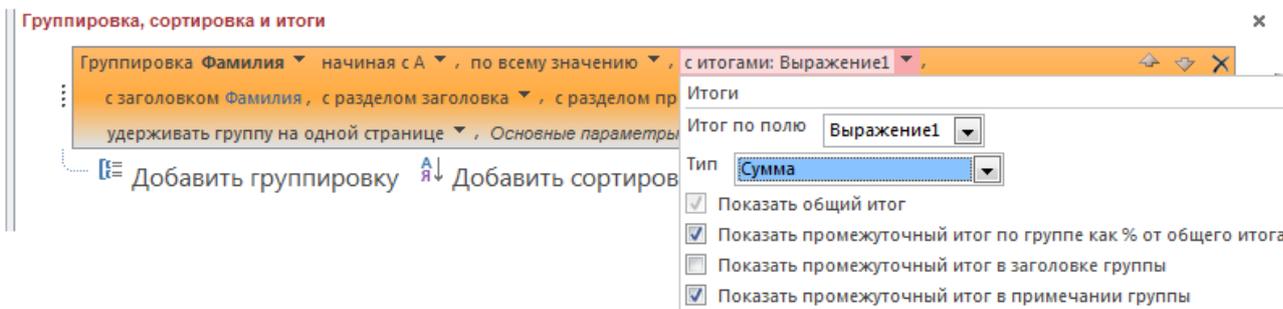


Рис. 3.91. Строка группировки при нажатой кнопки *Больше* с открытым списком параметра *с итогами: Выражение1*

Итоги можно подвести для нескольких полей группы. Для этого нужно выбирать каждое поле в раскрывающемся списке *Итог по полю* и требуемые параметры.

9. Перейдите в режим предварительного просмотра и просмотрите результаты введения группировки.
10. Установите книжную ориентацию отчёта, предварительно уплотнив, если необходимо, данные в нижнем колонтитуле.
11. Установите тонкую (1 пункт), синюю разделительную линию после каждой записи отчёта.
12. В области *Примечание группы* в поле, в котором вычисляется итоговая денежная сумма для каждого сотрудника установите соответствующий формат поля. Отделите это поле от основных данных так, чтобы при просмотре оно бросалось в глаза.
13. Отделите нижний колонтитул от данных отчёта линией сверху. Нижнюю линию удалите.
14. Установите краткий формат даты и времени.
15. Чтобы фамилия сотрудника не повторялась многократно необходимо поместить поле *Фамилия* в область заголовка группы, для этого:
 - выберите на ленте команду *Группировка*;
 - установите опцию *с разделом заголовка*;
 - перенесите фамилию сотрудника из области данных в область заголовка группы;
 - установите для заголовка группы более крупный шрифт, другой цвет и жирное начертание.
16. Произведите сортировку записей в группе в порядке убывания денежных сумм. Для этого вновь воспользуйтесь диалоговым окном *Группировка, сортировка и итоги*. Во второй строке (рис. 3.92) выберите название, данное вычисляемому полю (*Выражение1*), и установите сортировку от максимального к минимальному. Заголовок и примечание группы при этом не добавляйте.

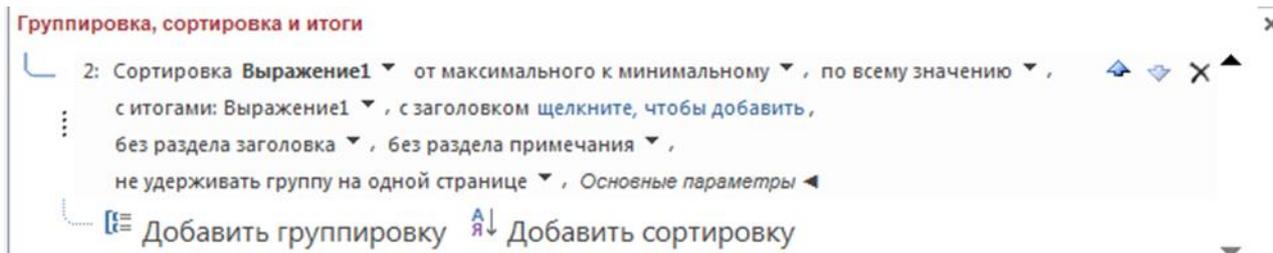


Рис. 3.92. Строка сортировки по полю *Выражение1*
(*Суммарная стоимость заказов клиента*)

17. Пронумеруйте записи отчёта с началом нумерации отдельно для каждого сотрудника.
18. Сохраните отредактированный отчёт (рис.3.93) с именем «Отчёт с группировкой-Стоимость заказов» и закройте его.

Стоимость заказов		22.04.2017
		20:58
Фамилия сотрудника	Название клиента	Суммарная стоимость заказов клиента
Акбаев		
1	Save-a-lot Markets	131 274,40€
2	Hungry Owl All-Night Grocers	75 580,00€
3	Ernst Handel	43 768,80€
29	Furia Bacalhau e Frutos do Mar	578,00€
30	Cactus Comidas para llevar	125,00€
Итоговая денежная сумма сотрудника		915 894,41€
Доля итоговой денежной суммы		7,13%
Общий итог денежных сумм сотрудников		12 841 739,18€

Рис. 3. 93. Фрагменты отчёта с группировкой *Стоимость заказов*

Редактирование источника записей отчёта

Источника записей отчёта можно изменить в режиме макета или конструктора добавив в отчёт поля из списка всех таблиц базы данных. Например необходимо в созданном выше отчёте отобразить не только фамилию, но имя сотрудника.

1. Откройте «Отчёт с группировкой-Стоимость заказов» в режиме макета. На вкладке ленты *Конструктор* в группе *Сервис* нажмите кнопку *Добавить поля*. В области *Список полей* будут отображены только поля, доступные в открытом отчёте. Чтобы отобразить другие поля надо щёлкнуть в верхней части области на строке *Показать все таблицы*. В открывшемся списке чтобы отобразить список полей нужной таблицы (в нашем случае таблица *Сотрудники*), следует щёлкнуть знак (+) рядом с её именем (рис. 3.94).

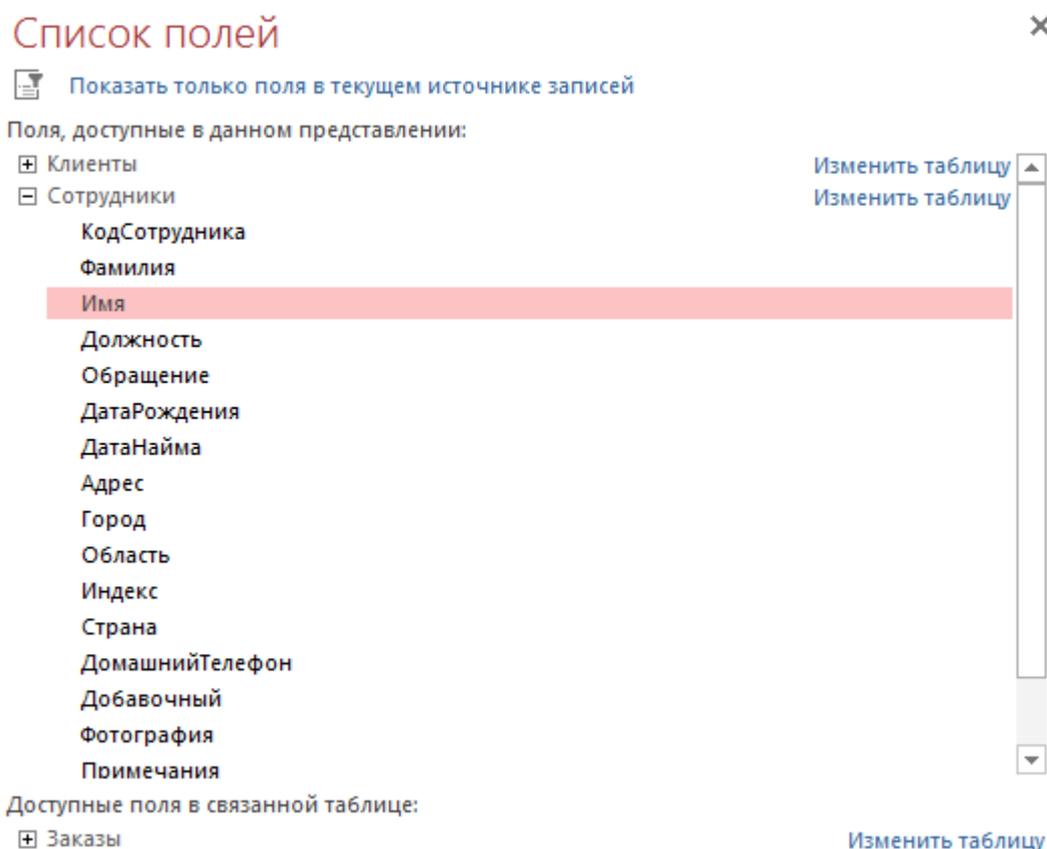


Рис 3.94. Область *Список полей* отчёта с отображением полей таблицы *Сотрудники*

2. Добавьте поле *Имя* к отчёту перетащив его из области *Список полей* в нужную область отчёта – в заголовок группы *Фамилия*. Чтобы поле *Имя* попало в заголовок группы *Фамилия*, поместите его справа от значения поля *Фамилия* (Акбаев) или над ним (рис. 3.95).

Стоимость заказов		22.04.2017
		21:19
Фамилия сотрудника	Название клиента	Суммарная стоимость заказов клиента
Акбаев		
1	Save-a-lot Markets	131 274,40€
2	Hungry Owl All-Night Grocers	75 580,00€
3	Ernst Handel	43 768,80€

Рис. 3.95. Добавление поля в отчёт в режиме макета

- После перемещения поля в заголовке группы отчёта создаются элемент управления и соответствующая подпись, связанные с полем. При этом автоматически изменяется источник записей отчёта. Надпись *Имя* нужно удалить, а в верхнем колонтитуле вместо поля *Фамилия сотрудника* введите *Фамилия и имя сотрудника*. Щёлкнув на кнопке *Страница свойств* вкладки ленты *Конструктор* и далее в окне свойств на вкладке *Данные* в строке *Источник записей* можно просмотреть добавленное в источник записей поле *Имя* таблицы *Сотрудники*.

Применение фильтрации для отображения отдельных данных

В режиме отчёта можно применять фильтры к данным. Например, в отчёте есть столбец *Название клиента* (см. рис.3.96) и требуется отобразить для всех сотрудников только строки, содержащие название клиента *Save-a-lot Markets*. Для этого, в отчёте найдите название клиента *Save-a-lot Markets* и щёлкните его правой кнопкой мыши. В контекстном меню выберите команду *Равно "Save-a-lot Markets"*. После этого будет создан и применится фильтр.

Стоимость заказов		22.04.2017
		22:01
Фамилия и имя сотрудника	Название клиента	Суммарная стоимость заказов клиента
Акбаев Иван		
1	Save-a-lot Markets	131 274,40€
2	Hungry Owl All-Night Grocers	75 580,00€

Рис. 3.96. Верхняя часть первой страницы отчёта в режиме предварительного просмотра с добавленным полем *Имя* сотрудника

Далее переключаться между режимами отображения с применением и без применения фильтров можно на вкладке *Главная* в группе *Сортировка и фильтр* с помощью кнопки *Применить (Удалить) фильтр*. При этом происходит лишь его включение и отключение, а фильтр не удаляется. Для его удаления следует в контекстном меню выбрать строку, с которой надо снять фильтр (в нашем случае *Save-a-lot Markets*) и выбрать команду *Снять фильтр с «имя поля»* (в нашем примере *имя поля* будет *Название клиента*). Если фильтр удалён, его нельзя включить с помощью кнопки *Применить (Удалить) фильтр*. Потребуется заново создать фильтр.

Если применить к отчёту фильтр, а затем сохранить и закрыть отчёт, фильтр также будет сохранён. Однако при следующем открытии отчёта фильтр не будет применён. Чтобы повторно применить данный фильтр, на вкладке *Главная* в группе *Сортировка и фильтр* следует нажать кнопку *Применить фильтр*.

Разработка отчёта с помощью мастера

Access упрощает работу над отчётами с группировкой. Можно создать простой отчёт с группировкой при помощи мастера отчётов, добавить группировку или сортировку в уже созданный отчёт, а также изменить уже заданные параметры группировки или сортировки. Мастер создаёт отчёт на основе ответов пользователя на заданные вопросы. После создания отчёта можно использовать его в готовом виде или изменить.

Задание. Построить предыдущий отчёт с использованием мастера отчётов.

1. Для создания отчёта мастером выберите в области навигации *Запрос для отчёта с группировкой*, необходимый для формирования источника записей отчёта. Выбранный запрос определит имя и заголовок отчёта по умолчанию, а также вид представления данных в отчёте.
2. На вкладке ленты *Создание* в группе *Отчёты* нажмите кнопку *Мастер отчётов*. Следуйте указаниям на страницах мастера отчётов.
3. В первом открывшемся окне мастера *Создание отчётов* (рис. 3.97) в группе *Таблицы и запросы* отобразится запрос: *Запрос для отчёта с группировкой* и *Доступные поля* его. Отсюда необходимо выбрать поля этого запроса, которые будут включены в отчёт. Выберите все поля.

Примечание: Если нужно добавить в отчёт поля из нескольких таблиц или запросов, не нажимайте в мастере отчётов кнопку *Далее* или *Готово* после выбора полей из первой таблицы или запроса. Повторите процедуру выбора для другой таблицы или запроса, а также выберите щелчком мыши другие поля, которые вы хотите добавить в отчёт. Затем только для продолжения нажмите кнопку *Далее* или *Готово*.

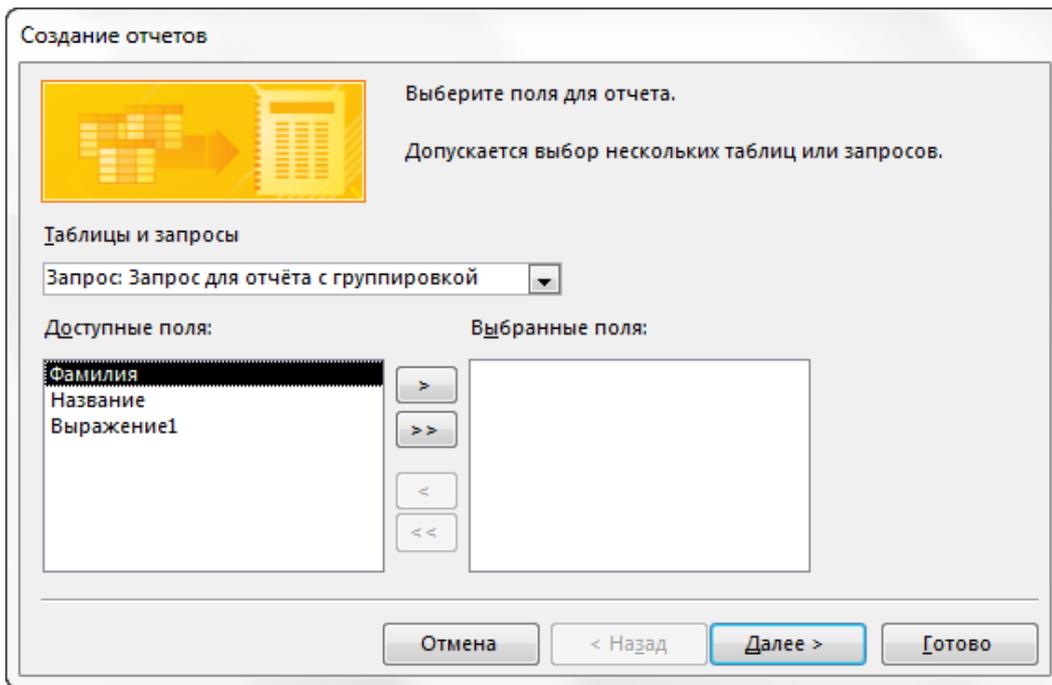


Рис. 3.97. Первый шаг мастера создания отчётов: Выбор полей отчёта

Далее важно не пропустить группировку, сортировку данных, а также итоговые операции.

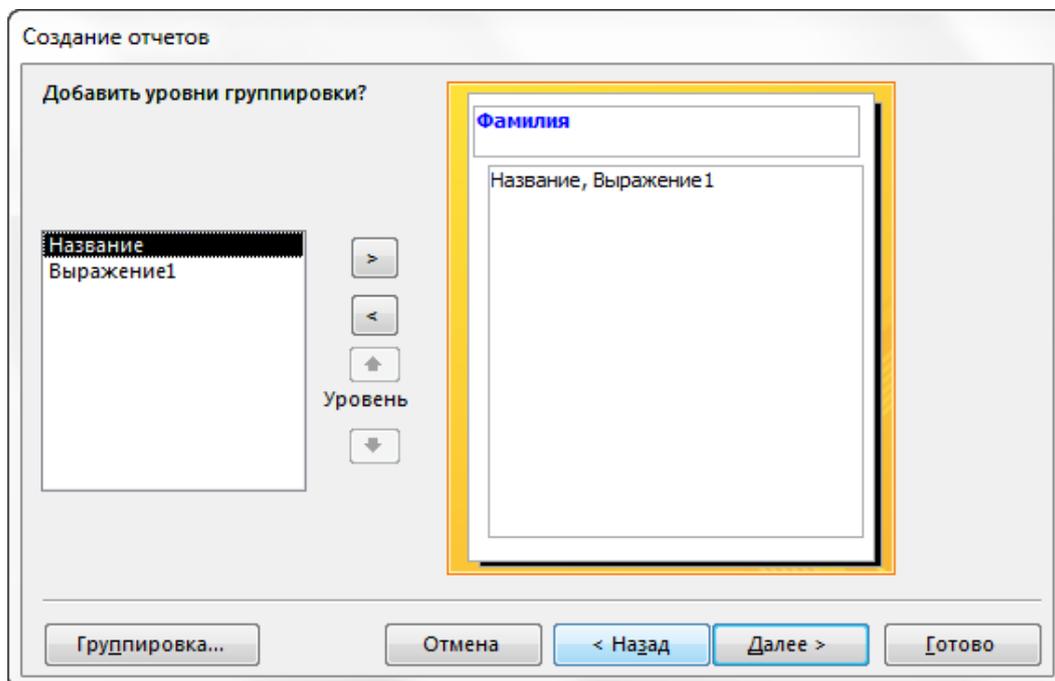


Рис. 3.98. Второй шаг мастера создания отчётов: Добавить уровни группировки

4. На втором шаге мастера (рис.3 98) добавляются уровни. Пользователь может выбрать несколько уровней группировки данных, максимальное количество которых четыре. Первое группируемое поле будет наиболее важным уровнем, а остальные следуют за ним в порядке выбора. В нашем случае для добавления имени поля из списка выберите поле *Фамилия* и нажмите кнопку  или сделайте двойной щелчок на выбранном поле, а затем кнопку *Группировка...*
5. В следующем окне мастера (рис.3 99) для каждого выбранного поля можно задать нужный интервал группировки. В нашем примере для поля группировки *Фамилия* оставьте в списке *Интервалы группировки* опцию *обычный*.

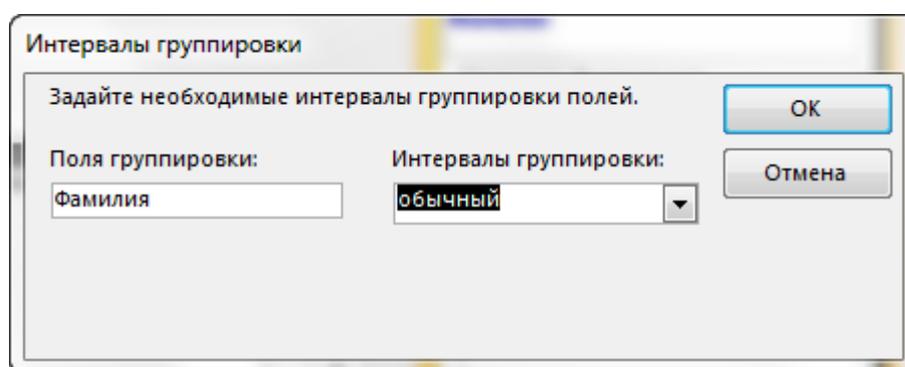


Рис. 3.99. Окно *Интервалы группировки* мастера создания отчётов

6. На следующем шаге мастера нужно выбрать порядок сортировки и вычисления, выполняемые для записей (рис. 3.100). Выберите последовательно поля *Фамилия*, *Название* и *Выражение1*. Затем установите курсор в поле *Выражение1* и нажмите кнопку *Итоги*.

Примечание: Кнопка *Итоги* доступна лишь в случае, если в разделе подробностей отчёта содержится одно или несколько числовых полей. Мастер отображает доступные числовые поля.

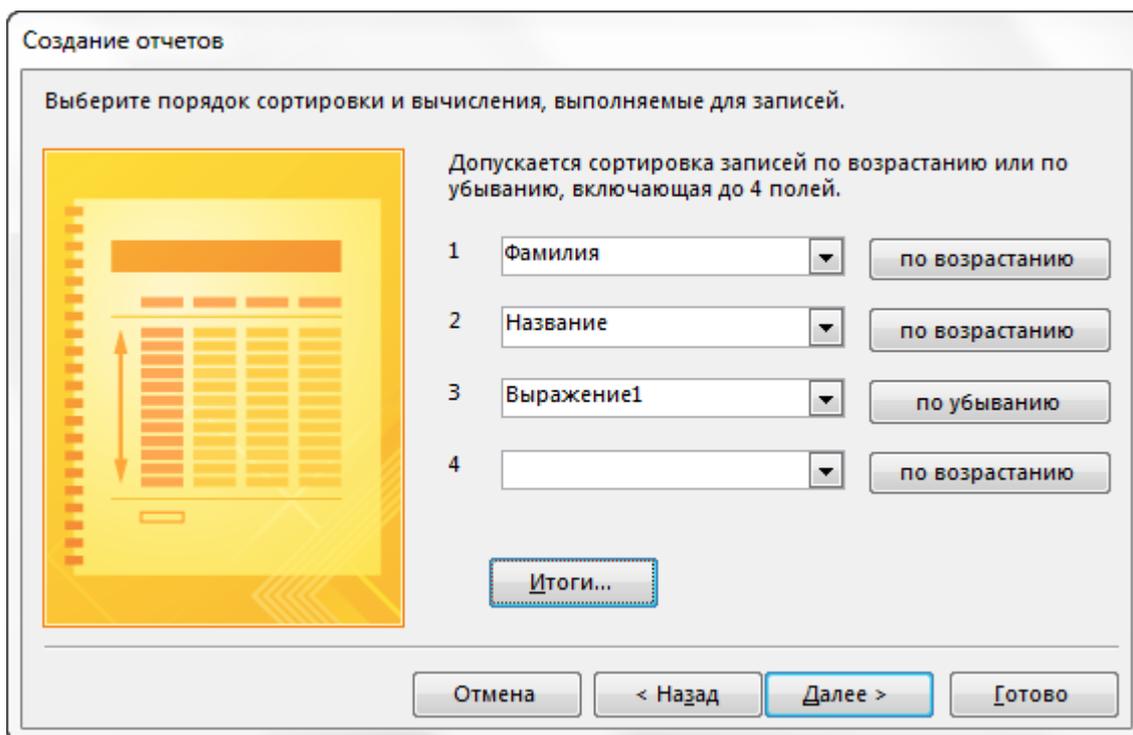


Рис. 3.100. Шаг мастера создания отчётов:
Выбор порядка сортировки и вычисления

7. В открывшемся окне (рис. 3.101) установите флажок в поле *Sum*, выберите *Показать: данные и итоги*, а также установите флажок *Вычислить проценты* и нажмите кнопку *OK*. Затем выберите вид макета для отчёта *ступенчатый*.

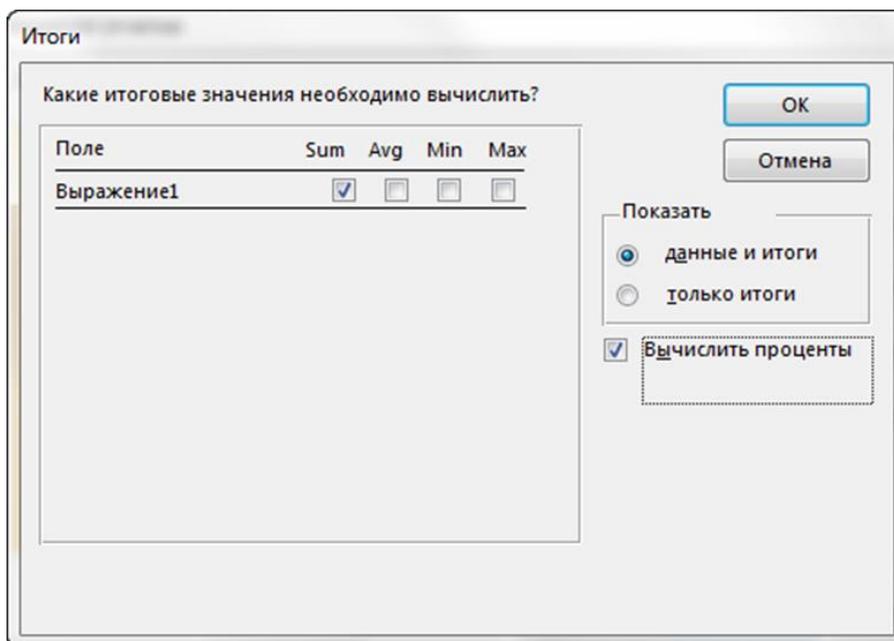


Рис. 3.101. Окно *Итоги* мастера создания отчётов

8. На последнем шаге можно изменить заголовок отчёта. Этот заголовок будет отображаться на первой странице отчёта, и Access сохранит отчёт, используя этот заголовок в качестве имени. В дальнейшем можно изменить как заголовок, так и имя отчёта. На последнем этапе следует выбрать *Изменить макет* или *Просмотреть отчёт* и нажать кнопку *Готово*.

Если по окончании работы мастера обнаружится существенное несоответствие полученного отчёта с выполненным отчётом вручную, удалите его и создайте вновь. Постарайтесь добиться наиболее полного соответствия отчёта, а затем его надо доработать в режиме конструктора.

Доработка отчёта в режиме конструктора

После окончания работы мастера, используя средства конструктора отчётов, можно подкорректировать структуру отчёта: добавить новые элементы управления и поля в отчёт, разместить, отформатировать надписи и поля в областях отчёта, убрать лишние поля. На странице свойств можно задать большое количество свойств для настройки отчёта.

Чтобы переключиться в режим конструктора, щёлкните правой кнопкой мыши имя отчёта в области навигации и выберите команду *Конструктор*. Для изменения свойств отчёта, элементов управления и разделов можно использовать страницу свойств.

Для добавления полей из базовой таблицы или запроса в структуру отчёта можно использовать область *Список полей* (см. выше *Редактирование источника записей отчёта*).

В нашем примере, после создания отчёта с помощью мастера, чтобы данные каждого сотрудника отображались с новой страницы отчёта, следует установить, для заголовка группы на вкладке *Макет* окна свойств в строке *Конец страницы*, значение *До раздела*.

Когда необходимо в отчёте сформировать титульную страницу, в свойствах заголовка отчёта на вкладке *Макет* в строке *Конец страницы* следует установить значение *После раздела*. Увеличивая размер области заголовка отчёта до размера высоты страницы, можно оформить титульную страницу по своему усмотрению.

Создание почтовых наклеек с помощью мастера

В Access для создания наклеек, содержащих данные из таблиц, предусмотрено несколько различных возможностей. Проще всего создать отчёт, отформатированный в соответствии с размером наклеек, воспользовавшись функцией Access «Создание наклеек». Наклейки могут содержать данные как из таблиц Access, так и данные, импортированные из других источников, например, книг Excel или списков контактов Microsoft Outlook. Можно выполнить слияние данных Access с документом Word и напечатать наклейки с

помощью приложения Word. В данном параграфе рассмотрим вопрос использования мастера Access для создания и печати наклеек.

Наклейки создаются в виде отчёта, который отформатирован с размером в соответствии с размером наклейки. Наклейки используются чаще всего для почтовых отправок, адресные данные для которых используются из таблиц или запросов. Но в формате наклеек можно напечатать любые данные Access. Для каждого адреса из источника записей получают отдельные наклейки, которые печатаются в отчёте.

Мастер наклеек создаёт отчёт согласно ответам пользователя на вопросы, выводимые на экран. Вопросы касаются данных, которые должны быть отображены в наклейках.

Пример

Задание. В учебной базе данных *Борей* создать почтовые наклейки с адресами клиентов.

Для исполнения задания данного выполните следующие действия.

1. Выделите таблицу *Клиенты* и на вкладке *Создать* в группе *Отчёты* нажмите кнопку *Наклейки*.
2. После начала работы мастера «Создание наклеек» откроется окно выбора формата наклеек и бумаги для них (рис. 3.102).

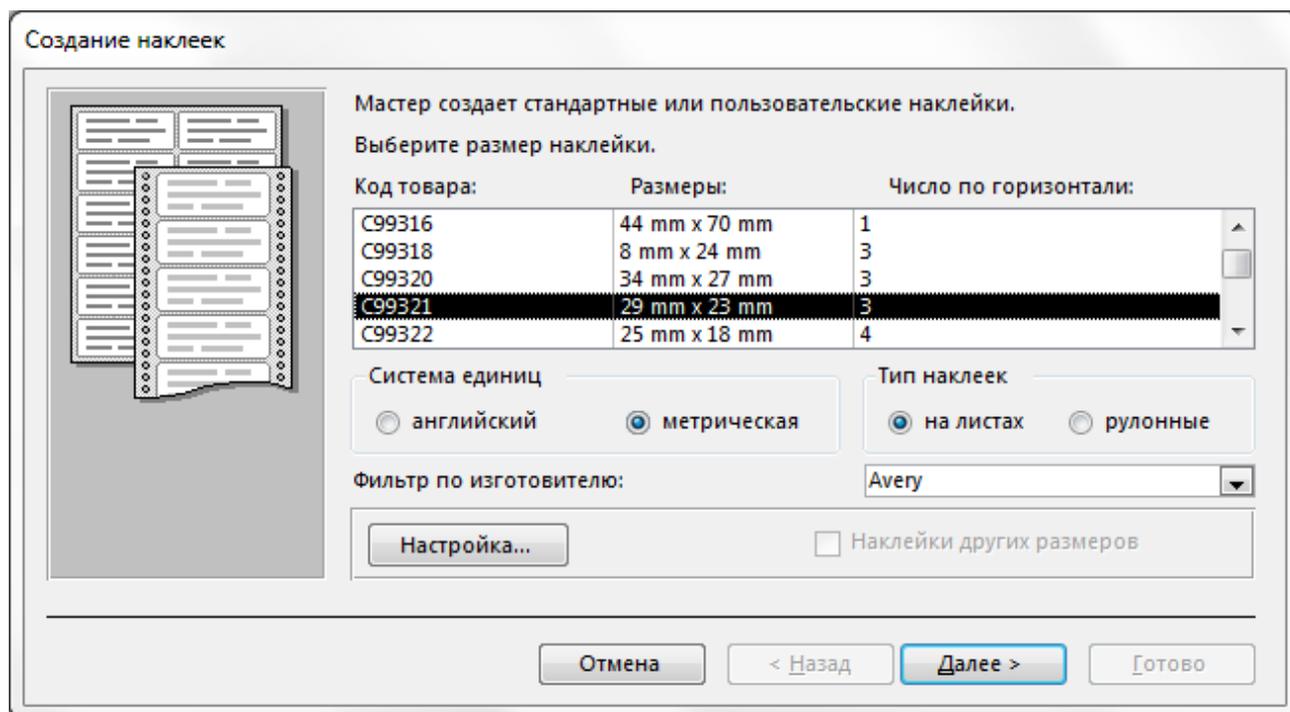


Рис. 3.102. Первый шаг работы мастера «Создание наклеек»

3. Выберите пункты *Тип наклеек* – на листах, *Система единиц* – метрическая и любой размер наклейки с расположением *Число по горизонтали* – по три в ряд. Нажмите кнопку *Далее*.

4. На втором шаге мастера (рис. 3.103) выберите шрифт – *Arial Cyr*, размер – *10 pt*, насыщенность – *Обычный*, не курсив и любой цвет. Нажмите кнопку *Далее*.

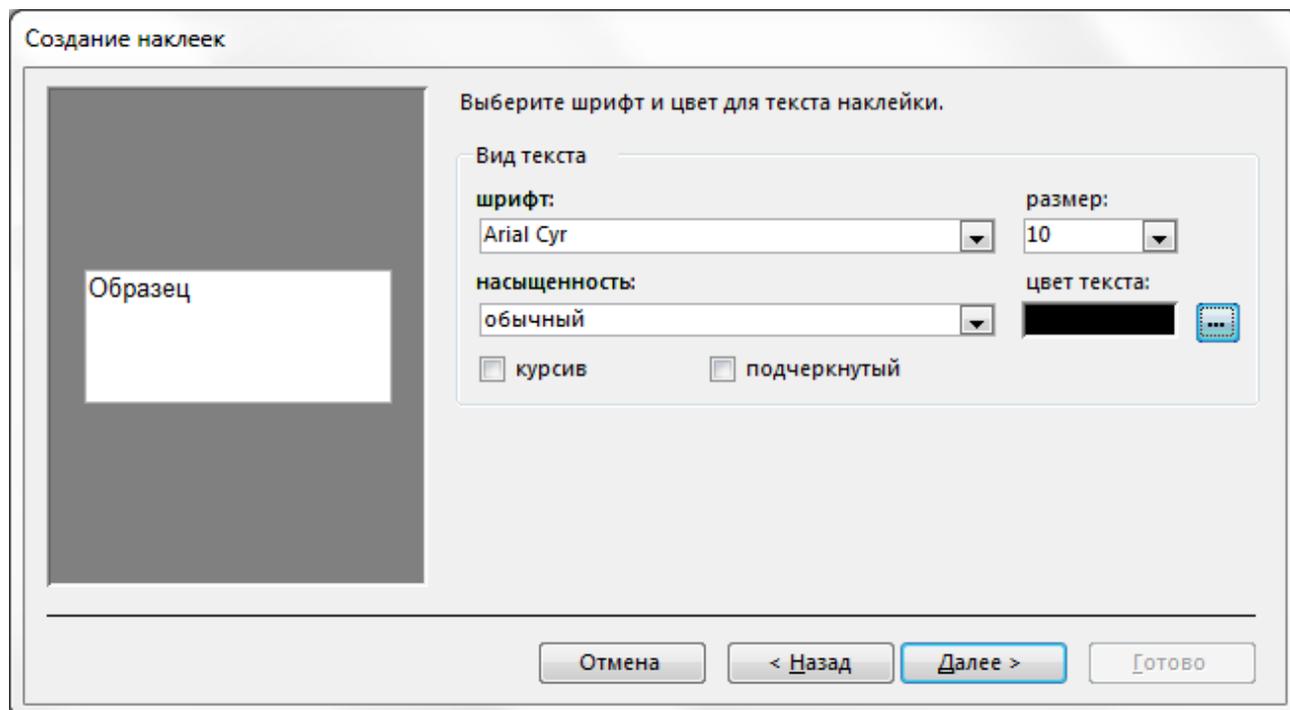


Рис. 3.103. Второй шаг мастера для выбора шрифта и цвета наклейки

5. В окне мастера *Какие поля требуется разместить в наклейке?* (рис 3.104) отберите поля, участвующие в формировании текста наклейки и расположите их следующим образом (точно так, как это предложено в образце, с соблюдением порядка полей и знаков пунктуации):

{Название}
{Адрес}
{Город}, {Область}
{Индекс}, {Страна}

После того как все поля будут расположены на наклейке необходимым образом, нажмите кнопку *Далее*.

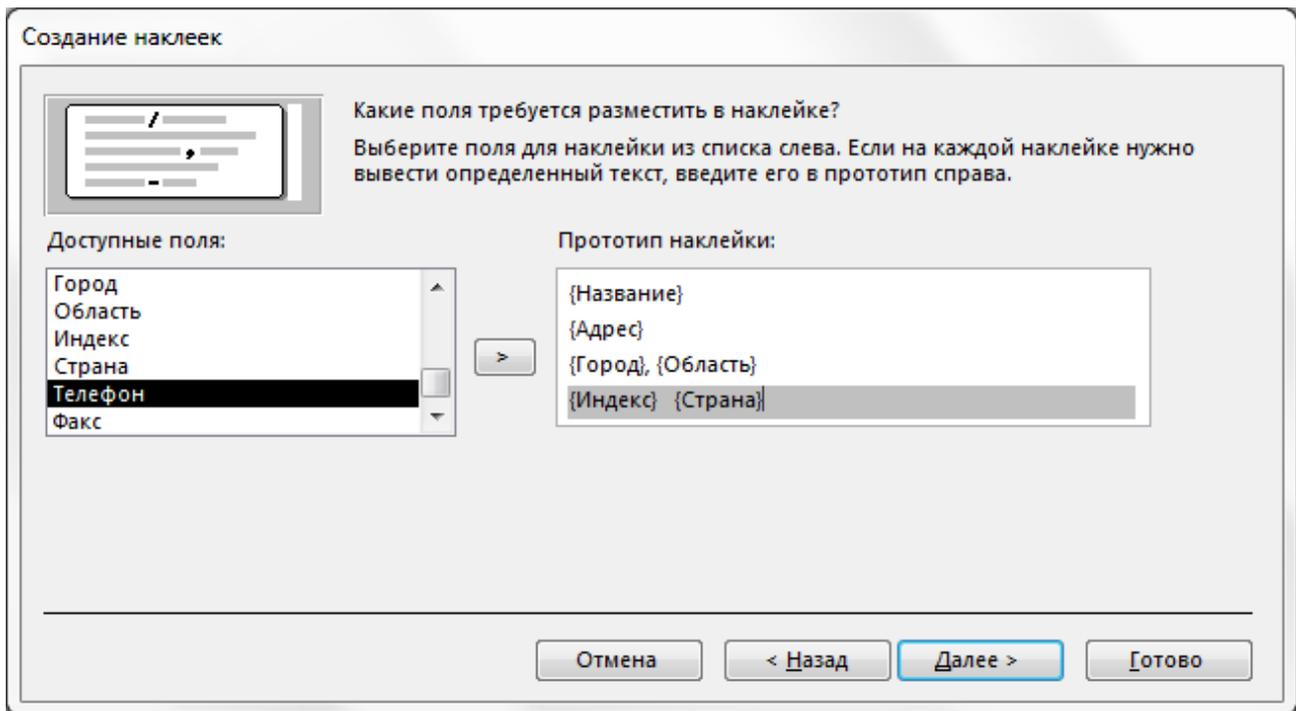


Рис. 3.104. Третий шаг мастера для выбора полей, которые требуется разместить в наклейке

6. В следующем окне (рис. 3.105) нужно выбрать поля, по которым будет выполняться сортировка наклеек. Выберите порядок сортировки из списка *Доступные поля* по полю *Страна*. После этого нажмите кнопку *Далее*.

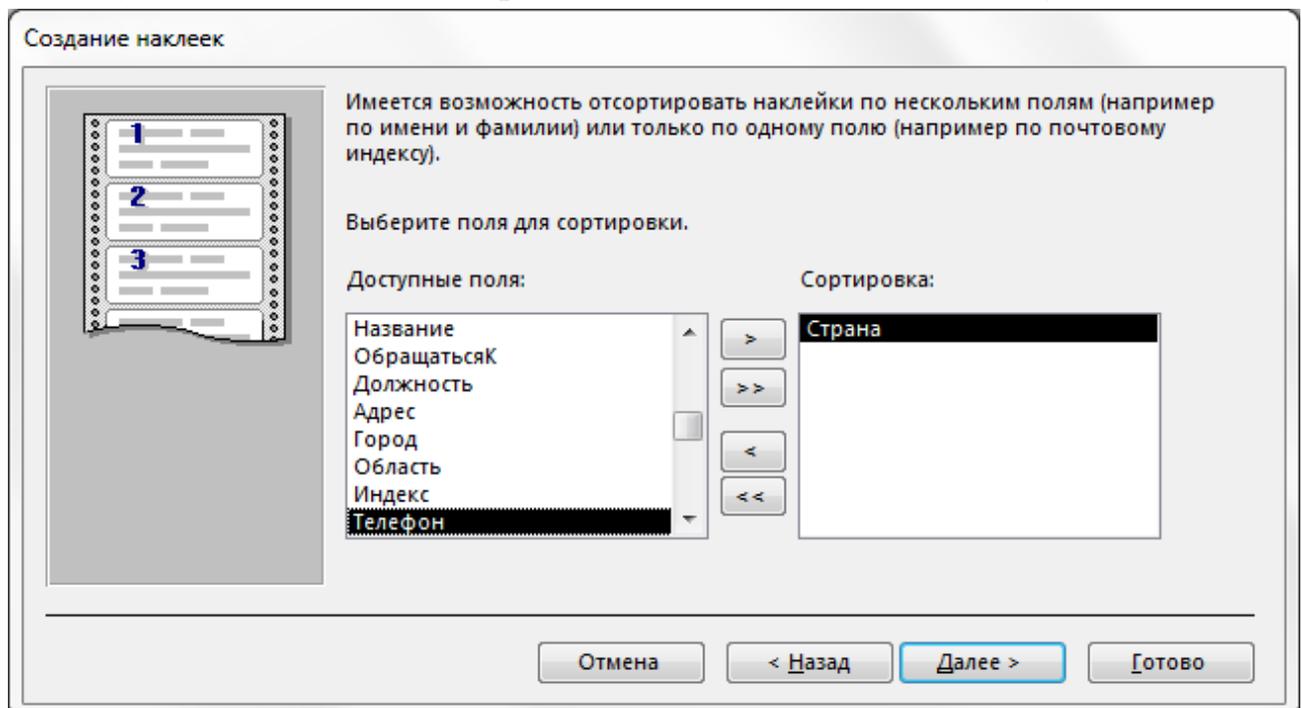


Рис. 3.105. Третий шаг мастера «Создание наклеек»

7. В открывшемся окне введите имя для отчёта *Печать этикеток адресов клиентов* и нажмите кнопку *Готово*.

При выборе в последнем окне мастера параметра *Просмотреть наклейки в том виде, как они будут напечатаны* наклейки будут показаны в режиме предварительного просмотра. Если был выбран параметр *Изменить макет наклеек*, открывается режим конструктора. Следует заметить, что *Предварительный просмотр* – это единственный режим, в котором столбцы отображаются в том порядке, в котором их расположил пользователь. При просмотре отчёта в режиме отчёта или макета все данные отображаются в одном столбце.

При открытии объекта Access (например, отчёта) иногда отображается диалоговое окно *Введите значение параметра*. При открытии объекта, содержащего идентификатор или выражение, которые Access не может интерпретировать, откроется, например, следующее диалоговое окно (рис.3.106).

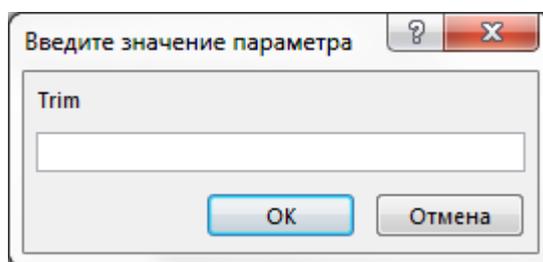


Рис. 3.106. Диалоговое окно *Введите значение параметра*

В определённых случаях это то, что нужно пользователю, а в других случаях появление этого диалогового окна нежелательно. Функция *Trim* возвращает значение содержащее копию указанной строки без начальных и конечных пробелов. В нашем случае она нам не нужна. Чтобы диалоговое окно *Введите значение параметра* больше не появлялось, необходимо в режиме конструктора удалить эту функцию.

В результате получим следующий отчёт (см. рис. 3.107 и рис. 3.108).

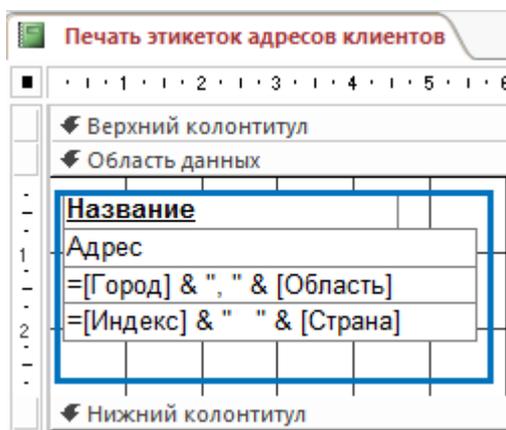


Рис. 3.107. Отчёт *Печать этикеток адресов клиентов* в конструкторе

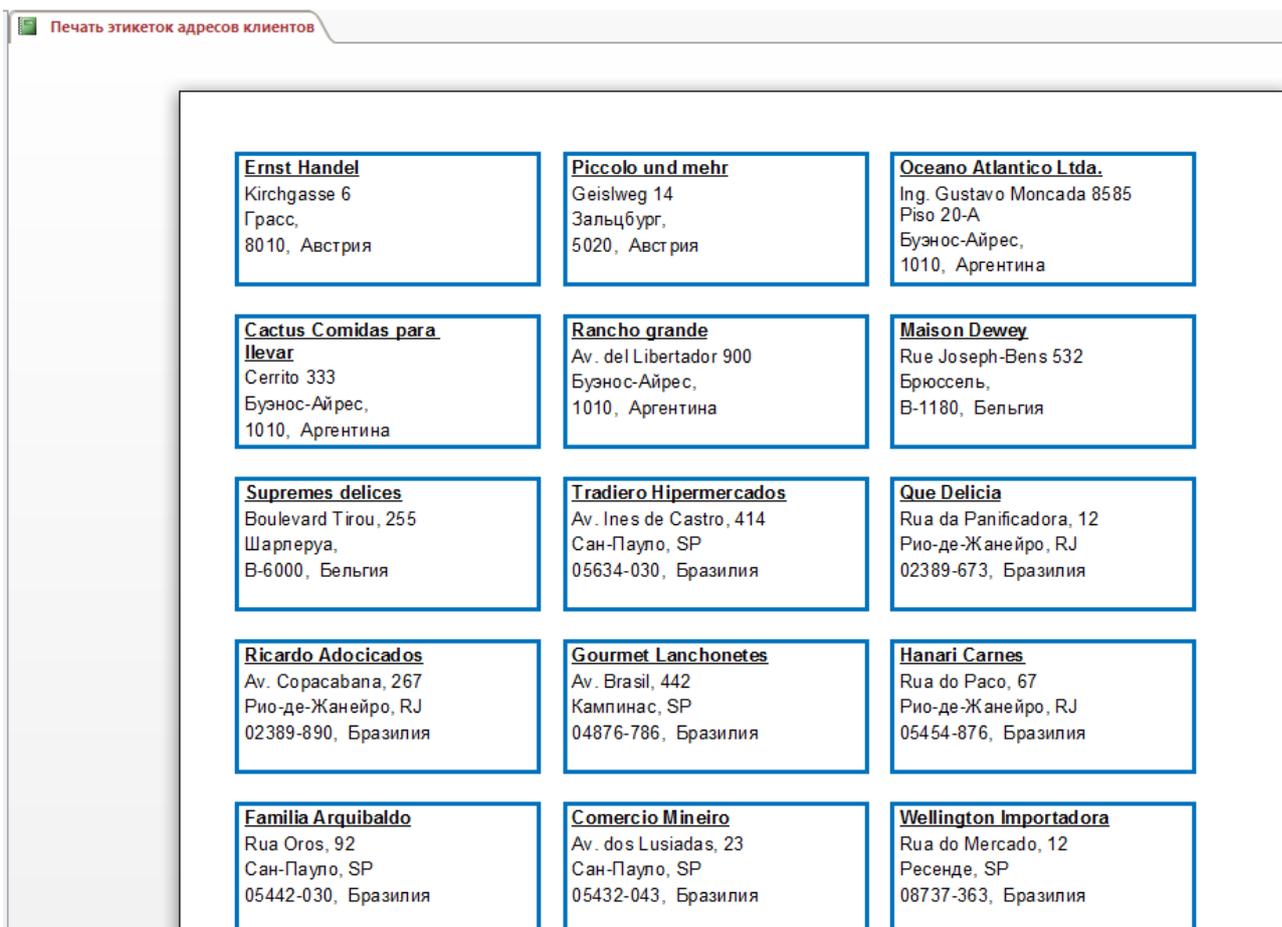


Рис. 3.108. Фрагмент отчёта *Печать этикеток адресов клиентов* в режиме *Предварительный просмотр*

Если получилась не те наклейки, которые нужны, можно настроить макет, открыв отчёт в режиме конструктора и внося в него изменения. Просмотр результатов изменений, сделанных в режиме конструктора осуществляется в режиме предварительного просмотра.

Для изменения полей или интервалов между наклейками в режиме конструктора на вкладке *Параметры страницы* в группе *Разметка страницы* необходимо щёлкнуть элемент *Колонки*. Появится диалоговое окно *Параметры страницы* (рис. 3.109) с открытой вкладкой *Колонки*, где можно изменить параметры в трёх разделах: *Параметры сетки*, *Размер столбца* и *Макет столбца*.

Диалоговое окно *Параметры страницы* можно открыть и в режиме предварительного просмотра, нажав на вкладке *Предварительный просмотр* в группе *Разметка страницы* кнопку *Колонки*. При нажатии кнопки *ОК* немедленно отражаются все изменения в режиме предварительного просмотра.

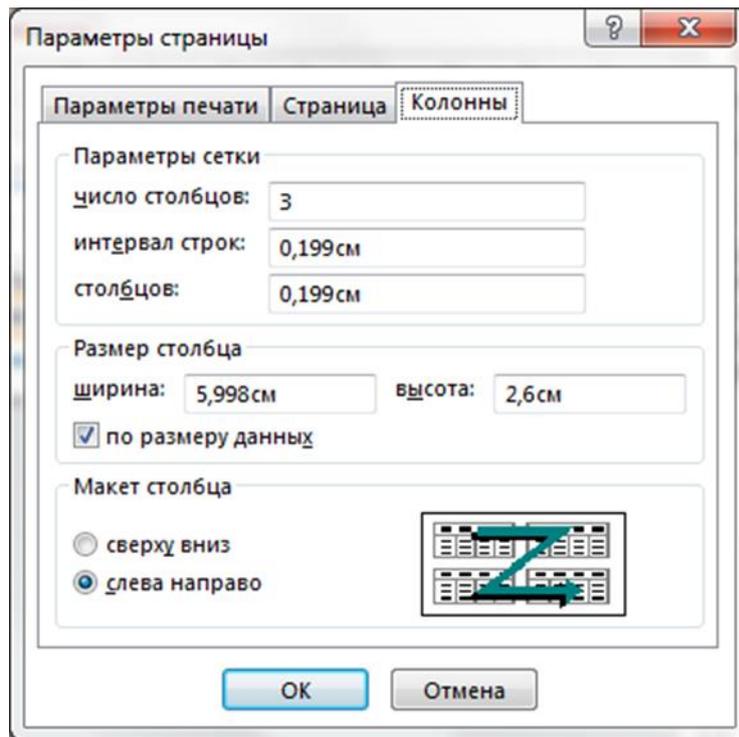


Рис. 3.109. Диалоговое окно *Параметры страницы* с открытой вкладкой *Колонки*

8. В режиме *Конструктора* откорректируйте отчёт следующим образом.
 - Поле *Название* оформите жирным подчёркнутым шрифтом.
 - Вокруг этикетки создайте рамку синего цвета.
 - В диалоговом окне *Параметры страницы* на вкладке *Параметры печати* установите все поля *10 мм*.
 - Отформатируйте размеры полей данных так, чтобы весь текст помещался в рамку и все размеры по горизонтали не превышали обычный диапазон печати.
 - Если при переходе в режим предварительного просмотра отображается сообщение об отсутствии достаточного пространства по горизонтали для отображения данных нужно переместить элементы управления и отрегулировать размер наклейки, изменив ширину или высоту раздела данных. Убедитесь, что все правильно сделано в режиме *Предварительный просмотр*.

СТРУКТУРА ТАБЛИЦ БАЗЫ ДАННЫХ «АВТОМОБИЛИ»

Таблица П1.

Данные для создания таблицы *Модели*

Имя поля	Тип данных	Описание поля
Код модели (первичный ключ)	Числовой	Код заказа модели по заводскому каталогу
Модель	Короткий текст	Тип кузова
Мощность двигателя	Короткий текст	Мощность двигателя (кВт/л.с.)
Цвет	Короткий текст	Цвет кузова
Количество дверей	Числовой	Двухдверный или четырехдверный салон
Коробка передач	Короткий текст	Автоматическая или ручная
Обивка	Короткий текст	Велюр, кожа, ткань
Другое оснащение	Длинный текст	Дополнительные аксессуары
Заводская цена	Денежный	Заводская продажная нетто-цена
Транспортные издержки	Денежный	Издержки на доставку
Предпродажная подготовка	Денежный	Издержки на предпродажную подготовку
Специальная модель	Логический	Специальная модель или стандартное исполнение

Данные для оформления свойств полей таблицы *Модели*

Характеристика поля	Значение
Имя поля	<i>Код модели</i>
Размер поля	Длинное целое
Формат поля	Основной
Число десятичных знаков	0
Обязательное поле	Да
Индексированное поле	Да (совпадения не допускаются)
Имя поля	<i>Модель</i>
Размер поля	20
Индексированное поле	Да (допускаются совпадения)
Имя поля	<i>Мощность двигателя</i>
Размер поля	10
Имя поля	<i>Цвет</i>
Размер поля	10
Имя поля	<i>Количество дверей</i>
Размер поля	Байт
Формат поля	Основной
Число десятичных знаков	0
Значение по умолчанию	4
Условие на значение	2 or 4
Сообщение об ошибке	Данное поле может содержать только 2 или 4!
Имя поля	<i>Коробка передач</i>
Размер поля	15
Значение по умолчанию	Ручная
Условие на значение	Ручная or Автоматика
Сообщение об ошибке	Допустимы только значения «Ручная» или «Автоматика»!
Имя поля	<i>Обивка</i>
Размер поля	10

Характеристика поля	Значение
Имя поля	<i>Другое оснащение–</i>
Размер поля	Это Мемо–поле, поэтому никаких ограничений на размер поля не существует
Имя поля	<i>Заводская цена</i>
Формат поля	Денежный
Число десятичных знаков	2
Имя поля	<i>Транспортные издержки</i>
Формат поля	Денежный
Число десятичных знаков	2
Имя поля	<i>Предпродажная подготовка</i>
Формат поля	Денежный
Значение по умолчанию	80
Число десятичных знаков	2
Имя поля	<i>Специальная модель</i>
Формат поля	Да/Нет
Индексированное поле	Нет

Данные для заполнения таблицы *Модели*

Код модели	Модель	Мощность двигателя	Цвет	Количество дверей	Коробка передач	Обивка	Другое оснащение	Заводская цена	Транспортные издержки	Предпродажная подготовка	Специальная модель
12579	Volvo	53/75	Красный	4	Ручная	Ткань	Радио/плеер, раздвижная крыша	30700	950	90	Нет
12580	KIA	69/90	Зеленый	4	Ручная	Ткань	Радио/плеер, лаковое покрытие «металлик»	32200	1200	95	Нет
12653	BMW	100/139	Черный	4	Автоматика	Кожа	Радио/плеер, алюминиевые «дворники»	41100	975	105	Нет
12743	BMW	100/140	Черный	2	Ручная	Кожа	Радио/плеер	37900	900	80	Да
12651	Toyota	90/135	Голубой	2	Ручная	Велюр	Раздвижная крыша, автоподъемник окон	37900	1050	105	Да
12422	Mazda	60/90	Белый	4	Ручная	Ткань	Навесной багажник	36200	1100	105	Нет
12410	Mazda	80/100	Черный	4	Автоматика	Велюр	Радио/плеер, раздвижная крыша	50200	1600	115	Да
11052	BMW	100/130	Красный	4	Ручная	Кожа	Радио/плеер	37100	975	105	Нет

Данные для создания таблицы *Клиенты*

Имя поля	Номер клиента (первичный ключ)
Тип данных	Числовой
Размер поля	Длинное целое
Формат	Основной
Число десятичных знаков	0
<hr/>	
Имя поля	Фамилия
Тип данных	Короткий текст
Размер поля	30
<hr/>	
Имя поля	Имя
Тип данных	Короткий текст
Размер поля	15
<hr/>	
Имя поля	Населенный пункт Списочное поле со значениями «Харьков»; «Киев»; «Одесса» с возможностью введения другого города
Тип данных	Короткий текст
Размер поля	20
<hr/>	
Имя поля	Почтовый адрес
Тип данных	Короткий текст
Размер поля	40
<hr/>	
Имя поля	Почтовый индекс
Тип данных	Числовой
Размер поля	Длинное целое
Число десятичных знаков	0
Индексированное поле	Да (совпадения допускаются)
<hr/>	
Имя поля	Телефон
Тип данных	Короткий текст
Размер поля	20

Данные для проектирования таблицы *Купля автомобилей*

Имя поля	<i>Номер заказа</i> (первичный ключ)
Тип данных	Счетчик
<hr/>	
Имя поля	<i>Номер клиента</i>
Тип данных	Числовой
Размер поля	Длинное целое
Формат	Основной
Число десятичных знаков	0
Индексированное поле	Да (совпадения допускаются)
<hr/>	
Имя поля	<i>Код модели</i>
Тип данных	Числовой
Размер поля	Длинное целое
Формат	Основной
Число десятичных знаков	0
Индексированное поле	Да (совпадения допускаются)
<hr/>	
Имя поля	<i>Дата заказа</i>
Тип данных	Дата/Время
Формат поля	Краткий формат даты
Индексированное поле	Да (совпадения допускаются)
<hr/>	
Имя поля	<i>Скидка</i>
Тип данных	Числовой
Размер поля	Одинарное с плавающей точкой
Формат	Процентный
Число десятичных знаков	0
Индексированное поле	Нет

Данные для создания таблицы *Мастерские*

№ мастерской	Адрес	Категория	Телефон	Директор	Средняя цена	Наценка
34	Харьков, ул. Мира, 24	высшая	322-14-81	Вставьте свою фамилию	2500	13%
83	Харьков, пер. Правды, 9	первая	741-18-79	Козак	2000	7%
47	Киев, ул. Патриотов, 1	высшая	124-81-09	Чумак	2600	8%
161	Одесса, ул. Морская, 80	первая	350-11-72	Чалый	2100	10%
3	Львов, ул. Киевская, 20	вторая	644-01-17	Иващенко	1500	
40	Киев, пр. Победы, 32/1	высшая	101-91-50	Левицкий	2700	15%

ГОРЯЧИЕ КЛАВИШИ MS ACCESS



Ниже приведены несколько советов о том, как ускорить работу, сделать ее более простой и приятной, как равномерно распределить нагрузку на кисти рук

с помощью так называемых «горячих клавиш» или *клавиш быстрого доступа* – комбинации нескольких одновременно нажатых клавиш, которые позволяют выполнять наиболее распространенные команды без использования мыши и без вызова меню. В *MS Access*

для быстрого доступа к часто используемым командам и операциям для выполнения определенных действий, например, для выбора команд меню или элементов управления без помощи мыши, достаточно часто используется комбинация двух или трех клавиш. Любая комбинация содержит одну из специальных клавиш: *<Ctrl>*, *<Alt>*, *<Shift >* или *<Scroll Lock>*. Для ввода требуемой комбинации клавиш необходимо нажать клавишу, указанную в комбинации первой, и, не отпуская её, нажать вторую клавишу, а при необходимости и третью. Например, для ввода комбинации клавиш *<Ctrl>+<Enter>* нужно нажать клавишу *<Ctrl>* и, удерживая ее, нажать клавишу *<Enter>*. Если клавиши разделены запятой, например: *<End>*, *<Enter>*, значит надо нажать и отпустить первую клавишу *<End>*, затем нажать и отпустить вторую клавишу *<Enter>*.

Использование сочетаний клавиш значительно ускоряет работу и увеличивает количество возможных действий, выполняемых с помощью клавиатуры. Перечисленные ниже таблицы содержат списки сочетаний клавиш, доступных в *MS Access*.

Общие сочетания клавиш

Общие сочетания клавиш Microsoft Access

Клавиши	Действие
CTRL+N	Открытие новой базы данных
CTRL+O	Открытие существующей базы данных
ALT+F4	Выход из Microsoft Access
CTRL+P	Печать текущего или выделенного объекта
P или CTRL+P	Открытие диалогового окна Печать
S	Открытие диалогового окна Параметры страницы
C или ESC	Выход из режима предварительного просмотра
CTRL+S или SHIFT+F12 или ALT+SHIFT+F2	Сохранение объекта базы данных

Клавиши	Действие
F12 или ALT+F2	Открытие диалогового окна Сохранение
F4 или ALT+СТРЕЛКА ВНИЗ	Открытие поля со списком
F9	Обновление содержимого подстановочного поля (Поле подстановок. Поле, используемое в форме или отчёте в базе данных Microsoft Access, котором выводится либо список значений, загружаемый из таблицы или запроса, либо сохраняемый постоянный список значений.) списка или поля со списком
СТРЕЛКА ВНИЗ	Переход на одну строку вниз
PAGE DOWN	Переход на одну страницу вниз
PAGE UP	Переход на одну страницу вверх
TAB	Выход из окна списка или поля со списком
CTRL+F	В диалоговом окне Поиск и замена открытие вкладки Поиск (только в режимах таблицы и формы)
CTRL+H	В диалоговом окне Поиск и замена открытие вкладки Замена (только в режимах таблицы и формы)
SHIFT+F4	Поиск следующего вхождения текста, указанного в диалоговом окне Поиск и замена, при закрытом диалоговом окне (только в режимах таблицы и формы)

Общие сочетания клавиш в режиме конструктора

Клавиши	Действие
F2	Переключение между режимом редактирования (при отображении курсора) и режимом перемещения (Режим перехода. В Microsoft Access – режим выделения всего поля, когда курсор не отображается. При этом с помощью клавиш со стрелками можно выполнять переходы между полями)
F4	Переключение в окно свойств (режим конструктора формы или отчёта в базах данных или в проектах Microsoft Access)
F5	Переключение формы из режима конструктора в режим формы
F6	Переключение между верхней и нижней частями окна (в режиме конструктора для таблиц, макросов и запросов и только в окне расширенного фильтра)
F7	Переключение в построитель программ из формы или отчёта (в окне режима конструктора или окне свойств)
SHIFT+F7	Переключение из редактора Microsoft Visual Basic в режим конструктора формы или отчёта
ALT+V+P	Открытие окна свойств выбранного объекта

Общие сочетания клавиш в режиме конструктора в формах или отчетах

Клавиши	Действие
SHIFT+ENTER	Добавление элемента управления в раздел
CTRL+C	Копирование выделенного элемента управления в буфер обмена
CTRL+X	Вырезание выделенного элемента управления с копированием в буфер обмена
CTRL+V	Вставка содержимого буфера обмена в левый верхний угол выделенного раздела
CTRL+СТРЕЛКА ВПРАВО	Перемещение выделенного элемента управления вправо
CTRL+СТРЕЛКА ВЛЕВО	Перемещение выделенного элемента управления влево
CTRL+СТРЕЛКА ВВЕРХ	Перемещение выделенного элемента управления вверх
CTRL+СТРЕЛКА ВНИЗ	Перемещение выделенного элемента управления вниз
SHIFT+СТРЕЛКА ВНИЗ	Увеличение высоты выделенного элемента управления
SHIFT+СТРЕЛКА ВПРАВО	Увеличение ширины выделенного элемента управления
SHIFT+СТРЕЛКА ВВЕРХ	Уменьшение высоты выделенного элемента управления
SHIFT+СТРЕЛКА ВЛЕВО	Уменьшение ширины выделенного элемента управления
F11	Перевод окна базы данных на передний план
CTRL+F6	Переход между открытыми окнами по порядку
ENTER	Восстановление выделенного свёрнутого окна, когда все окна свёрнуты
CTRL+F8	Включение режима изменения размера активного окна, когда оно развернуто; для изменения размера окна используются клавиши со стрелками
ALT+ПРОБЕЛ	Открытие оконного меню
SHIFT+F10	Открытие контекстного меню
CTRL+W или CTRL+F4	Закрытие активного окна
ALT+F11	Переключение между редактором Microsoft Visual Basic и предыдущим активным окном
ALT+SHIFT+F11	Переключение в редактор сценариев Microsoft Script Editor из предыдущего активного окна
TAB	Переход к кнопке «Справка»  в мастере

Клавиши	Действие
ALT+C	Переход в следующее окно в мастере <i>Примечание.</i> Нельзя использовать сочетание клавиш CTRL+TAB для перехода из одного окна мастера в другое. CTRL+TAB перемещает курсор между диалоговыми окнами и элементами управления окна мастера.
ALT+B	Переход в предыдущее окно в мастере
ALT+F	Закрытие окна мастера
F2	Отображение полного адреса выделенной гиперссылки
F7	Проверка орфографии
SHIFT+F2	Открытие окна «Область ввода» для удобного ввода выражений и другого текста в небольшие области ввода
ALT+ENTER	Отображение окна свойств в режиме конструктора
ALT+F4	Выход из Microsoft Access, закрытие диалогового окна или окна свойств
CTRL+F2	Вызов построителя
CTRL+F11	Переключение между пользовательской строкой меню и встроенной строкой меню
CTRL+СТРЕЛКА ВПРАВО или CTRL+ЗАПЯТАЯ (,)	Переключение в следующий режим в таблице, запросе, форме, отчёте, странице, представлении, сводном списке, сводной диаграмме отчёта, сохранённой процедуре или в функциях проекта Microsoft Access (.adp). Если доступны дополнительные режимы, последовательное нажатие клавиш переключает в следующий доступный режим.
CTRL+СТРЕЛКА ВЛЕВО или CTRL+ТОЧКА (.)	Переключение в предыдущий режим в таблице, запросе, форме, отчете, странице, представлении, сводном списке, сводной диаграмме отчёта, сохранённой процедуре или в функциях (.adp). Если доступны дополнительные режимы, последовательное нажатие клавиш переключает в предыдущий режим.

Сочетания клавиш в окне базы данных

Редактирование и перемещение по списку «Объекты»

Клавиши	Действие
F2	Переименование выделенного объекта
СТРЕЛКА ВНИЗ	Переход на одну строку вниз
PAGE DOWN	Переход вниз на одно окно
END	Переход к последнему объекту
СТРЕЛКА ВВЕРХ	Переход на одну строку вверх
PAGE UP	Переход вверх на одно окно
HOME	Переход к первому объекту

Переходы и открытие объектов

Клавиши	Действие
CTRL+TAB	Циклический переход сверху вниз по списку Объекты
SHIFT+CTRL+TAB	Циклический переход снизу-вверх по списку Объекты
ENTER или ALT+O	Открытие выделенной таблицы или запроса в режиме таблицы, либо формы в режиме формы.
ENTER	Открытие выделенного отчёта в режиме предварительного просмотра.
ENTER	Открытие выделенной страницы доступа к данным в режиме страницы
ENTER	Запуск выделенного макроса
CTRL+ENTER или ALT+K	Открытие выделенной таблицы, запроса, формы, отчёта, страницы доступа к данным, макроса или модуля в режиме конструктора
ALT+C	Создание новой таблицы, запроса, формы, отчёта, страницы доступа к данным, макроса или модуля
F5	Обновление окна базы данных
CTRL+G	Вывод окна проверки в редакторе Visual Basic

Работа с меню

Клавиши	Действие
SHIFT+F10	Открытие контекстного меню
F10	Активизация строки меню
ALT+ПРОБЕЛ	Открытие оконного меню программы (в строке заголовка)
СТРЕЛКА ВНИЗ или СТРЕЛКА ВВЕРХ	Выбор следующей или предыдущей команды в открытом меню или подменю
СТРЕЛКА ВЛЕВО или СТРЕЛКА ВПРАВО	Выбор меню справа или слева либо (при открытом подменю) переключение между главным меню и подменю
HOME или END	Выбор первой или последней команды меню или подменю
ALT	Одновременный выход из видимых меню и подменю
ESC	Выход из видимых меню или, если на экран выведено только подменю, выход только из подменю

Работа с панелями инструментов

Клавиши	Действие
CTRL+TAB или CTRL+SHIFT+TAB	Выбор следующей или предыдущей панели инструментов
TAB или SHIFT+TAB	Выбор следующей или предыдущей кнопки, или меню на активной панели инструментов
ENTER	Открытие меню, выбранного на панели инструментов
ENTER	Выполнение действия кнопки, выбранной на панели инструментов

Работа в окнах и диалоговых окнах

Действие в окне программы

Клавиши	Действие
ALT+TAB	Переключение на следующую программу
ALT+SHIFT+TAB	Переключение на предыдущую программу
CTRL+ESC	Вывод меню Пуск Microsoft Windows
CTRL+W	Выход из активного окна базы данных
CTRL+F6	Переключение на следующее окно базы данных
CTRL+SHIFT+F6	Переключение на предыдущее окно базы данных
ENTER	Восстановление выделенного свёрнутого окна, когда все окна свёрнуты

Действие в диалоговом окне

Клавиши	Действие
CTRL+TAB или CTRL+PAGE DOWN	Переключение на следующую вкладку в диалоговом окне
CTRL+SHIFT+TAB или CTRL+PAGE UP	Переключение на предыдущую вкладку в диалоговом окне
TAB	Перемещение на следующий параметр или группу параметров
SHIFT+TAB	Перемещение на предыдущий параметр или группу параметров
Клавиши со стрелками	Перемещение между параметрами в выбранном раскрывающемся списке или между параметрами в группе параметров
ПРОБЕЛ	Выполнение действия, назначенного для выбранной кнопки; установка или снятие флажка
Клавиша с первой буквой в имени параметра (когда открыт раскрывающийся список)	Переход на параметр по первой букве в имени в раскрывающемся списке

Клавиши	Действие
ALT+клавиша с буквой	Выбор параметра либо установка или снятие флажка при помощи буквы, подчёркнутой в названии параметра
ALT+СТРЕЛКА ВНИЗ	Открытие выбранного раскрывающегося списка
ESC	Заккрытие выбранного раскрывающегося списка
ENTER	Выполнение действия кнопки по умолчанию в диалоговых окнах
ESC	Отмена команды и выход из диалогового окна
ALT+F4	Заккрытие диалогового окна

Действие при редактировании поля

Клавиши	Действие
HOME	Переход в начало поля
END	Переход в конец поля
СТРЕЛКА ВЛЕВО или СТРЕЛКА ВПРАВО	Переход на один знак вправо или влево
CTRL+СТРЕЛКА ВЛЕВО или CTRL+СТРЕЛКА ВПРАВО	Переход на одно слово вправо или влево
SHIFT+HOME	Выделение от позиции курсора до начала поля
SHIFT+END	Выделение от позиции курсора до конца поля
SHIFT+СТРЕЛКА ВЛЕВО	Расширение выделения на один знак влево
SHIFT+СТРЕЛКА ВПРАВО	Расширение выделения на один знак влево
CTRL+SHIFT+СТРЕЛКА ВЛЕВО	Расширение выделения на одно слово влево
CTRL+SHIFT+СТРЕЛКА ВПРАВО	Расширение выделения на одно слово вправо

Работа с диалоговыми окнами «Открыть файл базы данных», «Файл новой базы данных» и «Сохранение»

Клавиши	Действие
ALT+1	Переход к предыдущей папке ()
ALT+2	Открытие папки, расположенной на один уровень выше открытой папки (кнопка Переход на один уровень вверх )
ALT+3	Заккрытие диалогового окна и открытие страницы поиска в Интернете (кнопка Найти в Интернете )
ALT+4	Удаление выделенной папки или файла (кнопка Удалить )
ALT+5	Создание новой подпапки в открытой папке (кнопка Создать папку )

Клавиши	Действие
ALT+6	Переключение между режимами «Список», «Таблица», «Свойства» и «Просмотр»
ALT+7	Открытие меню Сервис (кнопка Сервис)

Работа с панелями задач

Клавиши	Действие
F6	Переход в область задач (Область задач. Область в окне программы Microsoft Office, содержащая часто используемые команды. Ее расположение и небольшие размеры позволяют использовать эти команды, не прекращая работу над файлами) из другой области окна программы. (Может потребоваться несколько нажатий клавиши F6) <i>Примечание.</i> Если при нажатии клавиши F6 область задач не открывается, попробуйте с помощью клавиши ALT перевести фокус на строку меню, а затем нажмите клавиши CTRL+TAB для перехода на область задач.
CTRL+TAB	Переход в область задач, когда активными являются меню или панель инструментов (Панель инструментов. Строка с кнопками и другими элементами управления, служащими для выполнения команд. Чтобы вызвать панель инструментов, нажмите клавишу ALT, а затем клавиши SHIFT+F10). (Может потребоваться несколько нажатий клавиш CTRL+TAB)
TAB или SHIFT+TAB	Когда область задач активна, выделение следующего или предыдущего параметра в области задач
CTRL+СТРЕЛКА ВНИЗ	Отображение полного набора команд в меню области задач
СТРЕЛКА ВНИЗ или СТРЕЛКА ВВЕРХ	Переходы между командами в выбранном подменю; переходы между параметрами в группе параметров
ПРОБЕЛ или ENTER	Открытие выделенного меню или выполнение действия, назначенного выделенной кнопке
SHIFT+F10	Открытие контекстного меню (Контекстное меню. Меню, содержащее список команд, относящихся к конкретному объекту. Для вызова контекстного меню щелкните объект правой кнопкой или нажмите клавиши SHIFT+F10); открытие раскрывающегося меню для выделенного элемента коллекции
HOME или END	Выбор первой или последней команды в открытом меню или подменю

Клавиши	Действие
PAGE UP или PAGE DOWN	Прокрутка вверх или вниз выбранного списка коллекции
CTRL+HOME или CTRL+END	Переход в начало или в конец выбранного списка коллекции

Сочетания клавиш, используемые в области задач Справка и в окне справки

Панель справки – это область задач, обеспечивающая доступ ко всему содержанию справки Microsoft Office. Она появляется при запуске каждого приложения. Окно справки отображает разделы и содержание справки, которое появляется рядом, но отдельно от основного окна приложения.

В области задач Справка

Клавиши	Действие
F1	Отображает область задач Справка
F6	Переключение между областью задач Справка и основным приложением
TAB	Выбор следующего элемента в области задач Справка
SHIFT+TAB	Выбор предыдущего элемента в области задач Справка
ENTER	Подтверждение выбора элемента
СТРЕЛКА ВНИЗ и СТРЕЛКА ВВЕРХ	Выбор в оглавлении следующего или предыдущего элемента, соответственно
СТРЕЛКА ВПРАВО и СТРЕЛКА ВЛЕВО	Отображать выбранный элемент в развёрнутом и свернутом виде, соответственно
SHIFT+СТРЕЛКА ВЛЕВО	Переход в предыдущую область задач
SHIFT+СТРЕЛКА ВПРАВО	Переход в следующую область задач
CTRL+ПРОБЕЛ	Открыть меню параметров панели
CTRL+F1	Закрыть и открыть заново текущую область задач
СТРЕЛКА ВПРАВО	Развернуть список (+/-)
СТРЕЛКА ВЛЕВО	Свернуть список (+/-)

В окне справки

Клавиши	Действие
TAB	Выбор следующего скрытого текста или гиперссылки, а также ссылок Показать все или Скрыть все в верхней части раздела
SHIFT+TAB	Выбор предыдущего скрытого текста или гиперссылки,

Клавиши	Действие
	а также кнопки поиска в обозревателе в верхней части статьи веб-узла Microsoft Office
ENTER	Выполнение действий выбранных ссылок Показать все, Скрыть все, скрытого текста или гиперссылки
SHIFT+СТРЕЛКА ВЛЕВО	Переход к предыдущему разделу справки
SHIFT+СТРЕЛКА ВПРАВО	Переход к следующему разделу справки
CTRL+P	Печать текущего раздела справки
СТРЕЛКА ВВЕРХ и СТРЕЛКА ВНИЗ	Медленная прокрутка вверх или вниз в пределах текущего раздела справки
PAGE UP и PAGE DOWN	Быстрая прокрутка вверх или вниз в пределах текущего раздела справки
ALT+U	Выбор места появления окна справки: рядом или в стороне от основного приложения
SHIFT+F10	Отображение меню команд окна справки; окно справки должно быть активным (выберите элемент в окне справки)

Отправка сообщений электронной почты

Клавиши	Действие
ALT+A+Ь+C	Отправка по электронной почте активного объекта базы данных
CTRL+SHIFT+B	Открытие адресной книги
ALT+П	Проверка имен в полях Кому, Копия и СК по адресной книге
TAB	Переход к следующему полю в заголовке сообщения или к тексту сообщения, если активно последнее поле заголовка
SHIFT+TAB	Выбор следующего или предыдущего поля в заголовке сообщения

Клавиши для работы с текстом и данными

Выделение текста в поле

Клавиши	Действие
SHIFT+СТРЕЛКА ВПРАВО	Расширение выделения на один знак вправо
CTRL+SHIFT+СТРЕЛКА ВПРАВО	Расширение выделения на одно слово вправо
SHIFT+СТРЕЛКА ВЛЕВО	Расширение выделения на один знак влево
CTRL+SHIFT+СТРЕЛКА ВЛЕВО	Расширение выделения на одно слово влево

Выделение поля или записи

Клавиши	Действие
ТАВ	Выделение следующего поля
F2	Переключение между режимом редактирования (при отображении курсора) и режимом перемещения (Режим перехода. В Microsoft Access – режим выделения всего поля, когда курсор не отображается. При этом с помощью клавиш со стрелками можно выполнять переходы между полями.)
SHIFT+ПРОБЕЛ	Переключение между выделением текущей записи и первого поля текущей записи в режиме перемещения
SHIFT+СТРЕЛКА ВВЕРХ	Расширение выделения на предыдущую запись, если текущая запись была выделена
SHIFT+СТРЕЛКА ВНИЗ	Расширение выделения на следующую запись, если текущая запись была выделена
CTRL+A или CTRL+SHIFT+ПРОБЕЛ	Выделение всех записей

Примечание. Для отмены выделения нажмите клавишу со стрелкой противоположного направления.

Расширение выделения

Клавиши	Действие
F8	Включение режима расширения выделения (в правом нижнем углу окна появляется «ВДЛ»); последовательное нажатие F8 выделяет слово, поле, запись (только в режиме таблицы) и все записи
СТРЕЛКА ВЛЕВО или СТРЕЛКА ВПРАВО	Расширение выделения на соседние поля в том же ряду в режиме таблицы
СТРЕЛКА ВВЕРХ или СТРЕЛКА ВНИЗ	Расширение выделения на соседние строки в режиме таблицы
SHIFT+F8	Отмена предыдущего расширения выделения
ESC	Отмена режима расширения выделения

Выделение и перемещение столбцов в режиме таблицы

Клавиши	Действие
CTRL+ПРОБЕЛ	Выделение текущего столбца или отмена выделения столбца
SHIFT+СТРЕЛКА ВПРАВО	Выделение колонки справа от текущей, если она была выделена
SHIFT+СТРЕЛКА	Выделение колонки слева от текущей, если она была вы-

Клавиши	Действие
ВЛЕВО	делена
CTRL+SHIFT+F8	Включение режима перемещения; нажмите затем СТРЕЛКА ВПРАВО или СТРЕЛКА ВЛЕВО для перемещения выделенного столбца (столбцов) направо или налево

Редактирование текста и данных

Примечание. Если курсор не виден, нажмите клавишу F2 для его отображения.

Перемещение курсора внутри поля

Клавиши	Действие
СТРЕЛКА ВПРАВО	Перемещение на один знак вправо
CTRL+СТРЕЛКА ВПРАВО	Перемещение на одно слово вправо
СТРЕЛКА ВЛЕВО	Перемещение на один знак влево
CTRL+СТРЕЛКА ВЛЕВО	Перемещение на одно слово влево
END	Перемещение в конец поля, размещённого в одну строку
CTRL+END	Перемещение в конец поля, размещённого в несколько строк
HOME	Перемещение в начало поля, размещённого в одну строку
CTRL+HOME	Перемещение в начало поля, размещённого в несколько строк

Копирование, перемещение и удаление текста

Клавиши	Действие
CTRL+C	Копирование выделенного фрагмента в буфер обмена
CTRL+X	Вырезание выделенного фрагмента с копированием в буфер обмена
CTRL+V	Вставка содержимого буфера обмена в место, указываемое курсором
BACKSPACE	Удаление выделенного фрагмента или знака слева от курсора
DELETE	Удаление выделенного фрагмента или знака справа от курсора
CTRL+DELETE	Удаление всех знаков справа от курсора

Отмена изменений

Клавиши	Действие
CTRL+Z или ALT+BACKSPACE	Отмена ввода с клавиатуры
ESC	Отмена изменений в текущем поле или в текущей записи. Для отмены изменений и в поле, и в записи, нажмите клавишу ESC два раза. Первое нажатие отменяет изменения в поле, второе – в записи

Ввод данных в режиме таблицы или режиме формы

Клавиши	Действие
CTRL+; (точка с запятой)	Вставка текущей даты
CTRL+SHIFT+ : (двоеточие)	Вставка текущего времени
CTRL+ALT+ПРОБЕЛ	Вставка в поле значения по умолчанию
CTRL+' (апостроф)	Вставка значения из того же поля предыдущей записи
CTRL+ПЛЮС (+)	Добавление новой записи
CTRL+МИНУС (-)	Удаление текущей записи
SHIFT+ENTER	Сохранение изменений в текущей записи
ПРОБЕЛ	Переключение значений флажка или переключателя
CTRL+ENTER	Вставка новой строки

Обновление полей текущими данными

Клавиши	Действие
F9	Пересчёт полей в окне
SHIFT+F9	Обновление запроса (Повторный запрос. Повторное выполнение базового запроса активной формы или объекта в режиме таблицы для отображения изменений в записях, вывода добавленных записей или исключения удалённых записей) к базовым таблицам. Для подчинённой формы обновление коснется только базовых таблиц подчинённой формы
F9	Обновление содержимого подстановочного поля

Клавиши для перехода по записям

Перемещение в режиме конструктора

Клавиши	Действие
F2	Переключение между режимом редактирования (при отображении курсора) и режимом перемещения
F4	Переключение в окно свойств (в режиме конструктора формы или отчета в базе данных и проектах Microsoft Access)
F5	Переключение формы из режима конструктора в режим формы
F6	Переключение между верхней и нижней частями окна (только в режиме конструктора для окна расширенного фильтра)
F7	Переключение в построитель программ из формы или отчёта (в окне режима конструктора или окне свойств)
F8	Вызов списка полей в форме, отчёте или на странице доступа к данным (перемещает фокус на список полей, если он открыт)
SHIFT+F7	Переключение из редактора Visual Basic в режим конструктора формы или отчета
SHIFT+F7	Переключение из окна свойств элемента управления в режиме конструктора формы или отчёта на макет (фокус остаётся на элементе управления)
ALT+ENTER	Отображение окна свойств в режиме конструктора
ENTER или ALT+O	Открытие выбранной формы в режиме формы
CTRL+ENTER или ALT+K	Открытие выделенной таблицы, запроса, формы, отчёта, страницы доступа к данным, макроса или модуля в режиме конструктора
CTRL+TAB	Перемещение фокуса на подраздел, если фокус установлен на разделе
SHIFT+ENTER	Добавление элемента управления в раздел
CTRL+C	Копирование выделенного элемента управления в буфер обмена
CTRL+X	Вырезание выделенного элемента управления с копированием в буфер обмена
CTRL+V	Вставка содержимого буфера обмена в левый верхний угол выделенного раздела
СТРЕЛКА ВПРАВО	Перемещение выделенного элемента управления на один пиксел вправо по сетке страницы
СТРЕЛКА ВЛЕВО	Перемещение выделенного элемента управления на один пиксел влево по сетке страницы

Клавиши	Действие
СТРЕЛКА ВВЕРХ	Перемещение выделенного элемента управления на один пиксел вверх по сетке страницы
СТРЕЛКА ВНИЗ	Перемещение выделенного элемента управления на один пиксел вниз по сетке страницы
CTRL+СТРЕЛКА ВПРАВО	Перемещение выделенного элемента управления на один пиксел вправо (безотносительно к сетке страницы)
CTRL+СТРЕЛКА ВЛЕВО	Перемещение выделенного элемента управления на один пиксел влево (безотносительно к сетке страницы)
CTRL+ СТРЕЛКА ВВЕРХ	Перемещение выделенного элемента управления на один пиксел вверх (безотносительно к сетке страницы)
CTRL+СТРЕЛКА ВНИЗ	Перемещение выделенного элемента управления на один пиксел вниз (безотносительно к сетке страницы)
SHIFT+СТРЕЛКА ВПРАВО	Увеличение ширины на один пиксел (вправо) выделенного элемента управления
SHIFT+СТРЕЛКА ВЛЕВО	Уменьшение ширины на один пиксел (влево) выделенного элемента управления
SHIFT+СТРЕЛКА ВВЕРХ	Уменьшение высоты выделенного элемента управления на один пиксел (снизу)
SHIFT+СТРЕЛКА ВНИЗ	Увеличение высоты выделенного элемента управления на один пиксел (снизу)
SHIFT+TAB	Перевод фокуса с макета страницы на диалоговое окно адреса или темы при передаче страницы по электронной почте
ESC	Перевод фокуса из списка полей или структуры данных обратно на макет страницы доступа к данным

Перемещение в режиме таблицы

Клавиши	Действие
F5	Переход в поле номера записи Для перехода на конкретную запись достаточно ввести номер записи в это поле и нажать клавишу ENTER
TAB или СТРЕЛКА ВПРАВО	Перемещение на следующее поле
END	Переход на последнее поле текущей записи
SHIFT+TAB или СТРЕЛКА ВЛЕВО	Перемещение на предыдущее поле
HOME	Переход на первое поле текущей записи (в режиме перемещения)
СТРЕЛКА ВНИЗ	Переход на текущее поле следующей записи
CTRL+СТРЕЛКА ВНИЗ	Переход на текущее поле последней записи (в режиме перемещения)

Клавиши	Действие
CTRL+END	Переход на последнее поле последней записи (в режиме перемещения)
СТРЕЛКА ВВЕРХ	Переход на текущее поле предыдущей записи
CTRL+ СТРЕЛКА ВВЕРХ	Переход на текущее поле первой записи (в режиме перемещения)
CTRL+HOME	Переход на первое поле первой записи (в режиме перемещения)

Переход на другие экраны с данными

Клавиши	Действие
PAGE DOWN	Переход вниз на один экран
PAGE UP	Переход вверх на один экран
CTRL+PAGE DOWN	Переход вправо на один экран
CTRL+PAGE UP	Переход влево на один экран

Перемещение в подтаблицах

Клавиши	Действие
F5	Переход из подтаблицы в поле номера записи (Поле номера записи. Небольшое поле в левом нижнем углу окна таблицы или формы, в котором выводится номер текущей записи. Для перехода на конкретную запись достаточно ввести номер записи в это поле и нажать клавишу ВВОД.); затем следует ввести номер нужной записи и нажать клавишу ENTER для перехода к этой записи
CTRL+SHIFT+ СТРЕЛКА ВНИЗ	Развёртывание подтаблицы записи из таблицы
CTRL+SHIFT+ СТРЕЛКА ВВЕРХ	Свёртывание подтаблицы
TAB	Переход в подтаблицу из последнего поля предыдущей записи в таблице
SHIFT+TAB	Переход в подтаблицу из первого поля следующей записи таблицы
CTRL+TAB	Выход из подтаблицы и переход в первое поле следующей записи таблицы
CTRL+SHIFT+ TAB	Выход из подтаблицы и переход в последнее поле предыдущей записи таблицы
TAB	Переход в следующее поле таблицы из последнего поля подтаблицы
СТРЕЛКА ВНИЗ	Пропуск подтаблицы и переход к следующей записи таблицы
СТРЕЛКА ВВЕРХ	Пропуск подтаблицы и переход к предыдущей записи

Перемещение в режиме формы

Клавиши	Действие
F5	Переход в поле номера записи. Для перехода на конкретную запись достаточно ввести номер записи в это поле и нажать клавишу ENTER
TAB	Перемещение на следующее поле
SHIFT+TAB	Перемещение на предыдущее поле
END	Переход на последнее поле текущей записи
CTRL+END	Переход на последнее поле последней записи (в режиме перемещения)
HOME	Переход на первое поле текущей записи (в режиме перемещения)
CTRL+HOME	Переход на первое поле первой записи (в режиме перемещения)
CTRL+PAGE DOWN	Переход на текущее поле следующей записи
CTRL+PAGE UP	Переход на текущее поле предыдущей записи

Переходы в форме, состоящей из нескольких страниц

Клавиши	Действие
PAGE DOWN	Перемещение вниз на одну страницу; из конца записи перемещает на эквивалентную страницу следующей записи
PAGE UP	Перемещение вверх на одну страницу; из конца записи перемещает на эквивалентную страницу предыдущей записи

Переходы между главной и подчиненной формами

Клавиши	Действие
TAB	Вход в подчиненную форму из предшествующего поля в главной форме
SHIFT+TAB	Вход в подчиненную форму из следующего поля в главной форме
CTRL+TAB	Выход из подчиненной формы и переход на следующее поле или на следующую запись главной формы
CTRL+SHIFT+TAB	Выход из подчиненной формы и переход на предыдущее поле или на предыдущую запись главной формы

Перемещение в режимах предварительного просмотра и просмотра макета

Операции окна и диалогового окна

Клавиши	Действие
P или CTRL+P	Открытие диалогового окна Печать
S	Открытие диалогового окна Параметры страницы
Z	Изменение масштаба части страницы
C или ESC	Выход из режима предварительного просмотра

Просмотр различных страниц

Клавиши	Действие
F5	Переход в поле номера записи; затем следует ввести номер нужной записи и нажать клавишу ENTER для перехода к этой записи
PAGE DOWN или СТРЕЛКА ВНИЗ	Просмотр следующей страницы (при выбранном параметре Вписать в окно)
PAGE UP или СТРЕЛКА ВВЕРХ	Просмотр предыдущей страницы (при выбранном параметре Вписать в окно)

Перемещение в режимах предварительного просмотра

Клавиши	Действие
СТРЕЛКА ВНИЗ	Прокрутка вниз с небольшой скоростью
PAGE DOWN	Прокрутка вниз на один экран
CTRL+СТРЕЛКА ВНИЗ	Переход к концу страницы
СТРЕЛКА ВВЕРХ	Прокрутка вверх с небольшой скоростью
PAGE UP	Прокрутка вверх на один экран
CTRL+СТРЕЛКА ВВЕРХ	Переход к началу страницы
СТРЕЛКА ВПРАВО	Прокрутка вправо с небольшой скоростью
END или CTRL+СТРЕЛКА ВПРАВО	Переход к правому краю страницы
CTRL+END	Переход в правый нижний угол страницы
СТРЕЛКА ВЛЕВО	Прокрутка влево с небольшой скоростью
HOME или CTRL+СТРЕЛКА ВЛЕВО	Переход к левому краю страницы
CTRL+HOME	Переход в левый верхний угол страницы

Перемещение в окне схемы базы данных

Клавиши	Действие
ESC	Переход из ячейки таблицы в строку заголовка таблицы
ENTER	Переход из строки заголовка таблицы в последнюю измененную ячейку
TAB	Переход из строки заголовка таблицы в строку заголовка таблицы или переход из ячейки в ячейку внутри таблицы
ALT + СТРЕЛКА ВНИЗ	Раскрытие списка в ячейке таблицы
СТРЕЛКА ВНИЗ	Прокрутка элементов в раскрывающемся списке снизу-вверх
СТРЕЛКА ВВЕРХ	Переход к предыдущему элементу в списке
ENTER	Выбор элемента в списке и переход в следующую ячейку
ПРОБЕЛ	Изменение состояния флажка
HOME	Переход в первую ячейку в строке или переход в начало текущей ячейки
END	Переход в последнюю ячейку в строке или переход в конец текущей ячейки
PAGE DOWN	Переход на следующую «страницу» внутри таблицы или переход на следующую «страницу» схемы
PAGE UP	Переход на предыдущую «страницу» внутри таблицы или переход на предыдущую «страницу» схемы

Перемещение в конструкторе запросов

Клавиши	Действие
<i>Любая область</i>	
F6, SHIFT+F6	Переходы между областями конструктора запросов
<i>Область схемы</i>	
TAB или SHIFT+TAB	Переходы между таблицами, представлениями и функциями (и на линии связи, если имеются)
Клавиши со стрелками	Переходы между столбцами в таблице, представлении или функции
Клавиши ПРОБЕЛ или ПЛЮС	Выбор выделенного столбца данных для вывода в результатах запроса
Клавиши ПРОБЕЛ или МИНУС	Отказ от вывода выделенного столбца данных в результатах запроса
DEL	Удаление выделенной таблицы, представления, функции или линии связи из запроса

Примечание. Если выделено несколько элементов, нажатие клавиши ПРОБЕЛ действует на все элементы. Для выделения нескольких элементов щелкните эти элементы, удерживая нажатой клавишу SHIFT. Для изменения состояния выделения одного элемента щелкните его, удерживая нажатой клавишу CTRL.

Область сетки

Клавиши	Действие
Клавиши со стрелками, TAB или SHIFT+TAB	Переходы между ячейками
CTRL+СТРЕЛКА ВНИЗ	Переход в последнюю строку в текущем столбце
CTRL+СТРЕЛКА ВВЕРХ	Переход в первую строку в текущем столбце
CTRL+HOME	Переход в левую верхнюю ячейку в видимой части сетки
CTRL+END	Переход в правую нижнюю ячейку
СТРЕЛКА ВВЕРХ или СТРЕЛКА ВНИЗ	Переходы в раскрывающемся списке
CTRL+ПРОБЕЛ	Выделение столбца в сетке
F2	Переключение между режимом редактирования и режимом выделения
CTRL+C	Копирование выделенного в ячейке текста в буфер обмена (в режиме редактирования)
CTRL+X	Вырезание выделенного в ячейке текста и помещение его в буфер обмена (в режиме редактирования)
CTRL+V	Вставка текста из буфера обмена (в режиме редактирования)
INS	Переключение между режимами вставки и заборя при редактировании текста в ячейке
ПРОБЕЛ	Изменение состояния флажка в столбце «Вывод» <i>Примечание.</i> Если выделено несколько элементов, нажатие этой клавиши действует на все элементы.
DEL	Удаление выделенного содержимого
DEL	Удаление из запроса строки выделенного столбца данных <i>Примечание.</i> Если выделено несколько элементов, нажатие этой клавиши действует на все элементы.
DEL	Удаление всех значений из выделенного столбца сетки
INS (после выделения строки в сетке)	Вставка строки над выделенной строкой

СЛОВАРЬ ОСНОВНЫХ ТЕРМИНОВ

OLE (Object Linking and Embedding) – протокол связывания и встраивания объектов. Позволяет вставлять в документ или объект одного приложения (клиента OLE) объекты (OLE–объекты) других приложений, поддерживающих протокол OLE (серверов OLE). В Access объекты OLE могут быть вставлены, например, в поле, форму или отчёт.

SQL (Structured Query Language) – язык структурированных запросов для выборки, изменения и удаления данных из таблиц базы данных.

VBA (Visual Basic for Applications) – объектно–ориентированный язык программирования, разработанный компанией Microsoft для программирования в приложениях Microsoft Office. VBA является разновидностью языка Visual Basic.

Автозамена имен – средство автоматического устранения распространенных побочных эффектов, возникающих при переименовании форм, отчётов, таблиц, запросов или полей, а также элементов управления в формах и отчётах. Однако автозамена имен не позволяет исправить все ссылки на переименованные объекты.

Автоматически обновляемая связь – связь объекта OLE в Microsoft Access с OLE–сервером, благодаря которой объект в Microsoft Access автоматически обновляется после каждого изменения данных в файле объекта.

Автоформат – набор форматов, определяющих вид элементов управления и разделов формы и отчёта.

Архивариус – средство для создания отчёта с подробными сведениями об объектах базы данных.

База данных (Database) – множество данных определённой структуры, с которыми могут работать одна или несколько прикладных программ.

База данных Microsoft Access – набор данных и объектов (таких как таблицы, запросы и формы), относящихся к определённой задаче.

База данных SQL – база данных, для управления которой используется язык SQL.

База данных библиотеки – набор процедур и объектов базы данных, которые могут быть вызваны из любого приложения. Чтобы использовать элементы библиотеки, необходимо сначала определить в текущей базе данных ссылку на базу данных библиотеки.

Базовая таблица – таблица, являющаяся источником данных запроса, формы, отчёта или страницы доступа к данным. Данные в базовых таблицах можно изменять с помощью запросов на изменение.

Базовый запрос – запрос, который является источником данных для другого запроса, формы, отчёта или страницы доступа к данным.

Бланк – сетка, предназначенная для определения запроса или фильтра в конструкторе запроса или в окне расширенного фильтра.

Блокировка – состояние записи, набора записей или базы данных, при котором они доступны только для чтения всем пользователям, кроме пользователя, который изменяет их в данный момент.

Верхний колонтитул – используется для вывода названия, заголовков столбцов, даты или номера страницы вверху каждой страницы формы или отчёта. В формах верхние колонтитулы отображаются только при печати.

Вкладка объекта *Отношения* – вкладка объекта, на которой можно просмотреть, создать или изменить связи между таблицами или запросами.

Вкладка объекта *Отчёт* – окно, в котором отчёт выводится в режиме конструктора, макета или предварительного просмотра.

Вкладка объекта *Таблица* – в базе данных Microsoft Access – вкладка объекта, на которой пользователь работает с таблицами в режиме конструктора или в режиме таблицы.

Вкладка объекта *Форма* – вкладка объекта, на которой форма открывается в конструкторе, в представлении формы, в режиме таблицы или в режиме предварительного просмотра.

Владелец – при включённой системе безопасности это учётная запись пользователя, которая осуществляет управление базой данных или ее объектом. По умолчанию владельцем является учётная запись пользователя, создавшего базу данных или её объект.

Внешний ключ – одно или несколько полей (столбцов), содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ связи таблиц.

Внешняя база данных – источник импортируемых или связываемых таблиц для текущей базы данных или получатель экспортируемых таблиц.

Внешняя таблица (1) – таблица (например, «Заказы клиента»), содержащая поле внешнего ключа, которое является первичным ключом другой таблицы (например, «Клиенты»). Такое поле обычно находится на стороне «многие» отношения «один ко многим».

Внешняя таблица (2) – таблица, не принадлежащая к открытой в данный момент базе данных Microsoft Access или проекту Microsoft Access.

Внутреннее объединение – объединение таблиц в запросе, при котором в результирующий набор попадают те записи из двух связанных таблиц, у которых совпадают связанные поля.

Выбор поля – небольшой прямоугольник или полоса, при щелчке которых выделяется весь столбец в таблице.

Выбор формы – пересечение линеек в верхнем левом углу формы в конструкторе. Используется для выполнения операций на уровне формы (например, для выделения формы).

Вывод на экран – в Microsoft Access – процесс обновления или изменения изображения экране при выполнении макроса.

Выключатель – элемент управления, используемый для включения или отключения параметра в формах или отчётах. Имеет вид кнопки с текстом или рисунком и может использоваться отдельно или в группе переключателей.

Выражение (1) – любое значение или формула, возвращающая значение.

Выражение (2) – вычисляемое значение в объекте базы данных, часто синтаксически начинающееся со знака «равно» (=). Кроме операндов, может содержать математические и логические операции, вызовы функций: встроенных и определённых пользователем. Операндами могут быть константы, ссылки на элементы управления, поля таблиц и прочие объекты в базе данных и их свойства. Выражения применяются, например, в качестве значений элементов управления, а также свойств этих и других объектов. Чаще всего выражение можно построить с помощью *Построителя*.

Выражение даты – любое выражение, результат которого может быть интерпретирован как дата. К таким выражениям относятся литералы даты, числа и строки, имеющие вид даты, и даты, возвращаемые функциями.

Вычисляемое поле – поле в запросе, значение которого является результатом вычисления над другими полями этого же запроса.

Вычисляемый элемент управления – элемент управления в форме, отчёте или на странице доступа к данным, в котором выводится результат вычисления выражения. Результат пересчитывается при каждом изменении любого значения, входящего в выражение.

Главная таблица – таблица на стороне «один» при связи двух таблиц с отношением «один ко многим». В главной таблице должен существовать первичный ключ, а все записи в ней должны быть уникальными.

Группа «Пользователи» – учётная запись группы, в которую включаются все учётные записи пользователей. Microsoft Access автоматически добавляет создаваемые учётные записи пользователей в группу «Пользователи».

Группа переключателей – рамка в форме, отчёте или на странице доступа к данным, содержащая флажки, переключатели или выключатели. Группа используется для представления набора альтернативных вариантов, из которого пользователь выбирает единственный параметр.

Группировка (Grouping) – разделение данных на группы, по определённому критерию.

Данные – произвольная информация, представленная в символьной (цифровой) форме.

Двойная точность – характеристика числа, которое занимает в два раза больше места (два слова, обычно 8 байт) в памяти компьютера, чем требуется для хранения менее точного числа (числа одиночной точности). Обычно обрабатывается компьютером в виде числа с плавающей запятой.

Двухбайтовая кодировка – кодировка, в которой для представления знака используется 1 или 2 байта, что позволяет представить более 256 знаков.

Диаграмма – графический образ, отражающий количественную зависимость данных. Средство графического представления данных в форме, отчёте или на странице доступа к данным.

Диалоговое окно – один из основных элементов организации приложения.

Диалоговое окно настраиваемых свойств – окно свойств, которое позволяет пользователям задавать свойства элементов ActiveX.

Заголовок группы – раздел отчёта или страницы доступа к данным, содержащий элементы группировки. Используется для вывода данных, таких как название группы или итоговые значения для группы, в начале группы записей.

Заголовок отчёта – область отчёта, используемая для вывода данных (таких как заголовок, дата или введение) в начале отчёта.

Заголовок раздела – горизонтальная полоса над разделом формы или отчёта в режиме конструктора. В заголовке раздела отображается тип и имя раздела. Используется для доступа к окну свойств раздела.

Заголовок формы – область формы, используемая для вывода названия формы, инструкций по работе с формой или кнопок, открывающих связанные формы или выполняющих другие задачи. Заголовок формы выводится в верхней части представления формы и в начале ее распечатки.

Запись – структурированный тип данных, компоненты которого называются полями, имеют имя и тип.

Заполнение – вывод отчёта в масштабе, в котором окно моментального снимка отчёта заполняется по высоте или ширине страницы, в зависимости от того, имеет ли отчёт книжную или альбомную ориентацию.

Запрос – обращение к СУБД, содержащее задание на выборку, добавление, изменение или удаление записей. Вопрос о данных, хранящихся в таблицах, или требование на выполнение определённых действий с данными. Запрос позволяет объединить данные, находящиеся в разных таблицах, использовать их в качестве источника данных для формы или отчёта.

Запрос на выборку – запрос, в котором задаётся вопрос о данных, хранящихся в таблицах, и возвращается результирующий набор в форме таблицы без изменения самих данных.

Запрос на добавление – запрос на изменение, добавляющий записи из результирующего набора в конец существующей таблицы.

Запрос на изменение – запрос, в котором выполняется копирование или изменение данных. В число запросов на изменение входят запросы на добавление, удаление, создание таблицы и обновление. Имена таких запросов обозначены восклицательным знаком (!) в области переходов.

Запрос на обновление – запрос на изменение (инструкция SQL), в котором изменяется набор записей, удовлетворяющих указанному условию поиска.

Запрос на объединение – запрос, в котором используется оператор UNION для объединения результатов нескольких запросов на выборку.

Запрос на создание таблицы – запрос (инструкция SQL), создающий новую таблицу, в которую копируются записи (строки) из существующей таблицы или результаты запроса.

Запрос на удаление – запрос (инструкция SQL), который удаляет записи, удовлетворяющие указанному условию, из одной или нескольких таблиц.

Запрос с параметрами – запрос, в котором одно или несколько условий вводятся в интерактивном режиме пользователем. Запрос с параметрами не является отдельным типом запроса; он расширяет гибкость обычного запроса.

Защита на уровне пользователя – при включении защиты на уровне пользователя в базе данных Microsoft Access администратор базы данных или владелец объекта предоставляет определенные разрешения отдельным пользователям и группам пользователей на следующие объекты: таблицы, запросы, формы, отчеты и макросы.

Защищённая рабочая группа – рабочая группа Microsoft Access, для входа в которую необходимо ввести имя пользователя и пароль и в которой доступ к объектам базы данных ограничен разрешениями, предоставленными отдельным учётным записям пользователей и групп.

Знак выбора всех столбцов – прямоугольник в верхней части столбца. Для выделения всего столбца в бланке запроса или фильтра достаточно щелкнуть знак выбора всех столбцов.

Значение Null – значение, которое можно ввести в поле или использовать в выражениях и запросах для указания отсутствующих или неизвестных данных. В Visual Basic ключевое слово **Null** указывает значение Null. Некоторые поля, такие как поля первичного ключа, не могут содержать значения Null.

Значение по умолчанию – значение, которое автоматически вводится в поле или в элемент управления при добавлении новой записи. Можно принять значение по умолчанию или ввести вместо него любое другое.

Импорт – копирование данных из текстового файла, файла электронной таблицы или базы данных в таблицу Microsoft Access. Можно использовать импортированные данные для создания таблицы или добавить их в существующую таблицу с подходящей структурой.

Имя класса – имя, используемое для ссылок на модуль класса. Если этот модуль класса является модулем формы или отчёта, то перед именем класса указывается тип модуля, например, `Form_OrderForm`.

Индекс – средство, обеспечивающее быстрый доступ к данным в таблице на основе значений одного или нескольких столбцов. Индекс представляет собой упорядоченный список значений и ссылок на те записи, в которых хранятся эти значения.

Индексы списка – последовательность номеров элементов списка (0 для первого элемента списка, 1 для второго и т. д.).

Индикатор развёртывания – кнопка для развёртывания и свёртывания групп записей; на кнопке отображается знак «плюс» (+) или «минус» (–).

Источник данных – совокупность данных, являющихся источником для представления и обработки данных в приложениях, связанных с базами данных. В Access источником данных называют таблицу или запрос, на основе данных из которых заполняются форма, отчёт, страница доступа к данным, элемент управления или другой объект.

Источник записей – базовый источник данных для формы, отчёта или страницы доступа к данным. В базе данных Microsoft Access источником записей может быть таблица, запрос или инструкция SQL.

Итоговый запрос – запрос, в котором выводятся результаты статистических расчётов, например, среднего или суммы значений различных полей из одной или нескольких таблиц. Итоговые запросы не являются отдельным типом запросов; скорее, они расширяют гибкость запроса на выборку.

Каскад – процесс активации действия другим действием. Например, когда между несколькими таблицами определено отношение каскадного обновления, обновление первичного ключа в главной таблице автоматически активирует изменения во внешней таблице.

Каскадное обновление – обновление всех связанных записей в связанной таблице или таблицах при изменении записи в главной таблице (при наличии отношений, обеспечивающих целостность данных между таблицами).

Каскадное событие – последовательность событий, возникающих, если процедура обработки событий прямо или косвенно вызывает сама себя. Эту последовательность называют также каскадом событий или рекурсивной. При использовании каскадных событий следует проявлять осторожность, поскольку они часто приводят к переполнению стека или другим ошибкам во время выполнения.

Каскадное удаление – средство поддержания целостности данных в связанных таблицах, которое при удалении записи в главной таблице обеспечивает удаление всех связанных записей в подчинённой таблице.

Ключевое поле – поле, которое однозначно идентифицирует каждый объект в таблице, т. е. позволяет чётко отличить один объект от другого. Может быть частью составного ключа.

Кнопка – элемент управления, используемый для запуска макроса, вызова функции Visual Basic или запуска процедуры обработки событий. В других программах кнопки иногда называют командными кнопками.

Кнопки перехода – кнопки, используемые для перемещения по записям. Находятся в левом нижнем углу окон режима таблицы и представления формы. Кнопки перехода доступны также в окне предварительного просмотра и используются для перемещения между страницами документа.

Кнопочная форма – средство, обеспечивающее пользователям доступ к функциям приложения.

Колонтитул отчёта – область отчёта, используемая для сведений, которые следует напечатать в конце отчёта, таких как дата, номера страниц и итоговые значения.

Конструктор – режим, в котором отображается макет следующих объектов базы данных: таблицы, запросы, формы, отчёты и макросы. В конструкторе можно создавать новые объекты базы данных или изменять макет существующих.

Контекстное меню – меню специального вида, которое появляется при нажатии правой кнопки мыши.

Конфликт – конфликт, который возникает при пакетных обновлениях. Клиент считывает данные с сервера, а затем пытается изменить их в пакетном обновлении, но при этом до выполнения попытки обновления другой клиент изменяет исходные данные на сервере.

Литерал даты – любая последовательность знаков с допустимым форматом, до и после которой есть знаки #. Допустимым является формат даты, заданный в параметрах языкового стандарта программы, или универсальный формат даты.

Личный код – строка, включающая от 4 до 20 букв и цифр (с учётом регистра), которая вместе с именем учётной записи определяет пользователя или группу в рабочей группе Microsoft Access.

Макрокоманда – основной стандартный блок макроса; автономная инструкция, которая может применяться отдельно или в сочетании с другими макрокомандами для автоматизации задач. В других языках макросов макрокоманды иногда называют просто командами.

Макрос – макрокоманда или набор макрокоманд, используемый для автоматизации задач, которые выполняются либо последовательно, либо в порядке, заданном определёнными условиями.

Максимальное число записей – для повышения производительности в проекте Microsoft Access можно указать максимальное число записей, загружаемых в форму или в таблицу из базы данных Microsoft SQL Server.

Маркер данных – отрезок, область, точка, сегмент или другой объект диаграммы, соответствующий одной точке данных или значению. Связанные маркеры данных на диаграмме образуют ряд данных.

Маркер перемещения – большой квадрат в верхнем левом углу выделенного элемента управления в режиме конструктора или макета. Для перемещения элемента управления или макета элемента управления в другое место можно перетащить маркер перемещения.

Маркер развертывания – кнопка на странице доступа к данным, предназначенная для отображения и скрытия содержимого одного уровня группировки.

Маркеры размера – небольшие квадраты по краям выделенного элемента управления, раздела формы или отчёта, с помощью которых можно менять размеры формы или элемента управления.

Маска ввода – формат, состоящий из постоянных символов (таких как скобки, точки или дефисы) и символов маски, указывающих, в какие позиции, в каком количестве и какого типа данные можно вводить.

Мастера объектов – мастера для создания объектов базы данных Access (*форма, запрос, отчёт*).

Мастера элементов управления – мастера создания встроенных элементов управления и элементов ActiveX для добавления их в формы, отчёты и страницы доступа к данным.

Метка – элемент управления, в котором в форме, отчёте или на странице доступа к данным выводится поясняющий текст, например заголовок, подпись или инструкция. Метки могут быть свободными или связанными с другими элементами управления.

Метка данных – подпись с дополнительными сведениями о маркере данных, представляющем одну точку данных или значение.

Многозначное поле – поле подстановки, в котором может храниться несколько значений.

Многопользовательская (общая) база данных – база данных, в которой несколько пользователей могут одновременно получать доступ к одному и тому же набору данных и изменять его.

Модель данных – совокупность принципов организации базы данных. В СУБД Access используется реляционная модель данных.

Моментальный снимок – статическое отображение набора данных, такого как записи, возвращённые в запросе.

Моментальный снимок отчёта – файл (с расширением имени файла SNP), содержащий точную копию каждой страницы отчёта Microsoft Access с сохранением двумерного макета, графики и других внедрённых объектов.

Монопольный доступ – режим доступа к базе данных, совместно используемой в сети. Если база данных открыта в режиме монопольного доступа, другие пользователи не могут открыть её.

Набор вкладок – элемент управления, с помощью которого создаются формы или диалоговые окна, содержащие несколько различных страниц, каждая со своей вкладкой, на которых расположены сходные элементы управления, например, текстовые поля или переключатели. Когда пользователь щелкает вкладку, страница становится активной.

Набор записей – общий термин для объектов Recordset табличного, динамического и моментального типа, представляющий набор записей, который имеет характеристики объекта.

Навигационные кнопки – кнопки, расположенные на нижней границе формы или таблицы и используемые для перемещения по записям.

Настраиваемая группа – элемент поля настраиваемых групп. Пользовательская группа содержит несколько элементов из поля строк или столбцов.

Настраиваемая панель инструментов – в Microsoft Access 2003 и более ранних версиях – панель инструментов, создаваемая пользователем для собственного приложения. В отличие от неё, стандартная панель инструментов устанавливается вместе с Microsoft Access.

Нижний колонтитул – используется для вывода итоговых значений по странице, даты или номера страницы внизу каждой страницы формы или отчёта. В формах нижние колонтитулы отображаются только при печати.

Нормализация – минимизация количества повторяющихся данных в реляционной базе данных за счёт более эффективной структуры таблиц. Нормализацию базы данных можно выполнить с помощью мастера анализа таблиц.

О – область формы, используемая для вывода инструкций по работе с формой, кнопок или свободных элементов управления, в которые можно вводить данные. Выводится в нижней части представления формы и в конце её распечатки.

Область выделения отчёта – пересечение линеек в верхнем левом углу отчёта в конструкторе. Используется для выполнения операций на уровне отчёта, таких как его выделение.

Область выделения раздела – поле слева от строки раздела в режиме конструктора. Используется для выполнения операций на уровне раздела, таких как его выделение.

Область выделения строки – небольшое поле или прямоугольник, при щелчке по которым выделяется вся строка в таблице или в режиме конструктора макроса, а также при сортировке и группировке записей в конструкторе отчёта.

Область данных – часть представления режима «Сводная таблица» или «Сводная диаграмма», содержащая итоговые данные. Значения в области данных отображаются как записи в режиме «Сводная таблица» или как точки данных в режиме «Сводная диаграмма».

Область зависимостей объектов – в данной области отображаются объекты, зависящие от выделенного, а также объекты, от которых зависит выделенный объект.

Область маркировки – небольшое поле или прямоугольник с левого края записи в режиме таблицы или представлении формы, при выборе которого выделяется вся запись.

Область переходов – область, которая открывается при открытии базы данных Microsoft Access или проекта Microsoft Access. В области переходов отображаются объекты базы данных. Её можно настраивать различными способами для сортировки и группировки объектов.

Область списка полей – область, в которой перечислены все поля базового источника записей или объекта базы данных.

Область столбцов – часть режима «Сводная таблица», содержащая поля столбцов.

Область строк – часть представления «Сводная таблица», содержащая поля строк.

Область фильтра – часть режима «Сводная таблица» или «Сводная диаграмма», содержащая поля фильтра.

Обновление (1) – в базе данных Microsoft Access – повторный вывод записей в форме или таблице с учетом изменений, выполненных другими пользователями. В проекте Microsoft Access – повторное выполнение базового запроса для активной формы или таблицы с целью отображения изменений в записях.

Обновление (2) – обновление содержимого экрана.

Обновление (3) – принятие изменений данных в записи. Изменения сохраняются в базе данных при переходе на новую запись в форме или таблице, а также при сохранении записи.

Общая строка меню – в Microsoft Access 2003 и более ранних версиях – специальная настраиваемая строка меню, заменяющая стандартные строки меню во всех окнах приложения базы данных за исключением тех, для которых определены настраиваемые строки меню для формы или отчёта.

Объект Microsoft Access – объект, определённый в Microsoft Access, который относится к приложению Access, его интерфейсу или формам и отчётам приложения. Кроме того, объект Microsoft Access можно использовать для программирования элементов интерфейса для ввода и отображения данных.

Объект OLE – объект, поддерживающий протокол OLE для связывания и внедрения. Объект OLE из OLE-сервера (например, рисунок Windows Paint или лист Microsoft Excel) можно связать с полем, формой или отчётом либо внедрить в них.

Объектная модель – совокупность взаимосвязанных объектов, описывающих программную систему.

Объекты DAO – Объекты базы данных – в базу данных Microsoft Access могут входить такие объекты, как таблицы, запросы, формы, отчёты, страницы, макросы и модули.

Обычный порядок сортировки – порядок сортировки по умолчанию, определяющий сортировку символов во всей базе данных, т. е. в таблицах, запросах и отчётах. Обычный порядок сортировки следует применять, если

планируется использовать базу данных в версиях Microsoft Access для разных языков.

Ограничение – ограничение, налагаемое на значения, которые могут вводиться в столбец или в строку. Например, можно запретить ввод в поле «Возраст» значений меньше 0 или больше 110.

Окно базы данных – в Microsoft Access 2003 и более ранних версиях – окно, которое открывается при открытии базы данных или проекта Microsoft Access. В нем выводятся ярлыки для создания объектов базы данных и открытия существующих объектов. В Microsoft Access 2007 и более поздних версиях окно базы данных заменено областью переходов.

Окно базы данных – окно, которое содержит списки всех объектов базы данных и обеспечивает доступ к этим объектам.

Окно запросов – окно, в котором запрос открывается в конструкторе, в режиме таблицы, в режиме SQL, или в режиме предварительного просмотра.

Окно расширенного фильтра – окно, в котором можно самостоятельно создать фильтр. В бланке вводятся условия отбора, с помощью которых из всех записей в открытой форме или таблице выделяется подмножество, удовлетворяющее данным условиям.

Окно свойств – окно, предназначенное для просмотра и изменения свойств таблиц, запросов, форм, отчетов, страниц доступа к данным и элементов управления.

Операнд – элемент данных, участвующий в операции.

Оператор – допустимое в языке программирования высокого уровня предложение, задающее целостное законченное действие или описание.

Оператор сравнения – оператор, выполняющий сравнение двух значений или выражений. Например, < (меньше), > (больше) и = (равно).

Операция – действие, выполняемое над данными.

Определение данных – поля в базовых таблицах и запросах, а также выражения, образующие источник записей для страницы доступа к данным.

Основная форма – форма, содержащая одну или несколько подчинённых форм.

Отключенный элемент управления – элемент управления, который затенён в форме. Отключённый элемент управления не может получать фокус и не реагирует на щелчки мыши.

Отладка – поиск и исправление ошибок в макросах или процедурах VBA.

Отношение – связь, установленная между общими полями (столбцами) в двух таблицах. Существуют отношения «один к одному», «многие ко многим» и «один ко многим».

Отношение «многие ко многим» – если две таблицы связаны отношением «многие ко многим», то одной записи в каждой таблице могут соответ-

ствовать несколько записей в другой таблице. Для установления отношения «многие ко многим» необходимо создать третью таблицу и добавить в неё поля первичного ключа из обеих таблиц.

Отношение «один к одному» – связь двух таблиц, при которой значению первичного ключа каждой записи в главной таблице соответствует значение В
в связанном поле или полях в одной (и только в одной) записи связанной таблицы.

Отношение «один ко многим» – связь двух таблиц, при которой значению первичного ключа каждой записи в главной таблице соответствует значение в связанном поле или полях в нескольких записях связанной таблицы.

Отчёт – объект базы данных Microsoft Access, предназначенный для вывода на печать данных, организованных и отформатированных в соответствии с требованиями пользователя. С помощью отчётов составляются сводки по продажам, списки телефонов и почтовые наклейки.

Панель инструментов – один из видов панелей команд.

Панель команд – панель, на которой могут размещаться элементы управления.

Панель элементов – набор инструментов для создания элементов управления в форме, отчёте или страницах доступа к данным в режиме конструктора.

Первичный ключ – одно или несколько полей (столбцов), значения которых однозначно определяют каждую запись в таблице. Первичный ключ не допускает значений Null и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.

Переключатель – элемент управления, который обозначает альтернативный вариант. Такие элементы управления обычно используются в виде группы переключателей. При этом пользователь может выбрать только один переключатель.

Перекрёстный запрос – запрос, в котором подсчитывается сумма, среднее, число значений или другое итоговое значение, после чего результаты группируются по двум наборам данных, один из которых выводится в левой части, а другой – в верхней части таблицы.

Переменная процедуры – переменная, которая объявлена внутри процедуры. Переменные процедуры всегда являются закрытыми для процедуры, в которой они объявлены.

Перемещаемый – допускающий перемещение в собственном окне. Перемещаемое окно всегда выводится поверх других окон. Перемещаемыми являются построитель выражений, Архивариус, панель элементов и палитры.

Повторный запрос – повторное выполнение базового запроса для активной формы или таблицы в целях отображения изменений в записях, вывода добавленных записей или исключения удалённых записей.

Подмножество – набор записей, который определяется таблицей, запросом или инструкцией SQL. Статистические функции по подмножеству возвращают сведения о конкретном подмножестве или наборе записей.

Подсказки – краткие описания названий команд и кнопок на ленте, которые появляются при наведении на них указателя мыши.

Подстановочные знаки – эти знаки используются в запросах и выражениях для включения всех записей, имён файлов или других элементов, которые начинаются с определённых знаков или соответствуют заданному образцу.

Подтаблица – таблица, вложенная в другую таблицу и содержащая данные, связанные или объединённые с первой таблицей.

Подчинённая форма, подчинённый отчёт – форма (отчёт), которая встраивается в главную форму (отчёт).

Поле (1) – элемент данных в записи.

Поле (2) – элемент управления, предназначенный для отображения текстовых или числовых данных.

Поле гиперссылки – поле, в котором хранятся адреса гиперссылок. В базе данных Microsoft Access это поле с типом данных гиперссылки. В проекте Microsoft Access это поле, у которого свойство *IsHyperlink* имеет значение *True*.

Поле итогов – поле, отображающее итоговое значение по данным из базового источника записей. Для расчёта итоговых значений в поле может использоваться статистическая функция, такая как *Sum* или *Count*, или выражение.

Поле подстановки – поле, используемое в форме или отчёте в базе данных Microsoft Access, в котором либо выводится список значений, загружаемый из таблицы или запроса, либо хранится статический набор значений.

Поле рядов – поле, выводящееся в области ряда диаграммы, которое содержит элементы ряда. Ряд представляет набор связанных между собой точек данных.

Поле с номером записи – небольшое поле в левом нижнем углу режима таблицы или представления формы, в котором выводится номер текущей записи. Для перехода на конкретную запись достаточно ввести в это поле её номер и нажать клавишу ВВОД.

Поле со значением Null – поле, содержащее значение Null. Такое поле отличается от поля, содержащего пустую строку (""), или значение 0.

Поле со списком – элемент управления в форме, являющийся сочетанием текстового поля и списка. Пользователь может ввести значение в это поле или раскрыть список и выбрать из него значение.

Пользовательская функция – запрос, который принимает входные параметры и возвращает результат аналогично хранимой процедуре. Типы: скалярная (несколько инструкций; возвращает одно значение), встроенная (одна инструкция; обновляемое табличное значение), табличная (несколько инструкций; табличное значение).

Пользовательский объект – настраиваемый объект, описанный в модуле класса формы или отчёта. В модуле класса можно создавать методы и свойства нового объекта, создавать новые экземпляры объекта и управлять им с помощью этих свойств и методов.

Пользовательский порядок – порядок сортировки, определяемый пользователем.

Построитель – средство Microsoft Access, упрощающее выполнение конкретной задачи. Например, сложные выражения быстро создаются с помощью построителя выражений.

Построитель выражений – средство Microsoft Access, предназначенное для создания выражений. Построитель включает список часто используемых выражений.

Построитель макросов – вкладка объекта, на которой создаются и изменяются макросы. Построитель макросов можно запускать из различных объектов, таких как форма или отчёт, или напрямую с вкладки **Создать** на ленте.

Правило проверки – свойство, определяющее допустимые входные значения для поля или записи в таблице, а также для элемента управления в форме. Когда это правило нарушается, Microsoft Access выводит сообщение, текст которого задаётся в свойстве *ValidationText*.

Представление формы – представление, в котором форма выводится для отображения или ввода данных. Это основное представление для ввода или изменения данных в таблицах. В этом режиме также можно изменить макет формы.

Приложение базы данных – набор объектов, среди которых могут быть таблицы, запросы, формы, отчёты, макросы и модули кода, которые используются вместе и упрощают использование базы данных. Обычно приложение базы данных развёртывается для группы пользователей.

Примечание группы – используется для вывода данных, таких как название группы или итоговые значения для группы, в конце группы записей.

Примечание группы – раздел отчёта, в котором выводятся итоговые значения группы.

Присоединенная рамка объекта – элемент управления в форме или отчёте, используемый для отображения объектов OLE, хранящихся в таблицах, и управления ими.

Присоединенный столбец – столбец в списке, поле со списком или раскрывающемся списке, присоединённый к полю, которое указано в свойстве *ControlSource* элемента управления.

Присоединенный элемент управления – элемент управления в форме, отчёте или на странице доступа к данным, в котором выводятся данные из поля базовой таблицы, запроса или инструкции SQL.

Проверка – процесс проверки введённых данных на соответствие определённым условиям или ограничениям.

Прямоугольник выделения – прямоугольник, образуемый выделенными строками (записями) и столбцами (полями) в режиме таблицы.

Пустая строка – строка, не содержащая символов. Пустая строка используется для указания того, что значение для этого поля отсутствует. Для ввода пустой строки с клавиатуры следует ввести два символа прямых кавычек без пробела ("").

Раздел – часть формы или отчёта, например, заголовок, колонтитул или раздел подробностей.

Раздел подписей – раздел страницы доступа к данным с группировкой, используемый для отображения заголовков столбцов данных. Выводится перед заголовком группы. В раздел подписей невозможно добавить связанный элемент управления.

Раздел подробностей – основной раздел формы или отчёта. Этот раздел обычно содержит элементы управления, связанные с полями в источнике записей. Кроме того, он может содержать свободные элементы управления, такие как метки, определяющие содержимое полей.

Разделители компонентов даты – знаки, используемые для разделения дня, месяца и года при форматировании значений дат. Эти знаки определяются параметрами системы или задаются с помощью функции **Формат**.

Разделитель – символ, используемый для разделения текста или чисел.

Разделитель строк – Текстовые символы, используемые для выделения строки внутри другой строки. Разделителями строк являются символы одинарных (') и прямых (") кавычек.

Разрешения – набор атрибутов, определяющих доступ пользователя к данным или объектам в базе данных.

Раскрывающийся список – элемент управления на странице доступа к данным, при щелчке которого выводится список, в котором можно выбрать значение. Ввести значение в раскрывающийся список невозможно.

Режим SQL – вкладка объекта, на которой выводится инструкция SQL текущего запроса или которая используется для создания запроса SQL (запроса на объединение, запроса к серверу или запроса определения данных). При создании запроса в режиме конструктора Microsoft Access автоматически создаёт эквивалентную инструкцию SQL в режиме SQL.

Режим макета – режим в Microsoft Access, в котором при просмотре динамических данных можно изменять различные параметры структуры форм и отчётов.

Режим перемещения – режим, в котором можно перемещать столбцы в режиме таблицы с помощью клавиш со стрелками ВЛЕВО и ВПРАВО.

Режим таблицы – режим, в котором данные из таблицы, формы, запроса, представления или хранимой процедуры выводятся в формате строк и столбцов. В режиме таблицы можно изменять поля, добавлять или удалять данные

и выполнять их поиск. В Microsoft Access в этом режиме также можно изменять и добавлять поля таблицы.

Реляционная СУБД – СУБД, базирующаяся на реляционной модели данных, в которой связи между наборами данных реализованы на основании совпадения значений полей.

Ряд данных – набор связанных между собой точек данных на диаграмме. Каждому ряду данных на диаграмме соответствует отдельный цвет или шаблон. Диаграммы могут содержать несколько рядов данных.

Свободная рамка объекта – элемент управления в форме или отчёте, в который помещают свободный объект. Свободным объектом является, например, рисунок, значение которого не основано на данных в таблице.

Свободная форма или отчёт – формы или отчёты, не связанные источниками записей, такими как таблица, запрос или инструкция SQL. (Свойство *RecordSource* формы или отчёта является пустым).

Свободный элемент управления – элемент управления в форме, отчёте или на странице доступа к данным, не связанный с данными.

Сводная таблица – веб-компонент Microsoft Office, который используется для анализа данных в интерактивном режиме на веб-странице. Данные, которые отображаются в формате строк и столбцов, можно перемещать, фильтровать, сортировать и рассчитывать так, как требуется пользователям.

Свойства документа – свойства, такие как название, тема и имя автора, которые сохраняются для каждой страницы доступа к данным.

Свойства страницы доступа к данным – атрибуты страницы доступа к данным, определяющие базу данных, к которой она подключена, а также её внешний вид и характеристики.

Свойства таблицы – в базе данных Microsoft Access – атрибуты таблицы, определяющие её внешний вид или её общие характеристики. Значения свойств таблицы задаются в режиме конструктора таблицы аналогично свойствам полей.

Свойства формы – атрибуты формы, определяющие её внешний вид или характеристики. Например, свойство формы *DefaultView* определяет, будет ли форма автоматически открываться в представлении формы или же в режиме таблицы.

Свойство – характеристика объекта, которая может принимать определённое значение.

Свойство по умолчанию – свойство, которое можно задать для элемента управления, чтобы все созданные элементы управления такого типа имели одно и то же значение.

Связанная таблица (1) – таблица, которая хранится в файле, не принадлежащем к открытой базе данных, но записи которой доступны Microsoft Access. Можно добавлять, удалять и изменять записи в связанной таблице, но изменять её структуру нельзя.

Связанная таблица (3) – внешняя таблица, которая доступна из текущего файла базы данных.

Связанный рисунок – элемент управления в форме, отчёте или на странице доступа к данным, используемый для связи рисунка с полем объекта OLE в базе данных Access или со столбцом типа image в проекте Access.

Связанный элемент управления – элемент управления в форме, отчёте или на странице доступа к данным, используемый для изменения или отображения данных из базовой таблицы, запроса или инструкции SQL. Имя поля, с которым связан элемент, хранится в свойстве *ControlSource* элемента управления.

Связывание (таблиц) – создание подключения к данным в другой программе, которое позволяет просматривать и изменять данные и в исходной программе, и в Microsoft Access.

Связь – логическое отношение между объектами, представленными таблицами.

Связь OLE/DDE – связь между объектом OLE и его OLE–сервером или между исходным и конечным документами при связи DDE.

Связь, обновляемая вручную – связь, при которой для обновления данных после их изменения в исходном документе требуются действия пользователя.

Сгруппированные элементы управления – несколько элементов управления, которые обрабатываются как единое целое при разработке формы или отчёта. При размещении элементов или задании их свойств можно выбрать группу, не выделяя элементы по отдельности.

Сетка (режим таблицы) – вертикальные и горизонтальные линии, визуально разделяющие столбцы и строки данных на ячейки в таблице, запросе, форме, представлении или хранимой процедуре. Можно скрывать и отображать линии сетки.

Система управления базами данных (СУБД) – комплекс программ и языковых средств, предназначенных для создания, ведения и использования баз данных.

Соединение – одна из операций над таблицами баз данных.

Сортировка – изменение порядка, в котором представлены данные.

Составной ключ – первичный ключ таблицы, который состоит из нескольких полей.

Спецификация импорта и экспорта – спецификация, в которой хранятся сведения, необходимые Microsoft Access для импорта или экспорта текстовых файлов с полями фиксированного размера или разделителями.

Список макрокоманд – список, который вызывается с помощью стрелки в столбце *Макрокоманда* вкладки объекта *Макрос*.

Список объектов – область в левой части окна базы данных, в которой отображаются все объекты Access.

Стандартная панель инструментов – в *MS Access 2003* и более ранних версиях – панель инструментов, устанавливаемая как компонент пользовательского интерфейса Microsoft Access. В отличие от неё настраиваемая панель инструментов создаётся пользователем для собственного приложения базы данных. В Microsoft Access 2007 и более поздних версиях панели инструментов заменила лента, на вкладках которой команды располагаются в связанных группах. Кроме того, часто используемые команды можно добавлять на панель быстрого доступа.

Стандартное отклонение – параметр, определяющий распределение функции вероятности вокруг среднего значения, который равняется корню квадратному моменту квадрата отклонения.

Статистическая функция – функции, такие как *Sum*, *Count*, *Avg* или *Var*, используемые для расчётов итоговых значений.

Стиль элемента управления по умолчанию – значение свойства по умолчанию для элемента управления. Тип элемента управления имеет смысл задавать перед созданием нескольких похожих элементов управления, чтобы не настраивать их по отдельности.

Столбец – часть таблицы базы данных, в которой хранятся данные определённого типа. Кроме того, столбец является наглядным представлением поля таблицы и (в базе данных Microsoft Access) бланка запроса или фильтра.

Столбец подстановок – список значений, которые могут быть введены в текстовое или числовое поле. Список значений может быть, как фиксированным, так и может содержаться в таблице или запросе.

Страница (хранилище данных) – часть файла базы данных, в которой хранятся данные записи. В зависимости от размеров записей, страница (имеющая размер 4 КБ) может содержать несколько записей.

Страница доступа к данным – веб-страница, предназначенная для просмотра данных и работы с ними через Интернет или интрасеть. Обычно эти данные хранятся в базе данных Microsoft Access.

Страница доступа к данным с группировкой – страница доступа к данным с несколькими уровнями группы.

Страница свойств – область, предназначенная для просмотра и изменения свойств таблиц, запросов, полей, форм, отчётов, страниц доступа к данным и элементов управления.

Строка – последовательность символов, заключённая в двойные кавычки.

Строка итогов – строка таблицы, в которой отображаются итоговые данные для каждого поля в зависимости от типа данных в этом поле.

Строка меню (Menu Bar) – панель команд, на которой расположены элементы управления, представляющие собой раскрывающиеся списки команд меню.

Строка подключения – строковое выражение, используемое для открытия внешней базы данных.

Схема базы данных – графическое представление любой части схемы базы данных. Схема может представлять полную или частичную структуру базы данных. Она включает таблицы, содержащиеся в них столбцы и отношения между таблицами.

Схема данных – графическое представление структуры базы данных Access.

Таблица – объект базы данных, в котором данные хранятся в виде записей (строк) и полей (столбцов). Данные в отдельной таблице обычно относятся к определённой категории (например, сведения о сотрудниках или заказах).

Текстовое поле – элемент управления, который используется в формах или отчётах для отображения или ввода данных. Может иметь метку.

Текстовый тип данных (Короткий текст) – тип данных поля в базе данных Microsoft Access. Текстовые поля могут содержать до 255 знаков или меньшее число знаков, указанное в значении свойства поля.

Текстовый файл с полями фиксированного размера – файл, содержащий данные, в котором каждое поле имеет фиксированную ширину.

Текущая запись – запись в наборе записей, в которой в данный момент можно изменять или читать данные. Текущей записью в наборе является только одна из записей. Однако в наборе записей может и не быть текущей записи, например, после удаления записи из набора записей типа «динамическое подмножество данных».

Тип данных «да/нет» – тип данных поля, при котором поле может содержать только одно из двух значений, например «Да» и «Нет» или «Истина» и «Ложь». Значения Null не допускаются.

Тип данных «Дата/время» – тип данных баз данных Microsoft Access, предназначенный для хранения значений даты и времени.

Тип данных «объект OLE» – тип данных поля, используемый для объектов, созданных в других программах, которые могут быть связаны с базой данных Microsoft Access или внедрены в неё.

Тип данных «Счетчик» – Тип данных поля в базе данных Microsoft Access, при котором для каждой добавляемой в таблицу записи автоматически сохраняется уникальный номер.

Тип данных (Data Type) – атрибут поля в таблице, который определяет, какие данные могут содержаться в этом поле.

Тип объекта – тип объекта, предоставляемый приложением через автоматизацию, например Application (приложение), File (файл), Range (диапазон) или Sheet (лист). Полный список доступных объектов см. в окне просмотра объектов в редакторе Visual Basic или в документации на программу.

Типы данных поля – характеристика поля, определяющая, какие данные могут в нем храниться. Например, в поле с типом «Текст» можно вводить как текст, так и числа, а в полях с типом «Число» допускается хранение только числовых данных.

Точка ряда – отдельное значение, отображаемое на диаграмме в виде полосы, столбца, линии, сектора, кольцевого сектора или другого маркера данных.

Уникальный индекс – индекс, определенный для поля путем задания для свойства *Индексированное поле* значения *Да* (*Совпадения не допускаются*). При этом ввод в индексированное поле повторяющихся значений становится невозможным. Для ключевых полей уникальный индекс создается автоматически.

Уровень группировки – уровень вложенности группы в отчете или на странице доступа к данным.

Уровень группы – уровень вложенности конкретной группы в отчете или на странице доступа к данным. Вложенные группы образуются при группировке наборов записей по нескольким полям, выражениям или источникам записей группы.

Уровень модуля – уровень модуля описывает все переменные и константы, объявленные в разделе описаний модуля Visual Basic для приложений (VBA) или вне процедуры. Переменные или константы, описанные на уровне модуля, являются доступными для всех процедур в том же модуле.

Уровень процедуры – описывает любые переменные или константы, объявленные в процедуре. Такие переменные и константы доступны только внутри процедуры.

Условная фильтрация – фильтрация поля для отображения *n* первых или последних элементов на основании итогового значения. Например, с помощью фильтра можно отобрать три города с максимальными показателями продаж или пять товаров, дающих наименьшую прибыль.

Условное форматирование – форматирование содержимого элемента управления в форме или отчете на основании одного или нескольких условий. Условие может содержать ссылку на другой элемент управления, элемент управления, имеющий фокус, или на определяемую пользователем функцию Visual Basic для приложений.

Условные обозначения – поле, определяющее цвет или шаблон для ряда данных или категории на диаграмме.

Учетная запись пользователя – учётная запись пользователя, определяемая именем и личным кодом, которая создаётся для управления разрешениями пользователя на доступ к объектам базы данных в рабочей группе Microsoft Access.

Файл данных Microsoft Access – файл базы данных Microsoft Access или проекта Microsoft Access. В базе данных Microsoft Access 2007 объекты и данные сохраняются в *accdb*-файле, а в более ранних версиях Microsoft Access используется формат *mdb*.

Фильтр – набор условий, применяемых для отбора подмножества данных или для сортировки данных. В Microsoft Access для фильтрации данных можно использовать различные способы, например фильтрацию по выделенному или фильтр по форме.

Фильтр по форме – способ фильтрации данных, в котором используется версия текущей формы или таблицы с пустыми полями. В эти поля вводятся значения, которые должны содержать отфильтрованные записи.

Фильтрация на основе ввода – способ фильтрации записей, при котором отбираются только записи, содержащие указанное значение или удовлетворяющие заданному выражению.

Фильтрация по выделенному – способ фильтрации записей в форме или таблице, при котором отбираются только записи, содержащие выбранное значение.

Фильтрация, исключая выделение – способ фильтрации записей в форме или таблице, при котором отбираются только записи, не содержащие выбранное значение.

Флажок – элемент управления, указывающий на то, что параметр включён. Если параметр включён, на флажке появляется метка.

Фон окна приложения – фоновая область окна приложения.

Форма – объект базы данных Microsoft Access, в который добавляются элементы управления, реагирующие на действия пользователей или служащие для ввода, отображения и изменения данных в полях.

Форма – объект базы данных, который используется для ввода, просмотра или редактирования записей в таблицах Access.

Формат – формат определяет отображение данных на экране или при печати. В базах данных Microsoft Access определён ряд стандартных форматов для некоторых типов данных. Также можно создавать пользовательские форматы.

Функция – запрос, который принимает входные параметры и возвращает результат аналогично хранимой процедуре. Типы: скалярная (несколько инструкций; возвращает одно значение), встроена (одна инструкция; обновля-

емое табличное значение), табличная (несколько инструкций; табличное значение).

Целостность данных – правила, обеспечивающие сохранение установленных отношений между таблицами при добавлении, обновлении или удалении записей. Система правил, используемых для поддержания связей между записями в связанных таблицах.

Числовой тип данных – тип данных поля в базе данных Microsoft Access, предназначенный для хранения чисел, используемых в математических расчётах. Для хранения денежных значений и для выполнения расчётов с ними рекомендуется использовать денежный тип данных.

Экспорт – копирование данных и объектов базы данных в другую базу данных, электронную таблицу или файл другого формата для их использования в других базах данных или программах. Поддерживается экспорт данных в различные базы данных, программы и форматы.

Элемент группировки – элемент управления на форме, отчёте или странице доступа к данным, по значению которого производится группировка значений других элементов управления.

Элемент подчинённой формы или отчёта – элемент управления, который выводит в форме подчинённую форму, а в отчёте – подчинённую форму или подчинённый отчёт.

Элемент управления – средство графического интерфейса пользователя, предназначенное для ввода/вывода информации и управления работой программы. Воздействие пользователя на элемент управления вызывает событие в системе.

Элемент управления перехода по записям – элемент управления, используемый на странице доступа к данным для отображения панели инструментов кнопок перехода. На странице с группировкой элемент управления перехода по записям можно добавить на каждый уровень группировки. Настройка этого элемента управления осуществляется только путём изменения свойств.

Элемент управления с гиперссылкой – элемент управления, позволяющий пользователю выполнить переход к документу, веб-странице или объекту (например, в поле, связанное с полем таблицы, содержащем гиперссылку).

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Бекаревич Ю. Б. Самоучитель Microsoft Access 2013 / Ю. Б. Бекаревич, Н. В. Пушкина. – СПб.: БХВ-Петербург, 2014. – 464 с.
2. Днепров А. Г. Microsoft Access 2007 / А. Г. Днепров. – СПб. : Питер, 2008. – 400 с.
3. Компьютерные технологии и информатика. MS Access. : методические указания по изучению курса для студентов фак. «Бизнес–управление» / ХГИ «НУА» ; [сост.: Белоус Е. В. , Кравец О. А.] – Харьков : Изд-во НУА, 2000. – 72 с.
4. Кирвас В. А. Компьютерные технологии и информатика. Модуль: «Система управления базами данных MS Access» : практикум для студентов фак. «Референт–переводчик», обучающихся по направлению подгот. 6.020303 – Филология (кредит.–модул. система) / В. А. Кирвас; Нар. укр. акад., [каф. информ. технологий и математики]. – Харьков : Изд–во НУА, 2010. – 132 с.
5. Microsoft. Access 2013 видео и учебники. [Электронный ресурс] // Microsoft. – Режим доступа: <https://support.office.com/ru-ru/article/Access-2013-%D0%B2%D0%B8%D0%B4%D0%B5%D0%BE-%D0%B8-%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA%D0%B8-%D0%BF%D0%BE-a4bd10ea-d5f4-40c5-8b37-d254561f8bce?ui=ru-RU&rs=ru-RU&ad=RU> (дата обращения: 15.05.2017). – Загл. с экрана.
6. Основные задачи и сложные запросы в Access [Электронный ресурс] // Microsoft. – Режим доступа: <http://office.microsoft.com/ru-ru/access-help/HA101829991.aspx> (дата обращения: 15.05.2017). – Загл. с экрана.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. БАЗЫ ДАННЫХ И СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ	5
Основные понятия и термины баз данных	5
Отличительные особенности электронных БД:	5
Пользователи БД	5
Классификация БД	6
Классификация СУБД	7
Типы моделей данных	8
Жизненный цикл БД	13
2. ОСОБЕННОСТИ ПОСТРОЕНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ В	
MSROSOFT ACCESS	16
Общие понятия и требования СУБД MS Access	16
Требования к операционной системе для Access 2013	16
Объекты СУБД MS Access	17
Разработка структуры базы данных	19
Типы данных MS Access 2013	21
Свойства полей MS Access 2013	24
Создание связей между таблицами	26
Проектирование структуры реляционной базы данных «Автомобили»	28
3. ОСНОВЫ РАБОТЫ В ПРОГРАММЕ MS ACCESS 2013	33
Запуск Access	33
Интерфейс пользователя Access	37
Работа с таблицами базы данных	40
Создание таблицы в режиме конструктора	40
Определение первичного ключа	43
Создание поля с фиксированным списком	46
Создание подстановки (поля со списком) на основе таблицы или запроса ..	50
Создание маски ввода	54
Создание таблицы в режиме таблицы	55
Создание таблицы путём импорта	61
Оформление внешнего вида таблиц	68
Создание схемы данных	71
Создание связей между таблицами БД	73
Обеспечение целостности данных	77
Работа с формами в базе данных	79
Разработка однотоабличной формы	79

Создание разделённой формы	83
Редактирование формы в режиме макета	85
Редактирование формы в режиме конструктора	86
Создание новой формы в режиме конструктора	91
Проектирование составной формы с использованием мастера	93
Оформление составной формы.....	94
Встраивание в форму кнопок.....	97
Изменение свойств формы.....	98
Создание запросов	98
Запросы на выборку	99
Построение простых запросов с помощью конструктора по одной таблице	100
Использование логических функций при построении запросов	103
Особенности построения запросов по нескольким таблицам.....	103
Добавление вычисляемых полей в запросы.....	104
Импорт таблиц.....	106
Использование шаблонов при организации запросов.....	107
Анализ данных с помощью запросов (итоговые запросы).....	109
Условия отбора в запросах с групповыми операциями.....	111
Отображение строки итогов по столбцу.....	112
Параметрические запросы.....	113
Отчёты в базе данных Access.....	116
Общие сведения об отчётах	116
Создание отчётов	119
Доработка отчёта.....	122
Группировка и сортировка данных в отчётах	126
Редактирование источника записей отчёта	130
Применение фильтрации для отображения отдельных данных	132
Разработка отчёта с помощью мастера.....	133
Доработка отчёта в режиме конструктора	137
Создание почтовых наклеек с помощью мастера.....	137
Приложение 1. СТРУКТУРА ТАБЛИЦ БАЗЫ ДАННЫХ "АВТОМОБИЛИ"	144
Приложение 2. ГОРЯЧИЕ КЛАВИШИ MS ACCESS	151
Приложение 3. СЛОВАРЬ ОСНОВНЫХ ТЕРМИНОВ.....	171
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	193

Навчальне видання

Кірвас Віктор Андрійович

**ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ.
СИСТЕМА УПРАВЛЕННЯ БАЗАМИ ДАНИХ
MS ACCESS**

Навчальний посібник

для студентів II курсу факультету «Референт-перекладач»,
які навчаються за спеціальністю – Філологія
(російською мовою)

В авторській редакції
Комп'ютерний набір *В. А. Кірвас*

Підписано до друку 20.05.2017. Формат 60×84/16.
Папір офсетний. Гарнітура «Таймс».
Ум. друк. арк. 12,25. Обл.-вид. арк. 11,39.
Тираж 300 пр. Зам. № /

План 2016/17 уч. р. поз. № 4 у перепілку робіт кафедри

Видавництво
Народної української академії
Свідоцтво № 1153 від 16.12.2002.

Надруковано у видавництві
Народної української академії

Україна, 61000, Харків, МСП, вул. Лермонтовська, 27.