



МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Инженерная экономика»

О. А. Лавренова

СЕТЕВЫЕ ТЕХНОЛОГИИ И БАЗЫ ДАННЫХ

Пособие



Минск
БНТУ
2024

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Инженерная экономика»

О. А. Лавренова

СЕТЕВЫЕ ТЕХНОЛОГИИ
И БАЗЫ ДАННЫХ

Пособие
для обучающихся по специальности
6-05-0718-01 «Инженерная экономика»
профилиций «Бизнес-процессы промышленных предприятий»,
«Цифровое производство»

В 2 частях

Часть 1

*Рекомендовано учебно-методическим объединением по образованию
в области экономики и организации производства*

Минск
БНТУ
2024

УДК 004.65(075.8)
ББК 32.973-018.2я7
Л13

Рецензенты:
Т. Ф. Старовойтова, К. С. Курочка

Лавренова, О. А.

- Л13 Сетевые технологии и базы данных : пособие для обучающихся по специальности 6-05-0718-01 «Инженерная экономика» профилизиаций «Бизнес-процессы промышленных предприятий», «Цифровое производство» : в 2 ч. / О. А. Лавренова. – Минск : БНТУ, 2024. – Ч. 1. – 326 с.
ISBN 978-985-583-983-6 (Ч. 1).

В первой части издания изложены основные сведения в области теории баз данных, рассмотрены приемы работы с реляционной СУБД из интегрированного офисного пакета Microsoft Office, представлены задания к лабораторным работам, приведены указания к их выполнению и оформлению отчетов.

Издание предназначено для обучающихся дневной и заочной форм обучения специальности 6-05-0718-01 «Инженерная экономика», а также может быть полезным при изучении теории баз данных и освоении практических приемов работы с ними для других специальностей экономического профиля.

УДК 004.65(075.8)
ББК 32.973-018.2я7

ISBN 978-985-583-983-6 (Ч. 1)
ISBN 978-985-583-984-3

© Лавренова О. А., 2024
© Белорусский национальный
технический университет, 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
Часть 1. ОСНОВЫ РАБОТЫ С РЕЛЯЦИОННЫМИ БАЗАМИ ДАННЫХ И СУБД.....	7
1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ	7
Тема 1. ОСНОВЫ ТЕОРИИ БАЗ ДАННЫХ.....	7
1.1. Обзор рынка СУБД.....	7
1.2. Основные понятия теории баз данных и СУБД.....	10
1.3. Основные структурные единицы реляционных БД. Требования к таблицам реляционных БД	14
1.4. Этапы проектирования реляционных БД.....	17
Вопросы для самопроверки	24
Тема 2. ОСНОВНЫЕ СВЕДЕНИЯ О РЕЛЯЦИОННОЙ СУБД MS ACCESS.....	26
2.1. Основные объекты СУБД MS Access	26
2.2. Способы создания БД.....	28
2.3. Разработка макетов таблиц.....	29
2.4. Типы данных MS Access	39
2.5. Свойства полей	43
2.6. Порядок формирования схемы БД.....	50
Вопросы для самопроверки	53
Тема 3. АНАЛИЗ ДАННЫХ В РЕЛЯЦИОННЫХ БД	54
3.1. Инструменты анализа данных. Назначение и виды запросов	54
3.2. Способы создания запросов на выборку	57
3.3. Формирование условий отбора в запросах	62
3.4. Вычисляемые поля в запросах	69
3.5. Применение функций MS Access.....	75
3.6. Использование групповых операций в итоговых запросах.....	81
3.7. Создание перекрестных запросов	87
3.8. Применение параметров в запросах	95
3.9. Назначение и особенности запросов действия. Методика их формирования	98
Вопросы для самопроверки	105
Тема 4. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС БД.....	107
4.1. Назначение и виды форм, способы их создания.....	107
4.2. Работа с формами в окне Конструктора.....	114

4.3. Работа с элементами управления	125
4.4. Особенности создания и настройки кнопочных форм	130
4.5. Работа с макросами	135
4.6. Понятие и назначение отчетов, особенности создания и настройки	140
Вопросы для самопроверки	148
Тема 5. СПЕЦИАЛЬНЫЕ ПРИЕМЫ РАБОТЫ	
С СУБД MS ACCESS	149
5.1. Взаимодействие MS Access с приложениями MS Office. Технологии интеграции реляционных БД и офисных приложений	149
5.2. Использование процедуры Слияние в Word для подготовки типовой рассылки	151
5.3. Настройка рабочей среды MS Access	158
Вопросы для самопроверки	161
Тема 6. ОСНОВЫ ИСПОЛЬЗОВАНИЯ SQL	
6.1. Понятие SQL, варианты реализации, достоинства и особенности	162
6.2. Категории операторов SQL	166
6.3. Основы синтаксиса SQL	172
6.4. Типы данных SQL	173
Вопросы для самопроверки	174
2. ЛАБОРАТОРНЫЕ РАБОТЫ	
Лабораторная работа № 1. ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ С РЕЛЯЦИОННОЙ СУБД MS ACCESS	
	175
Лабораторная работа № 2. ОСВОЕНИЕ ЭТАПОВ СОЗДАНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ	
	180
Лабораторная работа № 3. РАЗРАБОТКА ТАБЛИЦ ИНДИВИДУАЛЬНОЙ БАЗЫ ДАННЫХ	
	199
Лабораторная работа № 4. АНАЛИЗ ДАННЫХ ПРИ ПОМОЩИ ЗАПРОСОВ ВЫБОРКИ	
	211
Лабораторная работа № 5. ИСПОЛЬЗОВАНИЕ ВЫЧИСЛЯЕМЫХ ПОЛЕЙ И ПАРАМЕТРОВ В ЗАПРОСАХ MS ACCESS	
	222
Лабораторная работа № 6. ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ MS ACCESS В ЗАПРОСАХ ВЫБОРКИ	
	238
Лабораторная работа № 7. ПРОЕКТИРОВАНИЕ И ИСПОЛЬЗОВАНИЕ ТАБЛИЧНЫХ ФОРМ	
	248

Лабораторная работа № 8. ПОДГОТОВКА ОТЧЕТОВ СРЕДСТВАМИ СУБД	260
Лабораторная работа № 9. РАЗРАБОТКА И НАСТРОЙКА ГЛАВНОЙ КНОПОЧНОЙ ФОРМЫ	264
Лабораторная работа № 10. ВЗАИМОДЕЙСТВИЕ СУБД MS ACCESS С ПРИЛОЖЕНИЯМИ MS OFFICE.....	270
Лабораторная работа № 11. РАЗРАБОТКА ПРЕЗЕНТАЦИИ БАЗЫ ДАННЫХ СРЕДСТВАМИ MS POWERPOINT	281
3. КОНТРОЛЬНАЯ РАБОТА	286
3.1. Общие сведения о структуре работы и выборе варианта заданий	286
3.2. Список теоретических вопросов	289
3.3. Примерный перечень предметных областей.....	291
3.4. Указания к выполнению и оформлению практической части	292
3.5. Список типовых заданий к аттестации (на примере учебной БД)	307
3.6. Краткие требования к оформлению контрольной работы.....	312
СПИСОК СОКРАЩЕНИЙ	313
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	314
ПРИЛОЖЕНИЕ А	317

ВВЕДЕНИЕ

Внедрение стека технологий «Индустрия 4.0» с 2011 года является главным мировым трендом в повышении конкурентоспособности и эффективности предприятий. Цифровая трансформация предприятий сопровождается внедрением комплексных автоматизированных информационных систем, которые являются необходимым элементом эффективной системы управления. В свою очередь, основу любой информационной системы составляет база данных (БД), для управления которой применяется соответствующая система управления базами данных (СУБД). Таким образом, широкое распространение информационно-коммуникационных технологий требует от современного специалиста высокой информационной культуры, свободного владения технологиями структурирования и обработки данных средствами современных СУБД.

В теоретической части пособия «Сетевые технологии и базы данных» представлены основные сведения в области теории реляционных баз данных и СУБД, изложены вопросы проектирования баз данных и разработки средствами офисной СУБД объектов базы данных для хранения и анализа данных, организации интерфейса, также уделено внимание основам использования SQL и вопросам интеграции баз данных с офисными приложениями.

Лабораторные работы направлены на практическое освоение этапов проектирования и применения реляционных баз данных для хранения и анализа данных. Выполнение работ позволит обучающимся приобрести навыки визуального программирования БД, создания объектов реляционных БД и управления ими, а также применения механизмов интеграции БД с офисными приложениями.

Комплексные задания контрольной работы нацелены на закрепление полученных знаний и навыков путем самостоятельной работы над конкретной предметной областью. В данном разделе приведены указания к выполнению и оформлению работы, пример ее оформления будет полезен для студентов заочной формы обучения.

Материал пособия является актуализацией ранее изданных по дисциплине «Сетевые технологии и базы данных» конспекта лекций [14] и лабораторных работ [8, 18], разделов электронного учебно-методического комплекса [22] и других изданий, исправленных и дополненных пояснениями и практическими примерами.

Часть 1. ОСНОВЫ РАБОТЫ С РЕЛЯЦИОННЫМИ БАЗАМИ ДАННЫХ И СУБД

1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

Тема 1. ОСНОВЫ ТЕОРИИ БАЗ ДАННЫХ

- 1.1. Обзор рынка СУБД.
- 1.2. Основные понятия теории баз данных и СУБД.
- 1.3. Основные структурные единицы реляционных БД. Требования к таблицам реляционных БД.
- 1.4. Этапы проектирования реляционных БД.

1.1. Обзор рынка СУБД

Основные идеи современных информационных технологий базируются на концепции, согласно которой данные должны быть организованы в *базы данных* (БД) с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей. БД составляют основу информационного обеспечения современных систем и являются эффективным средством представления структурированных данных и манипулирования ими.

Базы данных создаются и функционируют под управлением специальных программных комплексов, называемых *системами управления базами данных* (СУБД), например: Oracle, MS SQL Server, MySQL, Informix, DB2, MS FoxPro, MS Access, OpenOffice Base, Azure Tables и Azure DocumentDB (NoSQL от Microsoft) и др.

В настоящее время рынок СУБД развивается по двум направлениям: 1) поддержка оперативной деятельности компаний и 2) аналитическая обработка накопленных массивов данных. Причем, характер решаемых задач по каждому направлению выдвигает достаточно противоречивые требования: так, для первого направления одним из ключевых требований является возможность оперативного изменения содержимого БД, а для второго – скорость поиска и выборки требуемой информации.

Для СУБД первого направления, ориентированных на решение задач оперативной деятельности компаний, аналитики Gartner при-

меняют термин Operational Database Management Systems (OPDBMS, «операционные СУБД») – программное обеспечение для управления реляционными и нереляционными базами данных для традиционных транзакций, поддерживающих бизнес-процессы. Основное назначение систем OPDBMS (операционных СУБД) – поддержка широкого спектра корпоративных транзакционных приложений уровня предприятия, таких как ERP, CRM, систем Интернета вещей и др.

Смену терминологии (OPDBMS вместо применявшегося ранее термина OLTP – Online Transaction Processing, онлайн-транзакционная обработка) аналитики объясняют тем, что при ведении современного бизнеса для информационной поддержки систем бизнес-процессов необходима обработка большого объема как *структурированной* информации (*например*, учетных данных), так и использование *неструктурированной* информации, причем часто в режиме реального времени.

Аналитики Gartner выделяют в ИТ-системах два новых класса данных, используемых как в аналитических задачах, выполняемых в фоновом режиме, так и в оперативной деятельности компаний [7]:

1. *Interaction data* (данные взаимодействия) – это данные, генерируемые людьми при их взаимодействии, в первую очередь, с использованием социальных методов общения (блоги, посты и т. д.), а также в архивах электронной почты, аудио- и видеозаписях и т. д.

2. *Observation data* (данные наблюдений) – это данные, которые генерируются различными «соединенными» устройствами, как правило, в автоматическом режиме. *Например*, координаты геолокации, получаемые с мобильных устройств, сведения о свойствах устройств Интернета вещей и т. д.

Факторами трансформации рынка СУБД выступают как запросы потребителей на обработку данных, так и современные технологии, такие как Open Source, Cloud Computing, non-SQL, In-memory. Меняются и физические устройства хранения данных: на смену жестким дискам приходят твердотельные накопители, флэш- и оперативная память. Также изменяются и модели реализации (доставки пользователям) и использования СУБД: на смену традиционной автономной установке ПО (stand-alone) приходят, *например*, облачные (публичные, частные, гибридные), а также программно-аппаратные комплексы. Следует отметить, что если в начале своего развития операци-

онные СУБД имели чаще всего реляционную архитектуру, то в настоящее время наблюдается рост и нереляционных систем.

В отчете «Магический квадрант для облачных систем управления базами данных» («Magic Quadrant for Cloud Database Management Systems») ежегодно аналитики компании Gartner определяют места для 20 поставщиков реляционных и нереляционных СУБД, хранилищ документов и хранилищ данных. Ведущими игроками на рынке операционных СУБД по данным компании Gartner с 2022 года выступают такие компании, как Amazon, Microsoft, Google, Oracle и др. [4].

Аналитики Gartner для рынка облачных СУБД отмечают, что предлагаемые решения могут обслуживать несколько моделей (*сведения о моделях данных приведены в п. 1.2*) и типов данных – реляционные, нереляционные (документоориентированные, хранилища триплетов (RDF), хранилища «ключ-значение», графовые, векторные, геопространственные и другие. В рейтинге DB-Engines (<https://db-engines.com/en/ranking>) среди наиболее популярных СУБД лидируют реляционные и мультимодельные СУБД (рис. 1.1).

Rank			DBMS	Database Model
Mar 2024	Feb 2024	Mar 2023		
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ
6.	6.	6.	Redis +	Key-value, Multi-model ⓘ
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model ⓘ
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model ⓘ
9.	9.	↑ 11.	Snowflake +	Relational
10.	10.	↓ 9.	SQLite +	Relational
11.	11.	↓ 10.	Microsoft Access	Relational
12.	12.	12.	Cassandra +	Wide column, Multi-model ⓘ
13.	13.	13.	MariaDB +	Relational, Multi-model ⓘ
14.	14.	14.	Splunk	Search engine
15.	↑ 16.	↑ 16.	Microsoft Azure SQL Database	Relational, Multi-model ⓘ

Relational DBMS

Document store

Graph DBMS

RDF store

Spatial DBMS

Vector DBMS

Рис. 1.1. Рейтинг СУБД по версии DB-Engines (<https://db-engines.com/en/ranking>)

Поставщики предлагают пользователям аналитических БД набор передовых технологий, таких как: встроенное машинное обучение, разработка low-code или мультимодальность, которые уже становятся стандартными [1].

Также для современных решений на рынке СУБД уже стандартны возможности эластичной масштабируемости приложений, поддержка SQL (в том числе для нереляционных СУБД), возможность сочетать оперативную и аналитическую обработку данных.

1.2. Основные понятия теории баз данных и СУБД

База данных – структурированная совокупность логически взаимосвязанных данных конкретной предметной области, организованная на магнитных носителях средствами СУБД [21].

Под *предметной областью* (ПО) понимается совокупность объектов реального мира и их связей, относящихся к некоторой области человеческой деятельности и имеющих практическую ценность для пользователей при их отражении в БД. *Например*, производственная деятельность, торговля, учебная деятельность и др.

СУБД (система управления базами данных) – программное средство, предназначенное для создания и обслуживания БД на внешних запоминающих устройствах (ВЗУ).

Любая СУБД поддерживает минимальный набор *основных* функций: *ввод, обновление и анализ* данных.

К *дополнительным* функциям СУБД относятся:

- 1) проверка состояния БД;
- 2) выдача справочной информации;
- 3) разграничение прав доступа пользователей к объектам БД, инструментам анализа информации и др.

Для выполнения этих функций в состав СУБД входят *язык описания данных*, позволяющий создать структуру описания данных в базе, и *язык манипулирования данными*, позволяющий производить различные операции с данными.

Проектирование и разработка базы данных выполняет команда *аналитиков и системных программистов*, которые взаимодействуют с заказчиком проекта по разработке БД. При разработке базы данных учитывается специфика предметной области и запросы *конечных пользователей*.

Работу действующей БД обслуживает команда *администраторов*, которые выполняют подключение к системе новых пользователей, определяют нормы и правила доступа к данным, создают копии данных, проверяют и восстанавливают информацию.

Классификацию СУБД можно выполнить по различным признакам [15, 19]:

1. *По месту размещения БД:*

а) *локальные* – содержимое БД располагается на одном компьютере (*например*, сервере компании);

б) *распределённые* – содержимое БД частично находится на разных компьютерах (*например*, в облачных хранилищах).

2. *По принципу организации хранения и обработки данных:*

а) *клиент-серверные* – СУБД и БД хранятся на одном компьютере, к которому пользователи отправляют запросы на обработку, причем доступ к клиент-серверной СУБД можно получить с любого компьютера. *Например*, так работают СУБД и БД интернет-магазинов;

б) *файл-серверные* – БД находится на сервере, а СУБД – на компьютерах или любых устройствах пользователя, с которых к БД отправляются запросы.

Такие СУБД используются для организации работы локальных сервисов компании, *например*, CRM-систем (обрабатывают данные о клиентах компании и их взаимодействии с компанией);

в) *встраиваемые* – это локальные СУБД, представляющие собой отдельный модуль на языке программирования для управления данными внутри приложений.

Например, Microsoft Access, 1С:Бухгалтерия.

3. *По поддержке языка запросов:*

а) *SQL* – язык структурированных запросов, поддерживается большинством популярных СУБД.

Например, PostgreSQL, Microsoft SQL Server, MySQL;

б) *NoSQL* – язык неструктурированных запросов, основанный на другом языке программирования, *например*, Python или JavaScript, обычно используется при работе с большими данными.

Пример такой NoSQL СУБД – MongoDB, которую используют такие крупные компании, как Google, Forbes, IBM и др., а также многие интернет-магазины.

4. *По структуре и организации данных (модели данных).*

Поскольку этот признак классификации является важным для понимания особенностей процесса обработки данных, необходимо дать определение понятию «модель данных».

Модель данных представляет собой множество структур данных, ограничений целостности и операций манипулирования данными [21]. Модель данных является ядром любой БД. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

К известным моделям данных относятся: иерархические, сетевые, реляционные, постреляционные БД, резидентные БД («in-memo») и др. Краткая характеристика некоторых моделей приведена ниже.

1. *Иерархические* – данные представлены в виде деревьев. Вершины – информационные единицы, дуги – связи. Каждый объект может подчиняться только одному объекту более высокого уровня. Существует единая точка входа (рис. 1.2).

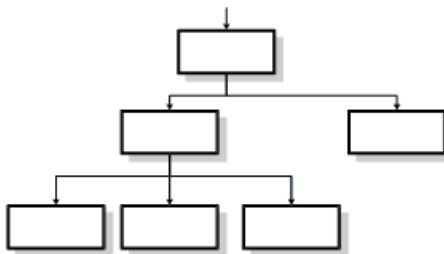


Рис. 1.2. Иерархическая модель данных

2. *Сетевые* – данные связаны системой отношений в виде произвольной сети. Любой объект может быть связан с любым количеством других элементов. Существует несколько точек доступа (рис. 1.3).

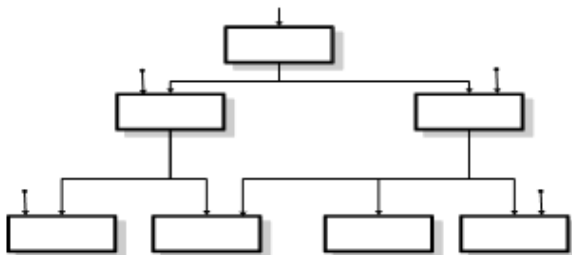


Рис. 1.3. Сетевая модель данных

Существенными недостатками как иерархической, так и сетевой моделей являются дублирование данных и неоптимальный доступ к ним.

3. *Реляционные* – данные сгруппированы в двумерные таблицы-отношения, между которыми установлены связи по ключам (рис. 1.4).

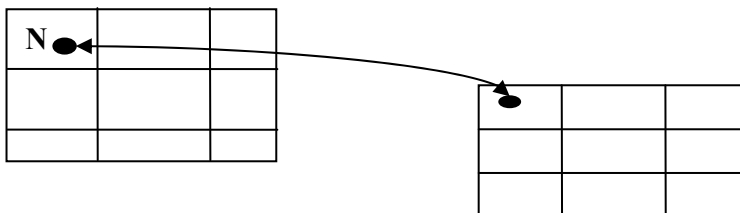


Рис. 1.4. Реляционная модель данных

Впервые термин «реляционная модель данных» появился в статье сотрудника фирмы IBM доктора Кодда (Codd E. F., A Relational Model of Data for Large Shared Data Banks, 1970). Кодд (математик по образованию) предложил использовать для обработки данных аппарат теории множеств (объединение, пересечение, разность, декартово произведение). Он показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известного в математике как *отношение* (*relation*, англ.).

При работе с реляционными моделями используется как математическая терминология, так и терминология, исторически принятая в сфере обработки данных. Основные формальные реляционные термины и соответствующие им неформальные эквиваленты приведены на рис. 1.5.

Формальный реляционный термин	Неформальный эквивалент
Отношение	Таблица
Кортеж	Запись, строка
Атрибут	Поле, столбец

Рис. 1.5. Основные формальные реляционные термины и их эквиваленты

Реляционная модель описывает данные в виде взаимосвязанных таблиц. Однако это требование относится только к логической структуре базы данных, определяемой на уровне внешнего, концептуального проектирования. На физическом уровне в БД могут использоваться различные структуры для хранения данных и механизмы доступа к ним.

Широкому распространению достаточно простых для понимания реляционных (табличных) СУБД способствовало увеличение объемов хранимых данных, усложнение их структуры, а также значительный рост числа пользователей информационных систем.

Несмотря на развитие различных моделей данных и в 2023 году доля СУБД с реляционной моделью в рейтинге популярности DB-Engines (https://db-engines.com/en/ranking_categories) была преобладающей и превышала 72 %.

Независимо от применяемой СУБД при работе с любой БД происходит преобразование данных из внешнего представления во внутреннее в соответствии с ее логической структурой, основанной на применяемой модели данных.

Описание общей логической структуры организации БД называется *схемой* БД. В схему БД входит полное описание всех типов данных, хранящихся в базе, а также всех типов операций над ними.

1.3. Основные структурные единицы реляционных БД. Требования к таблицам реляционных БД

Основными структурными единицами реляционных БД являются следующие:

1. *Таблица* – множество одинаковых по структуре записей со значениями в полях.

2. *Запись* – совокупность полей, соответствующая логически связанным реквизитам. *Структура записи* определяется составом и последовательностью полей, входящих в нее. Каждая запись однозначно идентифицируется *ключом* (ключевым полем).

3. *Поле* – элементарная единица логической организации данных, которая соответствует неделимой единице информации – реквизиту. *Реквизит* – простейшая единица информации, отображающая качественную или количественную характеристику объекта предметной области.

Поля могут быть (рис. 1.6) *ключевыми* и *описательными* (информационными).



Рис. 1.6. Виды полей

Ключевое поле (ключ) – одно или несколько полей, значения которых однозначно определяют запись в таблице.

Ключевые поля бывают *первичные* и *внешние*:

1. *Первичный ключ* – одно или несколько полей главной таблицы однозначно идентифицирующих запись (одно поле – простой ключ, иначе – составной). Значения первичного ключа *уникальны*.

2. *Внешний ключ* – одно или несколько полей в таблице-связке, содержащих ссылку на ключевое поле или поля в главной таблице. Значения внешнего ключа *могут повторяться* в нескольких записях таблицы-связки.

При описании логической структуры данных указывают *структуру записи*, т. е. поля и порядок их следования, также для каждого поля задаются *имя, тип данных* (определяет содержание поля, формат и точность данных) и для ключевых полей – *признак ключа*.

Реляционная БД представляет собой набор двумерных таблиц-отношений, связи между которыми установлены при помощи ключей (ключевых полей). Другими словами, в реляционной БД все доступные пользователю данные организованы в виде набора двумерных таблиц, а все операции над данными сводятся к операциям над этими таблицами.

Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в простейшей базе данных являются *поля* и *записи*. Если записей в таблице пока нет, значит, ее структура образована только набором полей.

Таблицы-отношения реляционной модели обладают следующими свойствами:

- каждый столбец таблицы соответствует элементу (атрибуту) данных. Повторяющиеся атрибуты отсутствуют;
- в каждой строке таблицы содержится по одному значению в соответствующем столбце;
- все столбцы (поля) таблицы – однородные;
- столбцам присвоены однозначные имена, определяющие атрибуты;
- один или несколько атрибутов являются ключом таблицы, который однозначно идентифицирует запись таблицы;
- в таблице не может быть двух одинаковых строк (записей);
- общее количество строк (записей) не ограничено.

Важным требованием, предъявляемым к таблицам реляционной модели, является *нормализация данных* – минимизация количества повторяющихся данных.

Существует несколько нормальных форм реляционной модели, которые вводят ограничения и позволяют минимизировать дублирование данных и обеспечить целостность данных [21].

Первично нормализованная таблица содержит простые атрибуты (одному атрибуту соответствует одно значение).

При второй нормальной форме атрибуты являются простыми и каждый описательный (неключевой) атрибут функционально полно зависит от ключа (одному значению ключа соответствует одно значение неключевого атрибута).

При третьей нормальной форме реляционной модели (канонической модели) выполняются следующие *требования нормализации*:

1) каждая таблица должна содержать уникальный ключ (простой или составной);

2) все описательные (неключевые) атрибуты должны быть взаимно независимы;

3) все атрибуты составного ключа должны быть взаимно независимы;

4) каждый описательный атрибут должен функционально полно зависеть от ключа (каждому ключу соответствует одно значение описательного атрибута);

5) при составном ключе каждый описательный атрибут должен функционально полно зависеть от всей совокупности атрибутов ключа (не допускается зависимость описательного атрибута от части ключа);

6) каждый описательный атрибут не может зависеть от ключа транзитивно, т. е. через другой промежуточный атрибут. В случае транзитивной зависимости совокупность атрибутов разделяется с образованием двух таблиц.

Для реализации БД средствами реляционной СУБД достаточно обеспечить соблюдение требований нормализации третьей нормальной формы.

К реляционным СУБД относятся: PostgreSQL, Microsoft SQL Server, MySQL, Oracle и др.

Как отмечалось выше, реляционная модель достаточно проста для понимания пользователя, имеет развитый математический аппарат, обеспечивающий строгую структуру и построчную обработку данных.

Вместе с тем реляционная модель имеет несколько *ограничений* по применению, связанных, в первую очередь, с жесткой структурой данных, также с возможностью выбора небольшого числа записей из БД за один запрос и ограничением по объему данных, который не должен превышать нескольких терабайт, в противном случае существенно снижается скорость обработки данных.

1.4. Этапы проектирования реляционных БД

Проектирование БД является очень важным этапом, от которого зависят последующие этапы ее реализации средствами СУБД. При проектировании БД разработчики решают основные задачи:

1. Обеспечение хранения в БД всей необходимой информации.
2. Обеспечение возможности анализа и извлечения данных по запросам конечного пользователя.
3. Сокращение избыточности и дублирования данных.
4. Обеспечение целостности данных.

Перед разработкой БД необходимо располагать *первичными документами* выбранной предметной области, чтобы выполнить ее описание, составить перечень всей необходимой информации для удовлетворения предполагаемых запросов пользователя и определить потребности в обработке данных.

Укрупненно процесс разработки реляционной базы данных состоит из четырех этапов (рис. 1.7).

1 этап. Определение цели создания базы данных.

2 этап. Выделение информационных объектов предметной области.

3 этап. Определение логической структуры БД.

4 этап. Создание и заполнение объектов базы данных средствами конкретной СУБД.

На первом этапе проектирования БД необходимо определить:

- цель создания БД;
- основные функции БД;
- информацию, которую должна содержать БД.

Фактически определяют основные темы таблиц базы данных и информацию, которую будут содержать таблицы.

База данных должна отвечать требованиям конечных пользователей. Поэтому необходимо четко определить темы, которые должна охватывать БД, и отчеты, которые она должна выдавать, проанализировать формы, которые в настоящий момент используются для записи данных. На этом этапе целесообразно изучить существующие аналоги баз данных и выполнить сравнительный анализ их возможностей.

Информационный объект (ИО) – это описание некоторой сущности (реального объекта, процесса, явления или события) в виде совокупности логически связанных информационных элементов (атрибутов, реквизитов). Информационный объект образуется совокупностью логически взаимосвязанных реквизитов, представляющих его качественные и количественные характеристики. *Например,*

информационные объекты предметной области «Торговые операции»: *Клиент, Товар, Поставщик, Сотрудник.*



Рис. 1.7. Этапы разработки реляционной БД

На втором этапе выделяют информационные объекты предметной области.

Выделение информационных объектов осуществляют в следующем порядке [21]:

а) проводят анализ первичных документов предметной области, при котором определяется состав и структура данных;

б) выполняют описание предметной области в виде списка реквизитов, извлекаемых из первичных документов – источников загрузки БД;

в) формируют информационные объекты путем функционального анализа и группировки реквизитов, извлеченных из первичных документов, для чего необходимо:

– установить в списке реквизитов функциональные зависимости между реквизитами;

– разделить все реквизиты на ключевые и описательные, установить между ними соответствие;

– образовать информационный объект, сгруппировав описательные реквизиты, одинаково зависимые от ключевого;

– определить структурные связи.

На рис. 1.8 представлен пример выделения информационных объектов для предметной области «Хозяйственная деятельность» [21].

В результате анализа реквизитов, извлеченных из первичного документа «Договор», для предметной области «Хозяйственная деятельность» выделены три информационных объекта с набором реквизитов (ключевых и описательных) [21]:

1. *Договор – номер договора (ключ)*, дата, сумма, количество, (код заказчика, код изделия – ключи для связи объектов).

2. *Заказчик – код заказчика (ключ)*, наименование заказчика, адрес, банк.

3. *Изделие – код изделия (ключ)*, наименование изделия, единица измерения, минимальная партия, цена.

В соответствии с требованиями реляционной модели каждому информационному объекту будет соответствовать таблица БД.

Проектирование структуры таблиц БД является одним из наиболее сложных и важных этапов, так как результаты, которые должна выдавать база данных (отчеты, выходные формы и др.) не всегда дают полное представление о структуре ее таблиц.

Документ	Реквизиты	Функциональные зависимости
Договор	Номер	←
	Дата	
	Сумма	
	Количество	
	Код заказчика	←
	Наименование заказчика	
	Адрес заказчика	
	Банк	
	Минимальная партия	←
	Цена	
	Наименование изделия	
	Единица измерения	
	Код изделия	←

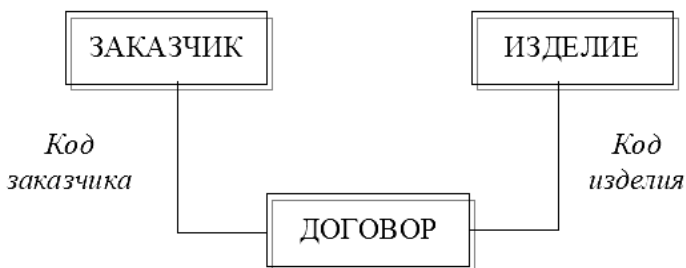


Рис. 1.8. Выделение информационных объектов для предметной области «Хозяйственная деятельность»

Проектирование структуры БД как совокупности информационных объектов (таблиц) осуществляется до использования СУБД. Графическое отображение информационно-логической модели предметной области по сути представляет проект реляционной БД. Ее структура представляется в виде взаимосвязанных таблиц с соблюдением требований к их содержанию:

1. Информация в таблице не должна дублироваться. Не должно быть повторений и между таблицами.

Когда определенная информация хранится только в одной таблице, то и изменять ее придется только в одном месте. Это делает работу более эффективной, а также исключает возможность несоответствия информации в разных таблицах. *Например*, в одной таблице должны содержаться адреса и телефоны клиентов.

2. *Каждая таблица должна содержать информацию только на одну тему.*

Сведения на каждую тему обрабатываются намного легче, если они содержатся в независимых друг от друга таблицах. *Например*, адреса и заказы клиентов хранятся в разных таблицах, чтобы при удалении заказа информация о клиенте осталась в базе данных.

3. *Каждая таблица должна содержать первичный ключ.*

Для того чтобы связать данные из разных таблиц, *например*, данные о клиенте и его заказы, каждая таблица должна содержать поле или набор полей, которые будут задавать индивидуальное значение каждой записи в таблице. Такое поле или набор полей называют *первичным ключом*.

4. *В таблице должна присутствовать вся необходимая информация.*

Информацию в таблицы БД рекомендуется *закладывать с избытком*. Каждая таблица содержит информацию на отдельную тему, а каждое поле в таблице содержит отдельные сведения по теме таблицы. *Например*, в таблице с данными о корпоративном клиенте целесообразно включить поля с названием компании, ее адресом, должностью контактного лица и номером телефона.

5. *Каждое поле должно быть связано с темой таблицы.*

6. *Не рекомендуется включать в таблицу поля для данных, которые являются результатом вычислений.* *Например*, в таблице с данными о заказах клиентов не включается поле, в котором придется ввести стоимость заказа, как число, рассчитанное вручную.

7. *Информацию следует разбивать на наименьшие логические единицы (например, поля «Фамилия», «Имя» и «Отчество», а не общее поле «ФИО»;* вместо одного поля «Адрес» рекомендуется предусмотреть набор полей «Страна», «Индекс», «Город», «Адрес», причем последнее поле будет содержать название улицы, дом, квартиру и т. д.).

На третьем этапе определяют логическую структуру БД. Фактически определяют тип связей между объектами (таблицами), учи-

тывая, что реальное отношение определяется отношением между экземплярами двух типов информационных объектов.

Отношения между объектами могут быть следующих типов:

1. *Одно-однозначные* ($1 : 1$) – имеют место, когда каждой записи одной таблицы соответствует одна запись другой. Такие таблицы могут быть легко объединены в одну, и первичным ключом может стать любой ключ из таблицы. Объекты равноправные.

2. *Одно-многозначные* ($1 : \infty$) – каждой записи одной таблицы может соответствовать несколько записей другой. Имеют место иерархические групповые отношения. Один объект определяется как главный, а другой – как подчиненный.

3. *Много-многозначные* ($\infty : \infty$) – каждой записи одной таблицы может соответствовать несколько записей другой и наоборот. Это сетевые групповые отношения. Такие отношения непосредственно не могут поддерживаться в реляционных СУБД. Обычно реализуются через третий объект, с которым исходные объекты связаны одно-многозначными отношениями, который вводится дополнительно и называется *объектом-связкой*.

В реляционных СУБД реализуются отношения «один-ко-многим» и «один-к-одному», а реальные отношения между информационными объектами чаще всего являются отношениями «многие-ко-многим», которые непосредственно не поддерживаются в реляционных СУБД. Поэтому реальные отношения «многие-ко-многим» трансформируются в отношения «один-ко-многим» путем ввода объекта-связки.

На этапе определения связей между объектами применяют следующие *практические приемы нормализации* отношений:

1. Необходимо проверить, что между каждым полем и записью в таблице существует связь $1 : 1$.

2. Если между некоторыми данными в таблице имеется связь $1 : \infty$, следует поместить эти данные в другую таблицу, которая содержит первичный ключ первой таблицы в качестве внешнего ключа.

3. Связь $\infty : \infty$ между двумя таблицами разбивают на две связи $1 : \infty$ и создают третью таблицу, которая содержит первичные ключи обеих таблиц в качестве внешних.

После завершения третьего этапа получают *информационно-логическую модель предметной области* (ИЛМ ПО) – графическое

отображение совокупности информационных объектов и связей между ними. Другими словами, ИЛМ ПО – это проект будущей БД.

После проектирования таблиц, полей и связей необходимо еще раз проанализировать структуру будущей базы данных и выявить возможные недочеты в организации данных. Желательно сделать это на этапе формирования информационно-логической схемы БД.

На четвертом этапе средствами конкретной СУБД осуществляют создание и заполнение объектов базы данных.

Первоначально необходимо сформировать макеты всех главных, а затем подчиненных таблиц (см. п. 2.3), исключив все возможные повторения данных; установить связи между таблицами – создать схему БД (см. п. 2.6).

До эксплуатации БД, чтобы выполнить проверку работоспособности БД, необходимо создать таблицы, определить связи между ними, а затем ввести несколько контрольных записей в каждую таблицу. Если структуры таблиц отвечают требованиям к их содержанию, то выполняется ввод реальных данных и заполнение БД.

Рекомендуется создать контрольные запросы, формы и отчеты, чтобы оценить, насколько база данных соответствует заявленным требованиям и проверить возможности работы с требуемой информацией. Если контрольные результаты удовлетворительны, то можно создавать любые запросы, формы, отчеты, макросы и модули, в противном случае следует откорректировать структуру и содержание таблиц БД, заново установить связи между ними (изменить схему БД).

Вопросы для самопроверки

1. Сформулируйте понятие «база данных». Какое программное обеспечение необходимо для создания и функционирования баз данных?

2. Что такое СУБД? Перечислите основные функции СУБД и приведите примеры современных СУБД.

3. Назовите признаки классификации СУБД.

4. По каким направлениям развивается рынок СУБД? Какие факторы оказывают влияние на его трансформацию?

5. Какие компании являются крупнейшими поставщиками облачных СУБД? Какими возможностями отличаются облачные СУБД?

6. Приведите определение понятия «модель данных». Назовите основные модели данных и кратко опишите их особенности.

7. Кто и когда сформулировал термин «реляционная модель данных»? Какими свойствами обладают таблицы-отношения реляционной модели данных?

8. Что понимают под «нормализацией данных»? Перечислите требования нормализации при третьей нормальной форме реляционной модели.

9. Что представляет собой «реляционная БД»? Назовите основные структурные единицы реляционной БД и сформулируйте их определения.

10. Что такое «поле» таблицы БД? Какие бывают поля? Приведите примеры ключевых и описательных (информационных) полей.

11. Что такое «ключевое поле» таблицы? Какие бывают ключевые поля? Сформулируйте назначение и особенности «первичных» и «внешних» ключей.

12. Перечислите этапы проектирования реляционной БД и кратко characterizeуйте их содержание.

13. Что понимают под «предметной областью»? Приведите примеры.

14. Сформулируйте понятие «информационный объект предметной области»? Опишите порядок выделения информационных объектов при проектировании БД.

15. Дайте определение понятию «информационно-логическая модель предметной области» (ИЛМ ПО)? Как формируется ИЛМ ПО?

16. Какие требования следует соблюдать при проектировании информационных объектов (таблиц) реляционной БД?

17. Какие виды отношений существуют между информационными объектами? Какие из них поддерживаются реляционными СУБД?

Тема 2. ОСНОВНЫЕ СВЕДЕНИЯ О РЕЛЯЦИОННОЙ СУБД MS ACCESS

- 2.1. Основные объекты СУБД MS Access.
- 2.2. Способы создания БД.
- 2.3. Разработка макетов таблиц.
- 2.4. Типы данных MS Access.
- 2.5. Свойства полей.
- 2.6. Порядок формирования схемы БД.

2.1. Основные объекты СУБД MS Access

СУБД MS Access является доступной реляционной СУБД, которая входит в состав интегрированного пакета для офиса Microsoft Office, и может быть использована для работы с небольшими базами данных офисного назначения. Важным достоинством MS Access (особенно для начинающего пользователя) является стандартный графический интерфейс, унифицированные приемы настройки и управления, характерные для всех приложений офисного пакета.

MS Access не только предоставляет пользователю широкий набор средств для обработки и представления данных, но и обеспечивает интеграцию с другими приложениями MS Office с помощью буфера обмена, технологии OLE, операций импорта и экспорта объектов в приложения MS Word, MS Excel и может использоваться для разработки уникальных решений в малых и средних БД.

В окне БД, открытом в СУБД MS Access, в области навигации отображается список доступных в СУБД объектов (*например*, на рис. 2.1 представлен список таблиц учебной БД *Борей*).

В MS Access база данных представляет собой один файл с расширением *.accdb*, в котором содержатся *шесть типов объектов*:

Таблицы – объекты СУБД MS Access, предназначенные для структурированного хранения данных. Именно взаимосвязанные таблицы составляют основу реляционной БД, с которых начинается работа над БД.

Запросы – объекты СУБД MS Access, предназначенные для анализа данных из таблиц.

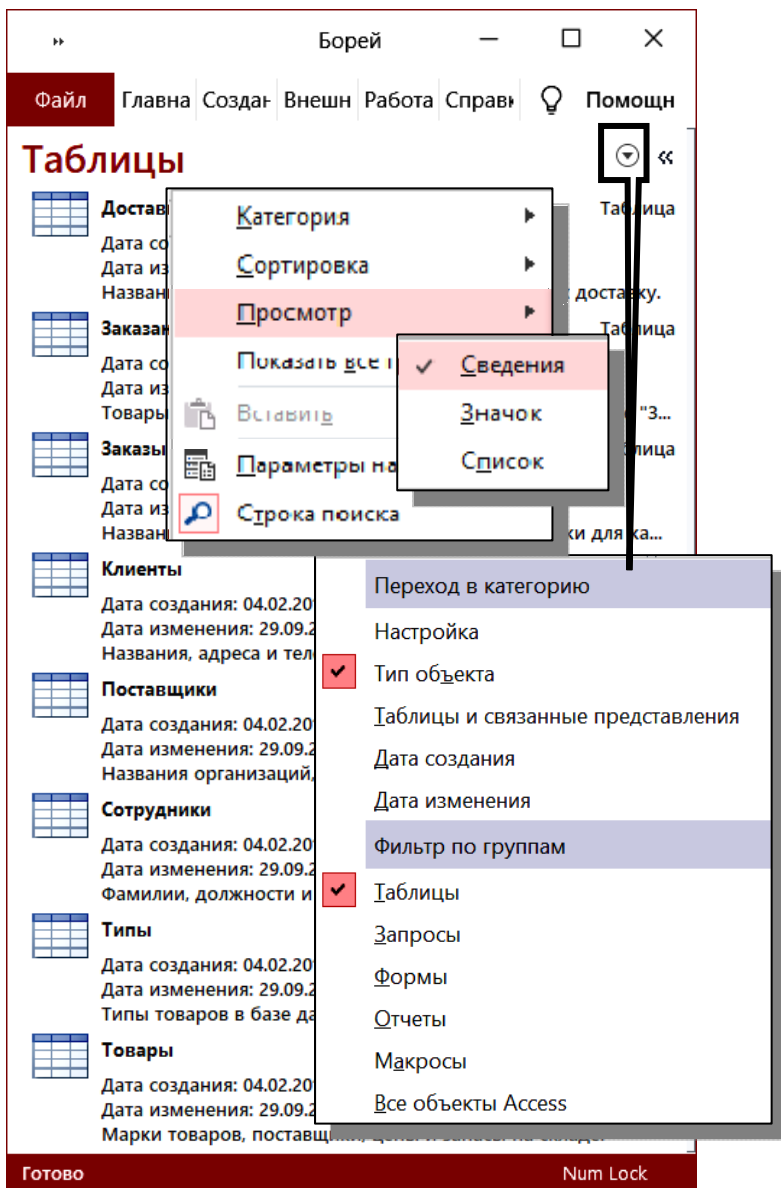


Рис. 2.1. Представление объектов БД на вкладке *Таблицы* (в режиме просмотра сведений)

Формы – объекты СУБД MS Access, предназначенные для организации пользовательского интерфейса: 1) наглядного отображения, ввода и редактирования данных из таблиц (*табличные формы*) и 2) управления объектами базы данных (*кнопочные формы*).

Отчеты – объекты СУБД MS Access, предназначенные для подготовки и вывода на печать итоговых документов в удобном для пользователя виде.

Макросы – объекты СУБД MS Access, представляющие собой набор инструкций (макрокоманд) и предназначенные для автоматизации часто повторяющихся действий пользователя.

Модули – объекты СУБД MS Access, представляющие собой законченные программы на языке VBA и предназначенные для решения прикладных задач пользователя.

С любыми объектами СУБД MS Access можно работать как минимум в двух режимах:

– в режиме *Конструктора*;

– в режиме *Объекта* (режиме *Таблицы, Формы* и т. д.).

Переключение между режимами осуществляется с помощью кнопки «Режим» на вкладке *Главная* в группе команд *Режимы*.

В режиме *Конструктора* создается *макет объекта* (таблицы, запроса, формы и т. д.), а в режиме *Объекта* (*Таблицы, Формы* и т. д.) – отображается соответствующий результат обработки данных (записи таблицы, результат запроса и т. д.).

2.2. Способы создания БД

Для создания новой БД в СУБД MS Access используется команда *Создать* в меню *Файл*. Далее выбирают один из вариантов:

1. *Пустая БД*. При создании пустой БД необходимо указать ее местоположение (*диск, папку*) и *имя файла* (по умолчанию новая БД получает имя *DataBase1.accdb*, которое не рекомендуется использовать, а следует заменить на уникальное). Далее следует самостоятельно создать макеты таблиц, сформировать схему БД, заполнить таблицы и приступить к созданию других объектов БД.

2. Один из *готовых шаблонов* MS Access (*например, Контакты, Управление мероприятиями, Отслеживание активов, Отслеживание задач* и др., рис. 2.2). В результате будет создана база с набором макетов объектов – таблиц (без записей), запросов, форм, отчетов.

После заполнения таблиц реальными данными можно использовать другие объекты БД для работы с данными.

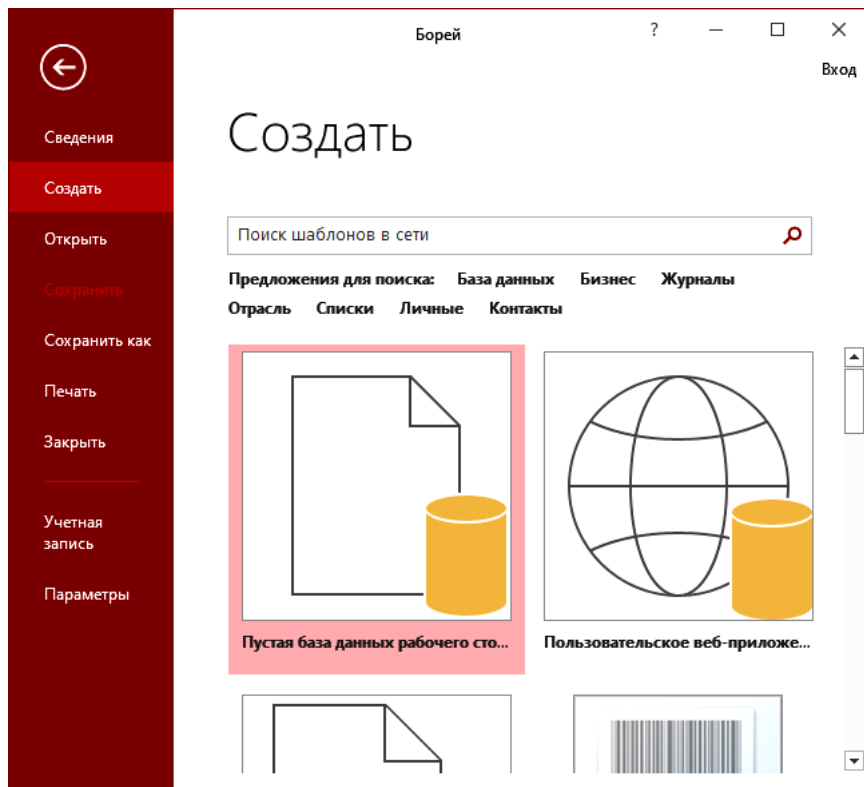


Рис. 2.2. Выбор шаблона для создания новой БД

2.3. Разработка макетов таблиц

Основу реляционной базы данных составляют взаимосвязанные таблицы. Поэтому при разработке базы данных «с нуля» (при выборе варианта «Пустая БД») в пустой базе данных создаются таблицы: в первую очередь – главные, затем – подчиненные.

Для создания таблицы в новой БД на вкладке *Создание* в группе команд *Таблицы* выбирают один из способов создания (рис. 2.3):

- 1) *Таблица*;
- 2) *Конструктор таблиц*.

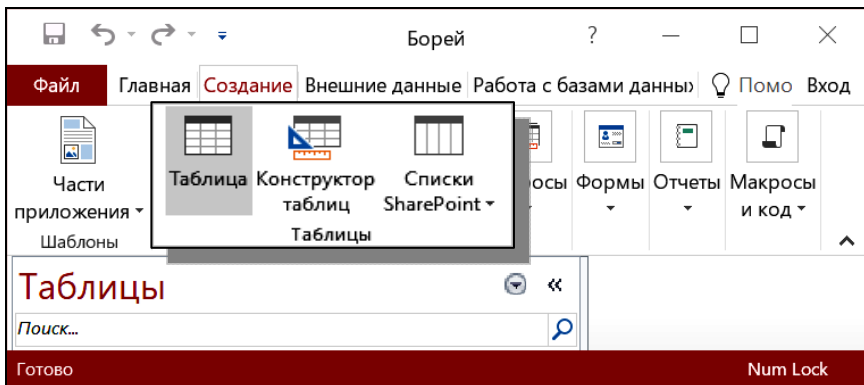


Рис. 2.3. Режимы создания таблиц БД

Режим таблицы – создает новую таблицу с именем *Таблица1* из одной записи, состоящей из одного поля *Код* и дополнительного столбца с именем «Щелкните для добавления» (рис. 2.4), в котором находится кнопка раскрывающегося списка для выбора типа данных нового поля.

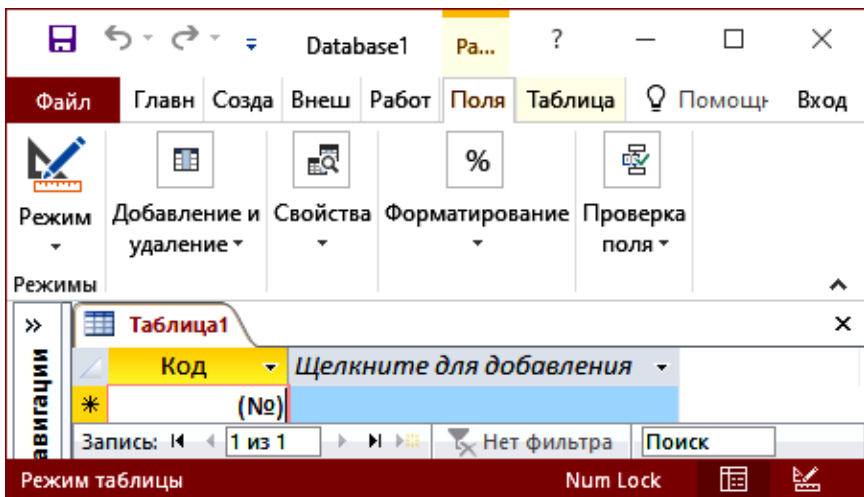


Рис. 2.4. Результат выполнения команды *Создание таблицы* в режиме *Таблицы*

При выборе в новом поле типа данных из раскрывающегося списка (рис. 2.5) или при вводе значения в новое поле, полю автоматически присваивается имя «Поле1» и появляется следующий столбец для добавления поля. Если же выполнить ввод данных в поле с именем «Щелкните для добавления», то при сохранении таблицы тип данных будет определен автоматически в соответствии с содержанием поля, а первое поле *Код* по умолчанию назначается ключевым (рис. 2.6).

Такой вариант создания подходит для простых таблиц, с небольшим числом полей и однотипными данными.

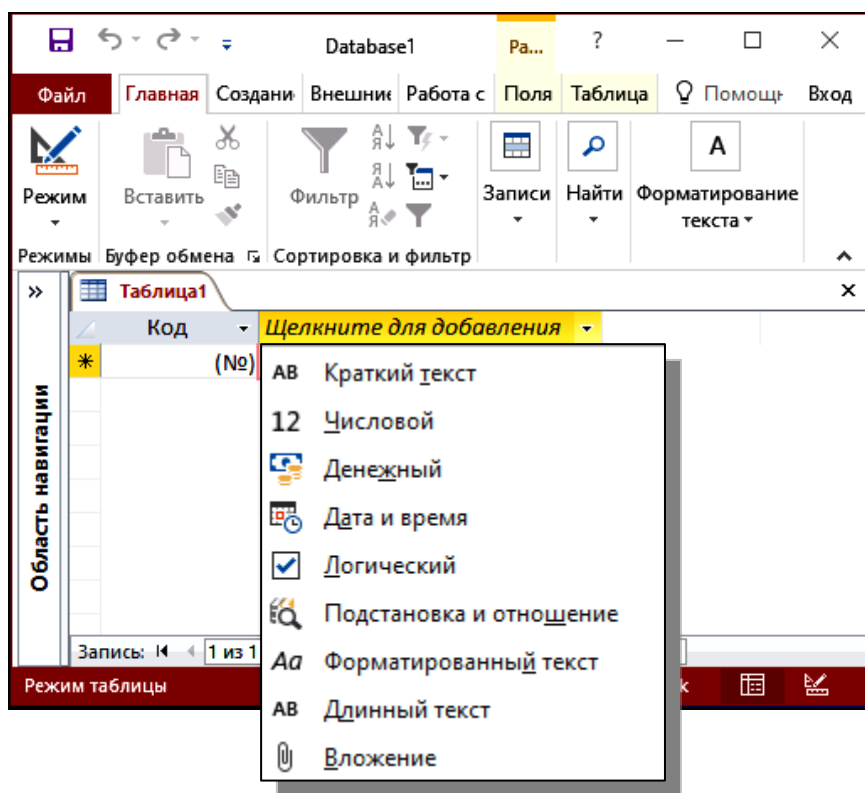


Рис. 2.5. Создание в режиме *Таблицы* нового поля путем выбора типа данных

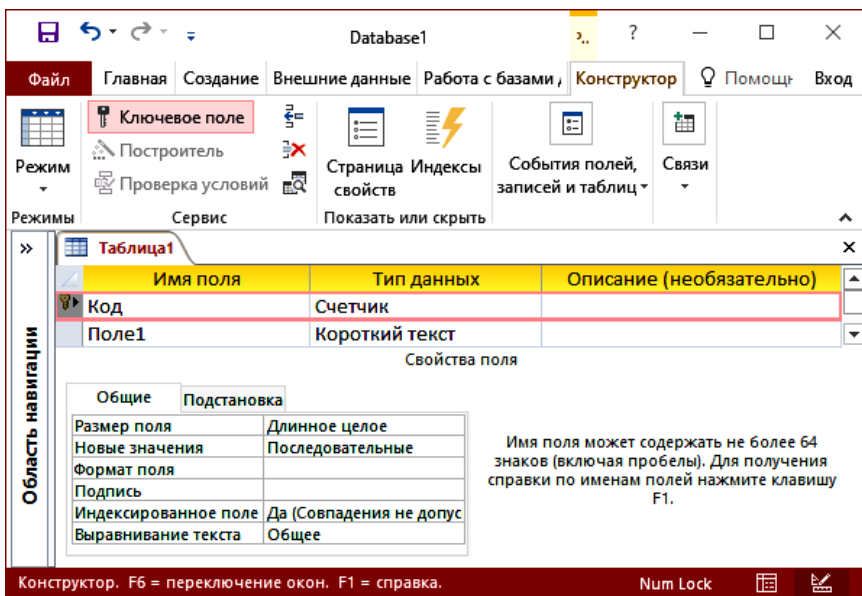


Рис. 2.6. Макет таблицы, созданной в режиме *Таблицы*, с ключевым полем *Код* типа *Счетчик*

Для профессиональной работы над макетом таблицы с возможностью выбора типа данных, настройкой свойств полей на вкладке *Создание* рекомендуется выбрать режим *Конструктор таблиц*.

При выборе режима *Конструктор таблиц* для создания новой таблицы открывается окно, содержащее три столбца: *Имя поля*, *Тип данных*, *Описание*, а также вкладки *Общие* и *Подстановка* в области *Свойства поля* (рис. 2.7).

Обязательными для заполнения являются: *Имя поля*, *Тип данных*, *Свойства*.

Необязательно, но рекомендуется заполнить *Описание* – содержание этого раздела будет отображаться в строке состояния при вводе данных в таблицу.

Имя каждого поля задается с учетом особенностей данных, которые будут храниться в поле, и *правил присвоения имен* (полей и таблиц):

1. Длина имени поля – до 64 символов.
2. Не допускаются символы: [] ! . (точка).

3. Пробелы могут быть, но не должны быть в начале имени.
4. Не могут присутствовать в имени управляющие символы (с кодами ASCII 00 – 31).

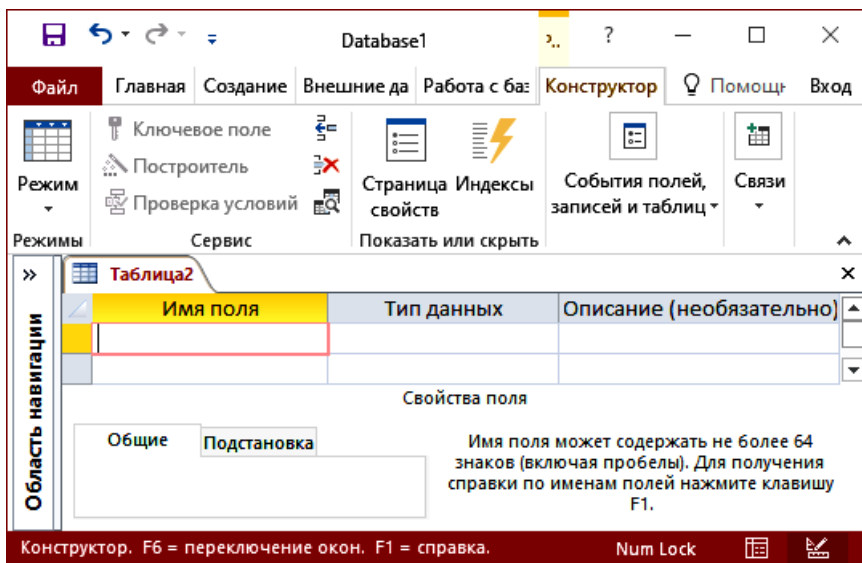


Рис. 2.7. Создание таблицы в режиме *Конструктора*

Для первого поля в таблице, которое обычно назначается ключевым, выбирают имя с учетом названия таблицы, например, *КодКлиента* – ключевое поле для таблицы *Клиенты*.

Как правило, разработчики БД – программисты задают имена таблиц и полей в виде: *tbl_Clients* (таблица *Клиенты*), *IdClient* (Код клиента), *Id_Region* (Код региона) и т. д. Однако в СУБД MS Access допускается запись имен объектов и полей на русском языке. В рамках данного пособия для большей наглядности и простоты восприятия информации при работе с объектами БД будут использоваться имена, записанные на русском языке.

При перемещении курсора в столбец *Тип данных* появляется список (рис. 2.8), из которого нужно выбрать необходимый тип в соответствии с хранимой в поле информацией (*сведения о типах данных приведены в п. 2.4*).

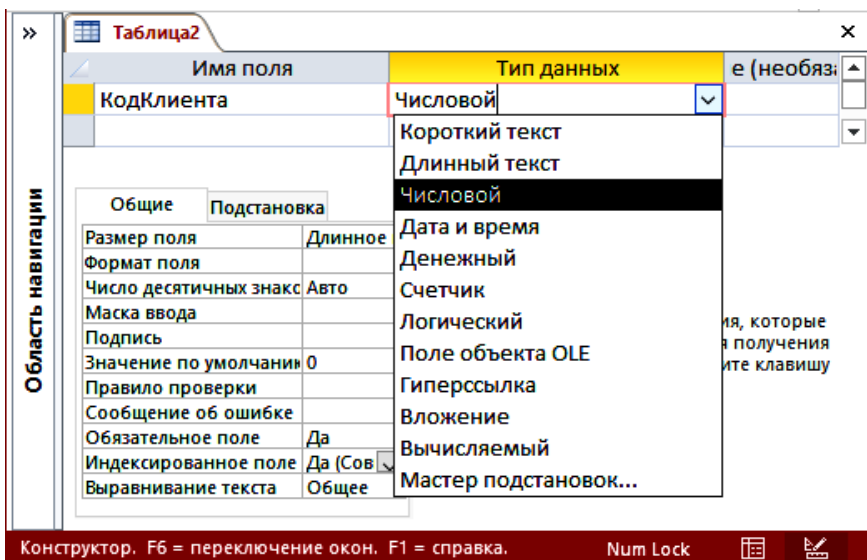


Рис. 2.8. Создание таблицы в режиме *Конструктора* (выбор типа данных для поля *КодКлиента*)

После выбора типа данных рекомендуется определить и *свойства поля* – набор характеристик, определяющих способ ввода, отображения, обработки и сохранения данных (*сведения о свойствах полей приведены в п. 2.5*).

До закрытия таблицы необходимо задать *поле первичного ключа* (для связи с другими таблицами) на вкладке *Конструктор* в группе команд *Сервис*, нажав кнопку «Ключевое поле».

Если ключевое поле не было задано, то автоматически при сохранении таблицы в качестве ключевого будет выбрано поле с типом данных *Счетчик* (если таковое присутствует в таблице), либо будет создано новое ключевое поле (имя поля – *Код*, тип данных – *Счетчик*).

При сохранении созданной таблицы указывается *имя таблицы*. Имя таблицы также должно удовлетворять вышеописанным правилам присвоения имен.

2.4. Типы данных MS Access

При проектировании макета таблицы для каждого поля необходимо выбрать подходящий тип данных.

Тип данных определяет набор допустимых значений в поле и операции над ними.

Как правило, тип данных выбирается в соответствии с тем, является это поле ключевым или описательным, какие характеристики объекта будут храниться в данном поле: качественные или количественные.

Например, для ключевых полей (первичных) чаще всего выбирают тип данных *Счетчик*, который обеспечивает автоматическую нумерацию записей. Для описательных полей, содержащих качественные характеристики, *например*, фамилию, имя, адрес, представляющие собой текстовые строки, выбирается соответствующий тип данных – *Короткий текст*, если длина текстовой строки не превышает 255 символов (рис. 2.9).

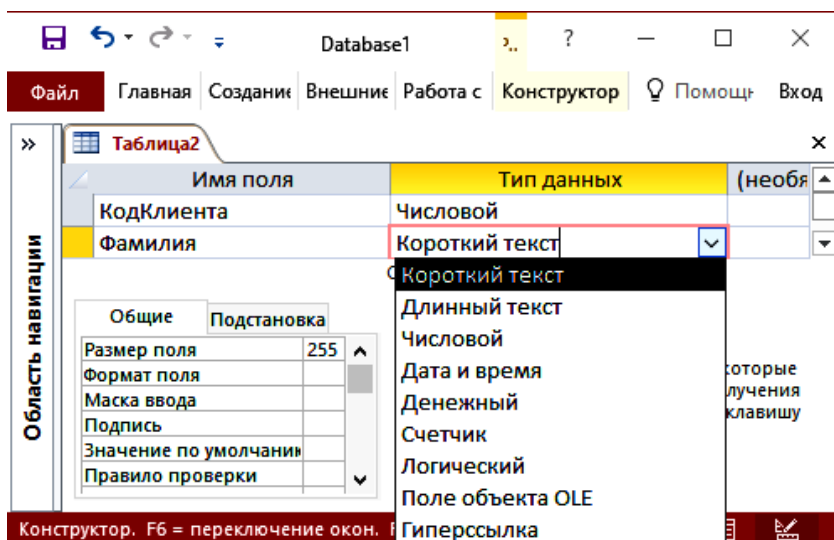


Рис. 2.9. Выбор типа данных для текстового поля *Фамилия*

В табл. 2.1 приведена краткая характеристика типов данных, которые определены в СУБД MS Access. В данной таблице приведены

примеры полей, для которых выбирается каждый из типов данных, ограничения по размеру поля для каждого типа, а также возможности индексирования полей и использования в качестве ключей (первичных или внешних).

Таблица 2.1

Краткая характеристика типов данных MS Access

Тип данных	Описание, примеры полей	Размер	Индексирование	Ключ
1	2	3	4	5
Короткий текст (Текст до 2013)	Текст, числа, не требующие вычислений, и их комбинации. <i>Например,</i> Номер телефона, Почтовый индекс, Адрес, Фамилия	До 255 символов	+	-/+ первичный, внешний
Длинный текст (МЕМО до 2013)	Длинный текст, комбинация текста и чисел. <i>Например,</i> Описание, Аннотация, Характеристика	До 63 999 символов	-	-
Числовой	Числа, используемые для математических вычислений (кроме денежных операций). Типы чисел и их точность зависят от свойства <i>Размер поля</i> (см. п. 2.5) <i>Например,</i> Количество, Год выпуска, Скидка (%)	1, 2, 4, 8, 12 байт (16 байт для кода репликации)	+/-	+ первичный, внешний

Продолжение табл. 2.1

1	2	3	4	5
Денежный	<p>Значения валют, используемые для денежных операций.</p> <p>Тип данных <i>Денежный</i> допускает 15 знаков целой части числа и 4 знака в дробной части, предотвращает округление при вычислениях.</p> <p>Обозначение валюты зависит от выбранной национальной настройки, которая устанавливается в Панели Управления.</p> <p><i>Например,</i> Цена товара, Стоимость аренды за сутки</p>	8 байт	+/-	-
Дата/время	<p>Значения дат и времени (от 100 до 9999 года).</p> <p>Значения в поле хранятся как десятичные числа (дата – целая часть числа, время – дробная, доля суток от полуночи).</p> <p>Вычисления выполняются в единицах измерения времени</p> <p><i>Например,</i> Дата заказа, Дата рождения</p>	8 байт	+/-	-

Продолжение табл. 2.1

1	2	3	4	5
Счетчик	<p>Уникальные последовательные или случайные числа в полях <i>первичного ключа</i>, которые вводятся в поле автоматически при создании новой записи.</p> <p>Значения в поле не могут быть изменены пользователем вручную.</p> <p>В таблице может быть <i>только одно поле</i> типа <i>Счетчик</i> (AutoNumber).</p> <p><i>Например,</i> Код клиента, Код заказа</p>	<p>4 байта 16 байт (код репликации)</p>	+	<p>+</p> <p>первичный</p>
Логический	<p>Одно из двух значений: Истина/Ложь, Да/Нет, Вкл/Выкл (On/Off). Обычно имеет вид флажков.</p> <p><i>Например,</i> Заказано, Оплачено</p>	1 бит	+/-	-
Поле объекта OLE	<p>Объекты, созданные в других приложениях, поддерживающих технологию OLE (документы, рисунки, аудиозаписи, видеоклипы и др.).</p> <p>Объекты могут быть связанными или внедренными.</p> <p><i>Например,</i> Отчет Word, Фото клиента, Видео, Аудиозапись звонка</p>	<p>До 1 Гбайта (размер ограничен доступным объемом диска)</p>	-	-

1	2	3	4	5
Вложение	Все поддерживаемые типы файлов, которые можно просматривать и редактировать. Рационально используется место на диске, так как не создается растровое изображение файла	Размер ограничен доступным объемом диска	–	–
Гиперссылка	Текст или сочетание текста и чисел, сохраненных как текст и используемых в качестве адреса гиперссылки. <i>Адрес гиперссылки</i> представляет собой <i>путь</i> к объекту, документу или веб-странице, размещенных в локальной или глобальной сети	До 2048 символов в каждой из 4-х частей	–	–
	<p>Адрес гиперссылки может состоять максимум из 4-х частей (каждая часть не более 2048 символов) и записывается в следующем формате:</p> <p>[экранный_Текст]#адрес#[доп_Адрес]#[всплывающая_Подсказка]</p> <p><i>Адрес</i> гиперссылки может представлять <i>URL-адрес</i> в Интернет (<i>например</i>, http://www.someone.homepage/def.html) или <i>сетевой маршрут UNC</i> к файлу в локальной сети (<i>например</i>, \\MyPC\Samples\Борей.accdb).</p> <p><i>Дополнительный адрес</i> гиперссылки может содержать некоторую специальную адресную информацию (<i>например</i>, имя объекта базы данных, закладку Microsoft Word или диапазон ячеек Microsoft Excel, на которые указывает адрес)</p>			

При выборе типа данных необходимо учитывать, что данные с меньшим размером занимают меньше дискового пространства, оперативной памяти и быстрее обрабатываются. Но слишком малый размер поля может привести к *искажению* или *потере данных*.

Вариант *Мастер подстановок* из списка типов данных не является самостоятельным типом данных, но позволяет создать поле со списком (столбец подстановки) для выбора значений из фиксированного набора значений, введенных пользователем, или из заданного поля таблицы БД (для внешнего ключа). После применения *Мастера подстановок* для поля устанавливается *тип данных источника подстановки*. Такие поля могут индексироваться.

Например, для поля *Обращение* в таблице с данными клиентов с помощью мастера подстановок, выбрав вариант создания списка подстановки с фиксированным набором значений, можно создать список из двух значений: *Уважаемый* и *Уважаемая* (рис. 2.10).

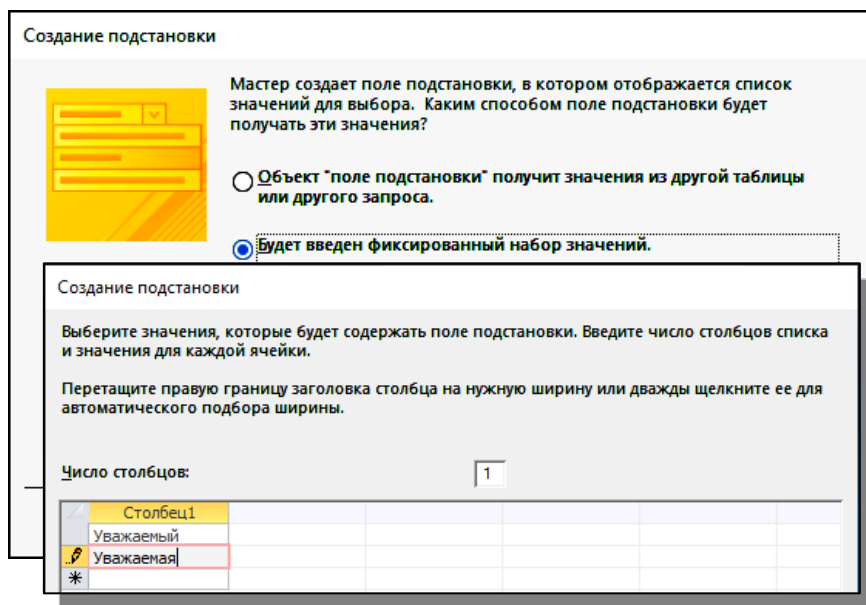


Рис. 2.10. Создание фиксированного набора для списка подстановки в поле *Обращение*

Тип данных для поля *Обращение* по умолчанию будет определен как *Короткий текст*, а на вкладке *Подстановка* отображаются все свойства сформированного поля со списком (рис. 2.11).

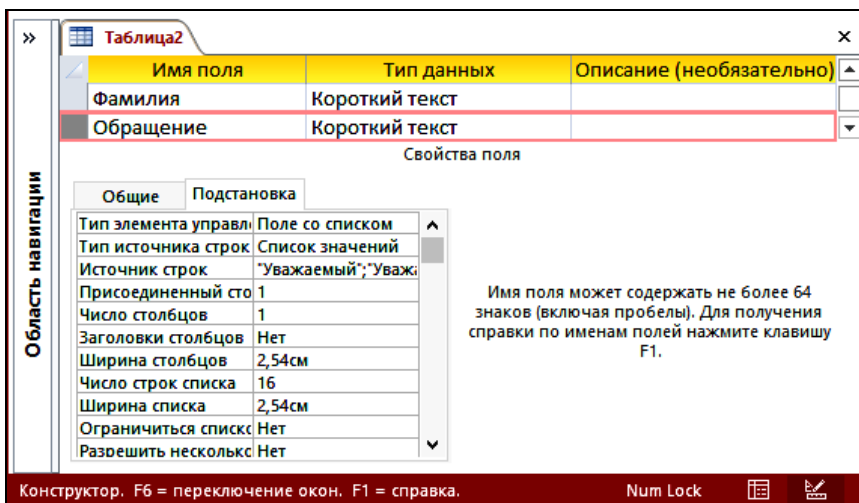


Рис. 2.11. Свойства списка подстановки в поле *Обращение*

В режиме *Таблицы* в поле, созданном с помощью *Мастера подстановок*, находится раскрывающийся список, позволяющий выбрать требуемое значение (рис. 2.12).

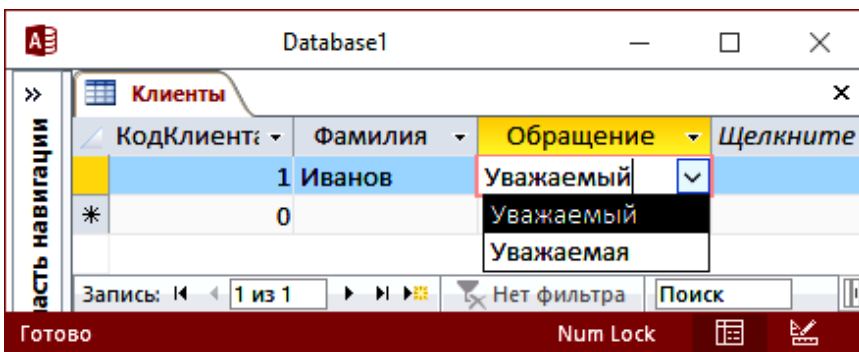


Рис. 2.12. Выбор значений из списка подстановки в поле *Обращение*

Мастер подстановок, как правило, используется при создании *внешних ключей* в подчиненной таблице, в этом случае в качестве источника строк для подстановки выбираются значения первичного ключа главной таблицы (рис. 2.13).

Рис. 2.13. Свойства списка подстановки в поле внешнего ключа *КодТипа*

2.5. Свойства полей

Каждое поле в таблице БД имеет *свойства* – набор характеристик, определяющих параметры отображения, обработки и хранения данных. Свойства поля управляют поведением значений в поле. Набор свойств поля зависит от выбранного типа данных. В табл. 2.2 приведена краткая характеристика свойств полей, которые определены в СУБД MS Access.

Таблица 2.2

Краткая характеристика основных свойств полей

Свойство, тип данных	Значение, описание
1	2
Размер поля	Определяет количество байт для хранения данных. Значение свойства по умолчанию устанавливается в настройках параметров БД (меню <i>Файл</i> , команда <i>Параметры</i> , вкладка <i>Конструктор объектов</i>)
Короткий текст	1 – 255 символов
Числовой	<i>Байт</i> (1 байт, целые числа от 0 до 255) <i>Целое</i> (2 байта, целые числа от –32 768 до 32 767) <i>Длинное целое</i> (4 байта, целые числа от –2 147 483 648 до 2 147 483 647) <i>Одинарное с плавающей точкой</i> (4 байта, действительные числа с точностью до 7 значащих цифр, от –3,410 ³⁸ до 3,410 ³⁸) <i>Двойное с плавающей точкой</i> (8 байт, действительные числа с точностью до 15 значащих цифр, от –1,79710 ³⁰⁸ до 1,79710 ³⁰⁸) <i>Код репликации</i> (16 байт, только при создании реплик БД для глобальных уникальных идентификаторов) <i>Действительное</i> (12 байт, действительные числа с заданной точностью, от –10 ²⁸ до 10 ²⁸) <i>Примечание:</i> при определении размера для числового поля следует учитывать, какие значения: целые или дробные будут храниться в поле. Например, для поля Скидка, %, выбирается размер <i>Одинарное с плавающей точкой</i>

1	2
Формат поля	Определяет способ отображения данных в поле, не влияет на хранение данных
Короткий текст	<p>Для задания способа отображения данных в поле используются специальные форматы, состоящие из двух секций (разделитель «;» – точка с запятой):</p> <p style="text-align: center;"><i>текст; пустые значения</i></p> <p>Для создания формата используются коды: @ – обязательный символ (@@@-@@-@@ – № тел.); & – не обязательный символ; > – преобразует буквы в прописные; < – преобразует буквы в строчные</p>
Числовой, Денежный, Счетчик	<p>Основной Денежный Евро Фиксированный С разделителями разрядов Процентный Экспоненциальный <i>Специальный</i> формат, создаваемый пользователем, состоит из 4 секций (разделитель «;» – точка с запятой): <i>положительные; отрицательные; ноль; пустые значения</i> <i>Например,</i> # ##0,0” \$”; -# ##0,0” \$”; 0 – пользовательский формат, отображающий положительные и отрицательные значения с разделителями групп разрядов, с одним десятичным знаком и обозначением валюты (\$), а ноль – в виде 0</p>
Дата/время	<p>Полный формат даты (дата + время) Длинный формат даты (из Панели управления) Средний формат даты Краткий формат даты (не рекомендуется) Длинный формат времени Средний формат времени Краткий формат времени</p>
Логический	<p>Да/Нет (по умолчанию) Истина/Ложь Вкл/Выкл</p>

1	2
Число десятичных знаков	Число знаков справа от десятичной запятой
Числовой, Денежный	0–15 (по умолчанию – Авто)
Подпись Все типы	Используется как заголовок поля в режиме <i>Таблицы</i> , в элементах управления форм и отчетов. Чаще всего совпадает с именем поля
Маска ввода	Определяет формат ввода данных в поле (в отличие от формата поля)
Короткий текст, Дата/Время	Для создания маски ввода используются специальные коды: 0 – обязательная цифра; 9 – необязательная цифра или пробел; # – необязательная цифра, пробел, знаки плюс, минус; L – обязательная буква; ? – необязательная буква; A – обязательная буква или цифра; a – необязательная буква или цифра; & – обязательный произвольный символ или пробел; C – необязательный произвольный символ или пробел; > – преобразует все символы справа к верхнему регистру; < – преобразует все символы справа к нижнему регистру; ! – указывает, что маску заполняют справа налево (символы слева являются необязательными); \ – следующий символ воспринимать буквально; «литерал» – литерал в кавычках воспринимается буквально. <i>Например,</i> >LLLL – маска ввода для текстовых значений из 4 прописных букв; (999)000-000 – маска ввода для телефонного номера
Значение по умолчанию	Значение, которое автоматически отображается в поле при создании каждой новой записи
Короткий текст, Дата/время, Гиперссылка	По умолчанию используется значение «NULL»

1	2
Числовой, Денежный	По умолчанию используется значение «0»
Логический	По умолчанию используется значение «Ложь»
Правило проверки Все, кроме Счетчик, Поле объекта OLE, Вложение	<p><i>Условие</i>, которое должно быть истинным при вводе или редактировании данных (в таблице, форме, запросе).</p> <p><i>Правило проверки</i> представляет собой <i>выражение</i>, которое состоит из <i>операторов сравнения</i> (например, =, >, >=, <, <=, <>) и <i>операндов</i> (констант, литералов, идентификаторов объектов, функций).</p> <p>Операторы сравнения с образцом (см. п. 3.3): Like, In, Is. Between ... And ... – оператор для проверки попадания значения в заданный диапазон (например, если предоставляемая скидка не превышает 70 %, правило проверки для поля <i>Скидка</i> имеет вид: Between 0 And 0,7).</p> <p>Несколько сравнений могут быть связаны при помощи логических операторов AND, OR (>=50 AND <=100)</p> <p>Для формирования условия обычно используют <i>Построитель выражений</i>.</p> <p>Если вводимые данные не соответствуют заданному условию, то выводится значение свойства <i>Сообщение об ошибке</i> (если не задано – системное сообщение)</p>
Сообщение об ошибке	Текст, который выводится в диалоговом окне MS Access, если вводимые данные не соответствуют заданному
Все, кроме Счетчик, Поле объекта OLE, Вложение	<i>Правило проверки</i>
Обязательное поле	Если нельзя оставить поле незаполненным, устанавливаются значение «Да» в этом случае в поле не допускается значение «NULL»
Все, кроме Счетчик	
Пустые строки	Свойство позволяет ввести в поле пустую строку – "" (две пары двойных кавычек без пробела), что будет означать, что для поля не существует значения.
Короткий текст, Длинный текст	Позволяет различать поля с пустыми значениями типа NULL – незаполненное или неизвестное значение, от полей, не имеющих значений – "" (пустая строка)

1	2
Индексированное поле	Индексирование ускоряет доступ к данным, хранящимся в поле.
Короткий текст, Числовой, Дата/время, Денежный, Счетчик, Логический	Индексирование применяется к полям, для которых часто выполняются операции <i>поиска, сортировки, группировки</i> . <i>Например</i> , Страна, Город, Фамилия. <i>Индекс</i> – это специальная таблица, содержащая в упорядоченном виде возможные значения поля и соответствующие им местоположения записей с данным значением; выполняет роль оглавления или указателя; обеспечивает быстрый доступ к данным в поле. Индексы могут быть созданы по одному или нескольким полям. <i>Первичные ключи индексируются автоматически</i> , по умолчанию свойству присваивается значение: «Да (Совпадения не допускаются)». Для <i>внешних</i> ключей для свойства выбирают значение: «Да (Допускаются совпадения)»
Сжатие Юникод	Позволяет сжимать до 1 байта все символы, первый байт которых в кодировке Unicode равен 0.
Короткий текст, Длинный текст	При сохранении происходит сжатие, при выборке – восстановление значения. По умолчанию установлено значение «Да»

На рис. 2.14–2.17 приведены примеры полей различных типов с заданными свойствами, включая формат поля, значение по умолчанию, правило проверки и сообщение об ошибке.

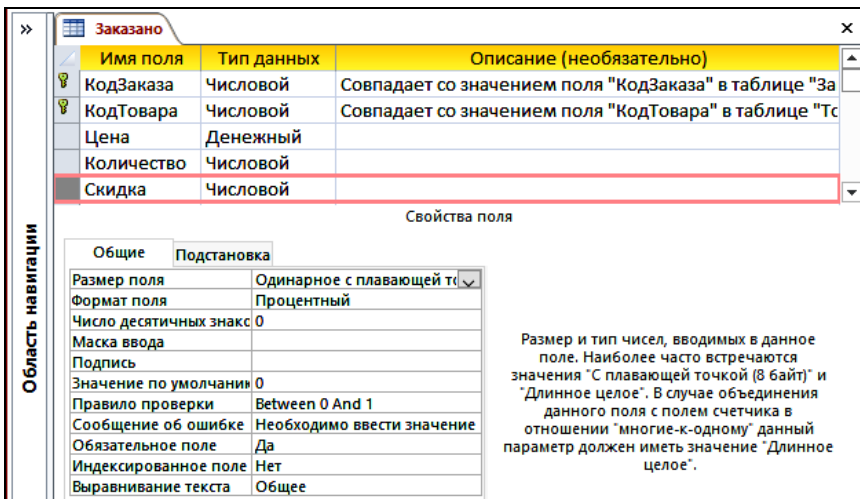


Рис. 2.14. Свойства поля *Скидка* (тип – Числовой)

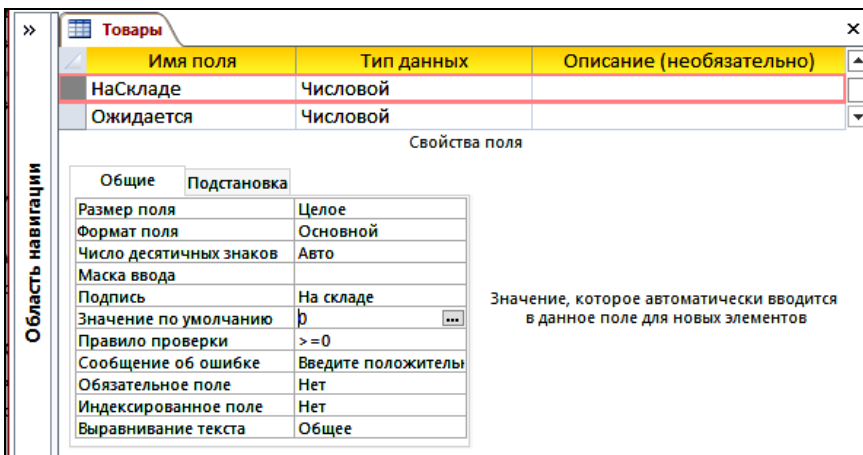


Рис. 2.15. Свойства поля *На складе* (тип – Числовой)

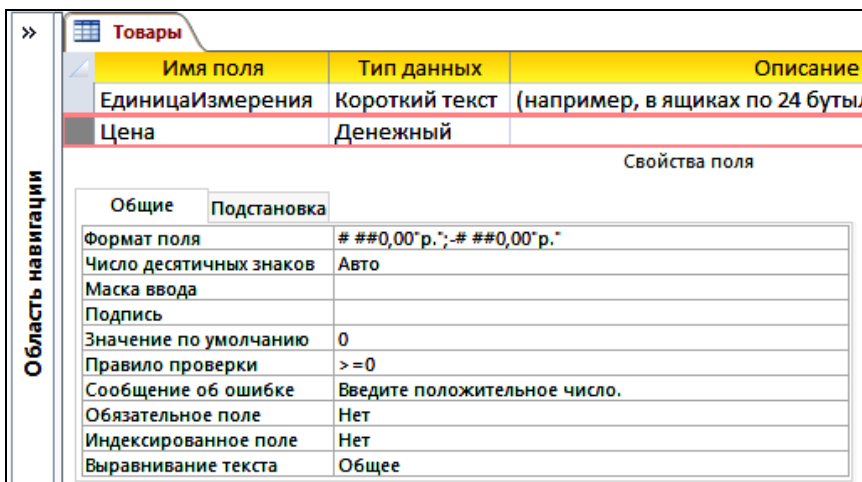


Рис. 2.16. Свойства поля *Цена* (тип – Денежный)

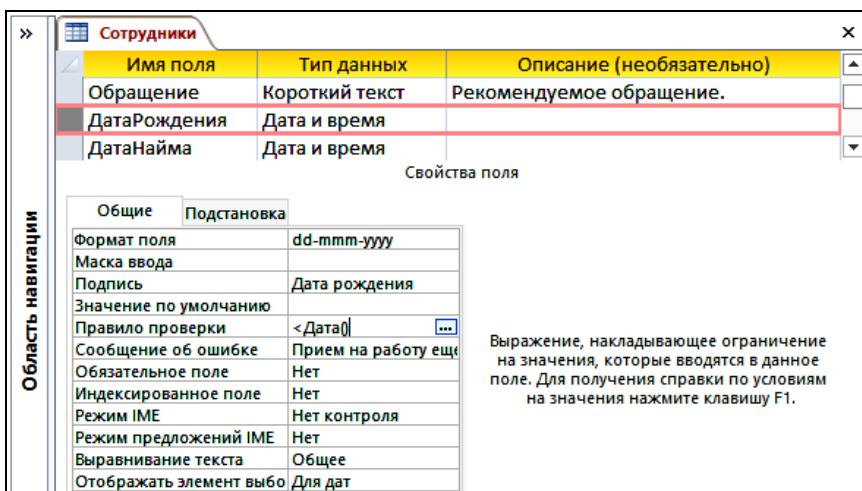


Рис. 2.17. Свойства поля *Дата рождения* (тип – Дата и время)

2.6. Порядок формирования схемы БД

Схема БД представляет собой графическое отображение информационно-логической модели предметной области.

Установленные в схеме БД связи между таблицами обеспечивают корректную обработку данных при эксплуатации БД.

Для отображения схемы БД необходимо на вкладке *Работа с базами данных* в группе команд *Отношения* нажать кнопку «Схема данных».

Пример схемы учебной БД «Борей» приведен на рис. 2.18.

Формирование схемы обычно выполняют после завершения работы над всеми таблицами БД, соблюдая следующий порядок:

1. Закрыть все таблицы БД (создать связи при открытых таблицах нельзя).
2. На вкладке *Работа с базами данных* в группе команд *Отношения* нажать кнопку «Схема данных».
3. На вкладке *Конструктор* в группе команд *Связи* нажать кнопку «Отобразить таблицу», далее в диалоговом окне «Добавление таблицы» выбрать таблицы и нажать кнопку «Добавить». После добавления таблиц закрыть диалоговое окно.
4. Далее в окне «Схема данных» упорядочить главные и подчиненные таблицы: рекомендуется главные таблицы расположить выше подчиненных, а таблицу-связку расположить уровнем ниже в центре между главными таблицами.
5. Для формирования связи между таблицами необходимо перетаскать поле первичного ключа из главной таблицы на поле внешнего ключа подчиненной таблицы.
6. В диалоговом окне «Изменение связей» (рис. 2.19) необходимо проверить корректность связи (совпадение по полям) и установить флажки в группе *Обеспечение целостности данных*.
7. Повторить шаги 5–6 для оставшихся таблиц.
8. Сохранить макет схемы данных.

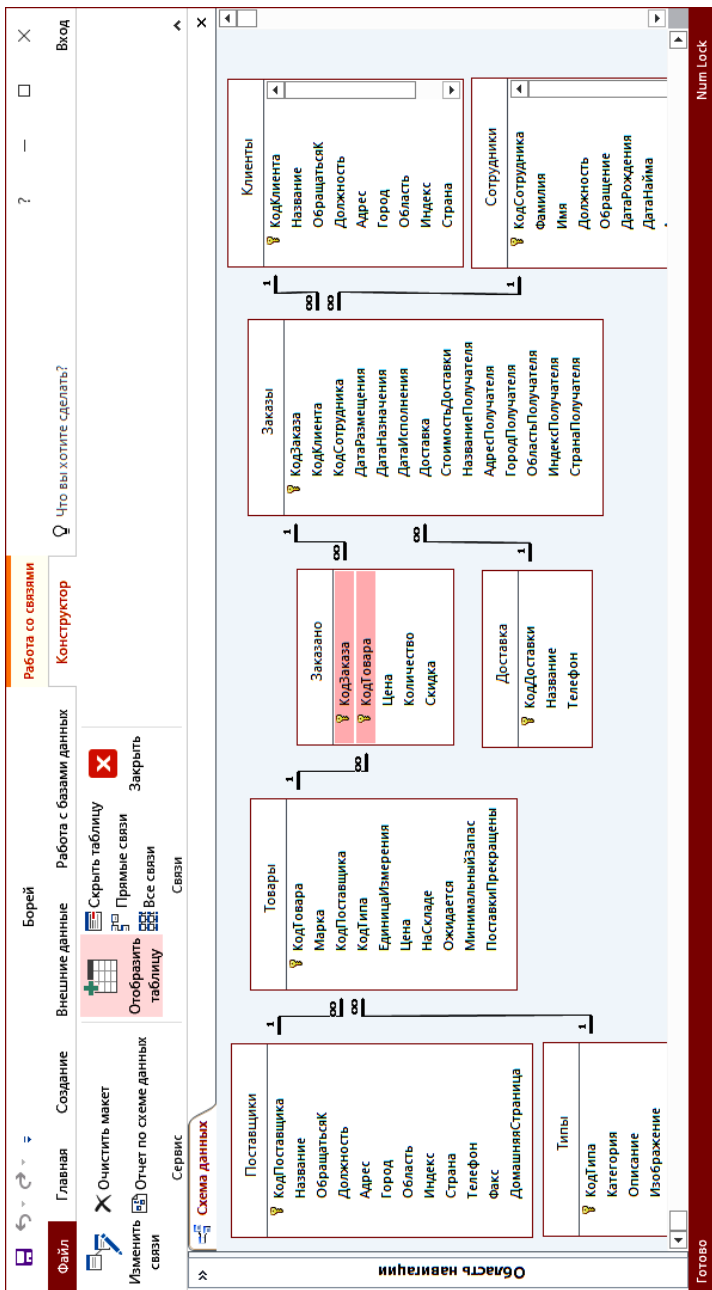


Рис. 2.18. Схема учебной БД Борей и вкладка модульной ленты Конструктор из набора Работа со связями

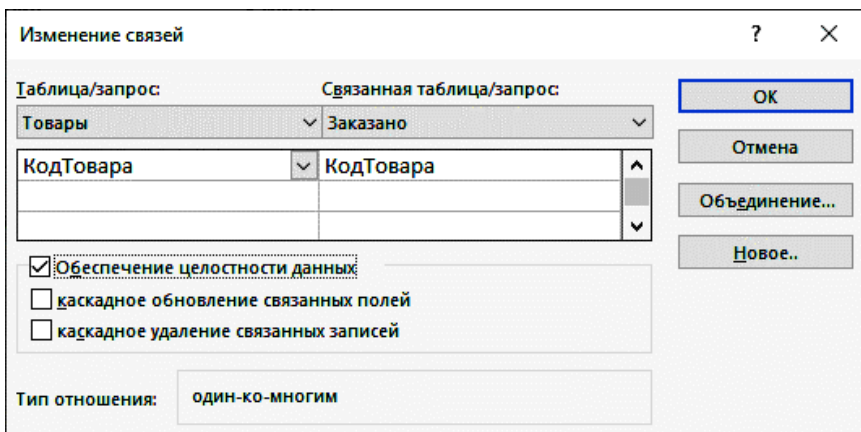


Рис. 2.19. Изменение межтабличных связей

Для успешного формирования связи между таблицами необходимо, чтобы ключевые поля (первичные и внешние) имели одинаковый тип данных, содержимое одного типа и одинаковый размер поля. *Существует исключение*: поле счетчика можно связать с числовым полем, соблюдая некоторые требования (табл. 2.3).

Таблица 2.3

Требования к ключевым полям
при формировании межтабличных связей

Характеристика поля	Первичный ключ (главной таблицы)	Внешний ключ (подчиненной таблицы)
1. Тип данных	Счетчик	Числовой
2. Размер поля	Длинное целое	Длинное целое
3. Обязательное поле	(по умолчанию – Да)	Да
4. Индексированное поле	Да, совпадения не допускаются	Да, допускаются совпадения

Вопросы для самопроверки

1. Какая модель данных используется в СУБД MS Access?
2. Перечислите способы создания БД в СУБД MS Access. Какие расширения файлов может иметь БД, созданная с помощью СУБД MS Access?
3. Какие типы объектов в БД могут быть созданы в СУБД MS Access? С создания каких объектов начинают работу над БД?
4. Сформулируйте назначение режима *Конструктора таблиц* и его возможности для работы над макетом таблицы БД.
5. Какой длины может быть имя поля и имя таблицы? Какие символы нельзя использовать в именах полей и таблиц?
6. Что понимают под «типом данных»? Как выбирают тип данных для конкретного поля? Приведите краткую характеристику типов данных (с примерами их выбора для конкретных полей).
7. Какие типы данных рекомендуется выбирать для первичных ключевых полей? Как правильно выбрать тип данных для внешнего ключа? Как задается ключевое поле в таблице?
8. Для чего применяется «Мастер подстановок»? Каким образом могут быть получены значения для подстановки?
9. Что понимают под «свойствами поля»? Какое свойство поля определяет тип чисел и их точность, длину текстовой строки?
10. Для чего используется свойство *Формат поля*? Приведите примеры форматов для числовых значений, значений типа даты/времени? Какой код формата позволит отобразить текстовые значения в верхнем/нижнем регистре?
11. Какие свойства рекомендуется использовать для предотвращения ошибок при вводе данных? Приведите примеры.
12. Для каких полей выполняется индексирование? Как индексируются первичные и внешние ключи?
13. Что понимают под схемой БД? Как она формируется в СУБД MS Access?
14. Какие условия должны быть соблюдены для успешного формирования межтабличных связей в схеме БД?

Тема 3. АНАЛИЗ ДАННЫХ В РЕЛЯЦИОННЫХ БД

- 3.1. Инструменты анализа данных. Назначение и виды запросов.
- 3.2. Способы создания запросов на выборку.
- 3.3. Формирование условий отбора в запросах.
- 3.4. Вычисляемые поля в запросах.
- 3.5. Применение функций MS Access.
- 3.6. Использование групповых операций в итоговых запросах.
- 3.7. Создание перекрестных запросов.
- 3.8. Применение параметров в запросах.
- 3.9. Назначение и особенности запросов действия. Методика их формирования.

3.1. Инструменты анализа данных. Назначение и виды запросов

Для анализа данных в БД могут применяться такие инструменты, как *сортировка*, *фильтрация* и *запросы*. Основным инструментом анализа данных в БД являются *запросы*.

Запрос – объект СУБД MS Access, предназначенный для анализа данных из таблиц БД.

Назначение запросов:

- выбор по условиям отбора конкретной информации из одной или нескольких взаимосвязанных таблиц;
- выполнение вычислений и представление результатов в виде таблицы (в т. ч. с использованием функций);
- группирование записей с одинаковыми значениями в полях с применением групповых (статистических) функций;
- обновление данных в исходных таблицах, добавление и удаление записей;
- создание новых таблиц БД на основе данных из существующих таблиц.

Запросы создаются на основе:

- исходных таблиц БД;
- таблиц, полученных в результате выполнения запросов на создание таблиц;
- результатов других запросов (временных таблиц с результатами).

Классификацию запросов MS Access (рис. 3.1) можно выполнить по 2-м признакам: по методу создания и по функциональному назначению.



Рис. 3.1. Виды запросов

1. По методу создания запросы бывают:

1) *QBE-запросы (Query By Example)* – запросы по образцу, которые создаются при помощи *Конструктора запросов* – графического средства MS Access, включающего подсхему данных и бланк запроса. Для формирования макета запроса по образцу достаточно добавить таблицы с источниками данных в подсхему данных, перемес-

тить при помощи мыши необходимые поля из таблиц в бланк запроса, добавить условия отбора данных, задать функциональный тип запроса (по умолчанию создается запрос типа выборки);

2) *SQL-запросы (Structured Query Language)* – запросы, для формирования которых применяются специальные инструкции языка структурированных запросов SQL-инструкции (*дополнительные сведения об SQL приведены в п. 6.1–6.4*).

QBE-запрос может быть трансформирован в SQL-запрос с помощью команды *Режим SQL*, которая выполняется на вкладке *Конструктор* из набора вкладок *Работа с запросами* в группе команд *Результат* после нажатия кнопки «Режим». А изменения, вносимые в SQL-инструкции, автоматически отображаются в режиме *Конструктора запроса*.

2. По функциональному признаку запросы можно разделить на две группы:

1) *запросы выборки данных (запросы выборки)* – формируют результат обработки данных в виде набора записей *временной таблицы*, существующей до закрытия запроса. Причем, в отличие от таблиц, эти временные наборы записей не существуют физически в базе данных, но сам запрос сохраняется как объект в БД.

Запросы выборки, в свою очередь, могут быть:

а) простые;

б) с условиями;

в) с вычислениями;

г) с параметрами (особенность таких запросов заключается в том, что значения, необходимые для формирования условия отбора или вычисляемого выражения в запросе, задаются пользователем в диалоговом окне непосредственно при выполнении запроса);

д) с групповыми операциями (такие запросы обеспечивают статистическую обработку данных по группе записей с применением агрегатных функций: Sum, Count, Avg и др.);

е) перекрестные – обеспечивают обработку данных с применением статистических функций и их компактное отображение путем объединения однотипной информации по строкам и столбцам, на пересечении которых находится итоговое значение.

Например, Строка – Продавец; столбец – Покупатель, Итоговое значение – Сумма заказов.

2) *запросы действия* (или *запросы на изменение*) – приводят к изменениям данных в исходных таблицах или созданию новых таблиц, сохраненных как новый объект.

Различают запросы на:

- а) создание таблицы;
- б) обновление данных;
- в) добавление данных;
- г) удаление данных.

Результаты обработки информации по запросу представляют собой:

– *для запроса выборки* – временная таблица, содержащая набор записей, удовлетворяющих критериям запроса. Такое подмножество данных, отображаемых запросом, называется *динамическим набором данных* (DYNASET), поскольку при каждом выполнении запроса отображает «свежие» данные и *существует только до закрытия запроса* (макет запроса сохраняется как объект БД);

– *для запроса действия* – изменение данных в базовых таблицах или новый объект в БД – таблица.

Поля в таблице с результатами запроса наследуют свойства полей исходных таблиц.

Последовательное выполнение ряда QBE-запросов позволяет решать сложные задачи, не прибегая к традиционному программированию.

3.2. Способы создания запросов на выборку

Создание QBE-запросов в СУБД MS Access выполняется на вкладке *Создание* в группе команд *Запросы* в двух режимах:

1. Режим *Мастер запросов*.

После нажатия кнопки «Мастер запросов» в диалоговом окне необходимо выбрать вариант создаваемого запроса (рис. 3.2).

В режиме *Мастера запросов* пошаговые инструкции позволяют сформировать следующие виды запросов:

– *простой запрос* на выборку данных, который может включать данные из одной или нескольких взаимосвязанных таблиц. Выбор источника запроса и полей для запроса выполняется в диалоговом окне, отличительная особенность заключается в том, что непосред-

ственно в диалоговых окнах при построении простого запроса в режиме Мастера нельзя задать условия отбора;

- *перекрестный запрос* (компактное отображение данных в формате строка-столбец, аналог сводной таблицы MS Excel);
- *запрос для поиска повторяющихся записей* (повтор значений в заданных полях, *например*, клиенты из одной страны или города);
- *запрос для поиска записей без подчинения* (*например*, позволяет найти клиентов, не имеющих заказов).

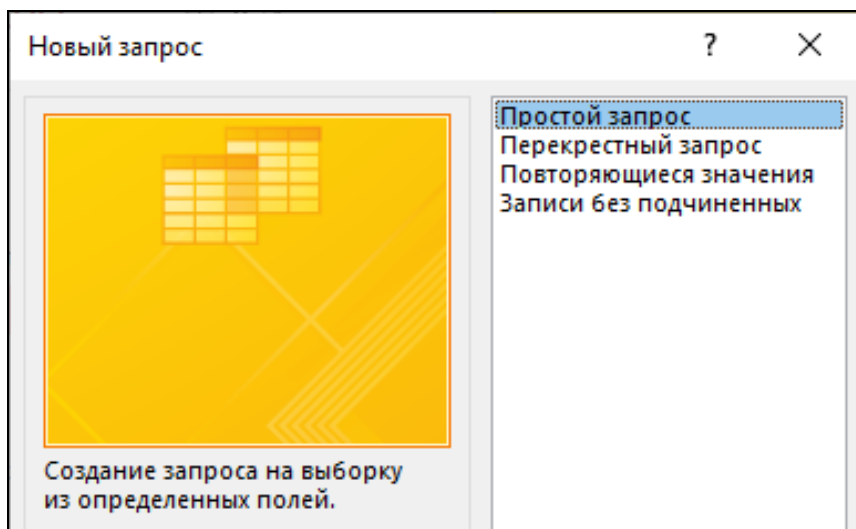


Рис. 3.2. Режимы *Создания запросов*

После выбора одного из вариантов создания запроса и нажатия кнопки «ОК» открывается соответствующее диалоговое окно «Создание ... запроса», в котором выбирают источник для отбора полей и формируют структуру запроса.

2. Режим *Конструктор запросов*.

Если выбрать режим *Конструктора*, то открывается окно *Конструктора запроса* (рис. 3.3), которое имеет 2 области: *Подсхему данных* и *Бланк запроса*.

Подсхема данных содержит исходные таблицы (в виде списка полей) с установленными связями. Если связи между таблицами отсутствуют, то для добавления недостающих таблиц-связок

в подсхему данных следует использовать команду *Добавить таблицу* из контекстного меню.

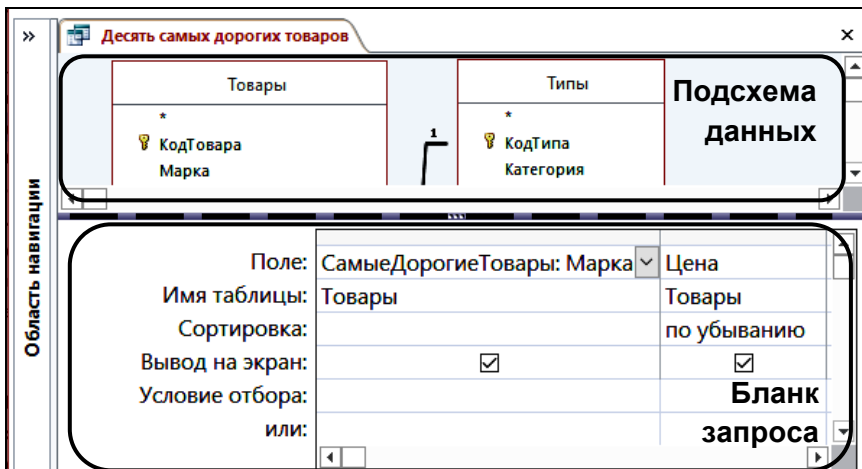


Рис. 3.3. Окно Конструктора запроса

Кроме того, результат запроса зависит от вида соединения таблиц (внутренне или внешнее), выбранного для каждой связи в диалоговом окне «Параметры объединения» (рис. 3.4).

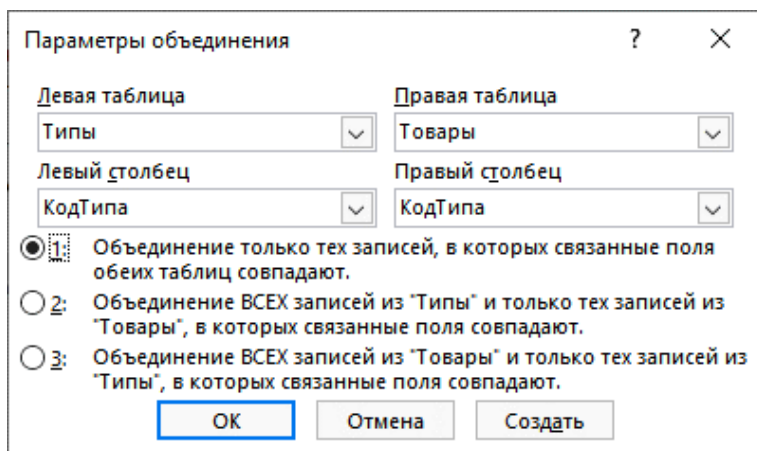


Рис. 3.4. Окно Конструктора запроса

Бланк запроса по образцу (нижняя панель) содержит *столбцы* (имена полей, включаемых в запрос) и *строки*:

– *Поле* – отображает имя поля, выбранного из таблицы, или имя вычисляемого поля (или выражение);

– *Имя таблицы* – отображает имя таблицы, из которой выбрано поле, или остается пустой для вычисляемого поля;

– *Сортировка* – определяет порядок и поля сортировки (до 3-х полей в запросе) и содержит список из трех значений: по возрастанию, по убыванию, (отсутствует);

– *Вывод на экран* – содержит флажок, который по умолчанию установлен и обеспечивает вывод выбранного поля в результирующей таблице при выполнении запроса, а если поле участвует в запросе, но не должно выводиться, флажок следует снять;

Условие отбора / или – в этих строках записываются инструкции (см. п. 3.3), при помощи которых отбираются записи, включаемые в динамический набор данных.

Таким образом, именно строка *Поле* определяет состав и порядок следования полей, включенных в запрос и отображаемых в результате его выполнения (в динамическом наборе данных). В строке могут быть как имена полей из таблиц, так и имена полей с вычислениями, по любому из них могут быть заданы условия для выборки и порядок сортировки, при этом отдельные поля могут не отображаться в результирующей таблице.

Добавить поля из таблиц в *Бланк запроса* можно несколькими способами:

1. Выбрать поле из раскрывающегося списка в строке *Поле* (при этом имя соответствующей таблицы автоматически появится в строке ниже).

2. Выполнить двойной щелчок левой кнопкой мыши (ЛКМ) по полю непосредственно в таблице, добавленной в подсхему данных. При этом выбранное поле будет размещено в столбце справа от последнего добавленного в запрос поля.

3. Выделить поле непосредственно в таблице, добавленной в подсхему данных и, удерживая ЛКМ, перетащить выделенное поле на любой столбец в *Бланке запроса*.

4. Для добавления в *Бланк запроса* *всех полей* из таблицы необходимо выбрать «*» (звездочку) непосредственно в таблице, добавленной в подсхему данных, и перетащить ее в область *Бланка за-*

проса. Следует учитывать, что если в *Бланк запроса* включены все поля с использованием «*», то *нельзя задать условия отбора*.

Строка *Условие отбора* позволяет задать инструкции (в виде *выражения*, состоящего из операндов и операторов сравнения), при помощи которых выбираются записи, включаемые в динамический набор данных.

Если условия отбора в *Бланке запроса* расположены на одной строке, то к ним применяется логическая операция И, а если на разных строках, то к условиям применяется логическая операция ИЛИ.

Для полей, включаемых в запрос, можно с помощью КЗМ отобразить *Окно свойств* и дополнительно установить или изменить значения свойств (подпись, формат и др., рис. 3.5).

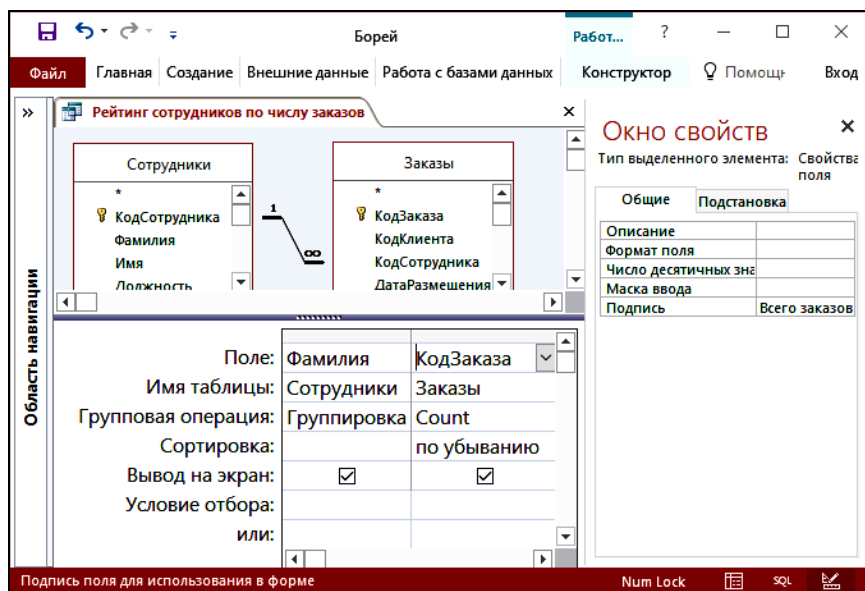


Рис. 3.5. Свойство *Подпись* для поля *КодЗаказа* в запросе выборки

Также с помощью КЗМ в области подсхемы данных можно отобразить *Окно свойств* для запроса, в котором задать свойства самого запроса, например, ограничить число выводимых в результате запроса записей, изменив значение свойства *Набор значений* (рис. 3.6).

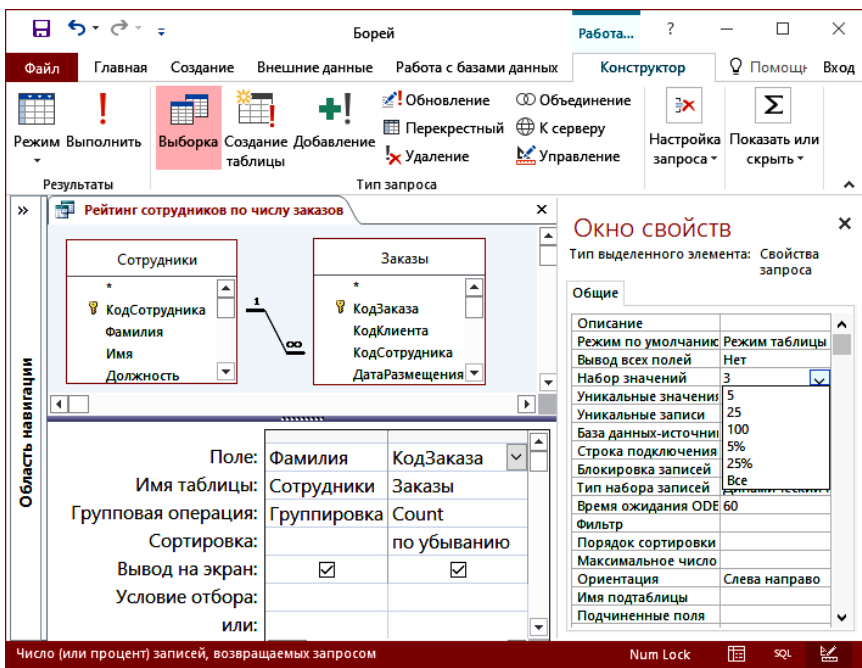



Рис. 3.6. Изменение свойства запроса *Набор значений*

После заполнения *Бланка запроса* необходимо выполнить запрос, нажав на вкладке *Конструктор запросов* кнопку «Выполнить»  или выбрать режим отображения *Таблица*.

При закрытии окна запроса необходимо подтвердить сохранение запроса и присвоить *имя*, изменив имя по умолчанию – Запрос 1.

3.3. Формирование условий отбора в запросах

Условие отбора (УО) представляет собой *логическое выражение*, состоящее из *операндов* и *операторов сравнения*, в результате работы которого возвращается единственное из значений – ИСТИНА (True) или ЛОЖЬ (False). Если возвращаемое значение – ИСТИНА, то запись включается в результат работы запроса, в противном случае – нет.

Операнды – это объекты, над которыми выполняются действия. В качестве операндов в выражениях могут использоваться:

- 1) *константы*: "" (пустая строка), Null, Истина, Ложь;
- 2) *литералы*: заданные пользователем числовые или текстовые значения, значения даты/времени, *например*, 100, Минск, 01.01.2024;
- 3) *идентификаторы объектов* БД: имена полей в таблицах, отчетах и формах в формате [Имя объекта]![Имя поля], *например*, [Товары]![Цена];
- 4) *функции* (сведения о функциях приведены в п. 3.5).

Операторы обозначают *действия*, выполняемые над операндами (табл. 3.1). В MS Access формально все операторы MS Access представлены в 4-х категориях: *арифметические* операторы, *строковый (текстовый)* оператор, операторы *сравнения* и *логические* операторы. Особую группу составляют *операторы идентификации*, которые используются для формирования имен объектов и имеют самый высокий приоритет.

Таблица 3.1

Операторы MS Access

Оператор	Описание	Пример
1	2	3
<i>1. Операторы идентификации объектов (результат – объект)</i>		
«!»	Позволяют объединить имена объектов, чтобы идентифицировать поля в таблицах, выполнять отбор объектов	Класс Объекта! Имя Объекта
«.»		Имя объекта.Метод
<i>2. Арифметические операторы (результат – число)</i>		
– унарный минус	Меняет знак операнда на противоположный	–123456
^ возведение в степень	Возводит основание степени в показатель степени	n^2 [Основание]^ [Показатель]
\ целочисленное деление	Делит один операнд на другой нацело	[Количество] \ 3

Продолжение табл. 3.1

1	2	3
Mod остаток от деления	Возвращает остаток от деления операнда на число	[Количество] Mod 10
* и / умножение и деление	Перемножает операнды. Делит первый операнд на второй	[Товары]![Цена]*1,2 [Количество] / 3
+ и – сложение и вычитание	Суммирует операнды. Из первого операнда вычитает второй операнд	[итог] + [надбавка] Date() – 7
<i>3. Оператор объединения текстовых строк (результат – строка)</i>		
& (амперсанд) конкатенация	Объединяет текстовые операнды в единую текстовую строку	"Проживает по адресу: " & [Клиенты]![Адрес]
<i>4. Операторы сравнения (результат – True or False)</i>		
=	Равно	100 = 100 (True)
<	Меньше	10 < 100 (True)
>	Больше	10 > 100 (False)
>=	Больше либо равно	100 >= 10 (True)
<=	Меньше либо равно	100 <= 10 (False)
<>	Не равно	10 <> 100 (True)
<i>5. Операторы сравнения с образцом (результат – True or False)</i>		
Between	Определяет, находится ли числовое значение в определенном диапазоне значений	Between 0 And 100
In	Определяет, является ли строковое значение элементом списка значений	In ("Минск", "Брест")
Like	Определяет, включает ли строковое значение указанные символы	Like "*лавр*"

1	2	3
<i>6. Логические операторы (результат – True or False)</i>		
Not Логическое отрицание	Возвращает значение <i>False</i> , если выражение имеет значение <i>True</i> , в обратном случае – <i>True</i>	Not A
And Логическое И	Конъюнкция Возвращает значение <i>True</i> , если оба выражения имеют значение <i>True</i>	A And B
Or Логическое ИЛИ	Дизъюнкция Возвращает значение <i>True</i> , если хотя бы одно из выражений имеет значение <i>True</i>	A Or B
Xor Исключающее ИЛИ	Возвращает значение <i>True</i> , если только одно из выражений имеет значение <i>True</i>	A Xor B
Eqv	Логическая эквиваленция	A Eqv B
Imp	Логическая импликация	A Imp B

Операторы сравнения в условиях отбора позволяют производить поиск данных более чем по одному значению, а для записи сложных условий применяются логические операторы (логические операторы эквиваленции и импликации практически не используются при записи условий отбора).

Если в условии отбора значение записано *без оператора*, то подразумевается оператор *равно* (=).

Для задания сложных условий отбора используются логические операторы – NOT, AND, OR, XOR (табл. 3.2).

Условия отбора, размещенные в разных полях, но на одной строке объединяются при помощи логического оператора И (AND), а если на разных строках – ИЛИ (OR) (рис. 3.7).

Результатом обработки выражения в строке УО является одно из значений *True* (Истина) или *False* (Ложь).

Таблица истинности логических операторов

Логический оператор	Результат выполнения логического оператора при заданных значениях А и В			
	А – Истина В – Истина	А – Истина В – Ложь	А – Ложь В – Истина	А – Ложь В – Ложь
Not A	Ложь	Ложь	Истина	Истина
Not B	Ложь	Истина	Ложь	Истина
A And B	Истина	Ложь	Ложь	Ложь
A Or B	Истина	Истина	Истина	Ложь
A Xor B	Ложь	Истина	Истина	Ложь

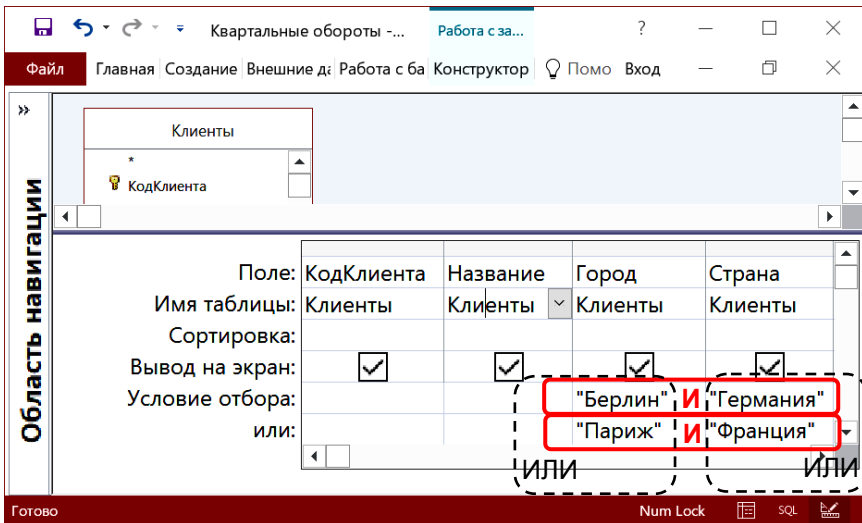


Рис. 3.7. Условия отбора по текстовым полям *Город* и *Страна* (подразумевается логическая операция И по двум столбцам в каждой строке и логическая операция ИЛИ по двум строкам каждого столбца)

Условие отбора для полей различных типов можно установить следующими способами:


1. Если поле имеет тип *числовой, денежный или счетчик*, то в строку УО Бланка запроса вводится оператор сравнения и число без дополнительных символов (*например, >=200* – так можно записать УО по полю *Цена*, для отбора товаров с ценой от 200 рублей).

2. Если поле *текстовое*, то в строку УО Бланка запроса вводится текст-образец в виде литерала или текст с символами шаблона * и ? (шаблон «*» заменяет любое число символов, в том числе и их отсутствие, а «?» – один символ). После ввода текста образца в строке отобразится *Like "текст"*, при этом *Like* и "" система добавляет автоматически (*например, Like "*"Минск"* – так может выглядеть УО для поля *Город*).

3. Для полей типа *дата/время* в строку УО Бланка запроса вводится оператор сравнения и дата – критерий отбора. *Например*, для отбора заказов, сформированных после 1 сентября 2023 года в качестве УО по полю *ДатаЗаказа* вводится выражение *>1.09.2023*, которое преобразуется к виду *>#01.09.2023#* (знаки «#» к распознанным значениям даты система добавляет автоматически).

4. В УО для полей с *логическим* типом данных используются константы *Да, Нет* (Истина, Ложь). *Например*, для поля *Поставки прекращены* из таблицы *Товары* необходимо ввести в УО значение *Нет*, чтобы отобразить все записи о товарах, которые еще поставляются.

5. Для поиска *пустых полей* в полях любого типа в качестве УО используется выражение *Is Null*.

Для формирования сложных выражений в условии отбора рекомендуется использовать *Мастер построения выражений*, который вызывается в строке УО для поля с помощью КЗМ/*Построить* или на вкладке *Конструктор запросов* в группе команд *Настройка запроса* нажатием кнопки «Построитель» 

Это приведет к выводу на экран диалогового окна «Построитель выражений» (рис. 3.8), которое в верхней части содержит *Область ввода выражения*, где и отображается формируемое *условие отбора* или *вычисляемое поле*.

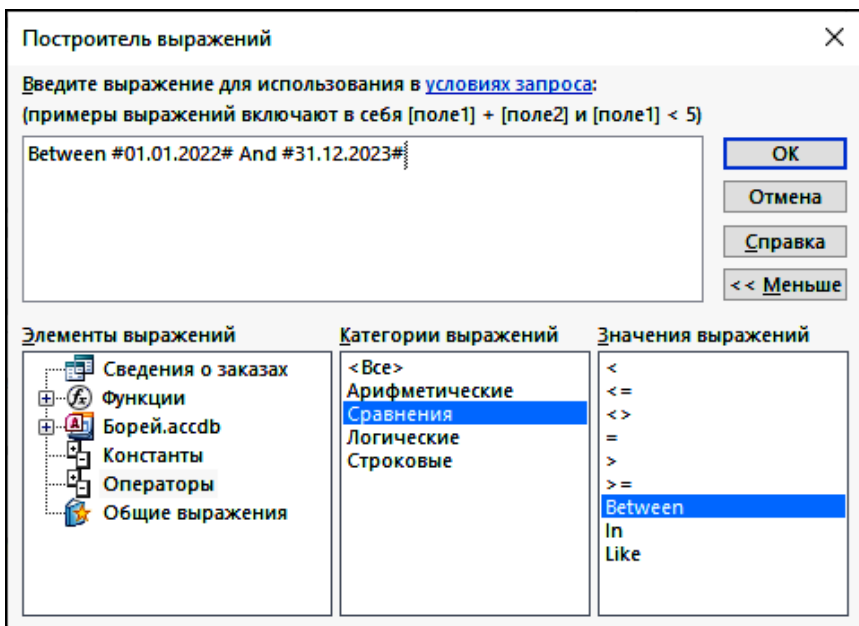


Рис. 3.8. Окно *Построителя выражений* содержит условие отбора с оператором `Between #01/01/2022# And #31/12/2023#` (для поля со значением типа *дата и время*)

Для формирования выражения используются доступные в нижней части окна «Построитель выражений» три области с элементами выражений:

1. В крайней левой области «Элементы выражений» выводится список объектов, включая: активный (текущий) объект; *Функции* (встроенные и определенные пользователем); *файл БД*, содержащий объекты базы данных (таблицы, запросы, формы); разделы *Константы*, *Операторы* и *Общие выражения*.

2. В средней области «Категории выражений» отображаются определенные элементы или типы элементов для объекта, выбранного в левой области «Элементы выражений». Именно в этой области отображаются категории стандартных функций, названия полей для выбранной таблицы БД, группы операторов и т. п.

Например, если выделить в левой области построителя раздел *Операторы*, то в области «Категории выражений» будет выведен

список групп операторов: *Арифметические, Сравнения, Логические, Строковые*.

3. В крайней правой области «Значения выражений» выводится список значений (если они существуют) для выделенных слева элементов.

Например, если выбрать в области «Элементы выражений» раздел *Операторы*, а в области «Категории выражений» – строку с типом операторов *Сравнения*, то в правой области будет выведен список всех встроенных операторов сравнения, в котором можно выделить необходимый и выполнить по нему двойной щелчок ЛКМ, чтобы вставить выбранный оператор в область ввода выражения.

Для завершения работы с окном «Построитель выражений» необходимо нажать кнопку «ОК».

Для проверки работоспособности сформированного условия отбора следует выполнить запрос. Если в результате запроса отображается пустая строка, следует убрать выражение из строки УО и повторить запрос, проверив корректность исходных данных в таблицах.

3.4. Вычисляемые поля в запросах

Запросы используются для выполнения расчетов. Для этого в строке Поле Бланка запроса создается новое (*вычисляемое*) поле, которое, включает три элемента: *имя поля*, «:» (двоеточие) и *выражение* (рис. 3.9).

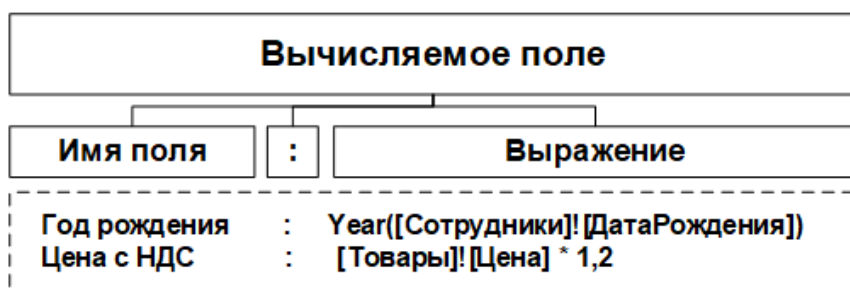


Рис. 3.9. Структура вычисляемого поля

Выражение представляет собой последовательность операндов, разделенных операторами (рис. 3.10).

Вычисление значения выражения выполняется слева направо, в порядке убывания приоритета операторов.

Для изменения порядка используются скобки ().



Рис. 3.10. Структура выражения для вычисляемого поля

Результат в вычисляемом поле *пересчитывается при каждом выполнении запроса* (на основе текущих значений операндов).

Имя вычисляемого поля должно быть уникальным и соответствовать правилам формирования имен (см. п. 2.3).

Если самостоятельно не задать имя, а сразу построить выражение для расчета значения в поле, то MS Access самостоятельно сформирует перед введенным выражением *стандартное имя* для вычисляемого поля – *Выражение1*, и отделит его от расчетной формулы с помощью : (двоеточия). В этом случае заголовком поля в таблице с результатами запроса будет имя «*Выражение1*».

В качестве *операндов* при записи выражения для расчета значения вычисляемого поля, так же как и при записи условий отбора, используются *константы, литералы, идентификаторы объектов и функции* (см. п. 3.3).

Операторы обозначают действия, выполняемые над операндами (см. табл. 3.1). Вычисления в выражении выполняются слева направо, в порядке убывания приоритета операторов. Для изменения порядка вычислений в выражении используются скобки ().

Например, необходимо рассчитать стоимость заказов, сформированных каждым сотрудником, с учетом цены товара, проданного количества и скидки (см. учебную БД *Борей*).

Для построения данного запроса необходимо:

1. В режиме *Конструктора* создать запрос выборки, выбрав в качестве источника взаимосвязанные таблицы *Сотрудники*, *Заказано* и *Заказы*.

2. Включить в *Бланк запроса* поле *Фамилия* из таблицы *Сотрудники*.

3. Для формирования вычисляемого поля с собственным именем выполнить следующие действия:

- в пустом поле *Бланка* запроса вызвать Построитель выражений (рис. 3.11),
- ввести *имя* вычисляемого поля (уникальное),
- ввести : (*двоеточие*),
- после : (*двоеточия*) построить выражение, выбирая необходимые объекты БД, значения, функции, операторы (рис. 3.12):

$$\text{Стоимость заказа: [Заказано]![Цена] * [Заказано]![Количество] * (1 - [Заказано]![Скидка])}$$

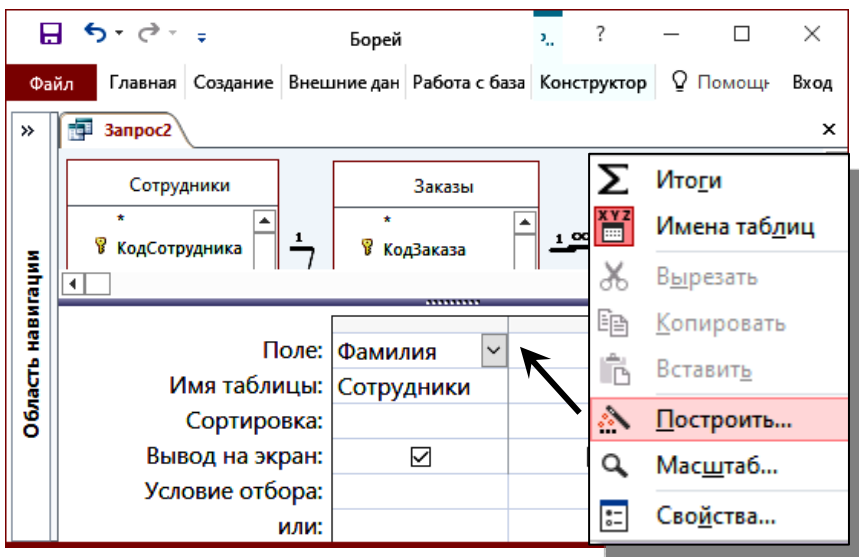


Рис. 3.11. Вызов *Построителя выражения* для создания вычисляемого поля

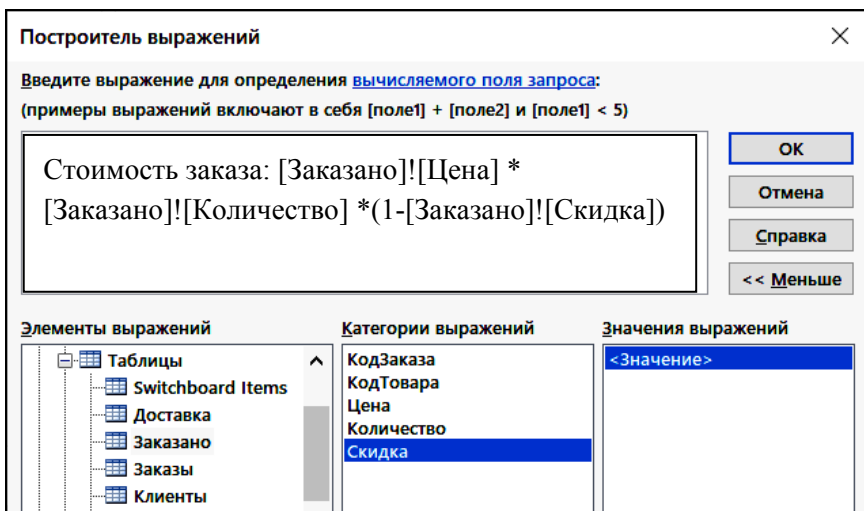


Рис. 3.12. Формирование выражения для вычисляемого поля

Во избежание ошибок имена объектов БД (например, поля таблицы) рекомендуется не вводить вручную, а выбирать из списка в *Построителе выражений*.

После закрытия окна *Построителя выражения* в поле *Бланка запроса* отобразится результат построения вычисляемого поля. Строка *Имя таблицы* для вычисляемого поля остается пустой (рис. 3.13). Результат вычислений приведен на рис. 3.14.

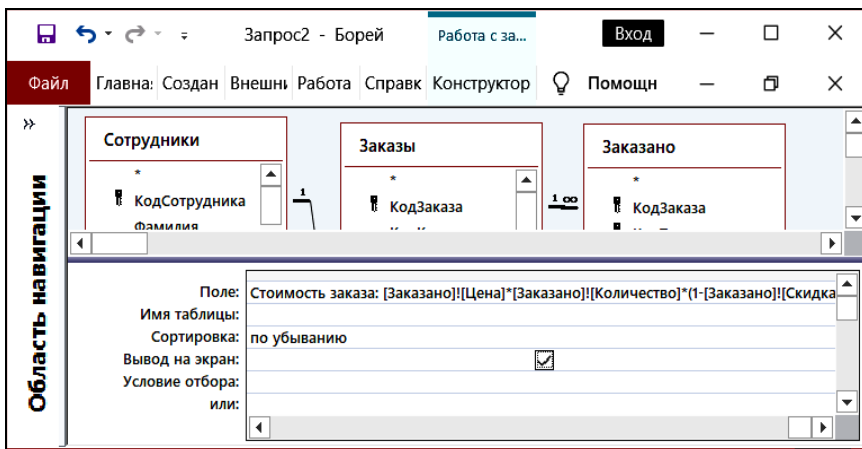


Рис. 3.13. Вычисляемое поле в *Бланке запроса*

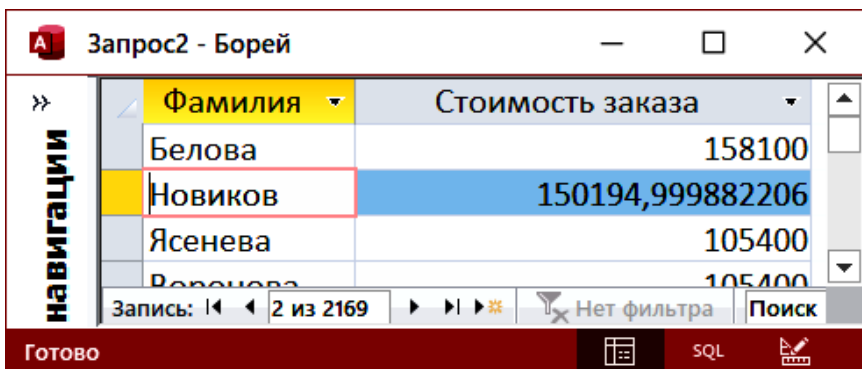


Рис. 3.14. Результат вычисления стоимости заказа (без изменения свойства *Формат* для вычисляемого поля)

При построении текстовых выражений используется текстовый оператор (конкатенация) – &. Фрагменты соединяемого текста заключаются в двойные кавычки "".

Например, для формирования вычисляемого поля *Контакт*, которое должно возвращать текстовую строку, содержащую фамилию, имя и должность сотрудника из таблицы *Сотрудники*, создается вычисляемое поле следующего вида (рис. 3.15):

Контакт: [Сотрудники]![Фамилия] & " " & [Сотрудники]![Имя] & ", "&[Сотрудники]![Должность]

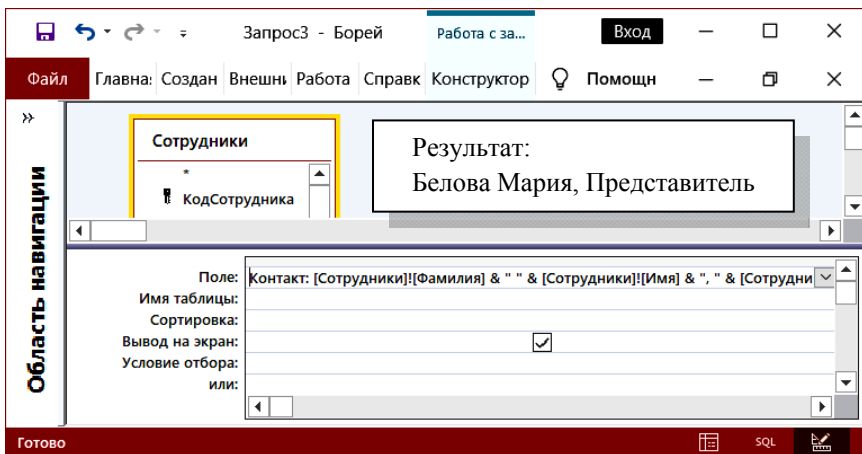


Рис. 3.15. Вычисляемое текстовое поле в *Бланке запроса* и пример возвращаемой строки результата

При создании вычисляемых полей и формировании условий отбора рекомендуется операнды и аргументы функций, представляющие собой имена объектов БД, выбирать ЛКМ из списка в области «Категории выражений» окна «Построитель выражений», а не набирать вручную. Такой подход позволяет сократить количество ошибок при записи выражений.

3.5. Применение функций MS Access

В выражениях MS Access могут использоваться функции из 14 категорий (табл. 3.3), при этом функции могут быть аргументами других функций (вложенные функции), что определяется решаемой задачей.

Таблица 3.3

Описание категорий функций MS Access

Категория	Назначение функций
1	2
База данных	С помощью функций для работы с объектами баз данных осуществляется обращение к объектам баз данных: таблицам, запросам, формам, отчетам, макросам и модулям
Дата/время	Используются для проведения операций со значениями даты и времени (табл. А.1)
Массивы	Используются для определения размерностей массивов при программировании на VBA
Математические	Используются для выполнения различных математических и тригонометрических вычислений
Обработка ошибок	Функции этой категории используются для отслеживания ошибок. Применение таких функций в выражении считается признаком хорошего тона, поскольку используемое выражение, корректно обрабатывает ошибочный результат и предупреждает об этом пользователя
Общий	Используются в основном при программировании на языке VBA для извлечения вспомогательной информации (код цвета и значение цвета)
По подмножеству	Статистические функции по подмножествам записей аналогичны статистическим функциям SQL, но работают с вычисляемыми значениями, а не со значениями, содержащимися в полях запросов
Преобразование	Применяются для преобразования выражений одного типа данных в другой (табл. А.2)

1	2
Проверка	Функции проверяют тип аргумента, тип результата вычисления выражения, отсутствие данных и т. п. (табл. А.3)
Сообщения	Позволяют выводить сообщения или вводить новые данные, а также устанавливать различные параметры ввода/вывода данных
Статистические	Статистические функции SQL обобщают данные в полях для выбранных записей, применяются при построении итоговых запросов, перекрестных запросов (табл. А.4)
Текстовые	Позволяют проводить различные операции над строками (табл. А.5)
Управление	Используются для выбора значения из нескольких альтернатив (табл. А.6)
Финансовые	Аналогичны финансовым функциям в MS Excel и используют те же аргументы

При работе с функциями рекомендуется выбирать функции и задавать аргументы не вручную, а с помощью *Построителя выражений*, в котором представлены 14 категорий встроенных функций различного назначения.

Функции возвращают в выражение вместо имени функции *значение*, тип которого (текст, число, значение даты/времени) зависит от категории и назначения функции (см. табл. 3.3).

Каждая функция выделяется круглыми скобками (), следующими сразу же за ее идентификатором. Наличие круглых скобок после имени функции синтаксически обязательно, даже если функция не имеет аргументов.

Например, в категории *Дата и время* (см. табл. А.1) ряд встроенных функций не имеют аргументов:

- функция Date () содержит пустой список аргументов и возвращает системную дату вида ДД.ММ.ГГГГ (01.09.2023);
- функция Now() не имеет аргументов и возвращает системную дату и время вида ДД.ММ.ГГГГ ЧЧ:ММ:СС (01.09.2023 15:12:41);
- функция Time () аналогично не имеет аргументов и возвращает системное время вида ЧЧ:ММ:СС (15:12:41).

Например, в категории *Базы данных* не имеет аргументов функция `CurrentUser()`, которая возвращает имя активного пользователя базы данных (рис. 3.16).

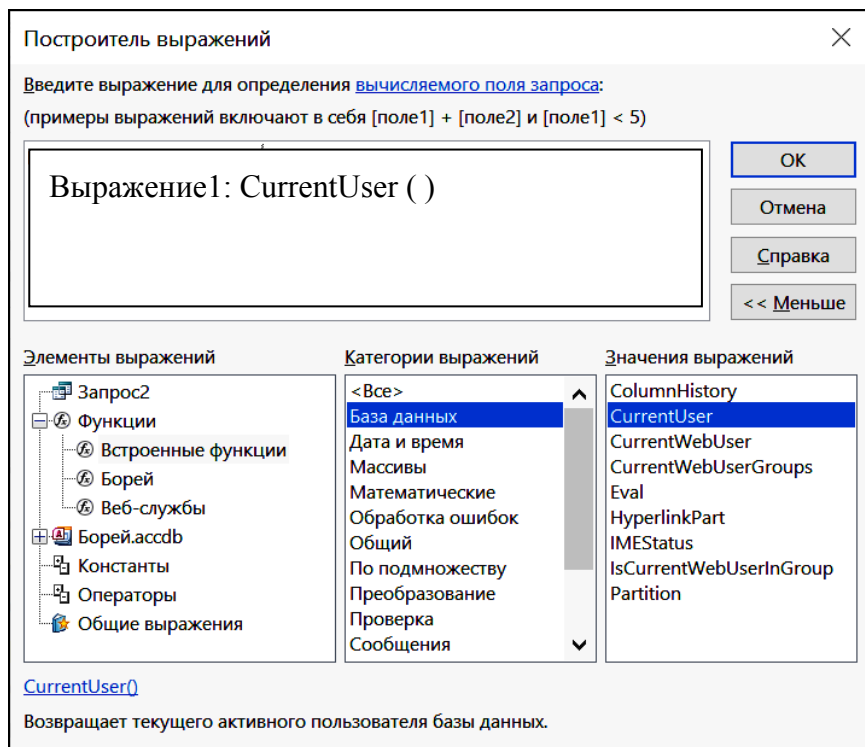


Рис. 3.16. Функция `CurrentUser()` из категории *База данных*

Многие функции требуют наличия *аргументов*, которые записываются в скобках при обращении к функции.

Аргументами могут быть: *константы*, литералы, *идентификаторы*, а также другие *функции*, которые в этом случае будут *вложенными функциями*.

Например, многие функции категории *Дата и время* (см. табл. А.1) имеют один обязательный аргумент «*Дата*» – любое числовое выражение или строковое выражение или любое сочетание, которое может представлять *дату*.

Если аргумент «Дата» содержит значение Null, то и функции возвращают значение Null.

Так, функция Year («Дата») возвращает *целое число*, соответствующее *году* указанной даты. Значениями аргумента «Дата» могут быть константы, значения полей таблиц типа дата/время или функция Date(), *например*:

Year (#01.09.2023#) – аргумент функции представляет собой литерал (значение даты 01.09.2023), результат работы функции – число соответствующее году = 2023;

Year ([Заказы]![Дата]) – аргумент функции – значение поля *Дата* из таблицы *Заказы*;

Year (Date()) – в качестве аргумента функции используется вложенная функция Date (), имеющая пустой список аргументов.

Полезны при обработке значений типа даты и времени и другие функции из категории *Дата и время*, которые позволяют решать различные задачи анализа данных, *например*, по определению срока выполнения заказа и своевременности его исполнения, вычислению возраста сотрудника по дате его рождения, стажа работы – по дате найма и т. д.

Например, с помощью функции Month() можно определить порядковый номер месяца, в котором был сделан заказ:

Месяц: Month ([Заказы]![ДатаРазмещения]).

Например, с помощью функции MonthName() можно определить название месяца, в котором был сделан заказ:

Месяц: MonthName (Month([Заказы]![ДатаРазмещения]); 0).

Например, с помощью функции DatePart () можно определить квартал, в котором был сделан заказ:

Квартал: DatePart ("q"; [Заказы]![ДатаРазмещения]).

Вместо аргумента "q" можно использовать значения "уууу", "m", "ww" и др., чтобы для исходной даты определить соответственно год, месяц, неделю и др.

С помощью функций If(), DatePart(), Date() и Year() можно определить возраст (по дате рождения) или стаж (по дате найма) сотрудника на текущую дату (число полных лет):

Число лет: If (DatePart("y"; «Дата»; 2) <= DatePart("y"; Date(); 2); Year(Date()) – Year(«Дата»); Year(Date()) – Year(«Дата») – 1).

Аргумент «Дата» в этой формуле может содержать значение поля *Дата рождения* (для расчета возраста) или *Дата найма* (для расчета стажа) сотрудника.

Аналогично можно рассчитать, сколько лет осталось до пенсии сотруднику, если известны пол сотрудника, дата его рождения и возраст выхода на пенсию.

Для определения интервала между датами используются функции DateAdd() и DateDiff().

Использование функций из категории *Преобразование* рассмотрим на примере запроса по расчету стоимости заказов сотрудников из п. 3.4: выражение вычисляемого поля было приведено на рис. 3.12, а результат работы запроса – на рис. 3.14. Очевидно, что результат представляет числовое значение без форматирования, в виде, неудобном для восприятия денежных сумм.

Для преобразования результата вычислений из числового значения к денежному используется функция CCur() (выражение с применением данной функции приведено на рис. 3.17, а результат выполнения запроса – на рис. 3.18).

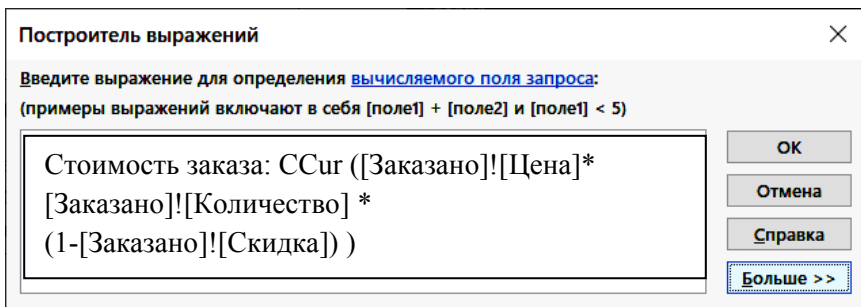


Рис. 3.17. Применение функции CCur() для преобразования значения выражения к типу *Денежный* (Currency)

Фамилия	Стоимость заказа
Белова	158 100,00 Р
Новиков	150 195,00 Р
Ясенева	105 400,00 Р

Рис. 3.18. Результат расчета стоимости заказа после применения функции *CCur()* (без изменения свойства *Формат* для вычисляемого поля)

Достаточно часто в запросах выполняется обработка *текстовых значений* с помощью стандартных функций из категории *Текстовые* (табл. А.5), доступ к которым в окне «Построитель выражений» обеспечивается следующей последовательностью действий: в области «Элементы выражений» ЛКМ раскрывается папка *Функции* и выбирается вариант *Встроенные функции*, далее в области «Категории выражений» с помощью ЛКМ выделяется категория функций *Текстовые* и в области «Значения выражений» двойной щелчок ЛКМ по требуемой функции вставляет ее непосредственно в область ввода выражения.

Обязательным аргументом любой текстовой функции является «строка» («string») – любое допустимое строковое выражение. Если строка содержит значение типа Null, то функции возвращают значение Null.

Например, текстовые функции *LCase* («строка») и *UCase* («строка») возвращают строковые значения, преобразованные в строчные и ПРОПИСНЫЕ буквы соответственно. При этом только прописные (строчные) буквы преобразуются в строчные (прописные), а все остальные буквы и небуквенные символы остаются без изменений.

Для извлечения части строки используются функции *Left* («строка»; «число_символов») и *Right* («строка»; «число_символов»), которые позволяют извлечь из строкового значения заданное число_символов слева (первые символы) и справа (последние символы) соответственно.

Например, чтобы создать в запросе поле, отображающее фамилию и инициалы клиента (в виде – Лавренова О.А.), необходимо в свободном поле *Бланка* запроса построить вычисляемое поле:

Фамилия_инициалы: [Клиенты]![Фамилия] & " " & Left([Клиенты]![Имя];1) & ". " & Left([Клиенты]![Отчество];1) & ". "

В данном случае применяется текстовая функция Left () , которая извлекает одну первую букву из значений в соответствующих полях: Left([Клиенты]![Имя] ; 1) и Left([Клиенты]![Отчество];1).

Например, в запросе создается вычисляемое поле, содержащее фамилию и инициалы, причем фамилия записывается прописными буквами (ЛАВРЕНОВА О.А.):

ФИО: UCase ([Клиенты]![Фамилия]) & " " & Left([Клиенты]![Имя];1) & ". " & Left([Клиенты]![Отчество];1) & ". "

В этом случае используется функция Ucase (), аргумент которой формируется с использованием функции Left () и оператора &.



Таким образом, применение встроенных функций из разных категорий позволяет решать разноплановые задачи анализа данных.

3.6. Использование групповых операций в итоговых запросах

В БД могут создаваться *итоговые запросы*, которые позволяют группировать данные и выполнять их статистическую обработку.

Итоговые запросы содержат в бланке запроса строку *Групповая операция*, которая используется для группировки записей и подведения итогов по группам записей с применением статистических функций.

Например, с помощью таких запросов подсчитывают число заказов или общую стоимость заказов для каждого сотрудника за конкретный период.

Для отображения в *Бланке запроса* строки *Групповая операция* необходимо на вкладке *Конструктор запроса* в группе команд *Показать или скрыть* нажать кнопку « Итоги» или в любом поле *бланка запроса* вызвать КЗМ и выбрать команду  Итоги.

В строке *Групповая операция* для каждого поля будет доступен список для выбора одного из вариантов: *Группировка*, список имен статистических функций (табл. 3.4), *Выражение* и *Условие*.

Таблица 3.4

Статистические функции для итоговых запросов

Функция	Действие
Sum	Суммирует значения для заданного поля в записях, отобранных запросом
Avg	Вычисляет среднее значение для заданного поля в записях, отобранных запросом
Min	Вычисляет минимальное значение для заданного поля в записях, отобранных запросом
Max	Вычисляет максимального значения определенного поля в записях, отобранных запросом
Count	Подсчитывает для заданного поля количество значений в записях, отобранных запросом
First	Определяет первое значение для заданного поля среди записей, отобранных запросом
Last	Определяет последнее значение для заданного поля среди записей, отобранных запросом
StDev	Вычисляет стандартное отклонение значений данного поля для всех записей, отобранных запросом
Var	Вычисляет вариации значений данного поля для всех записей, отобранных запросом

В итоговые запросы с групповыми операциями обычно включаются, как минимум, *два поля*, по первому из которых выполняется операция *Группировка*, а по второму применяется статистическая функция.

Данные в итоговых запросах обычно *сортируют* по полю со *статистической* функцией.

Также для поля со статистической функцией рекомендуется задать *свойство Подпись* (автоматически формируется подпись в виде текстовой строки, где первым словом записывается имя функции, а затем имя поля, к которому применяется функция. *Например*, имя поля *Count_КодЗаказа* получается, если по полю *КодЗаказа* из

таблицы *Заказы* была применена функция Count для подсчета числа заказов, сформированных каждым сотрудником. В таком случае целесообразно изменить подпись данного поля на более понятную, например, *Количество заказов* или *Всего заказов* или *Сформировано заказов*.

Рассмотрим применение итогового запроса для построения *рейтинга сотрудников компании по числу сформированных заказов* (БД Борей). Для определения рейтинга сотрудников необходимо построить таблицу, в которой в первом столбце отобразить фамилии сотрудников, а во втором – подсчитать число заказов, сформированных каждым сотрудником, и упорядочить по убыванию значения по данному полю (общее число заказов).

Для решения этой задачи необходимо поэтапно построить итоговый запрос:

1. Создать *запрос выборки*, добавив в подсхему данных взаимосвязанные таблицы *Сотрудники* и *Заказы*.

2. Включить в бланк запроса поля: *Фамилия* (из таблицы *Сотрудники*) и *Код заказа* (из таблицы *Заказы*).

3. Добавить в *Бланк запроса* строку *Групповая операция (Итоги)*.

4. В строке *Групповая операция* по полю *Фамилия* выбрать из списка вариант *Группировка*, а по полю *Код заказа* – статистическую функцию Count.

5. В поле *Код заказа* выполнить сортировку *по убыванию* и задать для свойства *Подпись поля* значение *Всего заказов*.

6. Сохранить запрос с именем *Рейтинг сотрудников по числу заказов*.

7. Выполнить запрос.

Макет итогового запроса для вывода рейтинга сотрудников по числу заказов приведен на рис. 3.19.

Результат выполнения итогового запроса с изменением свойства *Подпись поля* для поля со статистической функцией приведен на рис. 3.20.

Результат выполнения итогового запроса без изменения свойства *Подпись поля* для поля со статистической функцией приведен на рис. 3.21 (поле со статистической функцией получило имя CountКодЗаказа).

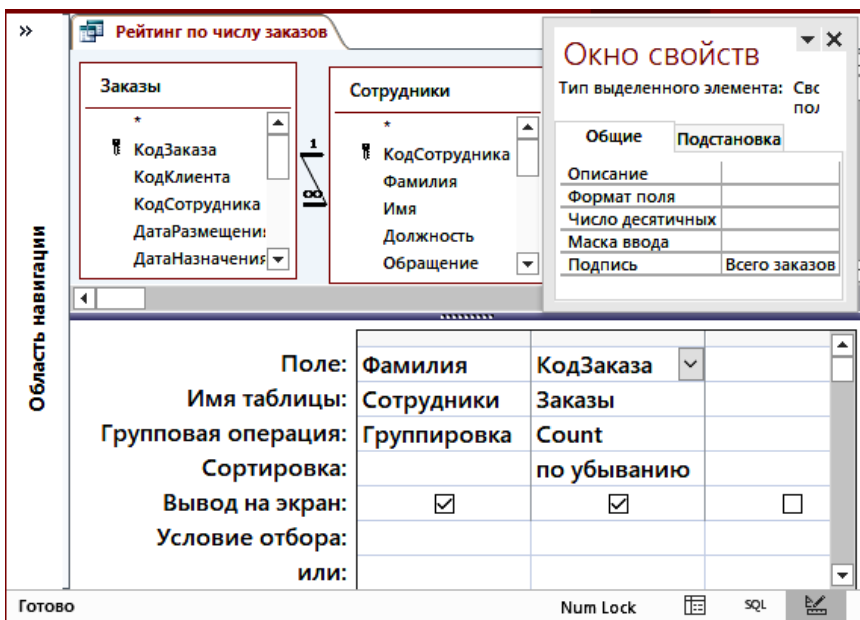


Рис. 3.19. Макет итогового запроса для определения рейтинга сотрудников по числу заказов (с групповой операцией Count по полю КодЗаказа)

Рейтинг по числу заказов

Фамилия	Всего заказов
Воронова	156
Бабкина	126
Белова	122
Крылова	105
Новиков	96
Кралев	73
Акбаев	67
Ясенева	43
Кротов	42

Запись: 1 из 9 Нет фильтра

Рис. 3.20. Результат итогового запроса с изменением подписи поля

Фамилия	Count-КодЗаказа
Воронова	156
Бабкина	126
Белова	122
Крылова	105
Новиков	96
Кралев	73
Акбаев	67
Ясенева	43
Кротов	42

Запись: 1 из 9 Нет фильтра

Рис. 3.21. Результат запроса определения рейтинга сотрудников по числу заказов (без изменения для поля *КодЗаказа* свойства *Подпись поля*)

Расширить возможности анализа итогов работы сотрудников можно с помощью дополнительного вычисляемого поля, подсчитывающего сумму сформированного заказа с учетом скидки, к которому применяется статистическая функция Sum (рис. 3.22).

Поле:	Фамилия	КодЗаказа	Сумма заказа: [Зака]
Имя таблицы:	Сотрудники	Заказы	
Групповая операция:	Группировка	Count	Sum
Сортировка:			по убыванию
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			
или:			

Рис. 3.22. Макет итогового запроса для определения рейтинга сотрудников по числу заказов и по стоимости заказов

Вычисляемое поле в запросе для расчета суммы каждого заказа имеет вид:

$$\text{Сумма заказа: } SS_{\text{Cur}}([\text{Заказано}]![\text{Цена}] * [\text{Заказано}]![\text{Количество}] * (1 - [\text{Заказано}]![\text{Скидка}])).$$

Таким образом, запрос формирует рейтинг сотрудников по объемам продаж в стоимостном выражении (рис. 3.23).

Фамилия	Число заказов	Сумма заказа
Воронова	422	2 320 520,71 Р
Бабкина	318	2 012 632,43 Р
Белова	343	1 912 942,39 Р
Новиков	241	1 665 377,55 Р
Крылова	268	1 329 776,77 Р
Кралев	181	1 254 751,10 Р
Ясенева	109	915 894,41 Р
Акбаев	168	739 131,29 Р
Кротов	119	690 712,52 Р

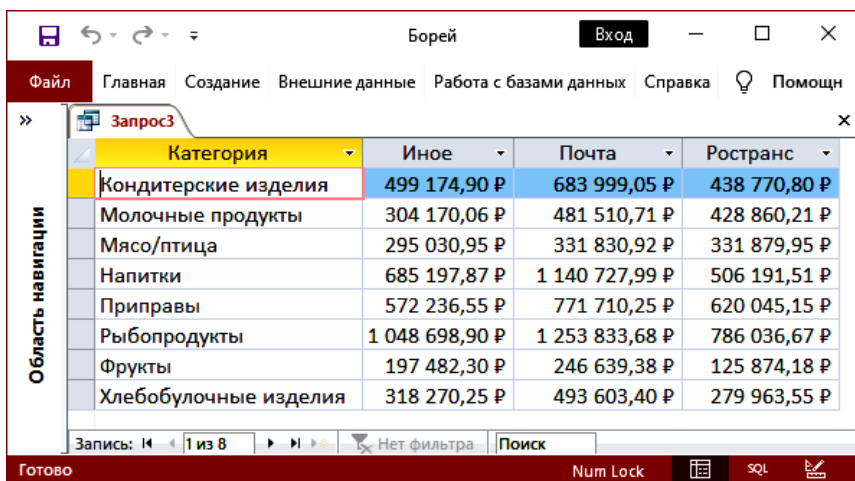
Рис. 3.23. Результат запроса для определения рейтинга сотрудников по общей стоимости заказов

В таблице видно, что сотрудник Ясенева при меньшем количестве сформированных заказов опережает в рейтинге по объемам продаж в стоимостном выражении сотрудников Акбаева и Кротова.

3.7. Создание перекрестных запросов

Перекрестный запрос является разновидностью итоговых запросов и позволяет компактно отобразить данные, используя объединение однотипной информации (в формате строк-столбцов, на пересечении которых вычисляется итоговое значение).

Перекрестные запросы по структуре подобны сводным таблицам MS Excel, но без области фильтра отчета. *Например*, перекрестный запрос представляет собой сводную ведомость по стоимости товаров каждой категории, доставленных потребителям разными способами доставки (рис. 3.24), причем в БД *Борей* исходный массив данных о заказах включает более 2000 записей.



Категория	Иное	Почта	Ространс
Кондитерские изделия	499 174,90 ₽	683 999,05 ₽	438 770,80 ₽
Молочные продукты	304 170,06 ₽	481 510,71 ₽	428 860,21 ₽
Мясо/птица	295 030,95 ₽	331 830,92 ₽	331 879,95 ₽
Напитки	685 197,87 ₽	1 140 727,99 ₽	506 191,51 ₽
Приправы	572 236,55 ₽	771 710,25 ₽	620 045,15 ₽
Рыбопродукты	1 048 698,90 ₽	1 253 833,68 ₽	786 036,67 ₽
Фрукты	197 482,30 ₽	246 639,38 ₽	125 874,18 ₽
Хлебобулочные изделия	318 270,25 ₽	493 603,40 ₽	279 963,55 ₽

Рис. 3.24. Перекрестный запрос «Сводная ведомость по общей стоимости заказов по категориям товаров, доставленных каждым видом доставки»

Перекрестные запросы обладают рядом достоинств:

- возможностью обработки значительного объема данных и вывода их в формате, который удобен для дальнейшего построения графиков и диаграмм;
- простотой и скоростью разработки сложных запросов с несколькими уровнями детализации.

К недостаткам перекрестных запросов можно отнести невозможность сортировки таблицы результатов по значениям, содержащимся

в столбцах, поскольку в большинстве случаев одновременное упорядочивание данных в столбцах по всем строкам невыполнимо. Перекрестные запросы удобны для представления данных в виде таблицы, с целью дальнейшей визуализации на диаграммах, но на их основе довольно сложно построить отчеты.

Перекрестные запросы можно создать двумя способами:

- 1) в режиме *Мастера* перекрестного запроса;
- 2) непосредственно в режиме *Конструктора*.

1. Создание в режиме *Мастера перекрестного запроса* требует предварительной подготовки: необходимо создать дополнительный запрос выборки, содержащий поля из всех таблиц, используемых для перекрестного запроса. Далее в пошаговом режиме выполнить создание в режиме *Мастера перекрестного запроса*:

- выбрать в качестве источника запрос;
- задать поля – заголовки строк и заголовки столбцов;
- выбрать значения, которые отображаются на пересечении, и итоговую функцию. Дополнительно можно вычислить итоги по строкам.

Например, необходимо создать перекрестный запрос, позволяющий подсчитать объем продаж (в натуральном выражении, ед.) товаров каждой марки, проданных каждым сотрудником компании (БД *Борей*). Порядок решения задачи с применением *Мастера перекрестного запроса* включает следующие действия:

1. Предварительно необходимо создать и сохранить запрос выборки *Исходные данные*, включающий все необходимые поля из связанных таблиц *Сотрудники*, *Заказы*, *Заказано*, *Товары*.

2. Далее выполнить команду *Мастер запросов* в группе *Запросы* на вкладке *Создание* и в диалоговом окне «Новый запрос» выбрать вариант *Перекрестный запрос*.

3. На первом шаге построения перекрестного запроса выбрать в качестве источника предварительно созданный запрос *Исходные данные*.

4. На следующих 2-х шагах сформировать макет перекрестного запроса: в качестве *заголовков строк* выбрать поле *Марка*, в качестве *заголовков столбцов* – поле *Фамилия*.

5. На следующем шаге для области перекрестного запроса *Значения* выбрать поле *Количество* и функцию *Сумма* для поля *Количество* (*определить объем продаж в натуральном выражении*),

установить флажок *Вычислить итоговое значение для каждой строки*. Диалоговое окно «Создание перекрестного запроса» со сформированным макетом перекрестного запроса приведено на рис. 3.25.

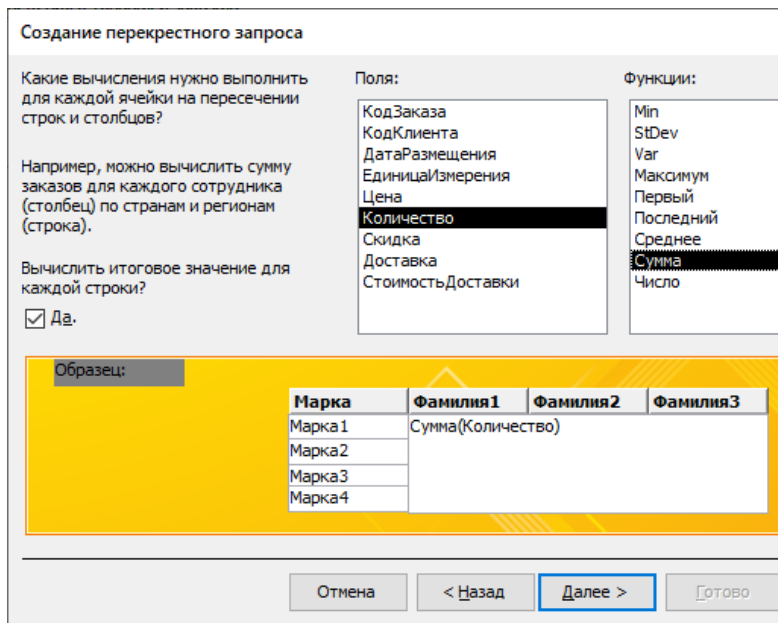


Рис. 3.25. Готовый макет перекрестного запроса в режиме *Мастера*

6. Перейти к завершающему шагу, задать имя запроса (по умолчанию – *Исходные данные перекрестный*), подтвердить выполнение запроса. По умолчанию открывается таблица с результатом выполнения запроса (рис. 3.26). При необходимости изменить макет перекрестного запроса можно непосредственно в режиме *Конструктора* (рис. 3.27).

Для общего итога по полю *Марка* создано дополнительное поле вида: *Итоговое значение Количество: Количество*, к которому применена групповая операция SUM и которое используется как заголовки строк. Особенность перекрестного запроса в том, что итоги рассчитываются *только для заголовков строк*. Это позволяет выбрать поле для заголовков строк, для которых требуется подведение итогов, а следовательно, определить и структуру перекрестного запроса в целом.

Файл Главная Создание Внешние данные Работа с базами данных Справка Борей Выход

Исходные данные_перекрестный

Итоговое з

Марка	Итоговое з	Акбаев	Бабкина	Белова	Воронова	Кралев	Крс
Alice Mutton	329	20	80	68	26	31	
Aniseed Syrup	241		64	1	60	25	
Boston Crab Meat	982	118	192	84	121	195	
Camembert Pierrot	608	50	80	5	156	75	
Carnarvon Tigers	458	6	20	6	225	69	
Chai	740	80	123	51	245	59	
Chang	747	20	208	57	261	67	
Chartreuse verte	1105	53	393	90	298	66	
Chef Anton's Cajun Seasoning	184		15		52	27	
Chef Anton's Gumbo Mix	799		105	149	135	197	
Chocolade	520	12	40	148	95	42	
Cote de Blaye	793	151	109	127	120	3	
Escargots de Bourgogne	1496	146	109	104	255	206	

Область навигации

Запись: 1 из 77 Нет фильтра Поиск

Режим таблицы Num Lock SQL

Рис. 3.26. Результат выполнения перекрестного запроса, созданного в режиме Мастера перекрестных запросов

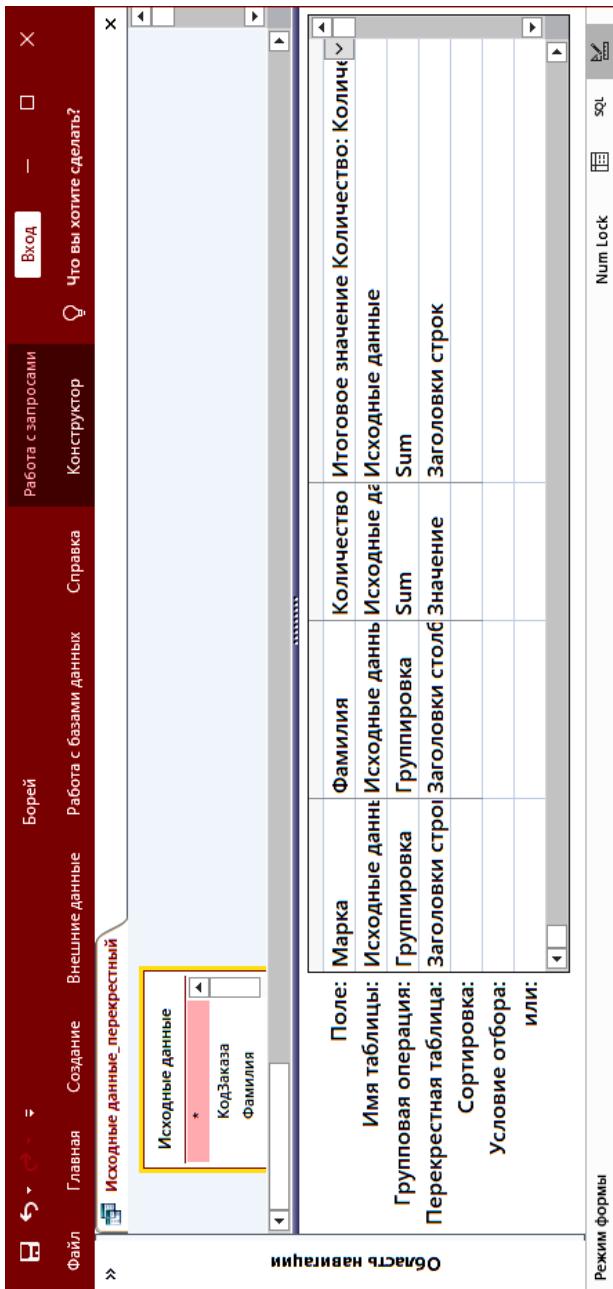


Рис. 3.27. Макет сформированного перекрестного запроса в режиме *Конструктора* (с дополнительным полем *Итоговое значение* *Количество*: *Количество*)

По умолчанию запросу, построенному в режиме *Мастера перекрестных запросов*, будет присвоено имя *Исходные данные_перекрестный* (к имени запроса – источника данных, добавляется название типа запроса – *перекрестный*).

Создание аналогичного перекрестного запроса в режиме *Конструктора* не требует предварительного создания запроса-источника данных и выполняется в следующем порядке:

1. Создать запрос выборки и добавить в подсхему данных все таблицы, содержащие необходимые для построения перекрестного запроса поля (*например*, таблицы из БД Борей: *Товары, Заказано, Заказы, Сотрудники*). В окне подсхемы данных следует проконтролировать наличие связей между таблицами, в случае отсутствия связей следует найти таблицу-связку для несвязанных таблиц и также добавить ее в подсхему данных.

2. Далее на вкладке *Конструктор запроса* в группе команд *Тип запроса* выбрать тип запроса – *Перекрестный*. При этом в бланк запроса автоматически будут добавлены дополнительные строки: *Групповая операция* и *Перекрестная таблица*.

3. Далее в бланк запроса следует добавить 3 поля: *Марка, Фамилия, Количество*. При этом в строке *Групповая операция* для каждого поля автоматически отображается вариант *Группировка*.

Для полей *Марка, Фамилия* это значение остается неизменным, поскольку данные будут сгруппированы по этим полям в формате строк-столбцов, а по полю *Количество* следует выбрать статистическую функцию *Sum* для подведения итогов.

4. Следующее действие заключается в формировании структуры сводного отчета путем определении полей для заголовков строк и столбцов: в строке *Перекрестная таблица* для поля *Марка* следует выбрать вариант *Заголовки строк*, для поля *Фамилия* – *Заголовки столбцов*, а для поля *Количество* – *Значение*. По полю *Марка* можно задать сортировку.

Макет перекрестного запроса, сформированного непосредственно в режиме *Конструктора* представлен на рис. 3.28, а результат на рис. 3.29.

Борей

Вход

Что вы хотите сделать?

Справка

Работа с базами данных

Внешние данные

Создание

Главная

Файл

Область навигации

Перекрестный в конструкторе

Марка	Анбаев	Бабкина	Белова	Воронова	Кралев	Кротов	Кры
Alice Mutton	20	80	68	26	31		
Aniseed Syrup		64	1	60	25		
Boston Crab Meat	118	192	84	121	195	42	
Camembert Pierrot	50	80	5	156	75	4	
Carnarvon Tigers	6	20	6	225	69	12	
Chai	80	123	51	245	59	15	
Chang	20	208	57	261	67	31	
Chartreuse verte	53	393	90	298	66	64	
Chef Anton's Cajun Seasoning		15		52	27		
Chef Anton's Gumbo Mix		105	149	135	197	40	
Chocolate	12	40	148	95	42		
Cote de Blaye	151	109	127	120	3	108	
Escargots de Bourgogne	146	109	104	255	206	144	

Записи: 1 из 77

Нет фильтра

Поиск

Num Lock

Режим таблицы

Рис. 3.29. Результат выполнения перекрестного запроса в режиме Таблицы

3.8. Применение параметров в запросах

Запросы с параметрами не являются самостоятельным типом запросов. Они расширяют возможности запросов выборки при использовании условий отбора и выполнении вычислений.

Необходимость запросов с параметрами обусловлена тем, что запросы выборки не всегда выполняются пользователем при неизменных критериях отбора. Чаще всего запросы представляют собой незначительно видоизмененные варианты базового запроса, условия которых изменяются от случая к случаю, но несущественно. Для их реализации предусмотрены *параметры*, значения которых задаются пользователем в диалоговом окне при выполнении запроса. Таким образом, под *параметром* в запросах будем понимать не конкретное значение, а некоторый прототип значения.

Параметры используются:

- в условиях отбора;
- в вычисляемых полях.

Для проектирования параметрического запроса в строке УО или вычисляемом поле используется, как было сказано выше, не конкретное числовое или текстовое значение, а *название параметра*, заключенное в квадратные скобки []. В этом случае, при выполнении запроса конкретное значение параметра будет запрошено у пользователя в диалоговом окне «Введите значение параметра».

Название параметра в запросе должно быть *уникальным* и не должно совпадать с именами полей таблицы.

Например, необходимо вывести из таблиц БД *Борей* список товаров из указанной категории с ценой не выше определенного значения и рассчитать их цену в валюте по заданному курсу. Для этой цели создается *запрос выборки*, содержащий поля *Марка*, *Категория*, *Цена* из базовых таблиц и вычисляемое поле *цена в валюте*. Параметры применяются в Условии отбора и в вычисляемом поле:

1. В условии отбора по полю *Категория* из таблицы *Типы* можно использовать параметр [Введите название категории].

Однако можно выполнить отбор значений *по текстовому фрагменту*, а не точному значению, если использовать универсальное условие отбора в виде выражения с параметром:

Like "*" & [Введите фрагмент категории] & "*"

2. В строке *Условие отбора* по полю *Цена* из таблицы *Товары* записывается логическое выражение с параметром:

\leq [Цена не выше значения (введите максимальную цену)]

3. В вычисляемом поле *Цена в валюте* для пересчета значения цены используется параметр [Введите курс \$] при записи выражения для расчета:

Цена в валюте: [Товары]![Цена]/ [Введите курс \$]

4. Для вычисляемого поля необходимо изменить свойство *Формат поля* и задать пользовательский формат (рис. 3.30).

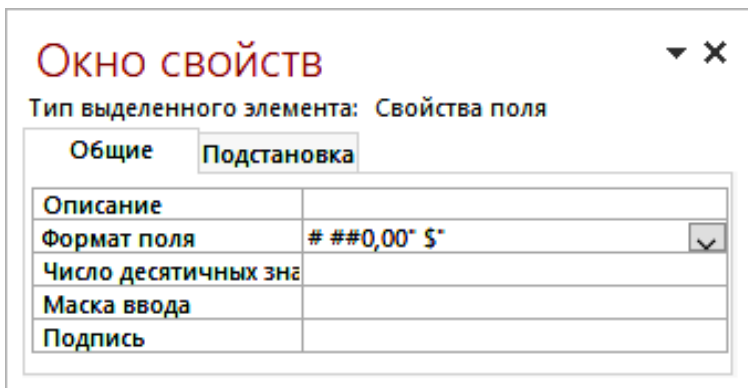


Рис. 3.30. Свойство Формат для вычисляемого поля *Цена в валюте*

Макет построенного запроса с двумя параметрами в условии отбора и одним параметром в вычисляемом поле приведен на рис. 3.31, окна параметров при выполнении запроса – на рис. 3.32.

Если в запросе используется несколько параметров, то порядок их появления определяется порядком расположения полей в *Бланке запроса*.

3.9. Назначение и особенности запросов действия. Методика их формирования

С помощью четырех видов запросов действия пользователь может *создавать* новые таблицы, *обновлять*, *добавлять* или *удалять* группы записей в базовых таблицах:

1. *Запрос на создание новой таблицы* применяется для *архивирования* старых записей БД или *сохранения резервных копий* таблиц. В этом случае из динамического набора (результата запроса выборки) формируется новая таблица – постоянный объект БД, сохраняемый на вкладке БД *Таблицы*.

Сохранить новую таблицу можно как в текущей БД, так и в новой БД (в момент выполнения запроса на создание таблицы *новая БД должна быть закрыта*).

Чтобы посмотреть результат, необходимо перейти в окне БД на вкладку *Таблицы* и открыть созданную таблицу.

2. *Запрос на обновление записей* применяется для изменения значений в полях группы записей, отобранных по заданному условию.

При создании запроса в строку *Обновление* бланка запроса вводятся новые значения для изменяемых полей. В качестве новых значений могут выступать как константы, так и вычисляемые выражения.

3. *Запрос на добавление записей* используется для добавления всех или отобранных по условию записей одной таблицы к другой. Может применяться даже в тех случаях, когда макеты таблиц имеют некоторые различия по количеству и составу полей.

При построении запроса после выбора типа запроса – *Запрос на добавление*, в диалоговом окне запрашивается имя и местоположение таблицы, в которую необходимо добавить отобранные записи, а в бланке запроса появится строка *Добавление*, в которую автоматически вводятся имена полей запроса, совпадающие с именами таблицы.

Можно исключить добавляемые поля из запроса, в этом случае в целевой таблице они окажутся пустыми.

4. *Запрос на удаление записей* используется для удаления отобранных по условию записей из таблицы.

В строке *Удаление* бланка запроса автоматически появляется значение «Условие», в которую можно ввести конкретное *значение*

или *параметр*, на основании которых выполняется удаление записей из таблицы.

Для удаления записей, содержащих незаполненные поля, используется условие отбора Is Null.

Отличительные особенности запросов на изменение заключаются в следующем:

1. Эти запросы *не создают динамического набора данных*, а вносят изменения в существующие таблицы БД или создают новые таблицы (объекты) БД на вкладке *Таблица*.

2. Невозможно отменить результат выполнения запросов на изменение.

3. Для корректной работы запросов действий в БД для всех межтабличных связей в схеме БД должен быть установлен параметр *Обеспечение целостности данных* и обеспечивающие эту целостность флажки *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей* (окно вызывается из контекстного меню для выделенной межтабличной связи с помощью команды *Изменить связь*, рис. 3.33).

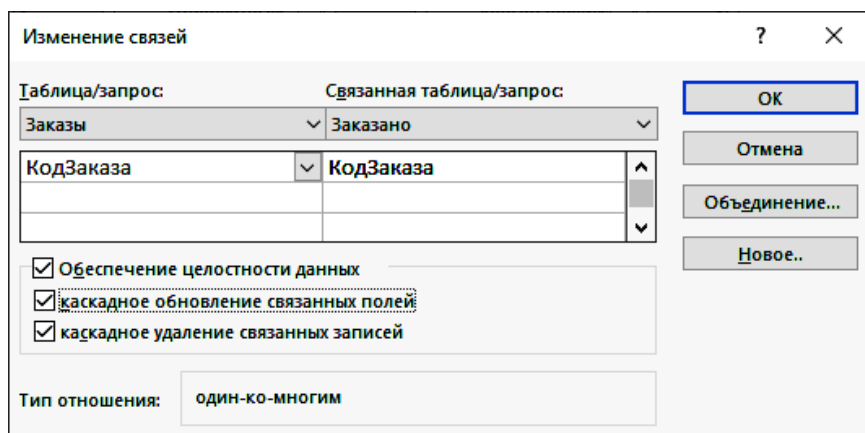


Рис. 3.33. Параметры обеспечения целостности данных

Целостность данных означает систему правил Microsoft Access для поддержания установленных межтабличных связей при вводе, редактировании и удалении записей. Целостность данных обеспе-

чивает корректную обработку данных, защиту от случайного удаления или изменения связанных данных. Установить параметры целостности данных для межтабличных связей можно только в том случае, если выполнены следующие условия:

- связанное поле главной таблицы является *ключевым полем* (или имеет уникальный индекс);

- связанные поля имеют *одинаковый тип данных*. Возможно *исключение*: поле с типом данных *Счетчик* может быть связано с числовым полем, если свойство *Размер поля* для обоих полей имеет значение *Длинное целое* или *Код репликации*;

- обе таблицы принадлежат одной базе данных MS Access. Для связанных таблиц из баз данных других форматов установить целостность данных невозможно.

Если установлен только флажок *Обеспечение целостности данных*, то накладываются следующие *ограничения*:

1. *Невозможно ввести в поле внешнего ключа связанной таблицы значение, не содержащееся в ключевом поле главной таблицы.* Однако в поле внешнего ключа возможен ввод пустых значений, показывающих, что записи не являются связанными.

Например, нельзя сохранить запись, регистрирующую заказ для несуществующего клиента, но можно создать запись для заказа, который пока не отнесен ни к одному из клиентов, если ввести пустое значение в поле *КодКлиента*.

2. *Не допускается удаление записи из главной таблицы, если существуют связанные с ней записи в подчиненной таблице.*

Например, невозможно удалить запись из таблицы *Сотрудники*, если в таблице *Заказы* имеются заказы, относящиеся к данному сотруднику.

3. *Невозможно изменить значение ключевого поля в главной таблице, если существуют записи, связанные с данной.*

Например, невозможно изменить *КодСотрудника* в таблице *Сотрудники*, если в таблице *Заказы* имеются заказы, относящиеся к этому сотруднику.

Если установлен *только флажок Обеспечение целостности данных*, то любая попытка выполнить действие, нарушающее одно из перечисленных выше правил, приведет к выводу на экран предупреждения, а *само действие выполнено не будет*.

Чтобы преодолеть ограничения на удаление или изменение связанных записей, сохраняя при этом целостность данных, следует использовать средства поддержки целостности данных – установить флажки: *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*.

Если установлен флажок *Каскадное обновление связанных полей*, то при изменении ключевого поля главной таблицы автоматически изменяются и соответствующие значения связанных записей. MS Access выполняет каскадное обновление без вывода предупреждающих сообщений.

Если установлен флажок *Каскадное удаление связанных записей*, то при удалении записи в главной таблице удаляются и все связанные записи в подчиненной таблице. Если записи удаляются вручную из таблицы или формы при установленном флажке *Каскадное удаление связанных записей*, MS Access выводит *предупреждение* о возможности удаления связанных записей.

Если записи из таблицы удаляются с помощью *запроса на удаление записей*, то удаление выполняется автоматически, *без вывода предупреждения*.

Формирование запросов действия в режиме в режиме *Конструктора* выполняется в следующем порядке:

1. Сформировать запрос выборки, выбрав необходимые поля и задав условия отбора.

2. Выполнить запрос выборки, проверив корректность заданных условий (вкладка *Конструктор*, группа команд *Результаты*, кнопка «Выполнить», отобразить динамический набор данных).

3. Вернуться в режим *Конструктора* (кнопка «Конструктор» на вкладке *Главная* в группе команд *Режимы*).

4. Выбрать требуемый *вид запроса действия* в группе команд *Тип запроса* (рис. 3.34) на вкладке *Конструктор* из набора вкладок *Работа с запросами*:

- Создание таблицы;
- Обновление;
- Добавление;
- Удаление.

Для запроса на создание таблицы необходимо дополнительно указать имя создаваемой таблицы (рис. 3.35) и имя внешней БД, если таблица создается не в текущей БД.

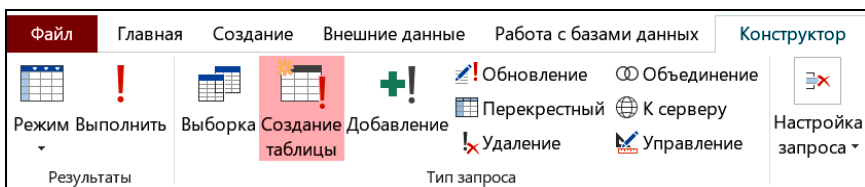


Рис. 3.34. Выбор типа запроса действия на вкладке *Конструктор*

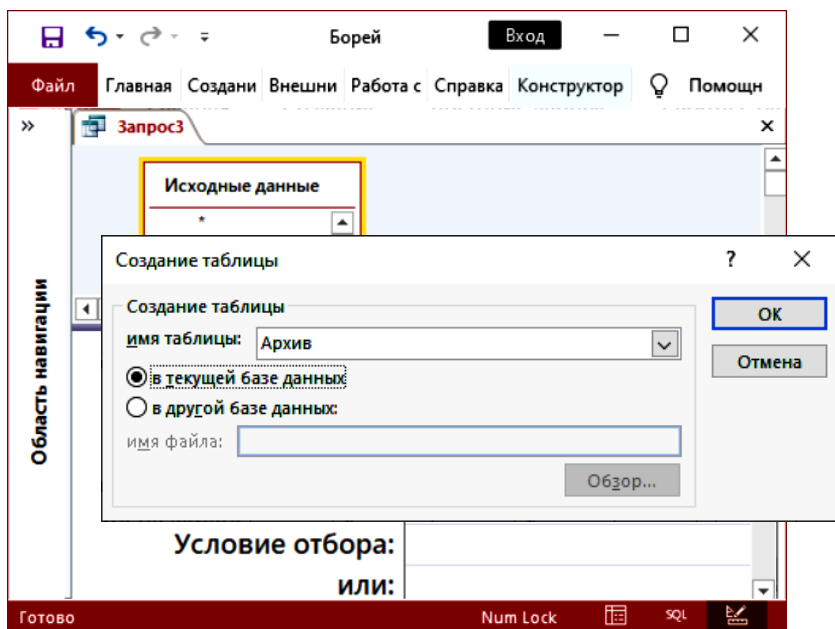


Рис. 3.35. Макет запроса на создание таблицы (в текущей БД)

5. Выполнить запрос действия, нажав кнопку «Выполнить» (на вкладке *Конструктор* в группе команд *Результаты*).

6. При выполнении запроса действия на экран выводится диалоговое окно (или последовательность диалоговых окон) с информацией для пользователя о выполняемых действиях, а также с запросами на подтверждение выполняемых действий.

7. После выполнения запроса действия для контроля изменения данных в БД, необходимо открыть таблицы на вкладке *Таблицы* или создать запросы выбора.

Например, необходимо увеличить цену некоторого товара на определенный процент. Для этого создается запрос на обновление, макет которого приведен на рис. 3.36.

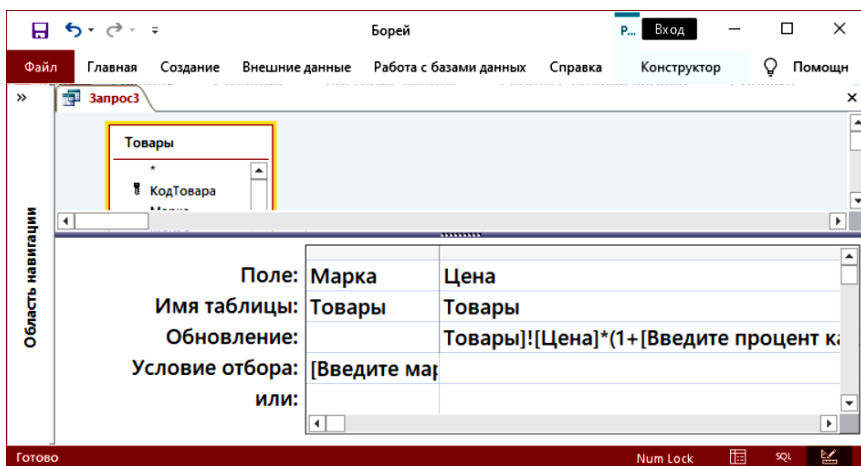


Рис. 3.36. Макет запроса на обновление (выражение для пересчета цены: $[Товары]![Цена]*(1+[Введите процент как целое число 0-100]/100)$)

При выполнении запроса будут запрошены значения 2-х параметров (рис. 3.37):

- 1) [Введите марку] – из условия отбора по полю *Марка*;
- 2) [Введите процент как целое число] – из выражения в строке *Обновление* по полю *Цена*.

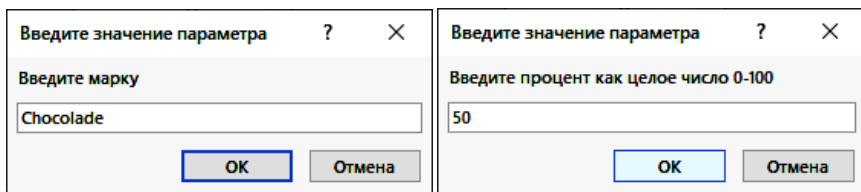


Рис. 3.37. Ввод параметров для запроса на обновление цены товара (марки *Chocolade* на 50 %)

Для удаления из таблицы записей о сотруднике по его фамилии можно построить запрос на удаление с использованием параметра в условии отбора для поля *Фамилия* (рис. 3.38).

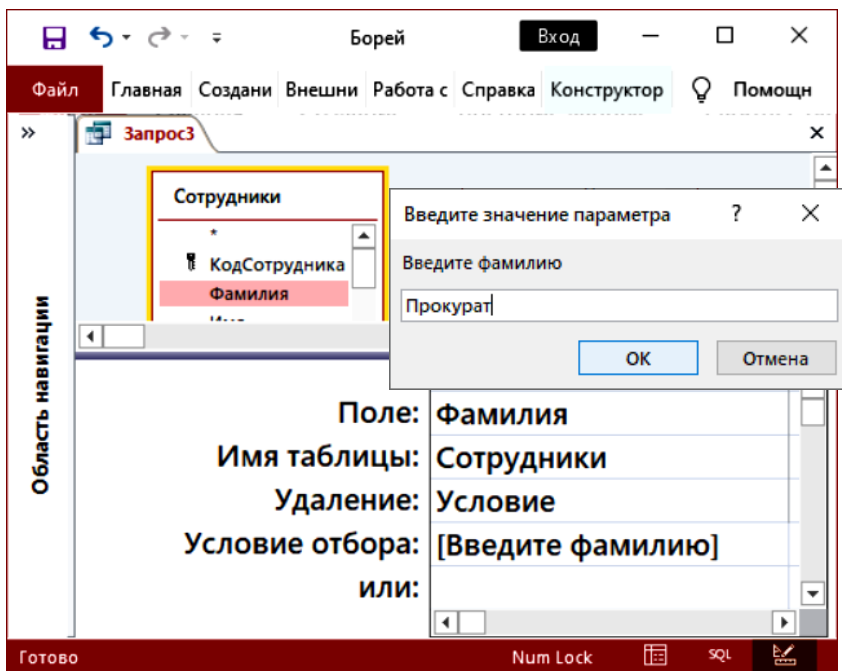


Рис. 3.38. Макет запроса на удаление и окно ввода параметра

После ввода значения в окно параметра система выводит сообщение о количестве удаляемых записей (рис. 3.39).

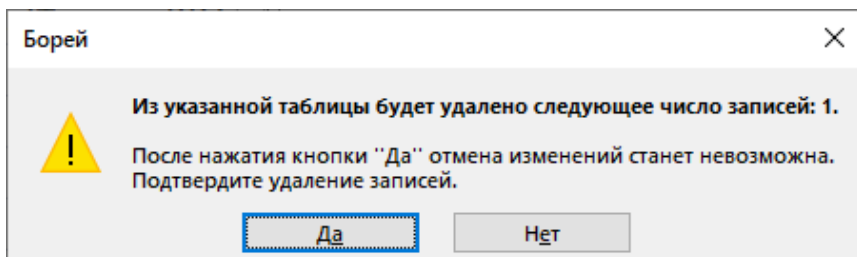


Рис. 3.39. Окно на подтверждение выполнения запроса на удаление

Вопросы для самопроверки

1. Какие инструменты СУБД применяются для анализа данных из таблиц БД?

2. Сформулируйте понятие «запроса», опишите назначение запросов. Назовите признаки классификации запросов, их виды и особенности.

3. Какие объекты могут быть источниками для формирования запросов СУБД MS Access? Что представляют собой результаты работы запросов выборки и запросов действия?

4. Назовите способы создания запросов СУБД MS Access. Какие запросы создаются в режиме Мастера?

5. Назовите особенности и порядок создания запроса выборки в режиме Конструктора.

6. Что представляет собой выражение для условия отбора в запросе? Приведите примеры формирования условий отбора для текстовых и числовых полей. Как записывается условие отбора для поиска пустых значений?

7. Сформулируйте понятие выражения MS Access, опишите его элементы. В каких строках бланка запроса окна Конструктора могут использоваться выражения? Приведите примеры.

8. Какие виды операндов используются в выражениях? Приведите примеры.

9. Назовите виды операторов, используемых в выражениях. Какие из них имеют самый высокий приоритет?

10. Как формируется вычисляемое поле в запросе MS Access? Какое имя присваивается ему по умолчанию? Приведите примеры вычисляемых полей.

11. Опишите возможности Построителя выражений. Какие преимущества дает его использование при записи выражений?

12. Каковы правила записи функций в вычисляемых полях? Какие категории стандартных функций применяются в MS Access?

13. Какие функции применяются для обработки текстовых значений? Приведите примеры.

14. Какие функции позволяют выполнять вычисления в зависимости от некоторых условий? Приведите примеры их использования.

15. Какие функции применяются для обработки значений типа даты/время? Приведите примеры функций, с помощью которых можно извлечь часть даты (день, месяц, квартал, год).

16. В чем заключаются особенности запросов с параметрами и в каких случаях их целесообразно использовать? Приведите примеры.

17. Что понимают под итоговыми запросами? Каковы особенности их построения?

18. Перечислите основные статистические функции, которые применяются при построении итоговых запросов. Приведите примеры запросов с применением статистических функций.

19. Для чего применяются перекрестные запросы? Приведите примеры.

20. Назовите способы построения перекрестных запросов и опишите их особенности. Какова структура перекрестного запроса? По какому полю подводятся итоги в перекрестном запросе?

21. Что понимают под «параметрами» в запросах? Приведите примеры их применения в условиях отбора и в вычисляемых полях.

22. Для решения каких задач используются запросы действия? Какие условия должны быть соблюдены для их корректной работы?

23. Что такое «целостность данных» и как она обеспечивается в СУБД MS Access?

Тема 4. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС БД

- 4.1. Назначение и виды форм, способы их создания.
- 4.2. Работа с формами в окне Конструктора.
- 4.3. Работа с элементами управления.
- 4.4. Особенности создания и настройка кнопочных форм.
- 4.5. Работа с макросами.
- 4.6. Понятие и назначение отчетов. Особенности их создания и настройка.

4.1. Назначение и виды форм, способы их создания

Форма – объект СУБД MS Access, предназначенный для наглядного отображения, ввода и редактирования данных из таблиц (табличная форма, рис. 4.1), а также для управления объектами базы данных (кнопочная форма, рис. 4.2).

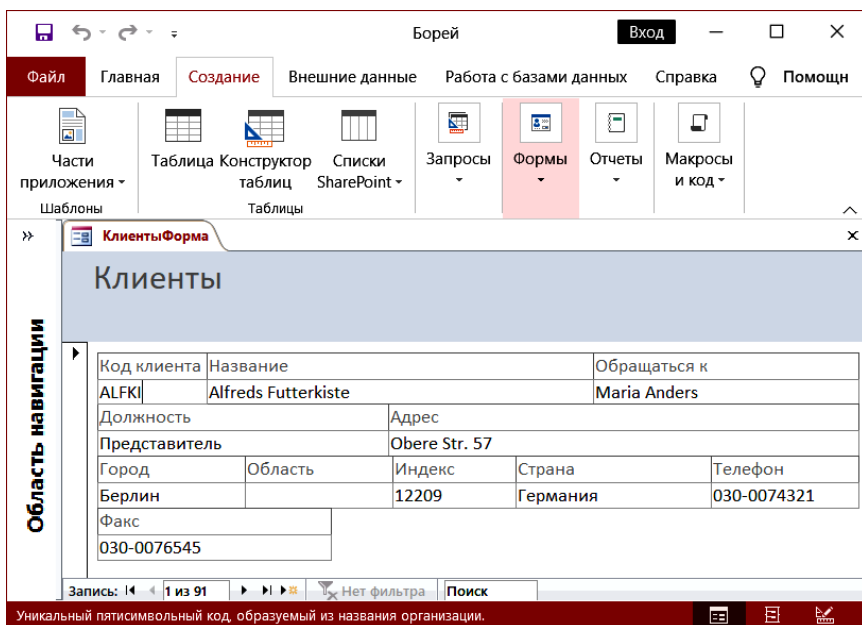


Рис. 4.1. Пример табличной формы *Клиенты*

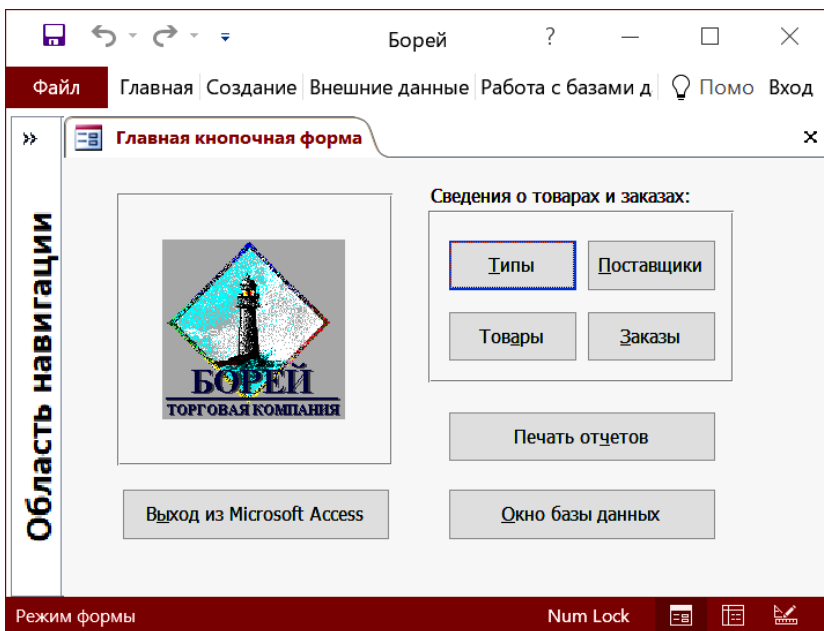


Рис. 4.2. Пример кнопочной формы

Формы, как правило, соответствуют формам первичных документов – источников данных для загрузки БД. При этом обеспечивается принципиальная особенность работы с базой данных – однократный ввод данных.

Формы бывают двух видов *табличные* и *кнопочные* (рис. 4.3).

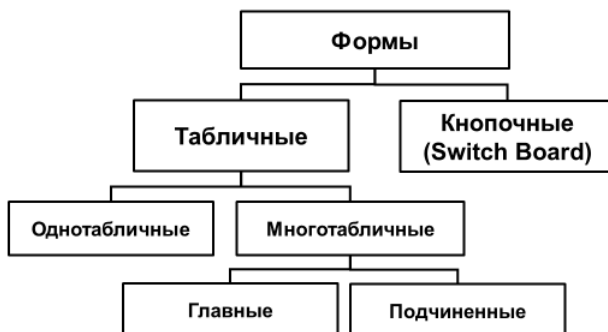


Рис. 4.3. Виды форм

Табличная форма позволяет изменить вид отображения данных. В табличной форме можно просматривать, редактировать и удалять записи. Табличные формы могут быть *однотабличные* и *многотабличные* (подчиненные и связанные).

Примеры табличных форм приведены на рис. 4.4–4.6.

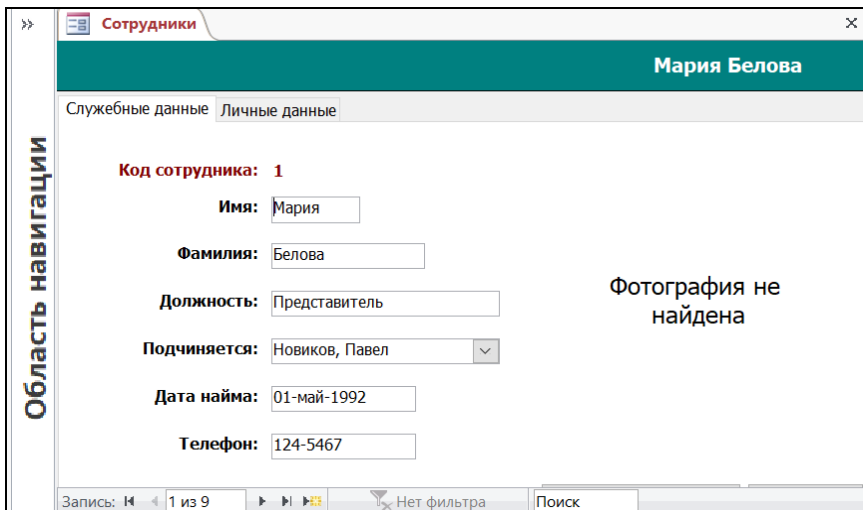


Рис. 4.4. Пример однотабличной формы с двумя вкладками (активна вкладка *Службные данные*)

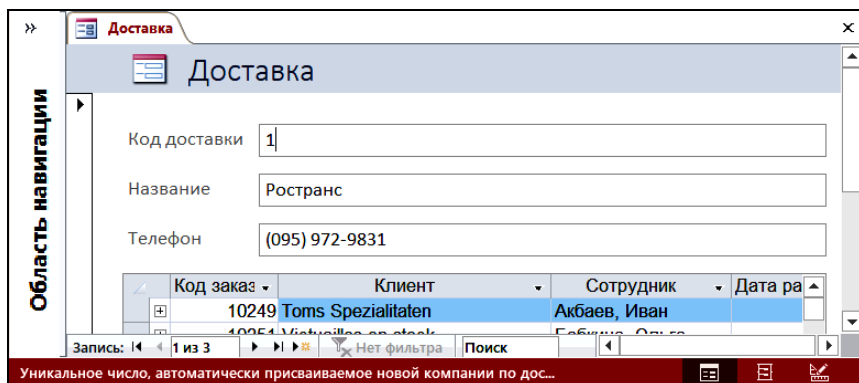


Рис. 4.5. Пример табличной формы *Доставка* с подчиненной формой *Заказы*

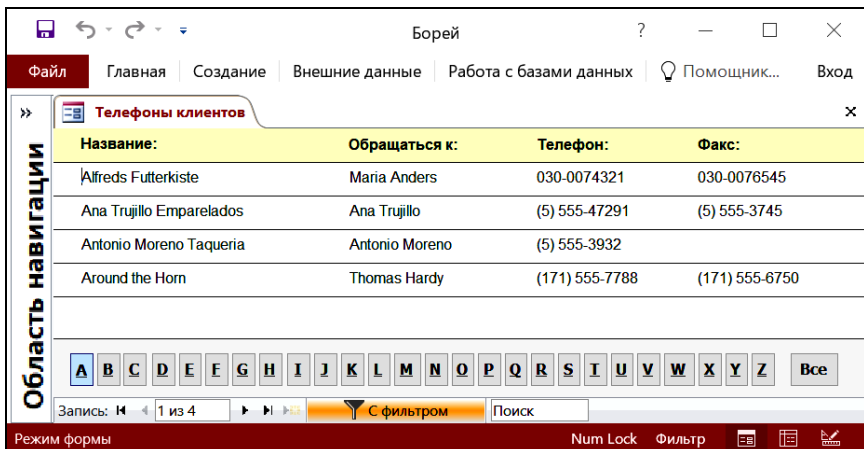


Рис. 4.6. Пример однотабличной формы с кнопками для фильтрации записей

Форму можно создать на основе любой таблицы или запроса. Порядок следования полей в форме не обязательно совпадает с порядком полей в таблице БД.

Для создания формы, использующей данные из многих таблиц, в качестве основы используется *запрос* (например, исходные данные), включающий необходимые данные из связанных таблиц.

Для создания формы используются команды в группе команд *Формы* на вкладке *Создание* (рис. 4.7).

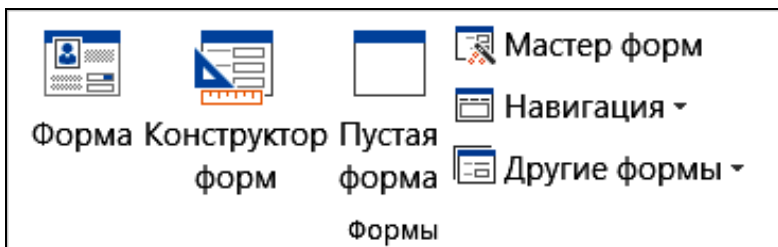


Рис. 4.7. Режимы создания форм (группа команд *Формы* вкладки *Создание*)

Для создания формы доступны следующие режимы:

1. *Форма* – предварительно на вкладке *Таблицы* или *Запросы* должен быть выбран объект, для которого будет создана автоформа.

2. *Конструктор форм* – создается пустая форма, не привязанная к источнику данных, которая открывается в режиме *Конструктора*. Этот режим используется в случае самостоятельного дизайна формы с использованием элементов управления.

3. *Пустая форма* – создается пустая форма, не привязанная к источнику данных, которая отображается в режиме макета.

4. *Мастер форм* – в этом режиме есть возможность построить как однотабличную, так и многотабличную форму, указав несколько источников для формы (взаимосвязанных таблиц или запросов).

В режиме *Мастера* также можно выбрать вариант расположения полей:

– *в один столбец* (поля одной записи располагаются в виде столбца из двух элементов, включая подпись и значение);

– *ленточный* (все записи расположены построчно, но подписи полей находятся только в заголовке над значениями, как таблица, причем ширина столбцов позволяет увидеть полное название полей);

– *табличный* (как таблица, все записи в строках);

– *выровненный* (компактное расположение полей).

5. *Навигация* – можно выбрать варианты расположения кнопок для формы с навигацией в виде горизонтальных или вертикальных вкладок, а также их комбинации.

6. *Другие формы* – в этом режиме можно создать форму для одновременного отображения не одной, а нескольких записей (вариант – несколько элементов), а также создать подчиненную форму или модальное окно.

Для освоения процесса создания форм, как элемента пользовательского интерфейса, создания формы «в один клик» разработчику (не специалисту в области дизайна), рекомендуется использовать режим *Мастера форм*, который предоставляет набор готовых базовых макетов и их элементов, а также дает возможность комбинировать их в соответствии с решаемой задачей.

Шаги *Мастера форм*:

1. Выбор *источника формы* (запроса или таблицы БД).

2. Выбор *полей*, отображаемых в форме (из указанного источника или других доступных объектов БД, из нескольких таблиц, рис. 4.8).

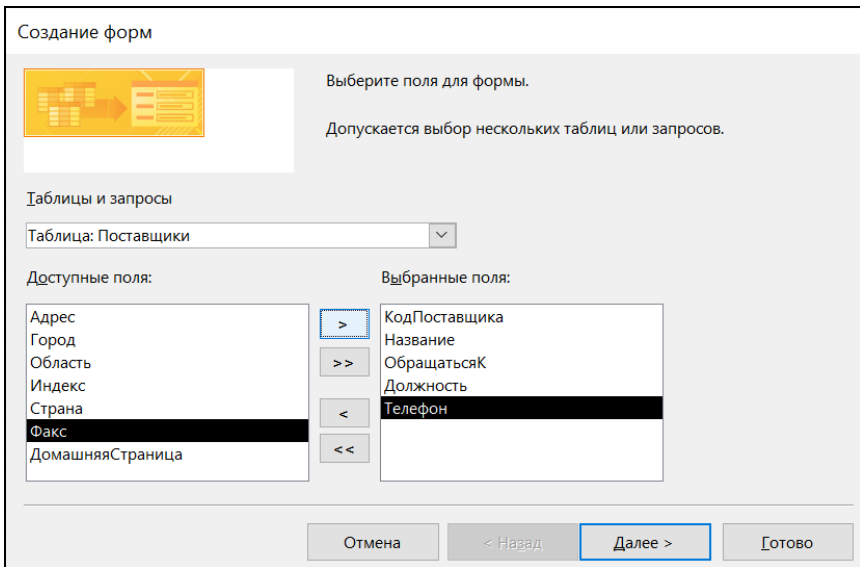


Рис. 4.8. Создание однотоабличной формы в режиме *Мастера*

3. Выбор *внешнего вида формы* – порядка расположения полей: в один столбец, ленточный, табличный, выровненный (рис. 4.9).

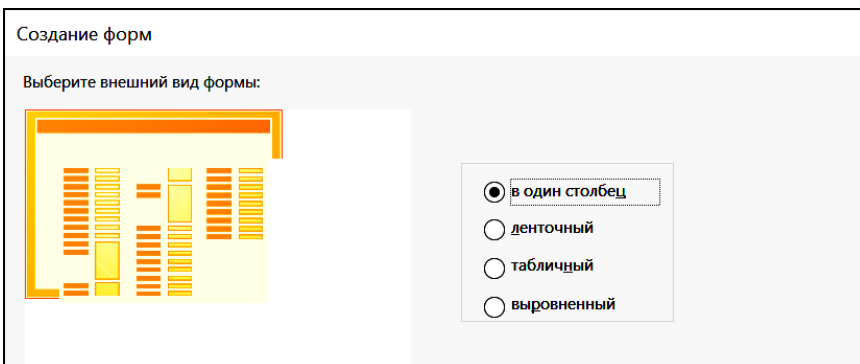


Рис. 4.9. Создание формы в режиме *Мастера* (выбор внешнего вида)

4. Ввод *Имени формы* (по умолчанию совпадает с именем источника, рис. 4.10).

Создание форм

Задайте имя формы:

ПоставщикиФорма

Указаны все сведения, необходимые для создания формы с помощью мастера.

Дальнейшие действия:

открыть форму для просмотра и ввода данных.

изменить макет формы.

Отмена < Назад Далее > Готово

Рис. 4.10. Завершение создания формы в режиме *Мастера*

После завершения работы *Мастера* создается форма, в которой размещены заданные поля из таблицы-источника.

Если включена подчиненная форма, то для нее выделена область, в которой указано только присвоенное ей имя.

По умолчанию созданная форма открывается для просмотра в режиме *Формы*, что позволяет выполнить ввод данных (рис. 4.11).

Если на последнем шаге *Мастера* (см. рис. 4.10) выбрать переключатель *Изменить макет формы*, то форма открывается в режиме *Конструктора*, в котором выполняется редактирование макета формы: настройка размеров и свойств областей формы, элементов управления и т. д.

Переключение между режимами отображения формы выполняется на вкладке *Главная* (или *Конструктор* из набора вкладок *Инструменты конструктора форм*) в группе команд *Режимы: Форма, Таблица или Конструктор*.

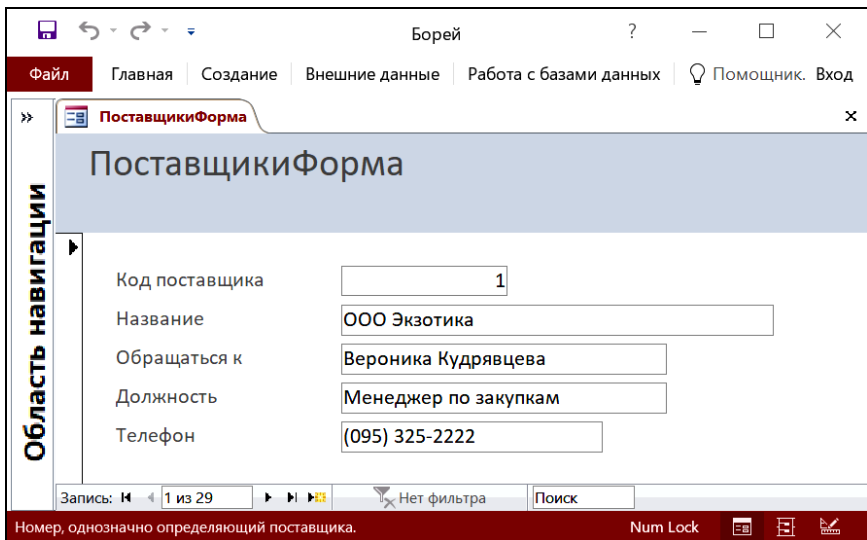


Рис. 4.11. Результат создания формы в режиме *Мастера*

При работе с формой в режиме *Формы* используются:

1. Кнопки перехода между записями в окне формы.
2. Кнопки на вкладке *Главная* для работы с записями:
 - создать новую запись (группа команд *Записи / Создать*);
 - удалить запись (группа команд *Записи / Удалить*);
 - найти запись по ее содержимому (группа команд *Найти*);
 - фильтры (удобны для анализа данных в табличной форме, группа команд *Сортировка и фильтр / Фильтр*);
 - сортировка (группа команд *Сортировка и фильтр / Сортировка*);
 - сохранить (группа команд *Записи/Сохранить запись*).

Дополнительно в режиме формы можно выполнить проверку орфографии в группе команд *Записи* на вкладке *Главная*.

4.2. Работа с формами в окне Конструктора

Объект *Форма* в режиме *Конструктора* (рис. 4.12) имеет следующую стандартную структуру (так же как и отчет):

- *область данных* – *обязательная область*, в которой размещаются элементы управления для оформления и вывода данных;
- *верхний/нижний колонтитул* – *необязательная область*, содержит элементы управления, текст или данные, которые выводятся при печати формы/отчета на каждой странице. В режиме *Формы* колонтитулы не отображаются;
- *заголовок/примечание* – *необязательная область*, содержит элементы управления, текст или данные, которые выводятся при печати формы/отчета на первой/последней странице.

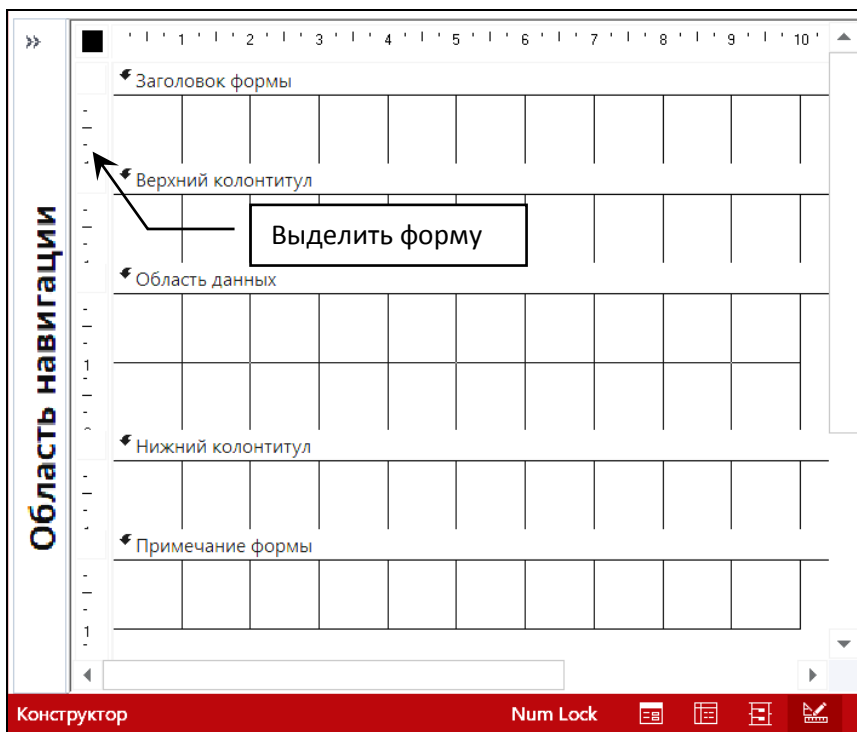


Рис. 4.12. Структура формы в режиме *Конструктора*

Для отображения или сокрытия указанных разделов формы (кроме области данных) используется контекстное меню (рис. 4.13).

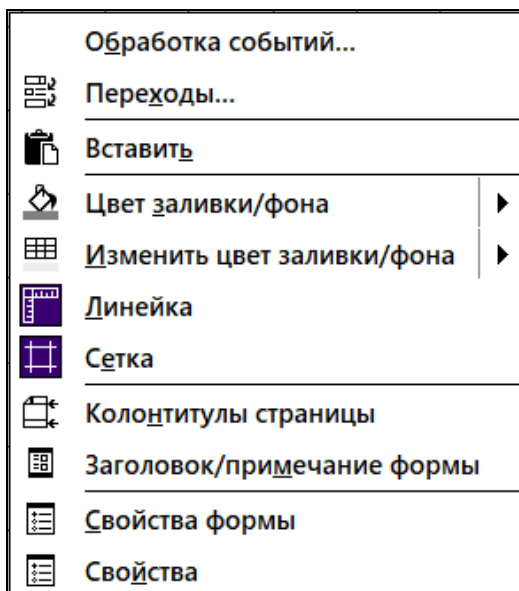


Рис. 4.13. Команды контекстного меню для работы с разделами формы в режиме *Конструктора*

При работе с формой в режиме *Конструктора* рекомендуется также отобразить элементы интерфейса *Линейка* и *Сетка*, затем использовать их для точного позиционирования элементов управления в разделах формы.

Команда КЗМ *Свойства* отображает область свойств для выделенного в форме элемента.

Работа с формой в режиме *Конструктора* связана с группами команд на вкладке *Конструктор* из набора вкладок *Инструменты конструктора форм* (рис. 4.14):

- *Режимы* – для переключения между режимами Конструктора, Макета и Формы;
- *Темы* – для изменения темы оформления, цветов и шрифтов;
- *Элементы управления* – для добавления элементов управления для вывода данных или оформления формы;
- *Колонтитулы* – для добавления логотипа, заголовка формы и вывода в разделе заголовков текущей даты;

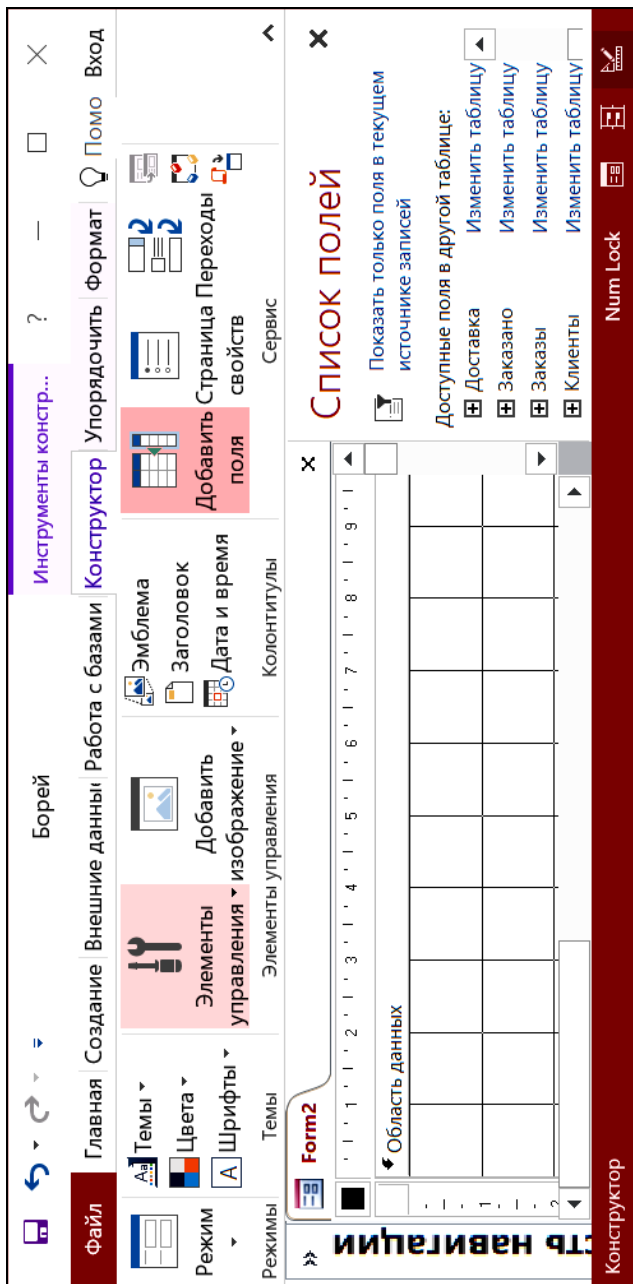


Рис. 4.14. Команды вкладки *Конструктор* из набора вкладок *Инструменты конструктора форм*

– *Сервис* – содержит ряд команд для работы с макетом формы в режиме *Конструктора*:

– *Добавить поля* – позволяет отобразить или скрыть список полей таблицы или запроса, для которых проектируется форма;

– *Страница свойства* – позволяет отобразить список свойств для выделенного объекта (формы, раздела, элемента управления);

– *Переходы* – позволяет установить порядок перехода между элементами управления, разделами формы во время работы с формой в режиме *Формы* при нажатии клавиш «Tab» или «Shift + Tab».

В начале работы над формой рекомендуется выбрать область данных и задать *размер формы*. Изменение размера формы выполняется либо в окне *Свойства формы* на вкладке *Макет*, либо с использованием ЛКМ на границе любой области формы, которую необходимо изменить (курсор мыши в этом режиме имеет вид двунаправленной стрелки).

Выделить форму можно с использованием кнопки «Выделить» (см. рис. 4.12) в виде черного квадрата на пересечении линеек, либо использовать команду в группе *Выделение* на вкладке *Формат* (рис. 4.15) из набора вкладок *Инструменты конструктора форм*.

Аналогично можно выделить любую область формы. Заголовок выделенной области в структуре формы визуальнo затемняется.

При работе в режиме *Конструктора* с элементами управления (см. п. 4.3) для их выделения и форматирования также применяются команды на вкладке *Формат* (см. рис. 4.15), а также команды на вкладке *Упорядочить* (рис. 4.16) из набора вкладок *Инструменты конструктора форм*.

Для перемещения между областями формы и элементами управления в режиме *Конструктора* можно использовать клавишу «Tab» и сочетание клавиш «Shift + Tab».

Для настройки формы необходимо открыть окно *Свойства формы* с помощью команды *Свойства формы* из КЗМ для области формы.

Все параметры сгруппированы на вкладках *Макет* (рис. 4.17), *Данные* (рис. 4.18), *События* (рис. 4.19), *Другие* (рис. 4.20) и *Все*.

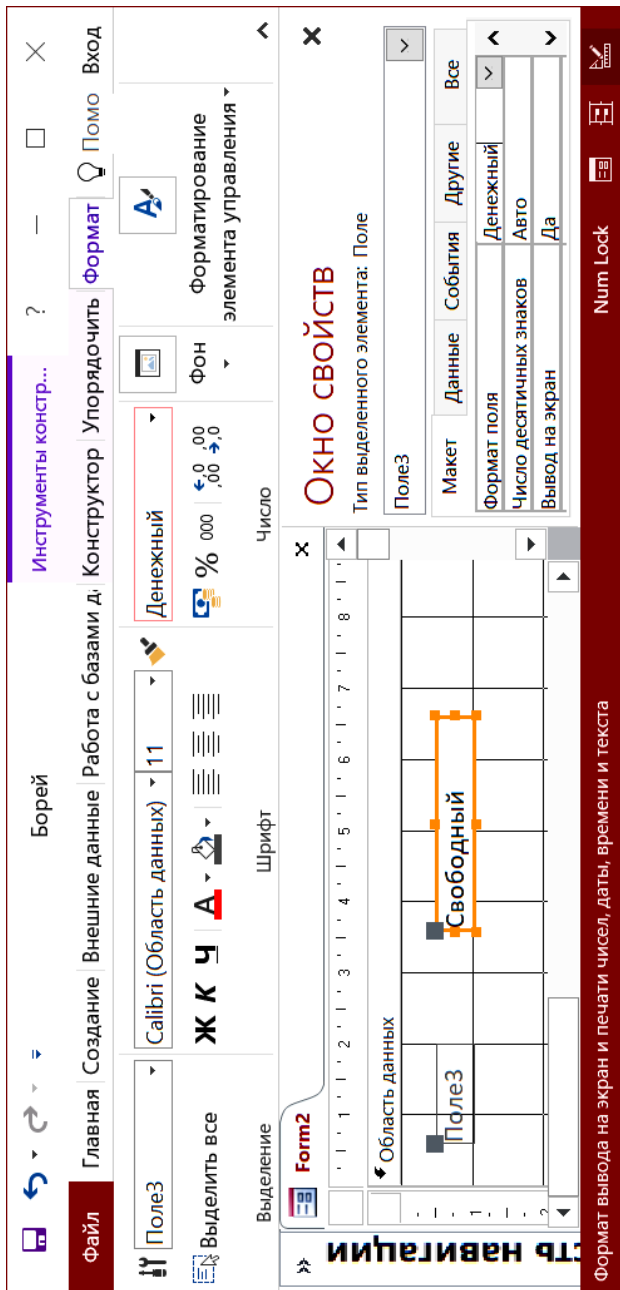


Рис. 4.15. Команды вкладки *Формат* из набора вкладок *Инструменты конструктора форм*

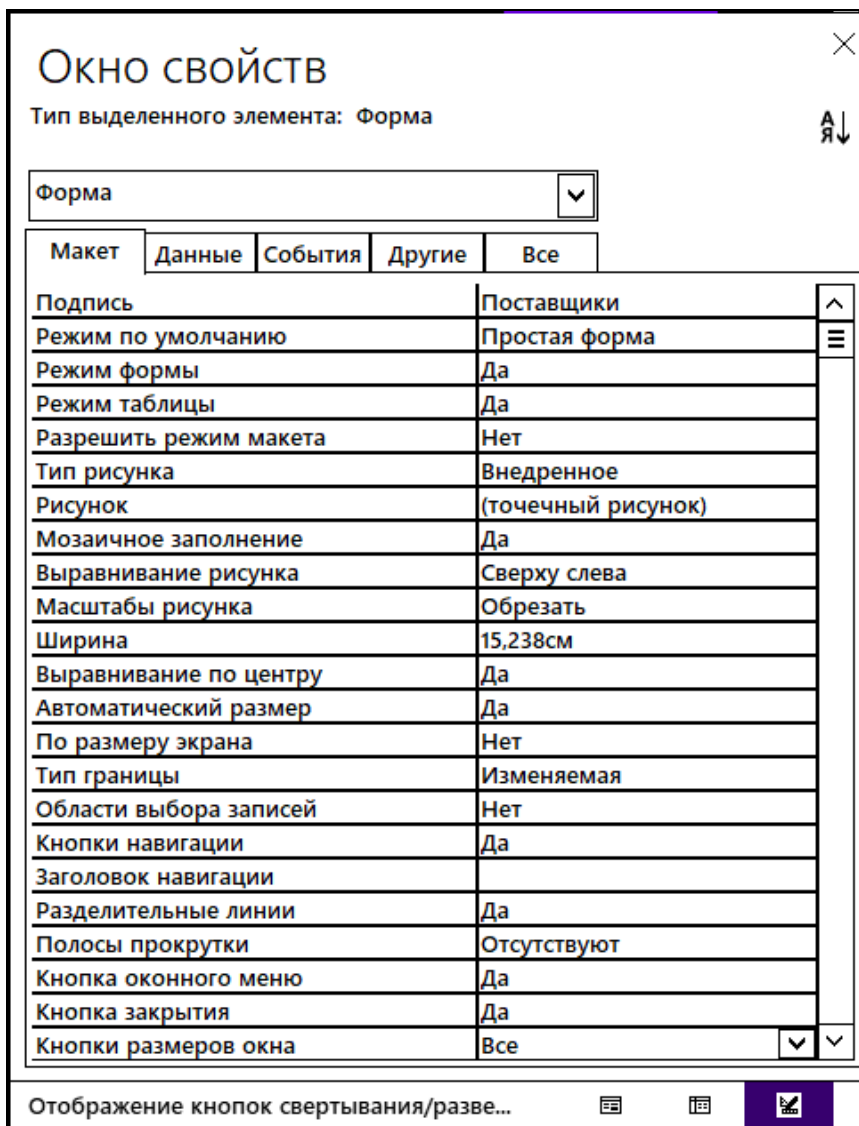


Рис. 4.17. Параметры для настройки макета формы

Окно свойств

Тип выделенного элемента: Форма

Форма

Макет Данные События Другие Все

Источник записей	Поставщики
Тип набора записей	Динамический набор
Загрузить значения по умолчанию	Да
Фильтр	
Фильтр при загрузке	Нет
Порядок сортировки	
Сортировка при загрузке	Да
Ожидание завершающей обработки	Нет
Ввод данных	Нет
Разрешить добавление	Да
Разрешить удаление	Да
Разрешить изменение	Да
Применение фильтров	Да
Блокировка записей	Отсутствует

Базовый объект или инструкция SQL для ...

Рис. 4.18. Параметры для настройки источника данных формы

Окно свойств

Тип выделенного элемента: Форма

Форма

Макет | Данные | События | Другие | Все

Текущая запись	[Процедура обработки]	^
Загрузка		☰
Нажатие кнопки		
После обновления		
До обновления	[Процедура обработки событ	
До вставки		
После вставки		
До подтверждения Del		
Удаление		
После подтверждения Del		
Внесены изменения		
Получение фокуса		
Потеря фокуса		
Двойное нажатие кнопки		
Кнопка вниз		
Кнопка вверх		
Перемещение указателя		
Клавиша вверх		
Клавиша вниз		
Нажатие клавиши		
Отмена		
Открытие		
Закрытие	[Процедура обработки событ	∨




Макрос или функция, запускаемые при пе...   

Рис. 4.19. Параметры для обработки событий в форме

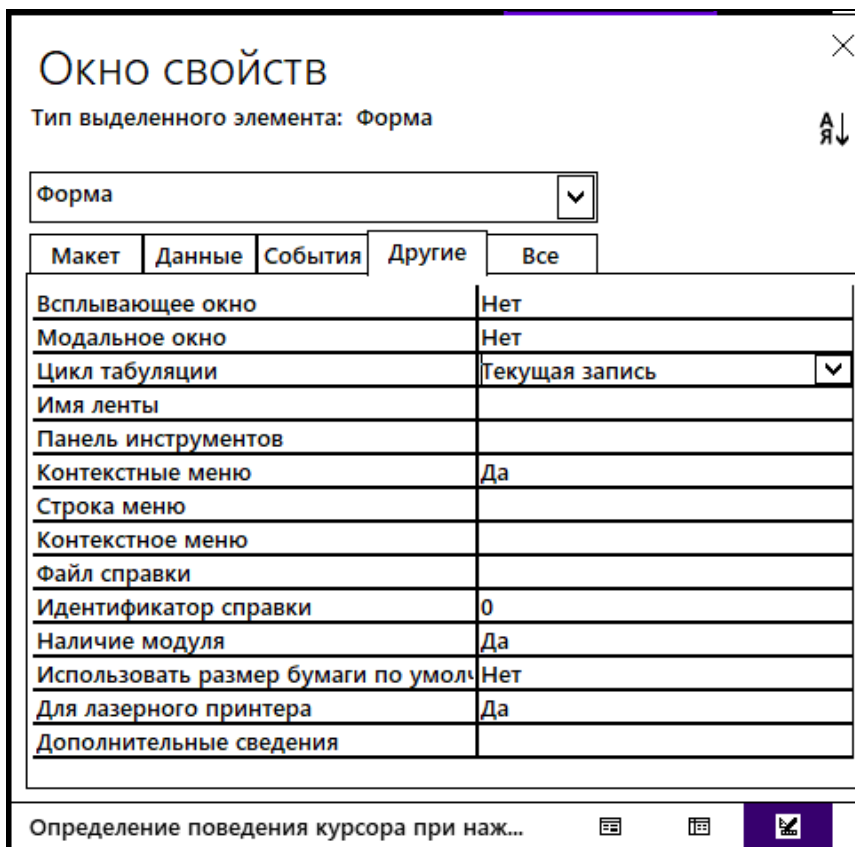


Рис. 4.20. Настройка дополнительных параметров формы

4.3. Работа с элементами управления

Элементами управления (ЭУ) называются объекты в формах, отчетах или на страницах доступа к данным, используемые для оформления, отображения данных или выполнения других действий.

Все ЭУ в формах/отчетах делятся на три группы:

1. *Связанные* (присоединенные) ЭУ – связаны с данными в таблице или запросе и предназначены для вывода данных.
2. *Свободные* ЭУ – независимы от данных в таблице, обычно используются для оформления.
3. *Вычисляемые* ЭУ – создаются для вывода результатов вычислений выражений, использующих данные из таблиц, результаты функций и т. п.

Для работы с формами необходимо в наборе вкладок *Инструменты конструктора форм* открыть вкладку *Конструктор* и отобразить группу команд *Элементы управления* (рис. 4.21). На рисунке видно, что по умолчанию активна кнопка «Выбор объекта» и включен режим автоматического использования мастера при добавлении в форму нового элемента управления. Обозначения элементов управления приведены на рис. 4.22.

Описание назначения основных ЭУ приведено в табл. 4.1.



Рис. 4.21. Команды группы *Элементы управления* на вкладке *Конструктор* из набора вкладок *Инструменты конструктора форм*

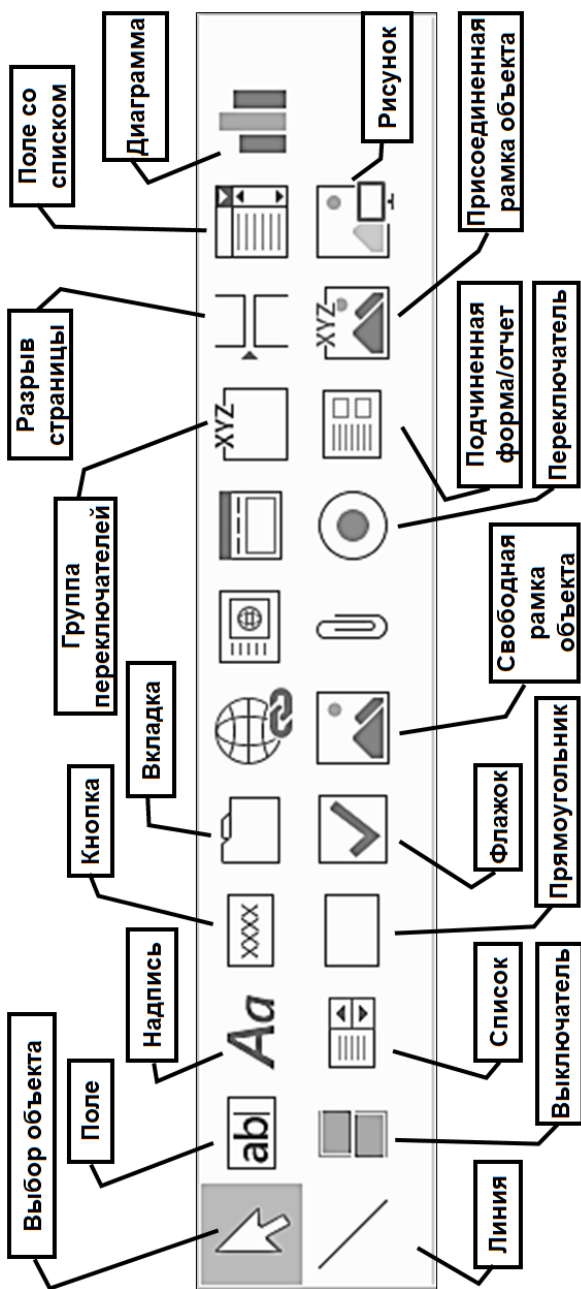


Рис. 4.22. Набор ЭУ в группе команд *Элементы управления* на вкладке *Конструктор* из набора вкладок *Инструменты конструктора форм*

Таблица 4.1

Характеристики элементов управления

Элемент управления	Вид	Назначение
<i>Надпись Label</i>	<i>Aa</i>	Отображение произвольного текстового выражения (пояснения)
<i>Поле TextBox</i>		Ввод текстовой информации, которая может преобразовываться в числа и даты
<i>Поле со списком ComboBox</i>		Хранение списка значений, из которого пользователь может выбрать или ввести с клавиатуры только одно значение
<i>Список ListBox</i>		Хранение списка значений, из которого пользователь может выбрать одно или несколько значений
<i>Флажок CheckBox</i>		Выбор из нескольких возможных вариантов
<i>Переключатель OptionButton</i>		Выбор одного из нескольких взаимоисключающих вариантов
<i>Рамка Frame</i>		Визуальная группировка элементов управления
<i>Кнопка Command Button</i>		Выполнение некоторых действий (макросов) при нажатии кнопки
<i>Набор вкладок TabStrip</i>		Создание нескольких вкладок в диалоговом окне
<i>Рисунок Image</i>		Отображение в форме графических файлов форматов: BMP, GIF, JPG, ICO, WMF

Одним из часто применяемых является ЭУ *Поле* – добавляет в форму *текстовое поле*, которое используется не только для отображения данных из таблиц, но и для вывода значений *вычисляемых выражений*.

Для добавления в форму поля из таблицы или запроса как *связанного ЭУ*, *предназначенного для вывода данных*, необходимо на

вкладке *Конструктор* из набора вкладок *Инструменты конструктора форм* в группе команд *Сервис* нажать кнопку «Добавить поля» (см. рис. 4.14), далее выделить в списке нужное поле и перетащить его в область данных. Для выделения всех полей в списке необходимо выбрать «*».

Форматирование и настройку свойств элементов управления чаще всего выполняют не для каждого элемента по отдельности, а сразу для нескольких – для выделенной группы.

Выделение ЭУ (смежных/несмежных) выполняется ЛКМ при нажатой клавише «Shift»/«Ctrl» соответственно. Выделенные ЭУ (рис. 4.23) по контуру имеют *размерные маркеры* и в верхнем левом углу каждого ЭУ обозначен *маркер перемещения*.

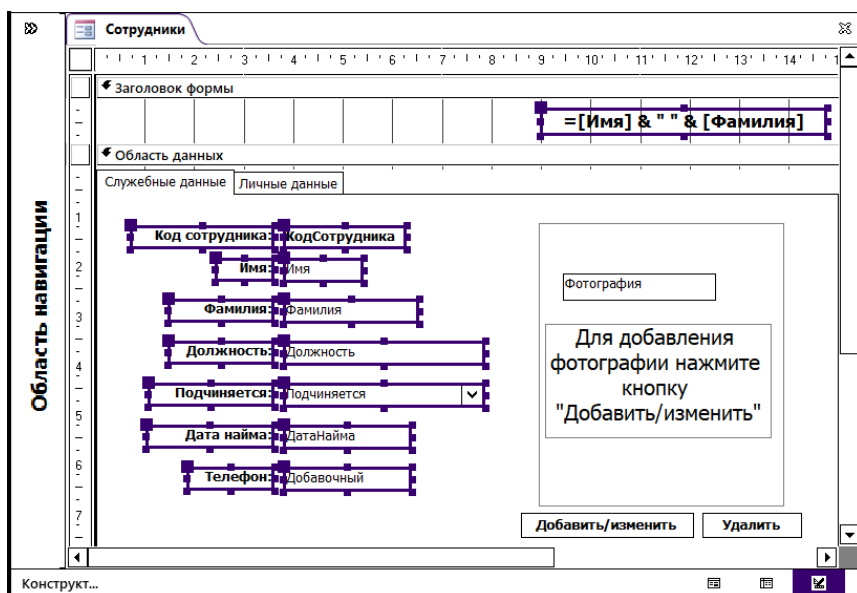


Рис. 4.23. Выделение группы ЭУ

Перемещение группы выделенных ЭУ выполняется ЛКМ без привязки к маркеру перемещения, а для того, чтобы переместить отдельный ЭУ следует ЛКМ нажать на его маркер перемещения.

Для форматирования ЭУ применяются команды на вкладке *Формат* (см. рис. 4.15), для компоновки ЭУ, их выравнивания от-

носительно друг друга применяются команды на вкладке *Упорядочить* (см. рис. 4.16) из набора вкладок *Инструменты конструктора форм*.

С помощью ЭУ *Поле* в формах и отчетах создаются *вычисляемые поля*. Например, в области заголовка формы, представленной на рис. 4.23, расположен ЭУ *Поле* для вывода текстового значения *вычисляемого выражения* вида:

=[Имя] & " " & [Фамилия].

Вычисляемое выражение для ЭУ *Поле* задается для свойства *Данные* на вкладке *Данные* в окне свойств (рис. 4.24) с помощью *Построителя выражений* (см. п. 3.4).

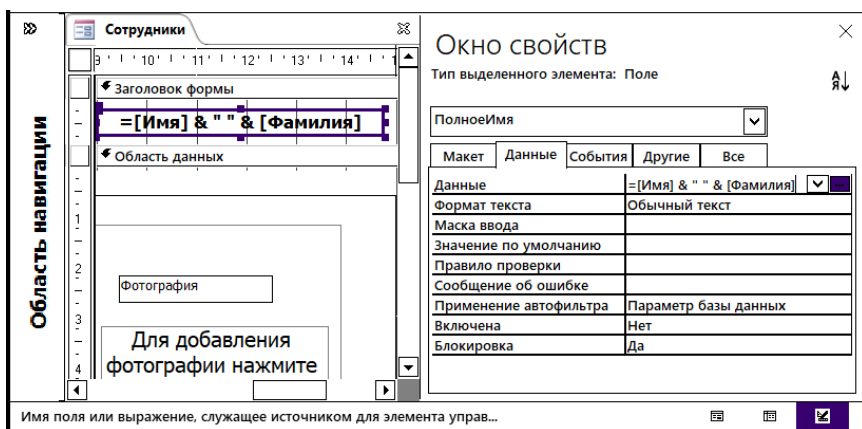


Рис. 4.24. ЭУ *Поле* с вычисляемым выражением
(значение свойства *Данные*)

Значения свойств для *связанного ЭУ* в форме могут не соответствовать значениям тех же свойств поля базовой таблицы, с которым связан ЭУ. Если значения различны, то значения в форме или отчете обычно перекрывают значения в таблице, но применимы только к текущей форме или отчету.

Рекомендуется задавать *в базовой таблице* или *запросе*, а не в форме или отчете, значения для следующих свойств ЭУ: *Формат*,

Число десятичных знаков, Маска ввода, Значение по умолчанию, Правило проверки и Сообщение об ошибке.

В этом случае можно быть уверенным, что для поля заданы нужные значения свойств независимо от того, добавляется ли оно к форме или отчету. Это обеспечивает согласованность настроек при добавлении полей в форму/отчет.

При создании *присоединенных ЭУ* в формах и отчетах путем перетаскивания полей из списка полей MS Access копирует некоторые свойства из поля базовой таблицы или запроса для создаваемого ЭУ. Например, если для поля *Цена* в таблице *Товары* свойство *Формат поля* имеет значение *Денежный*, а свойство *Число десятичных знаков* – значение *Авто*, то при создании присоединенного поля в форме путем перетаскивания поля *Цена* из списка полей для указанных свойств создаваемого ЭУ автоматически будут установлены такие же значения.

Свойства *Значение по умолчанию, Правило проверки и Сообщение об ошибке* не наследуются. Однако значения этих свойств проверяются при создании ЭУ, основанного на поле, для которого свойства были установлены.

Существует возможность изменения значений любых наследуемых свойств в окне свойств ЭУ. Изменение значения свойства ЭУ в форме не влияет на значение этого свойства для поля в базовой таблице или запросе. Аналогично, если изменить значение свойства для поля в таблице или запросе *после создания формы* с использованием этого поля, то значение свойства ЭУ автоматически *не обновляется*, обновление необходимо выполнить вручную.

4.4. Особенности создания и настройки кнопочных форм

Кнопочные формы являются элементами интерфейса БД и предназначены для управления объектами БД. *Главная кнопочная форма (Switch Board, Панель переключений)* обычно создается для того, чтобы ограничить круг задач, выполняемых в БД рядовыми пользователями в соответствии с их правами.

Особенность создания кнопочных форм заключается в том, что до разработки кнопочной формы должны быть созданы все объекты (таблицы, запросы, отчеты, табличные формы), к которым будет

обращаться кнопочная форма. Кнопочную форму можно создать несколькими способами: способами на вкладке *Создание* в группе команд *Форма*: 1) в режиме *Конструктора*; 2) использовать один из макетов, доступных по нажатию кнопки *Навигация*; 3) использовать *Диспетчер кнопочных форм*.

Порядок создания кнопочной формы в режиме *Конструктора*:

1. На вкладке *Создание* в группе команд *Форма* нажать кнопку «Конструктор форм» (не указывать источник формы).

2. Сформировать макет формы, добавить необходимое число ЭУ *Кнопка* и назначить им действия (макросы, процедуры обработки событий) на вкладке *События* окна свойств ЭУ *Кнопка* (рис. 4.25).



Рис. 4.25. Настройка свойств ЭУ *Кнопка*

3. Далее настроить *Свойства формы* на вкладке *Макет* (рис. 4.26):

- задать подпись;
- разрешить режим *Формы*;
- убрать полосы прокрутки, области выбора записей, кнопки навигации (перехода по записям);
- подобрать размеры формы и задать тип границы – окна диалога или тонкую, чтобы запретить возможность изменения размеров формы с помощью ЛКМ.

Окно свойств

Тип выделенного элемента: Форма

A Я ↓

Форма ▾

Макет	Данные	События	Другие	Все
Подпись				Главная кнопочная форма
Режим по умолчанию				Простая форма
Режим формы				Да
Режим таблицы				Нет
Разрешить режим макета				Нет
Тип рисунка				Внедренное
Рисунок				(отсутствует)
Мозаичное заполнение				Нет
Выравнивание рисунка				По центру
Масштабы рисунка				Обрезать
Ширина				10,921см
Выравнивание по центру				Да
Автоматический размер				Да
По размеру экрана				Нет
Тип границы				Окна диалога ▾
Области выбора записей				Нет
Кнопки навигации				Нет
Заголовок навигации				
Разделительные линии				Да
Полосы прокрутки				Отсутствуют
Кнопка оконного меню				Да
Кнопка закрытия				Да
Кнопки размеров окна				Все

Вид границы окна ☰

Рис. 4.26. Пример настройки свойств кнопочной формы

4. Для завершения работы над кнопочной формой следует настроить *параметры запуска формы* при открытии БД с помощью

команды *Параметры* в меню *Файл*, на вкладке *Текущая база данных* диалогового окна «Параметры Access» для параметра *Форма просмотра* выбрать кнопочную форму (рис. 4.27).

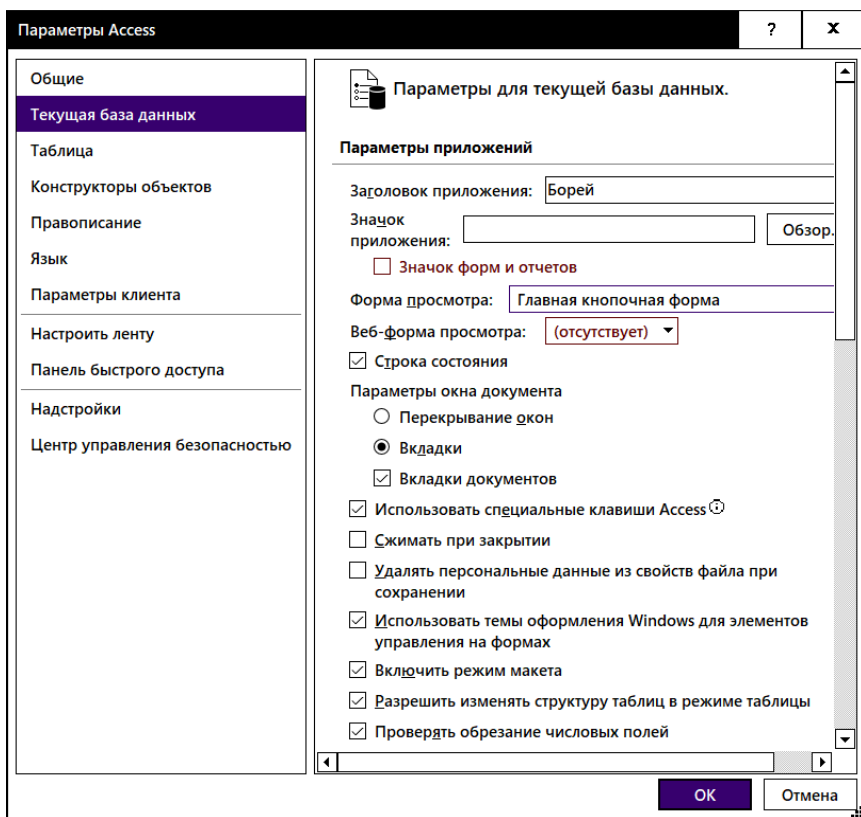


Рис. 4.27. Настройка параметров запуска БД (форма просмотра)

Дополнительно при настройке параметров запуска текущей БД можно задать параметр *Заголовок приложения*, который будет отображаться в строке заголовка при работе с БД вместо стандартного отображения полного пути к файлу БД.

При использовании на вкладке *Создание* в группе команд *Формы* кнопки *Навигация* пользователь может выбрать один из дизайнов панели навигации (рис. 4.28). Далее в режиме *Макета формы*

достаточно перетащить объект БД из области навигации на область вкладок в форме (рис. 4.29), чтобы создать кнопку для работы с выбранным объектом, или добавить необходимое количество кнопок и настроить их свойства.

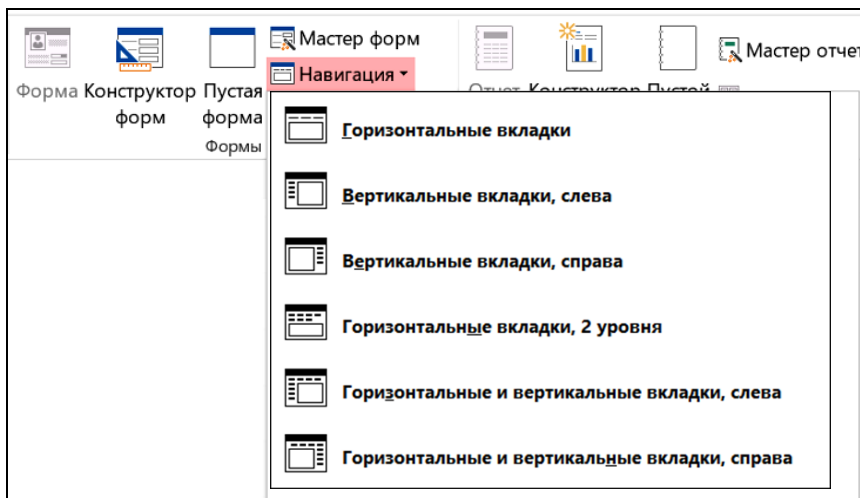


Рис. 4.28. Макеты для создания кнопочной формы для навигации в БД

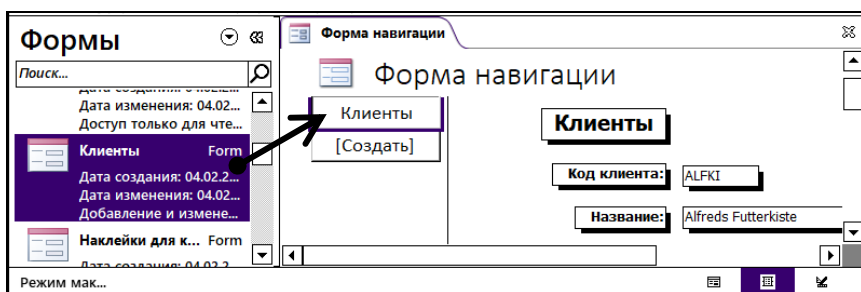


Рис. 4.29. Создание кнопки *Клиенты* (для открытия формы *Клиенты*) в режиме макета формы навигации

Для работы с *Диспетчером кнопочных форм* необходимо предварительно разместить кнопку для вызова команды на вкладке мо-

дульной ленты, используя команду *Параметры* в меню *Файл*, и далее в диалоговом окне «Параметры Access» на вкладке *Настроить ленту* в списке *Выбрать команды* найти в категории «Не на ленте» команду *Диспетчер кнопочных форм* и добавить ее в настраиваемую группу команд выбранной вкладки, например, *Создание*.

Работа над кнопочной формой в режиме *Диспетчера кнопочных форм* позволяет параллельно формировать *Главную* и *подчиненные* кнопочные формы. В этом случае будут созданы элементы управления, оформленные в едином стиле.

4.5. Работа с макросами

Макрос – последовательность инструкций (макрокоманд) на языке VBA, которая записывается единожды и исполняется многократно. Макросы предназначены для выполнения часто повторяющихся операций и для повышения эффективности работы с БД.

Для работы с VBA в MS Access используются 4 кнопки в группе команд *Макросы* на вкладке *Создание*:

- Макрос – непосредственно для записи макроса;
- Модуль;
- Модуль класса;
- Visual Basic.

В приложениях MS Office макрос обычно создается путем записи последовательных нажатий клавиш. Однако в MS Access используется принципиально другой способ создания: макросы записываются в режиме *Конструктора* (рис. 4.30) путем выбора команды из списка встроенных макрокоманд и задания необходимых аргументов.

После сохранения макрос, как правило, назначается кнопкам в формах БД (или разделам формы) и выполняется либо при нажатии на кнопку в форме, либо непосредственно из окна БД (после выбора объекта в области навигации БД на вкладке *Макросы* используется команда *Выполнить* из КЗМ или команда *Выполнить макрос* в группе команд *Макрос* на вкладке *Работа с базами данных*).

При создании макроса последовательно в левой области окна *Конструктора* добавляется *макрокоманда*, причем каждая располагается на отдельной строке в порядке выполнения и выбирается из стандартного списка.

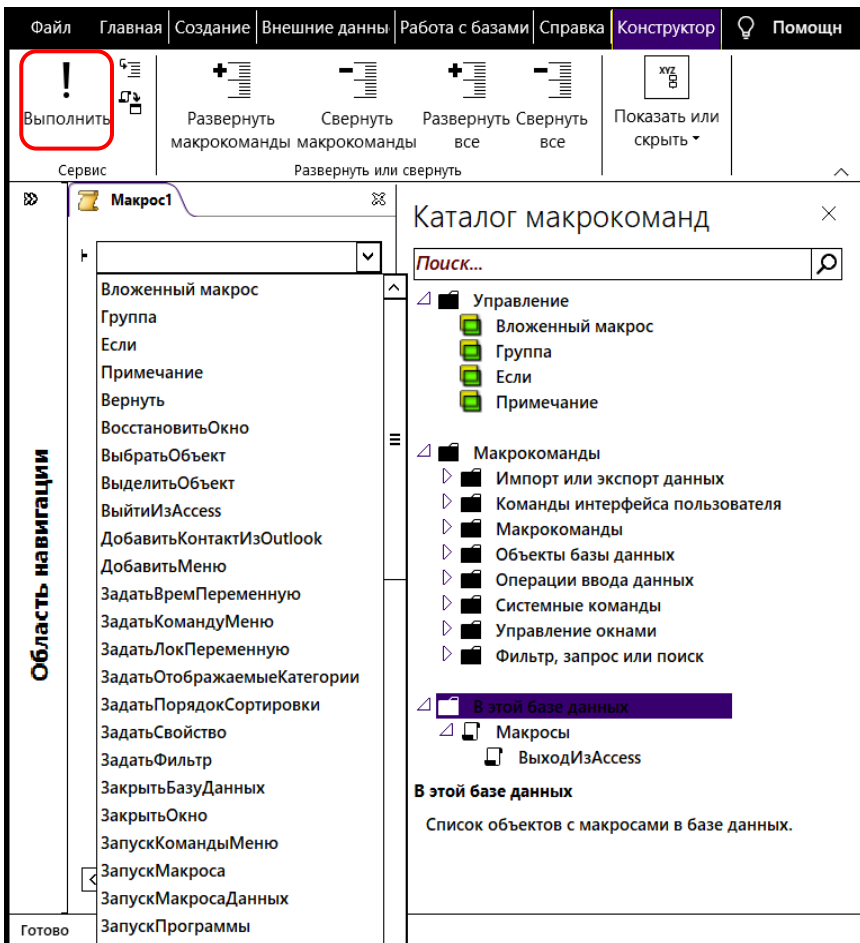


Рис. 4.30. Окно *Конструктора* макросов

Область *Аргументы макрокоманды* становится доступной только после выбора из списка макрокоманды. В этой области находится список аргументов для конкретной макрокоманды.

Большинство макрокоманд требует *здать значения аргументов* (рис. 4.31).

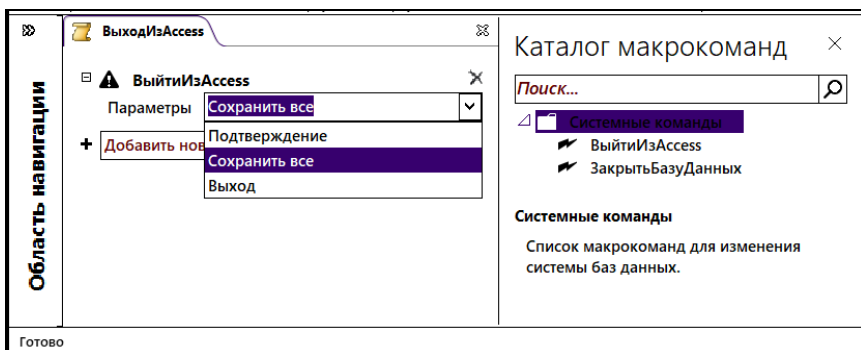


Рис. 4.31. Пример макрокоманды *ВыйтиИзAccess* с аргументом *Параметры*

При закрытии окна *Конструктора* выполняется запрос на сохранение макроса, в котором необходимо заменить присваиваемое по умолчанию имя *Макрос1* именем макроса, соответствующем выполняемому действию.

Например, для записи макроса *ОткрТаблКлиенты* (для открытия таблицы *Клиенты*), необходимо в режиме *Конструктора* добавить макрокоманду *ОткрытьТаблицу* (выбрать из списка) и задать для нее аргументы (рис. 4.32): *Имя таблицы*, *Режим* и *Режим данных*.

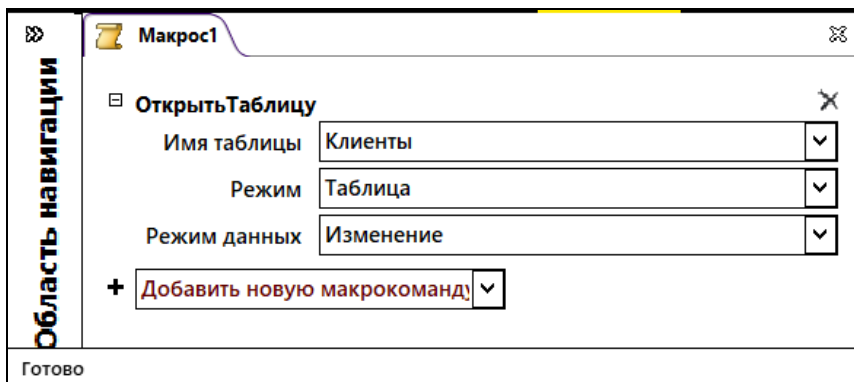


Рис. 4.32. Аргументы макрокоманды *Открыть таблицу*

После записи макрос, как и любой другой объект в режиме *Конструктора*, можно редактировать: удалять и добавлять строки с макроскомандами, изменять значения аргументов.

Выполнение макроса может осуществляться одним из следующих способов:

1) нажать кнопку «Выполнить» (см. рис. 4.30) на вкладке *Конструктор* из набора вкладок *Работа с макросами*;

2) в форме (в режиме *Формы*) нажать ЭУ *Кнопка*, которой был задан макрос для свойства *Нажатие кнопки*;

3) автоматически выполняется макрос из формы или отчета, например, при их открытии или закрытии, если назначить макрос соответствующей команде в окне *Свойств* на вкладке *Событие* для выбранной области или элемента управления.

Например, для текстового поля можно назначить макрос для свойств *Вход и/или Выход*, который будет выполняться при выборе поля или выходе из него (отмене);

4) при открытии БД автоматически выполняется макрос, сохраненный под именем *AutoExec*.

Например, при работе с БД необходимо регулярно выполнять следующие действия:

- 1) открыть таблицу *Клиенты* в режиме *Таблица*;
- 2) развернуть окно таблицы во весь экран;
- 3) переместить курсор к новой записи.

В этом случае целесообразно создать макрос *AutoExec* (специальный макрос с возможностью автозапуска при открытии БД). В режиме *Конструктора макросов* в области *Макроскоманда* необходимо последовательно выбрать для макроса 2 команды и задать их аргументы:

1. Команда *ОткрытьТаблицу* со значениями аргументов:
Имя = *Клиенты* (выбрать имя из списка таблиц);
Режим = *Таблица*;
Режим данных = *Изменение*.
2. Команда *На запись* со значениями аргументов:
Тип объекта = *Таблица*;
Имя объекта = *Клиенты*;
Запись = *Новая*.

Затем необходимо сохранить макрос, задав имя *AutoExec*.

Чтобы проверить работоспособность созданного макроса *AutoExec*, необходимо заново открыть БД. Следует помнить, что при открытии БД сначала выполняются *параметры запуска*, заданные для текущей БД в окне «Параметры Access» (см. рис. 4.27), а затем будет выполнен макрос *AutoExec* (поэтому между макрокомандами не должно быть противоречий).

Чтобы *пропустить* выполнение макроса *AutoExec*, необходимо при открытии БД удерживать клавишу «Shift».

Чтобы *отменить* автоматическое выполнение макроса *AutoExec* при открытии БД можно *переименовать* или *удалить* макрос *AutoExec*.

Дополнительные возможности при создании макроса:

1. Можно сформировать *Макрогруппу* и присвоить имена макросам, входящим в нее.

2. Можно добавить в макрос макрокоманды, выполняемые *по условию* (рис. 4.33), которые будут выполняться только в случае, если значения условия возвращает значение «Истина».

3. Можно выполнить *отладку* макроса по шагам (для обнаружения ошибок, в специальном окне), нажав кнопку «По шагам» в группе команд *Сервис* на вкладке *Конструктор*.

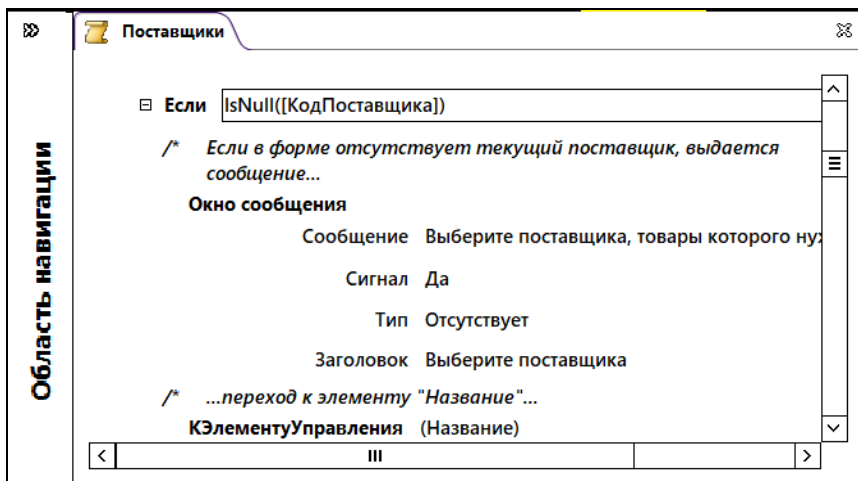


Рис. 4.33. Пример макроса с условием

4.6. Понятие и назначение отчетов, особенности создания и настройки

Отчет – объект СУБД MS Access, предназначенный для подготовки и вывода на печать итоговых документов в удобном для пользователя виде. Основой для создания отчетов служат таблицы и запросы БД.

Назначение отчетов:

1. Подготовка итоговых документов с возможностью группировки данных по различным признакам.
2. Создание почтовых наклеек.
3. Представление данных в виде диаграмм и сводных таблиц Excel.

В MS Access отчет, как и форма, в режиме *Конструктора* имеет стандартную структуру, пример которой приведен на рис. 4.34.



Рис. 4.34. Пример структуры отчета в режиме *Конструктора*

Структура отчета включает одну обязательную область – область *данных* и необязательные области: *верхний/нижний колонтитулы*, *заголовок/примечание отчета* и, в случае применения группировки по отдельным полям, дополнительно доступны области *заголовок/примечание группы*.

Так же как и в формах, в отчетах используются ЭУ для создания связи между отчетом и исходными данными.

Для создания отчета на вкладке *Создание* в группе команд *Отчет* используются кнопки:

1. *Отчет* – создается отчет на основе выделенного в области навигации БД объекта (таблицы или запроса), который включает все поля и записи из источника.

2. *Конструктор отчетов* – создается пустой отчет и открывается для последующей доработки в режиме *Конструктора*.

3. *Пустой отчет* – создается пустой отчет и открывается для оформления в режиме *Макета*.

4. *Мастер отчетов* – вызывает серию диалоговых окон для создания макета отчета, который можно редактировать в режиме *Конструктора*.

5. *Наклейки* – вызывает серию диалоговых окон для формирования макета почтовых наклеек с контактами получателей сообщений.

Для получения опыта создания профессионально оформленных отчетов рекомендуется использовать режим *Мастера*.

При создании отчета в режиме *Мастера* отчетов выполняются следующие шаги:

1) в диалоговом окне указывается *источник данных* – таблица или запрос, и выбираются поля, которые должны быть отражены в отчете. Можно добавить поля из другой таблицы или запроса (выбрав из списка в этом же окне);

2) на следующем шаге задается вид представления (срез) данных, *например*, по клиенту или сотруднику;

3) далее в соответствии с выбранным представлением выбираются уровни группировки данных, может быть задано до 4-х уровней группировки данных;

4) далее задается *сортировка записей* (до 4-х полей сортировки) и необходимые вычисления для подведения итогов (Sum, Avg, Max, Min) по числовым полям;

5) далее выбирается *макет и ориентация страницы* отчета (можно задать опцию «настроить ширину полей» для размещения отчета на одной странице);

6) на последнем шаге мастера необходимо задать *имя отчета*. Может быть выбрана опция «*Просмотр*» или «*Изменение структуры*».

После завершения работы мастера создается отчет и по умолчанию выводится на экран в режиме *Предварительного просмотра*. Если была выбрана опция «*Изменение структуры*», происходит переход в режим *Конструктора*.

Создание отчета в режиме Конструктора имеет ряд особенностей:

1. В режиме *Конструктора* необходимо вывести на экран список полей источника, нажав кнопку «*Добавить поля*» в группе команд *Сервис* на вкладке *Конструктор*. Далее с помощью мыши перетаскивать поля из списка в требуемые области отчета.

2. Для работы с отчетом в режиме *Конструктора* используются вкладки из набора вкладок *Работа с макетами отчетов*.

3. Для изменения внешнего вида ЭУ так же, как и в форме, необходимо выделить ЭУ и использовать набор свойств для изменения шрифта, цвета и др. Для изменения формата отображенных данных в ЭУ необходимо выделить ЭУ, вызвать контекстное меню *Свойства* и задать необходимые значения.

4. Для перемещения ЭУ, изменения их размера и выравнивания необходимо выделить группу ЭУ и использовать вкладку *Упорядочить* из набора вкладок *Работа с макетами отчетов* или команды контекстного меню.

5. Используя группу команд *Элементы управления* вкладки *Конструктор*, в отчет добавляются свободные и вычисляемые ЭУ.

Например, чтобы расположить в колонтитуле номер страницы отчета, в режиме Конструктора выполняется последовательность действий:

- 1) добавить в область колонтитула свободный ЭУ *Текстовое поле*;
- 2) для ЭУ вызвать окно *свойств* по команде *Свойства* из КЗМ;
- 3) в окне свойств на вкладке *Данные* для свойства *Данные* вызвать окно «*Построитель выражений*», в котором создать вычисляемое выражение: в категории элементов выбрать *Общие выражения*, в подкатегории выбрать *Номер страницы* и нажать кнопку «*Вставить*». После этого вычисляемое выражение отображается как

в окне свойств (рис. 4.35), так и непосредственно в самом ЭУ в области отчета, заменив текст по умолчанию – *Свободный*.

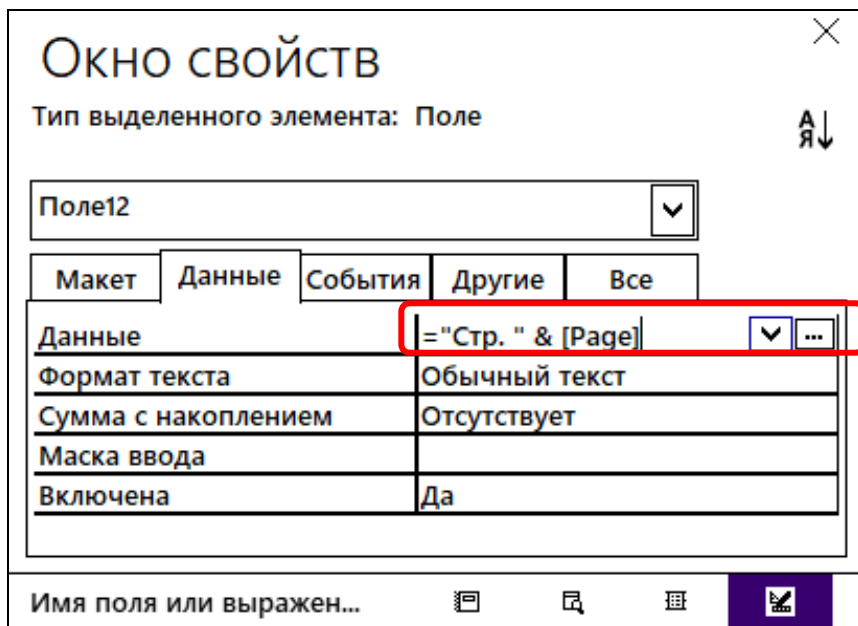


Рис. 4.35. Свойство свободного ЭУ для вывода номера страницы в отчете

Аналогично можно вывести в текстовом поле и другую информацию, *например*, текущую дату в области колонтитула или номер строки отчета.

Для нумерации строк отчета в области данных отчета размещается *вычисляемое поле* (рис. 4.36), для которого на вкладке *Данные* в окне свойств для параметра *Данные* задается начальное значение «=1» и для параметра *Сумма с накоплением* – значение «Для группы». Также в этом отчете для изменения порядка группировки используется кнопка «Группировка» на вкладке *Конструктор отчетов* (см. рис. 4.36).

Результат группировки и нумерации строк отчета в режиме предварительного просмотра приведен на рис. 4.37.

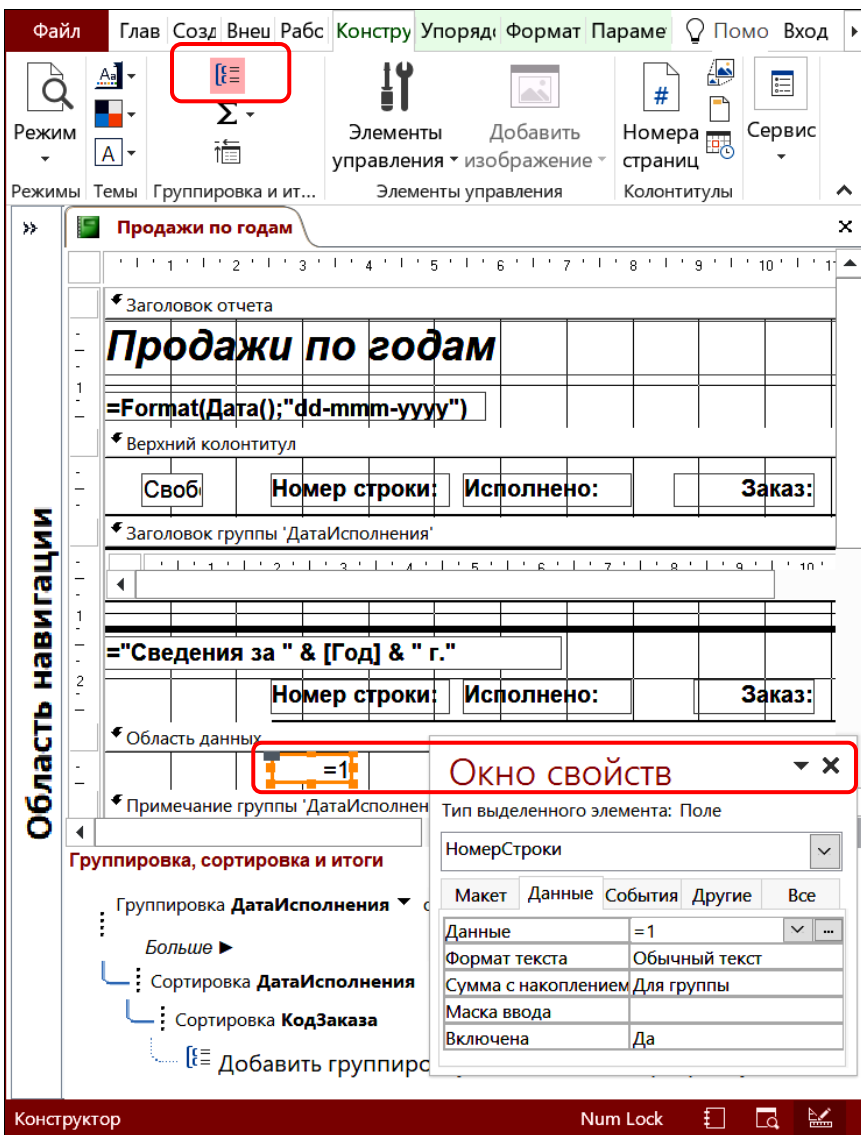


Рис. 4.36. Макет отчета с группировкой по полю *ДатаИсполнения* и полем для нумерации строк в пределах группы

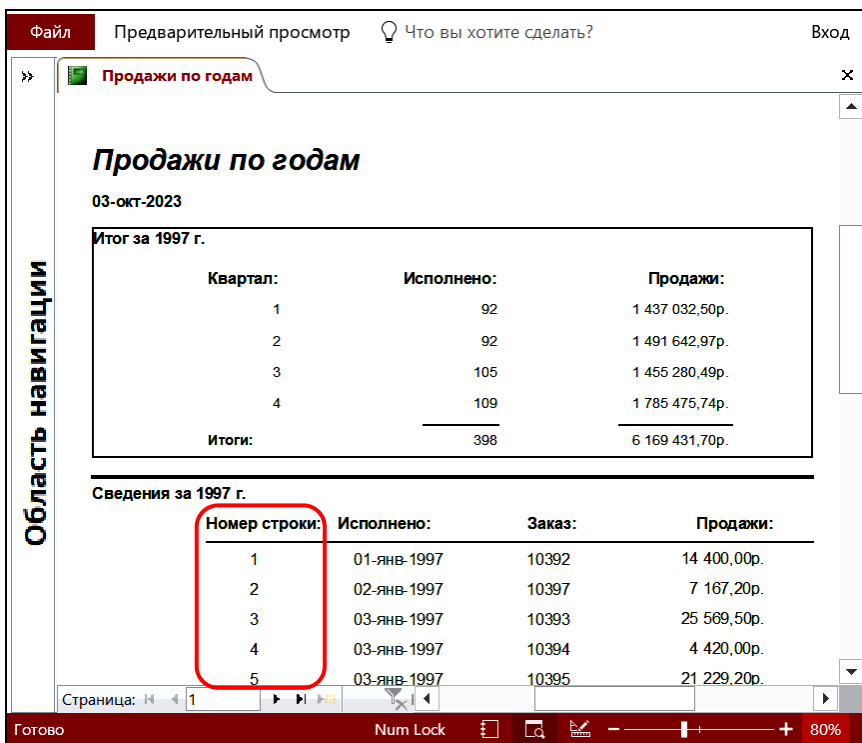


Рис. 4.37. Результат группировки данных и нумерации строк в отчете (режим *Предварительного просмотра*)

Таким образом, отчеты позволяют обобщать, группировать данные и получать печатную копию сводных данных в хорошо структурированном виде.

Так же, как и в форму, в отчет в режиме *Конструктора* можно добавить диаграмму (Microsoft Graph), используя соответствующий ЭУ *Диаграмма*.

Просмотр отчета можно выполнить как в режиме образца макета, так и в режиме *Предварительного просмотра* перед печатью (рис. 4.38).

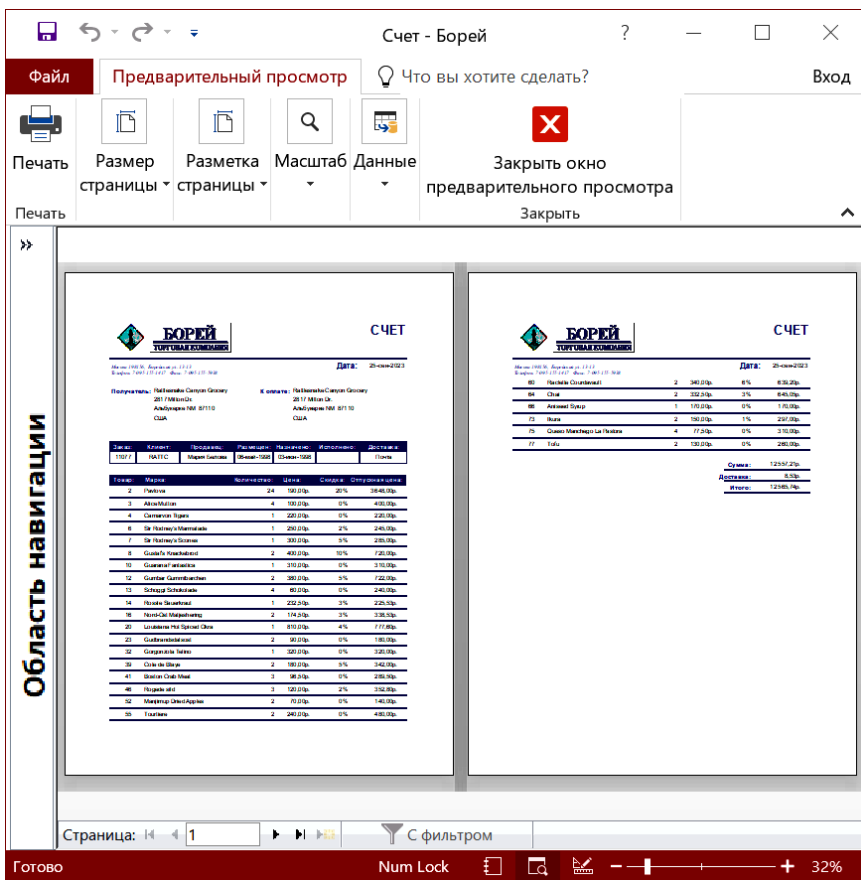


Рис. 4.38. Предварительный просмотр отчета в масштабе 2 страниц

До печати отчета обязательно задать формат и ориентацию бумаги, размеры полей на вкладке *Параметры страницы* (рис. 4.39). Стоит отметить, что при установке флажка *Печатать только данные*, при печати не выводятся элементы оформления отчета.

MS Access сохраняет настройки печати вместе с отчетом, поэтому их устанавливают один раз.

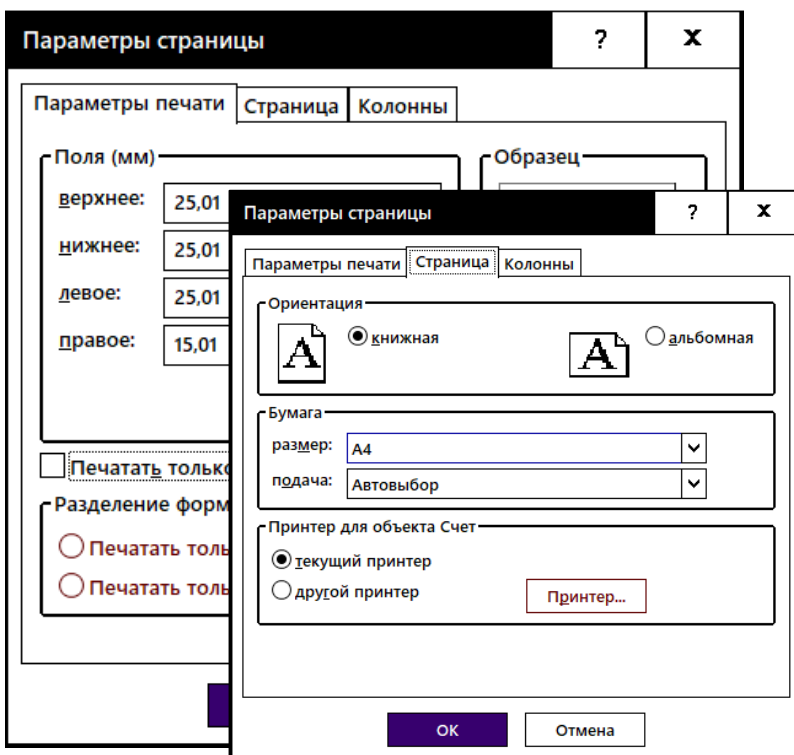


Рис. 4.39. Настройка параметров страницы для отчета

Отчеты могут быть не только распечатаны, но и отправлены в качестве вложений к сообщению электронной почты в виде PDF-файла или запроса к списку рассылки в виде книги Excel. Для этого достаточно в режиме *Предварительного просмотра* в группе команд *Данные* использовать соответствующие команды (рис. 4.40).

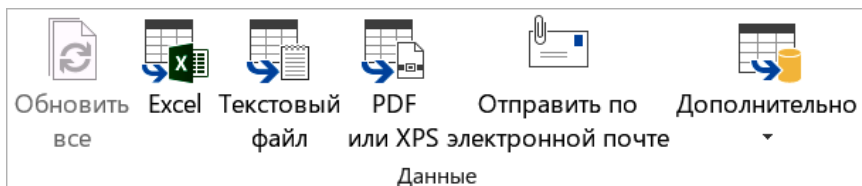


Рис. 4.40. Команды для вывода отчета в различных форматах

Вопросы для самопроверки

1. Какие инструменты СУБД MS Access применяются для организации пользовательского интерфейса БД?
2. Для чего предназначены формы? Какие виды форм можно создать в СУБД MS Access?
3. Какие объекты могут быть источниками при создании форм?
4. Назовите способы создания табличных форм в СУБД MS Access. Какие формы предпочтительно создавать в режиме *Мастера*?
5. Опишите порядок создания табличных форм в режиме *Конструктора*. Какую структуру имеет форма/отчет? В чем заключаются особенности создания многотабличных форм?
6. Что такое «элементы управления» в формах/отчетах? На какие группы подразделяются элементы управления в СУБД MS Access?
7. С помощью каких элементов управления организуют вывод данных в формах/отчетах? Приведите примеры.
8. Какие элементы управления используются для оформления форм/отчетов? Приведите примеры.
9. Каким образом в формах/отчетах создаются вычисляемые элементы управления? Опишите порядок создания и настройки свойств вычисляемых элементов управления в формах/отчетах.
10. Для чего предназначены кнопочные формы? Перечислите способы создания кнопочных форм и их особенности.
11. Опишите порядок настройки свойств кнопочных форм и параметров запуска при открытии БД.
12. Что такое «макрос»? Приведите порядок создания и настройки макросов. Как назначить макрос элементу управления или разделу формы/отчета?
13. Для чего предназначены отчеты? Какие они бывают? Что может быть источником отчета?
14. Перечислите способы создания отчетов. Опишите особенности создания отчетов в режиме *Конструктора*. Как создаются вычисляемые поля в отчетах? Приведите примеры.
15. Приведите типовые приемы работы с выделенной группой элементов управления при оформлении форм/отчетов.
16. Какие параметры настраивают перед печатью форм/отчетов? Как распечатать только данные форм/отчетов?

Тема 5. СПЕЦИАЛЬНЫЕ ПРИЕМЫ РАБОТЫ С СУБД MS ACCESS

5.1. Взаимодействие MS Access с приложениями MS Office. Технологии интеграции реляционных БД и офисных приложений.

5.2. Использование механизма слияния в MS Word для подготовки типовой рассылки.

5.3. Настройка рабочей среды MS Access.

5.1. Взаимодействие MS Access с приложениями MS Office. Технологии интеграции реляционных БД и офисных приложений

Взаимодействие MS Access с приложениями MS Office обеспечивают следующие технологии и инструменты:

1. *Буфер обмена.*

2. *Технология OLE* (тип данных – *Поле объекта OLE*).

3. *Технология ODBC* – набор функций API, позволяющих обрабатывать информацию из БД на SQL-серверах без непосредственной установки MS Access (*MS Query*).

4. Команды на вкладке *Внешние данные* (группа команд *Экспорт*) для экспорта объектов БД в форматы:

– *Excel* (рабочая книга с расширением .XLSX);

– *Текстовый файл* (с расширением .TXT);

– *XML-файл*;

– *PDF* или *XPS*;

– *Слияние в Word* (документ с расширением .DOCX);

– *Word* (документ формата RTF);

– а также доступные форматы БД (.ACCDB, .DBF).

Документы, полученные в результате экспорта объектов, по умолчанию сохраняются в папке, путь к которой указан в поле *Рабочий каталог* на вкладке *Общие* диалогового окна «*Параметры Access*».

В MS Access существует возможность *конвертирования данных* – преобразования из одного офисного формата представления данных в другой. Набор доступных форматов импорта и экспорта зависит от выбранного объекта БД (табл. 5.1) и параметров установки MS Access.

Варианты экспорта объектов БД в приложения MS Office

Объекты БД	Доступность для объектов БД команд на вкладке <i>Экспорт</i> в группе <i>Внешние данные</i>		
	Excel	Word	Слияние в Word
Таблица	+	+	+
Запрос	+	+	+
Форма	+	+	—
Отчет	+	+	—

Для конвертирования объектов БД используются:

- 1) команда *Сохранить объект как* в меню *Файл*;
- 2) набор команд на вкладке *Внешние данные* в группах *Импорт и связи* и *Экспорт*;
- 3) команда *Экспорт* из КЗМ для объекта, выделенного в области навигации БД (на рис. 5.1 представлен набор команд, доступных для экспорта запроса).

Чаще всего из БД экспортируются запросы в рабочие книги Excel, с целью дальнейшего анализа данных. В рабочей книге MS Excel экспортированный запрос (или таблица) располагается на рабочем листе и представляет собой обычный диапазон ячеек, который можно форматировать, выполнять вычисления, использовать все доступные в MS Excel технологии и инструменты обработки и анализа табличных данных [13]: сортировку, фильтрацию, промежуточные итоги, сводные таблицы и сводные диаграммы, инструменты статистического анализа.

Для визуального анализа табличных данных в MS Excel применяются инструменты деловой графики: диаграммы и линии тренда. Подробное описание технологий MS Excel для обработки и анализа табличных данных приведено в пособии [13].

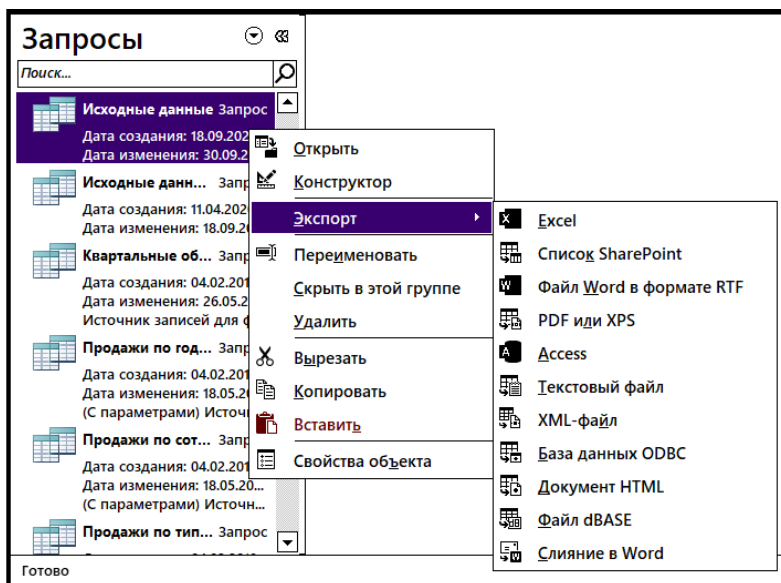


Рис. 5.1. Выбор варианта *Слияния*

5.2. Использование процедуры *Слияние в Word* для подготовки типовой рассылки

Слияние – стандартная процедура приложений MS Office, которая используется:

- 1) для создания документов на бланке;
- 2) для подготовки почтовых наклеек и конвертов;
- 3) для создания каталогов;
- 4) а также при массовой рассылке сообщений по факсу или по электронной почте.

В процессе слияния различают три типа документов: *основной документ* (документ на бланке), *источник данных* и *документ с результатом слияния* (итоговый многостраничный документ).

1. *Основной документ* – документ MS Word (письмо или документ на бланке) состоит из 2-х частей, в зависимости от характера размещаемой информации:

1) *постоянной части* (или *текста письма*), которая содержит данные, неизменные во всех производных документах при слиянии (*например*, обратный адрес отправителя, текст сообщения);

2) *переменной части*, которая содержит набор полей слияния из источника данных для отображения реквизитов получателя.

2. *Источник данных* – таблица с заголовками полей (столбцов), содержащая сведения (*например*, список имен и адресов получателей рассылки), предназначенные для объединения с основным документом. В качестве источника данных могут использоваться таблицы в файлах Word и Excel, а также таблицы или запросы БД. Для использования сведений из источника данных к нему необходимо подключиться в основном документе Word.

3. *Документ с результатом слияния* – многостраничный документ Word: письмо на бланке, почтовая наклейка, конверт для каждой записи (строки) из источника данных (таблицы, запроса).

Для подготовки серийного письма, на базе таблицы или запроса MS Access необходимо выполнить следующие действия:

1) выделить источник слияния в БД на вкладке *Таблицы* или *Запросы* и на вкладке *Внешние данные* в группе команд *Экспорт* выполнить команду *Слияние в Word*;

2) выбрать вариант слияния (рис. 5.2). При отсутствии готового документа с текстом слияния выбрать переключатель «Создать новый документ и установить связь с ним»;

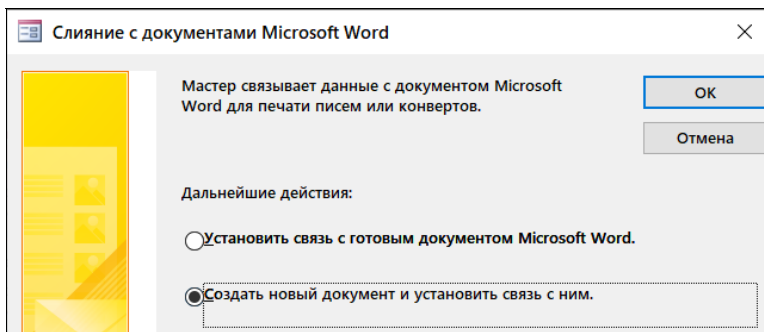


Рис. 5.2. Выбор варианта для механизма *Слияния*

3) в MS Word создать или отредактировать образец письма для рассылки – постоянную часть макета документа, включая, *например*, реквизиты компании-отправителя и стандартный текст, предназначенный для рассылки;

4) в документ MS Word добавить поля слияния из источника данных, используя команду *Вставить поле слияния* в группе ко-

манд *Составление документа и вставка полей* на вкладке *Рассылки* для выбора поля из списка или из диалогового окна (рис. 5.3);

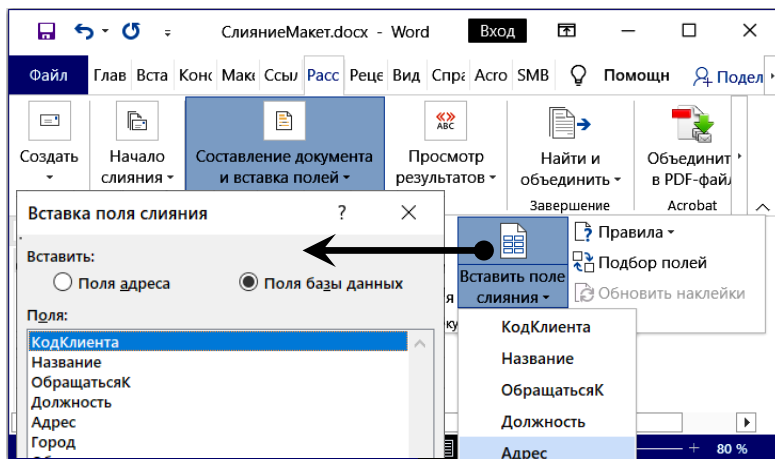


Рис. 5.3. Добавление полей *Слияния*

5) для персонализации обращения, *например*, к клиенту, можно добавить поле MS Word «If ... Then ... Else» (рис. 5.4), используя команду *Правила* в группе команд *Составление документа и вставка полей* на вкладке *Рассылки*;

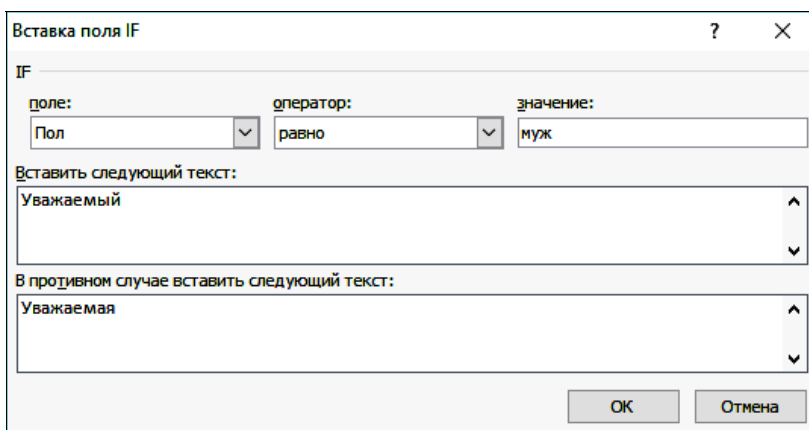


Рис. 5.4. Вставка поля MS Word *If ... Then ... Else* для персонализации обращения к клиенту

б) оценить полученный основной документ в режиме отображения кодов полей (рис. 5.5) и просмотреть макет документа MS Word в режиме отображения значений из источника данных, используя кнопку «Просмотреть результаты» в группе команд *Просмотр результатов* на вкладке *Рассылки* (рис. 5.6);

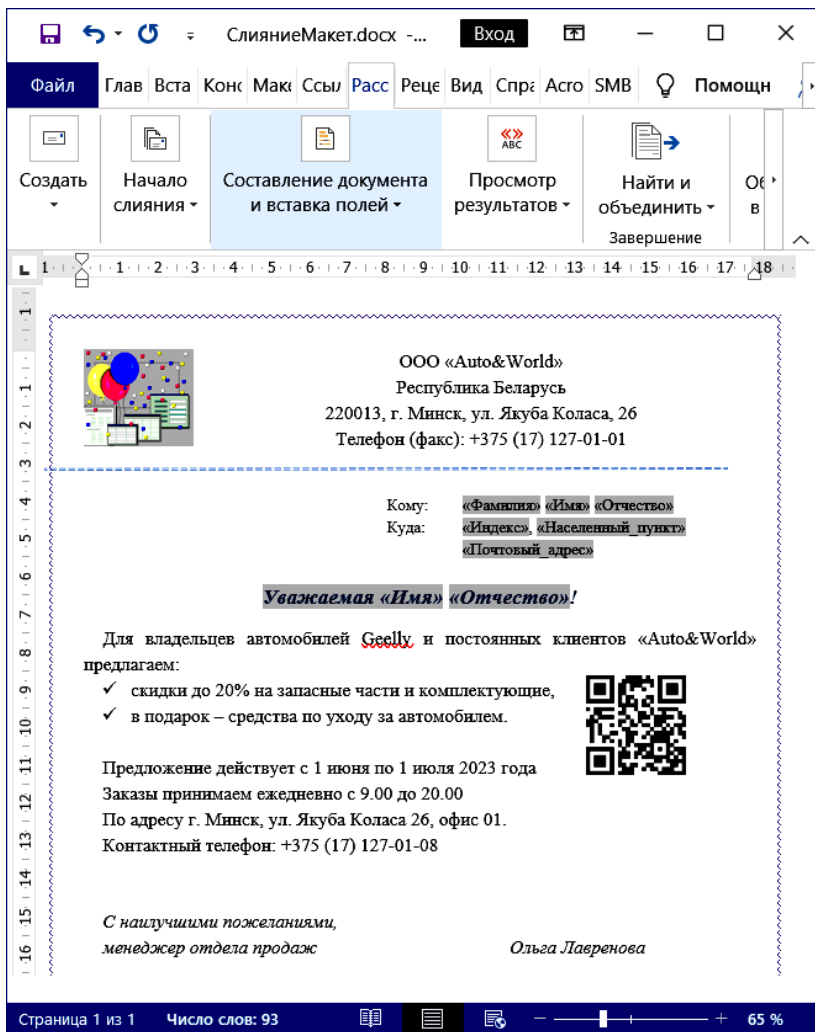


Рис. 5.5. Основной документ (макет письма слияния) в режиме отображения кодов полей слияния

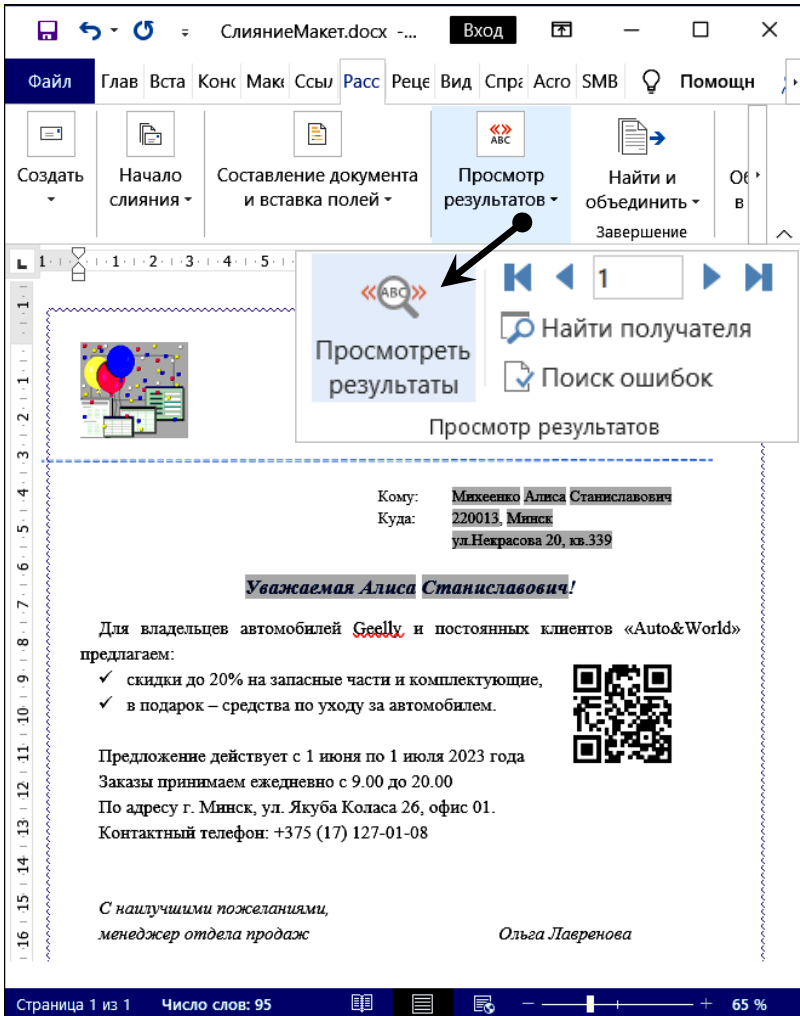


Рис. 5.6. Основной документ в режиме просмотра результатов (с данными в полях слияния первой записи из источника)

7) далее следует завершить создание рассылки, нажав на вкладке *Рассылки* в группе команд *Завершение* кнопку «Найти и объединить» и выбрав команду *Изменить отдельные документы*.

На этом шаге в диалоговом окне «Составные новые документы» есть возможность выбрать для слияния не все, а конкретные записи

из источника (рис. 5.7), затем подтвердить объединение данных из источника с текстом письма слияния и оценить полученный многостраничный документ (при одностраничном макете письма слияния число страниц готового документа рассылки будет равно числу записей в источнике данных, рис. 5.8). По умолчанию итоговый документ слияния получает имя *Письма1*.

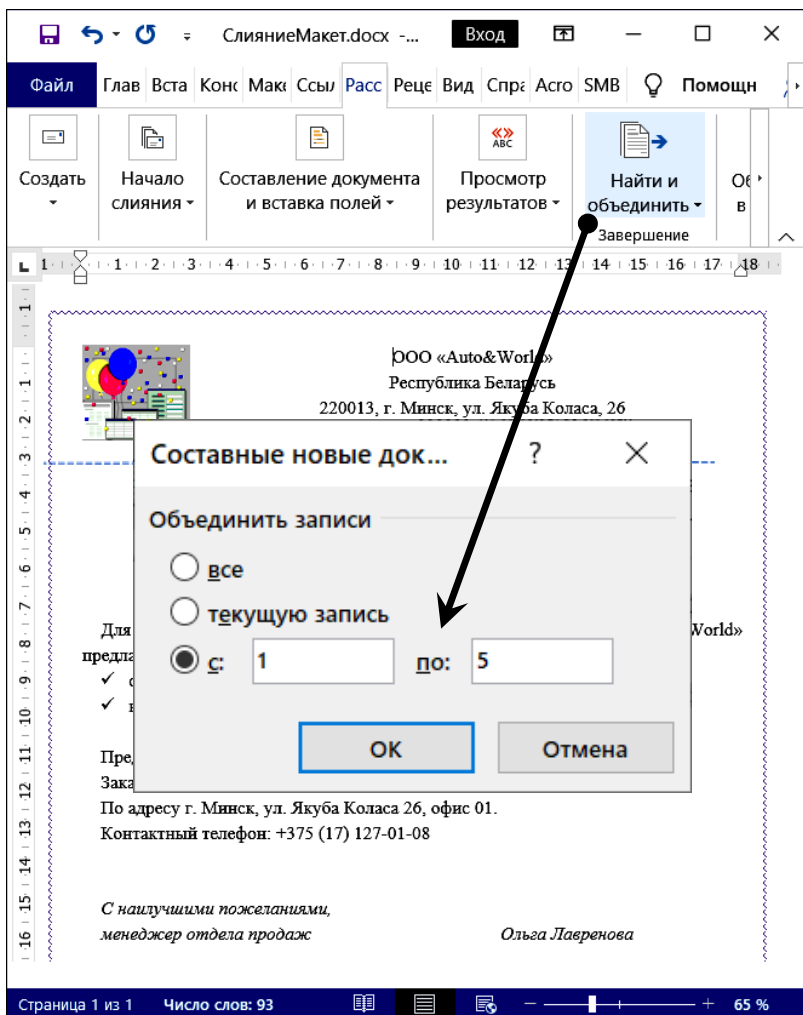


Рис. 5.7. Завершение процесса слияния (отбор первых 5-ти записей из источника)

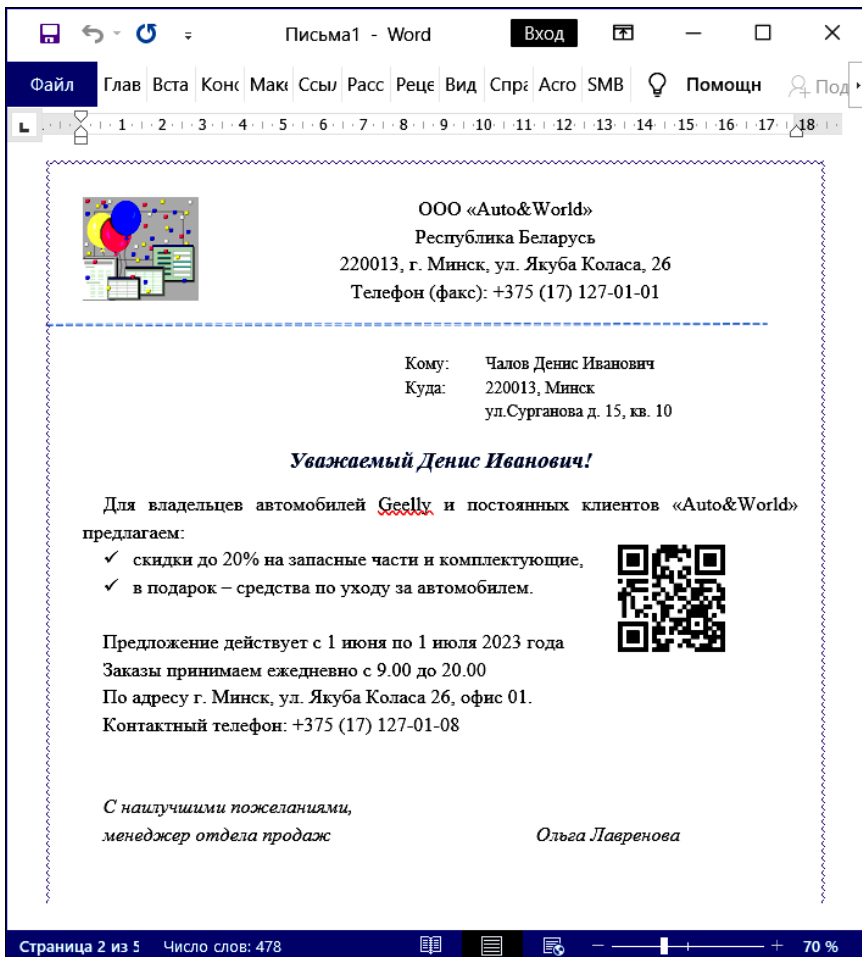


Рис. 5.8. Результат слияния – многостраничный итоговый документ (страница 2 из 5)

Таким образом, наличие возможностей экспорта данных их БД в приложения MS Office позволяет существенно расширить возможности анализа и визуализации данных, а процедура *Слияние в Word* обеспечивает автоматизацию процесса создания и рассылки типовых документов.

5.3. Настройка рабочей среды MS Access

Настройка параметров рабочей среды СУБД MS Access выполняется аналогично другим офисным приложениям с использованием команды *Параметры* в меню *Файл*.

В диалоговом окне «Параметры Access» на соответствующих вкладках выполняется настройка общих параметров рабочей среды СУБД, параметров загрузки текущей БД и окон документов, пользовательского интерфейса, включая вкладки модульной ленты, панель быстрого доступа и др.

Для настройки параметров загрузки текущей БД необходимо в диалоговом окне «Параметры Access» перейти на вкладку *Текущая база данных* (рис. 5.9) и задать значения для следующих элементов:

- *заголовок приложения* – ввести название, которое будет отображаться в строке заголовка текущей БД вместо имени файла;

- *значок приложения* – выбрать изображение, которое будет отображаться в строке заголовка текущей БД вместо стандартного логотипа СУБД;

- *форма просмотра* – выбрать из доступного списка форму, которая будет открываться при загрузке текущей базы данных.

При работе в режиме *Конструктора* следует изучить настройки параметров на вкладке *Конструкторы объектов* (рис. 5.10) такие как: тип данных, назначаемый для поля таблицы по умолчанию (короткий текст), размеры полей, назначаемые по умолчанию для текстовых (255 символов) и числовых (длинное целое) типов, а также параметры конструктора запросов, форм и отчетов, включая правила проверки ошибок при работе с формами и отчетами.

Для защиты БД от несанкционированного доступа в СУБД MS Access применяется шифрование с помощью пароля. Для этого необходимо открыть БД в монопольном режиме, далее в меню *Файл* выбрать команду *Сведения* и нажать кнопку «Зашифровать с использованием пароля», затем в диалоговом окне «Задание пароля базы данных» ввести надежный пароль, состоящий не менее чем из 8 знаков (букв в верхнем и нижнем регистре, цифр и символов). Следует учесть, что безопасность на уровне пользователя недоступна в файле БД формата accdb. Если пароль будет утерян, зашифрованную БД использовать невозможно.

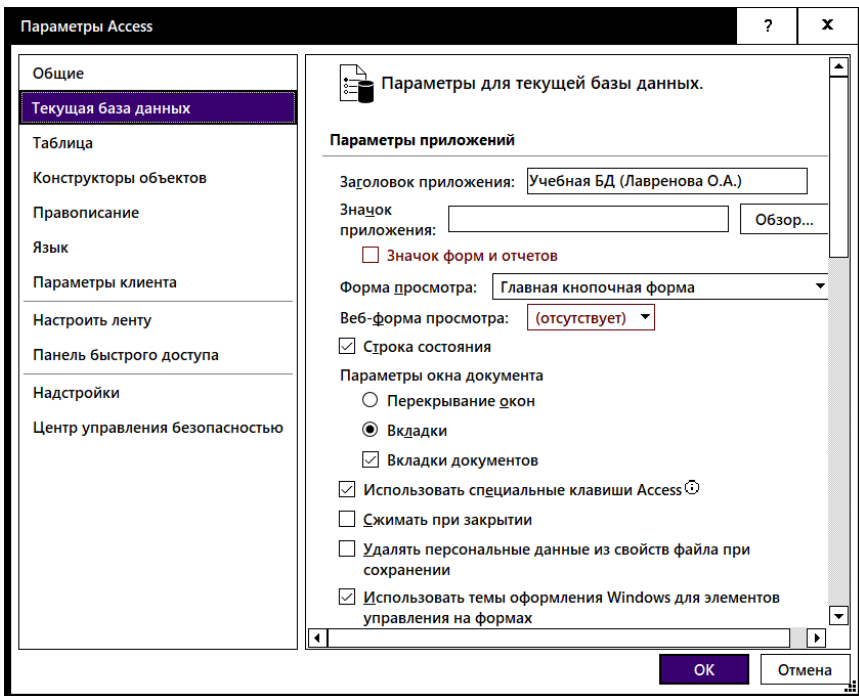


Рис. 5.9. Настройка параметров загрузки текущей БД

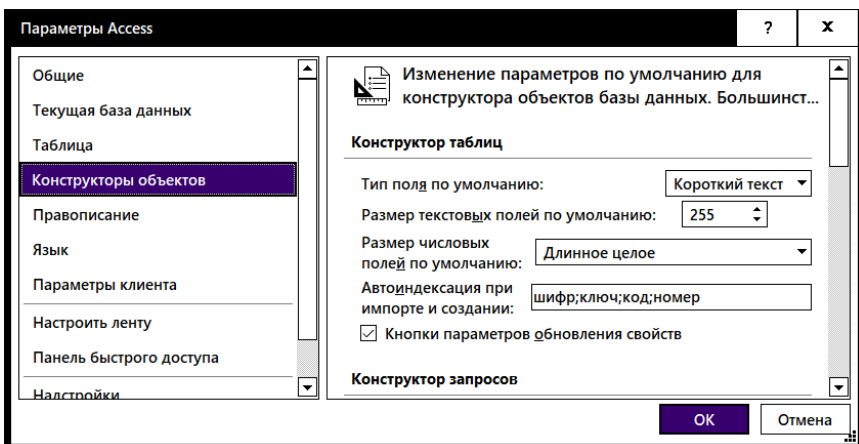


Рис. 5.10. Настройка параметров для режима конструкторов объектов

Возможности пользователя по настройке, управлению и обслуживанию БД расширяют встроенные в MS Access служебные программы, которые по функциональному признаку можно разделить на две группы: 1) для работы с объектами БД и 2) для работы с БД.

1. Команды для работы с объектами БД доступны на вкладках *Создание* и *Внешние данные*.

2. Команды для работы с БД доступны на вкладке *Работа с базами данных* (рис. 5.11) и обеспечивают:

- сжатие и восстановление БД;
- работу с отношениями (схемой данных);
- анализ БД (*Архивариус* – подготовка описания объектов БД, *Анализ быстрodeйствия*, анализ содержания таблиц) и др.

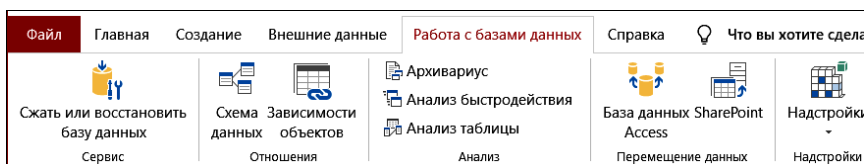


Рис. 5.11. Команды для работы с БД

Команда *Сжать или восстановить БД* используется, если необходимо физически сжать открытую БД: удалить из файла БД неиспользуемые участки, занятые, например, удаленными объектами, и уменьшить размер файла БД.

При работе с БД, особенно на этапе конструирования объектов, размер БД постоянно возрастает и даже при удалении неиспользуемых объектов не уменьшается. Это связано с тем, что объекты физически остаются в файле БД, т. к. их реальное удаление из файла и сокращение объема файла требуют физического перемещения всех объектов в файле БД. Обратите внимание, что параметр *Сжимать при открытии* по умолчанию не установлен (см. рис. 5.9).

Эта же команда *Сжать или восстановить БД* выполняет восстановление поврежденной БД при неустойчивой работе БД (например, при аварийном завершении работы с приложением).

Для выполнения операции необходимо:

- закрыть БД;
- вызвать команду *Сжать или восстановить БД*;

– выбрать исходный файл БД для операции сжатия или восстановления, задать имя файла назначения (можно задать одно и то же имя, но для безопасной работы рекомендуется ввести новое имя для БД после операции сжатия и восстановления).

При открытии поврежденной БД обычно выполняется автоматический запрос на ее восстановление.

В СУБД MS Access предусмотрена возможность сохранения БД как реплики – специальной копии БД, которая отличается от обычной копии файла тем, что позволяет выполнить синхронизацию изменений, внесенных пользователями при совместной работе с БД.

Вопросы для самопроверки

1. Назовите и кратко характеризуйте основные технологии взаимодействия СУБД MS Access с приложениями Microsoft Office.

2. Перечислите *форматы экспорта/импорта* объектов БД и способы конвертирования данных в форматы Microsoft Office.

3. Какие инструменты MS Excel используются для обработки и анализа данных, экспортированных из таблиц и запросов БД MS Access?

4. Сформулируйте назначение процедуры *Слияние в Word*. Какие объекты БД MS Access могут использоваться в качестве источника данных для слияния в Word?

5. Назовите документы, которые используются при слиянии в *Word*. В чем их особенности?

6. Опишите порядок создания серийных документов (шаги *Мастера слияния Word*).

7. Приведите приемы настройки рабочей среды MS Access. Каким образом настраиваются параметры запуска БД?

8. Каким образом в MS Access обеспечивается защита БД от несанкционированного доступа?

9. Какие встроенные программы MS Access расширяют возможности пользователи по управлению БД? На какие группы они делятся по функциональному признаку?

10. Каковы особенности применения команды *Сжать и восстановить базу данных*? Почему по умолчанию БД не сжимается автоматически при завершении работы с ней?

Тема 6. ОСНОВЫ ИСПОЛЬЗОВАНИЯ SQL

6.1. Понятие SQL, варианты реализации, достоинства и особенности.

6.2. Категории операторов SQL.

6.3. Основы синтаксиса SQL.

6.4. Типы данных SQL.

6.1. Понятие SQL, варианты реализации, достоинства и особенности

В условиях четвертой технологической революции, начало которой относят к 2011 году – к моменту появления немецкой инициативы «Индустрия 4.0», именно «данные стали критически важным сырьем XXI века. Не случайно данные назвали нефтью «Четвертой промышленной революции» [7].

В сфере бизнеса с развитием сети Интернет, электронной почты и почтовых рассылок, социальных сетей, внедрением CRM-систем и др. у компаний появляются широкие возможности добычи и обработки данных о потребителях, экономических и социальных процессах, анализ которых позволит выявлять актуальные закономерности, прогнозировать развитие событий в целях повышения эффективности работы компаний. Таким образом, аналитика данных является одним из технологических приоритетов современных компаний, а компетенции обработки данных, включая их извлечение, анализ и визуализацию, являются одними из критически важных для специалистов инженерно-экономического профиля.

Рынок труда формирует запросы на специалистов с компетенциями аналитиков баз данных, владеющих инструментами «больших данных» (BigData). Одним из таких востребованных инструментов является язык SQL, который позволяет эффективно решать задачи не только по созданию баз данных и управлению ими, но также позволяет строить запросы для поиска повторяющихся цепочек событий и выявления закономерностей. Навыки работы с SQL становятся обязательными для аналитиков, маркетологов, менеджеров и руководителей, а не только для разработчиков и тестировщиков БД. Поэтому требования к кандидатам в вакансиях, связанных с базами данных и анализом данных, включают обязательное знание SQL.

SQL (Structured Query Language, язык структурированных запросов) – язык программирования, предназначенный для управления данными в СУБД.

Основные этапы развития SQL:

Конец 1970-х годов – компания IBM предложила язык SQL.

1986 г. – первый вариант стандарта принят институтом ANSI.

1987 г. – вариант стандарта ANSI одобрен ISO.

1992 г. – принят новый (более полный) вариант SQL92, который заложил основу стабильности SQL, обеспечивший его использование во всех основных технологиях обработки данных. В последующие годы появлялись новые выпуски стандарта: 1999 (SQL: 1999), 2003 (SQL: 2003), 2008 (SQL: 2008) и т. д. С 2023 года действует стандарт ISO/IEC 9075-1:2023 [3].

Производители реляционных СУБД с различной степенью соответствия поддерживают стандарты SQL: Oracle, INGRES, Informix, SyBase, SQLBase, Microsoft SQL Server, DB2 (СУБД IBM), Interbase, Firebird. Язык SQL поддерживается и программными продуктами SQL/DC, Paradox, Access, Approach и многими другими.

С точки зрения технической реализации SQL является не самостоятельным языком, а скорее «подъязыком данных», который предназначен для организации взаимодействия с базой данных. SQL сам по себе не содержит средств, необходимых для разработки законченных программ, поэтому используется в виде одной из *трех прикладных реализаций*:

1. *Интерактивный или автономный SQL.*

Интерактивный SQL применяется для непосредственной работы с БД и дает возможность пользователям *извлекать* информацию из БД или *записывать* ее в БД. SQL-оператор выполняется непосредственно после ввода, и пользователь сразу видит результат его выполнения (или код ошибки).

Например, пользователь может запросить из БД список заказов за текущий месяц. В результате запроса информация из БД может быть выведена на экран, переадресована для записи в файл или отправлена для печати на принтер.

2. *Статический SQL.*

Статический SQL содержит операторы, которые непосредственно встроены в программы на других языках программирования. Статический SQL может быть *встроенным* и *модульным*.

Встроенный SQL – это код SQL, включенный в код исходного текста программы, написанной на другом языке, *например*, С или COBOL, однако, при обращении к БД используется именно SQL, а результаты выполнения SQL-операторов присваиваются переменным, определенным в базовой программе.

SQL-инструкции во встроенном SQL представляют собой заранее записанный, фиксированный, а не генерируемый во время выполнения программы код SQL, который обычно используется в приложениях.

Модульный SQL – в этом случае модули SQL скомпонованы с модулями кода других языков. Такой подход был принят в раннем стандарте SQL86, когда встроенный SQL не являлся официальной частью этого стандарта, хотя и был приведен в приложении к нему.

Стандарт SQL92 поддерживает обе версии статического SQL. Однако встроенный SQL пользуется большей популярностью среди разработчиков, чем модульный SQL.

3. *Динамический SQL.*

Динамический код SQL генерируется приложением непосредственно во время исполнения и его содержание зависит от того, какой выбор сделает пользователь во время выполнения приложения. Операторы динамического SQL часто генерируются такими средствами разработки, как, *например*, графические языки запросов.

Требования ко всем трем реализациям языка SQL различны, и это отражается на языковых конструкциях, используемых для каждой из них. Однако в большинстве случаев в применении языка нет принципиальных отличий для всех трех вариантов реализации SQL.

Широкое распространение SQL обусловлено рядом его существенных *достоинств*:

1. *Реляционная основа.* Табличная структура реляционной БД интуитивно понятна пользователям, способствует пониманию принципов работы SQL-запросов.

2. *Наличие стандартов.* Способствует применению языка SQL в различных системах.

3. *Независимость от конкретных СУБД.* Большинство SQL-запросов может быть перенесено из одной СУБД в другую, несмотря на существующие различия в синтаксисе диалектов языка.

4. *Интерактивные запросы.* SQL-запросы обеспечивают пользователям немедленный доступ к данным при выполнении специальных запросов.

5. *Различные представления данных.* С помощью SQL разработчики БД могут обеспечить различным категориям пользователей БД необходимые представления её структуры и содержимого.

6. *Динамическое определение данных.* SQL позволяет динамически изменять и расширять структуру БД даже во время работы пользователей с её содержимым.

7. *Язык для шлюзов БД.* В вычислительных сетях SQL часто используется в шлюзовой программе для обмена данными между разнородными СУБД.

8. *Переносимость с одной вычислительной системы на другую.* SQL-приложения, созданные для однопользовательских систем, по мере своего развития могут быть перенесены в более крупные.

9. *Поддержка архитектуры «клиент-сервер».* SQL используется для организации связи через локальную сеть с сервером БД, в которой хранятся совместно используемые данные, позволяет манипулировать БД, включая динамическое изменение и расширения её структуры, а также поддерживает возможность программного доступа к БД, в том числе и к распределенным БД.

Особенностями SQL являются:

1. *Несоответствие реляционной модели данных.* С точки зрения реляционной модели данных SQL не является истинно реляционным языком из-за наличия дубликатов, необязательности первичного ключа, возможности упорядочения результатов.

2. *Отклонения от стандартов.* Несмотря на наличие международного стандарта ANSI SQL92, многие компании, занимающиеся разработкой СУБД (Oracle, Sybase, Microsoft, MySQL AB), вносят собственные изменения в язык SQL, что приводит к появлению его диалектов.

3. *Сложность работы с иерархическими структурами.* заключается в том, что диалекты SQL ряда СУБД изначально не имели возможностей для манипуляции древовидными структурами.

4. *Сложность записи SQL-инструкций.* Для обычного пользователя БД, работающего в визуальной среде создания объектов БД, использование языка SQL может оказаться достаточно сложным, поэтому с SQL-запросами обычно работают программисты.

Следует отметить, что SQL является не классическим языком программирования, а *информационно-логическим языком*, который основывается на реляционной алгебре. В настоящее время это самый распространенный универсальный декларативный язык, применяемый для создания, модификации и управления данными в *реляционных базах данных*.

Примеры СУБД с поддержкой SQL: MySQL, Microsoft SQL Server, PostgreSQL и др.

6.2. Категории операторов SQL

Как правило, SQL используется в работе с большими базами данных, позволяет быстро извлекать данные, отфильтровывая их по множеству параметров. SQL позволяет выполнять все действия над данными: начиная от создания структуры БД, записи и чтения данных, до администрирования сервера СУБД.

Язык SQL содержит примерно 40 операторов, которые формируют шесть категорий в зависимости от выполняемых действий [16] (табл. 6.1).

Таблица 6.1

Категории операторов SQL

Категория	Назначение операторов
1	2
DDL	<i>Data Definition Language</i> , язык определения данных Операторы DDL: CREATE, DROP и ALTER, позволяют пользователю определять новые объекты БД, уничтожать и изменять их
DML	<i>Data Manipulation Language</i> , язык манипулирования данными Операторы DML: INSERT, DELETE и UPDATE, позволяют добавлять строки, удалять строки в таблице, а также непосредственно модифицировать данные
DCL	<i>Data Control Language</i> язык управления данными Операторы DCL: GRANT и REVOKE применяются для выполнения административных функций по присваиванию или отмене привилегий пользователю при работе с объектами БД

1	2
DRL	<i>Data Retrieval Language</i> , язык выборки данных Выборку данных из таблиц БД обеспечивает один оператор DRL – SELECT
TCL	<i>Transaction Control Language</i> , язык управления транзакциями Операторы TCL: COMMIT, ROLLBACK, SET TRANSACTION применяются для управления изменениями, выполненными группой операторов DML
CCL	<i>Cursor Control Language</i> язык управления курсором и др. Операторы CCL используются для управления курсором при выполнении запроса, для подготовки и выполнения операторов SQL, описания плана запроса и т. д.

Следует отметить, что непосредственно для работы с данными используются 4 группы операторов: DDL – операторы определения данных, DML – операторы манипуляции данными, DCL – операторы определения доступа к данным и DRL – операторы для выборки данных (SELECT), описание которых приведено в пп. 6.2.1–6.2.4.

6.2.1. Операторы определения данных DDL (*Data Definition Language*).

Операторы DDL позволяют пользователю определять новые объекты (*например*, таблицы и связанные с ними элементы, индексы, представления и др.), которые будут созданы в БД, удалять их, *как правило*, безвозвратно и изменять по своему усмотрению структуру существующего объекта, *например*, добавить столбцы в существующую таблицу:

CREATE TABLE – добавляет новую таблицу к БД;

DROP TABLE – удаляет таблицу из БД;

ALTER TABLE – изменяет структуру существующей таблицы;

CREATE VIEW – добавляет новое представление к БД;

DROP VIEW – удаляет представление из БД;

CREATE INDEX – создает индекс для поля;

DROP INDEX – удаляет существующий индекс для поля;

CREATE SCHEMA – создает новую схему в БД;

DROP SCHEMA – удаляет схему из БД;

CREATE DOMAIN – создает новый домен;
ALTER DOMAIN – переопределяет домен;
DROP DOMAIN – удаляет домен из БД.

Пример 6.1. SQL-оператор создает таблицу *Users* из 4-х столбцов *idUser*, *name*, *country*, *email*.

```
CREATE TABLE Users (  
    idUser INTEGER,  
    name VARCHAR(120),  
    country VARCHAR(50),  
    email VARCHAR(100)  
);
```

Большинство коммерческих баз данных SQL имеют собственные расширения в категории DDL, которые позволяют осуществлять контроль над нестандартными, но важными элементами конкретной системы.

6.2.2. Операторы манипуляции данными DML (Data Manipulation Language).

Эта категория содержит три оператора, которые позволяют пользователю выполнять изменения *данных* в таблицах:

INSERT – добавляет новые строки в таблицу;
DELETE – удаляет строки из таблицы;
UPDATE – изменяет данные в таблице.

Пример 6.2. SQL-оператор удаляет из таблицы *Users* записи со значением 13 в поле *idUser*.

```
DELETE FROM Users WHERE idUser = 13;
```

6.2.3. Операторы определения доступа к данным DCL (Data Control Language).

Операторы этой категории применяются для выполнения административных функций по назначению или отмене права (привилегии) на использование БД или таблицы БД, а также на выполнение отдельных операторов SQL:

GRANT – выполняет назначение привилегии;
REVOKE – отменяет привилегии.

Пример 6.3. SQL-оператор назначает разрешение каждому пользователю просматривать таблицу *Users*.

```
GRANT SELECT ON Users TO PUBLIC;
```

6.2.4. Оператор выборки данных DRL (Data Retrieval Language).

Эта категория содержит один оператор SELECT (*англ.* – *выбирать*), который обеспечивает *выборку* данных из таблиц БД.

Этот оператор является самым часто используемым при построении SQL-запросов [5, 6].

В простейшем виде, запись оператора SELECT имеет вид¹:

```
SELECT { * | <столбец1> [, <столбец2> ... ] }  
FROM <таблица1> [, <таблица2> ... ];
```

где SELECT – ключевое слово, после которого приводится либо * (означает выбрать все столбцы), либо список столбцов результирующего набора данных, возвращаемого оператором;

FROM – ключевое поле, после которого приводится имя таблицы (или список имен таблиц), из которых будет происходить выборка данных.

Пример 6.4. SQL-оператор выбирает все столбцы (используется «*») из таблицы *Users*.

```
SELECT * FROM Users;
```

Пример 6.5. SQL-оператор выбирает столбцы *name* и *email* из таблицы *Users*.

```
SELECT name, email  
FROM Users;
```

Расширенная запись оператора SELECT включает ряд дополнительных инструкций, указывающих условия для фильтрации данных, их группировки, упорядочивания и т. д.:

¹ При записи оператора использованы служебные символы:
{ ... | ... } – для обозначения вариантов альтернативного выбора;
<...> – для обозначения названий полей или таблиц;
[...] – для записи необязательных элементов (внутри скобок).

```

SELECT [DISTINCT | ALL] {*} | <Список_полей_выбора>
FROM <Список таблиц>
[WHERE <Условие отбора>]
[GROUP BY <Список полей для группировки> [ASC | DESC], ...]
[HAVING <Условие фильтрации данных для группировки>]
[ORDER BY <Список полей для сортировки> [ASC | DESC], ...]
[LIMIT <Число строк>];

```

DISTINCT используется для указания на то, что следует выбрать только уникальные значения в поле, ALL – все (по умолчанию).

WHERE используется для *фильтрации данных* в соответствии с *условием отбора*, которое записывается с помощью операторов сравнения, сравнения с образцом (LIKE, BETWEEN, IS, IN), логических операторов (AND, OR, NOT), ключевого слова NULL.

Пример 6.6. SQL-оператор выбирает из таблицы *Users* данные о пользователях из Беларуси.

```

SELECT *
FROM Users
WHERE country = 'Беларусь';

```

Пример 6.7. SQL-оператор выбирает из таблицы *Users* данные о всех пользователях по имени Ольга из Беларуси.

```

SELECT *
FROM Users
WHERE name='Ольга' AND country='Беларусь';

```

GROUP BY используется для группировки результата по столбцу или по нескольким столбцам. С GROUP BY связано использование агрегатных функций (возвращающих значение по набору значений поля: SUM, COUNT, MAX, MIN, AVG и др.) и HAVING.

Пример 6.8. SQL-оператор позволит определить по данным таблицы *Users* число пользователей из каждой страны.

```

SELECT country, COUNT(idUser) AS CountUsers
FROM Users
GROUP BY country;

```

HAVING используется в запросе аналогично условию WHERE, с той особенностью, что применяется к выходному набору, создан-

ному с помощью GROUP BY, и задает условия для фильтрации результата агрегатных функций.

Пример 6.9. SQL-оператор позволит определить по данным таблицы *Users* страны, в которых число пользователей более 1000.

```
SELECT country, COUNT(idUser)
FROM Users
GROUP BY country
HAVING COUNT(idUser) > 1000;
```

ORDER BY используется для сортировки выходного набора значений в выбранном направлении: ASC – по возрастанию, от меньших значений к большим (по умолчанию); DESC – по убыванию, от больших значений к меньшим. Сортировка может применяться как к числовым, так и строковым значениям, в этом случае упорядочивание происходит по алфавиту.

LIMIT n – записывается всегда в конце запроса и позволяет вывести из таблицы указанное число строк n, начиная с 1-й (или диапазон строк, например, с 5-й по 10-ю: LIMIT 5, 10).

Пример 6.10. SQL-оператор позволит определить по данным таблицы *Users* страны, где число пользователей более 1000, упорядочить список по возрастанию числа пользователей и вывести первых 5 строк.

```
SELECT country, COUNT(idUser) AS CountUsers
FROM Users
GROUP BY country
HAVING COUNT(idUser) > 1000
ORDER BY CountUsers ASC
LIMIT 5;
```

При записи оператора SELECT во избежание ошибок и некорректной обработки данных следует соблюдать установленный порядок использования ключевых слов:

1. SELECT (обязательное).
2. FROM (обязательное).
3. WHERE.
4. GROUP BY.

5. HAVING.
6. ORDER BY.
7. LIMIT.

6.3. Основы синтаксиса SQL

Для записи SQL-инструкций применяются следующие базовые элементы:

- зарезервированные ключевые слова (*например*, SELECT, FROM, WHERE);
- идентификаторы (имена) таблиц (*например*, Users) и столбцов таблиц (*например*, idUsers);
- идентификаторы (имена) операций и функций, используемых в выражениях (*например*, AND, OR, AVG, SUM);
- логические, арифметические и строковые выражения для формирования критериев поиска информации в БД, а также для вычисления значений полей для результирующих таблиц.

Правила записи идентификаторов:

- для имен таблиц и столбцов допустимо использовать буквы, цифры и знак «_» (подчеркивание), но *первым символом* имени обязательно должна быть *буква*;
- ПРОПИСНЫЕ и строчные буквы не различаются: SELECT = Select = select;
- для идентификаторов таблиц и имен столбцов нельзя использовать ключевые слова и имена функций SQL.

Правила построения SQL-операторов:

- оператор начинается с ключевого слова – глагола (*например*, CREATE, SELECT и т. п.) и заканчивается знаком «;» (точка с запятой);
- ключевые слова, *как правило*, записываются прописными буквами, а все остальные (*например*, имена таблиц и столбцов) – строчными;
- предложения, *как правило*, записываются на отдельных строках с использованием отступов для лучшего прочтения и удобства редактирования;
- ключевые слова нельзя сокращать и переносить с одной строки на другую.

6.4. Типы данных SQL

В языке SQL поддерживаются типы данных для обработки числовых, текстовых значений, значений типа дата/время, массивов (табл. 6.2).

Таблица 6.2

Типы данных SQL

Тип	Описание	Размер, байт
INTEGER	Длинное целое число в диапазоне от -2 147 483 648 до 2 147 483 647	4
SMALLINT	Короткое целое число в диапазоне от -32 768 до 32 767	2
FLOAT [(<i>len</i> , <i>dec</i>)]	Значение с плавающей запятой двойной точности в диапазоне от -1,79769313486232E308 до -4,94065645841247E-324 для отрицательных значений, от 4,94065645841247E-324 до 1,79769313486232E308 для положительных значений, а также 0, представляемое при выводе максимальным числом знаков, заданным аргументом <i>len</i> , и числом знаков после десятичной запятой, заданным аргументом <i>dec</i>	8
CHAR(<i>size</i>)	Строка символов фиксированной длины, Значение от 0 до 255 символов размер которой (в символах) определяется значением аргумента <i>size</i>	2 байта на символ
VARCHAR(<i>size</i>)	Строка символов переменной длины, максимальный размер которой (в символах) не может превышать значения, заданного аргументом <i>size</i>	
BINARY	Массив произвольных (двоичных) байтов (максимальный размер зависит от реализации, обычно это 65535 байт). Этот тип данных используется, например, для хранения изображений	1 байт на символ
DATETIME	Значение даты и времени между годами от 100 до 9999	8

Вопросы для самопроверки

1. Что понимают под языком SQL? Дайте общую характеристику SQL, назовите его достоинства и особенности.
2. В связи с чем на рынке труда возрастают запросы на специалистов, владеющих SQL? Какие специалисты в настоящее время должны иметь навыки работы с SQL? И почему?
3. Приведите примеры СУБД с поддержкой SQL. Какие существуют стандарты языка SQL?
4. На какие категории подразделяются операторы SQL? Какие из них используются для обработки данных?
5. Какие операторы SQL относятся к категории DDL? С помощью какого оператора SQL создаются таблицы? Приведите пример.
6. Какие операторы SQL относятся к категории DML? Опишите их назначение и приведите примеры.
7. К какой категории операторов относится оператор SELECT? Опишите его назначение и синтаксис.
8. Опишите алгоритм выполнения оператора SELECT. Как выполняется формирование выходного набора данных с помощью SELECT? Приведите примеры.
9. Правила формирования условия отбора при записи оператора SELECT. Приведите примеры записи WHERE с использованием логических операторов и операторов сравнения.
10. Как выполняется группировка данных в SQL? Какие агрегатные функции используются для получения сводной информации? Приведите примеры.
11. Каким образом выполняется упорядочение выходного набора данных при записи оператора SELECT? Как можно ограничить число строк в выходном наборе? Приведите примеры.
12. Какие типы данных используются в языке SQL (приведите примеры).

2. ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа № 1

ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ С РЕЛЯЦИОННОЙ СУБД MS ACCESS



Цель работы: – изучить назначение и интерфейс реляционной СУБД MS Access;
– приобрести практические навыки работы с объектами реляционной БД.



Содержание отчета

1. Название и цель работы.
2. Формулировка заданий и результаты их выполнения (включая распечатки окон БД).
3. Список контрольных вопросов и ответов на них.
4. Выводы.
5. Список использованных источников.



Контрольные вопросы

1. Сформулируйте основные понятия теории баз данных: «база данных», «реляционная база данных», «таблица», «запись», «поле».
2. Дайте определение и краткую характеристику основных моделей данных. Назовите достоинства и недостатки реляционной модели.
3. Сформулируйте понятие СУБД, перечислите функции (основные, дополнительные). Приведите примеры современных СУБД.
4. Назовите элементы интерфейса реляционной СУБД MS Access (описание команд меню *Файл* и вкладки *Главная*).
5. Определите состав и назначение объектов реляционной СУБД MS Access (таблицы, запросы, формы, отчеты, макросы, модули).
6. Для какой предметной области разработана учебная БД *Борей*? Опишите схему БД и укажите состав объектов.
7. Опишите основные приемы работы с таблицами в реляционной СУБД MS Access: добавление, удаление, модификация записей

в таблице; закрепление столбцов таблицы, их сокрытие/отображение, изменение ширины столбцов/высоты строк.

8. Сформулируйте назначение режимов *Таблицы* и *Конструктора таблиц*, приведите способы переключения между ними.



Порядок выполнения работы

1. Загрузите офисную систему управления реляционными базами данных MS Access. Изучите элементы интерфейса, состав и назначение меню *Файл*, вкладок *Главная*, *Создание* и *Работа с базами данных* (результат приведите в отчете).

2. Откройте учебную БД *Борей*, прочитайте текст ее заставки и определите назначение БД *Борей* – (сформулируйте в отчете).

3. Отобразите в *Области навигации* вкладку с типом объектов *Таблицы*, используя КЗМ для строки заголовка *Области навигации*, выберите команду *Просмотр*, далее команду *Сведения*, рис. 1.1.

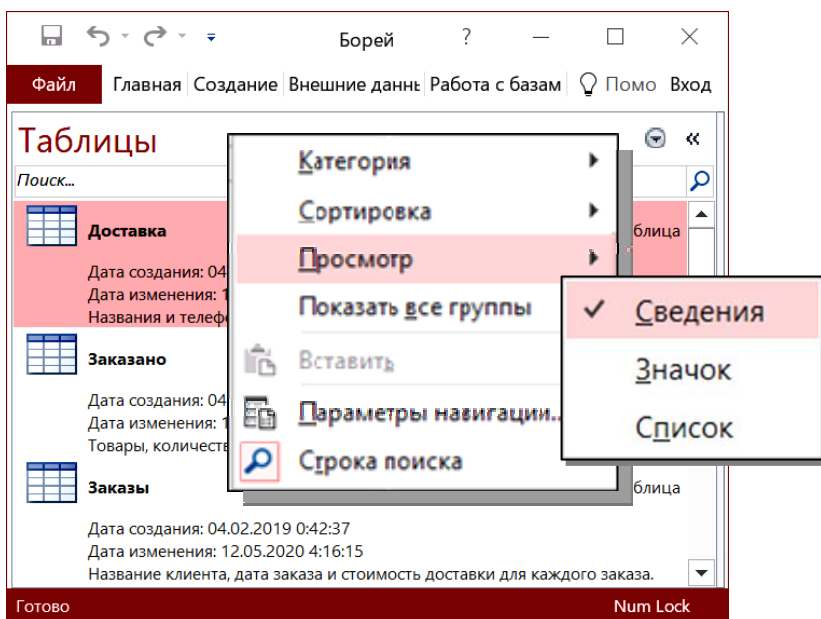


Рис. 1.1. Представление таблиц БД *Борей* на вкладке *Таблицы* в режиме просмотра сведений

4. Изучите количественный состав объектов БД *Борей*, их содержание и назначение. Заполните табл. 1.1.

Таблица 1.1

Состав и назначение объектов БД *Борей*

Объекты БД	Число в БД	Примеры объектов (открыть и изучить)	Назначение
Таблицы		Сотрудники, Клиенты, Типы, Товары	<i>Таблицы</i> – объекты СУБД MS Access, предназначенные для ...
Запросы		Десять самых дорогих товаров, Продажи по типам, Товары по типам	<i>Запросы</i> – объекты СУБД MS Access, предназначенные для ...
Формы		Главная кнопочная форма, Сотрудники, Телефоны клиентов, Типы, Товары	<i>Формы</i> – объекты СУБД MS Access, предназначенные для ...
Отчеты		Каталог, Наклейки для клиентов, Список товаров	<i>Отчеты</i> – объекты СУБД MS Access, предназначенные для ...
Макросы		<i>Открыть в Конструкторе:</i> Клиенты, Поставщики, Телефоны клиентов (и форму)	<i>Макросы</i> – объекты СУБД MS Access, предназначенные для ...
Модули		Заставка	<i>Модули</i> – объекты СУБД MS Access, предназначенные для ...

5. Отобразите *схему* БД *Борей* (рис. 1.2) с помощью кнопки «Схема данных» в группе команд *Отношения* на вкладке *Работа с базами данных*.

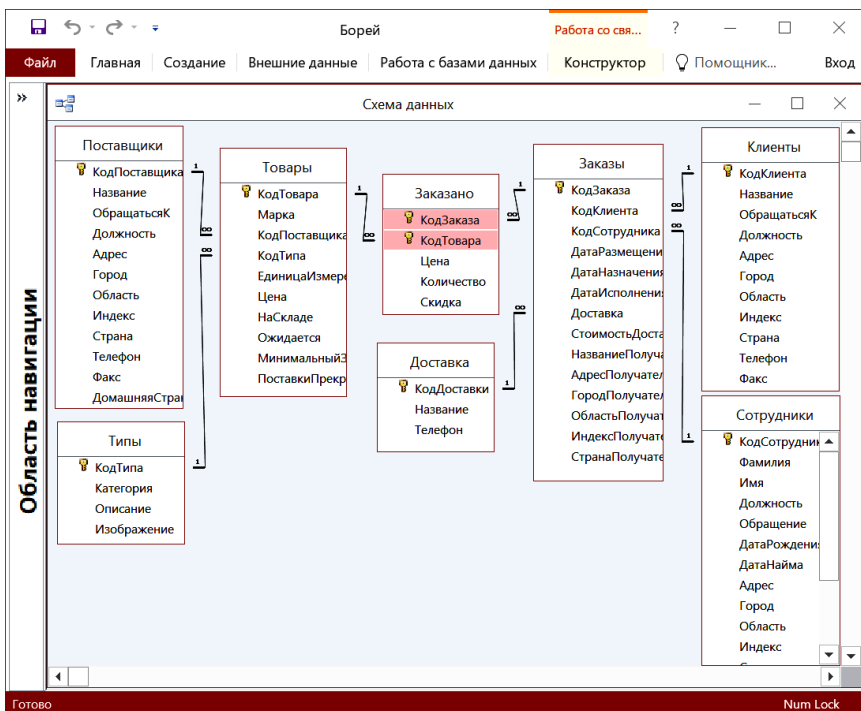


Рис. 1.2. Схема учебной БД *Борей*

6. Изучите схему БД *Борей* и определите взаимосвязи таблиц:

а) перечислите *главные* таблицы – _____;

б) перечислите *подчиненные* таблицы – _____.

В какой из таблиц находится *составной первичный* ключ? _____

7. Откройте таблицу *Сотрудники* в режиме *Таблицы* и определите *число записей* в таблице (приведите в отчете) – _____.

8. В режиме *Таблицы* добавьте в таблицу *Сотрудники* *новую запись* (с персональными данными); закройте таблицу после ввода данных *без операции ее сохранения*. Повторно откройте таблицу *Сотрудники* в режиме *Таблицы* и проверьте наличие ранее введенных персональных данных: _____.

9. В таблице *Сотрудники* (в режиме *Таблицы*) с помощью КЗМ последовательно скройте/отобразите поля *Должность* и *Обращение*.

10. В таблице *Сотрудники* (в режиме *Таблицы*) закрепите поле *Фамилия*.

11. Используя горизонтальную полосу прокрутки или клавишу *Tab*, выведите на экран поля *Примечание*, *Подчиняется*. Какое поле находится у левого края окна таблицы? _____.

12. Перейдите к первому столбцу таблицы *Сотрудники*. Отмените закрепление поля *Фамилия*.

13. Переместите поля таблицы *Фамилия* и *Код сотрудника*, используя технологию *drag-and-drop* (поменяйте местами поля).

14. С помощью ЛКМ измените *ширину* полей и *высоту* строк таблицы *Сотрудники*. Закройте таблицу *Сотрудники* без сохранения изменений макета.

15. Сформулируйте назначение режима *Таблицы* _____

16. Откройте таблицу *Сотрудники* в режиме *Конструктора*, изучите макет таблицы и определите *число полей* в таблице _____, в том числе *ключевых* _____ и *описательных* _____.

17. Сформулируйте *назначение* режима *Конструктора* таблиц: _____

18. Найдите не менее 3-х способов переключения между режимами *Таблицы* и *Конструктора* (приведите в отчете).

19. Завершите работу с СУБД MS Access.

20. Оформите отчет о выполнении работы.

Лабораторная работа № 2

ОСВОЕНИЕ ЭТАПОВ СОЗДАНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ



Цель работы: – изучить этапы создания реляционной БД;
– практически освоить создание таблиц реляционной БД средствами СУБД MS Access.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи.
3. Результаты выполнения заданий (включая макеты таблиц в режиме *Конструктора*, примеры заполнения таблиц в режиме *Таблицы* и схему БД).
4. Список контрольных вопросов и ответов на них.
5. Выводы.
6. Список использованных источников.



Контрольные вопросы

1. Перечислите основные этапы создания реляционной БД и поясните их содержание. Укажите источники загрузки БД, приведите примеры первичных документов.
2. Сформулируйте понятия «предметная область», «информационный объект», приведите примеры.
3. Что понимают под информационно-логической моделью предметной области? Опишите порядок ее формирования.
4. Перечислите виды отношений между информационными объектами? Назовите виды отношений, которые непосредственно не поддерживаются в реляционной БД? Какие отношения поддерживаются в реляционной БД?
5. Сформулируйте понятие «нормализация данных» и приведите требования нормализации при третьей нормальной форме реляционной модели. Приведите практические приемы нормализации.
6. Какие требования предъявляют к содержанию таблиц реляционной БД?

7. Какие виды полей различают в таблице реляционной БД? (отобразите в виде схемы, приведите примеры).

8. Что такое ключевое поле? Назовите виды ключевых полей. Каким образом задается первичный ключ в таблице?



Порядок выполнения работы

1. Постановка задачи.

Средствами реляционной СУБД Microsoft Office Access разработать БД для заданной предметной области.

Предметная область: деятельность салона по продаже автомобилей.

Входная информация: первичные документы предметной области – источники загрузки базы данных: каталог автомобилей, договор купли-продажи или бланк заказа.

Выходная информация: таблицы-отношения *Клиенты*, *Автомобили*, (дополнительно может быть предусмотрена и таблица *Сотрудники*), *Заказы* (другие виды объектов БД в данной лабораторной работе создаваться не будут).

Количественные ограничения: БД должна содержать информацию о 3 моделях автомобилей, о 10 клиентах, каждый клиент формирует не менее 3 заказов, (количество сотрудников не менее двух).

2. Предварительная подготовка. Анализ предметной области.

Перед началом работы по созданию базы данных необходимо:

- изучить первичные документы предметной области;
- выполнить описание выбранной предметной области, включая реальные объекты и процессы;
- определить потребности в обработке, анализе и отображении данных;
- подготовить всю необходимую информацию для удовлетворения предполагаемых запросов пользователя.

Например, при традиционной организации деятельность салона по продаже автомобилей находит свое отражение в таких первичных документах, как:

- каталог автомобилей, содержащий все необходимые характеристики модели автомобиля, фото образца модели;

– договор купли-продажи (бланк заказа), который содержит всю информацию о заключенной сделке, сведения о клиенте и сотруднике.

Ежемесячно руководство автосалона требует предоставления отчетов о продажах, сроках выполнения заказов, о своевременности оплаты заказов и т. д. Руководителю также может быть необходима сводная информация по результатам работы сотрудников салона в течение месяца, квартала, года.

Ручная обработка всей информации без использования вычислительной техники не может выполняться оперативно, требует значительных временных и трудовых затрат. Использование базы данных «Автосалон» позволит оперативно накапливать, систематизировать и эффективно обрабатывать информацию по различным срезам, формировать итоговые отчеты, выявлять слабые звенья в системе продаж.

3. Проектирование БД «Автосалон».

Укрупнено, процесс создания реляционной базы данных состоит из 4-х этапов (см. рис. 1.7 в п. 1.4 теоретического раздела пособия):

1 этап. Определение цели создания базы данных.

2 этап. Выделение информационных объектов предметной области.

3 этап. Определение логической структуры БД.

4 этап. Создание и заполнение объектов базы данных средствами конкретной СУБД.

В соответствии с постановкой задачи практическое освоение этапов проектирования реляционной БД будет выполняться на примере БД «Автосалон» для предметной области «Продажа автомобилей».

1-й этап. *Определение цели создания БД.*

На первом этапе проектирования БД определяют: цель создания БД, ее функции и примерный перечень информации (по результатам анализа первичных документов).

Целью создания БД «Автосалон» является *автоматизация* учета информации по продаже автомобилей.

Функции БД «Автосалон»:

– *хранение* информации о клиентах автосалона, о продаваемых автомобилях, сотрудниках и сформированных заказах;

– *обновление* и *добавление* информации (о клиентах, автомобилях, сотрудниках и заказах);

– анализ информации по различным срезам (клиент, автомобиль, сотрудник, заказ),

– выдача итоговой информации в виде отчетов по различным срезам.

На основе анализа предметной области выделяют *примерный перечень информации* проектируемой БД:

– сведения о клиенте автосалона,

– технические и ценовые характеристики автомобиля (модель, мощность двигателя, количество дверей, цвет кузова, обивка салона, дополнительное оснащение, цена автомобиля, транспортные и предпродажные издержки и т. п.),

– сведения о заказе (№ и дата заказа, скидка, сведения об оплате и т. п.) и сотруднике, который оформил заказ.

2-й этап. Выделение информационных объектов предметной области.

На втором этапе проектирования БД составляют описание предметной области в виде реквизитов, извлекаемых из первичных документов – источников загрузки БД:

– фамилия, имя, отчество клиента автосалона;

– адрес и телефон клиента;

– паспортные данные клиента;

– марка и модель автомобиля;

– мощность двигателя;

– тип коробки передач;

– количество мест в салоне;

– цвет кузова;

– обивка салона;

– дополнительное оснащение автомобиля;

– заводская цена автомобиля;

– издержки (транспортные и предпродажные);

– дата формирования заказа;

– фамилия и должность сотрудника, оформившего заказ;

– предоставляемые скидки, % и др.

Функциональный анализ реквизитов позволяет выделить следующие информационные объекты: *Клиент, Автомобиль и Сотрудник* (рис. 2.2).

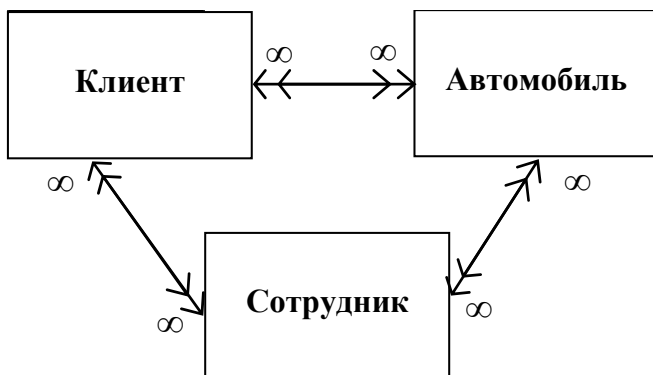


Рис. 2.2. Информационные объекты предметной области «Продажа автомобилей»

3-й этап. Определение логической структуры БД.

Реальные отношения между выделенными информационными объектами являются отношениями «многие-ко-многим», которые непосредственно не поддерживаются реляционными СУБД. Поэтому реальные отношения «многие-ко-многим» трансформируются в отношения «один-ко-многим» путем ввода объекта-связки *Заказ*.

Для установления связей каждому объекту назначается ключ (ключевое поле). Причем уникальные ключи объектов *Клиент*, *Автомобиль* и *Сотрудник* должны присутствовать как внешние ключи в объекте *Заказ* (рис. 2.3).



Рис. 2.3. Отношения 1 : ∞ между информационными объектами

Таким образом, между объектами *Клиент–Заказ*, *Автомобиль–Заказ* и *Сотрудник–Заказ* устанавливаются *одно-многозначные отношения* ($1 : \infty$), которые поддерживаются реляционной СУБД.

В соответствии с понятиями реляционной СУБД каждому *информационному объекту* в проектируемой БД будет соответствовать *таблица* (*Клиенты*, *Автомобили*, *Сотрудники*, *Заказы*).

Связь между таблицами устанавливается с помощью ключей *Код клиента*, *Код модели*, *Код сотрудника*, которые в главных таблицах *Клиенты*, *Автомобили* и *Сотрудники* будут являться первичными, а в таблице-связке *Заказы* – внешними.

На основе функционального анализа реквизитов предметной области определяется примерный состав и структура полей базовых таблиц (табл. 2.1). При этом учитываются основные *требования к содержанию таблиц реляционной БД*:

1. Каждая таблица содержит информацию только на одну тему.
2. Каждое поле должно быть связано с темой таблицы.
3. Информация в таблицах не дублируется.
4. Для связи между таблицами задаются *первичные ключи*, однозначно определяющие каждую запись в таблице.
5. В таблицу *не включаются поля*, значения в которых представляют собой готовый *результат вычислений*, а не вычисляемое выражение.
6. Информацию *разбивают на наименьшие логические единицы* (например, для описания места жительства задаются поля *Индекс*, *Населенный пункт*, *Почтовый адрес*, а не одно поле *Адрес*).

Таблица 2.1

Примерный состав и структура данных базовых таблиц БД

Таблица Клиенты	Таблица Автомобили	Таблица Сотрудники	Таблица Заказы
1	2	3	4
Код клиента (ключ)	Код авто (ключ)	Код сотрудника (ключ)	Код заказа (ключ)
Фамилия	Модель	Фамилия	Код клиента
Имя	Мощность двигателя	Имя	Код авто
Отчество	Цвет	Отчество	Код сотрудника

1	2	3	4
Индекс	Количество мест	Должность	Дата заказа
Страна	Заводская цена	Телефон	Скидка, %
Город	Издержки		Оплачено
Адрес	Спец. модель		
Телефон	Доп. оснащение		

При разработке таблиц реляционной БД учитываются и *требования нормализации*, с целью минимизации повторяющихся данных и обеспечения их целостности [21].

В результате выполнения теоретических этапов 1–3 получают проект логической структуры БД «Автосалон» (рис. 2.4), соответствующей информационно-логической модели предметной области «Продажа автомобилей».

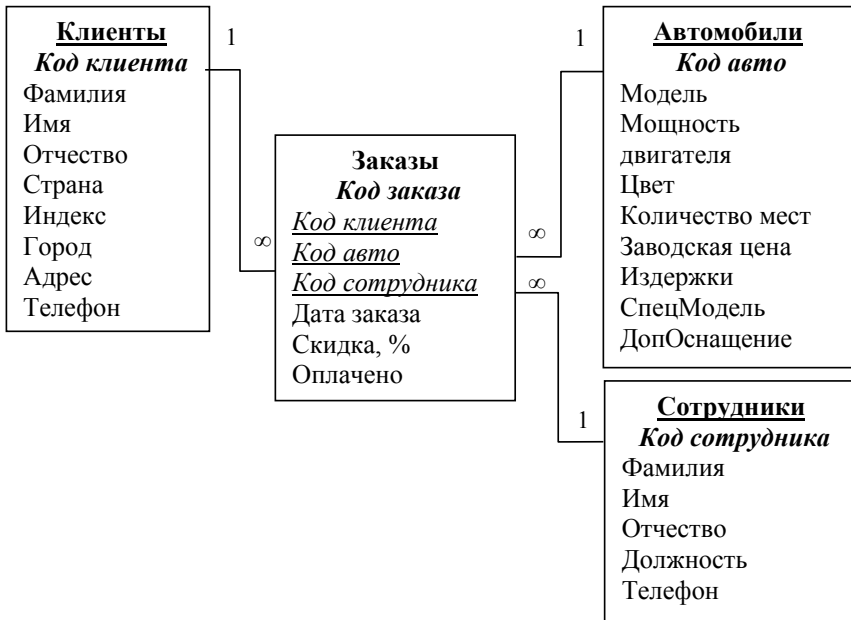


Рис. 2.4. Информационно-логическая модель предметной области – проект логической структуры БД «Автосалон»

4. Реализация БД «Автосалон».

4-й этап. Создание таблиц БД средствами СУБД MS Access.

4.1. Загрузите СУБД MS Access.

4.2. Создайте файл новой БД (*Файл/Создать/Новая БД...*): имя файла – *Фамилия_авто.accdb*, выберите расположение в *личной папке* и нажмите кнопку «Создать».

4.3. На вкладке *Создание* в группе команд *Таблицы* выберите команду *Конструктор таблиц* и на основании данных табл. 2.2 постройте макет таблицы *Автомобили*.

В табл. 2.2 приведено примерное описание макета таблицы *Автомобили*: имена полей, типы данных полей, свойства полей (определяют правила сохранения, отображения и обработки данных в поле), описания полей (необязательные для заполнения).

Внимание: все поля, за исключением поля *Другое оснащение*, должны быть *обязательными* для заполнения. Для этого при заполнении свойств полей в режиме *Конструктора* установите для свойства *Обязательное поле* значение «Да».

Таблица 2.2

Примерное содержание макета таблицы *Автомобили*

Имя поля	Тип данных	Описание	Свойства поля
1	2	3	4
<i>Код авто</i>	Числовой	<i>Ключевое поле</i> , код автомобиля в каталоге	Индексированное поле: Да/Совпадения не допускаются Ключевое поле задается в <i>контекстном меню</i> или на вкладке <i>Конструктор</i>
<i>Модель</i>	Короткий текст (далее – текстовый)	<i>Модель автомобиля марки Geely</i>	Размер поля: 20 Значение по умолчанию: Geely Atlas Pro Индексированное поле: Да/Совпадения допускаются
<i>Комплектация</i>	Текстовый	<i>Вариант комплектации</i>	Размер поля: 20

Окончание табл. 2.2

1	2	3	4
<i>Мощность двигателя</i>	Текстовый	Мощность двигателя (кВт/л. с.)	Размер поля: 10 Значение по умолчанию: 130/177
<i>Цвет</i>	Текстовый <i>*Мастер подстановок</i> , фиксированный набор значений	Цвет кузова Белый, Красный, Серебристый, Синий, Черный	Размер поля: 20 Значение по умолчанию: Белый
<i>Количество мест</i>	Числовой	Количество мест в салоне	Размер поля: Байт Формат: Основной Значение по умолчанию: 4
<i>Обивка</i>	Текстовый <i>*Мастер подстановок</i> , фиксированный набор значений	Велюр, Кожа, Ткань	Размер поля: 10 Значение по умолчанию: Велюр
<i>Коробка передач</i>	Текстовый <i>*Мастер подстановок</i> , фиксированный набор значений	6АКПП, 7DCT	Размер поля: 14 Значение по умолчанию: 6АКПП Правило проверки: 6АКПП Or 7DCT Сообщение об ошибке: Допустимы только значения 6АКПП или 7DCT
<i>Заводская цена</i>	Денежный	Заводская нетто-цена	Формат: Денежный Число десятичных знаков: Авто
<i>Транспортные издержки</i>	Денежный	Издержки на доставку	Формат: Денежный Число десятичных знаков: Авто Значение по умолчанию: 120
<i>Предпродажные издержки</i>	Денежный	Издержки на предпродажную подготовку	Формат: Денежный Число десятичных знаков: Авто Значение по умолчанию: 105

1	2	3	4
<i>Специальная модель</i>	Логический	Спец. модель или стандартная	Формат: Да/Нет
<i>Дополнительное оснащение</i>	Длинный текст	Дополнительные аксессуары	Значение по умолчанию: Кондиционер, круиз-контроль
<i>Цена отпускная</i>	Вычисляемый	Суммарные издержки	Выражение: [Заводская цена] + [Транспортные издержки] + [Предпродажные издержки] (создается с помощью <i>Построителя выражений</i>)

**Мастер подстановок* позволяет создать для указанного поля список, из которого при вводе данных будут выбираться значения (рис. 2.5).

Создание подстановки

Выберите значения, которые будут содержать поле подстановки. Введите число столбцов списка и значения для каждой ячейки.

Перетащите правую границу заголовка столбца на нужную ширину или дважды щелкните ее для автоматического подбора ширины.

Число столбцов:

Столбец1					
Велюр					
Кожа					
Ткань					
*					

Рис. 2.5. Создание подстановки с фиксированным набором значений

4.4. *Сохраните* созданный макет таблицы под именем *Автомобили* (рис. 2.6).

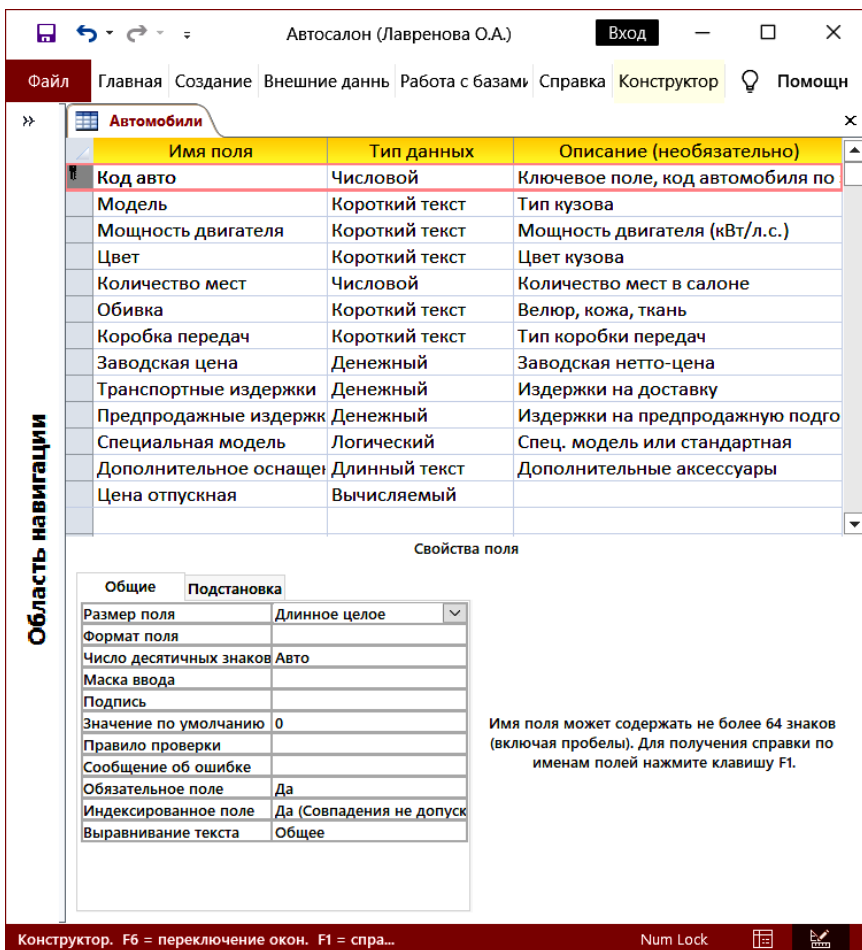


Рис. 2.6. Макет таблицы *Автомобили* в режиме *Конструктор*

4.5. В режиме *Таблицы* добавьте в таблицу *Автомобили* 3 записи (табл. 2.3). Для уточнения технических характеристик используйте материалы сайта geely.by. Приведите в отчете окно с записями таблицы *Автомобили* (в режиме *таблицы*).

Внимание: значение в поле *Цена приобретения*, отображающем суммарные издержки на приобретение автомобиля, рассчитывается и выводится в поле автоматически.

Таблица 2.3

Примерное содержание полей 3-х записей таблицы *Автомобили*

Имя поля таблицы	Примеры значений для трех записей №		
	1	2	3
Код модели	12580	12653	12651
Модель	Gelly Atlas Pro	Gelly Atlas Pro	Gelly Atlas Pro
Комплектация	Comfort	Luxury	Flagship
Мощность двигателя	130/177	130/177	140/190
Цвет	Белый	Черный	Серебристый
Количество мест	4	5	7
Обивка	Ткань	Велюр	Кожа
Коробка передач	6АКПП	6АКПП	7DCT
Заводская цена	77600	80800	91080
Транспортные издержки	776	808	910
Предпродажные издержки	385	425	480
Дополнительное оснащение	Кондиционер, панорамная крыша, камера заднего вида	Кондиционер, круиз-контроль, камера кругово- го обзора	Кондиционер, адаптивный круиз-контроль, бесключевой доступ
Специальная модель	Нет	Нет	Да

4.6. Создайте макет таблицы *Клиенты*.

Примерное описание макета таблицы *Клиенты* (имена полей, их свойства и типы данных) приведено в табл. 2.4.

Внимание: поля *Код клиента*, *Фамилия*, *Страна* должны быть *обязательными* для заполнения (свойство *Обязательное поле* : «Да»).

Примерное содержание макета таблицы *Клиенты*

Имя поля	Тип данных	Описание	Свойства поля
Код клиента	Счетчик	Ключевое поле, уникальный номер клиента в БД	Индексированное поле: Да/Совпадения не допускаются Ключевое поле задается в <i>контекстном меню</i> или на вкладке <i>Конструктор</i>
Фамилия	Текстовый	Фамилия	Размер поля: 40 Индексированное поле: Да/Совпадения допускаются
Имя	Текстовый	Имя	Размер поля: 20 Индексированное поле: Да/Совпадения допускаются
Отчество	Текстовый	Отчество	Размер поля: 25
Индекс	Текстовый	Почтовый индекс	Размер поля: 6 Маска ввода: 000 000
Страна	Текстовый	Название страны	Размер поля: 20 Значение по умолчанию: Беларусь Индексированное поле: Да/Совпадения допускаются
Населенный пункт	Текстовый	Название населенного пункта	Размер поля: 40 Значение по умолчанию: Минск Индексированное поле: Да/Совпадения допускаются
Почтовый адрес	Текстовый	Почтовый адрес (улица, дом, квартира)	Размер поля: 50
Телефон	Текстовый	Контактный телефон	Размер поля: 20 Маска ввода: (99)000\ -00\ -00

4.7. *Сохраните* созданный макет таблицы под именем *Клиенты* (макет таблицы приведите в отчете).

4.8. В режиме *Таблицы* добавьте в таблицу *Клиенты* 3 записи (значения полей задайте самостоятельно, результат – записи таблицы в режиме *Таблицы* приведите в отчете).

4.9. Таблицу *Сотрудники* импортируйте из учебной БД *Борей*.

На вкладке *Внешние данные* в группе команд *Импорт и связь* нажмите кнопку «Создать источник данных», выберите команду *Из базы данных*, и *Access*. Далее в диалоговом окне (рис. 2.7) установите переключатель *Импорт таблиц, запросов, форм, отчетов, макросов и модулей в текущую базу данных* и с помощью кнопки «Обзор» выберите в личной папке файл БД «Борей» и нажмите кнопку «ОК».

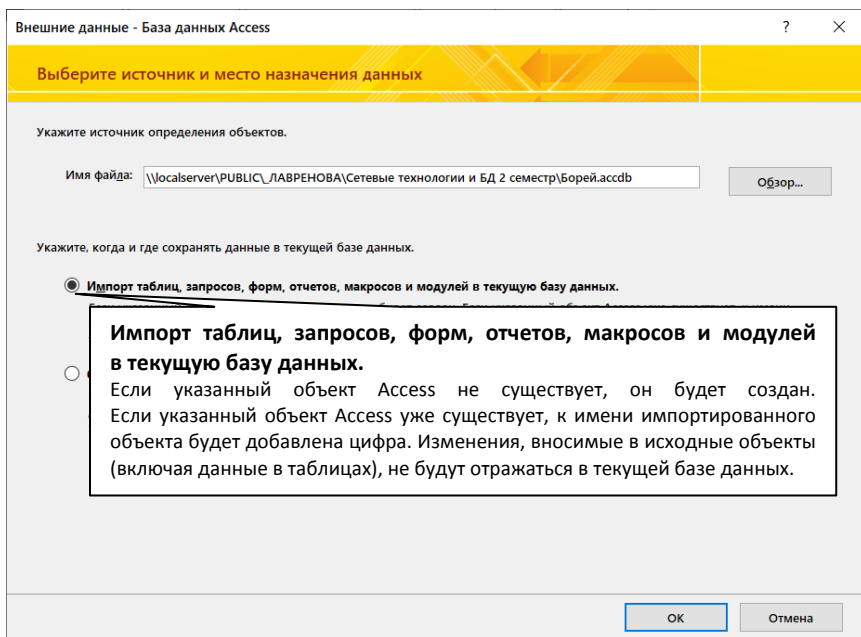


Рис. 2.7. Окно «Внешние данные – База данных Access» с выбором источника импорта БД

В открывшемся диалоговом окне «Импорт объектов» на вкладке *Таблицы* выберите таблицу *Сотрудники* (рис. 2.8) и нажмите кнопку «ОК». Приведите в отчете результат: записи таблицы *Сотрудники* в режиме *Таблицы*.

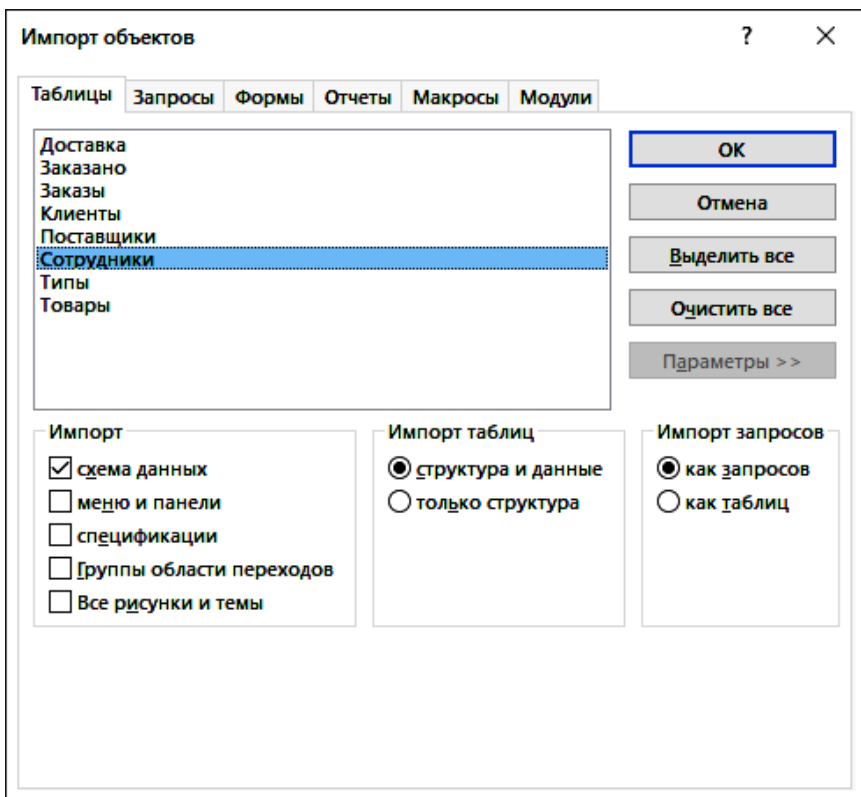


Рис. 2.8. Окно «Импорт объектов» – выбор таблицы *Сотрудники* в качестве объекта импорта

4.10. Создайте макет *таблицы Заказы*.

Примерное содержание макета таблицы *Заказы* (имена полей, их свойства и типы данных) приведено в табл. 2.5.

Внимание: *все поля*, за исключением поля *Скидка*, должны быть *обязательными* для заполнения (свойство *Обязательное поле*: Да). Самостоятельно добавьте в макет таблицы *Заказы* поле *Код сотрудника* для связи с таблицей *Сотрудники*. Тип данных и свойства поля выберите аналогично полю *Код клиента* (дополните табл. 2.5).

Таблица 2.5

Примерное содержание макета таблицы *Заказы*

Имя поля	Тип данных	Описание	Свойства поля
Код заказа	Счетчик	<i>Ключевое поле, уникальный номер заказа</i>	Индексированное поле: Да/Совпадения не допускаются
Код авто	Числовой, <i>*Мастер подстановок</i>	<i>Внешний ключ, для связи с таблицей Автомобили</i>	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
Код клиента	Числовой, <i>*Мастер подстановок</i>	<i>Внешний ключ, для связи с таблицей Клиенты</i>	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
Код сотрудника	Числовой, <i>*Мастер подстановок</i>	<i>Внешний ключ, для связи с таблицей Сотрудники</i>	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
Дата заказа	Дата/время	Дата формирования заказа в среднем формате даты ДД-МММ-ГГ	Формат: Краткий формат даты Значение по умолчанию: =Date() Индексированное поле: Да/Совпадения допускаются
Скидка	Числовой	Размер скидки в %, <i>не превышает 50 %</i>	Размер поля: Одинарное с плавающей точкой Формат: Процентный Значение по умолчанию: 0,05 Правило проверки: <i>Between 0 And 0,5</i>

*Используя *Мастер подстановок*, сформируйте для полей *Код авто*, *Код клиента* и *Код сотрудника* список значений из таблиц *Автомобили*, *Клиенты* и *Сотрудники* соответственно:

1) для поля *Код авто* выберите для столбца подстановки 2 поля: *Код авто* и *Модель* (снимите флажок *Скрыть ключевое поле*);

2) для поля *Код клиента* выберите для столбца подстановки 2 поля: *Код клиента* и *Фамилия* (не снимайте флажок *Скрыть ключевое поле*).

3) для поля *Код сотрудника* выберите для столбца подстановки 2 поля: *Код сотрудника* и *Фамилия* (не снимайте флажок *Скрыть ключевое поле*).

4.11. *Сохраните* макет таблицы под именем *Заказы*.

Внимание: в отчете приведите распечатку окна макета таблицы *Заказы* с активным полем *Код заказа* (вкладка *Общие*) и дополнительно – распечатки окон со свойствами полей *Код авто*, *Код клиента* и *Код сотрудника* (с активной вкладкой *Подстановка*).

4.12. В режиме *Таблицы* добавьте в таблицу *Заказы* не менее 15 записей (значения полей задайте самостоятельно, результат приведите в отчете).

4.13. *Сформируйте схему БД* (установите связи между таблицами БД в соответствии с информационно-логической моделью предметной области):

4.13.1. Закройте все таблицы БД (обязательное действие).

4.13.2. Выполните команду *Схема данных* на вкладке *Работа с базами данных* в группе *Отношения*.

4.13.3. В диалоговом окне «Добавление таблицы» последовательно добавьте все таблицы.

Внимание: окно «Добавление таблицы» *не отображается*, если в таблице *Заказы* для внешних ключей сформированы *списки подстановок* значений из соответствующих таблиц и связи между таблицами уже сформированы.

4.13.4. Далее, чтобы установить связь между главной и подчиненной таблицей, перетащите ключевое поле из главной таблицы на соответствующее поле подчиненной таблицы.

Например, установите связь между таблицами *Клиенты* и *Заказы*, перетащив ключевое поле *Код клиента* из главной таблицы *Клиенты* на соответствующее поле таблицы-связки *Заказы*.

В открывшемся диалоговом окне «Изменение связей» установите флажок *Обеспечение целостности данных* (рис. 2.9).

Внимание: если связь между таблицами уже сформирована при помощи *Мастера подстановок*, вызовите диалоговое окно «Изменение связей» (выделите связь, используйте КЗМ/*Изменить связь...*)

и в диалоговом окне установите флажок «Обеспечение целостности данных».

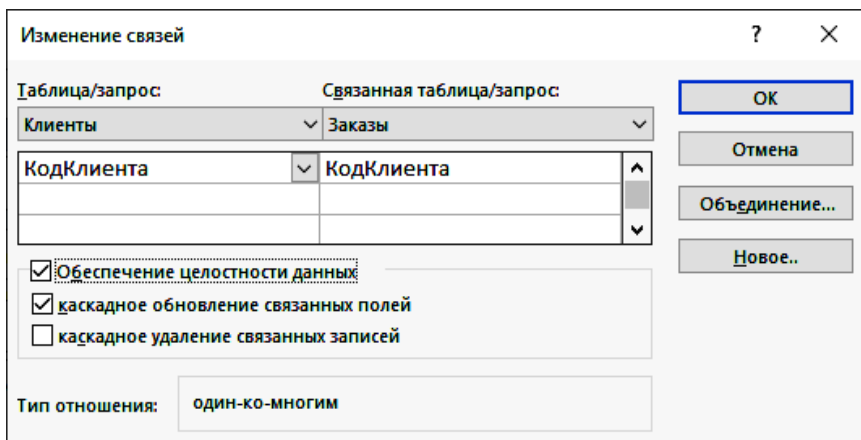


Рис. 2.9. Формирование связи между таблицами *Клиенты* и *Заказы*

4.13.5. Аналогично установите связь между остальными главными таблицами и таблицей–связкой *Заказы*, перетащив ключевое поле из каждой главной таблицы на соответствующее поле внешнего ключа в таблице–связке *Заказы* и установив в диалоговом окне «Изменение связей» флажок «Обеспечение целостности данных».

Внимание: Если при построении схемы в окне схемы данных отображаются не все таблицы БД, откройте диалоговое окно «Добавление таблицы», далее в списке таблиц выделите недостающие таблицы и нажмите кнопку «Добавить», закройте окно «Добавление таблицы». Выполните действия по созданию связей между таблицами.

4.13.6. Сохраните макет схемы данных (рис. 2.10).

Внимание: Выполните команду *Параметры* в меню *Файл* и в окне «Параметры Access» на вкладке *Текущая БД* задайте *Заголовок приложения* (рис. 2.11).

5. Завершите работу с СУБД MS Access.

6. Для отчета подготовьте распечатку окна БД, отобразите объекты вкладки *Таблицы* (список созданных таблиц в режиме *Просмотра сведений*), см. рис. 2.10).

7. Оформите отчет о выполнении работы.

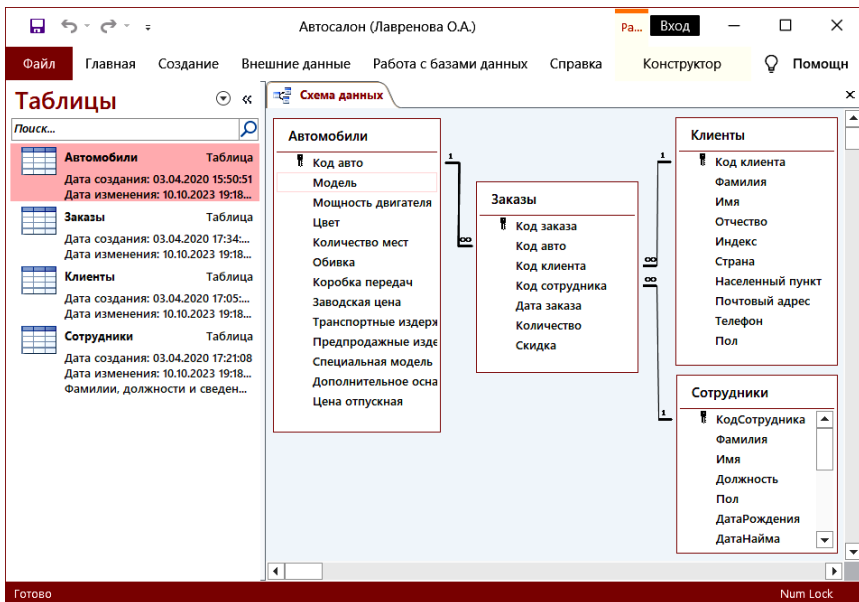


Рис. 2.10. Схема БД «Автосалон»

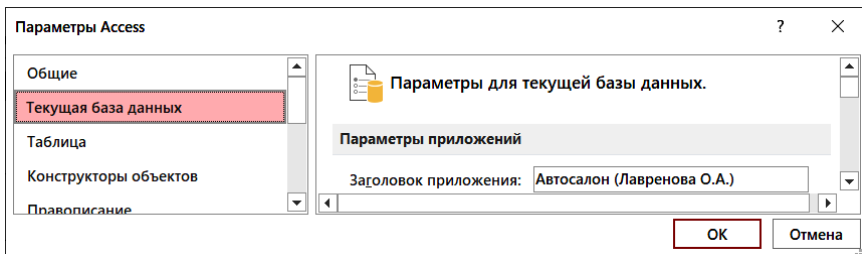


Рис. 2.11. Настройка заголовка приложения для БД «Автосалон»

Лабораторная работа № 3

РАЗРАБОТКА ТАБЛИЦ ИНДИВИДУАЛЬНОЙ БАЗЫ ДАННЫХ



Цель работы: – изучить требования к содержанию таблиц реляционных БД и способы их создания;
– приобрести практический опыт разработки таблиц реляционных БД.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи по варианту.
3. Содержание этапов проектирования индивидуальной БД.
4. Спецификации разработанных таблиц (имя поля, тип данных, свойства) и их макеты в режиме *Конструктора*.
5. Примеры заполнения таблиц согласно постановке задачи.
6. Структурная схема индивидуальной БД.
7. Список контрольных вопросов и ответов на них.
8. Выводы.
9. Список использованных источников.



Контрольные вопросы

1. Назовите способы создания таблиц в СУБД MS Access и укажите их отличительные особенности.
2. Приведите порядок создания макета таблицы в режиме *Конструктора*. Перечислите правила формирования имени поля таблицы.
3. Что понимают под типом данных? Какие типы данных можно обрабатывать в таблицах СУБД MS Access? Приведите их краткую характеристику с примерами конкретных полей.
4. Какие типы данных можно использовать при определении первичного ключа? внешнего ключа?
5. Что понимают под свойствами поля? Приведите краткую характеристику основных свойств полей таблицы БД.

6. Сформулируйте понятие схемы БД и приведите порядок ее формирования в СУБД MS Access. Какие параметры объединения можно установить при формировании связей между таблицами?



Порядок выполнения работы

1. Постановка задачи

Спроектировать и реализовать средствами СУБД MS Access реляционную БД для предметной области *по варианту*.

Вариант 1. БД «Договора подряда»

Предметная область: строительные работы.

Входная информация:

1. Фамилия, имя, Отчество сотрудника.
2. Домашний адрес и телефон сотрудника.
3. Характеристика сотрудника.
4. Название должности сотрудника.
5. Должностные оклады сотрудников.
6. Наименование проекта.
7. Дата начала/окончания проекта.
8. Руководитель проекта.
9. Описание проекта.
10. Количество дней, отработанных сотрудником в проекте

Количественные ограничения: БД должна содержать информацию о 10-ти сотрудниках, 5-ти должностях, 3-х проектах.

Предусмотреть, чтобы не менее 5-ти сотрудников работали в 2-х проектах одновременно.

Вариант 2. БД «Call-центр»

Предметная область: работа call-центра предприятия.

Входная информация:

1. Фамилия, имя, отчество абонента.
2. Адрес и телефон абонента.
3. Образование и социальный статус абонента.
4. Номер и линия оператора.
5. Фамилия, имя, отчество оператора.
6. Домашний адрес и телефон оператора.

7. Дата рождения оператора.
 8. Дата звонка, время звонка.
 9. Продолжительность разговора.
 10. Тип связи (мобильная/стационарная сеть/внутренняя) и тариф за 1 минуту.
 11. Тип звонка (входящий/исходящий).
 12. Тема и содержание разговора (звонка).
- Количественные ограничения:* БД должна содержать информацию о 10-ти абонентах, 5-ти операторах call-центра.
- Предусмотреть, чтобы каждый из 5-ти абонентов и 3-х операторов сделали не менее 2-х телефонных звонков на различные темы.

Вариант 3. БД «Торговые операции»

Предметная область: оптовая продажа товаров (определенной категории).

Входная информация:

1. Название фирмы-клиента.
2. Фамилия, имя, отчество руководителя.
3. Юридический адрес фирмы-клиента.
4. Контактный телефон.
5. Название и описание товара.
6. Единица измерения товара.
7. Цена товара.
8. Количество проданного товара.
9. Дата продажи.
10. Скидка, %.

Количественные ограничения: БД должна содержать информацию о 5-ти фирмах-клиентах, 10-ти товарах.

Предусмотреть, чтобы 5 фирм сделали не менее 3-х заказов различных товаров.

Вариант 4. БД «Почасовая оплата занятий»

Предметная область: почасовая работа преподавателей вуза.

Входная информация:

1. Фамилия, имя, отчество преподавателя.
2. Телефон преподавателя.
3. Место работы преподавателя.
4. Домашний адрес преподавателя.

5. Название должности преподавателя.
6. Должностная почасовая ставка преподавателя.
7. Предмет, который ведет преподаватель.
8. Общее число часов занятий по предмету.
9. Рабочая программа предмета.
10. Дата проведения занятий.
11. Количество проведенных часов.

Количественные ограничения: БД должна содержать информацию о 5-ти преподавателях, 4-х должностях, 5-ти предметах.

Предусмотреть, чтобы не менее 3-х преподавателей проводили занятия по 2-м предметам одновременно.

Вариант 5. БД «Методическая библиотека»

Предметная область: выдача учебно-методической литературы.

Входная информация:

1. Фамилия, имя, отчество читателя.
2. Домашний адрес читателя.
3. Телефон читателя.
4. Дата рождения читателя.
5. Номер читательского билета.
6. Название, автор, год издания и издательство книги, которую взял читатель.
7. Рубрика.
8. Цена книги.
9. Дата выдачи книги.
10. Дата возврата книги плановая и фактическая.

Количественные ограничения: БД должна содержать информацию о 5-ти читателях, 10-ти книгах.

Предусмотреть, чтобы каждый читатель взял не менее 2-х книг.

Вариант 6. БД «Банковские вклады»

Предметная область: операции по вкладам.

Входная информация:

1. Фамилия, имя, отчество вкладчика.
2. Серия и номер паспорта вкладчика.
3. Домашний адрес вкладчика.
4. Номер счета в банке.
5. Название вклада.

6. Вид вклада (рублевый, валютный).
 7. Процентная ставка и минимальный срок вклада.
 8. Особые условия вклада (пополнение вклада, капитализация процентов и т. д.).
 9. Дата операции.
 10. Тип и сумма операции (взнос/выдача).
- Количественные ограничения:* БД должна содержать информацию о 10-ти вкладчиках, 5-ти видах вкладов.
- Предусмотреть, чтобы каждый вкладчик совершал операции по вкладу не менее 2-х раз.

Вариант 7. БД «Бронирование мест»

Предметная область: бронирование мест в гостинице.

Входная информация:

1. Фамилия, имя, отчество клиента.
2. Домашний адрес клиента.
3. Телефон клиента.
4. Номер комнаты и тип комнаты (люкс, 2-, 3-местный).
5. Описание номера.
6. Дата заселения и дата выезда.
7. Время заселения/выезда.
8. Стоимость проживания за 1 сутки.
9. Доплата за бронь, %.
10. Доплата за тип номера, %.

Количественные ограничения: БД должна содержать информацию о 10-ти клиентах, 10-ти номерах.

Предусмотрите, чтобы каждый клиент бронировал номера в гостинице как минимум 2 раза.

Вариант 8. БД «Фотосервис»

Предметная область: работа фотоателье.

Входная информация:

1. Фамилия, имя, отчество клиента.
2. Домашний адрес клиента.
3. Телефон клиента.
4. Наименование фотоуслуги.
5. Описание фотоуслуги (характеристика).
6. Количество единиц заказа.

7. Цена за единицу.
8. Дата приемки/выдачи заказа.
9. Надбавка за срочность исполнения, %.
10. Скидка, %.

Количественные ограничения: БД должна содержать информацию о 10-ти клиентах, 5-ти видах услуг.

Предусмотрите, чтобы каждый клиент заказывал как минимум 2 фотоуслуги.

Вариант 9. БД «Коммунальные платежи»

Предметная область: оплата коммунальных услуг.

Входная информация:

1. Фамилия, имя, отчество квартиросъемщика.
2. Домашний адрес квартиросъемщика.
3. Номер лицевого счета.
4. Количество жильцов.
5. Площадь квартиры, кв.м.
6. Вид коммунальной услуги (название платежа).
7. Стоимость услуги на квадратный метр площади.
8. Стоимость услуги на 1 жильца.
9. Дата оплаты.
10. Скидка, % (льготы при оплате).

Количественные ограничения: БД должна содержать информацию о 7-ми квартиросъемщиках, 5-ти видах услуг (стоимость одних услуг должна определяться площадью квартиры, других – количеством жильцов).

Предусмотрите, чтобы каждый квартиросъемщик оплачивал не менее 3-х коммунальных услуг.

Вариант 10. БД «Салон проката»

Предметная область: работа салона проката бытовой техники.

Входная информация:

1. Фамилия, имя, отчество клиента.
2. Серия и номер паспорта клиента.
3. Домашний адрес и телефон клиента.
4. Наименование и модель бытовой техники.
5. Описание техники.
6. Залоговая стоимость техники.

7. Дата выдачи.
8. Дата возврата плановая и фактическая.
9. Стоимость проката за сутки.

Количественные ограничения: БД должна содержать информацию о 10-ти клиентах, 5-ти моделях бытовой техники.

Предусмотрите, чтобы 5 клиентов пользовались услугами проката не менее 2-х раз.

Вариант 11. БД «Посещаемость занятий»

Предметная область: контроль пропусков занятий студентами.

Входная информация:

1. Фамилия, имя, отчество студента.
2. Домашний адрес и телефон студента.
3. Форма обучения студента (бюджет, за оплату).
4. Дата рождения.
5. Название предмета и календарный план по предмету.
6. Дата занятия, и количество часов пропусков.
7. Причина пропуска (уважительная/неуважительная).
8. Вид занятия (лекция, лабораторная работа, практическое занятие, семинар).
9. Фамилия, имя, отчество и телефон преподавателя.

Количественные ограничения: БД должна содержать информацию о 10-ти студентах, 5-ти предметах, 2-х преподавателях.

Предусмотрите, чтобы 5 студентов пропустили не менее 2-х занятий по различным предметам.

Вариант 12. БД «Рейтинг студентов»

Предметная область: рейтинговая аттестация студентов.

Входная информация:

1. Фамилия, имя, отчество студента.
2. Домашний адрес и телефон студента.
3. Дата рождения студента.
4. Название предмета.
5. Дата выставления рейтинговой оценки.
6. Текущая рейтинговая оценка.
7. Максимальная рейтинговая оценка по предмету.
8. Фамилия, имя, отчество и телефон преподавателя.

Количественные ограничения: БД должна содержать информацию о 10-ти студентах, 5-ти предметах, 2-х преподавателях.

Предусмотрите, чтобы каждый из студентов имел рейтинговую оценку не менее чем по 2-м предметам.

Вариант 13. БД «Доставка пиццы»

Предметная область: работа пиццерии.

Входная информация:

1. Фамилия, имя, отчество клиента.
2. Адрес и телефон клиента.
3. Название пиццы.
4. Масса и цена пиццы.
5. Описание пиццы (ингредиенты).
6. Дата и время заказа.
7. Дата и время доставки заказа.
8. Количество заказанных порций.
9. Скидка, %.

Количественные ограничения: БД должна содержать информацию о 10-ти клиентах, 5-ти видах пиццы.

Предусмотрите, чтобы 5 клиентов сделали не менее 3-х заказов различных видов пиццы.

Вариант 14. БД «Туристическое бюро»

Предметная область: реализация туристических путевок.

Входная информация:

1. Фамилия, имя, отчество клиента.
2. Домашний адрес и телефон клиента.
3. Дата рождения клиента.
4. Название маршрута, длительность маршрута (в днях).
5. Описание маршрута.
6. Цена 1-й путевки.
7. Дата заказа и дата поездки.
8. Количество заказанных путевок.
9. Скидка, %.

Количественные ограничения: БД должна содержать информацию о 10-ти клиентах, 5-ти туристических маршрутах.

Предусмотрите, чтобы 5 клиентов заказали не менее 2-х путевок по различным маршрутам.

Вариант 15. БД «Видеосалон»

Предметная область: работа видеосалона.

Входная информация:

1. Фамилия, имя, отчество клиента.
2. Серия и номер паспорта клиента.
3. Домашний адрес и телефон клиента.
4. Наименование видеоматериала.
5. Жанр и описание видеоматериала.
6. Носитель видеоматериала и залоговая стоимость.
7. Стоимость проката/доступа за сутки.
8. Дата выдачи, дата возврата плановая и фактическая.

Количественные ограничения: БД должна содержать информацию о 10-ти клиентах, 5-ти видеоматериалах.

Предусмотрите, чтобы 5 клиентов пользовались услугами видеосалона не менее 3-х раз.

2. Описание предметной области *«Название по варианту»*.

Выполните *описание заданной предметной области* (согласно постановке задачи *по варианту*), включая:

- перечень первичных документов;
- состав информационных объектов и схемы их взаимодействия.

Внимание: дополнительно предусмотрите учет в создаваемой БД информации о сотруднике для вариантов, где это явно не указано в постановке задачи.

Сформулируйте потребности пользователя по автоматизации учета, обработки и анализа данных для выбранной предметной области (приведите примеры отчетов, итоговых документов, которые необходимы, *например*, для контроля деятельности).

Обоснуйте необходимость применения базы данных для автоматизации учета и обработки данных предметной области (см. п. 2 лабораторной работы № 2).

3. Поэтапное проектирование таблиц БД *«Название БД по варианту»*.

Выполните поэтапное проектирование информационно-логической модели предметной области по варианту (*приведите в отчете подробное описание процесса проектирования по примеру п. 3 лабораторной работы № 2*).

Внимание: результат выполнения (проект БД) предъявите преподавателю для проверки.

К созданию таблиц в MS Access приступайте только после подтверждения корректности модели и исправления выявленных ошибок!

4. Разработка таблиц БД «*Название БД по варианту*».

Создайте и реализуйте средствами СУБД MS Access таблицы индивидуальной БД в соответствии с порядком, приведенным в п. 4 лабораторной работы № 2.

Внимание: предварительно разработайте описания (спецификации) таблиц, включая имена полей, типы данных, свойства и описания.

При выборе типов данных и свойств полей руководствуйтесь тем, какая информация в них сохраняется.

Измените свойство *Значение по умолчанию* для полей *Город*, *Страна*, *Дата заказа* (и др. по выбору) и создайте маски ввода для полей *Индекс* и *Телефон*.

Для отчета подготовьте *спецификации таблиц* (см. лабораторную работу № 2, таблицы 2.2, 2.3, 2.4) и *макеты таблиц* в режиме *Конструктора* (см. рис. 2.6). Дополнительно приведите в отчете распечатки окон макета таблицы-связки с отображением вкладки *Подстановка для каждого внешнего ключа*.

Выполните информационное наполнение таблиц базы данных. Значения полей в таблицах задайте самостоятельно. В отчете приведите распечатки окон с записями таблиц БД в режиме *Таблица*.

Внимание: в созданной индивидуальной БД в последующих лабораторных работах *будут создаваться другие объекты* (запросы, табличные и кнопочные формы, отчеты, макросы).

Поэтому тщательно *проверяйте типы данных и свойства полей*, своевременно *исправляйте ошибки на этапе проектирования!*

5. Формирование схемы БД «*Название БД по варианту*».

В соответствии с порядком, приведенным в подпункте 4.13 п. 4 лабораторной работы № 2 сформируйте схему данных: установите связи между исходными таблицами, обеспечьте целостность данных.

В отчете приведите описание построения схемы БД и результат.

6. Документирование БД «*Название БД по варианту*».

6.1. Получите описание макетов таблиц, используя инструмент «Архивариус», который запускается кнопкой «Средство документиро-

вания базы данных» на вкладке *Работа с базами данных* в группе команд *Анализ* (рис. 3.1).

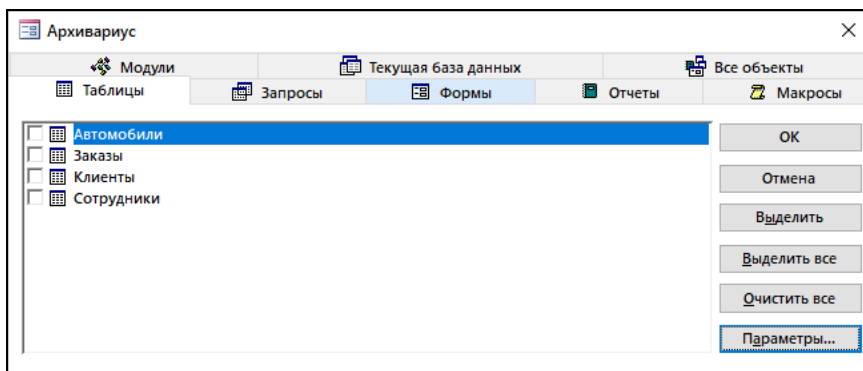


Рис. 3.1. Диалоговое окно «Архивариус» для документирования БД

«Архивариус» создает отчет, содержащий подробные сведения о каждой выбранной таблице, который открывается непосредственно в MS Access в режиме *Предварительного просмотра*.

Далее на вкладке *Предварительный просмотр* в группе команд *Данные* раскройте список *Дополнительно*, выберите команду *Word...* и выполните экспорт описания таблиц в файл формата RTF (укажите в какую папку и файл по умолчанию происходит экспорт).

В отчете приведите *пример одной страницы* с описанием таблицы-связки (в виде распечатки окна).

6.2. Для оценки корректности построения БД выполните анализ таблиц индивидуальной БД, используя команду *Анализ быстройдействия* на вкладке *Работа с базами данных* в группе команд *Анализ* (рис. 3.2).

В отчете приведите диалоговое окно с результатами анализа.

7. Для отчета подготовьте распечатку окна БД, вкладка *Таблицы* (список созданных таблиц в режиме *Просмотра сведений*).

8. Оформите отчет о выполнении работы.

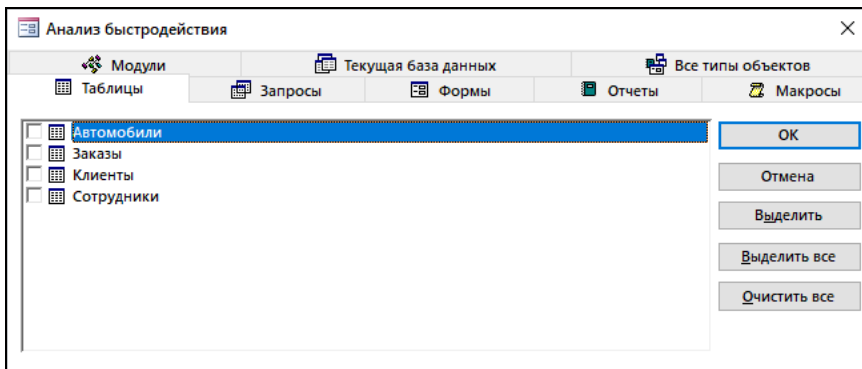


Рис. 3.2. Диалоговое окно «Анализ быстродействия»

Лабораторная работа № 4

АНАЛИЗ ДАННЫХ ПРИ ПОМОЩИ ЗАПРОСОВ ВЫБОРКИ



- Цель работы:**
- изучить инструменты анализа данных в реляционной БД;
 - изучить назначение и особенности запросов выборки, способы их создания;
 - приобрести практический опыт создания простых запросов выборки и запросов с условиями отбора для полей различных типов.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи по варианту.
3. Макеты запросов в режиме *Конструктора* (при необходимости дополнительно приведите окно *Мастера запросов*, окно *Свойства поля*, окно *Свойства запроса*).
4. Результаты выполнения запросов.
5. Окно БД, вкладка *Запросы* (список созданных запросов).
6. Список контрольных вопросов и ответов на них.
7. Выводы.
8. Список использованных источников.



Контрольные вопросы

1. Перечислите инструменты анализа данных из таблиц БД. Приведите понятие и назначение запросов СУБД MS Access.
2. Приведите классификацию запросов СУБД MS Access (в виде схемы) и их особенности. Назовите источники формирования запросов СУБД MS Access и результаты их работы.
3. Перечислите способы создания запросов СУБД MS Access. Какие запросы можно создать в режиме *Мастера*?
4. Приведите порядок создания запроса выборки в режиме *Конструктора*. Опишите структуру окна *Конструктора запроса* и *Бланка запроса*.

5. Что представляет собой условие отбора в запросе? Из каких элементов состоит выражение для условия отбора? Какие операторы используются при записи выражений в условиях отбора? Приведите примеры формирования условий отбора для полей различных типов (в виде таблицы, с описанием), запишите условие отбора для поиска пустых значений.

6. Какие свойства можно изменить для полей в запросе? В каких случаях для поля в запросе целесообразно задать подпись?

7. Как создать для поля специальный (пользовательский) формат? Опишите структуру пользовательского формата для текстовых и числовых полей, приведите примеры кодов, применяемых для создания специальных форматов.

8. Каким образом можно отобразить свойства запроса? Какое свойство запроса необходимо изменить, чтобы ограничить количество записей, отображаемых по запросу?



Порядок выполнения работы

Внимание: названия таблиц и полей, включаемых в запросы, приведены по вариантам из лабораторной работы № 3.

1. Простой запрос

В режиме Мастера «Простой запрос» сформируйте запрос выборки, позволяющий частично отобразить информацию из базовой таблицы, содержащей персональные сведения (*Клиенты, Операторы, Преподаватели, Студенты* и т. п. по варианту): *Фамилия, Имя, Отчество, Телефон*.

Укажите имя, предлагаемое по умолчанию для простого запроса:

Замените имя по умолчанию именем *01 Простой запрос*.

2. Исходные данные.

В режиме Мастера «Простой запрос», сформируйте запрос выборки, позволяющий отобразить всю информацию из базовых таблиц.

Внимание: в запрос не включаются первичные ключи из главных таблиц!

В отчете приведите:

– диалоговое окно «Мастера простого запроса» в режиме *Выбора полей* из одной (по выбору) главной таблицы по варианту;

– результат выполнения запроса.

Замените имя запроса по умолчанию именем *02 Исходные данные*.

3. *Записи без подчиненных*.

В режиме *Мастера «Записи без подчиненных»* на основе базовой таблицы, содержащей персональные сведения (*Клиенты, Операторы, Преподаватели, Студенты* и т. п. по варианту), сформируйте запрос выборки, включающий список фамилий (клиентов, операторов, преподавателей, студентов и т. п.), которым не соответствует ни одна запись в подчиненной таблице (таблице-связке).

Для отображения в запросе включите поля *Фамилия, Имя, Отчество, Телефон*.

Какое имя по умолчанию предлагается для запроса? _____.

В отчете приведите:

– диалоговое окно Мастера запроса «Поиск записей, не имеющих подчиненных» со списком полей из главной и подчиненной таблиц по варианту (*проверьте соответствие ключевых полей из выбранных таблиц*);

– результат выполнения запроса;

– макет запроса в режиме *Конструктора* (отобразите условие отбора по ключевому полю подчиненной таблицы).

Замените имя по умолчанию именем *03 Записи без подчинения*.

4. *Три самых (дешевых/дорогих)*.

В режиме *Конструктора* сформируйте запрос выборки, позволяющий частично отобразить информацию из базовых таблиц (по варианту).

Состав базовых таблиц и полей, необходимых для запроса, определите самостоятельно.

Поле для сортировки и порядок сортировки выберите самостоятельно в соответствии с заданием по варианту.

Для поля с числовым значением задайте подпись *Три самых (дешевых, дорогих и т. п.)*, используя свойство поля *Подпись*.

Ограничьте набор значений в запросе 3 записями, используя свойство запроса *Набор значений* (рис. 4.1).

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *04 Три самых (дешевых/дорогих ...*, название уточните по варианту).

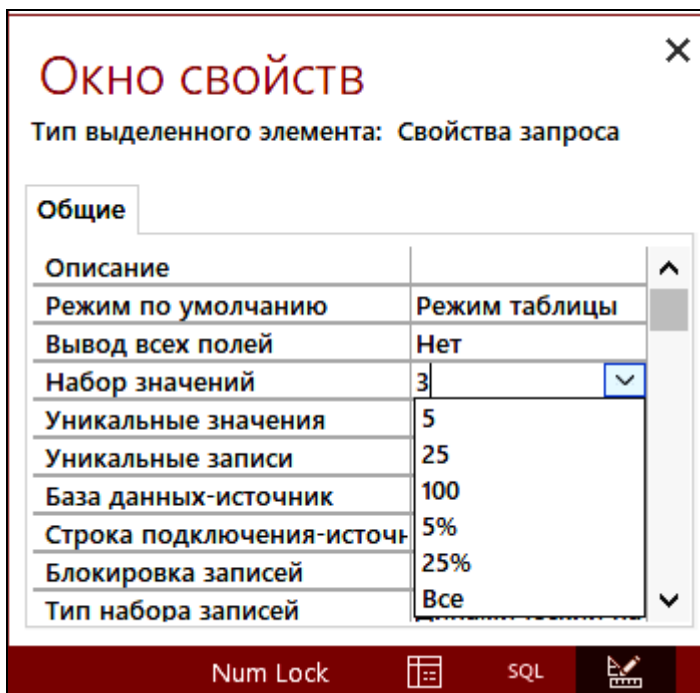


Рис. 4.1. Свойства запроса 04 (ограничение набора значений)

Вариант 1. Сформируйте список из трех самых низкооплачиваемых сотрудников, имеющих самые маленькие должностные оклады.

Вариант 2. Сформируйте список из трех самых «разговорчивых» абонентов, имеющих самую большую разовую продолжительность разговора.

Вариант 3. Сформируйте список из трех самых дешевых товаров.

Вариант 4. Сформируйте список из трех самых высокооплачиваемых преподавателей, имеющих самую большую должностную почасовую ставку.

Вариант 5. Сформируйте список из трех самых дорогих книг.

Вариант 6. Сформируйте список из трех самых высокодоходных вкладов, имеющих самую высокую процентную ставку.

Вариант 7. Сформируйте список из трех самых дешевых номеров, имеющих самую низкую стоимость проживания за сутки.

Вариант 8. Сформируйте список из трех самых дорогих фотоуслуг.

Вариант 9. Сформируйте список из трех самых дешевых коммунальных услуг.

Вариант 10. Сформируйте список из трех самых дорогостоящих товаров, имеющих самую высокую залоговую стоимость.

Вариант 11. Сформируйте список из трех самых прогуливающихся студентов, имеющих самое большое разовое число часов пропусков занятий по предмету.

Вариант 12. Сформируйте список из трех самых успевающих студентов, имеющих самую высокую текущую рейтинговую оценку по предмету.

Вариант 13. Сформируйте список названий из трех самых дорогих видов пиццы.

Вариант 14. Сформируйте список из трех самых дорогих маршрутов.

Вариант 15. Сформируйте список из трех самых дешевых видеоматериалов.

5. Список с условием.

В режиме *Конструктора* на основе базовых таблиц сформируйте запрос выборки (по варианту). Состав базовых таблиц и полей, необходимых для запроса, определите самостоятельно.

Поле для сортировки и порядок сортировки выберите самостоятельно (укажите в отчете).

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *05 Список с условием*.

Вариант 1. Сформируйте список сотрудников с должностным окладом не менее определенного значения. Величину оклада задайте самостоятельно.

Вариант 2. Сформируйте список операторов, разовая продолжительность разговора которых не менее определенного значения. Значение продолжительности разговора задайте самостоятельно.

Вариант 3. Сформируйте список товаров, цена которых не превышает определенного значения. Значение цены товара задайте самостоятельно.

Вариант 4. Сформируйте *список преподавателей*, должностная почасовая ставка которых не превышает определенного значения. Величину почасовой ставки задайте самостоятельно.

Вариант 5. Сформируйте *список книг*, цена которых не превышает определенного значения. Значение цены книги задайте самостоятельно.

Вариант 6. Сформируйте *список вкладчиков*, разовая сумма *взноса* которых не менее определенного значения. Величину суммы взноса задайте самостоятельно.

Вариант 7. Сформируйте *список номеров комнат*, стоимость проживания в которых не превышает определенного значения. Стоимость задайте самостоятельно.

Вариант 8. Сформируйте *список фотоуслуг*, цена которых не превышает определенного значения. Значение цены фотоуслуги задайте самостоятельно.

Вариант 9. Сформируйте *список коммунальных услуг*, стоимость которых не менее определенного значения. Величину стоимости услуги задайте самостоятельно.

Вариант 10. Сформируйте *список товаров*, залоговая стоимость которых не превышает определенного значения. Значение стоимости товара задайте самостоятельно.

Вариант 11. Сформируйте *список студентов*, число часов пропусков занятий которых более определенного значения. Величину пропусков задайте самостоятельно.

Вариант 12. Сформируйте *список студентов*, рейтинговая оценка которых не менее определенного значения. Значение рейтинговой оценки задайте самостоятельно.

Вариант 13. Сформируйте *список названий пиццы*, цена которых не превышает определенного значения. Значение цены задайте самостоятельно.

Вариант 14. Сформируйте *список маршрутов*, цена путевки которых не превышает определенного значения. Значение цены задайте самостоятельно.

Вариант 15. Сформируйте *список видеоматериалов*, залоговая стоимость которых не превышает определенного значения. Значение залоговой стоимости задайте самостоятельно.

6. *Список с условием по дате.*

В режиме *Конструктора* на основе базовых таблиц сформируйте запрос выборки (по варианту).

Состав базовых таблиц и полей, необходимых для запроса, определите самостоятельно.

Для условия отбора по полю со значением даты используйте оператор *Between ... And ...*.

Запишите в отчете, как будет выглядеть условие отбора по полю со значением даты, если использовать только *операторы сравнения* и *логические операторы*:

Поле для сортировки и порядок сортировки выберите самостоятельно (укажите в отчете).

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *Об Список с условием по дате*.

Вариант 1. Сформируйте *список сотрудников*, работавших в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 2. Сформируйте *список абонентов*, выполнивших звонки в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 3. Сформируйте *список товаров*, проданных в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 4. Сформируйте *список преподавателей*, проводивших занятия в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 5. Сформируйте *список книг*, выданных читателям в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 6. Сформируйте *список вкладчиков*, выполнявших операции по вкладам в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 7. Сформируйте *список номеров комнат*, заселенных в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 8. Сформируйте *список фотоуслуг*, заказанных в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 9. Сформируйте *список коммунальных услуг*, оплаченных в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 10. Сформируйте *список товаров*, выданных напрокат в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 11. Сформируйте *список студентов*, пропускавших занятия в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 12. Сформируйте *список студентов*, получивших рейтинговые оценки в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 13. Сформируйте *список названий пиццы*, заказанной в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 14. Сформируйте *список маршрутов*, путевки на которые были проданы в заданный период. Начальную и конечную дату периода задайте самостоятельно.

Вариант 15. Сформируйте *список видеоматериалов*, выданных напрокат в заданный период. Начальную и конечную дату периода задайте самостоятельно.

7. Алфавитный список с условием.

В режиме *Конструктора* на основе базовой таблицы, содержащей персональные сведения (*Клиенты, Операторы, Преподаватели, Студенты* и т. п. по варианту), сформируйте запрос выборки, который выдает список фамилий в указанном буквенном диапазоне (от *Начальная_буква* __ до *Конечная_буква* __ включительно, буквы задайте самостоятельно).

В бланк запроса включите поля *Фамилия, Имя, Телефон*.

По полю *Фамилия* задайте сортировку *По возрастанию*.

По полю *Фамилия* при помощи *Построителя выражений* задайте *условие отбора* с применением оператора сравнения *Between ... And ...* (условие отбора приведите в отчете).

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *07 Алфавитный список с условием*.

Например, сформируйте список сотрудников, фамилии которых начинаются с букв от А до В включительно (рис. 4.2).

Для решения задачи условие отбора по полю *Фамилия* задается в виде:

Between "А" And "Г"

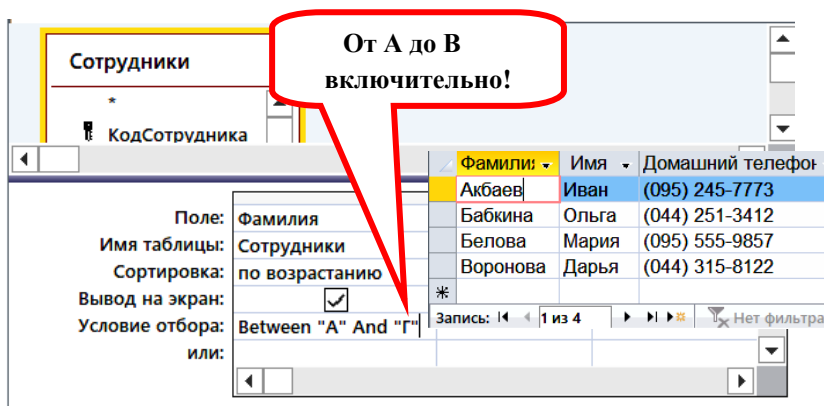


Рис. 4.2. Макет и результат запроса 07

8. Проживающие по адресу.

В режиме *Конструктора* на основе базовой таблицы, содержащей персональные сведения, сформируйте запрос выборки, который выдает список проживающих на улице _____ (название улицы задайте самостоятельно).

В бланк запроса включите поля *Фамилия*, *Имя*, *Адрес*.

По полю *Фамилия* задайте сортировку *По возрастанию*.

По полю *Адрес* при помощи *Построителя выражений* задайте условие отбора вида:

Like *фрагмент_названия_улицы*

(при формировании условия используйте шаблоны * и ?).

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *08 Проживающие по адресу*.

Например, сформируйте список клиентов, проживающих на улице Якуба Коласа (рис. 4.3).

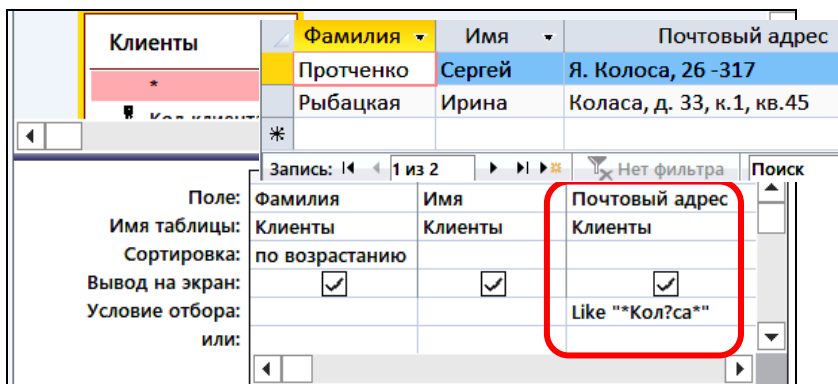


Рис. 4.3. Макет и результат запроса 08

9. Не проживающие в городе.

В режиме *Конструктора* на основе базовой таблицы, содержащей персональные сведения, сформируйте запрос выборки, который выдает список не проживающих в городе _____ (название города задайте самостоятельно).

В бланк запроса включите поля *Фамилия, Имя, Город, Адрес*.

По полю *Фамилия* задайте сортировку *По возрастанию*.

По полю *Город* при помощи *Построителя выражений* задайте условие отбора вида:

Not Like *фрагмент_названия_города*.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *09 Не проживающие в городе*.

10. Список по буквам фамилии и адресу.

В режиме *Конструктора* на основе базовой таблицы, содержащей персональные сведения (*Клиенты, Абоненты* и т. п. по варианту), сформируйте запрос выборки, который выдает список клиентов (абонентов и т. п. по варианту), не проживающих на одной улице _____ (название улицы задайте самостоятельно), фамилии которых могут начинаться на одну из двух букв _____ (буквы задайте самостоятельно).

В запрос включите поля *Фамилия, Имя, Отчество, Адрес*.

По полю *Фамилия* задайте условие на разных строках (*например*, Б* и Д*, если фамилии могут начинаться с букв Б или Д, соответственно).

По полю *Адрес* задайте условие:

Not Like *фрагмент_названия_улицы*.

Фрагмент названия улицы задайте самостоятельно (используйте шаблоны * и ?).

Значения в полях *Фамилия*, *Имя* преобразуйте к верхнему регистру, используя свойство *Формат поля* (>).

Отсортируйте список фамилий в алфавитном порядке.

Выполните запрос (отобразите результирующую таблицу), оцените корректность его выполнения.

Предложите альтернативный вариант записи условий отбора в запросе: _____.

Сохраните запрос с именем *10 Список по буквам фамилии и адресу*.

11. Для отчета подготовьте распечатку окна БД, вкладка *Запросы* (список созданных запросов в режиме *Просмотра сведений*).

12. Оформите отчет о выполнении работы.

Лабораторная работа № 5

ИСПОЛЬЗОВАНИЕ ВЫЧИСЛЯЕМЫХ ПОЛЕЙ И ПАРАМЕТРОВ В ЗАПРОСАХ MS ACCESS



- Цель работы:**
- изучить структуру вычисляемых полей и методику их создания в СУБД MS Access при помощи построителя выражений;
 - изучить назначение и особенности использования параметров в запросах выборки;
 - приобрести практический опыт создания запросов выборки с вычисляемыми полями и параметрами;
 - освоить применение статистических функций для построения итоговых и перекрестных запросов.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи по варианту.
3. Макеты запросов в режиме *Конструктора* (при необходимости дополнительно приведите окно *Параметров запросов*, окно *Свойства поля*, окно *Свойства запроса*).
4. Результаты выполнения запросов.
5. Окно БД, вкладка *Запросы* (список созданных запросов).
6. Список контрольных вопросов и ответов на них.
7. Выводы.
8. Список использованных источников.



Контрольные вопросы

1. Сформулируйте понятие выражения MS Access, в виде схемы приведите его структуру.
2. Дайте определение понятию «операнд», перечислите виды операндов, используемых в вычисляемых выражениях. Приведите в виде таблицы краткую характеристику и примеры операндов (констант, литералов, идентификаторов объектов и функций).

3. Что обозначают операторы в выражениях и какие виды операторов используются в MS Access? Ответ представьте в виде таблицы. Приведите таблицы истинности для логических операторов (And, Or, Not, Xor). Что означает понятие «приоритет операторов»?

4. Что понимают под вычисляемым полем в запросе MS Access? Опишите порядок создания вычисляемого поля. Какое имя присваивается вычисляемому полю по умолчанию? Приведите 2 примера записи вычисляемых полей.

5. Сформулируйте назначение *Построителя выражений* MS Access, перечислите способы его вызова. Опишите структуру окна, элементы выражений и порядок создания выражения (на примере применения текстовых функций).

6. Что представляют собой параметры в запросах и как они используются? Как формируется имя параметра? Приведите примеры. В каких случаях целесообразно использовать запросы с параметрами?

7. Назовите особенности построения итоговых запросов. Опишите назначение статистических функций.

8. Сформулируйте назначение перекрестных запросов, укажите способы и особенности их построения. Опишите структуру перекрестного запроса в режиме *Конструктора* и последовательность добавления итоговых значений.



Порядок выполнения работы

1. *Фамилия-верхний регистр.*

В режиме *Конструктора* на основе базовой таблицы, содержащей персональные сведения, сформируйте запрос выборки.

В запрос включите поля *Фамилия*, *Имя*, *Отчество*.

Создайте в строке *Поле Бланка* запроса новое вычисляемое поле с именем *Фамилия в верхнем регистре* и выражением, состоящим из текстовой функции *UCase()*, аргументом которой задайте поле *Фамилия*. Для формирования вычисляемого поля используйте *Построитель выражений*.

Отсортируйте список фамилий в алфавитном порядке.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *11 Фамилия-верхний регистр*.

2. *Фамилия-инициалы.*

В режиме *Конструктора* на основе базовой таблицы, содержащей персональные сведения, сформируйте запрос на выборку данных, который сформирует список всех фамилий и инициалов.

В запрос включите поля *Фамилия, Имя, Отчество*.

Дополнительно создайте вычисляемое поле *Фамилия_Инициалы* (*вычисляемое поле* строится с применением текстового оператора «&» и текстовой функции *Left()*). Для построения вычисляемого поля используйте *Построитель выражений*.

Отсортируйте список фамилий в алфавитном порядке.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *12 Фамилия-инициалы*.

3. *Запрос с расчетами.*

В режиме *Конструктора* на основе базовых таблиц (таблицы выберите самостоятельно) сформируйте запрос, включающий поля из базовых таблиц и вычисляемое поле по варианту (табл. 5.1).

Если в базовых таблицах присутствуют поля *Скидка, Надбавка, Доплата* и т. п., то формулу необходимо построить с учетом этих полей.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *13 Расчеты*.

Например, создайте запрос для отображения списка заказов и рассчитайте сумму каждого заказа с учетом предоставленной скидки.

В подсхему данных запроса необходимо включить таблицы *Заказы* и *Автомобили*, из которых в запросе будут использованы данные для отображения и вычислений.

В бланк запроса необходимо включить поля: *Код заказа, Дата заказа* (из таблицы *Заказы*) и вычисляемое поле *Сумма заказа* (создается в *Построителе выражений*, рис. 5.1):

$$\begin{aligned} \text{Сумма заказа: } & ([\text{Автомобили}]![\text{Заводская цена}] + \\ & + [\text{Автомобили}]![\text{Транспортные издержки}] + \\ & + [\text{Автомобили}]![\text{Предпродажные издержки}]) * \\ & *(1 - [\text{Заказы}]![\text{Скидка, \%}]) \end{aligned}$$

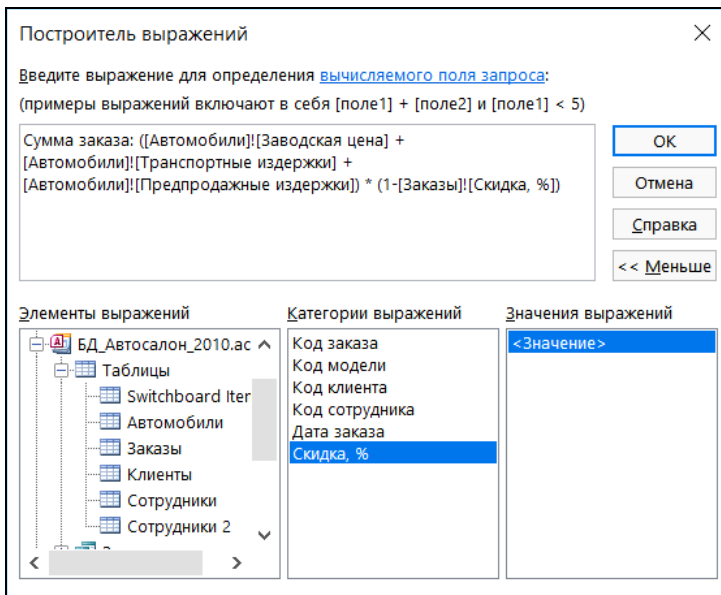


Рис. 5.1. Создание вычисляемого поля в построителе выражений

Для вычисляемого поля *Сумма заказа* свойству *Формат* задано значение *Денежный* (рис. 5.2).

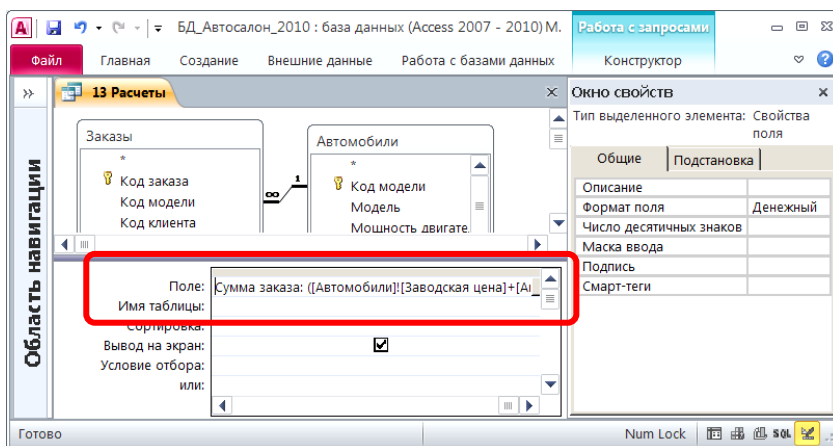


Рис. 5.2. Вычисляемое поле *Сумма заказа* и его свойства

Таблица 5.1

Исходные данные для запроса с расчетами

Вариант	Включить в запрос следующие поля:			Сортировать по полю, но поле на экран не выводить	
	1) поля из базовых таблиц	2) вычисляемое поле			
		имя	формула		формат
1	2	3	4	5	6
1	Фамилия, Имя, Отчество, Код сотрудника, Название проекта	Сумма к выплате	Оклад * Число отработанных дней / Число рабочих дней в месяце (задать самостоятельно)	Денежный	Код сотрудника
2	Дата звонка, Номер оператора, Фамилия, Имя, Отчество	Стоимость звонка	Тариф за мин * Продолжительность разговора	Денежный	Дата звонка
3	Название фирмы, Название товара, Цена товара, Количество, Дата продажи	Общая стоимость	Цена товара * Количество	Денежный	Дата продажи
4	Фамилия, Имя, Отчество, Название должности, Название предмета	Сумма к выплате	Должностная почасовая ставка * Количество часов	Денежный	Название должности

Продолжение табл. 5.1

1	2	3	4	5	6
5	Фамилия, Имя, Отчество, Номер билета, Название книги, Цена книги	Цена книги, увеличенная на 30 %	Цена книги * 1,3	Денежный	Номер билета
6	Фамилия, Имя, Отчество, Номер счета, Сумма операции	Сумма операции, увеличенная на 25 %	Сумма опера- ции * 1,25	Денежный	Номер счета
7	Фамилия, Имя, Отчество, Номер комнаты, Тип комнаты Дата заселения, Дата выбытия	Оплата за прожива- ние	Стоимость проживания за 1 сутки * Кол-во суток * (1 + Доплата за тип номера + Доплата за бронь)	Денежный	Дата выбытия
8	Фамилия, Имя, Отчество, Дата заказа, Название фотоуслуги, Цена, Количество	Стоимость заказа	Цена * Количе- ство	Денежный	Дата заказа

Продолжение табл. 5.1

1	2	3	4	5	6
9	Фамилия, Имя, Отчество, Номер лицевого счета, Количество жильцов, Площадь квартиры	Стоимость услуги	Стоимость за 1 кв. м * Площадь + Стоимость на 1 жильца * Кол-во жильцов	Денежный	Номер лицевого счета
10	Фамилия, Имя, Отчество, Дата выдачи, Название товара	Стоимость проката	Стоимость проката за 1 сут- ки * (Дата воз- врата – Дата выдачи)	Денежный	Дата выдачи
11	Фамилия, Имя, Отчество, Дата занятия, Название предмета, Количество часов пропусков	Стоимость отработки пропусков	Количество часов пропус- ков * <i>Цена 1 часа (цену задайте само- стоятельно)</i>	Денежный	Дата занятия
12	Фамилия, Имя, Отчество, Дата рождения, Название предмета, Рейтинг	Рейтинг, увеличен- ный на 10 %	Рейтинг * 1,1	Процент- ный	Дата рождения

1	2	3	4	5	6
13	Фамилия, Имя, Отчество, Дата заказа, Название пиццы, Цена, Количество	Стоимость заказа	Цена * Количе- ство порций	Денежный	Дата заказа
14	Фамилия, Имя, Отчество, Дата заказа, Название маршрута, Цена, Количество, Скидка	Стоимость заказа с учетом скидки	Цена * Количе- ство * (1 – Скид- ка)	Денежный	Дата заказа
15	Фамилия, Имя, Отчество, Дата выдачи, Название видео- материала	Стоимость проката	Стоимость проката за 1 сут- ки * (Дата воз- врата – Дата выдачи)	Денежный	Дата выдачи

4. Запрос с параметрами.

В режиме *Конструктора* на основе базовых таблиц (таблицы выберите самостоятельно) сформируйте запрос, включающий поля из базовых таблиц и вычисляемое поле по варианту (табл. 5.2).

В запросе используйте *2 параметра*:

1) в строке *Условие отбора* по полю в соответствии с вариантом используйте параметр с именем *[Введите ...]* (вместо многоточия укажите, например, название проекта, номер телефона и т. п. по варианту);

2) в выражении *вычисляемого поля* для пересчета значений используйте параметр с именем *[Введите курс USD]* (формулу для расчета значения по варианту постройте самостоятельно).

Для вычисляемого поля создайте пользовательский формат, например, для отображения значений с двумя обязательными знаками после запятой и обозначением валюты: `##0,00" $"`.

Выполните запрос, задав значения параметров.

В отчете приведите:

- макет запроса в режиме *Конструктора* (выражение для вычисляемого поля и выражение для условия отбора приведите в тексте отчета);

- окна параметров с введенными значениями при выполнении запроса;

- результат выполнения запроса.

Сохраните запрос с именем *14 Запрос с параметрами*.

Таблица 5.2

Исходные данные для запроса с параметрами

Вариант	Результат запроса	Имя вычисляемого поля	Параметры запроса
1	2	3	4
1	Список сотрудников, принимавших участие в указанном проекте и сумму к выплате в USD	Сумма к выплате в USD	Название проекта Курс USD, руб.
2	Список исходящих звонков указанного оператора и их стоимость в USD	Стоимость звонка в USD	Номер оператора Курс USD, руб.
3	Список товаров, проданных указанной фирме, и стоимость проданных товаров в USD	Стоимость товаров в USD	Название фирмы Курс USD, руб.
4	Список преподавателей, проводивших занятия по указанному предмету, и сумму к выплате в USD	Сумма к выплате в USD	Название предмета Курс USD, руб.
5	Список читателей, пользовавшихся указанной книгой, и стоимость этой книги в USD	Стоимость книги в USD	Название книги Курс USD, руб.

1	2	3	4
6	Список вкладчиков, имеющих указанный рублевый вклад, и сумму операций в USD	Сумма операции в USD	Название вклада Курс USD, руб.
7	Список клиентов, бронировавших указанный номер и стоимость проживания в USD	Стоимость проживания в USD	Номер комнаты Курс USD, руб.
8	Список клиентов, воспользовавшихся указанной фотоуслугой, и ее стоимость в USD	Стоимость заказа в USD	Название фотоуслуги Курс USD, руб.
9	Список квартиросъемщиков, пользующихся указанной коммунальной услугой, и стоимость этой услуги в USD	Стоимость оплаты услуги в USD	Название услуги Курс USD, руб.
10	Список клиентов, бравших напрокат указанный товар, и стоимость проката в USD	Стоимость проката в USD	Название товара Курс USD, руб.
11	Список студентов, пропустивших указанный вид занятий и стоимость отработки в USD	Стоимость отработки в USD	Вид занятия Курс USD, руб.
12	Список студентов и их рейтинговую оценку по указанному предмету, увеличенную на заданный процент	Рейтинг, увеличенный на заданный процент	Название предмета Процент увеличения
13	Список клиентов, заказавших указанную пиццу, и ее стоимость в USD	Стоимость заказа в USD	Название пиццы Курс USD, руб.
14	Список клиентов, купивших путевки по указанному маршруту, и их стоимость в USD	Стоимость путевок в USD	Название маршрута Курс USD, руб.
15	Список клиентов, бравших на прокат указанный видеоматериал, и стоимость проката в USD	Стоимость проката в USD	Название видеоматериала Курс USD, руб.

5. Запрос с групповой операцией AVG.

В режиме *Конструктора* на основе базовой таблицы и одного поля из нее сформируйте запрос на выборку данных, который позволяет определить среднее значение для данного поля *по варианту*.

Используйте для поля по варианту *групповую операцию Avg*.

Для поля измените свойство *Подпись*.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *15 Групповая операция Avg*.

Вариант 1. Определите средний должностной оклад.

Вариант 2. Определите среднюю продолжительность переговоров.

Вариант 3. Определите среднюю цену товара.

Вариант 4. Определите среднюю почасовую ставку.

Вариант 5. Определите среднюю цену книг.

Вариант 6. Определите среднюю сумму взноса.

Вариант 7. Определите среднюю стоимость проживания за 1 сутки.

Вариант 8. Определите среднюю стоимость фотоуслуги.

Вариант 9. Определите среднюю площадь квартиры.

Вариант 10. Определите среднюю стоимость проката за сутки.

Вариант 11. Определите среднее число часов пропусков.

Вариант 12. Определите среднюю рейтинговую оценку.

Вариант 13. Определите среднюю цену пиццы.

Вариант 14. Определите среднюю цену путевки.

Вариант 15. Определите среднюю залоговую стоимость видеоматериала.

6. Запрос с операцией SUM.

В режиме *Конструктора* на основе базовых таблиц сформируйте запрос выборки с *групповой операцией SUM*, который позволяет подвести общие итоги для группы записей по варианту.

Для поля со статистической функцией измените свойство *Подпись*.

Выполните сортировку по полю со статистической функцией.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *16 Итоги с операцией SUM*.

Например, сформируйте список сотрудников и подсчитайте общее число товаров, проданных каждым из них (рис. 5.3).

Для поля *Количество* задайте для свойства *Подпись* значение *Продано*.

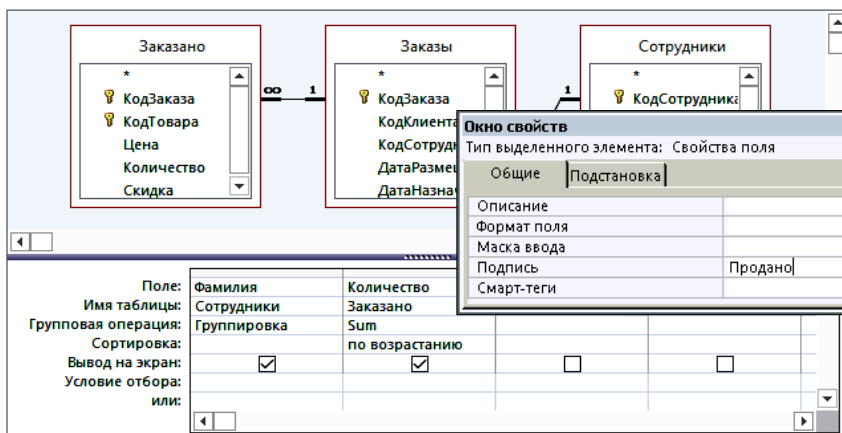


Рис. 5.3. Макет запроса с групповой операцией SUM

Внимание: в запросы с групповыми операциями включаются обычно 2 поля, по первому из которых выполняется группировка, а по второму применяется статистическая функция.

Данные в запросах с групповыми операциями обычно сортируют по полю со статистической функцией.

Вариант 1. Вывести список сотрудников и суммарное количество часов, отработанных каждым из них.

Вариант 2. Вывести список операторов и суммарную продолжительность разговоров для каждого из них.

Вариант 3. Вывести список фирм-клиентов и суммарное количество товаров, приобретенных каждой фирмой.

Вариант 4. Вывести список преподавателей и суммарное количество часов, отработанных каждым из них.

Вариант 5. Вывести список читателей и суммарную стоимость книг, выданных каждому из них.

Вариант 6. Вывести список вкладчиков и общую сумму взносов, сделанных каждым из них.

Вариант 7. Вывести список клиентов и суммарную стоимость проживания для каждого из них.

Вариант 8. Вывести список клиентов и суммарную стоимость фотослуг, оказанных каждому из них.

Вариант 9. Вывести список квартиросъемщиков и суммарную стоимость оплаченных коммунальных услуг для каждого из них.

Вариант 10. Вывести список клиентов и суммарную стоимость товаров, выданных напрокат каждому из них.

Вариант 11. Вывести список студентов и суммарное число часов пропусков каждого из них.

Вариант 12. Вывести список студентов и суммарную рейтинговую оценку каждого из них.

Вариант 13. Вывести список клиентов и суммарную стоимость пиццы, заказанной каждым из них.

Вариант 14. Вывести список клиентов и суммарную стоимость путевок, приобретенных каждым из них.

Вариант 15. Вывести список клиентов и суммарную стоимость видеоматериалов, выданных на прокат каждому из них.

7. Запрос с операцией COUNT.

В режиме *Конструктора* на основе базовых таблиц сформируйте запрос выборки с *групповой операцией COUNT* (по полю *первичного ключа таблицы-связки*), который позволяет подвести общие итоги для групп записей по варианту.

Для поля со статистической функцией измените свойство поля *Подпись*.

Выполните *сортировку* по полю со статистической функцией.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *17 Итоги с операцией COUNT*.

Вариант 1. Определите число проектов, в которых принял участие каждый сотрудник.

Вариант 2. Определите число звонков, сделанных каждым абонентом.

Вариант 3. Определите число заказов, сделанных каждой фирмой-клиентом.

Вариант 4. Определите число занятий, которые провел каждый преподаватель.

Вариант 5. Определите число выдач книг каждого читателя.

Вариант 6. Определите число операций, сделанных каждым вкладчиком.

Вариант 7. Определите число обращений каждого клиента по поводу бронирования места.

Вариант 8. Определите число обращений каждого клиента в фотоателье.

Вариант 9. Определите число оплат коммунальных услуг, выполненных каждым квартиросъемщиком.

Вариант 10. Определите число обращений каждого клиента в салон проката.

Вариант 11. Определите число пропусков занятий, отмеченных для каждого студента.

Вариант 12. Определите, число рейтинговых оценок, выставленных каждому студенту (по всем предметам).

Вариант 13. Определите число заказов, сделанных каждым клиентом.

Вариант 14. Определите число обращений каждого клиента в туристическое бюро.

Вариант 15. Определите число заказов, сделанных каждым клиентом видеосалона.

8. Перекрестный запрос 1.

В режиме *Мастера* «Перекрестный запрос», используя как источник запрос «02 Исходные данные», сформируйте *перекрестный запрос*, содержащий поля в соответствии с вариантом (табл. 5.3).

Таблица 5.3

Исходные данные для перекрестного запроса 1

Вариант	Поля источника запроса		
	Заголовки строк	Заголовки столбцов	Результирующее значение
1	2	3	4
1	Фамилия сотрудника	Название проекта	Суммарное число отработанных дней
2	Фамилия оператора	Тип звонка	Суммарная продолжительность переговоров
3	Название фирмы-клиента	Название товара	Суммарное число проданных товаров
4	Фамилия преподавателя	Название предмета	Суммарное число отработанных часов

1	2	3	4
5	Фамилия читателя	Название книги	Суммарная цена книг
6	Фамилия вкладчика	Название вклада	Число операций
7	Фамилия клиента	Номер комнаты	Количество заселений
8	Фамилия клиента	Наименование фотоуслуги	Число заказов
9	Фамилия квартиросъемщика	Вид услуги	Количество оплат
10	Фамилия клиента	Название товара	Количество обращений в прокат
11	Фамилия студента	Название предмета	Суммарное число пропущенных часов
12	Фамилия студента	Название предмета	Суммарная рейтинговая оценка
13	Фамилия клиента	Название пиццы	Суммарное количество заказанных порций
14	Фамилия клиента	Название маршрута	Суммарное количество проданных путевок
15	Фамилия клиента	Название видеоматериала	Число заказов

Для каждой строки вычислите итоговые значения (установив флажок в диалоговом окне «Мастер перекрестного запроса»).

В отчете приведите:

– диалоговое окно «Мастер перекрестного запроса» с макетом запроса по варианту (*проверьте соответствие статистической функции и поля для итоговых вычислений по варианту*);

– результат выполнения запроса;

– макет перекрестного запроса в режиме *Конструктора*.

Замените имя запроса по умолчанию именем *18 Перекрестный запрос 1*.

9. *Перекрестный запрос 2.*

В режиме *Конструктора* самостоятельно постройте перекрестный запрос, аналогичный запросу *18 Перекрестный запрос 1*, поменяв местами поля в заголовках строк и столбцов, результирующие значения оставьте в соответствии с вариантом (табл. 5.3).

Для каждой строки вычислите итоговые значения.

Задайте по итоговому полю сортировку по убыванию.

В отчете приведите:

– макет запроса в режиме *Конструктора*;

– результат выполнения перекрестного запроса.

Замените имя запроса по умолчанию именем *19 Перекрестный запрос 2*.

8. Для отчета подготовьте распечатку окна БД, вкладка *Запросы* (список созданных запросов в режиме *Просмотра сведений*).

9. Оформите отчет о выполнении работы.

Лабораторная работа № 6

ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ MS ACCESS В ЗАПРОСАХ ВЫБОРКИ



Цель работы: – изучить категории стандартных функций MS Access и особенности работы с ними;
– приобрести практический опыт использования стандартных функций в запросах СУБД MS Access.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи по варианту.
3. Макеты запросов в режиме *Конструктора* (при необходимости дополнительно приведите окно *Параметров запросов*, окно *Свойства поля*, окно *Свойства запроса*).
4. Результаты выполнения запросов.
5. Окно БД, вкладка *Запросы* (список созданных запросов в режиме *Таблицы*)
6. Краткие ответы на контрольные вопросы.
7. Выводы.
8. Список использованной литературы.



Контрольные вопросы

1. Дайте определение понятия «функция» MS Access. Что понимают под синтаксисом функции? Перечислите стандартные категории функций СУБД MS Access (назначение, примеры функций каждой категории).
2. Опишите функции из категории *Управление* (назначение, синтаксис, аргументы). Приведите примеры их использования.
3. Что представляют собой значения даты и времени в СУБД MS Access? Какие операции можно выполнять над значениями типа дата и время?
4. Какие функции возвращают значение текущей даты и текущего времени? Приведите примеры их практического применения.

5. В виде таблицы опишите функции (с аргументами), предназначенные для извлечения определенной части из значения даты (день, месяц, квартал, год), приведите примеры записи выражений и результат вычисления для даты выполнения задания.

6. Опишите алгоритм для определения возраста (стажа работы) по известной дате рождения (дате найма), приведите примеры записи вычисляемых выражений.



Порядок выполнения работы



Внимание. До начала выполнения работы необходимо:

1. Внесите *изменения в таблицы БД:*

– в БД должна присутствовать таблица *Сотрудники*, *обязательно* содержащие поля *Пол* (жен, муж) и *Дата найма* (на работу). Если таблица отсутствует – *импортируйте из учебной БД Борей* в индивидуальную БД;

– предусмотрите в БД сотрудника (клиента, студента, читателя и т. д.) с *датой рождения* (*датой найма* для работника) в *декабре* месяце;

– в *таблице-связке* проверьте наличие *внешнего ключа Код сотрудника*. При необходимости добавьте поле в таблицу-связку. С помощью *Мастера подстановок* сформируйте список для поля внешнего ключа и в режиме *Таблицы* заполните значения поля *Код сотрудника* в таблице-связке.

2. Внесите *изменения в схему БД:* проверьте наличие связей (при необходимости установите связи) между таблицей-связкой и таблицей *Сотрудники*.

3. *Изучите стандартные функции MS Access из категории Дата/Время* (табл. 6.1) и особенности их использования.

Таблица 6.1

Стандартные функции MS Access из категории Дата/Время

Функция (аргументы)	Результат	Примеры записи
1	2	3
Day (дата)	Целое число, день месяца (1...31)	Day([Заказы]![Дата])

1	2	3
Month (дата)	Целое число, номер месяца (1...12)	Month([Заказы]![Дата])
Year (дата)	Целое число, год (100...9666)	Year([Заказы]![Дата])
Weekday (дата [;первый_день_недели])	Целое число, номер дня недели (1...7) Нумерация первого дня определяется аргументом <i>первый_день недели</i> : 1 – Воскресенье, 2 – Понедельник	Weekday([Заказы]![Дата];2)
WeekdayName (день_недели [;аббр_имя [;первый_день_недели])	Строка – полное или краткое название дня недели (Понедельник...Воскресенье или Пн...Вс), определяется аргументом <i>аббр_имя</i>	WeekdayName (Weekday([Заказы]![Дата];2))
DatePart (“интервал”; дата [; <i>первый_день_недели</i>] [; <i>первая_неделя_года</i>])	Целое число, часть <i>даты</i> (определяется аргументом <i>интервал</i>): “уууу” Год “q” Квартал “m” Месяц “y” День года “d” День “w” День недели “ww” Неделя	DatePart (“q”; [Заказы]![Дата])

4. В БД по варианту создайте следующие запросы:

4.1. *Сведения за текущий месяц текущего года.*

В режиме *Конструктора* на основе базовых таблиц сформируйте запрос выборки данных *за текущий месяц текущего года* (по варианту). Состав базовых таблиц и полей, необходимых для запроса, определите самостоятельно.

Для решения задачи создайте два вычисляемых поля: *Месяц*, *Год* (используйте функции *Month()*, *Year()*, в качестве аргумента кото-

рых задайте поле с именем *Дата...* из таблицы-связки. Например, *Дата заказа* из таблицы *Заказы*).

В качестве условия отбора для вычисляемых полей используйте функции *Month()*, *Year()* с аргументом *Date()* для определения текущего месяца и года соответственно.

Поле для сортировки и порядок сортировки выберите самостоятельно.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *20 Сведения за текущий месяц текущего года*.

Вариант 1. Сформируйте список сотрудников, работавших в заданный период.

Вариант 2. Сформируйте список операторов, выполнивших исходящие звонки в заданный период.

Вариант 3. Сформируйте список товаров, проданных в заданный период.

Вариант 4. Сформируйте список преподавателей, проводивших занятия в заданный период.

Вариант 5. Сформируйте список книг, выданных читателям в заданный период.

Вариант 6. Сформируйте список вкладчиков, выполнявших операции по вкладам в заданный период.

Вариант 7. Сформируйте список номеров комнат, заселенных в заданный период.

Вариант 8. Сформируйте список фотоуслуг, заказанных в заданный период.

Вариант 9. Сформируйте список коммунальных услуг, оплаченных в заданный период.

Вариант 10. Сформируйте список товаров, выданных напрокат в заданный период.

Вариант 11. Сформируйте список студентов, пропускавших занятия в заданный период.

Вариант 12. Сформируйте список студентов, получивших рейтинговые оценки в заданный период.

Вариант 13. Сформируйте список названий пиццы, заказанной в заданный период.

Вариант 14. Сформируйте список маршрутов, путевки на которые были проданы в заданный период.

Вариант 15. Сформируйте список видеоматериалов, выданных напрокат в заданный период.

4.2. *Сведения за текущий квартал текущего года.*

В режиме *Конструктора* на основе базовых таблиц сформируйте запрос выборки данных за *текущий квартал текущего года* (по вариантам). Состав базовых таблиц и полей, необходимых для запроса, определите самостоятельно.

Для решения задачи создайте два вычисляемых поля: *Квартал*, *Год* (функции *DatePart()*, *Year()*, *Date()*).

По вычисляемым полям задайте условия отбора.

Поле для сортировки и порядок сортировки выберите самостоятельно.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *21 Сведения за текущий квартал текущего года*.

Вариант 1. Сформируйте список сотрудников, работавших в заданный период.

Вариант 2. Сформируйте список операторов, выполнивших исходящие звонки в заданный период.

Вариант 3. Сформируйте список товаров, проданных в заданный период.

Вариант 4. Сформируйте список преподавателей, проводивших занятия в заданный период.

Вариант 5. Сформируйте список книг, выданных читателям в заданный период.

Вариант 6. Сформируйте список вкладчиков, выполнявших операции по вкладам в заданный период.

Вариант 7. Сформируйте список номеров комнат, заселенных в заданный период.

Вариант 8. Сформируйте список фотоуслуг, заказанных в заданный период.

Вариант 9. Сформируйте список коммунальных услуг, оплаченных в заданный период.

Вариант 10. Сформируйте список товаров, выданных напрокат в заданный период.

Вариант 11. Сформируйте список студентов, пропускавших занятия в заданный период.

Вариант 12. Сформируйте *список студентов*, получивших рейтинговые оценки в заданный период.

Вариант 13. Сформируйте *список названий пиццы*, заказанной в заданный период.

Вариант 14. Сформируйте *список маршрутов*, путевки на которые были проданы в заданный период.

Вариант 15. Сформируйте *список видеоматериалов*, выданных напрокат в заданный период.

4.3. *Сведения за указанный квартал указанного года.*

В режиме *Конструктора* на основе базовых таблиц сформируйте *запрос с 2-мя параметрами* для выбора данных за *квартал года* (по варианту). Состав базовых таблиц и полей, необходимых для запроса, определите самостоятельно.

Для решения задачи создайте два вычисляемых поля: *Квартал*, *Год* (функции *DatePart()*, *Year()*).

По вычисляемым полям *в условиях отбора* задайте *параметры*: *[Введите год]*, *[Введите квартал]*.

Поле для сортировки и порядок сортировки выберите самостоятельно.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *22 Сведения за квартал года*.

Вариант 1. Сформируйте *список сотрудников*, работавших в заданный период.

Вариант 2. Сформируйте *список операторов*, выполнивших исходящие звонки в заданный период.

Вариант 3. Сформируйте *список товаров*, проданных в заданный период.

Вариант 4. Сформируйте *список преподавателей*, проводивших занятия в заданный период.

Вариант 5. Сформируйте *список книг*, выданных читателям в заданный период.

Вариант 6. Сформируйте *список вкладчиков*, выполнявших операции по вкладам в заданный период.

Вариант 7. Сформируйте *список номеров комнат*, заселенных в заданный период.

Вариант 8. Сформируйте *список фотоуслуг*, заказанных в заданный период.

Вариант 9. Сформируйте *список коммунальных услуг*, оплаченных в заданный период.

Вариант 10. Сформируйте *список товаров*, выданных напрокат в заданный период.

Вариант 11. Сформируйте *список студентов*, пропускавших занятия в заданный период.

Вариант 12. Сформируйте *список студентов*, получивших рейтинговые оценки в заданный период.

Вариант 13. Сформируйте *список пиццы*, заказанной в заданный период.

Вариант 14. Сформируйте *список маршрутов*, путевки на которые были проданы в заданный период.

Вариант 15. Сформируйте *список видеоматериалов*, выданных напрокат в заданный период.

4.4. Работа по дням недели.

В режиме *Конструктора* на основе базовых таблиц сформируйте запрос выборки, позволяющий проанализировать работу по дням недели (по варианту).

Для решения задачи создайте 2 вычисляемых поля:

1) вычисляемое поле *№ дня недели* (со значениями 1–7), по которому при подведении итогов будет выполняться *группировка* и *сортировка*;

2) вычисляемое поле *День недели*, по которому будет выполняться *группировка*.

Поле для подведения итогов и статистическую функцию выберите самостоятельно (по варианту).

Для поля со статистической функцией измените свойство *Подпись*.

Выполните сортировку по порядковому номеру дня недели.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *23 Работа по дням недели*.

Вариант 1. Определите, сколько проектов начиналось в каждый день недели.

Вариант 2. Определите продолжительность звонков в каждый день недели.

Вариант 3. Определите число покупок, сделанных в каждый день недели.

Вариант 4. Определите число часов проведенных занятий в каждый день недели.

Вариант 5. Определите число книг, выданных в каждый день недели.

Вариант 6. Определите сумму взносов, сделанных в каждый день недели.

Вариант 7. Определите число обращений по поводу бронирования места в каждый день недели.

Вариант 8. Определите число обращений в фотоателье в каждый день недели.

Вариант 9. Определите число коммунальных услуг, оплаченных в каждый день недели.

Вариант 10. Определите число обращений в прокат в каждый день недели.

Вариант 11. Определите общее число часов пропусков в каждый день недели.

Вариант 12. Определите количество рейтинговых оценок, выставленных в каждый день недели.

Вариант 13. Определите число порций, заказанных в каждый день недели.

Вариант 14. Определите число обращений в туристическое бюро в каждый день недели.

Вариант 15. Определите число заказов, сделанных в каждый день недели.

4.5. Возраст.

В режиме *Конструктора* на основе базовой таблицы сформируйте запрос с вычисляемым полем, позволяющий рассчитать возраст (число полных лет на текущую дату) сотрудника (клиента, студента, читателя и т. д.). Состав полей, необходимых для запроса, определите самостоятельно.

Для решения задачи создайте вычисляемое поле *Возраст* (для расчета используйте функции *If()*, *DatePart()*, *Date()*).

Порядок сортировки задайте по полю *Возраст*.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *24 Возраст*.

4.6. Стаж работы.

В режиме *Конструктора* на основе базовой таблицы *Сотрудники* (*Операторы*, *Преподаватели* и так далее по варианту) сформируйте

запрос с вычисляемым полем, позволяющий рассчитать стаж работы (число полных лет на текущую дату) сотрудника (оператора, преподавателя и т. д.).

Решение задачи выполните аналогично запросу *24 Возраст*.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *25 Стаж*.

4.7. Пенсионеры.

В режиме *Конструктора* на основе базовой таблицы *Сотрудники* сформируйте запрос с вычисляемым полем, позволяющий рассчитать возраст сотрудника и вывести фамилии тех, кто достиг пенсионного возраста.

Для решения задачи создайте вычисляемое поле *Возраст*.

Задайте условия отбора по двум полям: *Пол* и *Возраст*, учитывая актуальный пенсионный возраст для женщин и мужчин.

Порядок сортировки задайте по полю *Возраст*.

Выполните запрос (отобразите результирующую таблицу).

Сохраните запрос с именем *26 Пенсионеры*.

5. Задания для самостоятельной работы.

5.1. Определите день недели, в который родился каждый сотрудник. Имя запроса – *27 День рождения*.

Примените 2 способа решения задачи в одном запросе:

1) для поля *Дата рождения* создайте пользовательский формат вида *dd.mm.yyyy, ddd*;

2) создайте отдельное вычисляемое поле *День рождения: Week-DayName(...)*, аргументы функции задайте таким образом, чтобы выводилось полное название дня недели.

5.2. Определите, сколько лет осталось до пенсии каждому сотруднику. Имя запроса – *28 До пенсии осталось*.

Создайте вычисляемое поле *До пенсии осталось*. При вычислениях учитывайте, что возраст выхода на пенсию для женщин и мужчин различен: используйте для проверки пола сотрудника функцию управления *Иф()*, а также функции из категории *Дата/время* для расчета возраста (см. задание в п. 4.5 *Возраст*).

5.3. Выведите список сотрудников, которые выйдут на пенсию в течение ближайших 5 лет. Имя запроса – *29 До пенсии осталось меньше 5 лет*.

Для решения задачи используйте предыдущий запрос, дополнительно задайте условие отбора по вычисляемому полю *До пенсии осталось*.

5.4. Выведите список сотрудников, которые отметят юбилей в следующем году. Имя запроса – *30 Юбилей в следующем году*.

Для решения задачи создайте вычисляемое поле *Юбилей*, в котором с использованием функции `Year ()` рассчитайте возраст сотрудника в следующем году. При решении задачи юбилейным считайте возраст, кратный 5.

5.5. Выведите список заказов, сформированных в течение предыдущего месяца (квартала) от текущей даты.

Имена запросов – *31 Заказы в течение месяца*, *32 Заказы в течение квартала*.

Для решения задачи используйте функцию `DateAdd(...)` с соответствующими аргументами, определяющими интервал времени: “m”, “q” (см. табл. 6.1).

6. Для отчета подготовьте распечатку окна БД, вкладка *Запросы* (список созданных запросов в режиме *Просмотра сведений*).

7. Оформите отчет о выполнении работы.

Лабораторная работа № 7

ПРОЕКТИРОВАНИЕ И ИСПОЛЬЗОВАНИЕ ТАБЛИЧНЫХ ФОРМ



Цель

работы:

- изучить назначение и особенности создания табличных форм;
- приобрести практический опыт разработки табличных форм для ввода информации в базовые таблицы БД;
- приобрести практические навыки использования и форматирования элементов управления в формах MS Access.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи по варианту.
3. Макеты форм в режиме Конструктора (перечень и свойства используемых элементов управления).
4. Вид каждой формы в режиме Формы.
5. Окно БД, вкладка Формы (список созданных табличных форм).
6. Список контрольных вопросов и ответов на них.
7. Выводы.
8. Список использованных источников.



Контрольные вопросы

1. Сформулируйте назначение форм MS Access, перечислите их виды. Назовите режимы создания форм и приведите их особенности.
2. Опишите структуру (разделы) формы. Поясните особенности настройки свойств формы и разделов формы.
3. Перечислите типы элементов управления, используемых в формах, сформулируйте их назначение, приведите примеры. Приведите порядок создания и форматирования элементов управления, настройки их свойств.

4. Опишите порядок подготовки формы к печати: настройку параметров страницы и режимов печати формы (приведите вкладки окна с настройками параметров печати).



Порядок выполнения работы

1. Автоформа.

На вкладке БД *Таблицы* выделите таблицу по варианту (табл. 7.1) и создайте табличную автоформу с помощью кнопки «Форма» на вкладке *Создание* в группе команд *Формы*.

Таблица 7.1

Исходные данные для построения Автоформы

Вариант	Таблица/название формы	Вариант	Таблица/название формы
1	Должности	9	Коммунальные услуги
2	Абоненты	10	Товары
3	Товары	11	Предметы
4	Предметы	12	Предметы
5	Книги	13	Пицца
6	Вклады	14	Маршруты
7	Номера (комнаты)	15	Видеокассеты
8	Фотоуслуги		

Изучите и приведите в отчете структуру созданной Автоформы в режиме *Формы* (рис. 7.1) и в режиме *Конструктора* (рис. 7.2).

Добавьте в базовую таблицу одну запись с помощью разработанной Автоформы.

Сохраните форму с именем *01Автоформа*.

Борей

Файл Главная Создание Внешние данные Работа с базами данных Помощь Вход

Товары1

Товары

Код товара: 1

Марка: Genen Shouyu

Поставщик: Mayumi's

Тип: Приправы

Единица измерения: 24 бутылки по 250 мл

Цена: 697,50р.

На складе: 39

Ожидается: 0

Минимальный запас: 5

Поставки Прекращены

Код заказ	Цена	Количество	Скидка
10285	144,00р.	45	20%
10294	144,00р.	18	0%
10317	144,00р.	20	0%
10348	144,00р.	15	15%
10354	144,00р.	12	0%
10370	144,00р.	15	15%
10406	144,00р.	10	0%
10413	144,00р.	24	0%
10477	144,00р.	15	0%

Область навигации

Запись: 1 из 38

Нет фильтра

Поиск

Запись: 1 из 77

Нет фильтра

Уникальное число, автоматически присваиваемое новой записи.

Num Lock

Рис. 7.1. Автоформа (в режиме *Формы*)

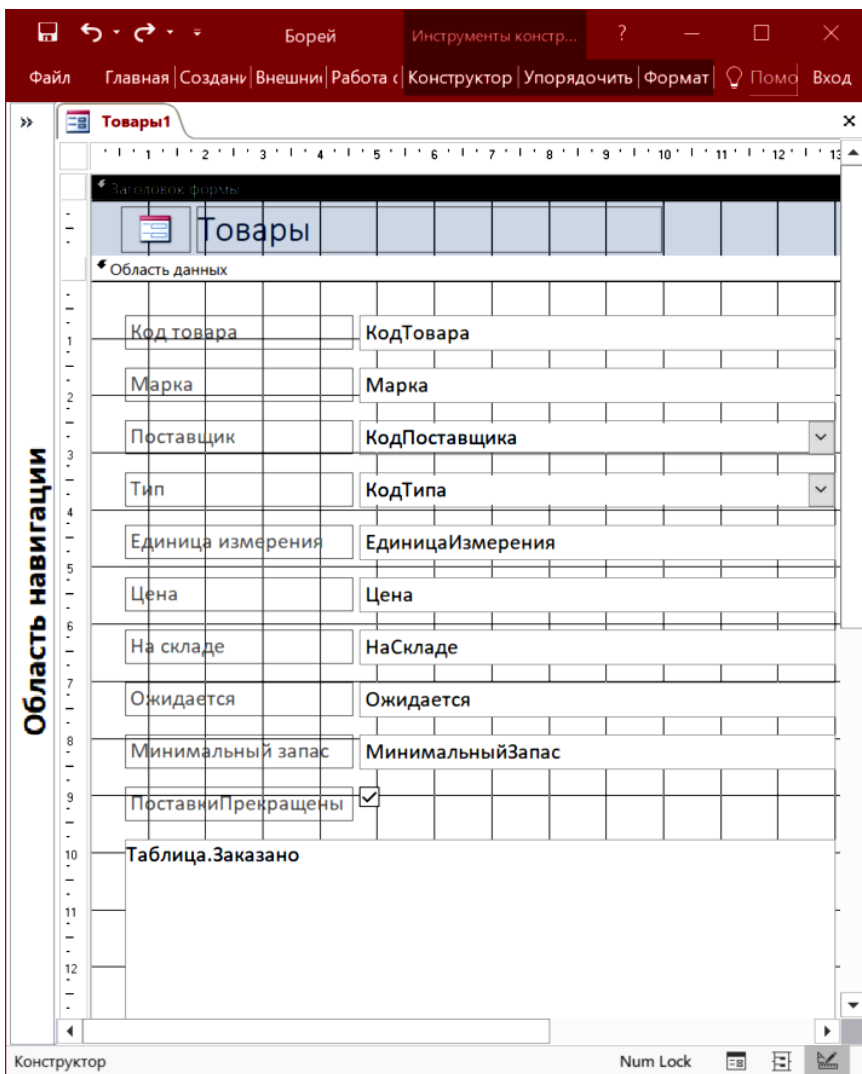


Рис. 7.2. Автоформа (в режиме *Конструктора*)

2. *Форма с элементом управления Поле со списком.*

В режиме *Мастера форм* создайте форму для ввода информации в подчиненную базовую таблицу (таблицу-связку).

В форму включите *все поля из таблицы-связки*.

Выберите *внешний вид формы* – выровненный.

Сохраните форму с именем *02 Форма с ЭУ Поле со списком*.

Выполните в режиме *Формы* ввод 2-х новых записей (ввод данных в поля внешних ключей из соответствующих связанных таблиц должен выполняться с использованием элемента управления *Поле со списком*).

Приведите в отчете макет созданной формы в режиме *Конструктора* и в режиме *Формы*.

3. *Форма с объектом OLE*.

Внимание: до построения формы добавьте в соответствующую базовую таблицу дополнительное поле с типом данных *поле Объекта OLE* по варианту (табл. 7.2).

Таблица 7.2

Исходные данные для построения *Формы с объектом OLE*

Вариант	Имя добавляемого поля	Вариант	Имя добавляемого поля
1	Фотография сотрудника	9	Фотография квартиросъемщика
2	Фотография оператора	10	Фотография клиента
3	Фотография руководителя фирмы-клиента	11	Фотография студента
4	Фотография преподавателя	12	Фотография студента
5	Фотография читателя	13	Фотография клиента
6	Фотография вкладчика	14	Фотография клиента
7	Фотография клиента	15	Фотография клиента
8	Фотография клиента		

В режиме *Конструктора форм* создайте табличную форму для работы с базовой таблицей с учетом дополнительного поля.

Используйте команду *Конструктор форм* на вкладке *Создание*, указав источник формы – базовую таблицу.

Для вывода *списка полей* в режиме *Конструктора* формы используйте команду *Добавить поля* на вкладке *Конструктор* в группе команд *Сервис*.

Размещение полей таблицы в области данных формы выполните следующим образом:

- в списке полей выделите *все поля* (ЛКМ + «Shift»);
- ЛКМ перетащите выделенные поля в область данных формы.

Для *позиционирования* и *форматирования* элементов управления в области данных формы используйте команды вкладки *Упорядочить* или команды *Выровнять* и *Размер* из контекстного меню, а также размерные маркеры элементов управления.

Для изменения *шрифта* и *начертания символов, границ полей* и их значений используйте команды группы *Шрифт* на вкладке *Формат* набора *Конструктора форм*.

Измените Свойства формы на вкладке *Макет* (рис. 7.3):

- подпись формы – название базовой таблицы;
- запретите режим *Таблицы*;
- уберите полосы прокрутки (контролируйте размер формы);
- уберите область выделения и разделительные линии;
- установите тип границы – тонкая (чтобы запретить изменение с помощью ЛКМ размеров формы в режиме *Формы*. Предварительно настройте размеры формы).

Сохраните форму с именем *03 Форма с объектом OLE*.

Заполните в режиме *Формы* поле объекта OLE для 1-й записи базовой таблицы.

Приведите в отчете макет созданной формы в режиме *Конструктора* и запись с добавленным объектом в режиме *Формы*.

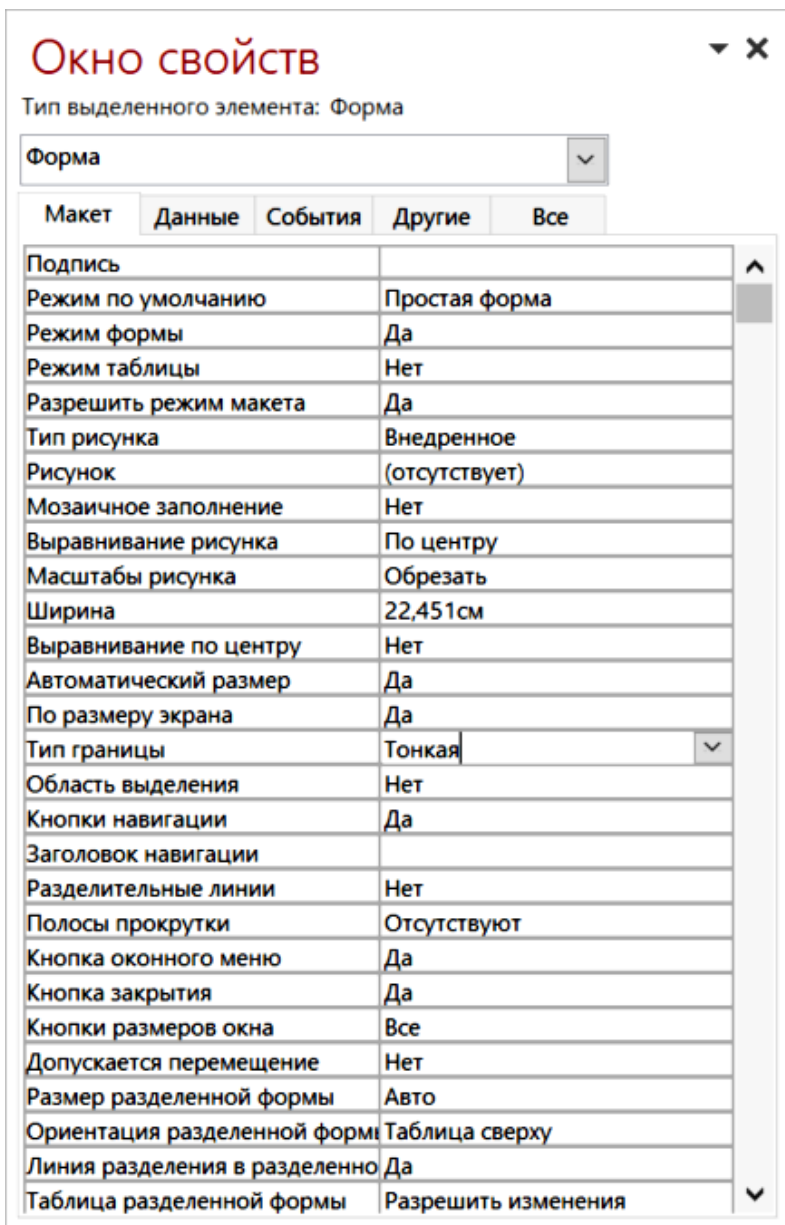


Рис. 7.3. Свойства табличной формы

4. Форма с диаграммой.

Для наглядного представления результатов анализа данных постройте форму с диаграммой на основе запроса *16 Итоги с операцией SUM*:

1. Создайте пустую форму, нажав кнопку «Пустая форма» в группе команд *Форма* на вкладке *Создание*.

2. Не закрывая созданную форму, перейдите в режим *Конструктора форм* и установите размер формы 10 × 10 см (в свойствах макета формы или с помощью ЛКМ).

3. В наборе вкладок *Инструменты конструктора форм* на вкладке *Конструктор форм* в группе *Элементы управления* выберите элемент управления *Диаграмма* и разместите ее в области данных по размеру пустой формы.

4. В открывшемся диалоговом окне «Создание диаграмм» выберите в качестве источника данных запрос *16 Итоги с операцией SUM*, нажмите кнопку «Далее».

5. На следующем шаге из списка полей запроса выберите 2 поля: текстовое – для оси категорий; числовое – для оси значений, нажмите кнопку «Далее».

6. В диалоговом окне «Создание диаграммы» выберите *тип диаграммы*, например, линейчатая (обоснуйте в отчете выбор типа диаграммы), нажмите кнопку «Далее».

7. В диалоговом окне «Создание диаграммы» при построении макета диаграммы выполните *двойной щелчок* ЛКМ по числовому полю (со статистической функцией Sum) на оси значений диаграммы и в списке диалогового окна «Вычисление итоговых значений» выберите вариант *Отсутствует* (рис. 7.4). Данная операция необходима для корректного отображения на диаграмме данных запроса с учетом выполненной в запросе сортировки.

После отмены итогов закройте окно, оцените диаграмму при помощи кнопки «Образец» и нажмите кнопку «Далее».

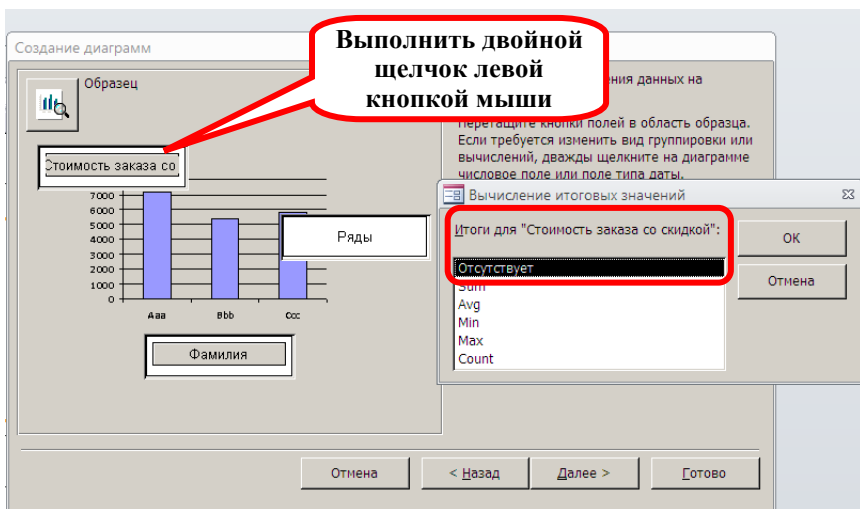


Рис. 7.4. Отмена итогов по числовому полю (SUM-Количество)

8. На последнем шаге построения диаграммы задайте ее название (по варианту, в соответствии с результатами запроса, *например, Рейтинг сотрудников по сумме заказов*) и завершите построение диаграммы.

9. Переключитесь в режим *Формы* и оцените полученный результат.

Сделайте выводы.

10. Далее переключитесь в режим *Конструктора* и при необходимости отредактируйте элементы диаграммы в окне приложения Microsoft Graph (рис. 7.5), открыв его двойным щелчком по диаграмме в режиме *Конструктора*.

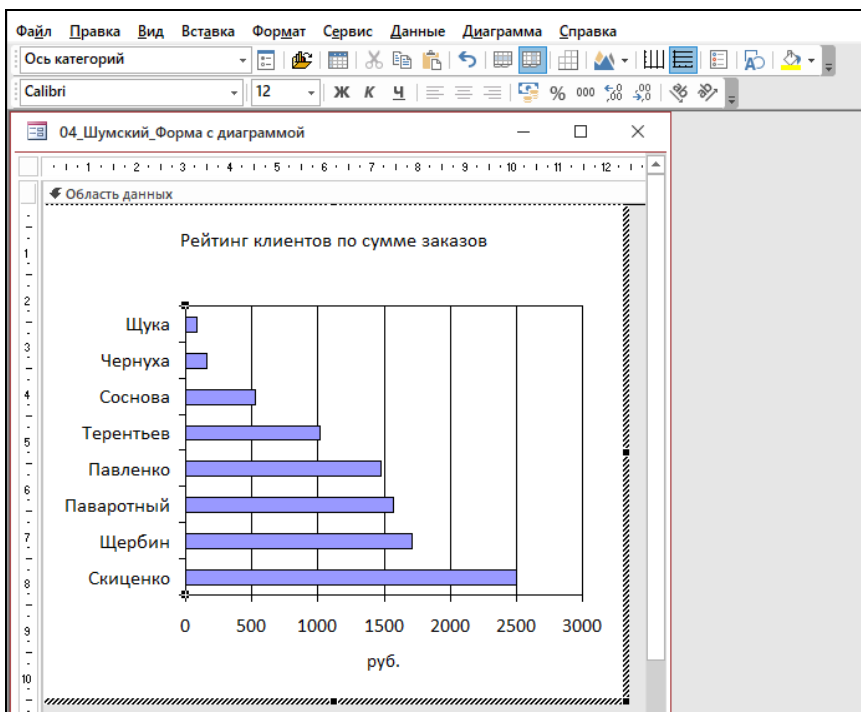


Рис. 7.5. Редактирование элементов диаграммы

11. В режиме *Конструктора* выполните настройку свойств формы с диаграммой (на вкладке *Макет*, рис. 7.6):

- *подпись формы* – Рейтинг... по ...;
- запретите режим *Таблицы*;
- уберите область выделения, разделительные линии, кнопки навигации (перехода по записям), полосы прокрутки и кнопки размеров окна (оставьте кнопку оконного меню и кнопку закрытия);
- подберите размер формы и задайте тип границы – *Тонкая*.

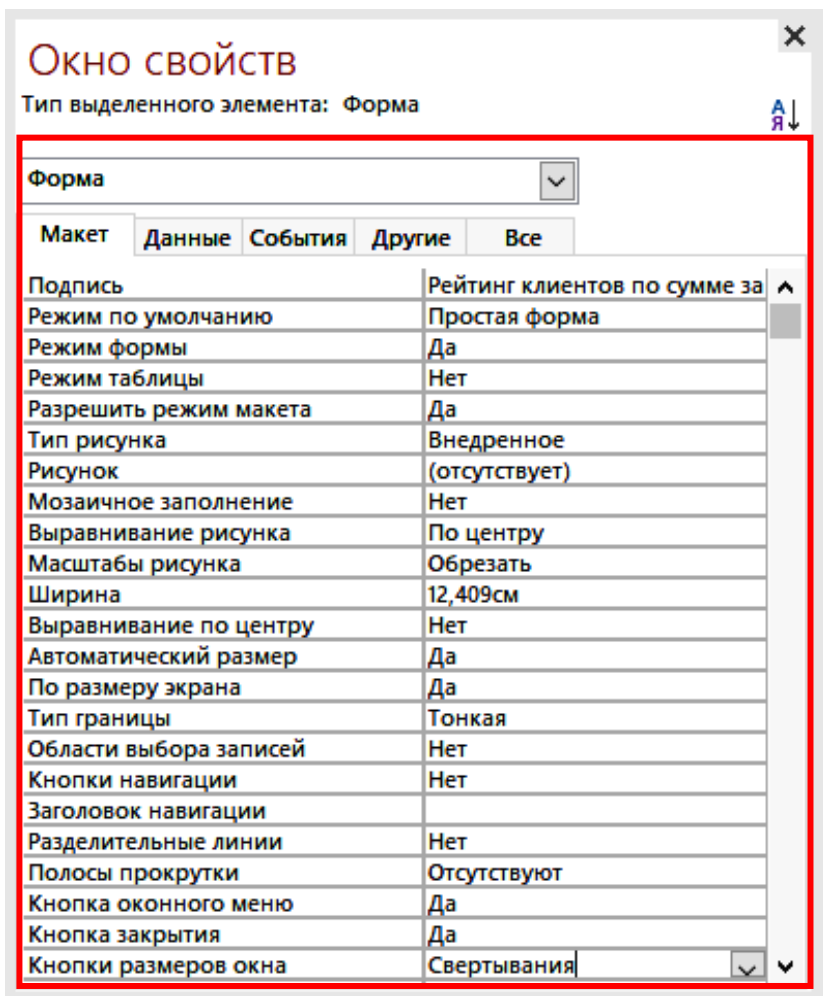


Рис. 7.6. Свойства формы с диаграммой

Примерный вид формы с диаграммой приведен на рис. 7.7.

Обратите внимание на отсутствие в построенной форме с диаграммой полос прокрутки и кнопок навигации, а также отсутствие кнопки изменения размеров окна (в ряду кнопок управления окном только 2 кнопки, стандартная кнопка *Свернуть в окно/Развернуть* отсутствует).

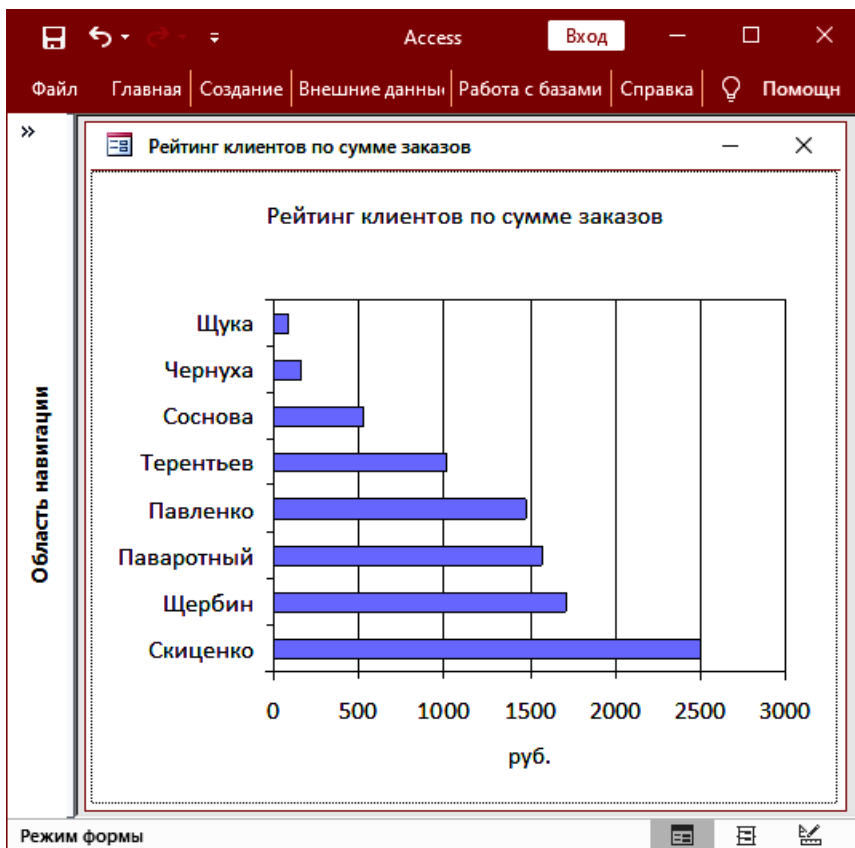


Рис. 7.7. Пример формы с диаграммой

Сохраните форму с именем *04 Форма с Диаграммой*.

Приведите в отчете макет созданной формы в режиме *Конструктора* и в режиме *Формы*.

5. Для отчета подготовьте распечатку окна БД, вкладка *Формы* (список созданных форм в режиме *Просмотра сведений*).

6. Оформите отчет о выполнении работы.

Лабораторная работа № 8

ПОДГОТОВКА ОТЧЕТОВ СРЕДСТВАМИ MS ACCESS



Цель работы:

- изучить назначение и особенности создания отчетов;
- приобрести практический опыт разработки отчетов для вывода данных с использованием сортировки и группировки записей, освоить нумерацию записей в отчете.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи по варианту.
3. Описание построения отчета (режим, уровни группировки, вид макета, итоги, стиль).
4. Вид каждого отчета (в режиме *Конструктора* и *Предварительного просмотра*).
5. Окно БД, вкладка *Отчеты* (список созданных отчетов).
6. Список контрольных вопросов и ответов на них.
7. Выводы.
8. Список использованных источников.



Контрольные вопросы

1. Сформулируйте назначение отчетов MS Access, перечислите их виды. Назовите режимы создания отчетов и приведите их особенности.
2. Опишите структуру (разделы) отчета MS Access в режиме *Конструктора*, приемы управления разделами отчета.
3. Приведите приемы группировки и сортировки записей в отчетах MS Access.
4. Опишите порядок создания и настройки свойств вычисляемых полей в отчетах MS Access. Как создать и настроить вычисляемое поле для нумерации записей отчета?
5. Назовите режимы просмотра и печати отчетов. Опишите порядок подготовки отчетов к печати, настройку параметров страницы и режимов печати отчета (приведите окно с настройками).

6. Опишите порядок подготовки отчетов MS Access к печати, настройку параметров страницы (приведите окно с настройками). Назовите режимы просмотра и печати отчетов.



Порядок выполнения работы

1. Автоотчет.

В качестве источника отчета на вкладке БД *Запросы* выделите запрос «23 Стаж» и сформируйте *Автоотчет* (рис. 8.1) с помощью кнопки «Отчет» на вкладке *Создание* в группе команд *Отчеты*.

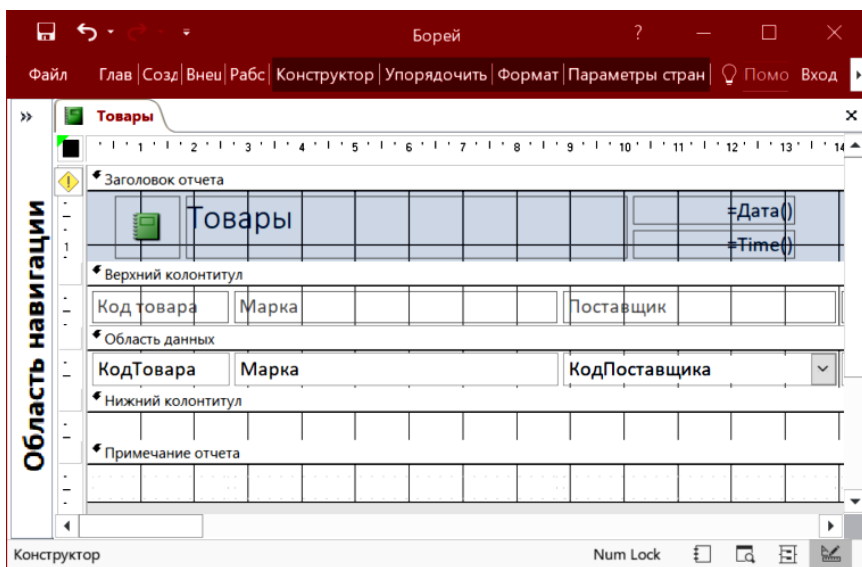


Рис. 8.1. Макет Автоотчета (в режиме *Конструктора*)

Перейдите в режим *Конструктора* и выполните дополнительное форматирование элементов управления:

- для названий полей измените шрифт, *например*, Times New Roman Суг, 10 пт, синего цвета;
 - для данных измените шрифт, *например*, Arial Суг, 9 пт;
 - задайте выравнивание данных по левому краю текстового поля.
- Сохраните отчет с именем *01 Автоотчет*.

2. Автоотчет выровненный.

В режиме *Мастер создания отчетов*, постройте Автоотчет на основе таблицы *Клиенты (Абоненты, Студенты, Читатели* и т. д. *по варианту)*, используя макет *выровненный* (уровни группировки и сортировку при построении не задавайте).

В режиме *Конструктора* выполните дополнительное форматирование элементов управления:

- для названий полей измените шрифт на Arial Cyr, 10 пт, полужирный, синего цвета;
- для данных измените шрифт на Arial Cyr, 10 пт, курсив, черного цвета;
- задайте выравнивание данных по левому краю текстового поля.

Сохраните отчет с именем *02 Автоотчет выровненный*.

3. Подробный итоговый отчет.

В режиме *Мастера отчетов* на основе запроса «13 Расчеты» сформируйте *подробный итоговый отчет*.

В отчет включите *все* поля запроса.

Группируйте данные по полю *Фамилия* (для *варианта 3* по полю *Название фирмы*).

Подсчитайте сумму и среднее по группам (окно приведите в отчете), а также *общий итог и среднее значение* по всем группам.

Результаты выведите в денежном формате или в числовом формате с точностью 2 знака после запятой.

Название отчета – «*Подробный итоговый отчет*».

В режиме *Конструктора пронумеруйте записи отчета*: создайте вычисляемое поле, настройте его свойства (см. рис. 4.36 и рис. 4.37 в п. 4.6 теоретического раздела пособия).

Окно свойств вычисляемого поля, макет отчета и полученный результат в режиме просмотра отчета приведите в отчете.

Сохраните отчет с именем *03 Подробный итоговый отчет*.

4. Почтовые наклейки.

В режиме *Мастера почтовых наклеек* сформируйте разновидность отчета: адресные этикетки для почтовой рассылки.

Размер наклейки, тип наклейки выберите самостоятельно.

Макет наклейки должен включать поля (рис. 8.2):

- Почтовый индекс;
- Страна, Населенный пункт;
- Почтовый адрес;
- Фамилия, Имя, Отчество.

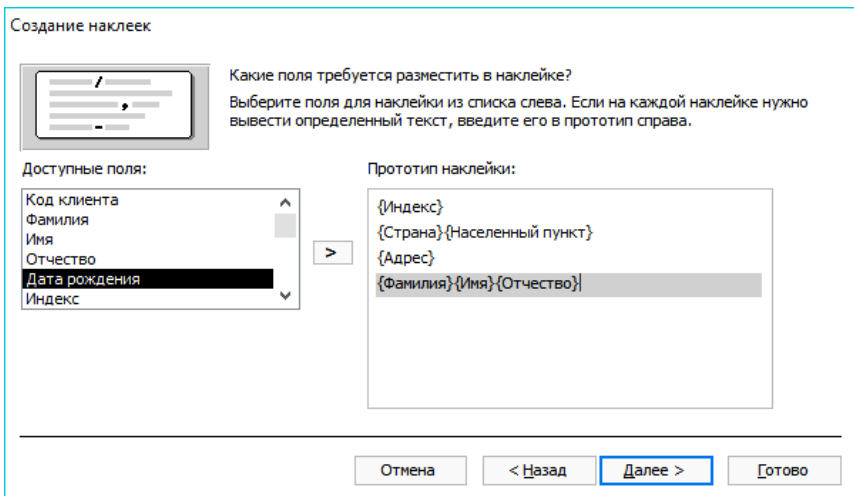


Рис. 8.2 Формирование макета почтовой наклейки

Сохраните отчет с именем *04 Почтовые наклейки*.

В режиме *Конструктора*:

- *отформатируйте поле с фамилией* – шрифт Arial, 10 пт, красного цвета, полужирный, курсив.

- оформите рамку для почтовой наклейки.

6. Для отчета подготовьте распечатку окна БД, вкладка *Отчеты* (список созданных отчетов в режиме *Просмотра сведений*).

7. Оформите отчет о выполнении работы.

Лабораторная работа № 9

РАЗРАБОТКА И НАСТРОЙКА ГЛАВНОЙ КНОПОЧНОЙ ФОРМЫ



Цель работы:

- изучить назначение и особенности создания кнопочных форм;
- приобрести практический опыт разработки кнопочных форм для управления БД;
- приобрести практические навыки настройки параметров запуска БД в MS Access.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи по варианту.
3. Проект главной кнопочной формы (ГКФ) в виде таблицы (перечень вкладок, кнопок и соответствующих им действий, имена макросов).
4. Макет ГКФ (1 вкладка) в режиме *Конструктора* и окно настройки свойств формы (отобразить вкладку *Макет*).
5. Вид ГКФ в режиме *Формы* (все вкладки)
6. Окно настройки параметров запуска БД (вкладка *Текущая база данных*).
7. Окно БД, вкладки *Формы* и *Макросы* (список созданных кнопочных форм и макросов в режиме *Таблицы*).
8. Список контрольных вопросов и ответов на них.
9. Выводы.
10. Список использованных источников.



Контрольные вопросы

1. Сформулируйте назначение и особенности кнопочных форм, перечислите способы их создания в СУБД MS Access.
2. Опишите порядок создания кнопочных форм в режиме *Конструктора*. Перечислите особенности настройки свойств кнопочной формы.

3. Перечислите элементы управления, применяемый в кнопочных формах. Приведите порядок создания кнопок и соответствующих им процедур обработки событий.

4. Сформулируйте назначение макросов и особенности их создания в MS Access. Опишите последовательность записи макрокоманд, задания аргументов и их значений (приведите примеры).

5. Приведите порядок настройка параметров запуска БД. В чем заключаются особенности макроса с именем *Autoexec*?



Порядок выполнения работы

1. В соответствии с вариантом разработайте проект *Главной кнопочной формы* (ГКФ) для управления объектами индивидуальной БД по различным срезам (считать, что срезу соответствует таблица БД).

Форма должна содержать логотип БД, кнопку выхода из БД; для управления данными по каждому срезу на вкладке предусмотрите не менее 4-х кнопок (рис. 9.1) для:

- просмотра базовой таблицы;
- открытия табличной формы для добавления записей в базовую таблицу;
- открытия запросов и/или формы с диаграммой с результатами анализа данных.

Внимание: при необходимости *дополнительно разработайте* запросы и формы для анализа данных по каждому срезу.

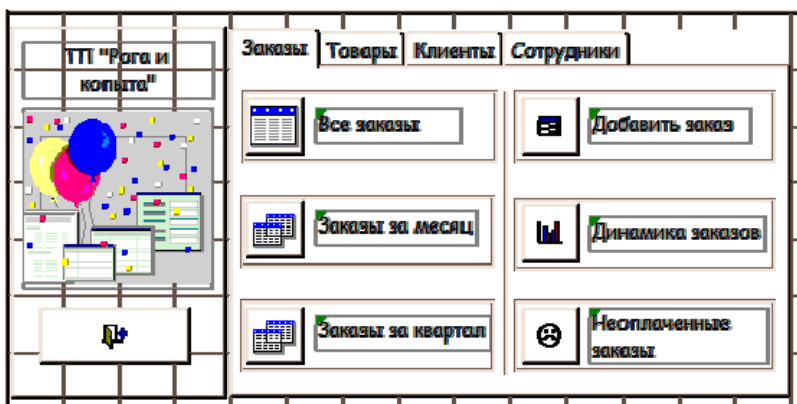


Рис. 9.1. Пример проекта вкладки *Главной кнопочной формы*

2. Оформите проект ГКФ в виде таблицы с описанием всех вкладок и кнопок, считая, что количество вкладок в форме соответствует количеству срезов для анализа (табл. 9.1).

Таблица 9.1

Описание проекта *Главной кнопочной формы*

ВКЛАДКА / Кнопки	Действие при нажатии кнопки	Имя макроса
ВКЛАДКА ЗАКАЗЫ		
<i>Все заказы</i>	Открыть таблицу <i>Заказы</i>	<i>ТаблЗаказы</i>
<i>Добавить заказ</i>	Открыть форму <i>Заказы</i> в режиме <i>Изменения данных</i>	<i>ФормаЗаказы</i>
<i>Заказы за месяц</i>	Открыть запрос <i>Заказы за месяц года</i>	<i>ЗапрЗаказыМесяц</i>
<i>Заказы за квартал</i>	Открыть запрос с параметрами <i>Заказы за квартал года</i>	<i>ЗапрЗаказыКвартал</i>
<i>Динамика заказов</i>	Открыть форму с диаграммой	<i>ФормаДинамика</i>
ВКЛАДКА ТОВАРЫ		
...	...	
ВКЛАДКА КЛИЕНТЫ		
...	...	

3. На вкладке БД *Создание* в группе команд *Формы* создайте пустую форму.

4. В режиме *Конструктора* в соответствии с разработанным проектом оформите *Главную кнопочную форму*. Используя команды на вкладке *Конструктор* из набора вкладок *Работа с макетами форм* добавьте в макет формы следующие элементы управления:

4.1. *Рисунок* с логотипом БД (самостоятельно подберите подходящий по габаритам рисунок).

4.2. *Вкладки* (по умолчанию создается набор из двух вкладок, самостоятельно добавьте дополнительные вкладки, используя команды контекстного меню, переименуйте вкладки).

4.3. *Кнопку* выхода из БД (при создании кнопки в режиме *Конструктора формы откажитесь* от настройки действий в окне *Мастера создания кнопок* (рис. 9.2), закройте окно мастера. Далее на

вкладке *Создание* в группе команд *Макросы и код* используйте команду *Макрос* для вызова режима *Конструктора макросов*, в котором самостоятельно создайте макрос *ВыходИзБД* (рис. 9.3): добавьте макрокоманду *ВыйтиИзAccess*, задайте значение для аргумента *Параметр = Подтверждение* и сохраните макрос).

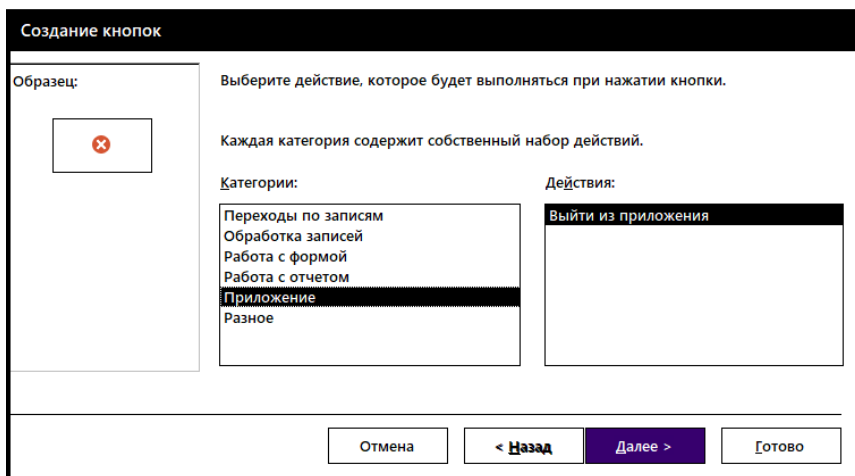


Рис. 9.2. Окно мастера «Создание кнопок» (закрыть)

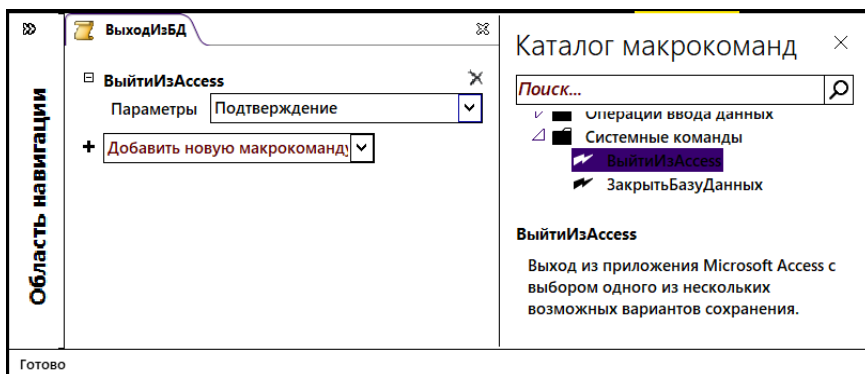


Рис. 9.3. Создание макроса *ВыйтиИзБД* (с параметром *Подтверждение*)

4.4. *Кнопки* (по 4 на каждой вкладке) для управления объектами БД (см. табл. 9.1).

Аналогично макросу *ВыходИзБД* самостоятельно создайте для каждой кнопки макрос и назначьте его кнопке, выбрав из списка для свойства *Нажатие кнопки* на вкладке *События*, в окне свойств кнопки (рис. 9.4).

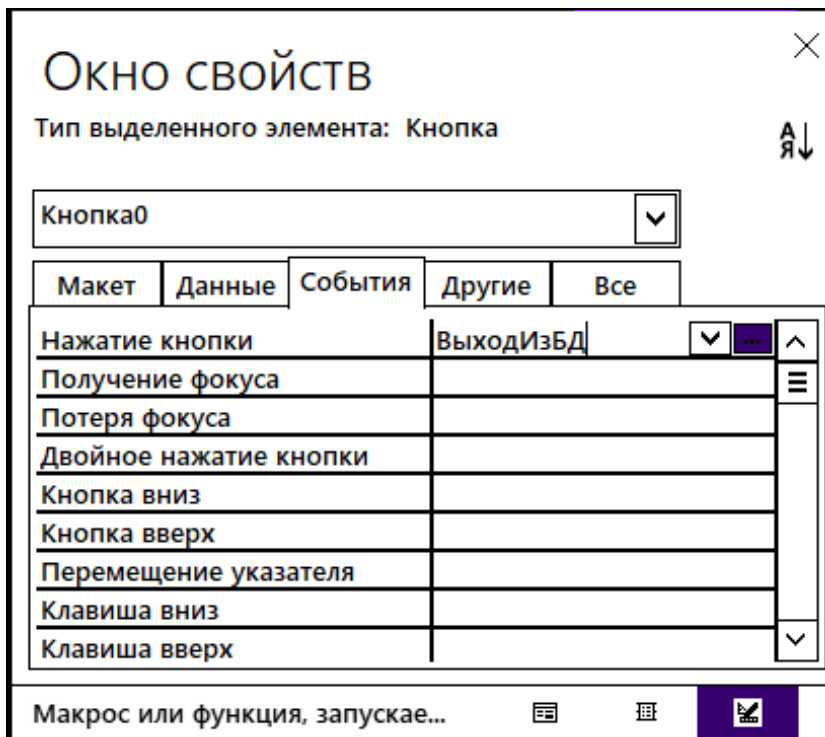


Рис. 9.4. Настройка для кнопки свойства *Нажатие кнопки*

Самостоятельно контролируйте список создаваемых макросов на вкладке *БД Макросы*.

4.5. *Элементы управления Линия и Прямоугольник* используйте для дополнительного оформления.

Пример макета кнопочной формы с текстом на кнопках приведен на рис. 9.5.

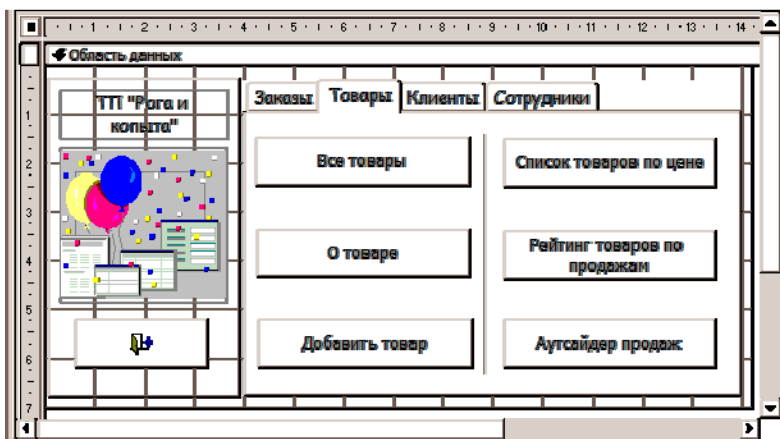


Рис. 9.5. Макет Главной кнопочной формы, вкладка Товары

5. На вкладке *Макет* настройте свойства *Главной кнопочной формы* (см. п. 4.4 теоретического раздела пособия):

- задайте подпись формы (название индивидуальной БД;
- уберите *полосы прокрутки, кнопки навигации* (перехода по записям) и *кнопки размеров окна* (оставьте только кнопку закрытия);
- подберите размер формы и задайте *тип границы* – Тонкая.

Внимание: Окно с настройками свойств формы (вкладка *Макет*) и результат отображения формы в режиме *Формы* приведите в отчете (все вкладки).

6. Настройте параметры запуска БД в окне «Параметры Access», отобразив его по команде *Параметры* из меню *Файл*. На вкладке *Текущая база данных* в области *Параметры для текущей базы данных*:

- задайте *Заголовок приложения* (название индивидуальной БД по варианту);
- в списке *Форма просмотра* выберите *Главную кнопочную форму*, которая будет автоматически открываться при загрузке БД. Окно с настройками приведите в отчете.

7. Проверьте выполненные настройки, закрыв и повторно открыв БД. Окно БД с загруженной формой приведите в отчете.

8. Для отчета подготовьте 2 распечатки окна БД: вкладки *Формы* и *Макросы* (список созданных кнопочных форм и макросов в режиме *Просмотра сведений*).

9. Оформите отчет о выполнении работы.

Лабораторная работа № 10

ВЗАИМОДЕЙСТВИЕ СУБД MS ACCESS С ПРИЛОЖЕНИЯМИ MS OFFICE



Цель работы:

- изучить технологии взаимодействия СУБД Access с приложениями MS Office;
- приобрести практический опыт использованием информации из БД при подготовке серийных документов в MS Word;
- приобрести практический опыт анализа информации из БД средствами MS Excel.



Содержание отчета

1. Название работы.
2. Цель работы.
3. Постановка задачи по варианту.
4. Описание выполнения задания (включая распечатки окон при создании документов MS Office).
5. Список контрольных вопросов и ответов на них.
6. Выводы.
7. Список использованных источников.
8. Приложение А (макет письма для слияния в режиме *Отображения полей слияния*, одностраничный документ).
9. Приложение Б (первый лист письма рассылки в режиме *Отображения данных после слияния*, многостраничный документ).



Контрольные вопросы

1. Назовите и кратко охарактеризуйте основные технологии взаимодействия СУБД MS Access с приложениями Microsoft Office. Приведите описание команд (по группам) вкладки *Внешние данные*.
2. Перечислите *форматы экспорта/импорта* объектов БД и способы конвертирования данных в форматы Microsoft Office (ответ оформите в виде таблицы).
3. Сформулируйте назначение и возможности процедуры *Слияния в MS Word*, виды документов для слияния.

4. Приведите порядок создания серийных документов в режиме *Мастера слияния*. Опишите назначение команд вкладки *Рассылки* приложения MS Word (по группам, в виде таблицы).

5. Приведите краткую характеристику инструментов MS Excel для обработки и анализа данных из БД (сортировка, фильтрация, промежуточные итоги, сводные таблицы).

6. Опишите назначение и особенности применения промежуточных итогов в MS Excel. Приведите порядок действий при подведении промежуточных итогов в таблицах MS Excel.

7. Сформулируйте понятие сводной таблицы MS Excel, назовите ее назначение и особенности создания. Опишите структуру сводной таблицы и приведите краткую характеристику ее элементов.

8. Перечислите инструменты графического анализа данных из БД MS Access в MS Excel. Приведите примеры выбора типа диаграмм для наглядного представления данных (обоснование выбора типа диаграммы по пяти типам сравнения).



Порядок выполнения работы

1. Экспорт таблиц в Microsoft Word.

Выделите в области навигации БД базовую таблицу по варианту (табл. 10.1) и экспортируйте ее в документ MS Word: на вкладке *Внешние данные* в группе команд *Экспорт* в списке *Дополнительно* используйте команду *MS Word* для вызова диалогового окна «Экспорт».

Укажите, в каком формате, под каким именем и в какую папку по умолчанию предлагается выполнить сохранение таблицы: _____

С помощью кнопки «Обзор» выберите личную папку.

Установите флажок *Открыть целевой файл после завершения операции экспорта* и завершите операцию экспорта.

В документе MS Word при необходимости выполните дополнительное форматирование таблицы и установку параметров страницы для наглядного отображения данных (окно с результатом приведите в отчете).

Исходные данные для экспорта таблиц в MS Word

Вариант	Таблица	Вариант	Таблица
1	Должности	9	Коммунальные услуги
2	Абоненты	10	Клиенты
3	Клиенты	11	Преподаватели
4	Предметы	12	Предметы
5	Книги	13	Клиенты
6	Вкладчики	14	Клиенты
7	Клиенты	15	Видеокассеты
8	Фотоуслуги		

2. Экспорт запроса в MS Excel.

Выделите в области навигации БД запрос с расчетами «13 Расчеты» и экспортируйте его в электронную таблицу MS Excel: на вкладке *Внешние данные* в группе команд *Экспорт* используйте команду *MS Excel* для вызова диалогового окна «Экспорт».

Укажите, в каком формате, под каким именем и в какую папку по умолчанию предлагается выполнить сохранение запроса: _____.

С помощью кнопки «Обзор» выберите личную папку, измените имя книги по умолчанию на *13_Фамилия_Расчеты*.

Установите флажок *Открыть целевой файл после завершения операции экспорта* и завершите операцию экспорта.

Укажите, какое имя по умолчанию получил рабочий лист? _____.

При необходимости выполните дополнительное форматирование таблицы, измените размеры ячеек для наглядного отображения данных (окно с полученным результатом приведите в отчете).

Сохраните изменения в рабочей книге MS Excel и выполните задание 3.

3. Обработка данных запроса в MS Excel.

Отобразите данные экспортированного запроса «13 Расчеты» на диаграмме, выполнив предварительную обработку табличной базы данных в рабочей книге MS Excel:

1) сортируйте по возрастанию данные по полю *Фамилия* (для варианта 3 – *Название фирмы-клиента*).

Внимание: операция сортировки по указанному полю *необходима* и *обязательна* для корректного подведения *промежуточных итогов*.

2) подведите *промежуточные итоги* [13] по полю *Фамилия* (для варианта 3 – *Название фирмы-клиента*), используя команду *Промежуточный итог* на вкладке *Данные* в группе *Структура*.

В диалоговом окне «Промежуточные итоги»:

– в области «При каждом изменении в» выберите из списка поле для группировки: *Фамилия* (для варианта 3 – *Название фирмы-клиента*),

– самостоятельно определите операцию и числовые поля для подведения итогов.

В отчете поясните свой выбор и приведите заполненное по варианту диалоговое окно «Промежуточные итоги»;

3) отобразите в полученной таблице с промежуточными итогами *2-й уровень структуры документа*;

4) выделите *несмежные диапазоны* с фамилиями (для варианта 3 – названиями) и числовыми данными, постройте *диаграмму* [13] (тип диаграммы и параметры ее форматирования выберите самостоятельно, обоснуйте свой выбор).

Приведите в отчете полученные результаты: окно книги с таблицей промежуточных итогов и готовую диаграмму.

Сохраните изменения в рабочей книге.

Сделайте выводы о возможностях инструмента «Промежуточные итоги».

4. Построение сводной таблицы по данным запроса в MS Excel.

Сводные таблицы MS Excel предназначены для обобщения (объединения, переработки) информации из табличной базы данных, также с их помощью можно вывести промежуточные итоги с любым уровнем детализации данных [13]. Макет сводной таблицы (рис. 10.1) включает четыре области, каждая из которых имеет свое назначение.

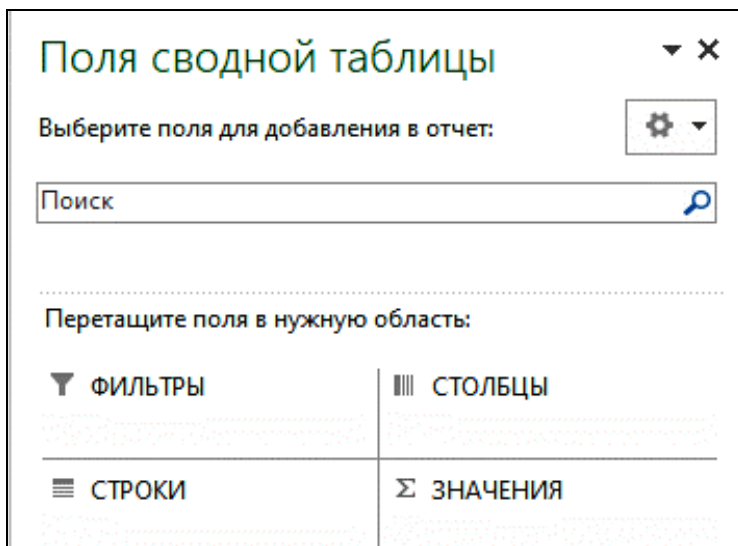


Рис. 10.1. Макет сводной таблицы

1. *Фильтры.* Поля, помещенные в эту область, используются для построения раскрывающихся списков для выбора варианта фильтрации данных. При этом в области значений сводной таблицы будут отображены итоговые данные для выбранного варианта. Область фильтров отчета определяет глобальный срез для анализа табличных данных.

2. *Строки.* Уникальные значения полей, помещенных в эту область, используются в качестве заголовков строк в сводной таблице. Если в области помещено одно поле исходной таблицы MS Excel, то количество строк в сводной таблице (без учета итогов) равно числу уникальных значений этого поля.

3. *Столбцы.* Уникальные значения полей, помещенных в эту область, используются в качестве заголовков столбцов в сводной таблице. Если в области помещено одно поле, то количество столбцов в сводной таблице (без учета итогов) равно числу уникальных значений этого поля.

4. *Значения.* Значения полей, помещенных в эту область, используются для заполнения ячеек сводной таблицы итоговыми данными – результатами суммирования, подсчета количества значений и т. д.

Для числовых полей по умолчанию применяется функция *Сумма*, а для текстовых – *Количество*.

Для построения сводной таблицы по варианту экспортируйте в MS Excel запрос «02 Исходные данные» (замените имя по умолчанию на: *02_Фамилия_Исходные данные* и сохраните файл рабочей книги в личную папку).

В рабочей книге на основе данных экспортированного запроса постройте *сводную таблицу*:

- выделите любую ячейку в области таблицы данных на листе MS Excel;
- выполните команду *Сводная таблица* на вкладке *Вставка* в группе команд *Таблица*;
- сформируйте макет сводной таблицы, расположив поля в соответствии с вариантом (табл. 10.2).

Таблица 10.2

Исходные данные для построения сводной таблицы

Вариант	Поля сводной таблицы в области			
	Фильтры (страницы)	Строки	Столбцы	Значения
1	2	3	4	5
1	Название проекта	Фамилия сотрудника	Название должности	Количество отработанных дней
2	Тема звонка	Дата разговора	Фамилия абонента	Продолжительность разговора
3	Название фирмы-клиента	Дата заказа	Название товара	Количество проданного товара
4	Название должности	Дата занятия	Название предмета	Количество проведенных часов
5	Фамилия читателя	Дата выдачи	Название книги	Цена книги
6	Название вклада	Дата операции	Фамилия вкладчика	Сумма операции
7	Фамилия клиента	Дата заселения	Номер комнаты	Стоимость проживания за 1 сут
8	Название услуги	Дата заказа	Фамилия клиента	Количество единиц заказа

1	2	3	4	5
9	Название услуги	Дата оплаты	Фамилия квартиросъемщика	Количество жильцов
10	Название товара	Дата выдачи	Фамилия клиента	Залоговая стоимость товара
11	Название предмета	Дата занятия	Фамилия студента	Число часов пропусков
12	Фамилия студента	Дата рейтинга	Название предмета	Рейтинговая оценка
13	Название пиццы	Дата заказа	Фамилия клиента	Количество заказанной пиццы
14	Название маршрута	Дата поездки	Фамилия клиента	Количество путевок
15	Наименование видеокассеты	Дата выдачи	Фамилия клиента	Залоговая стоимость

В построенной сводной таблице выполните *группировку данных* по строке по интервалам времени *год* и *месяц* (в отчете приведите распечатку диалогового окна «Группировка»).

В отчете приведите:

- макет сводной таблицы;
- полученный результат.

Для сводной таблицы на вкладке *Анализ* из набора вкладок *Работа со сводными таблицами* выполните команду *Сводная таблица/Параметры, Отобразить страницы фильтра отчета*.

Оцените полученный результат.

Сделайте выводы о возможностях сводной таблицы.

5. Слияние с Microsoft Word.

Выполните подготовку *массовой рассылки* на заданную тему (по варианту).

Внимание: до подготовки серийного письма:

– в таблицу с персональными данными (*Клиенты, Абоненты* и т. п. по варианту) в режиме *Конструктора* добавьте поле *Пол* (тип данных – *текстовый*, создайте столбец подстановки с фиксированными значениями – *жен, муж*);

– в режиме *Таблицы* заполните созданное поле значениями (*жен, муж*).

Вариант 1. Приглашение на семинар по инновационным технологиям или мастер-класс по 3d-моделированию.

Вариант 2. Опрос абонентов о качестве обслуживания (или сообщение о проведении рекламной акции).

Вариант 3. Рекламное предложение о распродаже некоторых моделей товаров или предпраздничных скидках на товары.

Вариант 4. Приглашение на ежегодную Международную научно-техническую конференцию или онлайн семинар.

Вариант 5. Уведомление о необходимости возврата книг или информация о проведении презентации.

Вариант 6. Сообщение об изменении ставки рефинансирования или предложение новых условий по вкладам.

Вариант 7. Уведомление об изменении условий бронирования номеров или предложение клиентам новых услуг.

Вариант 8. Сообщение о предоставлении дополнительных видов услуг или изменении стоимости услуг.

Вариант 9. Уведомление об изменении тарифов на оплату отдельных видов коммунальных услуг или предоставлении льгот.

Вариант 10. Напоминание о необходимости своевременного возврата взятой напрокат техники или о продлении срока проката.

Вариант 11. Уведомление об изменении стоимости оплаты за отработку пропущенных занятий и/или условий отработки пропущенных занятий.

Вариант 12. Сообщение об изменении правил текущей аттестации или условий предоставления скидки со сформированной стоимости обучения.

Вариант 13. Приглашение на презентацию нового ассортимента или мастер-класс по приготовлению диетической пиццы.

Вариант 14. Сообщение о распродажах «горящих» путевок или предложение экскурсий с тематическими мастер-классами.

Вариант 15. Сообщение о поступлениях новых видеоматериалов или проведении кинолектория по теме «Цифровое производство», «Цифровая трансформация».

В качестве *источника слияния* выделите в панели навигации индивидуальной БД таблицу с персональными данными (*Клиенты*, *Абоненты* и т. п. по варианту).

На вкладке *Внешние данные* в группе команд *Экспорт* выполните команду *Слияние в Word*, в диалоговом окне «Слияние с доку-

ментами MS Word» установите переключатель *Создать новый документ и установить связь с ним* и нажмите кнопку «Ок» (прозойдет открытие MS Word).

В MS Word самостоятельно подготовьте и отформатируйте текст стандартного серийного письма на заданную тему (по варианту).

Для подстановки данных адресата из базовой таблицы индивидуальной БД добавьте в текст письма *поля слияния MS Access*, используя кнопку «Вставить поле слияния» на вкладке *Рассылки* в группе команд *Составление документа и вставка полей* (см. рис. 5.3 в п. 5.2 теоретического раздела пособия).

Для персонализации обращения к адресату используйте в тексте письма *поле слияния MS Word «If... Then...Else»* (рис. 10.2) с условием по полю *Пол* (см. рис. 5.4 в п. 5.2 теоретического раздела пособия).

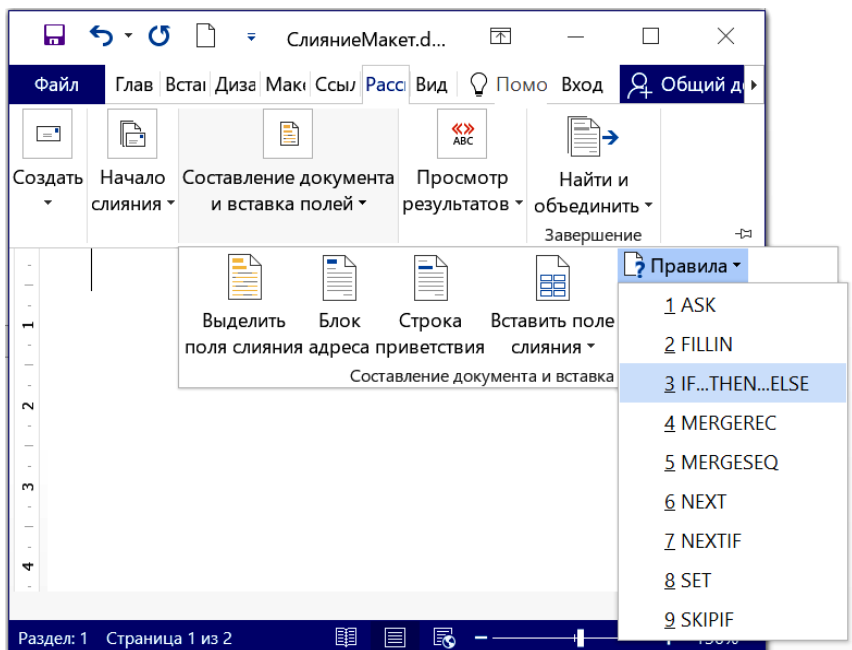


Рис. 10.2. Вставка поля Word «If... Then... Else»

Завершите слияние *в новый документ для всех записей* источника данных, для чего на вкладке *Рассылки* в группе команд *Заверше-*

ние нажмите кнопку «Найти и объединить» и выполните команду *Изменить отдельные документы*.

Сохраните в личной папке 2 документа MS Word и приведите в отчете:

– макет письма слияния (одностраничный документ, в приложении А);

– готовую рассылку (многостраничный документ, одна страница – в приложении Б).

7. Задания для самостоятельной работы по вариантам.

В индивидуальной БД самостоятельно выполните построение диаграмм для визуализации данных по варианту.

Обоснуйте макет запроса для выбора данных для визуализации (приведите в отчете параметры, выражения и т. п.).

Обоснуйте выбор типа диаграммы [13]. Тип, формат и параметры диаграммы задайте самостоятельно (приведите в отчете).

Вариант 1. Отобразите на диаграмме вклад каждого проекта (по стоимости работ) в общий объем выплат.

Вариант 2. Отобразите на диаграмме вклад каждого абонента (количество минут разговоров) в общую длительность разговоров.

Вариант 3. Отобразите на диаграмме вклад каждого товара (по стоимости заказов) в общий товарооборот.

Вариант 4. Отобразите на диаграмме для каждого преподавателя количество отработанных часов по указанному предмету.

Вариант 5. Отобразите на диаграмме число выдач для каждого читателя по указанному автору.

Вариант 6. Отобразите на диаграмме баланс по каждому виду вкладов для указанного вкладчика.

Вариант 7. Отобразите на диаграмме вклад каждого клиента (по стоимости проживания) в общий оборот гостиницы.

Вариант 8. Отобразите на диаграмме сумму заказов по заданной услуге для каждого клиента.

Вариант 9. Отобразите на диаграмме долю каждой услуги (сумма оплаты) в объеме оплат указанного квартиросъемщика.

Вариант 10. Отобразите на диаграмме вклад каждой модели техники (стоимость проката) в общий оборот салона проката.

Вариант 11. Отобразите на диаграмме пропуски каждого студента (число часов) по указанному предмету.

Вариант 12. Отобразите на диаграмме общий рейтинг каждого студента по указанному предмету.

Вариант 13. Отобразите на диаграмме вклад каждого вида пиццы (стоимость проданных порций) в общую сумму продаж.

Вариант 14. Отобразите на диаграмме вклад каждого маршрута (стоимость проданных путевок) в общий оборот туристического бюро.

Вариант 15. Отобразите на диаграмме вклад каждого видеоматериала (стоимость проката) в общий оборот видеосалона.

6. Для отчета подготовьте распечатку окна личной папки с созданными документами MS Office (представление файлов – в виде таблицы).

7. Оформите отчет о выполнении работы.

Лабораторная работа № 11

РАЗРАБОТКА ПРЕЗЕНТАЦИИ БАЗЫ ДАННЫХ СРЕДСТВАМИ MS POWERPOINT



Цель работы:

- изучить функциональные возможности программы подготовки презентаций MS PowerPoint;
- приобрести навыки разработки презентации базы данных как информационного продукта средствами MS PowerPoint.



Содержание отчета

1. Название и цель работы.
2. Постановка задачи.
3. Описание выполнения задания, включая (окна с настройками, распечатку одного слайда в режиме *Образца слайдов*, распечатку презентации в режиме *Сортировщика слайдов*).
4. Список контрольных вопросов и ответов на них.
5. Выводы.
6. Список использованных источников.
7. Приложение А (распечатка презентации по 6 слайдов на странице).



Контрольные вопросы

1. Что такое презентация? Какие бывают виды презентаций? Назовите основные форматы файлов MS PowerPoint и укажите их особенности. Перечислите способы создания презентаций в MS PowerPoint.
2. Перечислите режимы MS PowerPoint для работы с презентациями, опишите их назначение, укажите способы переключения между режимами (ответ представьте в виде таблицы).
3. Что такое слайд? Назовите виды слайдов MS PowerPoint. Перечислите способы создания (добавления) нового слайда, дублирования слайдов, их удаления и перемещения. В каком режиме выполняется работа с объектами на слайдах и заметками к слайдам?

4. Приведите краткую характеристику атрибутов слайда: макет (разметка) слайда, тема оформления и цветовая схема слайда, размер слайда, фон слайда. Назовите способы их изменения.

5. Как добавить мультимедийный объект на один слайд презентации? На каждый слайд презентации? Опишите возможности режима *Образец слайдов* для разметки и оформления слайдов.

6. Каким образом выполняется настройка анимации объектов на слайдах? Как можно изменить порядок и эффекты анимации?

7. Как настраиваются параметры переходов (смены слайдов) для выделенной группы слайдов? Для всех слайдов презентации? Опишите команды вкладки *Переходы* (по группам, в виде таблицы).

8. Опишите порядок добавления на слайд управляющих кнопок и настройки их параметров? Как настроить презентацию для управления только с помощью кнопок?

9. В каких режимах работы с презентацией можно скрыть слайды? Упорядочить, оценить время демонстрации слайда и эффекты перехода при показе слайдов? Как в процессе показа презентации выполнить переход к скрытому слайду?

10. Опишите возможности режима *Слайд-шоу*. Какие команды КЗМ доступны пользователю в этом режиме? Как запустить демонстрацию слайдов с первого слайда? С текущего слайда? Как можно добавить и сохранить заметки к слайдам в режиме слайд-шоу?

11. Сформулируйте назначение режима докладчика, опишите параметры режима и структуру экрана.



Порядок выполнения работы

1. Средствами MS PowerPoint подготовьте презентацию БД, разработанной в предыдущих лабораторных работах. Для автоматизации работы при подготовке презентации используйте Мастер содержания или выберите один из стандартных шаблонов презентаций MS PowerPoint.

2. Презентация должна включать до 20 слайдов (рис. 11.1):

1) титульный слайд (название БД, автор-разработчик) – 1 слайд;

2) назначение, область применения и достоинства разработанной БД – 2–3 слайда;

3) подробное описание Базы данных:

– хранение данных (таблицы) – 3–4 слайда;

- логическая схема БД – 1 слайд;
- анализ данных (запросы) – 2–4 слайда;
- интерфейс БД (формы) – 2–3 слайда;
- выдача итоговой информации (отчеты) – 2 слайда;
- 4) выводы (выгоды для пользователя БД) – 1 слайд;
- 5) контактная информация автора-разработчика – 1 слайд.

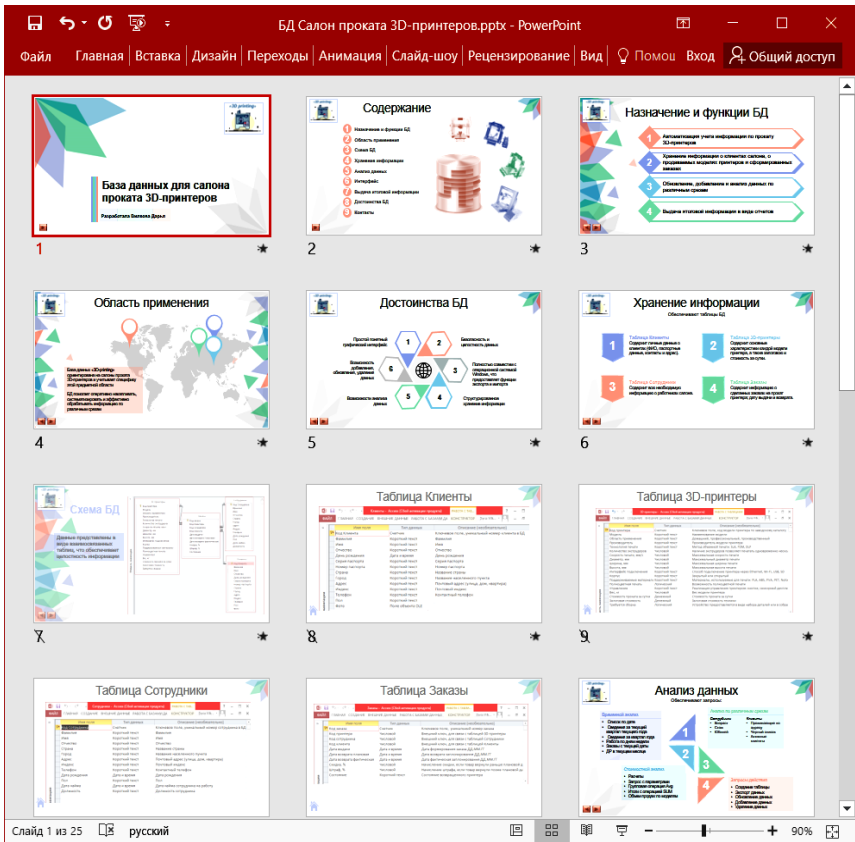


Рис. 11.1. Слайды презентации БД (в режиме *Сортировщика слайдов*)

3. На вкладке *Кондуктор* в группе команд *Настроить* установите широкоэкранный размер слайдов (16 : 9).

4. На вкладке *Анимация* настройте эффекты анимации текста и объектов на слайдах (рис. 11.2).

Эффекты назначаются по времени после предыдущего объекта (автоматически, не по щелчку мыши).

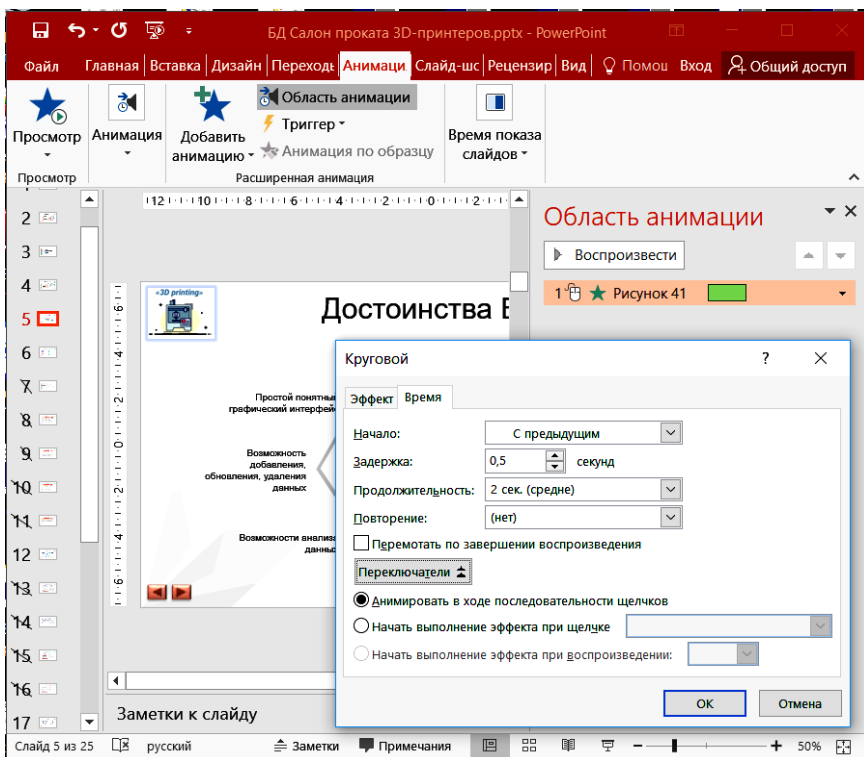


Рис. 11.2. Параметры настройки анимации объектов на вкладке *Анимация*

Для упрощения работы с настройкой анимации объектов на слайдах и обеспечения совместимости с различными версиями MS PowerPoint используйте простые эффекты анимации.

5. На вкладке *Переходы* настройте параметры смены слайдов и время показа слайдов (рис. 11.3).

Как правило, скорость перехода устанавливают быстро или средне, а смену слайдов – по щелчку.

Если необходима автономная демонстрация, без участия пользователя, то дополнительно устанавливается смена слайдов и по времени после определенного числа секунд.

6. Для приложения к отчету подготовьте распечатки слайдов (по 6 на листе), используя команду *Файл/Печать, напечатать все слайды как раздаточные материалы по 6 слайдов на странице*.

7. Оформите отчет о выполнении работы.

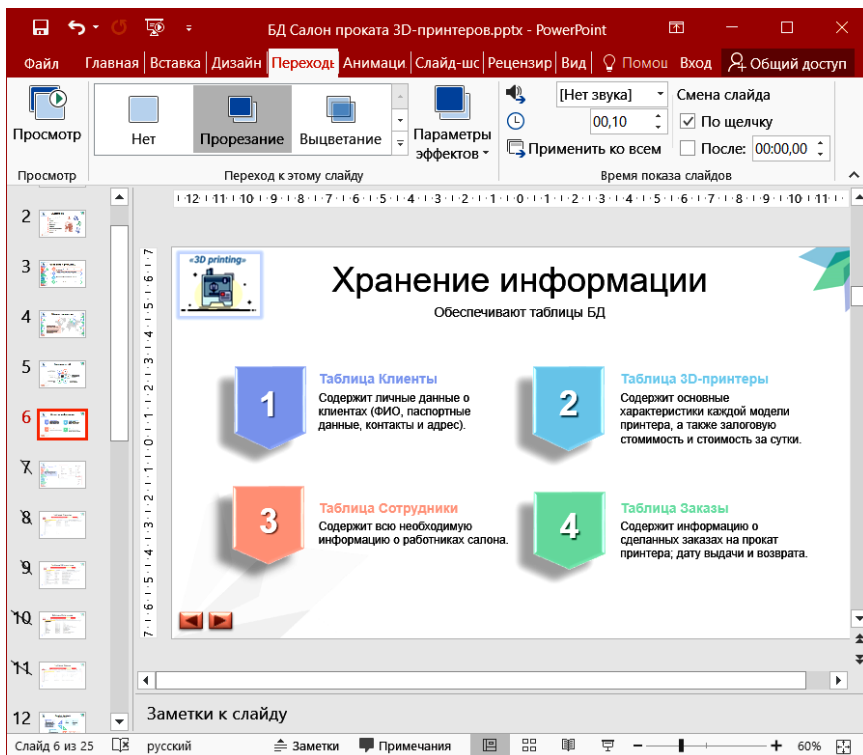


Рис. 11.3. Параметры настройки смены слайдов на вкладке *Переходы* (в группах команд *Переход к этому слайду* и *Время показа слайдов*)

3. КОНТРОЛЬНАЯ РАБОТА

При изучении дисциплины «Сетевые технологии и базы данных» в рамках самостоятельной работы студентов предусмотрена контрольная работа, выполнение которой обусловлено необходимостью активизации учебно-познавательной деятельности обучающихся, а также формирования у них умений и навыков как самостоятельного приобретения и обобщения знаний, так и самостоятельного применения полученных знаний на практике.

Контрольная работа используется как средство оценки эффективности образовательной деятельности и служит основанием для предварительной оценки знаний обучающихся перед аттестацией по дисциплине. Выполнение контрольной работы и ее защита предшествует аттестации по дисциплине.

Контрольная работа состоит из 2-х частей: теоретической части, включающей блок теоретических вопросов по разделу «Основы работы с реляционными базами данных и СУБД», и практической части, включающей задания по применению теоретических знаний в области работы с реляционной базой данных для выбранной предметной области (по вариантам).

К аттестации студент предоставляет распечатанную и сшитую работу, а также два файла: текст работы (документ MS Word) и электронный вариант базы данных (предварительно выполнив операцию сжатия БД). Имена файлов формируются по правилу: *№варианта_КР_Фамилия_студента*.

3.1. Общие сведения о структуре работы и выборе варианта заданий

В данном пункте приведены варианты заданий, необходимые сведения о структуре и объеме работы, содержании заданий.

Тема работы: Обработка информации средствами реляционных СУБД.

Цели работы:

1) изучение технологии структурирования, хранения и обработки информации с помощью современных СУБД;

2) получение практических навыков разработки реляционных БД с помощью офисной СУБД;

3) освоение технологий анализа и интеграции данных.

Объем работы составляет 25–30 страниц.

Структура работы включает следующие элементы:

Титульный лист.

Задание (выдается преподавателем).

Содержание (*автоматически сформированное*).

Введение.

1. Теоретическая часть².

1.1. БД как средство хранения структурированной информации. Краткая характеристика реляционных БД. Нереляционные базы данных (No-SQL) и их особенности. Технология обработки данных in-memory.

1.2. Понятие системы управления базами данных (СУБД), функции, классификация, обзор рынка СУБД.

1.3. Порядок создания макета таблицы БД в режиме Конструктора, правила выбора типов данных и назначения свойств полей в таблицах MS Access.

1.4. Назначение и порядок создания запросов выборки. Правила и примеры формирования условий отбора для числовых и текстовых полей.

1.5. Вычисления в запросах: структура вычисляемого поля, виды операндов и операторов. Примеры вычислений с применением функций.

1.6. Применение итоговых запросов, особенности построения перекрестных запросов.

1.7. Взаимодействие СУБД MS Access с MS Excel: экспорт данных и их анализ с применением фильтрации, промежуточных итогов, сводных таблиц.

² Приведен примерный перечень теоретических вопросов для самостоятельного изучения, который согласовывается с преподавателем и может быть изменен в зависимости от формы получения образования и уровня подготовки обучающихся.

Вопрос по варианту предполагает углубленное изучение, ответ должен содержать не только текст, но и примеры, иллюстрации, таблицы.

1.8. Использование из СУБД MS Access механизма «Слияние в Word» для подготовки типовых документов (списков рассылки).

1.9. Вопрос 1 по варианту (табл. 3.1).

Таблица 3.1

Задания по вариантам
(вариант выбирается по номеру зачетной книжки)

Вариант	Вопрос 1	Предметная область	Вариант	Вопрос 1	Предметная область*
1	11	Выбирается из предложенного списка по согласованию с преподавателем	17	36	Выбирается из предложенного списка по согласованию с преподавателем
2	12		18	38	
3	17		19	39	
4	18		20	23	
5	19		21	26	
6	20		22	27	
7	21		23	28	
8	22		24	31	
9	39		25	35	
10	24		26	15	
11	25		27	16	
12	29		28	40	
13	30		29	41	
14	32		30	42	
15	33		31	43	
16	34		32	44	

Примечание: выбор предметной области для студентов заочной формы обучения выполняется с учетом места работы и должностных обязанностей.

2. Практическая часть.

2.1. Проектирование БД для предметной области «*Название предметной области*» (по варианту)».

2.2. Реализация проекта БД средствами СУБД MS Access.

2.3. Проектирование запросов.

2.4. Проектирование форм и отчетов.

2.5. Интеграция БД с приложениями Microsoft Office.

Заключение.

Список использованных источников.

Приложения (*при необходимости*).

3.2. Список теоретических вопросов

1. Базы данных как средство хранения структурированной информации. Основные типы структур данных (запись, поле, отношение, ключевое поле) и их краткая характеристика.

2. Понятие модели (схемы) БД. Краткая характеристика основных моделей БД (иерархическая, сетевая, реляционная). Технология обработки данных in-memo.

3. Этапы проектирования реляционных БД и их краткая характеристика.

4. Понятие информационно-логической модели предметной области (ИЛМ ПО) и порядок ее формирования.

5. Понятие нормализации данных, требования нормализации таблиц реляционной БД (3-я нормальная форма).

6. Системы управления базами данных (СУБД). Понятие, функции, классификация. Назначение и функциональные возможности СУБД MS Access.

7. Способы создания новой БД в MS Access. Работа с мастерами и шаблонами баз данных.

8. Назначение объектов СУБД MS Access. Работа с объектами MS Access в режимах объекта и конструктора. Переключение между режимами.

9. Проектирование структуры БД. Разбиение данных на таблицы. Требования к содержанию реляционных таблиц.

10. Создание структуры таблицы MS Access в режиме Конструктора (наименование поля, тип данных, свойства поля, описание).

11. Понятие типа данных в MS Access. Типы данных: счетчик, числовой, денежный, короткий текст, длинный текст (МЕМО), OLE, гиперссылка, дата/время, логический. Особенности их применения и краткая характеристика. Использование Мастера подстановки.

12. Понятие свойства поля. Характеристика свойств полей таблиц MS Access: размер поля, формат, подпись, маска ввода, значение по умолчанию, правило проверки, сообщение об ошибке, обязательное поле, пустые строки, индексированное поле, сжатие Юникод.

13. Ключевое поле: понятие и виды ключевых полей. Установка связей между таблицами. Связи «один-к-одному», «многие-к-одному», «один-ко-многим», «многие-ко-многим».

14. Понятие схемы БД. Порядок и условия формирования схемы БД в MS Access. Параметры объединения ключевых полей и их особенности.

15. Понятие запроса в MS Access. Назначение и виды запросов. Источники запроса. Результаты выполнения запроса.

16. Запросы MS Access типа выборки. Общая методика создания. Работа с бланком запроса в режиме Конструктора.

17. Понятие выражения и его элементы (операнды и операторы). Работа с строителем выражений. Использование выражений в запросах MS Access.

18. Условие отбора в запросах MS Access. Особенности формирования условий отбора для полей различных типов.

19. Вычисляемые поля в запросах MS Access. Порядок создания вычисляемого поля и его элементы. Использование строителя выражений для создания вычисляемых полей.

20. Понятие параметров и их использование в запросах MS Access. Правила записи параметров. Примеры использования параметров в условиях отбора и вычисляемых полях.

21. Особенности применения групповых операций в запросах MS Access. Краткая характеристика статистических (итоговых) функций.

22. Назначение перекрестных запросов и способы их создания. Примеры перекрестных запросов.

23. Функции MS Access для обработки значений дата/время (описание функций из категории Дата/время, примеры использования).

24. Функции MS Access для обработки текстовых значений (описание функций из категории Текстовые, примеры использования).

25. Функции управления MS Access (описание функций из категории Управление, примеры использования).

26. Основные технологии связи СУБД MS Access с приложениями Microsoft Office. Публикация в Word. Использование процедуры Слияние для подготовки типовых документов (списков рассылки).

27. Основные технологии связи СУБД MS Access с приложениями Microsoft Office. Анализ данных в Excel (фильтрация, промежуточные итоги, сводные таблицы).

28. Экспорт/импорт объектов СУБД MS Access. Форматы сохранения объектов.

29. Запросы на изменение: назначение, виды и краткая характеристика. Общая методика создания. Понятие целостности данных, каскадное обновление данных и удаление записей.

30. Формы СУБД MS Access, их назначение и виды. Структура формы. Настройка параметров формы, режимы печати.

31. Способы создания табличных форм. Возможности работы с данными в режиме формы. Режим Конструктор форм.

32. Использование элементов управления для построения форм в режиме Конструктора. Назначение элементов управления и приемы работы.

33. Кнопочные формы, их назначение, особенности создания и настройки.

34. Инструменты СУБД MS Access для построения диаграмм. Правила выбора типа диаграммы для наглядного отображения данных.

35. Отчеты, их назначение и особенности создания. Виды стандартных отчетов. Структура отчета в режиме конструктора. Печать отчетов

36. Формирование отчетов с добавлением вычисляемых и итоговых полей. Создание почтовых наклеек.

37. Понятие и применение макросов в MS Access. Особенности создания и выполнения макросов в БД.

38. Настройка рабочей среды MS Access. Параметры запуска индивидуальной БД. Отключение параметров запуска БД.

39. Сжатие и восстановление БД.

40. Запросы SQL их особенности. Примеры применения.

41. Понятие и особенности языка SQL, его назначение и функции.

42. Типы данных SQL. Примеры применения.

43. Операторы SQL. Основы синтаксиса. Примеры записи.

44. Инструкции SQL. Использование выражений в инструкциях SQL. Примеры записи.

3.3. Примерный перечень предметных областей³

1. Автомастерская.
2. Автоперевозки грузовые/пассажи́рские.
3. Аренда помещений.

³ Предметная область выбирается по согласованию с преподавателем.

4. Видеосервис.
5. Выдача учебно-методической литературы.
6. Выполнение НИОКР.
7. Договора подряда.
8. Доставка пиццы/обедов в офис.
9. Каршеринг.
10. Коммунальные услуги.
11. Посещаемость занятий студентами.
12. Продажа техники (по категориям).
13. Прокат техники/инструмента/оборудования (по категориям).
14. Работа call-центра.
15. Работа кредитного отдела.
16. Работа медиалаборатории.
17. Резервирование билетов.
18. Ремонт бытовых приборов (по категориям).
19. Складской учет.
20. Торговые операции (промышленные товары).
21. Туристическое агентство (промышленный туризм).
22. Учет выполнения сменных заданий.
23. Учет материальных ценностей.
24. Учет НИР студентов.
25. Учет почасовой работы преподавателей.
26. Учет работ по договорам-подрядам.
27. Учет ремонтных работ в подразделении.
28. Учет успеваемости студентов.
29. Фотосервис.
30. Экскурсионное обслуживание (по направлениям).

3.4. Указания к выполнению и оформлению практической части

В данном разделе пособия приведено примерное содержание практической части, указания к ее выполнению и оформлению. В соответствии со структурой, приведенной в п. 3.1 данного раздела пособия, практическая часть контрольной работы имеет порядковый номер 2 и включает подпункты 2.1–2.5.

В связи с этим в тексте используется нумерация раздела, его пунктов и подпунктов, а также оформление заголовков, которые

соответствуют установленным на кафедре требованиям к структуре и оформлению контрольной работы и отличаются от требований к оформлению данного пособия.

Примерная структура практической части контрольной работы оформляется в виде:

2. Практическая часть.

2.1. Проектирование БД для предметной области «Прокат автомобилей».

2.1.1. Постановка задачи.

2.1.2. Описание и анализ предметной области «Прокат автомобилей».

2.1.3. Проектирование БД для предметной области «Прокат автомобилей».

2.2. Реализация проекта БД средствами СУБД.

2.2.1. Создание таблиц БД.

2.2.2. Формирование схемы БД.

2.3. Проектирование запросов.

2.4. Проектирование форм и отчетов.

2.5. Интеграция БД с приложениями Microsoft Office.

2.5.1. Анализ данных из БД в MS Excel.

2.5.2. Подготовка массовой рассылки с помощью процедуры Слияние в MS Word.

Содержание и указания к выполнению и оформлению пунктов и подпунктов практической части приведено ниже.

2. Практическая часть

2.1. Проектирование БД для предметной области «Прокат автомобилей»

2.1.1. Постановка задачи

1. Средствами реляционной СУБД Microsoft Access разработать БД для предметной области (*по варианту*), выполнить заполнение таблиц БД исходными данными контрольного примера и проверку работоспособности БД.

Исходные данные (заполняются по варианту):

Предметная область: «Прокат автомобилей» (деятельность салона по прокату автомобилей).

Входная информация: первичные документы предметной области (каталог автомобилей, договор аренды транспортного средства и др.).

Выходная информация: таблицы-отношения *Клиенты*, *Автомобили*, *Заказы* (*дополнительно может быть предусмотрена и таблица Сотрудники*), запросы, формы, отчеты.

Количественные ограничения: БД должна содержать информацию о не менее чем 5-ти моделях автомобилей, 5-ти клиентах, для каждого клиента оформляется не менее 3-х заказов.

2. Спроектировать запросы для анализа данных, объекты пользовательского интерфейса: формы для ввода данных, отчеты для отображения результатов.

3. Выполнить экспорт исходных данных из БД в книгу MS Excel, применить в рабочей книге MS Excel для анализа данных промежуточные итоги, построить сводную таблицу и диаграммы для визуализации данных по различным срезам: *клиент*, *автомобиль* и др. (*срезы определяются по варианту*).

2.1.2. Описание и анализ предметной области «Прокат автомобилей»

Указания по выполнению и оформлению п. 2.2:

1. В соответствии с этапами проектирования реляционной БД перед началом работы по созданию базы данных необходимо изучить первичные документы предметной области.

2. Далее на основании собранных документов составляется описание предметной области в виде текста объемом до 1 страницы,

а в приложении водятся примеры документов предметной области. Для понимания особенностей обработки информации для выбранной предметной области необходимо описать реальные объекты предметной области и процессы их взаимодействия.

3. Для определения потребностей в обработке информации следует изучить отчеты и другие документы, которые оформляют сотрудники в процессе работы, чтобы подготовить всю необходимую информацию для удовлетворения предполагаемых запросов пользователей и руководства салона.

4. Ниже приведен фрагмент описания объектов и процессов для предметной области «Прокат автомобилей» (описание и анализ индивидуальной предметной области должно быть достаточно детальным для понимания бизнес-процессов):

«При традиционной организации деятельности салона по прокату автомобилей информация о его бизнес-процессах находит свое отражение в таких первичных документах, как:

- каталог автомобилей, содержащий все необходимые технические и ценовые характеристики автомобиля, фото модели и др.;
- договор аренды транспортного средства, который содержит всю информацию о заключенной сделке, условиях и сроках проката, сведения о клиенте (и сотруднике).

В процессе оформления заказа на прокат транспортного средства происходит взаимодействие клиента и сотрудника. Сотрудник помогает клиенту в соответствии с индивидуальными критериями определиться с выбором марки и модели автомобиля, знакомит клиента с условиями проката, оформляет договор аренды транспортного средства.

Клиент получает в прокат и использует автомобиль в соответствии с определенными в договоре сроками и пробегом за сутки. В случае несвоевременного возврата транспортного средства клиенту начисляется штраф в установленном размере за каждый час задержки и каждые 100 км перепробега. В момент возврата автомобиля сотрудник осматривает автомобиль, проверяет и документально фиксирует его состояние после эксплуатации.

Сотрудники салона оформляют результаты своей работы в виде отчетов и регулярно (ежедневно, еженедельно, ежемесячно, ежеквартально) предоставляют их руководству для анализа.

Ручная обработка всей информации без использования вычислительной техники не может выполняться оперативно, требует значительных временных и трудовых затрат. Поэтому создание и использование базы данных «Автопрокат» позволит оперативно накапливать, систематизировать и эффективно обрабатывать информацию по различным срезам, формировать отчеты, выявлять слабые звенья в системе проката».

2.1.3. Проектирование БД для предметной области «Прокат автомобилей»

Укрупнено, процесс создания реляционной БД состоит из 4-х этапов, включая 3 проектных этапа: 1) определение цели создания базы данных; 2) выделение информационных объектов предметной области; 3) определение логической структуры БД, которые завершаются разработкой информационно-логической модели предметной области – проектом БД. На четвертом этапе выполняется реализации проекта БД средствами СУБД.

В соответствии с этой последовательностью выполняется поэтапное проектирование с описанием содержания каждого этапа.

Далее по тексту работы приводится содержание и результаты каждого этапа проектирования БД (по примеру п. 4 из лабораторной работы № 2).

2.2. Реализация проекта БД средствами СУБД MS Access

2.2.1. Создание таблиц БД

На четвертом этапе разработки БД на основании спроектированной информационно-логической модели с помощью СУБД MS Access создается файл БД (*приводится имя БД и ее расположение*).

Далее в файле БД формируются объекты для хранения – выполняется разработка макетов таблиц в режиме *Конструктор* с использованием команды *Конструктор таблиц* на вкладке модульной ленты *Создание*.

В первую очередь создаются макеты главных таблиц *Автомобили*, *Клиенты* (*и Сотрудники*), затем – подчиненная таблица *Заказы*.

В таблице *Заказы* для внешних ключей формируются поля со списком (с использованием *Мастера подстановки*).

При разработке макетов таблиц для каждого поля определяются:

- имя поля,
- тип данных и
- свойства поля в соответствии с описаниями (спецификациями)

таблиц.

Пример спецификации таблицы *Автомобили* приведен в таблице 2.1.

Таблица 2.1 – Примерное содержание макета таблицы *Автомобили*

Имя поля	Тип данных	Описание	Свойства поля
1	2	3	4
Код модели	Числовой	<i>Ключевое поле, код модели по каталогу</i>	Индексированное поле: Да/Совпадения не допускаются <i>Ключевое поле задается в контекстном меню</i>
Марка	Текстовый <i>*Мастер подстановок</i>	<i>Марка автомобиля Audi, BMW, Mazda и др.</i>	Размер поля: 20
Модель	Текстовый	<i>Тип кузова</i>	Размер поля: 20 Индексированное поле: Да/Совпадения допускаются (модель может встречаться в БД с вариантами оснащения)
Год выпуска	Числовой	<i>Год выпуска автомобиля</i>	Значение по умолчанию: 2021
Мощность	Текстовый	Мощность двигателя (кВт/л. с.)	Размер поля: 10 Индексированное поле: Нет
Цвет	Текстовый	Цвет кузова Белый Металлик Серебристый Черный	Размер поля: 20 Индексированное поле: Нет

Окончание таблицы 2.1

1	2	3	4
Количество дверей	Числовой	2 или 3 или 4	Размер поля: Байт Значение по умолчанию: 4 Правило проверки: 2 От 3 От 4 Сообщение об ошибке: Допустимы значения 2, 3, 4
Обивка	Текстовый <i>*Мастер подстановок</i>	Велюр, Кожа, Ткань	Размер поля: 10 Значение по умолчанию: Велюр
Коробка передач	Текстовый <i>*Мастер подстановок</i>	Автоматическая, Механическая	Размер поля: 14 Значение по умолчанию: Автоматическая Правило проверки: Автоматическая От Механическая Сообщение об ошибке: Допустимы только значения Автоматическая или Механическая
Другое оснащение	Длинный текст	Дополнительные аксессуары	Значение по умолчанию: Радио/плеер, круиз-контроль
Стоимость проката за сутки	Денежный	Стоимость проката за сутки	Формат: Денежный Значение по умолчанию: 45,00 р. Правило проверки: ≥ 45

Примечание: Все поля, за исключением поля Другое оснащение, обязательны для заполнения.

При уточнении типов данных и свойств полей принимается во внимание, какая информация в них сохраняется. С точки зрения ускорения ввода данных задаются значения свойства *Значение по умолчанию* для полей *Год выпуска*, *Коробка передач* и др., также для ряда полей – значения для свойства *Правило проверки*.

Макет таблицы *Автомобили* в режиме *Конструктора* приведен на рисунке 2.1.

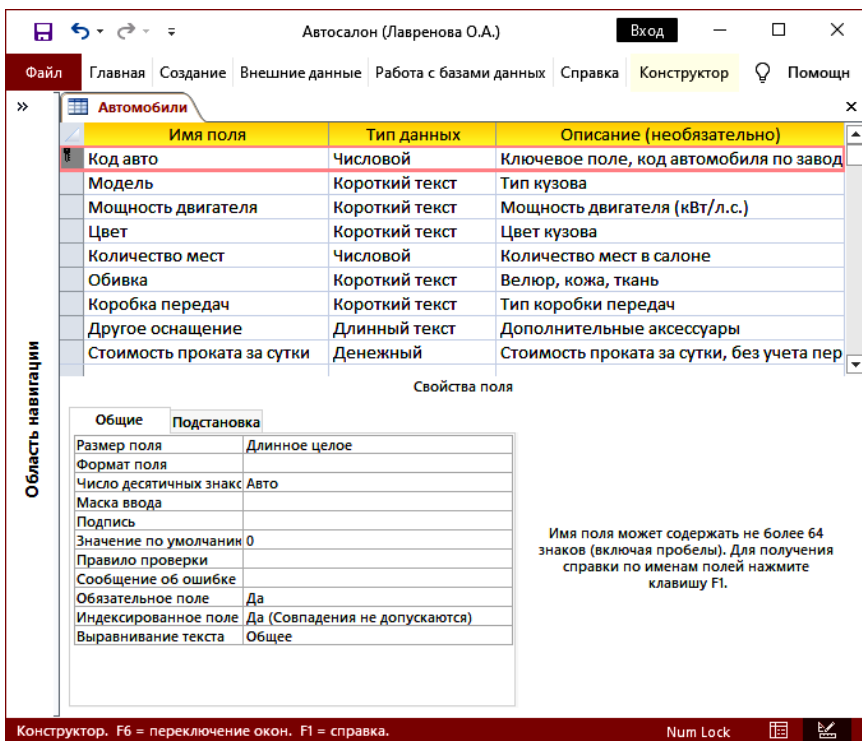


Рисунок 2.1 – Макет таблицы *Автомобили* в режиме *Конструктор*

Выполняется информационное наполнение таблицы *Автомобили*, записи таблицы в режиме *Таблица* приведены на рисунке 2.2.

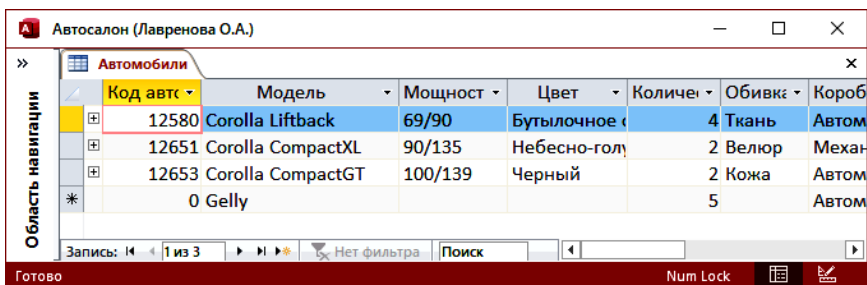


Рисунок 2.2 – Записи таблицы *Автомобили* в режиме *Таблицы*

Внимание: Аналогично оформляется построение других таблиц.

По тексту работы приводятся необходимые иллюстрации, включая рисунки и таблицы.

Для полей в таблицах задаются значения свойств: *Размер поля*, *Формат*, *Маска ввода*, *Значение по умолчанию*, *Правило проверки* (и *Сообщение об ошибке*), *Обязательное поле*, *Индексированное поле* (рекомендуется для полей, по которым будет выполняться сортировка и фильтрация данных).

Для полей, в которых содержатся повторяющиеся значения, рекомендуется создавать столбцы подстановки с фиксированным набором значений.

В таблице-связке *Заказы* для внешних ключевых полей рекомендуется использовать подстановку полей из главных таблиц. Например, при формировании подстановки для внешнего ключа *Код авто* из списка полей таблицы *Автомобили* выбирают 2 поля: *Код авто* и *Модель*; из таблицы *Клиенты* – *Код клиента* и *Фамилия*, из таблицы *Сотрудники* – *Код сотрудника* и *Фамилия*.

В тексте приводятся распечатки окон макета таблицы-связки с отображением вкладки *Подстановка* для внешних ключей.

2.2.2. Формирование схемы БД

Для корректной обработки данных формируется схема БД, устанавливаются связи между таблицами БД в соответствии с информационно-логической моделью предметной области. Для этого выполняется следующая последовательность действий:

1. Закрыть все таблицы БД.
2. Выполнить команду *Схема данных* на вкладке *Работа с базами данных* в группе *Отношения*.
3. Установить (изменить) по ключевым полям связи между таблицами.

Поскольку при создании макетов, как правило, создаются поля с подстановками для внешних ключей, то связи между таблицами формируются автоматически в процессе работы с Мастером подстановок. Поэтому для изменения связи ее достаточно выделить на схеме, затем с помощью команды *КЗМ/Изменить связь* вызвать диалоговое окно «Изменение связей», в котором установить флажок *Обеспечение целостности данных*.

Готовая схема данных отображает связи $1 : \infty$ между таблицами (рисунок 2.3, для отчета распечатка окна БД отображает таблицы на вкладке Таблицы в режиме просмотра сведений).

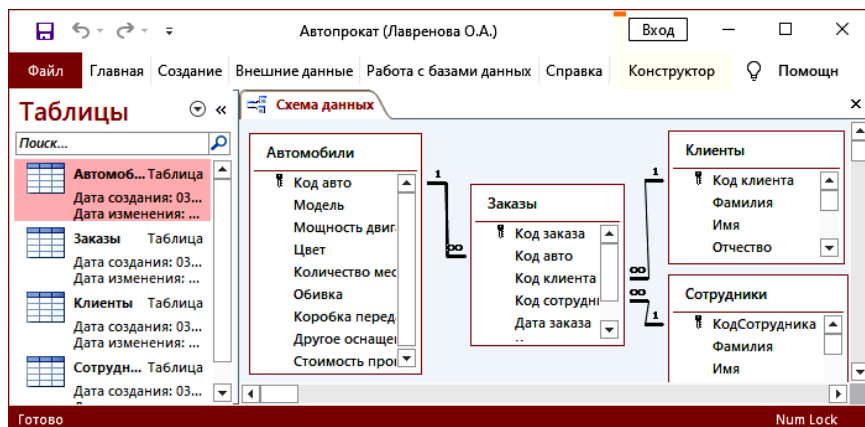


Рисунок 2.3 – Схема БД «Автопрокат»

2.3. Проектирование запросов

Для анализа информации в БД «Автопрокат» по срезам: *Автомобили*, *Клиенты*, *Сотрудники* и *Заказы* разрабатываются запросы. Создание макетов запросов выполняется в режиме *Конструктора*.

Макеты и результаты запросов приведены по тексту работы⁴.

Для анализа информации по срезу *Автомобили* создаются следующие запросы:

1. «*Самая дешевая (дорогая) модель проката*» – позволяет отобразить одну модель с минимальной (максимальной) стоимостью проката за сутки.

Создается на основе таблицы *Автомобили*, включает поля *Модель* и *Стоимость проката за сутки*.

По полю *Стоимость проката за сутки* выполняется сортировка по возрастанию (убыванию).

⁴ Приведен примерный список запросов и может быть изменен в соответствии со спецификой предметной области.

Набор выводимых значений ограничен (в *Свойствах запроса* выбран параметр *Набор значений* и задано значение = 1).

Макет запроса и результат его выполнения (итоговая таблица) приводятся на рисунках по тексту работы.

2. «*Список заказов на модель*» – позволяет сформировать список заказов, оформленных на конкретную модель автомобиля.

Создается на основе таблиц *Автомобили*, *Клиенты* и *Заказы*, включает поля *Код заказа*, *Дата заказа*, *Модель*, *Фамилия клиента*.

В условии отбора по полю *Модель* используется параметр [Введите модель].

Макет запроса и результат его выполнения (окно параметра и итоговая таблица) приводятся на рисунках по тексту работы.

3. «*Рейтинг популярности*» – позволяет сформировать список из трех автомобилей, которые сдавались напрокат чаще всего.

Создается на основе таблиц *Автомобили* и *Заказы*, включает поля *Модель* и *Код заказа*.

Используется групповая операция Count по полю *Код заказа*.

По полю *Код заказа* выполняется сортировка по убыванию.

Набор выводимых значений ограничен тремя (в *Свойствах запроса* выбран параметр *Набор значений* и задано значение = 3).

Макет запроса и результат его выполнения (итоговая таблица) приводятся на рисунках по тексту работы.

4. «*Рейтинг моделей по стоимости проката*» – позволяет сформировать список автомобилей и подсчитать общую стоимость проката.

Создается на основе таблиц *Автомобили* и *Заказы*, включает поля *Модель* и *Сумма заказа* (вычисляемое поле, в котором рассчитывается общая сумма каждого заказа с учетом стоимости проката за сутки и времени проката).

Используется групповая операция Sum по полю *Сумма заказа*.

По полю *Сумма заказа* выполняется сортировка по убыванию.

Макет запроса и результат его выполнения (итоговая таблица) приводятся на рисунках по тексту работы.

Для анализа информации по срезу Клиенты создаются запросы:

5. «Поиск клиента по фрагменту адреса» – позволяет отобразить сведения о клиенте (клиентах), проживающих по заданному адресу.

Создается на основе таблицы *Клиенты*, включает поля: *Фамилия клиента, Адрес, Телефон*.

По полю *Почтовый адрес* для построения выражения в условии отбора используется параметр [Введите фрагмент адреса].

Макет запроса и результат его выполнения (окно параметра и итоговая таблица) приводятся на рисунках по тексту работы.

6. «Список заказов клиента» – позволяет сформировать список заказов и рассчитать общую сумму проката для каждого клиента с учетом стоимости проката за сутки, сроков проката и своевременности возврата транспортного средства.

Создается на основе таблиц *Клиенты, Автомобили* и *Заказы*, включает поля: *Фамилия клиента, Код заказа, Дата заказа, Модель, Сумма заказа* (вычисляемое поле, в котором рассчитывается общая сумма каждого заказа на прокат с учетом стоимости проката за сутки, сроков проката и своевременности возврата транспортного средства).

Сортировка задается по полю *Фамилия клиента*.

По полю *Фамилия клиента* используется параметр [Введите фамилию клиента].

Макет запроса и результат его выполнения (окно параметра и итоговая таблица) приводятся на рисунках по тексту работы.

7. «Рейтинг активности клиентов по числу заказов» – позволяет сформировать список клиентов и упорядочить его по числу заказов.

Создается на основе таблиц *Клиенты* и *Заказы*, включает поля *Фамилия клиента* и *Код заказа*.

Используется групповая операция Count по полю *Код заказа*.

По полю *Код заказа* выполняется сортировка по убыванию.

Макет запроса и результат его выполнения (итоговая таблица) приводятся на рисунках по тексту работы.

8. «Рейтинг клиентов по сумме заказов» – позволяет сформировать список из 3-х клиентов, которые имеют самую большую сумму всех заказов на прокат.

Создается на основе таблиц *Клиенты* и *Заказы*, включает поля *Фамилия клиента* и *Сумма заказа* (вычисляемое поле, в котором рассчитывается общая сумма каждого заказа на прокат с учетом стоимости проката за сутки, сроков проката и своевременности возврата транспортного средства).

Используется групповая операция *Sum* по полю *Сумма заказа*.

По полю *Сумма заказа* выполняется сортировка по убыванию.

Набор выводимых значений ограничен тремя (в свойствах запроса выбрать параметр *Набор значений* и задано значение = 3).

Макет запроса и результат его выполнения (итоговая таблица) приводятся на рисунках по тексту работы.

Для анализа информации по срезу Заказы создаются запросы:

9. «*Неоплаченные заказы*» – позволяет сформировать список неоплаченных заказов.

Создается на основе таблиц *Клиенты*, *Автомобили* и *Заказы*, включает поля: *Код заказа*, *Дата заказа*, *Фамилия клиента*, *Модель*, *Сумма проката* (вычисляемое поле, в котором рассчитывается общая сумма каждого заказа на прокат с учетом стоимости проката за сутки, сроков проката и своевременности возврата транспортного средства), *Оплачено*.

По полю *Оплачено* (логическое поле) в строке *Условие отбора* указывается значение = *Нет*.

Макет запроса и результат его выполнения (итоговая таблица) приводятся на рисунках по тексту работы.

10. «*Список заказов за ... квартал ... года*» – позволяет сформировать список заказов, оформленных в указанном временном интервале.

Создается на основе таблиц *Клиенты*, *Автомобили* и *Заказы*, включает поля: *Код заказа*, *Дата заказа*, *Фамилия клиента*, *Модель*, *Сумма заказа* (вычисляемое поле, в котором рассчитывается общая сумма каждого заказа на прокат с учетом стоимости проката за сутки, сроков проката и своевременности возврата транспортного средства).

Дополнительно создаются 2 дополнительных вычисляемых поля *Год* и *Квартал* с использованием стандартных функций *Year()* и *DatePart ()* соответственно:

Год: Year (Дата заказа)
Квартал: DatePart (“q”; Дата заказа).

В строке *Условие отбора* для полей *Год* и *Квартал* используются параметры [Введите год] и [Введите квартал] соответственно.

Макет запроса и результат его выполнения (окна параметров и итоговая таблица) приводятся на рисунках по тексту работы.

11. «*Оборот за период ...*» – позволяет подвести итоги и рассчитать общую сумму всех заказов за указанный период (год, квартал, месяц).

Строится аналогично запросу 10, далее выполняется подведение итогов с использованием группировки по полям год (квартал, месяц) и статистической функции Sum по полю *Сумма заказа*.

Макет запроса и результат его выполнения (итоговая таблица) приводятся на рисунках по тексту работы.

Описание других запросов для анализа данных приводятся аналогично.

2.4. Проектирование форм и отчетов

Для удобной работы пользователя создаются формы и отчеты.

Для просмотра, ввода и редактирования информации из таблиц БД создаются формы: *Автомобиль*, *Клиент*, *Сотрудник*, *Заказ*.

Примеры разработанных форм приводятся на рисунках по тексту работы или в приложении (распечатки макета формы в режиме Конструктор и окна формы в режиме Форма).

Для вывода информации на печать в удобном для пользователя виде в БД «Автопрокат» разрабатываются отчеты: *Каталог автомобилей*, *Телефонный справочник по клиентам*, *Список неоплаченных заказов*, *Оборот за период*.

Отчеты создаются на основе соответствующих таблиц и запросов.

Примеры разработанных отчетов приводятся на рисунках по тексту работы или в приложении (распечатки окна с макетом отчета в режиме Конструктор и первой страницы отчета в режиме Предварительный просмотр).

2.5. Интеграция БД с приложениями Microsoft Office

Для связи MS Access с приложениями Microsoft Office применяются операции *Импорт* и *Экспорт* данных, которые выполняются на вкладке *Внешние данные* в двух группах команд: 1) *Импорт и связи*, 2) *Экспорт*.

2.5.1. Анализ данных из БД в MS Excel

Для анализа данных в MS Excel экспортируются запросы по срезам *Автомобиль*, ... (*перечислить другие срезы по варианту*). На листах в рабочей книге применяются такие инструменты анализа, как: промежуточные итоги, сводные таблицы и инструменты визуального анализа – диаграммы.

Промежуточные итоги подводятся по срезу *Марка автомобиля* (*срезы для анализа выбираются в соответствии с вариантом*) [13].

Предварительно данные сортируются по выбранному полю (*Марка автомобиля*), затем выполняется команда *Промежуточный итог* на вкладке *Данные* в группе команд *Структура* и заполняется диалоговое окно «Промежуточный итог».

Распечатки диалогового окна «Промежуточный итог» и таблицы с результатами подведения итогов с детализацией 2-го уровня приводятся по тексту работы.

Сводная таблица создается на отдельном листе для анализа данных, *например*, для обобщения данных о заказах, сформированных сотрудниками за весь период работы (с группировкой по месяцам), включая число заказов, в разрезе марок автомобилей. Поле *Марка автомобиля* размещается в области фильтра отчета, поскольку является срезом для анализа и получения отдельных страниц отчета сводной таблицы.

Распечатки макета сводной таблицы и итоговой таблицы с результатами приводятся по тексту работы.

Для визуального анализа данных выполняется построение диаграмм по срезу *Автомобиль* (*срезы для анализа определяются по варианту*):

– *круговая* (*приводится ссылка на рисунок*), для отображения доли каждой марки автомобиля в общей сумме заказов;

– *линейчатая (приводится ссылка на рисунок)*, для отображения рейтинга марок автомобилей по числу заказов;

– *гистограмма (приводится ссылка на рисунок)*, для демонстрации динамики заказов за год по кварталам и др.

Диаграммы приводятся на рисунках по тексту работы, в тексте обязательно приводится ссылка на рисунок.

2.5.2. Подготовка массовой рассылки с помощью процедуры Слияние в MS Word

Для автоматизации доставки клиентам салона типовых сообщений об изменениях стоимости и условий аренды транспортных средств создан макет письма рассылки и получен многостраничный документ рассылки с применением процедуры *Слияние в MS Word*.

Описание порядка использования процедуры Слияние в MS Word, включая макет письма и документ готовой рассылки, приводится по тексту работы.

3.5. Список типовых заданий к аттестации (на примере учебной БД)

Для проверки полученных практических навыков по анализу данных из таблиц реляционной БД предлагаются разноуровневые задания по анализу данных из таблиц на примере учебной БД, схема которой приведена на рис. 3.1.

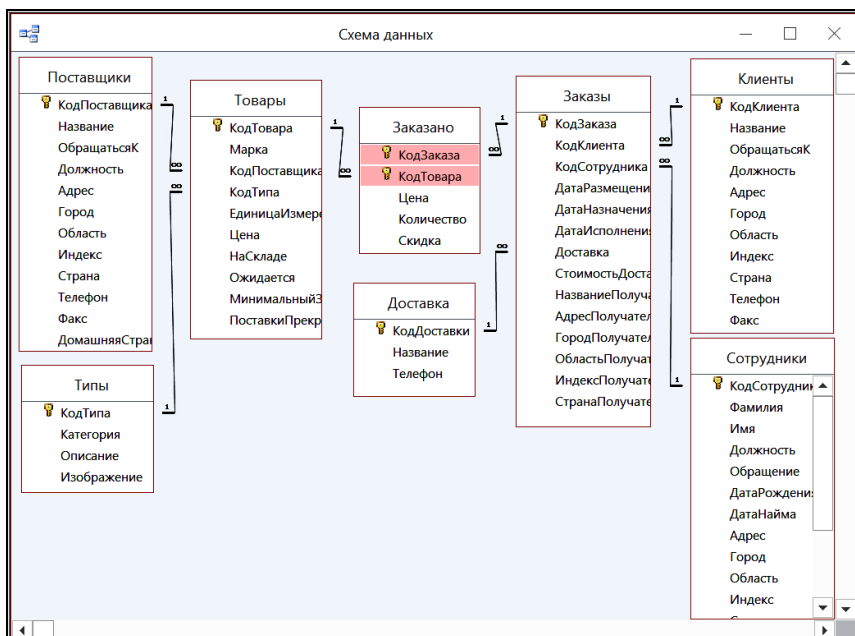


Рис. 3.1. Схема учебной БД

Вид результирующей таблицы (список и порядок полей при построении запроса) определяется в соответствии с постановкой задачи.

1. *Уровень 1.* Формирование запросов на выборку с условиями.

1.1. *Список сотрудников, проживающих по адресу.*

Сформировать запрос на выборку для вывода списка сотрудников, проживающих в городе (значение города выбрать самостоятельно) по адресу (указать фрагмент почтового адреса – улицу).

При записи условия отбора для города и почтового адреса использовать шаблоны * и ?.

1.2. *Список сотрудников, не проживающих в городе.*

Сформировать запрос на выборку для вывода списка сотрудников, не проживающих в городе (значение города выбрать самостоятельно).

Условие отбора задать по полю *Город* с использованием оператора *Not Like*.

1.3. *Список сотрудников, занимающих определенную должность.*

Сформировать запрос на выборку, позволяющий сформировать список сотрудников, занимающих определенную должность (значение должности выбрать самостоятельно).

При записи условия отбора использовать шаблоны * и ?.

1.4. *Самый дешевый (дорогой) товар в категории.*

Сформировать запрос на выборку, позволяющий отобразить информацию об одном товаре (самом дорогом или самом дешевом) из одной категории (значение категории выбрать самостоятельно).

Использовать сортировку по полю *Цена* (порядок сортировки выбрать по условию запроса) и ограничить набор отображаемых значений (1) в свойствах запроса.

1.5. *Список товаров в категории, поставки которых прекращены.*

Сформировать запрос на выборку, позволяющий отобразить список товаров из одной категории (значение категории выбрать самостоятельно), поставки которых уже прекращены.

2. *Уровень 1.* Формирование запросов на выборку с вычислениями и параметрами.

2.1. *Стоимость остатков товаров на складе.*

Сформировать запрос на выборку, который выдает список товаров по категориям, поставки которых прекращены, но которые еще есть на складе. Рассчитать стоимость остатков на складе.

Имя вычисляемого поля: *Стоимость остатков на складе.*

2.2. *Стоимость остатков товаров на складе (в евро).*

Сформировать запрос на выборку, который выдает список товаров по категориям, поставки которых прекращены, но которые еще есть на складе. Рассчитать стоимость остатков на складе в евро.

Имя вычисляемого поля: *Стоимость остатков на складе, евро.*

В выражении для расчета стоимости товаров в валюте используйте в качестве курса валюты *параметр* с именем *Курс евро*, а не конкретное числовое значение.

2.3. *Стоимость минимального запаса товаров.*

Сформировать запрос на выборку, который выдает список товаров из заданной категории, поставки которых не прекращены. Рассчитать стоимость минимального запаса товаров.

Имя вычисляемого поля: *Стоимость минимального запаса.*

В условии отбора для поля *Категория* использовать параметр.

2.4. *Список товаров из заданной категории, которые есть на складе.*

Сформировать запрос на выборку, который выдает список товаров из заданной категории, поставки которых не прекращены и которые есть на складе.

В условии отбора по полю *Категория* использовать *параметр* (для фрагмента названия категории), а не конкретное значение.

Имя вычисляемого поля: *Стоимость минимального запаса.*

2.5. *Стоимость заказов с учетом скидки.*

Сформировать запрос на выборку, который выдает список заказов на товары из заданной категории, сформированные конкретным сотрудником. Рассчитать стоимость заказов с учетом скидки.

Имя вычисляемого поля: *Стоимость заказа со скидкой.*

В условии отбора для полей *Категория* и *Фамилия сотрудника* использовать параметры.

3. *Уровень 3.* Формирование итоговых запросов и графическое представление результатов анализа.

При построении итоговых запросов использовать групповые операции, сортировку и ограничение набора значений.

3.1. *Пять стран с наибольшим числом клиентов.*

Сформировать итоговый запрос, отображающий список из пяти стран с наибольшим числом клиентов (*использовать групповую операцию, сортировку и ограничение набора значений*).

3.2. *Рейтинг сотрудников по числу сформированных заказов.*

Сформировать итоговый запрос, отображающий рейтинг сотрудников по числу сформированных заказов.

3.3. *Рейтинг сотрудников по стоимости сформированных заказов.*

Сформировать итоговый запрос, отображающий рейтинг сотрудников по стоимости сформированных заказов с учетом скидки.

3.4. *Рейтинг категорий товаров по сумме заказов.*

Сформировать итоговый запрос, отображающий рейтинг категорий товаров по общей сумме сформированных заказов (с учетом скидки).

3.5. *Товары – лидеры по объемам продаж.*

Сформировать список из 5-ти марок товаров, которые были проданы в наибольшем количестве.

3.6. *Сводная ведомость продаж.*

Сформировать перекрестный запрос – сводную ведомость продаж (в стоимостном выражении) для каждого сотрудника по каждой категории товаров.

Для поля *Категория* подвести итоги.

3.7. Динамика числа заказов по месяцам за каждый год.

Сформировать итоговый запрос, отображающий динамику числа заказов по месяцам за каждый год.

Для решения задачи создать вычисляемые поля, позволяющие определить год и месяц по известной дате заказа и рассчитать число заказов:

Год: Year (Дата заказа);

Месяц: Month (Дата заказа);

Число заказов: Count (Код заказа).

Групповые функции и порядок сортировки по полям задать самостоятельно.

3.8. Динамика объемов продаж по кварталам за каждый год.

Сформировать итоговый запрос, отображающий динамику объемов продаж по кварталам за каждый год.

Для решения задачи создать вычисляемые поля, позволяющие определить год и квартал по известной дате заказа и рассчитать стоимость заказов:

Год: Year (дата заказа);

Квартал: DatePart ("q"; дата заказа);

Стоимость заказа: Цена*Количество*(1-Скидка).

Групповые функции, условия отбора и порядок сортировки по полям задать самостоятельно.

Графическое представление результатов анализа выполняется в рабочей книге MS Excel после экспорта соответствующих запросов.

3.9. Диаграмма «Динамика заказов по месяцам за год».

Построить диаграмму для отображения динамики числа заказов по месяцам года (запрос 3.7).

Обосновать выбор типа диаграммы.

3.10. Диаграмма «Доля категорий в объеме продаж».

Построить диаграмму для отображения доли каждой категории в общем объеме продаж (запрос 3.4).

Обосновать выбор типа диаграммы.

3.11. Диаграмма «Рейтинг стран по числу клиентов»

Построить диаграмму для отображения рейтинга сотрудников по объемам продаж (запрос 3.3).

Обосновать выбор типа диаграммы.

3.6. Краткие требования к оформлению контрольной работы

Работа оформляется с использованием офисного редактора в виде текстового документа с соблюдением требований к оформлению текстовых документов, приведенных в ГОСТ 2.105-95.

Основные требования:

1. Размер бумаги А4, ориентация бумаги – книжная.

2. Размеры полей:

левое – 3 см;

правое – 1 см;

верхнее – 2,5 см;

нижнее – 2 см

3. Шрифт в документе – Times New Roman. Размер шрифта 12 пт. Для выделения заголовков – полужирный.

4. Междустрочный интервал выбирается таким образом, чтобы на странице размещалось не более 39–40 строк.

5. Абзацный отступ в тексте – 1,25 см.

6. Заголовки разделов Содержание, Введение, Заключение и Список использованных источников не нумеруются, записываются симметрично тексту (*по центру*) без абзацного отступа.

7. Заголовки основных разделов и подразделов нумеруются и располагаются с абзацного отступа.

8. В тексте работы обязательны ссылки на литературу, по образцу: [1].

9. В тексте работы обязательны рисунки и таблицы с соответствующими ссылками на них в тексте до рисунка или таблицы. Например, ... на рисунке 1 представлено ..., ... в таблице 1 приведены... и т. п.

10. Страницы нумеруются арабскими цифрами без точки в правом верхнем углу страницы. На титульном листе работы номер страницы не ставится, но учитывается в общей нумерации.

11. Работа распечатывается с одной стороны листа и сшивается по левому краю в папку-скоросшиватель.

12. На титульном листе отмечается дата сдачи работы на проверку, ставится подпись исполнителя.

СПИСОК СОКРАЩЕНИЙ

ANSI – American National Standards Institute.

API (Application Programming Interface) – программный интерфейс приложения/

ISO – International Standart Organization.

MS – Microsoft.

MS Access – Microsoft Office Access.

ODBC (Open Database Connectivity) – программный интерфейс доступа к базам данных, разработанный компанией Microsoft.

OLE (Object Linking and Embedding) – технология связывания и внедрения объектов в приложениях ОС Windows.

OPDBMS (Operational Database Management Systems) – операционные СУБД.

RDF (Resource Description Framework) – среда описания ресурса, графовая модель данных в виде триплетов.

SQL – Structured Query Language.

VBA – Visual Basic for Application.

БД – база данных.

БО – буфер обмена (Clipboard).

ИЛИМ ПО – информационно-логическая (инфологическая) модель предметной области.

ИО – информационный объект.

КЗМ – контекстно-зависимое (контекстное) меню.

КУК – клавиши управления курсором.

ЛКМ – левая кнопка мыши.

ООП – объектно-ориентированное программирование.

ООЯП – объектно-ориентированный язык программирования.

ОС – операционная система.

ПБД – Панель быстрого доступа (в приложениях ОС Windows).

ПЗ – Панель задач.

ПК – персональный компьютер.

ПКМ – правая кнопка мыши.

ПО – предметная область-

ППП – пакеты прикладных программ.

СУБД – система управления базами данных.

УО – условие отбора.

ЭУ – элемент управления.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Gartner: облачные базы данных стремительно взрослеют [Электронный ресурс]. – Режим доступа: <https://www.itweek.ru/its/article/detail.php?ID=225774/>. – Дата доступа: 01.07.2023.
2. INTUIT.ru: Интернет университет информационных технологий – дистанционное образование [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/>. – Дата доступа: 01.07.2023.
3. ISO/IEC 9075-1:2023(en). Information technology – Database languages SQL – Part 1: Framework (SQL/Framework). [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/76583.html>. – Дата доступа: 01.07.2023.
4. Magic Quadrant for Cloud Database Management Systems [Электронный ресурс]. – Режим доступа: <https://www.gartner.com/doc/reprints?id=1-2C00XM3H&ct=221213&st=sb>. – Дата доступа: 01.07.2023.
5. SQL: что это такое, как работает язык баз данных и СУБД - основы и структура. [Электронный ресурс]. – Режим доступа: <https://blog.skillfactory.ru/glossary/sql/>. – Дата доступа: 01.07.2023.
6. 2SQL.ru – Информационный ресурс о языке SQL [Электронный ресурс]. – Режим доступа: <http://2sql.ru/>. – Дата доступа: 01.07.2023.
7. Данные [Электронный ресурс]. – Режим доступа: <https://www.tadviser.ru/index.php/Статья:Данные>. – Дата доступа: 01.07.2023.
8. Дашкевич, Н. В. Основы проектирования реляционных баз данных средствами СУБД Microsoft Access [Электронный ресурс]: [методическое пособие по выполнению лабораторных работ по дисциплине «Компьютерные информационные технологии» для специальности 1-25 01 07 «Экономика и управление на предприятии»] / Н. В. Дашкевич, О. А. Лавренова, С. М. Козлов. – Минск : БНТУ, 2010.
9. Домашняя страница Access – Microsoft Office Online [Электронный ресурс]. – Режим доступа: <http://office.microsoft.com/ru-ru/access/>. – Дата доступа: 01.07.2023.
10. Иллюстрированный самоучитель по Access [Электронный ресурс]. – Режим доступа: <http://www.subd.org/>. – Дата доступа: 01.07.2023.

11. Информатика: базовый курс : учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симонович. – 3-е изд. – СПб. : Питер, 2017. – 637 с.

12. Лавренова, О. А. Информационные технологии в экономике : пособие для обучающихся по специальности 1-27 01 01 «Экономика и организация производства (по направлениям)» : в 2 ч. / О. А. Лавренова, Б. А. Железко. – Минск : БНТУ, 2021. – Ч. 1. – 129 с.

13. Лавренова, О. А. Информационные технологии в экономике : пособие для обучающихся по специальности 1-27 01 01 «Экономика и организация производства (по направлениям)» : в 2 ч. / О. А. Лавренова, Б. А. Железко. – Минск : БНТУ, 2022. – Ч. 2. – 2022. – 224 с.

14. Лавренова, О. А. Сетевые технологии и базы данных: Курс лекций для студентов специальности 1-27 01 01 «Экономика и организация производства» : в 2 ч. / О. А. Лавренова. – Минск : БНТУ, 2009. – Ч. 1: Основы работы с реляционными базами данных и СУБД. – 108 с.

15. НОУ ИНТУИТ. Лекция. Классификация БД и СУБД [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/3439/681/lecture/14019>. Дата доступа: 01.07.2023.

16. НОУ ИНТУИТ | Основы SQL | Информация. [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/5/5/info>. – Дата доступа: 01.07.2023.

17. Редмонд, Эрик. Семь баз данных за семь недель: введение в современные базы данных и идеологию NoSQL / Эрик Редмонд, Джим Р. Уилсон, под ред. Жаклин Картер, пер. А. А. Слинкин. – Москва : ДМК Пресс, 2015. – 383 с.

18. Сетевые технологии и базы данных : лабораторные работы для студентов специальности 1-27 01 01 «Экономика и организация производства» : в 2 ч. – Минск : БНТУ, 2008. – Ч. 1 : Основы работы с реляционными базами данных и СУБД / сост. О. А. Лавренова. – 2008. – 90 с.

19. СУБД: что такое системы управления базами данных, каких видов они бывают и для чего они нужны /SkillBox Media [Электронный ресурс]. – Режим доступа: <https://skillbox.ru/media/code/sistema-upravleniya-bazami-dannykh-cto-eto-takoe-i-zachem-ona-nuzhna/>. – Дата доступа: 01.07.2023.

20. Уроки по Microsoft Access. [Электронный ресурс]. – Режим доступа: https://www.youtube.com/channel/UC_7MFrFsvQZjzgay_gtItYw/. – Дата доступа: 01.07.2023.

21. Экономическая информатика: учебник для вузов / В. В. Евдокимов [и др.] ; под ред. В. В. Евдокимова. – СПб. : Питер Паблишинг, 1997. – 592 с.

22. Электронный учебно-методический комплекс по учебной дисциплине «Сетевые технологии и базы данных» для студентов специальности 1-27 01 01 «Экономика и организация производства (по направлениям)» / сост. О. А. Лавренова. – Минск : БНТУ, 2020. [Электронный ресурс] – Режим доступа: <https://rep.bntu.by/handle/data/81236>. – Дата доступа: 01.07.2023.

Приложение А

Таблица А.1

Примеры функций категории *Дата и время*

Функции	Описание, результат
1	2
Дата ()	Возвращает значение типа Variant (Date), текущую системную дату
DateAdd («interval»; «number»; «Дата»)	Возвращает значение типа Variant (Date), дату как результат прибавления к значению аргумента «Дата» числа интервалов времени, указанных в аргументах «number» и «interval» соответственно <i>Например:</i> DateAdd (“m”; 1; 01.09.2023) <i>Результат:</i> 01.10.2023
DatePart («interval»; «Дата» [; «firstdayofweek»] [; «firstweekofyear»])	Возвращает значение Variant (Integer), содержащее для данной даты часть, указанную аргументом «interval» <i>Например:</i> DatePart (“m”; 01.09.2023) <i>Результат:</i> 09
DateSerial («year»; «month»; «day»)	Возвращает значение типа Variant (Date), дату для указанного года, месяца и дня
Day («Дата»)	Возвращает значение типа Variant (Integer), целое число в диапазоне от 1 до 31 включительно, день месяца
Hour («time»)	Возвращает значение Variant (Integer), целое число в диапазоне от 0 до 23 включительно, час дня
Minute («time»)	Возвращает значение типа Variant (Integer), целое число в диапазоне от 0 до 59 включительно, минуты часа
Month («Дата»)	Возвращает значение Variant (Integer), целое число в интервале от 1 до 12, порядковый номер месяца года
MonthName («month» [; «abbreviate»])	Возвращает строку, указывающую название заданного месяца в полном или сокращенном виде

1	2
Now ()	Возвращает значение Variant (Date), указывающее системную дату и время в соответствии с настройками компьютера
Second («time»)	Возвращает переменную типа Variant (Integer), определяющую целое число в диапазоне от 0 до 59 включительно, которое представляет секунды, извлеченные из значения времени
Time ()	Возвращает Variant (Date), указывая текущее системное время
Weekday («Дата» [; «firstdayofweek»])	Возвращает значение переменной Variant (Integer), отображающее дни недели
WeekdayName («weekday» [; «abbreviate»] [; «firstdayofweek»])	Возвращает строку, в которой указан определенный день недели
Year («Дата»)	Возвращает значение Variant (Integer), содержащее целое число, которое представляет год

Описание аргументов:

«Дата» – значение или литерал, представляющий дату (любое числовое выражение или строковое выражение или любое сочетание, которое может представлять дату);

«abbreviate» – *необязательный аргумент*, значение, которое указывает, в полном или сокращенном виде выводится название (дня недели или месяца). Если значение аргумента не указано, то по умолчанию задается значение *False*, которое означает, что название не сокращается. Для вывода в сокращенном виде необходимо указать значение 2;

«day» – число или числовое выражение в диапазоне от 1 до 31;

«interval» – строковое выражение, обозначающее интервал времени, используемый в качестве аргумента:

“уууу” – год;

“q” – квартал;

“m” – месяц;

“y” – день года;

“d” – день месяца;

“w” – день недели;

“ww” – неделя;

«firstdayofweek» – *необязательный аргумент*, константа, задающая первый день недели. Если она не указана, по умолчанию принимается воскресенье. Для отсчета с понедельника необходимо указать значение 2;

«firstweekofyear» – *необязательный аргумент*, константа, задающая первую неделю года. Если она не указана, первой неделей является неделя, начинающаяся 1 января;

«month» – число или числовое выражение для обозначения месяца в диапазоне от 1 до 12;

«number» – число или числовое выражение, представляющее количество интервалов времени, которые требуется добавить к дате. Значение аргумента может быть положительным – для получения дат в будущем, или отрицательным – для получения дат в прошлом;

«time» – числовое выражение, строковое выражение или любое сочетание, которое может представлять время;

«weekday» – число или любое числовое выражение для обозначения дня недели в диапазоне от 1 до 7;

«year» – число или числовое выражение для обозначения года в диапазоне от 100 до 9999 включительно.

Таблица А.2

Примеры функций категории *Преобразование*

Функции	Описание, результат
1	2
Asc («string»)	Возвращает целое значение, представляющее код символа, который соответствует первому символу строки
CCur («expression»)	Преобразует выражение к типу Currency (денежный)
CDate («expression»)	Преобразует выражение к типу Date (дата)
CInt («expression»)	Преобразует выражение к типу Integer (целое)

1	2
FormatCurrency («expression»; [; «precision»] [;«leadingdigit»] [; «parensfornegative»] [; «groupdigits»])	Возвращает выражение в формате денежного значения с обозначением символа валюты, определенного настройками системы в панели управления
FormatDateTime («Дата»; «namedformat»)	Возвращает выражение, отформатированное как дата или время
FormatPercent («expression» [; «precision»] [;«leadingdigit»] [; «parensfornegative»] [; «groupdigits»])	Возвращает выражение в формате процентного значения (умноженного на 100) со знаком % после числа
FormatNumber («expression» [; «precision»] [;«leadingdigit»] [; «parensfornegative»] [; «groupdigits»])	Возвращает выражение, отформатированное в виде числа
Str («number»)	Возвращает значение типа Variant (String), содержащее строковое представление числа, заданного аргументом «number»
Val («string»)	Возвращает число, содержащееся в строке, как числовое значение соответствующего типа

Описание аргументов:

«Дата» – значение или литерал, представляющий дату (любое числовое выражение или строковое выражение или любое сочетание, которое может представлять дату);

«expression» – это любое строковое или числовое выражение;

«groupdigits» – *необязательный аргумент*, константа, указывающая, группируются ли числа с помощью разделителя групп разрядов, заданного в региональных настройках панели управления;

«leadingdigit» – *необязательный аргумент*, константа с тремя состояниями, определяющая, будет ли отображаться ноль в начале дробных значений меньше 1;

«namedformat» – *необязательный аргумент*, числовое значение, показывающее используемый формат даты/времени;

«number» – число или любое допустимое числовое выражение;
 «precision» – *необязательный аргумент*, числовое значение, указывающее, сколько знаков отображается справа от десятичного делителя. Значение по умолчанию = 1, указывающее, что используются региональные параметры в соответствии с настройками панели управления;

«parensfornegative» – *необязательный аргумент*, константа с тремя состояниями, определяющая, следует ли помещать отрицательные числа в скобки;

«string» – любое допустимое строковое выражение.

Таблица А.3

Примеры функций категории *Проверка*

Функции	Описание, результат
IsDate («expression»)	Возвращает значение <i>True</i> , если выражение является датой или распознается как допустимая дата или время, в противном случае возвращает значение <i>False</i>
IsError («expression»)	Возвращает значение типа Boolean, указывающее, является ли аргумент «expression» значением ошибки
IsNull («expression»)	Возвращает значение типа Boolean, которое указывает, не содержит ли аргумент «expression» несуществующих значений (Null)
TypeName («varname»)	Возвращает строку данных (String), которые предоставляют информацию о переменной
VarType («varname»)	Возвращает целое число, указывающее подтип переменной или тип свойства объекта по умолчанию

Описание аргументов:

«expression» – это любое строковое или числовое выражение;

«varname» – обязательный аргумент представляет собой любое строковое или числовое выражение, содержащий имя любой переменной.

Примеры функций категории *Статистические*

Функции	Описание, результат
Avg («expression»)	Вычисляет среднее арифметическое для набора значений, содержащихся в указанном поле
Count («expression»)	Вычисляет количество записей, возвращаемых запросом
Max («expression»)	Возвращает максимальное из набора значений, содержащихся в указанном поле
Min («expression»)	Возвращает минимальное из набора значений, содержащихся в указанном поле
Sum («expression»)	Возвращает сумму значений в наборе значений, содержащихся в указанном поле
StDev («expression») или StDevP («expression»)	Возвращает расчетное стандартное отклонение для генеральной совокупности или для ее выборки, представленной в виде набора значений, содержащихся в указанном поле. Если базовый запрос содержит менее двух записей (или, в случае функции StDevP, не содержит записей), эти функции возвращают значение Null, показывающее, что стандартное отклонение рассчитать невозможно
Var («expression») или VarP («expression»)	Возвращает оценки дисперсии для генеральной совокупности или ее выборки, представленной в виде набора значений, содержащихся в указанном поле. Если базовый запрос содержит менее двух записей, функции возвращают значение Null, которое показывает, что отклонение рассчитать невозможно

Описание аргументов:

«expression» – это любое строковое или числовое выражение для обозначения набора значений, содержащихся в указанном поле.

Примеры функций категории *Текстовые*

Функции	Описание, результат
Chr («charcode») Chr\$ («charcode»)	Возвращает значение типа String, которое содержит символ, связанный с указанным кодом символа
Format («expression»; [«format»]; [«firstdayofweek»]; [«firstweekofyear»])	Возвращает значение типа Variant (String), строковое выражение, содержащее аргумент «expression», отформатированный в соответствии с инструкциями, включенными в выражение формата
LCase («string»)	Возвращает значение типа Variant (String), содержащее указанную строку, символы которой преобразованы в нижний регистр
Left («string»; «length»)	Возвращает значение типа Variant (String), строку, содержащую указанное количество символов из левой части строки
Len («string»)	Возвращает значение типа Long, содержащее число символов в строке или число байтов, необходимое для хранения переменной
Mid («string»; «start»; «length»)	Возвращает значение типа Variant (String), содержащее указанное число «length» символов строки, выбранных, начиная с позиции «start»
Right («string»; «length»)	Возвращает значение типа Variant (String), строку, содержащую указанное число символов из правой части строки
StrComp («string1»; «string2» [; «compare»])	Возвращает переменную Variant (Integer), показывающую результат сравнения строк, в соответствии с заданным типом сравнения: -1, если string1 меньше, чем string2; 0, если string1 равна string2; 1, если string1 больше, чем string2; Null, если string1 или string2 равна Null
UCase («string»)	Возвращает значение типа Variant (String), содержащее указанную строку, символы которой преобразованы в верхний регистр

Описание аргументов:

«charcode» – число – код символа, который идентифицирует символ в таблице кодировки;

«compare» – *необязательный аргумент*, который задает тип сравнения строк: 0 – двоичное сравнение, 1 – текстовое сравнение,

2 – сравнение на основе сведений из базы данных (только Microsoft Access).

Если аргумент имеет значение Null, возникает ошибка. Если аргумент не задан (опущен), тип сравнения определяется параметром Option Compare;

«firstdayofweek» – *необязательный аргумент*, константа, задающая первый день недели: 0 – использовать настройку API многоязыковой поддержки, 1 – Воскресенье (по умолчанию), 2 – Понедельник, 3 – Вторник, 4 – Среда, 5 – Четверг, 6 – Пятница, 7 – Суббота;

«firstweekofyear» – *необязательный аргумент*, константа, задающая первую неделю года (1 – по умолчанию начать с недели, содержащей 1 января);

«format» – *необязательный аргумент*, допустимое именованное или определяемое пользователем выражение формата, с использованием кодов формата для строковых выражений (@, &, >, < !), для числовых значений (0, #, ., % и др.), для значений даты (d, dd, ddd, dddd, m, mm, mmm, mmmm, w, ww, y, yy, yuuu), времени (h, hh, n, nn, s, ss);

«length» – число или допустимое числовое выражение, указывающее, сколько знаков должно быть возвращено. Если значение аргумента равно «0», то возвращается пустая строка (""). Если значение аргумента равняется или превышает число знаков аргумента «string», то возвращается вся строка целиком;

«start» – позиция символа аргумента «string», с которой функция Mid выполняет отбор заданного числа символов. Если значение этого аргумента больше, чем число символов в аргументе «string», будет возвращена строка нулевой длины;

«string», «string1», «string2» – строка или любое строковое выражение.

Примеры функций категории *Управление*

Функции	Описание, результат
Choose («index»; «choice-0» [; «choice-1»; ... «choice-n»...])	<p>Выбирает и возвращает значение из списка аргументов</p> <p><i>Например:</i> =Choose ([Доставка]![ТипДоставки]; "Почта"; "Минтранс"; "Иное")</p>
If («expression»; «truepart»; «falsepart»)	<p>Возвращает одну из двух частей в зависимости от результата проверки выражения «expression»: «truepart» – если «expression» равно <i>True</i> и «falsepart» в противном случае</p> <p><i>Например:</i> =If ([Клиенты]![Страна]="Беларусь"; "Белорусский"; "Другой язык")</p>
Switch («expr-0»; «value-0» [; «expr-1»; «value-1»; ... «expr-n»; «value-n»])	<p>Оценивает список выражений «expr-0» ... «expr-n» и возвращает значение «value-0» типа <i>Variant</i> или выражение, которое связано с первым выражением в списке, имеющим значение <i>True</i></p> <p>Функция Switch возвращает значение Null в двух случаях: если ни одно из выражений «expr» не имеет значения <i>True</i> или с первым выражением «expr», имеющим значение <i>True</i>, связано выражение «value» со значением Null</p> <p><i>Например:</i> = Switch ([Клиенты]![Город] = "Лондон"; "Англия"; [Клиенты]![Город] = "Рим"; "Италия"; [Клиенты]![Город] = "Минск", "Беларусь")</p>

Описание аргументов:

«choice-|0» – выражение типа Variant, содержащее один из возможных вариантов выбора;

«choice-1» – *необязательный аргумент*, выражение из набора вариантов выбора;

«expression» – логическое выражение (условие для проверки), которое возвращает значение *True* или *False*;

«expr-|0» – логическое выражение (условие для проверки), возвращает значение *True* или *False*;

«falsepart» – значение или выражение, возвращаемое функцией, если значение аргумента «*expression*» равно *False*;

«index» – числовое выражение или поле, возвращающее значение в диапазоне от 1 до числа доступных вариантов выбора;

«truepart» – значение или выражение, возвращаемое функцией, если значение аргумента «*expression*» равно *True*;

«value-|0» – значение или выражение, которое необходимо вернуть, если соответствующее выражение «*expr-|0*» имеет значение *True*.

Учебное издание

ЛАВРЕНОВА Ольга Анатольевна

**СЕТЕВЫЕ ТЕХНОЛОГИИ
И БАЗЫ ДАННЫХ**

Пособие

для обучающихся по специальности
6-05-0718-01 «Инженерная экономика»
профилизаций «Бизнес-процессы промышленных предприятий»,
«Цифровое производство»

В 2 частях

Часть 1

Редактор *П. П. Горбач*
Компьютерная верстка *Н. А. Школьниковой*

Подписано в печать 29.04.2024. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.

Усл. печ. л. 19,01. Уч.-изд. л. 15,32. Тираж 100. Заказ 942.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет.
Свидетельство о государственной регистрации издателя, изготовителя, распространителя
печатных изданий № 1/173 от 12.02.2014. Пр. Независимости, 65. 220013, г. Минск.