

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
"БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА"

Кафедра "Информационные технологии"

Т. Л. ШИНКЕВИЧ, Т. А. ГОЛДОБИНА

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗ ДАННЫХ В MS ACCESS 2010

**Учебно-методическое пособие
по выполнению расчетно-графической работы**

Гомель 2014

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
"БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА"

Кафедра "Информационные технологии"

Т. Л. ШИНКЕВИЧ, Т. А. ГОЛДОБИНА

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗ ДАННЫХ В MS ACCESS 2010

*Одобрено научно-методической комиссией
гуманитарно-экономического факультета
в качестве учебно-методического пособия
по выполнению расчетно-графической работы
для студентов специальностей «Коммерческая деятельность» и
«Бухгалтерский учет, анализ и аудит (по направлениям)»*

Гомель 2014

УДК 004.65 (075.8)
ББК 32.81
Ш62

Рецензент – канд. экон. наук, доцент кафедры "Информационные технологии"
Л. А. Иоффе (УО "БелГУТ")

Шинкевич, Т. Л.

Ш62 Проектирование и разработка баз данных в MS Access 2010 : учеб.-метод. пособие / Т. Л. Шинкевич, Т. А. Голдобина ; М-во образования Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2014. – 70 с.

ISBN 978-985-554-273-6

Приведены краткие теоретические сведения, методические указания и справочные материалы, раскрывающие основы проектирования, разработки и использования баз данных в СУБД *MS Access 2010* на практических примерах пошагового выполнения расчетно-графической работы. Включает варианты заданий и требования к оформлению РГР.

Предназначено для студентов I курса специальностей "Бухгалтерский учет, анализ и аудит (по направлениям)", "Коммерческая деятельность" дневной формы обучения. Рекомендуется также студентам других специальностей в качестве дополнительного учебного материала по основам проектирования и разработки баз данных в СУБД *MS Access 2010*.

УДК 004.65 (075.8)
ББК 32.81

ISBN 978-985-554-273-6

© Шинкевич Т. Л., Голдобина Т. А., 2014
© Оформление. УО «БелГУТ», 2014

О Г Л А В Л Е Н И Е

Введение	5
1 Проектирование базы данных	7
1.1 Основные понятия	7
1.2 Проектирование баз данных	7
1.3 Нормализация данных, нормальные формы.....	9
1.4 Проектирование базы данных «Бронирование билетов».....	10
2 Создание базы данных	15
2.1 Запуск MS Access	15
2.2 Особенности интерфейса MS Access 2010.....	16
3 Разработка базы данных	18
3.1 Таблицы.....	18
3.1.1 Создание и сохранение таблицы	18
3.1.2 Типы данных	19
3.1.3 Заполнение полей таблицы в окне конструктора.....	20
3.1.4 Свойства полей	20
3.1.5 Задание маски ввода.....	21
3.1.6 Задание свойства Подстановка.....	22
3.1.7 Ключевое поле	23
3.1.8 Межтабличные связи.....	24
3.1.9 Создание схемы данных.....	24
3.1.10 Задание условий целостности.....	26
3.1.11 Работа с таблицами.....	27
3.1.12 Ввод данных в таблицу	27
3.2 Формы.....	28
3.2.1 Создание форм.....	28
3.2.2 Создание простой формы в режиме конструктора	30
3.2.3 Мастер форм	32
3.2.4 Создание подчиненной формы.....	32
3.2.5 Изменение структуры формы в режиме конструктора.....	33
3.2.6 Элементы управления	33
3.2.7 Создание заголовка формы	35
3.3 Запросы.....	35
3.3.1 Создание запроса в режиме Конструктора	36
3.3.2 Структура бланка запроса.....	36
3.3.3 Формирование условий отбора в запросах.....	37
3.3.4 Создание запроса с помощью Мастера.....	38
3.3.5 Создание запроса с параметром	38
3.3.6 Создание запроса на совпадение	40
3.3.7 Создание итоговых запросов	40
3.3.8 Построитель выражений. Создание вычисляемых полей в запросах.....	42

3.3.9	Использование встроенных функций для вычислений	43
3.3.10	Создание перекрестного запроса.....	44
3.3.11	Создание запроса на обновление данных в таблице.....	47
3.3.12	Формирование запроса на удаление записей из таблиц базы данных	48
3.3.13	Разрешения на выполнение запросов манипулирования данными в MS Access 2010.....	49
3.4	Отчеты	50
3.4.1	Способы создания отчетов.....	50
3.4.2	Панель элементов	52
3.4.3	Создание отчета при помощи мастера	53
3.4.4	Создание вычисляемых и итоговых полей	54
3.5	Кнопочная форма	55
3.5.1	Создание макросов	55
3.5.2	Главная кнопочная форма.....	55
4	Задания на расчетно-графическую работу	59
4.1	Общие требования	59
4.2	Варианты заданий	60
4.3	Требования к оформлению	68
4.4	Требования к форматированию итогового отчета	69
	Список использованной и рекомендуемой литературы	70

ВВЕДЕНИЕ

Информация – самое ценное достижение человечества. Накапливаемые данные и знания необходимо сохранять, обрабатывать и результативно использовать. В современных экономических условиях успешное функционирование различных фирм, организаций и предприятий невозможно без развитой *информационной системы* (ИС), которая позволяет автоматизировать сбор и обработку данных, помогает эффективно управлять предприятием. Появление информационных систем явилось ответом на требования мира бизнеса в области применения *персональных компьютеров* (ПК).

Современные *автоматизированные информационные системы* (АИС) позволяют собирать, надежно хранить и оперативно обрабатывать информацию. Современная система обработки данных должна:

- 1) обеспечивать получение общих или детализованных данных работы предприятия за любой промежуток времени;
- 2) позволять вовремя определять тенденции изменения важнейших показателей деятельности предприятия;
- 3) предоставлять критическую информацию и обеспечивать принятие оперативных решений;
- 4) выполнять полный и точный анализ данных.

Любая целенаправленная деятельность начинается с систематизации изучаемых объектов и отношений между ними. Главными функциями всех систем управления являются накопление и поиск информации, которые успешно реализуются путем создания баз данных.

Ядром современной системы управления предприятия являются *базы данных* (БД) и *системы управления базой данных* (СУБД).

База данных (БД) – поименованная и организованная (структурированная) совокупность взаимосвязанных данных, которые отражают состояние объектов и их отношений в рассматриваемой предметной области и находятся под центральным программным управлением.

Предметная область – это некоторая область человеческой деятельности или часть реального мира, на основе которой создается БД и её структура. База данных содержит такой минимально необходимый объем информации, который позволяет удовлетворять запросы каждого из многочисленных пользователей.

Доступ пользователя к базе данных и все операции с данными реализуются программой, называемой *системой управления базами данных* (СУБД, от англ. DBMS – Database Management System). СУБД – это совокупность языковых и программных средств, предназначенных для создания, наполнения, обновления и удаления баз данных.

СУБД можно рассматривать как программную оболочку, которая находится между базой данных и пользователем (рисунок 1). Она обеспечивает централизованный контроль защиты и целостности данных, доступ к данным, их обработку, формирование отчетов на основе базы данных и другие операции и процедуры. СУБД является универсальным программным продуктом общего назначения.



Рисунок 1 – Система управления базой данных

Понятия *база данных* и *система управления базой данных* образуют во взаимосвязи понятие *банк данных* – хранилище реальной совокупности данных, система специальным образом организованных данных (баз данных), программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования. База данных и банк данных имеют архивы, которые в совокупности образуют *фонд данных*.

Организационной основой разработки любой системы служит ее *жизненный цикл*. Жизненный цикл информационной системы включает три основные стадии:

- проектирование;
- программную реализацию;
- эксплуатацию.

1 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

1.1 Основные понятия

Реляционная база данных – это структура для хранения взаимосвязанной информации.

Все данные такой информационной структуры представлены в виде *двумерных таблиц*, каждая из которых содержит некоторое количество строк, называемых *записями*, и один или несколько столбцов, называемых *полями*.

Примечание – В современной теории баз данных, которая базируется на математической теории множеств, таблицы называют *отношениями (сущностями)*, строки – *кортежами*, а столбцы – *атрибутами*. Реляционные системы были предложены в конце 1968 года доктором Э. Ф. Коддом из фирмы IBM, который для строки таблицы использовал термин "кортеж длины n ", или просто "кортеж", ввел ряд понятий в области управления базами данных и дал более четкие определения существующим.

Записи в таблице могут располагаться в произвольном порядке и, по сути, различаются только по содержащимся данным. Поэтому каждая таблица должна иметь один или несколько столбцов с уникальными значениями для идентификации каждой строки.

1.2 Проектирование баз данных

Проектирование баз данных – это многоэтапный процесс, который включает: анализ информационной модели предметной области, выделение информационных объектов данной предметной области, определение их атрибутов, выделение ключевых атрибутов, описание типов данных, определение связей между информационными объектами.

Задачи проектирования БД:

- обеспечение хранения в БД всей необходимой информации;
- обеспечение получения требуемой информации по всем запросам;
- обеспечение целостности данных, правильности их содержания;
- сокращение избыточности и повтора данных;
- недопустимость наличия противоречий в данных;
- исключение потери данных.

Проектирование БД разбивают на два основных этапа:

- *инфологическое* (концептуальное);

- *дatalogическое*, которое, в свою очередь, подразделяется на *логическое* и *физическое* проектирование.

На этапе *инфологического проектирования* выполняется изучение и описание предметной области, построение формализованной модели предметной области. Такая модель строится с использованием стандартных языковых средств, обычно графических, например, ER-диаграмм – диаграмм «Сущность-связь» (от англ. Entity-Relationship, т. е. сущность-связь). В них сущности изображаются в виде прямоугольников или ромбов, а связи между ними – линиями со стрелками, над которыми проставляется степень связи (1 или буква M, заменяющая слово «много»). Такая модель строится без ориентации на какую-либо конкретную СУБД.

Инфологическая модель отображает данные предметной области в виде совокупности информационных объектов (сущностей), их реквизитов (атрибутов) и связей между ними.

Сущность – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в БД. Сущностями могут быть люди, места, товары, самолеты, вкус, цвет и т. д. Каждая сущность имеет несколько экземпляров. Например, сущность ЧЕЛОВЕК может иметь несколько экземпляров – Иванов, Петров, Сидоров и т. д.

Атрибут – поименованная характеристика сущности. Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Примерами атрибутов для сущности ЧЕЛОВЕК являются ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, АДРЕС, ДАТА РОЖДЕНИЯ, НОМЕР ПАСПОРТА и т. д.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Для сущности ЧЕЛОВЕК ключом является атрибут НОМЕР ПАСПОРТА.

Связь – ассоциирование двух или более сущностей. Если бы назначением БД было только хранение отдельных, не связанных между собой данных, то её структура могла бы быть очень простой. Однако одно из основных требований к организации БД – это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи.

Целью логического проектирования является выбор конкретной СУБД и преобразование инфологической (концептуальной) модели в логическую. Для реляционной БД этот этап состоит в разработке структуры таблиц, связей между ними и определении ключевых атрибутов. Структура таблицы определяется составом атрибутов сущности. Ключевой атрибут сущности образует уникальный ключ реляционной таблицы. Такой ключ называется *первичным ключом* таблицы.

Физическое проектирование – реализация даталогической модели средствами конкретной СУБД, а также выбор решений, связанных с физической средой хранения данных: выбор метода управления дисковой памятью, методов доступа к данным, методов сжатия данных и т. д.

1.3 Нормализация данных, нормальные формы

Нормализация представляет собой процесс реорганизации данных путем ликвидации повторяющихся групп и иных противоречий в хранении данных. Таблица находится в данной *нормальной форме*, если она удовлетворяет определенному набору требований. Теоретически существует пять нормальных форм, но на практике обычно используются только первые три. Более того, первые две нормальные формы являются по существу промежуточными шагами для приведения базы данных к третьей нормальной форме.

О нормализации таблиц в базе данных необходимо заботиться на раннем этапе проектирования БД, так как при введенных данных довольно трудно менять структуру базы. Иногда процесс нормализации порождает добавочные таблицы, которые не были включены в первоначальный проект. Если узнать об этом как можно раньше, то не придется напрасно тратить силы на их разработку.

Первая нормальная форма. Для того чтобы таблица считалась нормализованной к первой нормальной форме, каждое из ее полей должно быть неделимым и не должно содержать никаких повторяющихся групп. Поле считается неделимым, если оно содержит только один элемент данных. Например, поле *Адрес*, которое содержит не только название улицы, но также и города, почтовый код, не является неделимым. Чтобы соответствовать первой нормальной форме, такие поля должны быть разбиты на несколько полей (название улицы, номер дома, номер квартиры, город, индекс). *Повторяющаяся группа* – это поле, которое повторяется внутри определения записи с целью хранения нескольких значений для атрибута.

Вторая нормальная форма. Для приведения таблицы ко второй нормальной форме необходимо, чтобы:

- 1) она находилась в первой нормальной форме;
- 2) все не ключевые поля полностью зависели от первичного ключа таблицы или от каждого поля в первичном ключе, если последний состоит из нескольких полей. Это значит, что каждое не ключевое поле должно однозначно определяться первичным ключом и полями, его составляющими.

Третья нормальная форма. Для того чтобы таблица была приведена к третьей нормальной форме, необходимо, чтобы она находилась в первой и второй нормальной формах и чтобы все не ключевые поля полностью зависели от первичного ключа таблицы и не зависели друг от друга.

1.4 Проектирование базы данных «Бронирование билетов»

Существует несколько подходов к проектированию баз данных:

- база данных проектируется на основе предлагаемых полей – реквизитов, описывающих некоторую предметную область;
- на основе общей постановки;
- на основе бумажных документов.

При проектировании на основе бумажных документов основой проектирования являются бланки и другие виды документации, с которыми работают конечные пользователи. Например, при разработке базы данных «Бронирование билетов» за основу можно взять представленный ниже документ «Билет» (рисунок 2), содержащий информацию о направлении следования пассажира, дате, времени, номере поезда, номере вагона и места, стоимости проезда.

БИЛЕТ № _____	
Номер поезда ____ вагон ____ место ____	
Станция отправления _____	
Станция прибытия _____	
Дата отправления _____	Время отправления _____
Дата прибытия _____	Время прибытия _____
Стоимость билета _____	

Рисунок 2 – Пример бланка, на основе которого проектируется база данных

Предметная область – бронирование пассажирами билетов на поезд.
Инфологическая модель представлена на рисунке 3.

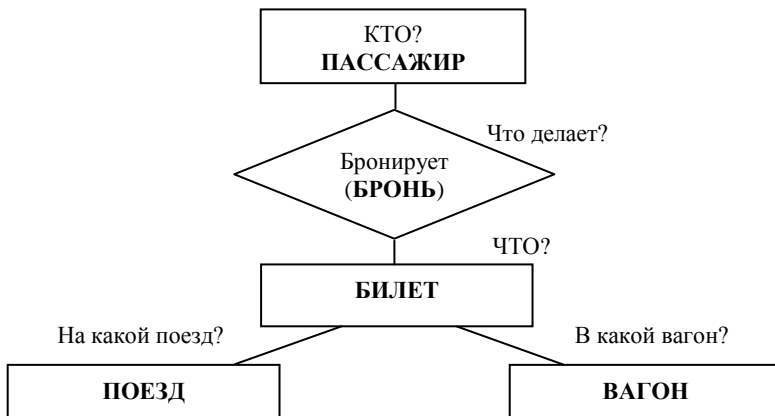


Рисунок 3 – Упрощенная инфологическая модель

На основании инфологической модели можно выделить следующие информационные объекты (сущности):

- ПАССАЖИР;
- БИЛЕТ;
- ПОЕЗД;
- ВАГОН;
- БРОНЬ.

ВНИМАНИЕ! – Сущность БРОНЬ – это сущность-отношение, которая соединяет сущности ПАССАЖИР и БИЛЕТ.

Определим атрибуты, характеризующие каждую сущность (таблица 1).

Таблица 1 – Сущности и их атрибуты

Сущность	Атрибуты
ПАССАЖИР	НомерПаспорта Фамилия Имя Отчество Адрес E-mail
БРОНЬ	НомерЗаказа ДатаБронирования ← НомерПаспорта ← НомерБилета
БИЛЕТ	НомерБилета НомерПоезда ТипВагона НомерВагона Место Цена ← ЧТО?
ПОЕЗД	НомерПоезда ПунктОтправления ПунктНазначения ДатаОтправления ДатаПрибытия ВремяОтправления ВремяПрибытия
ВАГОН	КодТипаВагона КраткоеНазвание ПолноеНазвание КоличествоМест

Определим *ключевые атрибуты* для каждой сущности. Для сущности **ПАССАЖИР** ключом является атрибут **НомерПаспорта** (этот атрибут однозначно определяет пассажира, который бронирует билет), для сущности **БРОНЬ** ключом является атрибут **НомерЗаказа**, для сущности **БИЛЕТ**

ключом является атрибут **НомерБилета**, для сущности **ПОЕЗД** ключом является **НомерПоезда**, для сущности **ВАГОН** ключом является **ТипВагона**. Эти ключевые атрибуты называются *первичным ключом*. По ним устанавливаются связи между сущностями.

Для того чтобы ответить на вопрос «Кто бронирует билет?», к атрибутам сущности **БРОНЬ** необходимо добавить атрибут **НомерПаспорта**, этот атрибут называется *внешним ключом* сущности **БРОНЬ** и по нему устанавливается связь с сущностью **ПАССАЖИР** по его первичному ключу – **НомерПаспорта**. Для того чтобы ответить на вопрос «Что бронирует пассажир?», необходимо к атрибутам сущности **БРОНЬ** добавить атрибут **НомерБилета**, который будет вторым внешним ключом для сущности **БРОНЬ** и по нему будет установлена связь с сущностью **БИЛЕТ**.

Рассмотрим сущность **БИЛЕТ**. Она характеризуется следующими атрибутами: **НомерБилета**, **НомерПоезда**, **ТипВагона**, **НомерВагона**, **Место**, **Цена**. Атрибуты **НомерПоезда** и **ТипВагона** можно назвать внешними ключами сущности **БИЛЕТ**, так как по ним можно установить связь с сущностью **ПОЕЗД** по атрибуту **НомерПоезда** и с сущностью **ВАГОН** по атрибуту **ТипВагона**. Атрибут **НомерПоезда** для сущности **ПОЕЗД** является первичным ключом (однозначно определяет номер поезда), атрибут **ТипВагона** для сущности **ВАГОН** является первичным ключом, поскольку однозначно определяет тип вагона (рисунок 4).

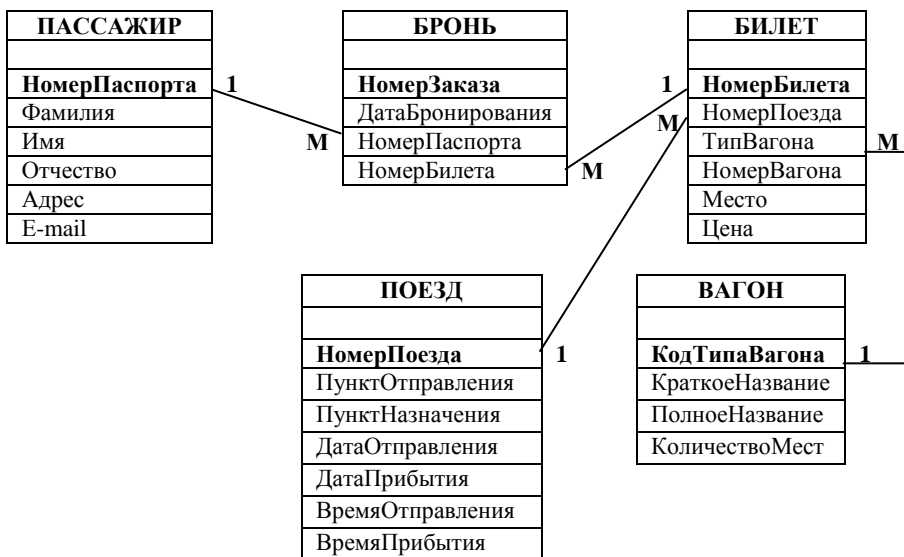


Рисунок 4 – Инфологическая модель базы данных «Бронирование билетов»

Установим связи между сущностями по ключу.

Каждый ПАССАЖИР может бронировать билеты несколько раз или несколько билетов. Поэтому связь ПАССАЖИР – БРОНЬ имеет тип «один ко многим»: ключевое поле НомерПаспорта сущности ПАССАЖИР связывается с неключевым полем сущности БРОНЬ.

Связь БИЛЕТ – БРОНЬ «один ко многим».

Связь ПОЕЗД – БИЛЕТ «один ко многим».

Связь ВАГОН – БИЛЕТ «один ко многим».

В результате получим информационно-логическую модель (см. рисунок 4).

Каждой сущности поставим в соответствие таблицу базы данных, в которой определим ключевые поля, типы данных, хранящихся в базе данных, и свойства полей:

Таблица «Пассажир»			
Поле	Ключевое поле	Тип поля	Свойства поля
НомерПаспорта	Ключ	Текстовое	
Фамилия		Текстовое	
Имя		Текстовое	
Отчество		Текстовое	
Адрес		Текстовое	
E-mail		Текстовое	

Таблица «Бронь»			
Поле	Ключевое поле	Тип поля	Свойства поля
НомерЗаказа	Ключ	Счетчик	
ДатаБронирования		Дата/Время	Маска ввода
НомерПаспорта		Текстовое	Мастер подстановок из таблицы «Пассажир»
НомерБилета		Числовое	Мастер подстановок из таблицы «Билет»

Таблица «Билет»			
Поле	Ключевое поле	Тип поля	Свойства поля
НомерБилета	Ключ	Счетчик	
НомерПоезда		Числовое	Мастер подстановки из таблицы «Поезд»
ТипВагона		Текстовое	Мастер подстановки из таблицы «Вагон»
НомерВагона		Числовое	
Место		Числовое	
Цена		Денежное	

Таблица «Поезд»			
Поле	Ключевое поле	Тип поля	Свойства поля
НомерПоезда	Ключ	Числовое	
ПунктОтправления	Текстовое	Текстовое	
ПунктНазначения	Текстовое	Текстовое	
ДатаОтправления	Дата/Время	Дата/Время	Маска ввода
ДатаПрибытия	Дата/Время	Дата/Время	Маска ввода
ВремяОтправления	Дата/Время	Дата/Время	Маска ввода
ВремяПрибытия	Дата/Время	Дата/Время	Маска ввода


Таблица «Вагон»			
Поле	Ключевое поле	Тип поля	Свойства поля
КодТипаВагона	Ключ	Числовое	
КраткоеНазвание		Текстовое	
ПолноеНазвание		Текстовое	
КоличествоМест		Числовое	

Убедимся, что все таблицы приведены к третьей нормальной форме: все поля таблиц неделимы и неключевые поля зависят только от ключей в каждой таблице.

После проделанной работы по проектированию базы данных можно приступать к созданию и разработке базы, определив инструмент – СУБД *MS Access 2010*.

2 СОЗДАНИЕ БАЗЫ ДАННЫХ

2.1 Запуск MS Access

Для запуска *MS Access 2010* в главном меню *Windows* (кнопка **Пуск**) необходимо выбрать **Все программы / MS Access 2010 / Создать**. Откроется окно (рисунок 5), в правой части которого предусмотрен блок **Новая база данных**. Далее следует указать папку для сохранения базы данных с помощью кнопки **Открыть** – , имя базы данных и нажать кнопку **Создать** –



. Сохранить файл базы данных надо сразу, до создания таблиц. Таким образом, создается пустая база данных, в которую впоследствии можно поместить таблицы, формы, запросы, отчеты и другие объекты.

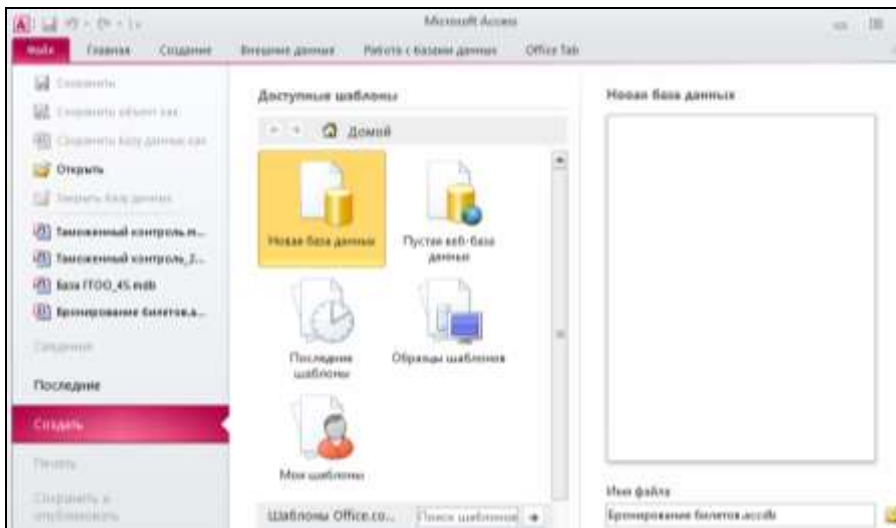


Рисунок 5 – Окно создания базы данных

Если базу данных планируется переносить на другой компьютер, открывать и использовать в более ранних версиях *MS Access*, то перед созданием

файла базы данных выполняют **Файл / Параметры / Общие**, затем в списке *Формат файла по умолчанию для пустой базы данных* выбирают: *Access 2002–2003*.

2.2 Особенности интерфейса MS Access 2010

В верхней части окна расположен аналог главного меню более ранних версий *MS Office* – лента. *Лента* – набор вкладок, которые можно изменять (рисунок 6).

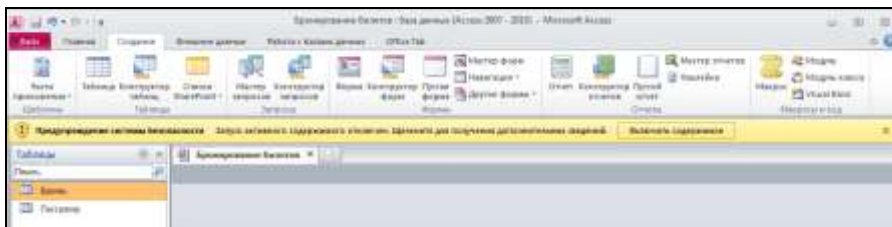


Рисунок 6 – Вкладка *Создание* ленты СУБД *MS Access*

В левой части ленты *MS Access* находится кнопка/вкладка **Файл** красного цвета, содержащая иерархическое меню.

Лента может присутствовать на экране постоянно, либо появляться после щелчка на заголовке вкладки. Смена режима отображения ленты происходит после щелчка на кнопке, расположенной слева от знака вопроса (рисунок 7).

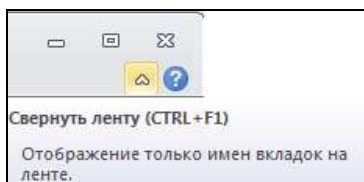



Рисунок 7 – Изменение режимов отображения ленты

Каждая вкладка содержит набор команд, относящихся к одной категории. Некоторые вкладки видны всегда, а некоторые появляются по мере необходимости. Так, вкладки **Файл**, **Главная**, **Создание**, **Внешние данные** и **Работа с базами данных** присутствуют во всех режимах.

Вкладка **Главная** содержит пункты меню, касающиеся правки, форматирования, сортировки, фильтрации, поиска, управления окнами. Вкладка **Создание** позволяет создавать новые объекты базы данных – таблицы, формы, запросы, отчеты, макросы. Вкладка **Внешние данные** включает операции импорта/экспорта. **Работа с базами данных** – это такие инструменты, как схема данных, переход в редактор встроенного языка программирования VBA, анализ быстродействия и т. д.

Слева в окне *MS Access* находится навигационная панель (рисунок 8), предоставляющая доступ к объектам базы данных.

Щелчок по кнопке  выводит на экран меню (рисунок 9).

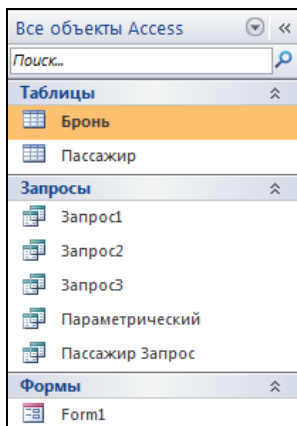


Рисунок 8 –
Навигационная панель

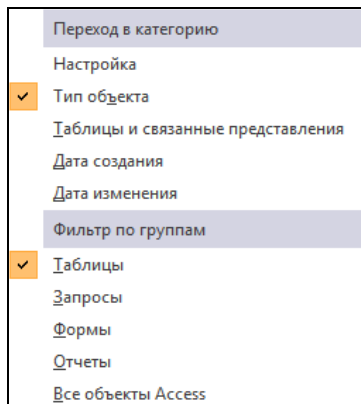


Рисунок 9 – Меню
навигационной панели

Здесь можно выбрать способ отображения объектов. Если выбран пункт **Все объекты Access**, то на навигационной панели будут отражаться все созданные объекты базы данных. Если выбрать пункт **Таблицы**, то будут отображаться только созданные таблицы и т. д.

3 РАЗРАБОТКА БАЗЫ ДАННЫХ

3.1 Таблицы

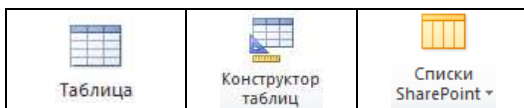
3.1.1 Создание и сохранение таблицы

В *MS Access* вся информация хранится в виде таблиц. Таблицей является совокупность данных, которые относятся к одному информационному объекту. Каждому информационному объекту соответствует отдельная таблица, что позволяет избежать повторения сохраняемых данных. Каждая строка таблицы представляет собой запись, а столбец – поле. Запись содержит набор данных об одном объекте: например, номер паспорта, фамилия, имя, отчество, адрес, e-mail, а поле – однородные данные обо всех объектах (например, номера паспортов всех занесенных в таблицу пассажиров, которые бронируют билеты). Количество полей в записи и их тип определяют в процессе проектирования базы данных.

В *MS Access 2010* создать таблицу можно двумя основными способами:

- путем ввода данных (кнопка **Таблица**, расположенная на вкладке **Создание** ленты, рисунок 10, а);
- в режиме конструирования (кнопка **Конструктор таблиц** вкладки **Создание**, рисунок 10, б).

Команда **Списки SharePoint** используется для создания таблиц на удаленном сервере (рисунок 10, в).



а)

б)

в)

Рисунок 10 – Инструменты *MS Access* для создания таблиц

Для создания таблицы на вкладке **Создание** щелкают по кнопке **Конструктор таблиц**.

Конструктор таблиц содержит три столбца (рисунок 11):

- Имя поля – это заголовок столбца;
- Тип данных – формат содержащихся в столбце данных;

- Описание – комментарий, который можно написать для себя, чтобы не забыть, какие данные будут храниться в этом поле.

Примечание – Имена полей и других объектов базы данных на практике часто записывают без пробелов, латинскими буквами. Знаки препинания и другие спецсимволы недопустимы.

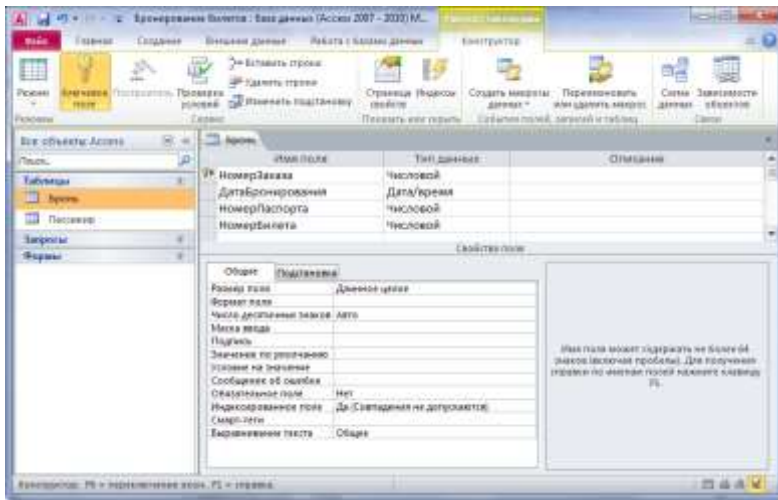


Рисунок 11 – Создание таблицы в режиме конструктора

При закрытии окна конструктора таблицы *MS Access* напомнит, что созданную структуру таблицы надо сохранить. Для сохранения структуры выбирают пункт *Да* и вводят имя таблицы. При выборе пункта *Нет MS Access* не позволит выйти из режима конструктора.

3.1.2 Типы данных

MS Access позволяет выбрать один из следующих типов данных:

- **Текстовый** – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов);
- **Поле Мемо** – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле. Он хранится в другом месте базы данных, а в поле хранится указатель на него, но для пользователя такое разделение заметно не всегда;
- **Числовой** – тип данных для хранения действительных чисел;
- **Дата/время** – тип данных для хранения календарных дат и текущего времени;
- **Денежный** – тип данных для хранения денежных сумм;

- **Счетчик** – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей;

- **Логический** – для хранения логических данных (могут принимать только два значения, например, *Да* или *Нет*);

- **Поле объекта OLE** – специальный тип данных, предназначенный для хранения объектов OLE, например, мультимедийных. Реально, конечно, такие объекты в таблице не хранятся. Как и в случае с полями MEMO, они находятся в другом месте внутренней структуры файла базы данных, а в таблице хранятся только указатели на них (иначе работа с таблицами была бы чрезвычайно замедленной);

- **Гиперссылка** – специальное поле для хранения URL-адресов Web-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне;

- **Мастер подстановок** – это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбрать из раскрывающегося списка.

По умолчанию устанавливается тип данных *Текстовый*. Для изменения необходимо щелкнуть мышью на строке модифицируемого поля в ячейке столбца *Тип данных (Data Type)*, затем выбрать из раскрывающегося списка требуемый тип данных.

3.1.3 Заполнение полей таблицы в окне конструктора

Для каждого поля в окне конструктора последовательно вводятся имя, затем тип данных. По желанию можно ввести произвольное описание поля в столбец **Описание**. Переход к следующей ячейке строки осуществляется с помощью клавиши **Tab** или щелчком мыши на этой ячейке. Переход к заполнению следующего поля осуществляется клавишами **Tab**, **Enter** или щелчком мыши.

3.1.4 Свойства полей

Каждое поле имеет свой набор свойств, от которых зависит, как содержимое этого поля отображается на экране, хранится в базе данных, проверяется или обрабатывается. Некоторые свойства являются общими для всех типов полей, другие относятся исключительно к отдельным типам данных.

Некоторые свойства задаются по умолчанию. Свойства полей не являются обязательными. Их можно настраивать по желанию или вообще не задавать.

Все типы полей, кроме полей счетчика и объекта OLE, имеют следующие основные свойства:

- **Формат поля (Format)** – определяет, как должно отображаться содержимое поля;

- **Подпись** (*Caption*) – определяет текст, который будет отображаться рядом с полем или в отчете;
- **Значение по умолчанию** (*Default Value*) – определяет значение, которое вводится в поле автоматически при добавлении новой записи;
- **Условие на значение** (*Validation Rule*) – определяет область или диапазон значений данных, допустимых в поле;
- **Сообщение об ошибке** (*Validation Text*) – задает текст сообщения, которое появляется в строке состояния при попытке ввести в поле недопустимое значение;
- **Обязательное поле** (*Required*) – содержит значение **Да** (*Yes*) или **Нет** (*No*), в зависимости от того, обязательно ли должно это поле иметь какую-либо запись.

Следующие свойства относятся, в основном, к *текстовым полям*:

- **Размер поля** (*Field Size*) – максимально допустимое количество символов для данного поля;
- **Маска ввода** (*Input Mask*) – обеспечивает ввод данных в указанном формате;
- **Индексированное поле** (*Indexed*) – указывает *MS Access*, должны ли создаваться индексы для этого поля. Индексирование поля ускоряет процесс сортировки, поиска и фильтрации содержимого. Поле, являющееся первичным ключом таблицы, индексировано всегда. Стоит задавать индексацию тех полей, по которым возможно проведение сортировки в формах или отчетах;
- **Сжатие Unicode** (*Unicode Compression*) – представляет собой схему кодировки всех известных символов, для чего вместо одного байта каждый символ представлен двумя. Когда эта функция отключена, *MS Access* при сохранении компрессирует все символы, начинающиеся с байта 0, и распаковывает их при обращении к ним.

Поля числового и денежного типов данных имеют одно дополнительное свойство – это *число десятичных знаков*, которое определяет, какое количество десятичных знаков после запятой будет отображаться на экране и храниться в поле.

Чтобы установить свойство поля, надо выделить строку поля. При этом в левой нижней части окна конструктора откроется список свойств данного поля. Далее следует щелчком мыши установить курсор на строку свойства и выбрать необходимое значение.

3.1.5 Задание маски ввода

При вводе данных, например, в поле ДатаБронирования, было бы удобно, чтобы в этом поле заранее вводилась точка в качестве разделителя – тогда ее не надо будет вводить вручную.

Форматы для ввода данных называются *масками ввода*.

В состав *MS Access* входит инструмент *Мастер ввода масок*. Чтобы задать маску ввода, необходимо:

- 1) установить курсор на поле, для которого надо задать маску ввода;
- 2) щелкнуть по строке свойства **Маска ввода**, при этом появится кнопка с тремя точками **...**;
- 3) после щелчка по кнопке **...** пользователю будет предложено сохранить таблицу, после чего откроется диалоговое окно **Создание масок ввода**;
- 4) выбрать тип маски ввода, например, *краткий формат даты* для поля ДатаБронирования, и щелкнуть по кнопке **Готово**. Окно **Создание масок ввода** закроется, и маска ввода будет внесена в свойства поля.

Маску ввода можно задать и вручную, введя в строку **Маска ввода** шаблон ввода, например, для даты это 00.00.0000.

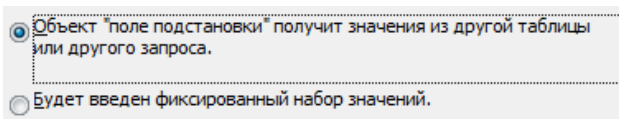
Хранение данных в базе вместе с маской ввода увеличивает размер базы данных, но позволяет облегчить ввод данных.

3.1.6 Задание свойства Подстановка

Поля в таблице могут содержать ограниченное число значений. Поэтому эти значения лучше не вводить, а выбирать из списка.

Чтобы создать для поля список значений, необходимо:

- 1) открыть таблицу в режиме Конструктора;
- 2) щелкнуть мышью в столбце **Тип данных** соответствующего поля;
- 3) выбрать в раскрывающемся списке значение *Мастер подстановок*, при этом будет запущен Мастер;
- 4) в диалоговом окне **Создание подстановки** выбрать опцию

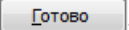


и нажать **Далее >**;

5) на следующем шаге Мастера, если в п. 4 установлена опция «будет введен фиксированный набор значений», надо указать число столбцов (например, 1), ввести значения данного поля и завершить формирование списка нажатием кнопки **Готово**. Если в п. 4 установлена опция «объект "поле подстановки" будет использовать значения из таблицы или запроса», то надо выбрать имя таблицы или запроса, из которого будет выполнена подстановка и нажать **Далее >**;

6) выбрать с помощью кнопок **>** или **>>** поля, которые будут отображаться в таблице подстановки, нажать **Далее >**;

7) при необходимости выполнить сортировку, **Далее >**;

8) если для подстановки в п. 6 было выбрано более одного поля, то опцию «Скрыть ключевой столбец» можно не отменять, в этом случае вместо цифры для выбора будет открыт список значений из других выбранных полей. Нажать кнопку .

В области **Свойства поля** на вкладке **Подстановка** для выбранного поля появятся новые свойства.


После задания подстановки при заполнении таблицы появится возможность не вводить данные вручную, а выбирать из списка.

3.1.7 Ключевое поле

В реляционной базе данных таблицы должны быть связаны друг с другом. Это позволит получить общую информацию в запросе или отчете. Связь между таблицами устанавливается с помощью уникальных полей, т. е. полей, в которых значения не могут повторяться. Например, в таблице **Пассажир** уникальным полем может быть НомерПаспорта.

Поле или комбинация полей, которое однозначно идентифицирует запись в таблице, называется *первичным ключом*. Если в таблице отсутствует поле, которое могло бы служить ключом, можно создать сложный ключ, объединив несколько полей. Для этого надо, удерживая нажатой клавишу **Ctrl**, выделить поля, которые необходимо включить в ключ, щелкнуть правой клавишей мыши и в контекстном меню выбрать пункт **Ключевое поле**.

Если никакая комбинация полей не подходит для ключа, то можно ввести в таблицу дополнительное поле, например, порядковый номер записи, присвоив ему тип данных *Счетчик*. Данное поле будет уникальным.

Чтобы задать ключевое поле, надо установить там курсор, нажать на ленте кнопку  или выбрать из контекстного меню поля пункт **Ключевое поле** (рисунок 12).

Если к моменту выхода из режима конструктора первичный ключ для создаваемой таблицы не будет указан, *MS Access* выдаст запрос о необходимости включения в таблицу ключевого поля.

Для ключевого поля *MS Access* автоматически устанавливает в строке свойства **Обязательное поле** значение *Да (Yes)*, а в строке свойства **Индексированное поле** – значение *Да (совпадение не допускается)*. Если в качестве первичного ключа служит группа из нескольких полей, то все поля являются обязательными, но совпадения внутри каждого из них допускаются.

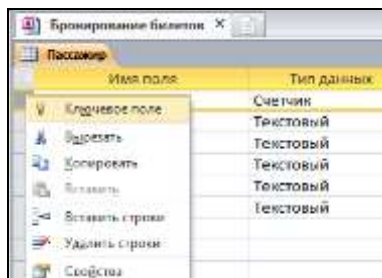


Рисунок 12 – Задание ключевого поля

3.1.8 Межтабличные связи

MS Access позволяет строить реляционные базы данных, отдельные таблицы которых могут быть связаны между собой. Связь между таблицами определяет тип отношения между их полями. Как правило, связывают ключевое поле одной таблицы с соответствующим ему полем другой таблицы, которое называется *полем внешнего ключа*.


При наличии связей между таблицами *MS Access* автоматически будет выбирать связанные данные из таблиц в отчетах, запросах, формах.

Существуют три типа связей:

- один-к-одному – каждая запись таблицы А может быть связана только с одной записью таблицы Б; в этом случае ключевое поле таблицы А связывается с ключевым полем таблицы Б;
- один-ко-многим – одна запись в таблице А может быть связана со многими записями таблицы Б; в этом случае ключевое поле таблицы А связывается с неключевым полем таблицы Б;
- многие-ко-многим – каждая запись в таблице А может быть связана со многими записями в таблице Б, а каждая запись в таблице Б – со многими записями в таблице А.

Нельзя установить связи между полями разного типа: числовое поле можно связать только с числовым полем или полем *Счетчик*, но нельзя связать с текстовым полем.

3.1.9 Создание схемы данных

Чтобы создать связи между таблицами, надо на вкладке **Работа с базами данных** выбрать  *Схема данных*.

Примечание – Все изменения в схему данных можно вносить только при закрытых таблицах. Аналогично таблицы можно изменять, если схема данных закрыта.

В отобразившемся на экране окне **Схема данных** можно определить или просмотреть связи между таблицами базы данных.

Если связи между таблицами устанавливаются для базы данных впервые, то окно **Схема данных** окажется пустым и следует начать с добавления таблиц, для которых планируется создать связи. Для этого в окне **Добавление таблицы** выделяют название таблицы и нажимают кнопку **Добавить**. Таким образом, выбирают все таблицы, между которыми надо установить связи.

Для создания связи необходимо поле первичного ключа главной таблицы перетащить с помощью мыши к соответствующему полю связанной таблицы. На экране появится окно **Изменение связей**. В этом окне главная таблица и первичный ключ изображаются в левом столбце, а подчиненная таблица и внешний ключ – в правом столбце. Далее следует установить флажки «Обеспечение целостности данных», «Каскадное обновление связанных полей» и «Каскадное удаление связанных записей» (рисунок 13).

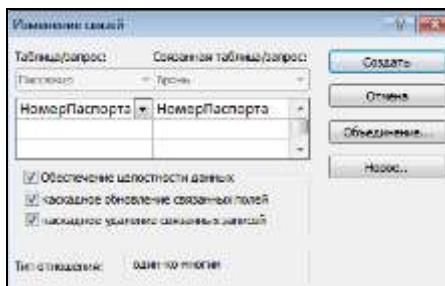


Рисунок 13 – Изменение связей между таблицами

В нижней части окна будет предложен тип связи между таблицами (один-ко-многим). После нажатия на кнопку **Создать** схема данных примет вид, представленный на рисунке 14.

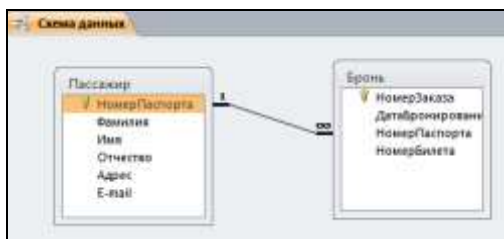


Рисунок 14 – Двухтабличная схема данных

Для удаления связи в схеме данных достаточно щелкнуть по ней правой клавишей мыши и выбрать в отобразившемся контекстном меню команду **Удалить** (рисунок 15).



Рисунок 15 – Удаление связи между таблицами

Примечание – Как правило, MS Access блокирует многие изменения в таблицах, включенных в межтабличные связи. Поэтому прежде чем изменять структуру таблицы, например, переименовывать или добавлять поля, переназначать ключевое поле и т. п., следует удалить из схемы соответствующую связь, а ещё лучше – и модифицируемую таблицу, после чего выполнить сохранение и закрытие схемы данных.

Схема данных для базы «Бронирование билетов» показана на рисунке 16.

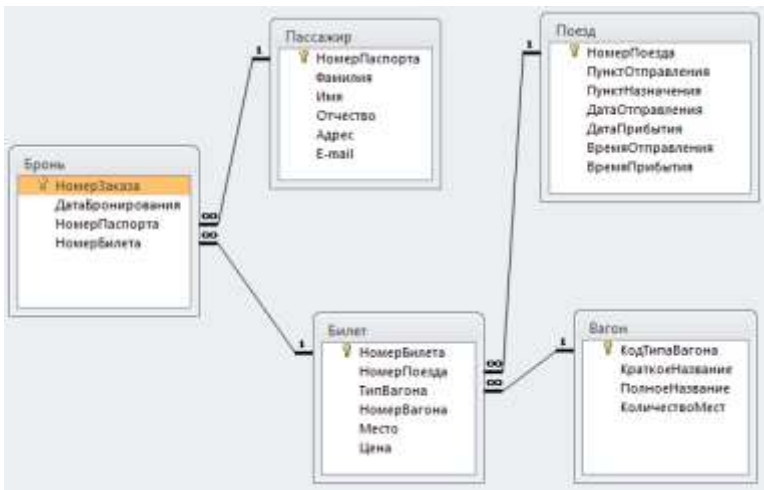


Рисунок 16 – Схема данных базы «Бронирование билетов»

3.1.10 Задание условий целостности

Условие целостности предназначено для того, чтобы обеспечить соответствие записей подчиненной таблицы записям в главной таблице. Оно предотвратит непреднамеренное удаление пользователем тех записей главной таблицы, которые связаны с записями в подчиненной таблице, воспрепятствует добавлению в подчиненную таблицу записей, которым нет соответствия в главной таблице.


Если допускается *каскадное обновление связанных полей*, то при изменении значений первичного ключа в главной таблице происходит автоматическое обновление значений внешнего ключа в подчиненной таблице.

Если разрешено *каскадное удаление связанных записей*, то удаление записи в главной таблице приводит к удалению всех связанных записей в подчиненной таблице.

Для задания условий целостности необходимо:

- на вкладке Работа с базами данных выбрать пункт Схема данных. Откроется окно Схема данных;
- двойным щелчком мыши на существующих линиях связи открыть диалоговое окно **Изменение связей** и в нем установить требуемые флажки. Щелкнуть по кнопке **ОК**;
- закрыть окно **Схема данных**, сохраняя внесенные изменения.

3.1.11 Работа с таблицами

После сохранения и закрытия таблицы можно найти, выбрав с помощью кнопки  на навигационной панели пункт **Таблицы** (рисунок 17).

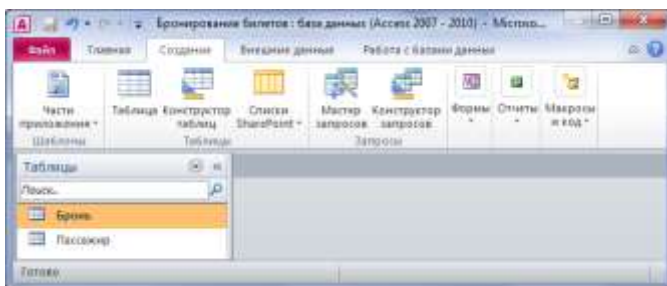



Рисунок 17 – Выбор объектов MS Access

Изменение структуры таблицы выполняется в режиме Конструктора (кнопка  ленты, расположенная на вкладке **Главная**).

Для *ввода данных* таблица открывается двойным щелчком мыши или выбором соответствующей команды из контекстного меню.

Все доступные *команды для работы с открытой таблицей* находятся в контекстном меню, вызванном на вкладке имени таблицы (рисунок 18).



Рисунок 18 – Контекстное меню таблицы

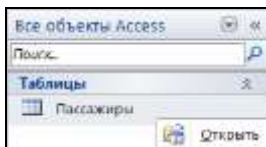
3.1.12 Ввод данных в таблицу

После задания всех полей и определения их свойств выполняется ввод данных в таблицу.



ВНИМАНИЕ! Данные вводятся сначала в главную таблицу, а затем в подчиненную.

Чтобы заполнить таблицу:

- 1) выделяют имя таблицы на навигационной панели



и дважды щелкают левой клавишей мыши. Можно также в контекстном меню выбрать команду **Открыть**;

2) если в данный момент открыто окно Конструктора таблицы, необходимо перейти в режим отображения, нажав кнопку  **Режим**, расположенную на вкладке **Конструктор**. При этом вид кнопки изменится на , а ее повторное нажатие приведет к возврату в окно Конструктора таблицы;

3) информацию вводят в первое поле, щелкнув в нем мышью. Для перехода к следующему полю нажимают клавишу **Tab** или **Enter**. Затем можно продолжить ввод информации с нажатием клавиши **Tab** после заполнения каждой ячейки, пока не будет достигнут конец записи. Для перехода к следующей записи необходимо нажать **Tab** или **Enter** в конце строки.

Чтобы изменить ширину столбца, указатель мыши наводят на разделительную линию между заголовками столбцов и, когда указатель примет вид двунаправленной стрелки, линию перемещают в требуемом направлении.

3.2 Формы

3.2.1 Создание форм

Форма – объект базы данных, предназначенный для ввода данных в таблицу, *кнопочная форма* – для открытия других объектов базы данных.

Перед созданием формы необходимо создать структуру таблицы или запроса, для которых будет создаваться форма.

Формы обеспечивают пользователям возможность ввода данных без необходимости знать, как устроена таблица. Одна форма может содержать данные из многих таблиц, благодаря чему обеспечивается возможность ввода данных в различные таблицы из одного места. Макет формы можно сконструировать таким образом, что форма на экране будет выглядеть точно так же, как и первоисточник: заявление о приеме, квитанция, заполняемая клиентами, и прочие документы, используемые для сбора данных, подлежащих вводу в базу данных.

Существует несколько способов создания форм. Доступ к ним осуществляется с помощью области **Формы** вкладки **Создание** ленты (рисунок 19).

Примечание – Обращайте внимание на всплывающие подсказки, возникающие при наведении на ту или иную кнопку ленты. Помимо названия они могут содержать краткое описание работы инструмента.

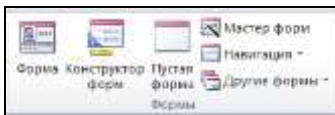


Рисунок 19 – Инструментарий для создания форм в MS Access

Рассмотрим *способы создания форм*.

Форма. Позволяет автоматически создать форму, которая выводит в окне данные по одной записи. При создании такой формы таблица или запрос, для которой создается форма, должна быть открыта или выделена в навигационной панели. В нижней части формы расположены элементы управления для перемещения по записям и добавления новых записей, окно для ввода критериев поиска записей и информация о включенном/выключенном фильтре. Если в момент создания формы в таблице был включен фильтр, то его действие сохраняется в форме. Поля записей в данном виде формы располагаются в столбец.

Если таблица, для которой создается форма, связана с другой таблицей, то в нижнюю часть окна формы помещаются поля из подчиненной таблицы (рисунок 20).

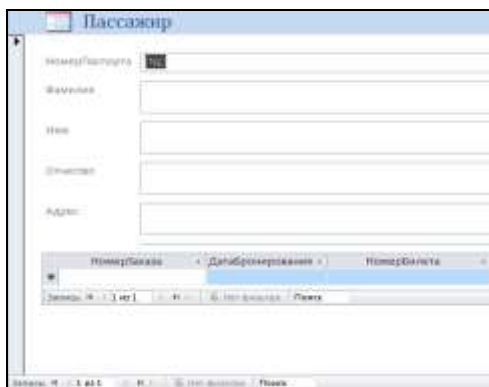


Рисунок 20 – Двухтабличная форма

Конструктор форм. Расположение полей и других элементов управления создаются вручную.

Пустая форма. Создает макет формы, не связанной ни с какими таблицами. Отличается от режима Конструктора только тем, что первоначально открывается в режиме формы, а не Конструктора.

Мастер форм. Наиболее общий способ создания форм.

Навигация. Создание управляющих кнопочных форм. Их назначение – вызывать другие формы.

Другие формы. Здесь можно создать веб-форму (*Несколько элементов*), сводные таблицы и диаграмму, разделенную форму, которая совмещает в одном окне форму и таблицу. С помощью пункта **Таблица** можно создать форму, имеющую такой же вид, как и таблица в режиме Таблицы.

3.2.2 Создание простой формы в режиме конструктора

Для создания новой формы с помощью Конструктора необходимо:

1) выбрать **Конструктор форм** в области **Формы** вкладки **Создание** ленты. На экране появится окно, содержащее макет будущей формы (рисунок 21);

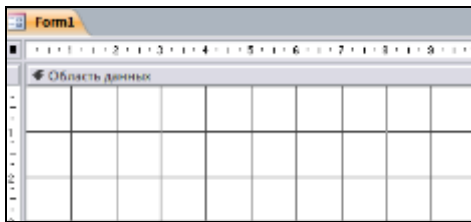



Рисунок 21 – Макет пустой формы

2) установить связь с источником данных, т. е. с таблицей, для которой создается форма, например, с таблицей **Пассажир**. Для этого в области **Сер-**

вис вкладки **Конструктор** выбирают кнопку , которая открывает страницу изменения свойств объекта.

Далее на вкладке **Данные** окна свойств, в строке **Источник записей** из списка имеющихся объектов выбирают таблицу **Пассажир** (рисунок 22);

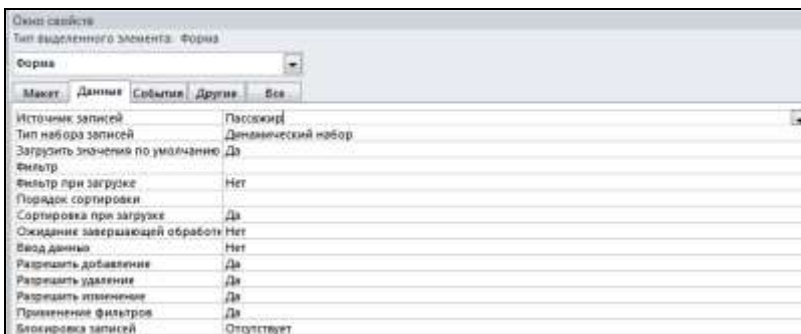



Рисунок 22 – Выбор источника записей для формы

3) вызвать список полей таблицы **Пассажир** (рисунок 23) с помощью кнопки , расположенной в области **Сервис**, и, перетаскивая поля из списка с помощью мыши, расположить их на макете формы.

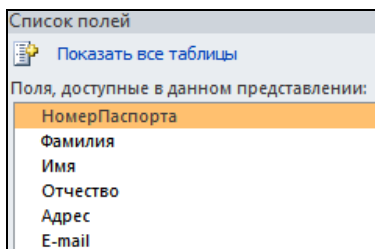
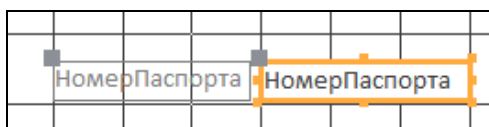


Рисунок 23 – Выбор полей таблицы **Пассажир**

На макете формы отображаются сами поля и надписи к ним, например:



Выделить поле и надпись можно щелчком мыши. Для перемещения поля или надписи по отдельности, курсор мыши наводят непосредственно на черный квадрат в верхнем левом углу требуемого объекта, при этом вид курсора видоизменяется, и только после этого выполняют перемещение;

4) закрыть форму, сохранив ее.

Примечание – Если поле и его надпись необходимо разместить в разных разделах формы, то выделяют отдельно, например, надпись, вырезают её, а затем вставляют в другой раздел (см. п. 3.2.5).

В процессе проектирования формы в режиме Конструктора в любой момент можно перейти в режим Формы, чтобы оценить результат работы. Для этого достаточно в списке **Режим**, расположенном слева на вкладке **Конструктор** ленты, выбрать **Режим формы**. При этом лента перейдет на вкладку **Главная**. Всевозможные режимы для работы с формой показаны на рисунке 24.

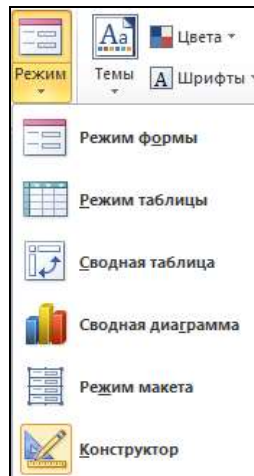






Рисунок 24 – Доступные режимы работы с формой

3.2.3 Мастер форм

Для создания простых и настраиваемых форм в области **Формы** выбирают  **Мастер форм**. Использование Мастера позволяет включать в форму поля из различных таблиц.

Для создания формы с помощью Мастера:

1) в появившемся окне **Создание форм** в списке **Таблицы/запросы** выбирают таблицу, из которой в форму будут включены поля, а в списке **Доступные поля** – поля для создаваемой формы. С помощью кнопок  или  поля переносят в список **Выбранные поля**. В списке **Таблицы/запросы** последовательно можно выбирать таблицы, из которых необходимо включать поля в форму. При включении в форму полей из различных таблиц будет создана многотабличная форма. После завершения отбора нажимают кнопку  **Далее >**;

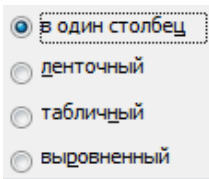

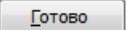


Рисунок 25 – Возможный внешний вид формы

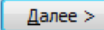

2) выбирают *внешний вид формы*. Для этого активизируют один из переключателей, представленных на рисунке 25, и нажимают  **Далее >**;

3) вводят *имя формы*, выбирают одну из опций: «открыть форму для просмотра и ввода данных» или «изменить макет формы», .

3.2.4 Создание подчиненной формы

Подчиненная форма – это вложенная форма, позволяющая отображать данные из связанной таблицы в тех случаях, когда одной записи из главной таблицы может соответствовать несколько записей подчиненной таблицы (связь *один-ко-многим*). Подчиненная форма внедряется в главную связь. Если главная форма уже достаточно сложна, то можно выбрать отображение подчиненной формы не как вложенной, а как связанной. В этом случае на главной форме автоматически создается кнопка, щелчком по которой открывается подчиненная форма.

Чтобы для таблиц, связанных *один-ко-многим*, создать форму:

- 1) в области **Формы** выбирают **Мастер форм**;
- 2) в списке **Таблицы/запросы** выбирают главную таблицу и копируют требуемые поля в список **Выбранные поля**;
- 3) в списке **Таблицы/запросы** выбирают подчиненную таблицу и копируют требуемые поля в список **Выбранные поля**, нажимают  **Далее >**;
- 4) указывают вид представления данных: хотите ли вы, чтобы форма основывалась на главной или на связанной форме, при этом в качестве исходной можно выбрать главную или подчиненную таблицу, .

Примечание – Опции **Подчиненные формы** и **Связанные формы** определяют отношения между таблицами. Выбрав опцию **Подчиненные формы**, создают главную форму, содержащую подчиненную форму. Если выбрать параметр **Связанные формы**, то в главной форме будет содержаться кнопка, щелчком на которой открывается подчиненная форма;


5) выбирают *внешний вид* подчиненной формы, Далее >;

6) вводят *заголовок главной формы*. Для подчиненной формы следует оставить имя, предложенное программой;

7) устанавливают одну из опций: «открыть форму для просмотра или ввода данных» или «изменить макет формы». При выборе второй опции после нажатия Готово на экране появится проект формы.

3.2.5 Изменение структуры формы в режиме конструктора

Форма в режиме Конструктора имеет три раздела и несколько различных элементов управления (рисунок 26):

- **Заголовок формы** – этот раздел находится в начале первой страницы формы и используется для размещения заголовка формы. Если заголовок и примечание отсутствуют, то их можно добавить из области **Колонтитулы** щелчком по кнопке  **Заголовок**;

- **Примечание формы** – этот раздел расположен в конце последней страницы формы и используется для размещения советов, адресованных пользователю, и другой полезной информации;

- **Область данных** – основная область, в которой располагаются записи данных.

Для *изменения размера области данных* надо переместить указатель мыши в конец области данных, туда, где начинается раздел примечаний. Когда указатель примет вид двунаправленной стрелки, разделительную полосу раздела «Примечание формы» перетаскивают вниз, увеличивая тем самым размер раздела «Область данных». Аналогично изменяется размер раздела заголовка.



Рисунок 26 – Конструктор формы

3.2.6 Элементы управления

Элементами управления в формах называются все объекты, с которыми можно выполнять какие-то действия, в том числе и текстовые поля.

Для *выделения объекта* достаточно щелкнуть на нем мышью – вокруг объекта появятся маркеры выделения.

Для выделения нескольких элементов управления необходимо установить указатель над левым верхним объектом и, нажав клавишу мыши, протянуть по всем объектам выделения. После того как кнопка мыши будет отпущена, все элементы управления, через которые прошла линия, будут выделены.

Если элементы управления, которые необходимо выделить, не располагаются рядом, можно выделить один элемент, а затем выделять другие, удерживая клавишу **Shift**.

Удаление выделенных элементов управления производится клавишей **Delete**. Элементы управления могут быть:

- **присоединенными**, т. е. связанными с полем базовой таблицы или запроса. Такие элементы управления используются для отображения, ввода или обновления значений из полей базы данных;
- **свободными**. Они предназначены для вывода на экран линий, прямоугольников, рисунков или независимых данных.

Элементы управления, которыми может пользоваться разработчик, представлены на **Панели элементов** (рисунок 27), а их назначение – в таблице 2.



Рисунок 27 – Панель элементов


Таблица 2 – Назначение кнопок панели элементов

Кнопка	Назначение	Кнопка	Назначение	Кнопка	Назначение
	Выбрать		Группа переключателей		Прямоугольник
	Поле		Разрыв страницы		Флажок
	Надпись		Поле со списком		Свободная рамка объекта
	Кнопка		Диаграмма		Переключатель
	Вкладка		Линия		Подчиненная форма
	Гиперссылка		Выключатель		Присоединенная рамка объекта
	Набор вкладок		Список		Рисунок


Выбор элемента управления выполняется одним щелчком на его значке в Панели элементов, после чего следующим щелчком на поле формы отмечается место, куда он должен быть поставлен. Вместе с элементом в поле формы вставляется его присоединенная надпись. По умолчанию эта надпись стандартная, например, для переключателей – это *Переключатель1*, *Переключатель2* и т. д. Редактированием свойства элемента управления (доступ к свойствам открывается через контекстное меню) можно дать элементу управления более содержательную подпись.

Изменение размеров элементов управления выполняется так же, как это принято для графических объектов – перетаскиванием маркеров изменения размера, которые расположены по углам и сторонам выделенных элементов управления.

3.2.7 Создание заголовка формы

Для создания заголовков и других свободных надписей в форме служит элемент управления **Надпись** – .

Для создания заголовка необходимо:

- 1) щелкнуть на панели элементов по кнопке **Надпись** – .
- 2) переместить указатель мыши в область заголовка, он примет вид прописной буквы «А» с перекрестием;
- 3) щелкнуть на том месте формы, где необходимо начать ввод текста;
- 4) после ввода текста щелкнуть на любом месте формы, чтобы закрыть элемент управления;
- 5) выделить созданный элемент и использовать панель форматирования для изменения надписи.

3.3 Запросы

Запросы используются для просмотра, изменения и анализа данных различными способами. Запросы также можно использовать в качестве источников записей для форм, отчетов. В *MS Access 2010* запросы можно создать с помощью Конструктора и с помощью Мастера. На вкладке **Создание** ленты расположены соответствующие кнопки (рисунок 28).

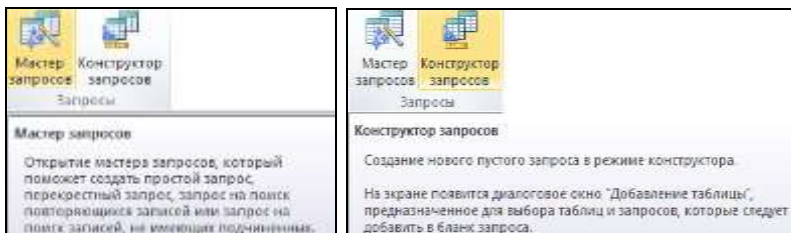


Рисунок 28 – Кнопки создания запросов

3.3.1 Создание запроса в режиме Конструктора

Для создания запроса:

- 1) щелкают по кнопке **Конструктор запросов**;
- 2) в окне **Добавление таблицы** на вкладке **Таблицы** устанавливают курсор на таблицу, поля которой будут использованы в запросе, и нажимают кнопку **Добавить**. Таким образом, можно добавить несколько таблиц в запрос. Но эти таблицы должны быть связаны, иначе запрос будет работать неверно. Закрывают окно **Добавление таблицы**;
- 3) двойным щелчком мыши по полю, которое необходимо включить в запрос, добавляют поля из окна таблицы в бланк запроса (рисунок 29).

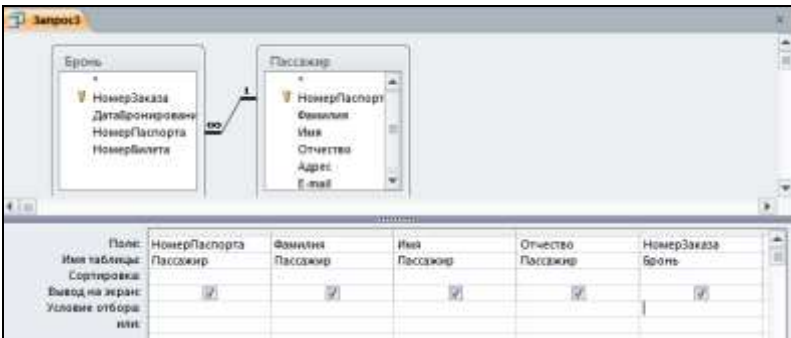





Рисунок 29 – Простой запрос в режиме Конструктора

Далее с помощью кнопки , расположенной на вкладке **Конструктор** ленты, можно запустить запрос на выполнение. При этом кнопка  на вкладке **Конструктор** будет заменена кнопкой  вкладки **Главная**. Этими кнопками выполняют переключение из режима Конструктора в режим просмотра записей, внося изменения в структуру запроса и сразу просматривая результат работы запроса.

При попытке закрыть окно Конструктора на экране появится сообщение о том, что можно сохранить запрос, присвоив ему имя.

3.3.2 Структура бланка запроса

Запрос в режиме Конструктора состоит из двух основных частей, представленных на рисунке 29:

- списка полей объектов БД, на основе которых создается запрос (в данном случае это поля таблиц **Бронь** и **Пассажир**);
- бланка запроса.

Бланк запроса представляет собой таблицу, столбцы которой являются полями запроса, а строки – свойствами полей запроса.

Рассмотрим свойства полей запроса:

- **поле** – в данной строке выводится название поля запроса;
- **имя таблицы** – имя таблицы, в которой находится поле запроса;
- **сортировка** – свойство, предназначенное для выбора типа сортировки записей соответствующего поля запроса. Возможные варианты сортировки: по возрастанию, по убыванию, отсутствует;
- **вывод на экран** – регулирует вывод на экран записей соответствующего поля: – не выводить, – выводить;
- **условие отбора...или** – условия отбора записей соответствующего поля.

3.3.3 Формирование условий отбора в запросах

Условия отбора для различных типов данных формируются при помощи логических операций и операций отношения, представленных в таблице 3.

Таблица 3 – Операции для формирования условий отбора

Логические операции		
And	логическое умножение	необходимо одновременное выполнение заданных условий
Or	логическое сложение	достаточно выполнение одного из заданных условий
Not	логическое отрицание	инверсия условия
Like	совпадение	совпадение по маске
Операции сравнения		
>	больше	
<	меньше	
>=	больше или равно	
<=	меньше или равно	
<>	неравно	

Примеры формирования условий отбора представлены в таблице 4.

Таблица 4 – Формирование условий отбора

Тип данных	Постановка задачи: вывод записей	Пример формирования условия отбора
Текстовый	с фамилией пассажира Иванов	Иванов
Дата/время	с датой бронирования 09.12.2013	09.12.2013
Числовой	с номером заказа 101 или 102	101 Or 102
Дата/время	с датой бронирования 01.12.2013 или 09.12.2013	#01.12.2013# Or #09.12.2013#
Числовой	с номером заказа между 103 и 105	Between 103 And 105

Окончание таблицы 4

Тип данных	Постановка задачи: вывод записей	Пример формирования условия отбора
Числовой	с номером заказа в интервале (102;106)	>102 And <106
Числовой	с номером заказа в интервале [101;105]	>=101 And <=105
Числовой	с номером заказа больше 106	>106
Дата/время	с датой бронирования между 01.04.2013 и 15.04.2013	Between #01.04.2013# And #15.04.2013#
Дата/время	со временем отправления с 6:00 до 10:00	>=#6:00:00# And <=#10:00:00#

Примеры использования масок при формировании условий отбора записей в запросах представлены в таблице 5.

Таблица 5 – Использование масок в условиях отбора

Тип данных	Постановка задачи: вывод записей	Маска
Текстовый	в фамилии пассажира встречается сочетание «ич»	*ич*
Текстовый	фамилия пассажира начинается на «В»	В*

3.3.4 Создание запроса с помощью Мастера



Для создания запросов с помощью Мастера:

1) на вкладке **Создание** ленты в области **Запросы** нажимают на кнопку **Мастер запросов**;

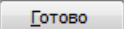
2) в окне **Новый запрос** выбирают **Простой запрос** и нажимают **ОК**;

3) в появившемся диалоговом окне выполняют следующие действия:

- выбирают из списка **Таблицы и Запросы** таблицу, поля которой будут использованы в запросе, например, **Пассажир**;

- переносят при помощи кнопок  или  поля, например, Номер-Паспорта, Фамилия, Имя, Отчество, Адрес, E-mail из списка **Доступные поля** в список **Выбранные поля**;

- нажимают кнопку ;

4) в последнем диалоговом окне задают *имя запроса*, например, **Запрос 1**, и нажимают .

Если необходимо включить в запрос поля из нескольких таблиц, то сначала выбирают поля из одной таблицы, затем из второй и т. д.

3.3.5 Создание запроса с параметром

Запрос с параметром – это запрос, который при выполнении требует ввода с клавиатуры интересующего пользователя значения критерия отбора записей.

Запрос можно создать с помощью Конструктора или с помощью Мастера, а затем в режиме Конструктора для поля, которое предполагается использо-

вать как параметр, например, для поля НомерЗаказа, ввести в ячейку строки **Условие отбора** (рисунок 30) текст-приглашение, заключенное в квадратные скобки – **[Введите Номер заказа]**. Это приглашение будет выводиться при запуске запроса. Текст-приглашение начинается со слова «Введите» и должно отличаться от имени поля, но может включать его.

Квадратные скобки являются для MS Access признаком параметра.

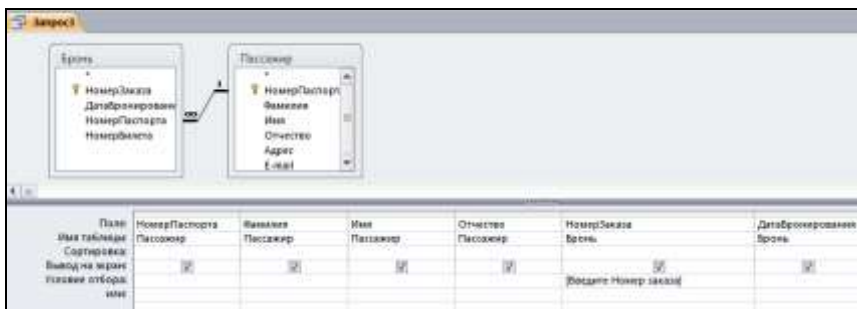



Рисунок 30 – Создание параметрического запроса

Для просмотра результатов нажимают . В появившемся диалоговом окне (рисунок 31) вводят требуемое значение параметра, например, 106. После нажатия кнопки **ОК** отобразится результат запроса – запись с номером заказа, равным 106.

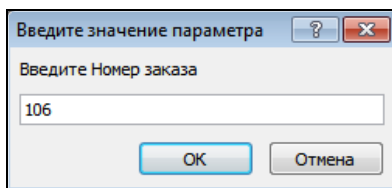


Рисунок 31 – Выполнение параметрического запроса

Для поля, которое выводит даты, можно ввести приглашение следующего вида: **"Введите начальную дату:"** и **"Введите конечную дату:"**, чтобы задать границы диапазона значений. Для этого в ячейке **Условие отбора** соответствующего поля набирают:

Between [Введите начальную дату:] And [Введите конечную дату:]

Чтобы запросить у пользователя один или несколько символов для поиска записей, которые начинаются с этих символов или содержат их, создают запрос с параметром, использующий оператор *Like* и подстановочный символ "звездочка" (*).

Например, выражение

- **Like [Введите первый символ для поиска:] & *** выполняет поиск слов, начинающихся с указанного символа;
- **Like * & [Введите любой символ для поиска:] & *** выполняет поиск слов, которые содержат указанный символ.

3.3.6 Создание запроса на совпадение

Запрос на совпадение отличается от запроса с параметром тем, что для выбора записей, отвечающих заданному критерию, достаточно в поле **Условие отбора** бланка запроса ввести определенный критерий отбора. Например, для выбора записи с номером заказа 106 для поля НомерЗаказа в условии отбора вводят число 106 (рисунок 32).

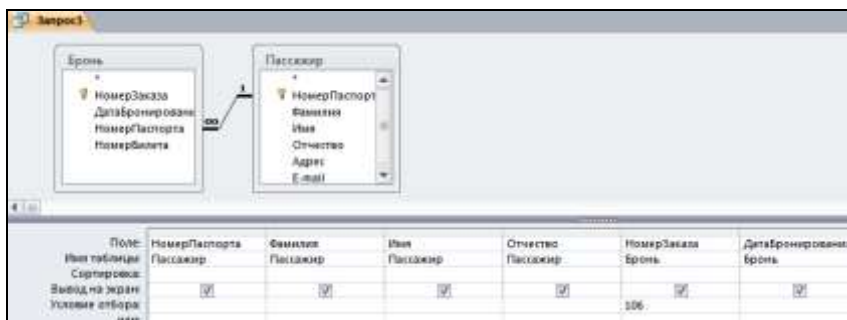



Рисунок 32 – Запрос на совпадение

3.3.7 Создание итоговых запросов

Для того чтобы подсчитать количество заказанных билетов по датам бронирования, необходимо сгруппировать данные по дате бронирования и найти количество заказов, приходящееся на каждую дату.

Для создания этого запроса необходимо выполнить следующие действия:

- 1) на вкладке **Создание** в области **Запросы** щелкнуть по кнопке **Конструктор запросов**;
- 2) в окне **Добавление таблицы** выбрать таблицу **Бронь** и добавить в бланк запроса;
- 3) поместить поля ДатаБронирования и НомерЗаказа в бланк запроса;
- 4) щелкнуть по кнопке  панели инструментов, после чего в нижней части таблицы появится строка **Групповая операция**.

Для каждого столбца таблицы можно выбрать либо группировку, либо одну из следующих операций: Sum (*Сумма*), Avg (*Среднее*), Min (*Минимум*), Max (*Максимум*) Count (*Количество значений*) и т. д.

Окончательно запрос для подсчета количества заказанных билетов по датам бронирования в Конструкторе будет выглядеть, как показано на рисунке 33.

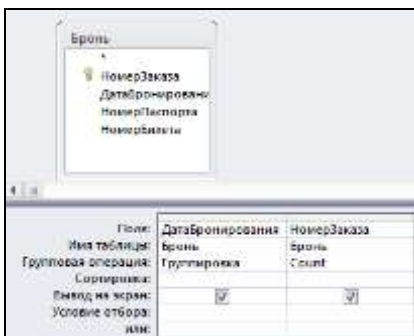


Рисунок 33 – Итоговый запрос

Для создания итоговых запросов можно использовать и Мастер запросов. Предположим, что надо подсчитать, на какую сумму забронировано билетов на каждый поезд. Для создания такого запроса необходимо:

- 1) на вкладке **Создание** в области **Запросы** выбрать **Мастер запросов**;
- 2) в окне **Новый запрос** выбрать **Простой запрос** и нажать **ОК**;
- 3) в появившемся диалоговом окне выполнить следующие действия:
 - выбрать из списка **Таблицы и Запросы** таблицу **Поезд**, поля из которой будут использованы в запросе;
 - перенести при помощи кнопки поле **НомерПоезда** из списка **Доступные поля** в список **Выбранные поля**;
 - выбрать из списка **Таблицы и Запросы** таблицу **Билет**;
 - перенести при помощи кнопки поле **Цена** из списка **Доступные поля** в список **Выбранные поля**;
 - нажать ;
- 4) выбрать опцию **Итоговый**, нажать кнопку ;
- 5) для поля **Цена** указать функцию **Sum** (рисунок 34), нажать **ОК**, затем кнопку ;

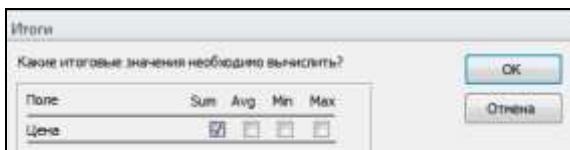
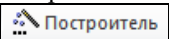


Рисунок 34 – Выбор функции подсчета итогов

- 6) в последнем диалоговом окне задать *имя запроса* и нажать .

3.3.8 Построитель выражений. Создание вычисляемых полей в запросах

Построитель выражений используется для удобства создания выражений и вычисляемых полей в базе данных.

Вызвать окно построителя выражений можно командой **Построить** контекстного меню или кнопкой  панели инструментов.

Окно **Построитель выражений** (рисунок 35) состоит из двух основных частей. В верхнюю часть (окно ввода формулы) вводится выражение. Нижняя часть окна разбита на три области. В левой области выводится список объектов базы данных, в центральной области – список полей объектов базы данных. Справа значения выражений уточняются.

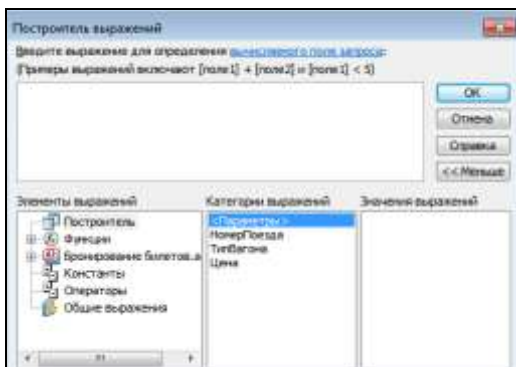


Рисунок 35 – Окно построителя выражений

Пусть требуется создать запрос, переводящий цену билета в евро.

Для этого выполняют следующие действия:

1) создают запрос в Конструкторе, включив в него поля **НомерПоезда**, **ТипВагона**, **Цена** из таблицы **Билет** и сохраняют его;

2) в бланке запроса устанавливают курсор в строке **Поле** нового столбца и из контекстное меню выбирают команду **Построить**. Появится окно построителя выражений, представленное на рисунке 35;

3) вводят выражение (рисунок 36) и нажимают **ОК**. Выбор полей при этом выполняется двойным щелчком по названию поля в центральной области построителя. Набор основных арифметических и логических операций осуществляется при помощи соответствующих кнопок или клавиатуры;



Рисунок 36 – Создание выражения

4) в бланке запроса появится новое поле с именем **Выражение 1**. Эту фразу следует заменить, введя новое имя поля, например, **Цена билета в евро** (рисунок 37);

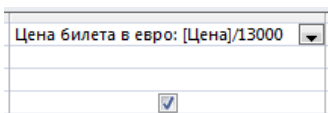


Рисунок 37 – Вычисляемое поле в режиме Конструктора

5) далее можно изменить свойства поля «Цена билета в евро». Для этого из контекстного меню поля выбирают опцию **Свойства**. В окне свойств (рисунок 38) указывают: **Формат поля** – Евро, **Число десятичных знаков** – 1. Сохраняют запрос.

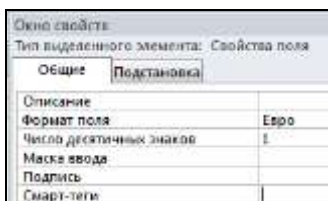


Рисунок 38 – Свойства поля

3.3.9 Использование встроенных функций для вычислений

Для создания вычисляемых полей в *MS Access* можно использовать встроенные функции (рисунок 39). При выборе категории в правой части окна построителя выражений открывается список функций соответствующей категории. В таблице 6 приведены некоторые функции.

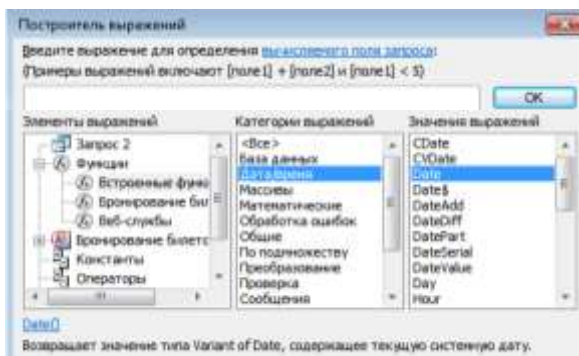


Рисунок 39 – Встроенные функции MS Access

Таблица 6 – Некоторые встроенные функции MS Access

Функции Дата/Время	
Date()	возвращает текущую системную дату
Day(date)	из даты выделяет день месяца
Month(date)	из даты выделяет месяц года
Year(date)	из даты выделяет год
DateDiff("d";date1;date2)	число интервалов времени между указанными датами (<i>d</i> – дней; <i>m</i> – месяцев; уууу – лет и т. д.)
Текстовые функции	
Left(stringexpr;n)	выделяет из строки символов <i>n</i> символов слева
Right(stringexpr;n)	выделяет из строки <i>n</i> символов справа
Len(stringexpr)	возвращает длину строки <i>stringexpr</i>
Функции управления	
IIf(expr>truepart>falsepart)	проверяет условие <i>expr</i> , если оно истинно, то выполняется <i>truepart</i> , иначе – <i>falsepart</i>

Примеры использования встроенных функций представлены в таблице 7.

Таблица 7 – Применение встроенных функций

Пример	Результат
Year(Date())	возвращает 2013
Year([ДатаБронирования])	возвращает год из даты бронирования
DateDiff("d";[ДатаБронирования];Date())	возвращает количество дней, прошедших между текущей датой и датой бронирования билетов
Left([Фамилия];1)	возвращает первую букву фамилии
Left([Имя];1)	возвращает первую букву имени
Left([Отчество];1)	возвращает первую букву отчества
IIf([Вагон]="общий";[Билет]/2;[Билет]/[Цена])	если вагон общий, то цена билета уменьшается в 2 раза, иначе – остается прежней

3.3.10 Создание перекрестного запроса

Перекрестный запрос, предназначенный для представления данных в структурированном, удобном для пользователя виде, можно создать двумя способами:

- в режиме Конструктора (вручную);
- с помощью Мастера, выбрав способ создания запроса *Перекрестный*.

Создание перекрестного запроса в режиме Конструктора.

Пусть требуется создать запрос, в котором данные по цене билета сгруппированы в таблицу, в строках которой расположены номера поездов, а в столбцах – типы вагонов.

Для создания такого перекрестного запроса в *режиме Конструктора*:


1) выполняют последовательно **Создание / Конструктор запросов**;
2) добавляют таблицы, из которых в запрос будут выбираться поля (таблица **Билет** и таблица **Вагон**);

3) последовательно перетаскивают:

- *поле*, значения которого будут заголовками строк перекрестной таблицы (поле **НомерПоезда** из таблицы **Билет**);

- *поле*, значения которого будут заголовками столбцов перекрестной таблицы (поле **ПолноеНазвание** из таблицы **Вагон**);

- *поле*, по которому подводится итог с использованием статистических функций (поле **Цена** из таблицы **Билет**);

4) на вкладке **Конструктор** нажимают кнопку . В результате в бланке запроса появятся две новые строки: **Групповая операция** и **Перекрестная таблица**;

5) в строке **Перекрестная таблица** для полей со значениями в роли заголовков строк выбирается из раскрывающегося списка значение **Заголовки строк** (для поля **НомерПоезда**), а для поля со значениями в роли заголовков столбцов (для поля **ПолноеНазвание**) – **Заголовки столбцов**;

6) строке **Групповая операция** для поля, по которому подводится итог (поле **Цена**), из раскрывающегося списка выбирается необходимая статистическая функция, а в строке **Перекрестная таблица** – **Значение** (рисунок 40);

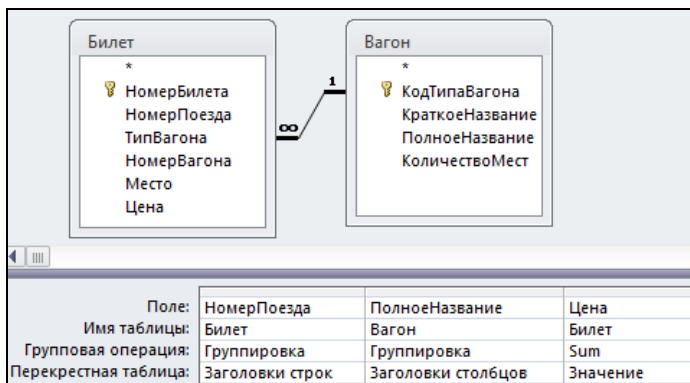


Рисунок 40 – Перекрестный запрос

7) для ввода суммарной цены по **НомеруПоезда** по всем типам вагонов необходимо добавить еще одно поле **Цена**, в строке **Групповая операция** из

списка выбрать функцию Sum, а в строке **Перекрестная таблица – Заголовки строк** (рисунок 41);



Рисунок 41 – Добавление суммарных значений

8) далее в контекстном меню поля изменяют название поля, выбрав в свойствах **Подпись** и введя новое название итогового поля – *Итого*.

9) запрос сохраняют и запускают на выполнение. Результат представлен на рисунке 42.

НомерПоезда	Итого	купейный	межобластной	общий	плацкартный
35	2 010 000,00р.	890 000,00р.		1 120 000,00р.	
109	210 000,00р.	150 000,00р.			60 000,00р.
355	40 000,00р.		40 000,00р.		

Рисунок 42 – Результат выполнения перекрестного запроса

Создание перекрестного запроса с помощью мастера.

Внимание! – Для включения полей из нескольких таблиц сначала создают обычный запрос, содержащий все необходимые поля.

Пусть имеется созданный в Конструкторе запрос с именем **Отбор**, включающий поля **НомерПоезда** и **Цена** из таблицы **Билет** и поле **ПолноеНазвание** из таблицы **Вагон**.


Далее для создания перекрестного запроса *с помощью Мастера*:

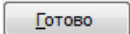
1) выполняют последовательно **Создание / Мастер запросов / Перекрестный запрос** и следуют указаниям Мастера;

2) выбирают опцию **Запросы**, а затем – запрос, который будет использован в качестве основы для создания перекрестного запроса, в нашем случае это запрос **Отбор**,

3) выбирают поле, данные из которого будут использоваться в качестве заголовков строк (поле **НомерПоезда**),

4) выбирают поле, данные из которого будут использоваться в качестве заголовков столбцов (поле **Вагон**),

5) выбирают поле, содержащее числа, над которыми будут проводиться итоговые вычисления (поле **Цена**), и итоговую функцию для этих вычислений (функция Sum). Указывают, требуется ли итоговый результат по каждой строке, ;

б) вводят имя перекрестного запроса и щелкают по кнопке .

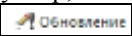
Примечание – При создании перекрестного запроса поле, по которому выбирается функция, должно быть числовым или денежным.

3.3.11 Создание запроса на обновление данных в таблице

Пусть требуется создать запрос, который увеличит значение поля **Цена** билета в таблице **Билет** для указанного номера поезда на 5 %.

Для создания запроса:

- 1) последовательно выполняют **Создание / Конструктор запросов**;
- 2) добавляют таблицу **Билет**, из которой будут в запрос выбираться поля;
- 3) помещают поля НомерПоезда и Цена в бланк запроса и сохраняют запрос, не закрывая Конструктор;

4) нажимают кнопку  на панели инструментов вкладки **Конструктор**. При этом в таблице конструктора запросов появится новая строка **Обновление**;

5) для поля НомерПоезда в строке **Условие отбора** указывают номер поезда, для которого необходимо обновить цену билета, например, 55;

б) для поля Цена в строке **Обновление** вводят формулу с помощью Построителя выражений: $[Цена]*1,05$ или $[Цена] + [Цена]/100*5$ (рисунок 43). Запрос сохраняют;

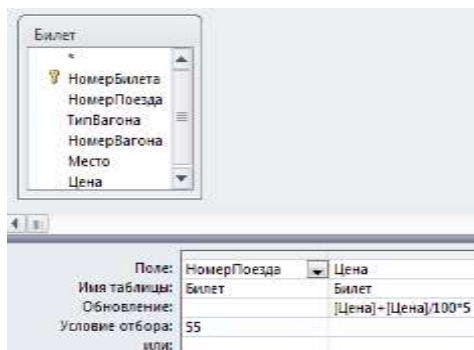




Рисунок 43 – Запрос на обновление

7) с помощью кнопки  панели инструментов можно просмотреть все записи, которые подлежат обновлению;

8) с помощью кнопки  панели инструментов запрос запускают на выполнение. На экране появится окно (рисунок 44), в котором следует подтвердить обновление или отказаться от него.

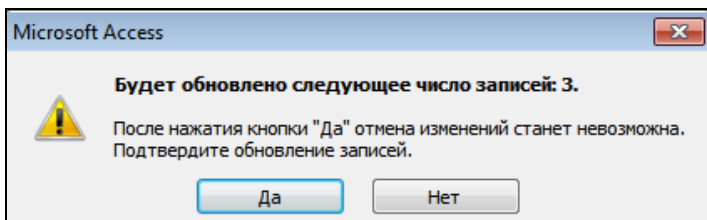


Рисунок 44 – Подтверждение обновления в базе данных

При каждом запуске запроса поле **Цена** в таблице **Билет** будет обновляться для поезда номер 55.

3.3.12 Формирование запроса на удаление записей из таблиц базы данных

При удалении записей из базы данных необходимо **помнить** следующее.

1 Если между таблицами установлена связь с обеспечением целостности данных и с каскадным удалением записей, то создается запрос на удаление записей только из главной таблицы.

2 Если между таблицами установлена связь с обеспечением целостности данных, но без каскадного удаления записей, то сначала создается запрос на удаление записей из подчиненной таблицы, а затем – из главной.

Пусть необходимо создать запрос на удаление записей из таблиц с номером паспорта НВ0100001. Для этого:

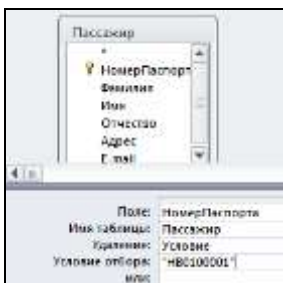
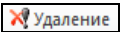


Рисунок 45 – Запрос на удаление записей из главной таблицы

1) проверяют целостность данных и каскадное удаление (см. схему данных);
2) удаляют записи из главной таблицы. Для этого:

- выбирают режим Конструктор;
- добавляют главную таблицу **Пассажир**, перетаскивают поле **НомерПаспорта** в бланк запроса;
- на панели инструментов нажимают кнопку . Появится строка **Удаление**. В строку **Условие отбора** вводят условие НВ0100001 (рисунок 45);
- сохраняют и запускают запрос на выполнение. В результате появится окно,

в котором для удаления записей выбирают *Да*, для отмены – *Нет*.

Примечания

1 Проверьте, удалены ли записи из подчиненной таблицы.

2 Обратите внимание на вид значка запроса.


Пусть надо создать запрос на удаление из таблиц записей с номером паспорта НВ0100003, если *не задано каскадное удаление записей*. Для этого:

1) открывают схему данных и изменяют связь, убрав каскадное удаление;

2) удаляют записи из подчиненной таблицы. Для этого:

- выбирают режим Конструктор;
- добавляют главную таблицу **Пассажир** и подчиненную таблицу **Бронь** (связь *один-ко-многим*);

- перетаскивают * из подчиненной таблицы **Бронь** в бланк запроса и поле **НомерПаспорта** из главной таблицы **Пассажир** в бланк запроса;

- на панели инструментов нажимают кнопку . Появится строка **Удаление**, а ниже – строка **Условие отбора**, в которую вводят условие удаления записей, т. е. НВ0100003;

- сохраняют и запускают запрос на выполнение; при этом будут удалены записи из подчиненной таблицы с номером паспорта НВ0100003;

- в окне Конструктора **удаляют** таблицу **Бронь**.

Если *подчиненных таблиц несколько*, то следует:

1) добавить в бланк запроса вторую подчиненную таблицу, удалить записи из нее, затем удалить подчиненную таблицу в окне Конструктора. И так поступают со всеми подчиненными таблицами;

2) удалить записи из главной таблицы.

3.3.13 Разрешения на выполнение запросов манипулирования данными в MS Access 2010

В *MS Access 2010* выполнение всех запросов на удаление, обновление по умолчанию заблокировано, поэтому при попытке их запустить ничего не происходит. Для того чтобы выполнить такой запрос, следует щелкнуть на кнопке **Включить содержимое** на специальной строке безопасности, расположенной ниже ленты (рисунок 46).

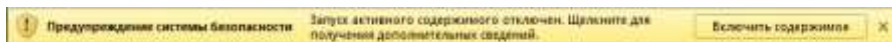


Рисунок 46 – Строка безопасности

Включение активного содержимого также означает разрешение запуска макросов и модулей, созданных в приложении.

3.4 Отчеты

3.4.1 Способы создания отчетов

Отчет – объект базы данных, позволяющий подготовить материалы базы данных к печати, создав выходные документы в соответствии с требованиями конкретного пользователя.

Существует несколько способов создания отчетов. Доступ к ним осуществляется с помощью области **Отчеты** вкладки **Создание** (рисунок 47).

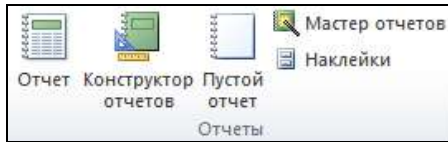


Рисунок 47 – Вкладка для создания новых отчетов

Примечание – При наведении курсора мыши на кнопки вкладки возникает всплывающая подсказка, содержащая полезную информацию.

Кнопка **Отчет** позволяет создать простой отчет на базе таблицы, выделенной на навигационной панели.

Конструктор отчетов создает пустой отчет в режиме конструирования. Отчет состоит из трех основных областей:

- верхний колонтитул;
- нижний колонтитул;
- область данных.

Добавить или удалить заголовок/примечание можно с помощью контекстного меню, в котором выбирают **Колонтитулы страницы** или **Заголовок/примечание отчета**. Заголовок печатается один раз в самом начале отчета, а примечание – в конце отчета. Колонтитулы при печати отображаются на каждой странице.

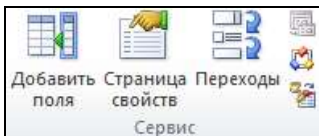
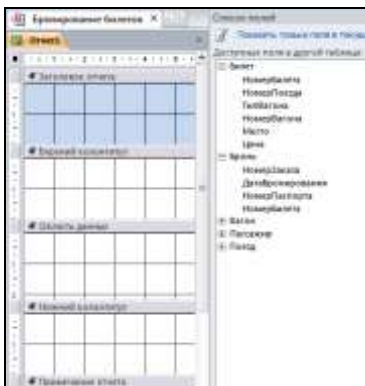


Рисунок 48 – Область Сервис

Для добавления полей необходимо на вкладке **Конструктор** в области **Сервис** выбрать **Добавить поля** (рисунок 48). Появится окно **Список полей**, содержащее все таблицы базы данных, в случае, если ни одна таблица/запрос не была выбрана (рисунок 49, а). Если предварительно в окне свойств на вкладке **Данные** была выбрана таблица/запрос, то в верхней части окна списка полей будут отображаться только поля выбранной таблицы, а в нижней – список полей в связанных таблицах (рисунок 49, б).



а)



б)

Рисунок 49 – Конструктор отчета:

а – источник для отчета предварительно не выбран;

б – источник для отчета выбран

Для изменения свойств на вкладке **Конструктор** в области **Сервис** выбирают **Страница свойств** (см. рисунок 48). В окне свойств можно установить свойства самого отчета, его разделов и элементов управления (рисунок 50). Набор свойств в окне зависит от того, какой объект является активным (т. е. выделен).

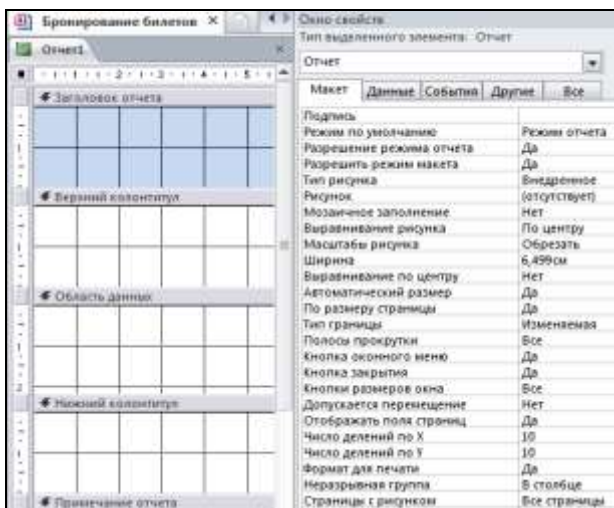




Рисунок 50 – Общий вид окна отчета

При работе с отчетом используются вкладки **Конструктор**, **Упорядочить** и **Формат**. На вкладке Конструктор слева находится список , содержащий четыре позиции:

- **Представление** – просмотр готового отчета на экране монитора, но без возможности распечатать, в макете дополнительно демонстрируется разметка, которую можно изменить;

- **Предварительный просмотр** – можно установить размер бумаги, параметры страницы, альбомную или книжную ориентацию бумаги. Выход из этого режима выполняется щелчком на пиктограмме ;

- **Режим макета** – позволяет изменять внешний вид отчета;
- **Конструктор**.

Из списка полей в отчет можно добавлять поля нескольких связанных таблиц.

3.4.2 Панель элементов


При конструировании отчета используется также **Панель элементов** для создания различных элементов управления (рисунок 51). Вызвать её можно из контекстного меню или на вкладке **Конструктор**, область **Элементы управления**.



Рисунок 51 – Панель элементов

К основным элементам управления относятся:

1) **Надпись** – предназначена для ввода поясняющего текста в любую часть отчета. Для создания надписи выполняют следующие действия:

- выбирают кнопку  на Панели элементов;
- фиксируют курсор в месте вставки и вводят текст;

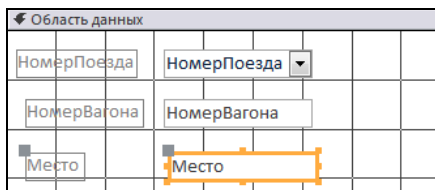




Рисунок 52 – Элемент управления Поле

2) **Поле** – предназначено для вывода значения записей из полей базы данных. Состоит из связанной с ним надписи и области вывода (рисунок 52).

Для создания поля, выводящего данные из таблицы или запроса, достаточно перетащить их из списка полей в область данных отчета;

3) **Переключатели и флажки** – предназначены для выбора одного из двух вариантов: *Истина* или *Ложь*. Используются для логических полей. Для создания переключателя (флажка) выполняют следующие действия:

- выбирают левой клавишей мыши переключатель  или флажок  на панели элементов;
- наводят указатель мыши на требуемое поле в списке полей и перетаскивают его в область данных отчета.

Форматирование отдельных элементов отчета осуществляется кнопками, расположенными на вкладке **Формат**.

3.4.3 Создание отчета при помощи мастера

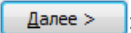
Пусть необходимо создать отчет, в котором выводятся данные из таблицы **Бронь** (поля НомерЗаказа, ДатаБронирования) и из таблицы **Билет** (поля НомерБилета, НомерПоезда, НомерВагона, Место, Цена).

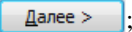
Для создания этого отчета с помощью Мастера:


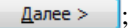
1) в окне базы данных выбирают вкладку **Создание**, затем в области **Отчеты – Мастер отчетов**;

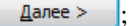
2) в появившемся диалоговом окне выполните следующие действия:

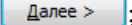
- в списке **Таблицы и запросы** выбирают таблицу **Бронь**, поля из которой будут использованы в отчете;
- переносят поля НомерЗаказа, ДатаБронирования для отчета из списка **Доступные поля** в список **Выбранные поля**;

• повторяют вышеуказанные действия для таблицы **Билет**,  ;

3) в появившемся диалоговом окне выбирают вид представления данных – **Бронь**,  ;

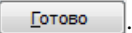
4) выбирают при помощи кнопки  поле для группировки – **ДатаБронирования**,  ;

5) при необходимости выполняют сортировку записей по полю **НомерЗаказа**,  ;

6) нажимают кнопку **Итоги** и устанавливают функцию **Sum** для поля **Цена**, **ОК**,  ;

7) выбирают **Макет отчета**, **Ориентацию страницы**,  ;

8) выбирают стиль оформления отчета,  ;



9) в последнем диалоговом окне задают имя отчета, например, **Отчет 1**, и нажимают кнопку  .

3.4.4 Создание вычисляемых и итоговых полей

Вычисляемые поля предназначены для вывода результатов расчета по одному или нескольким полям базы данных.

Итоговые поля предназначены для вывода итоговых значений по числовым полям базы данных.

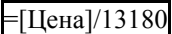
Для создания вычисляемых (итоговых) полей:

- 1) открывают отчет в Конструкторе;
- 2) выбирают кнопку **Поле** –  на Панели элементов;
- 3) фиксируют поле в области данных (при создании вычисляемого поля) или в примечании отчета (при создании итогового поля);
- 4) выбирают команду **Свойства** из контекстном меню элемента поля **Свободный**, где переходят на вкладку **Все**. В строке **Данные** нажимают кнопку вызова построителя выражений ;
- 5) вводят расчетное выражение, например, =avg([Цена]);
- 6) закрывают окно свойств поля;
- 7) вводят текст надписи, связанной с полем.


Примечание – При создании вычисляемых и итоговых полей расчетные формулы можно вводить с клавиатуры. При вводе необходимо соблюдать регистр букв в названиях полей.

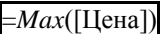
Задание. Отредактируйте **Отчет 1** следующим образом:

- 1) откройте **Отчет 1** в Конструкторе;
- 2) создайте в области данных поле, вычисляющее цену в евро:




- 3) создайте в примечании отчета итоговое поле, вычисляющее минимальное и максимальное значения цены билета по всей базе данных:





- 4) создайте в примечании отчета итоговое поле, вычисляющее среднее значение цены по всей базе данных:




- 5) в нижний колонтитул вставьте нумерацию страниц: область **Колонтитулы, Номер страницы**.
- 6) сохраните отчет с именем **Отчет 2**.

3.5 Кнопочная форма

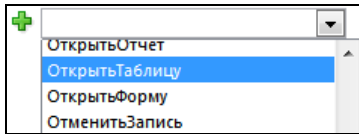
3.5.1 Создание макросов

Макрос – набор из одной или нескольких макрокоманд, каждая из которых выполняет определенное действие.

Для создания макроса:

1) в окне базы данных на вкладке **Создание** в области **Макросы** выбирают кнопку  ;

2) в списке указывают действие, которое должен выполнить макрос:



Откроется окно, представленное на рисунке 53;



Рисунок 53 – Окно настройки макроса

3) в поле **Имя таблицы** выбирают имя таблицы, для открытия которой создается макрос. В списке **Режим** указывают режим открытия таблицы: Таблица, Конструктор, Просмотр, Сводная таблица или Сводная диаграмма. В списке **Режим данных** выбирают, для чего будет открыта таблица: для добавления данных, для изменения данных или только для чтения;

4) закрывают макрос, выполняя сохранение. При сохранении указывают имя макроса, соответствующее выполняемому действию, например, «Открыть таблицу Бронь».

Примечание – Макросы будут использованы при создании кнопок Главной кнопочной формы.

3.5.2 Главная кнопочная форма

Главная кнопочная форма предоставляет интерфейс пользователя для доступа ко всем объектам базы данных. Для создания кнопочной формы:

1) на вкладке **Работа с базами данных** в области **Администратор** выбирают **Диспетчер кнопочных форм**;

2) на вопрос «Не удастся найти кнопочную форму в этой базе данных. Создать кнопочную форму?», нажимают кнопку **Да**.

3) в диалоге **Диспетчер кнопочных форм** представлена **Главная кнопочная форма**, которая открывается по умолчанию (рисунок 54).

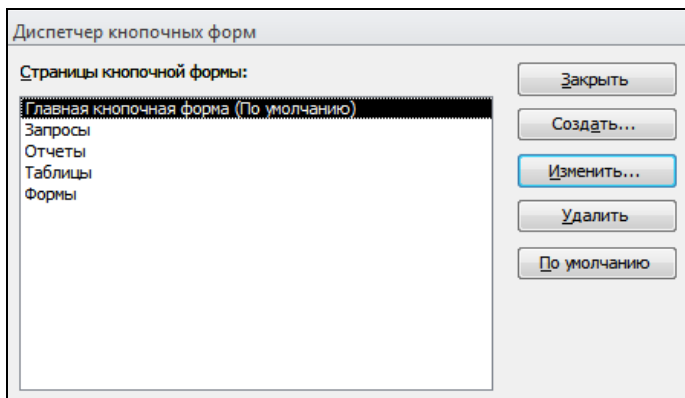


Рисунок 54 – Создание кнопочных форм

В этом окне могут быть сформированы кнопочные формы **Таблицы**, **Формы**, **Запросы**, **Отчеты**, к которым будет переход из главной кнопочной формы. Для этого надо нажать кнопку **Создать**, ввести название кнопочной формы. Например, для создания кнопочной формы **Таблицы** нажимают **Создать**, вводят с клавиатуры **Таблицы** и нажимают **ОК**. Аналогично создаются кнопочные формы **Запросы**, **Формы**, **Отчеты** (см. рисунок 54).

Кнопка **Изменить** вызывает диалог **Изменение страницы кнопочной формы** и позволяет настраивать элементы выделенной кнопочной формы. Например, выделив кнопочную форму **Таблицы** и нажав кнопку **Изменить**, можно создать кнопки для открытия всех таблиц базы данных и кнопку перехода на главную кнопочную форму (рисунок 55). В диалоге **Изменение элемента кнопочной формы** задается *текст* подписи к кнопке, *команда*, выполняемая по нажатию этой кнопки, и *выбирается имя объекта*, который должен быть открыт. Команды могут быть следующие: открыть отчет, открыть форму, выполнить макрос, выйти из приложения. Для открытия таблиц и запросов необходимо выбрать команду **Выполнить макрос**. Макросы для открытия таблиц и запросов создаются заранее.

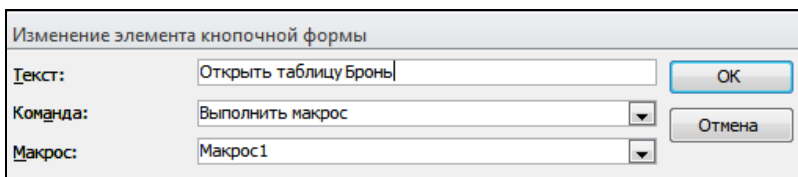


Рисунок 55 – Изменение элемента кнопочной формы

При создании кнопки возврата на главную кнопочную форму в окне задают: **Текст** – На главную, **Команда** – Перейти к кнопочной форме, **Кнопочная форма** – Главная кнопочная форма. Кнопочная форма Таблицы представлена на рисунке 56.

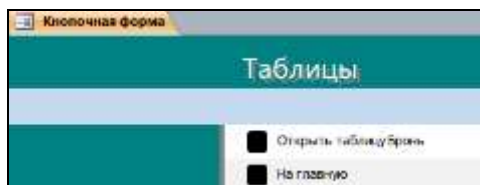


Рисунок 56 – Кнопочная форма Таблицы

Для связывания Главной кнопочной формы с кнопочными формами Таблицы, Запросы и др. и создания кнопок Главной кнопочной формы:

1) выбирают имя Главная кнопочная форма (По умолчанию) и нажимают **Изменить**;

2) нажимают кнопку **Создать** и в окне **Изменение элемента кнопочной формы** в строке **Текст** вводят **Таблицы**, в строке **Команда** выбирают из списка **Перейти к кнопочной форме**, в строке **Кнопочная форма** выбирают имя кнопочной формы, к которой будет выполняться переход – **Таблицы**. Нажимают **ОК**. Указанные действия повторяют несколько раз для создания перехода к кнопочным формам **Формы**, **Запросы**, **Отчеты**.

Для создания кнопки **Выход**:

1) нажимают кнопку **Создать** и в окне **Изменение элемента кнопочной формы** в строке **Текст** вводят текст **Выход**;

2) в строке **Команда** выбирают из списка **Выйти из приложения**.

Примечание – Используйте кнопки **Вверх** и **Вниз** для правильного размещения кнопок.

Таким образом, созданы кнопки на панели Главной кнопочной формы (рисунок 57). Для возврата в окно **Диспетчер кнопочных форм** нажимают кнопку **Заккрыть**.

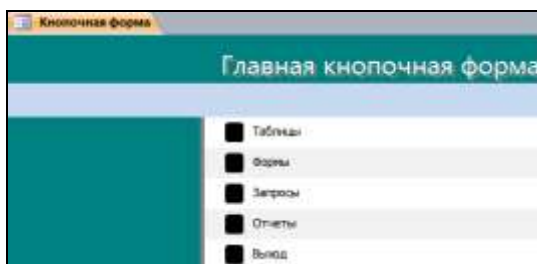


Рисунок 57 – Главная кнопочная форма

ВНИМАНИЕ! Для создания кнопок для открытия таблиц, запросов и т. д. необходимо создать макросы, которые будут открывать данные объекты. В этом случае в окне **Изменение элемента кнопочной формы** в строке **Команда** выбрать в списке команду **Выполнить макрос**, в строке **Макрос** выбрать макрос, выполняющий соответствующее действие.

При создании кнопочной формы с помощью Диспетчера кнопочных форм на вкладке **Таблицы** автоматически создаётся **таблица «Switch board Items»**, которая описывает текст и действия кнопок.

ВНИМАНИЕ! Нельзя удалять таблицу **«Switch board Items»** и **Главную кнопочную форму!**

Чтобы *изменить кнопочную форму* или *добавить кнопки*, надо на вкладке **Работа с базами данных** в области **Администратор** выбрать **Диспетчер кнопочных форм**. Будет открыта главная кнопочная форма, в которой можно выполнить добавление кнопок, изменения.

4 ЗАДАНИЯ НА РАСЧЕТНО-ГРАФИЧЕСКУЮ РАБОТУ

4.1 Общие требования

На основании предложенной оперативной информации (вариант выбрать по номеру в списке группы) спроектировать базу данных (разработать структуры таблиц для справочной и оперативной информации, указать типы полей, свойства полей, выбрать ключевые поля, установить связи). Необходимую справочную информацию выбрать самостоятельно.

На основании спроектированной базы данных выполнить следующие операции:

- создать базу данных с помощью СУБД *MS Access*;
- создать таблицы для хранения данных (для справочной и оперативной информации), создать схему данных;
- создать формы для просмотра и ввода данных в таблицы (ленточные или обычные формы для справочных таблиц и составную форму для оперативных данных);
- заполнить таблицы данными;
- создать запросы по условию задания;
- сформировать отчеты по условию задания;
- при помощи диспетчера кнопочных форм создать главную форму приложения, которая должна открываться при запуске базы данных. Форма должна состоять из страниц:
 - **главная** с кнопками «Переход к таблицам», «Переход к формам», «Переход к отчетам», «Переход к запросам» и «Выход»;
 - «**Таблицы**» с кнопками для открытия всех таблиц и кнопкой «На главную»;
 - «**Формы**» с кнопками вызова всех рабочих форм и кнопкой «На главную»;
 - «**Отчеты**» с кнопками вызова всех отчетов и кнопкой «На главную»;
 - «**Запросы**» с кнопками вызова всех запросов и кнопкой «На главную».

Страницы и кнопки главной кнопочной формы могут быть переименованы в соответствии с выбранной темой.

4.2 Варианты заданий

Вариант 1 – Расчетный листок предприятия

РАСЧЕТНЫЙ ЛИСТОК			
за _____ месяц ____ год			
ФИО _____ Должность _____ Оклад _____			
Таб. номер _____ Отдел _____			
Наименование начисления	Сумма, руб.	Наименование удержания	Сумма, руб.
Начислено:		Удержано:	
Итого к выдаче: _____			

Запросы.

1 Рассчитать сумму премии (50 % от оклада, если стаж работника до 10 лет, и 70 %, если стаж более 10 лет).

2 Рассчитать сумму отчисления в пенсионный фонд (1 %), в профсоюзный фонд (1 %), подоходный налог 12 %, если зарплата (оклад + премия) до 2 млн руб., и 14 %, если более 2 млн руб.

3 Рассчитать сумму начислений, удержаний и итоговых выплат по каждому работнику.

4 Получить данные о сотруднике с минимальной заработной платой.

5 Рассчитать общую сумму заработной платы по всей организации.

Отчеты.

1 Отчет для печати предложенного «Расчетного листка» за определенный месяц, по определенному отделу.

2 Отчет о заработной плате по отделам за определенный месяц.

Вариант 2 – Автовокзал

СПРАВКА ПО НОМЕРУ РЕЙСА					
Номер рейса _____					
Марка автобуса _____					
День недели	Пункт отправления	Время отправления	Пункт назначения	Время прибытия	Цена билета

Запросы.

- 1 Минимальное и максимальное количество мест по марке автобуса.
- 2 Общая стоимость билетов на рейс.
- 3 Сведения о рейсах в воскресенье.
- 4 Сведения о количестве мест по маркам автобуса (марку автобуса ввести с клавиатуры).
- 5 Сведения обо всех рейсах из пункта отправления Гомель.

Отчеты.

- 1 Отчет для печати приведенных в задании оперативных данных с детализацией по номеру рейса и марке автобуса.
- 2 Расписание по дням недели.

Вариант 3 – Услуги БЕЛТЕЛЕКОМ

ИЗВЕЩЕНИЕ

На оплату связи БЕЛТЕЛЕКОМ за месяц _____

Абонент _____ (ФИО)

Адрес _____ телефон _____

Наименование услуг	Объем услуг, мин.	Сумма, руб.
Всего к оплате:		

Запросы.

- 1 Выбрать данные об абонентах по фамилии Иванов.
- 2 Рассчитать сумму к оплате по каждому абоненту.
- 3 Получить информацию об абоненте, который потратил на услуги связи наибольшую сумму денег за определенный месяц.
- 4 Получить информацию о международных или междугородних телефонных соединениях.
- 5 Получить информацию об абоненте с общим объемом услуг до 100 минут.

Отчеты.

- 1 Сформировать отчет для печати вышеприведенного извещения на оплату услуг связи.
- 2 Сформировать отчет о выплатах по каждому абоненту за определенный месяц.

Вариант 4 – Кредитование

СПРАВКА О ПОКУПКАХ В КРЕДИТ

Номер договора	ФИО	Паспорт	Сумма кредита	Срок погашения в месяцах

Запросы.

- 1 Рассчитать ежемесячные выплаты по каждому договору.
- 2 Получить сведения о покупателе, который последний погасит кредит.
- 3 Получить список покупателей, взявших кредит больше 10 млн руб.
- 4 Рассчитать сумму кредита в евро.
- 5 Получить общую сумму выданных кредитов.

Отчеты.

- 1 Сформировать отчет для печати «Справки о покупке в кредит» по конкретному покупателю. Номер паспорта вводится с клавиатуры.
- 2 Сформировать отчет об общей сумме выданных кредитов.

Вариант 5 – Изготовление деталей

НАРЯД НА СДЕЛЬНУЮ РАБОТУ

По цеху _____ на дату _____

Табельный номер	ФИО	Наименование детали	Принято годных, штук	Забраковано, штук	Расценка за штуку, руб.

Запросы.

- 1 Рассчитать, сколько заработал каждый рабочий данного цеха (цех вводится с клавиатуры) за день.
- 2 Рассчитать, сколько заработал каждый рабочий данного цеха (цех вводится с клавиатуры) за месяц.
- 3 Получить список рабочих, допустивших брак, и сумму брака.
- 4 Получить сумму брака по цехам за указанный месяц.
- 5 Вывести список работников-мужчин пенсионного возраста.

Отчеты.

- 1 Сформировать отчет об изготовленных деталях по каждому цеху на определенную дату (дату ввести с клавиатуры).
- 2 Сформировать отчет о работе каждого рабочего на определенную дату (дата вводится с клавиатуры).

Вариант 6 – Товарооборот

СВЕДЕНИЯ О ТОВАРООБОРОТЕ

По магазину № _____ за _____ год

Месяц	План товарооборота (руб.)	Фактический товарооборот (руб.)

Запросы.

- 1 Рассчитать план товарооборота каждого магазина за год.
- 2 Рассчитать фактический товарооборот каждого магазина за год.
- 3 Рассчитать процент выполнения плана товарооборота по месяцам по каждому магазину.
- 4 Рассчитать процент выполнения плана товарооборота по каждому магазину за год.
- 5 Выдать информацию о магазине с наибольшим фактическим товарооборотом за год.

Отчеты.

- 1 Сформировать отчет о товарообороте по магазинам, по годам.
- 2 Сформировать отчет по магазину № 1.

Вариант 7 – Поступление товаров

ВЕДОМОСТЬ ПОСТУПЛЕНИЯ ТОВАРОВ

На склад за _____ месяц

Наименование товара	План поступления, шт.	Поступило, шт.	Реализовано, шт.

ИТОГО на сумму: _____

Запросы.

- 1 Рассчитать сумму всех реализованных товаров за определенный месяц (месяц ввести с клавиатуры).
- 2 Рассчитать остаток по каждому товару на складе за определенный месяц.
- 3 Показать товары, которые имеют план более 1000 шт. по месяцам.
- 4 Показать товар с самым большим планом в определенном месяце, введенном с клавиатуры.
- 5 Выдать список всех товаров на складе, отсортированный по наименованию товара и по месяцам.

Отчеты.

- 1 Сформировать ведомость поступления товаров на склад по месяцам.
- 2 Сформировать отчет об остатках на складе по каждому месяцу.

Вариант 8 – ЖСК

ВЕДОМОСТЬ ПОГАШЕНИЯ ССУДЫ

Членами ЖСК-1

ФИО	Сумма ссуды	Сумма ежемесячного взноса	Процент отчисления на кап. ремонт	Итого сумма ежемесячного взноса

Запросы.

- 1 Рассчитать процент отчисления на капитальный ремонт (3 % от суммы ссуды, если ссуда менее 40 млн руб., и 6 %, если ссуда более 40 млн руб.)
- 2 Рассчитать сумму ежемесячного взноса с учетом отчислений на кап. ремонт по каждому члену ЖСК.
- 3 Определить члена ЖСК, у которого наибольшая сумма ссуды.
- 4 Рассчитать общую сумму ежемесячного взноса кооператива.
- 5 Определить количество членов ЖСК, сумма ссуды которых больше 20 млн руб.

Отчеты.

- 1 Сформировать отчет для печати «Ведомости погашения ссуды».
- 2 Сформировать отчет по ежемесячным выплатам ссуды в бел. руб. и в евро, курс вводить с клавиатуры.

Вариант 9 – Учет рабочего времени

КАРТОЧКА УЧЕТА ОТРАБОТАННОГО ВРЕМЕНИ

в выходные и праздничные дни по цеху _____

Дата работы	ФИО	Табельный номер	Профессия	Время прихода	Время ухода

Запросы.

- 1 Рассчитать ежедневные выплаты работнику за отработанное сверхурочное время.
- 2 Рассчитать выплаты по каждому работнику за месяц.
- 3 Рассчитать выплаты по каждому цеху за месяц.

4 Определить в каждом цехе работника, который отработал наибольшее количество часов за определенный месяц.

5 Вывести список работников, табельный номер которого кратен 5.

Отчеты.

1 Сформировать отчет для печати карточки учета по каждому сотруднику за определенный месяц.

2 Сформировать сводный отчет об отработанном времени по всем цехам за определенный месяц.

Вариант 10 – Банковские вклады

ВЫПИСКА ИЗ ЛИЦЕВОГО СЧЕТА № _____ в бел. руб. (валюте)

За период с _____ по _____

Вкладчик _____ (ФИО)

Дата операции	Наименование операции	Сумма операции

ИТОГО _____

Запросы.

1 Рассчитать сумму начисленных процентов за весь срок, с учетом 26 % годовой ставки.

2 Определить вкладчика с наибольшим вкладом.

3 Вывести все операции на определенную дату, введенную с клавиатуры.

4 Вывести все операции по всем счетам Смирнова Ю. Ю.

5 Получить данные о вкладчике с самой ранней датой вклада.

Отчеты.

1 Сформировать отчет для печати выписки по каждому счету.

2 Сформировать отчет по каждому вкладчику (счет-сумма итогов).

Вариант 11 – Учет рабочего времени

ТАБЕЛЬ УЧЕТА РАБОЧЕГО ВРЕМЕНИ

по цеху _____ за месяц _____ год _____

Табельный номер	ФИО	Отработано часов (О)	Почасовая ставка (С)	Разряд (1–10)

Запросы.

1 Рассчитать сумму премии по каждому сотруднику за месяц при условии, что сумма премии составляет 5 % от заработка, если разряд от 1 до 4, сумма премии 10 %, если разряд 5–8, и сумма премии 20 %, если разряд 9–10.

2 Рассчитать заработную плату по каждому сотруднику в разрезе цехов по формуле (О·С + премия).

3 Вывести список работников с табельным номером, кратным 5.

4 Получить список работников 10-го разряда.

5 Получить данные о сотруднике, табельный номер которого вводится с клавиатуры.

Отчеты.

1 Заработная плата за определенный месяц по цехам с детализацией по сотрудникам.

2 Список сотрудников с детализацией по разрядам.

Вариант 12 – Трудозатраты

ЗАТРАТЫ ТРУДА НА ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ

по операциям по цеху _____

Наименование детали	Код операции	Норма времени на операцию, час.	Расценка по операции, руб.	План, шт.

Запросы.

1 Рассчитать норму времени на изготовление каждой детали (сумма норм всех операций по данной детали).

2 Рассчитать стоимость каждой детали.

3 Рассчитать затраты в рублях на выполнение плана по каждой детали.

4 Выбрать детали, название которых начинается на букву Т.

5 Показать наименее и наиболее трудоемкое изделие.

Отчеты.

1 Сформировать отчет для печати плана выпуска деталей по цеху (Деталь-план-стоимость).

2 Сформировать развернутый отчет по цехам, отражающий затраты труда на изготовление деталей по операциям.

Вариант 13 – Обменный пункт

РАБОТА ОБМЕННОГО ПУНКТА

за _____ месяц _____ год

Дата	Валюта	Покупка (количество)	Продажа (количество)

Запросы.

- 1 Рассчитать сумму покупки/продажи валюты за день.
- 2 Рассчитать сумму покупки/продажи валюты за день по валютам.
- 3 Рассчитать сумму покупки/продажи валюты за месяц.
- 4 Рассчитать сумму покупки/продажи валюты за месяц по валютам.
- 5 Обеспечить выбор данных по дате и по валюте.

Отчеты.

- 1 Отчет о работе обменного пункта за определенный месяц по дням, с итогами за месяц.
- 2 Сумма приобретенной и проданной валюты за определенный период с детализацией по валюте.

Вариант 14 – Оборудование

СПРАВКА

о наличии оборудования в цехе № _____

Шифр оборудования	Наименование оборудования	Балансовая стоимость (Б), руб.	Процент износа (ПР)

Запросы.

- 1 Рассчитать остаточную стоимость оборудования по формуле $B - (B \cdot ПР / 100)$.
- 2 Показать оборудование, процент износа которого более 50 %.
- 3 Показать оборудование с наибольшей балансовой стоимостью.
- 4 Показать сведения об оборудовании по определенному цеху. Номер цеха ввести с клавиатуры.

5 Рассчитать балансовую и остаточную стоимость всего оборудования по цеху и по всем цехам предприятия.

Отчеты.

1 Сформировать отчет для печати справки о наличии оборудования в цехе, включив остаточную стоимость оборудования.

2 Сформировать отчет для вывода на печать балансовой стоимости определенного оборудования (вести с клавиатуры) по указанному цеху, номер цеха ввести с клавиатуры.

Вариант 15 – Экзаменационная ведомость

ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ		
Семестр _____ Группа _____		
Предмет _____ Преподаватель _____		
ФИО студента	Шифр	Оценка

Запросы.

1 Выбрать данные о студентах определенной группы, которые сдали экзамен по дисциплине «История Беларуси».

2 Выбрать данные о сдаче экзаменов определенного студента, группу, фамилию, имя ввести с клавиатуры.

3 Выбрать данные о студентах, которые получили «9» по экзаменам.

4 Определить студента, который набрал наибольшее количество баллов за все экзамены.

5 Вывести списки студентов по группам, отсортированные по фамилии.

Отчеты.

1 Сформировать отчет для получения приведенной в задании ведомости.

2 Сформировать список преподавателей, которые принимали экзамены в группах.

4.3 Требования к оформлению

Оформление итогового отчета по РГР выполняется средствами текстового процессора *MS Word*.

Отчет должен иметь следующую структуру:

- титульный лист;
- вариант индивидуального задания на РГР;
- содержание (составляется автоматически);

- введение (в данном разделе описывается цель РГР);
- Раздел 1. Проектирование базы данных (в виде таблиц *MS Word*, сопровождаемых комментариями);
- Раздел 2. Таблицы базы данных (скриншоты таблиц из *MS Access* в режиме конструктора и в режиме отображения данных);
- Раздел 3. Схема данных;
- Раздел 4. Формы базы данных;
- Раздел 5. Запросы к базе данных;
 - Раздел 5.1. Запросы в конструкторе;
 - Раздел 5.2. Запросы в режиме отображения данных;
- Раздел 6. Отчеты базы данных (скриншот отчета из *MS Access* в режиме конструктора и в режиме отображения данных);
- Раздел 7. Главная кнопочная форма базы данных;
- заключение (в данном разделе подробно описываются все этапы создания базы данных, создание и назначение каждого объекта базы данных и область применения разработанной базы);
- список литературы.

4.4 Требования к форматированию итогового отчета

1 Каждый основной раздел должен начинаться с новой страницы и иметь заголовок, выполненный стилем Заголовок 1, со следующими параметрами: размер шрифта – 14; тип шрифта – Times New Roman; начертание – полужирное; выравнивание текста – по центру, прописными.

2 Подразделы основных разделов формируются стилем Заголовок 2 со следующими параметрами: размер шрифта – 13; тип шрифта – Times New Roman; начертание шрифта – полужирное; выравнивание текста – по центру.

3 Разделы и подразделы должны быть автоматически пронумерованы.

4 Основной текст отчета формируется стилем Главный со следующими параметрами: размер шрифта – 12; тип шрифта – Times New Roman; отступ первой строки – 1 см; выравнивание текста – по ширине; межстрочный интервал – одинарный.

5 В верхний колонтитул документа необходимо по центру ввести ФИО, группу и вариант.

6 Страницы документа должны быть пронумерованы. Номер страницы вводится в нижний колонтитул по правому краю и формируется шрифтом размером 13. Титульный лист не нумеруется, хотя и включается в общую нумерацию страниц документа.

7 В документе должны быть установлены следующие поля: левое – 25 мм; правое – 10 мм; верхнее – 15 мм; нижнее – 15 мм.

Примечание – Требования к форматированию итогового отчета могут быть перепределены согласно варианту.

СПИСОК ИСПОЛЬЗОВАННОЙ И РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1 **Информатика : Базовый курс:** учеб. пособие для студ. высш. техн. учеб. заведений / С. В. Симонович [и др.] ; под общ. ред. С. В. Симоновича. – СПб. : Питер, 2005. – 640 с.

2 **Иоффе, Л. А.** Основы компьютерных информационных технологий : учеб.-метод. пособие / Л. А. Иоффе, Т. Л. Шинкевич, Т. А. Голдобина ; М-во образования Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2010. – 166 с.

3 **Голдобина, Т. А.** Форматирование многостраничных документов в MS Word : учеб.-метод. пособие / Т. А. Голдобина ; М-во образования Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2012. – 48 с.

4 **Лыч, Ю. П.** Технологии организации, хранения и обработки данных / Ю. П. Лыч, Т. Л. Шинкевич. – Гомель : БелГУТ, 2003. – Ч. 1 : Основы создания баз данных : пособие для сам. работы. – 157 с.

5 **Лыч, Ю. П.** Технологии организации, хранения и обработки данных / Ю. П. Лыч. – Гомель : БелГУТ, 2003. – Ч. 2 : Основы программирования в СУБД Access: пособие для сам. работы. – 68 с.

6 **Морозевич, А. Н.** Основы экономической информатики : учеб. пособие / А. Н. Морозевич. – Минск. : БГЭУ, 2006. – 438 с.

Учебное издание

ШИНКЕВИЧ Татьяна Леонидовна
ГОЛДОБИНА Татьяна Александровна

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА
БАЗ ДАННЫХ В MS ACCESS 2010**

Учебно-методическое пособие
по выполнению расчетно-графической работы

Редактор *А. А. Павлюченкова*
Технический редактор *В. Н. Кучерова*

Подписано в печать 20.03.2014 г. Формат 60×84 $\frac{1}{16}$.
Бумага офсетная. Гарнитура Times. Печать на ризографе.
Усл. печ. л. 4,18. Уч.-изд. л. 3,26. Тираж 250 экз.
Зак. № . Изд. № 17.

Издатель и полиграфическое исполнение
Белорусский государственный университет транспорта:
ЛИ № 02330/0552508 от 09.07.2009 г.
ЛП № 02330/0494150 от 03.04.2009 г.
246653, г. Гомель, ул. Кирова, 34.