

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего  
профессионального образования  
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

---

**СОЗДАНИЕ БАЗ ДАННЫХ В СРЕДЕ MS ACCESS**

Методические указания к выполнению лабораторных работ

ГУАП  
Санкт-Петербург  
2014

Составитель: Т.В. Семененко

Рецензент д.т.н., профессор С.И. Зиятдинов

Методические указания содержат описание и порядок выполнения лабораторных работ по дисциплине "Управление данными".

Указания предназначены для студентов, обучающихся по направлению 230400 «Информационные системы и технологии».

Подготовлены кафедрой информационно-сетевых технологий и рекомендованы к изданию редакционно-издательским советом Санкт-Петербургского государственного университета аэрокосмического приборостроения.

Редактор  
Компьютерная верстка

---

Сдано в набор    Подписано к печати    Формат  
Бумага офсетная. Усл.-изд.л.    усл.печ.л. Тираж    экз. Заказ №    .

---

Редакционно-издательский центр ГУАП  
190000, Санкт-Петербург, Б. Морская ул., 67  
Санкт-Петербургский государственный  
университет аэрокосмического  
приборостроения (ГУАП), 2014

## Лабораторная работа №1

### СОЗДАНИЕ БАЗЫ ДАННЫХ И ОБЪЕКТОВ ТИПА “ТАБЛИЦА” В СРЕДЕ MS ACCESS. ОПРЕДЕЛЕНИЕ СВЯЗЕЙ МЕЖДУ ТАБЛИЦАМИ, ЗАДАНИЕ СХЕМЫ БАЗЫ ДАННЫХ

*Цель работы:* изучение процесса создания базы данных и объектов типа “Таблица” с помощью *Конструктора* и с помощью *Мастера*, освоение методов задания свойств полей таблицы и манипуляций с ними, заполнение таблицы данными и редактирование; изучение способов экспорта/импорта данных из других БД, из БД иной архитектуры, из текстовых файлов; освоение способов определения связей между таблицами.

#### 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

В данной работе рассматриваются способы создания объектов базы данных типа «Таблица», свойства этих объектов, построение схемы данных.

##### Таблицы

MS Access поддерживает реляционную модель данных, то есть база данных состоит из множества взаимосвязанных отношений, при этом каждому отношению соответствует отдельная таблица.

Таблицы состоят из строк и столбцов. Столбцы таблицы отражают общие свойства или характеристики объектов реального мира, которые моделируются с использованием данной таблицы. Для каждого свойства пользователь выделяет один столбец. Строки таблицы соответствуют различным экземплярам реальных объектов и называются записям.

Желательно для каждой таблицы определить первичный ключ – столбец или набор столбцов однозначно определяющих строку таблицы.

##### Создание таблицы

Прежде чем создать таблицу, необходимо открыть базу данных, в которой MS Access будет хранить эту таблицу. Затем в окне базы данных следует выбрать закладку “ТАБЛИЦА”, кнопка “Режим” позволяет выбрать необходимый режим создания таблицы: *Таблица* или *Конструктор*.

MS Access предлагает большое количество таблиц-примеров, содержащих данные из различных областей (в меню «Создание» кнопка “Шаблоны таблиц”).

Далее в режиме *Таблица* MS Access предлагает установить первичный ключ.

Для изменения структуры таблицы необходимо воспользоваться режимом *Конструктор*. Поля таблицы расположены друг под другом. Для каждого поля – имя, тип и описание. Поле первичного ключа таблицы, как правило, может иметь тип данных “Счетчик” (AutoNumber). Этот тип данных гарантирует, что во время ввода данных MS Access автоматически нумерует строки таблицы в возрастающей последовательности.

В режиме *Конструктора* можно изменить имеющиеся поля и добавить новые. Если таблица уже содержит данные, то при изменении структуры таблицы они

теряются только в исключительных случаях. MS Access предупреждает об этом соответствующим сообщением.

В именах полей запрещено использовать символы ‘!’ , ‘.’ , ‘[’ , ‘]’. Максимальная длина имени 64 символа. Не рекомендуется использовать в именах полей пробелы, т.к. в дальнейшем могут некорректно выполняться запросы.

В качестве типа данных поля можно использовать следующие стандартные типы: числовой, текстовый, счетчик, поле MEMO, денежный, логический, дата/время, поле объекта OLE и т.д..

Для каждого поля можно ввести произвольный комментарий, относящийся к полю данных. Типичным комментарием является описание назначения поля.

Для каждого поля можно задать дополнительные характеристики. Например, для текстовых полей можно определить размер поля, а для числового определить длину разрядной сетки и количество разрядов после запятой.

При создании таблиц с помощью *Конструктора* MS Access автоматически задает первичный ключ. Чтобы определить в качестве первичного ключа иной набор полей, следует выделить соответствующие поля и нажать пиктограмму “КЛЮЧЕВОЕ ПОЛЕ” на панели инструментов. Во время ввода данных в поля, определенные в качестве первичного ключа, MS Access автоматически следит за тем, чтобы вводились только уникальные значения. MEMO-поля и поля объекта OLE не могут быть первичными ключами.

### Таблица как объект проектирования в MS Access

Каждая таблица имеет свойства, значения которых определяют ее вид и поведение. Ниже приведены основные имеющиеся свойства таблицы (таблица 1).

Таблица 1 – Свойства таблиц

1. AllowZeroLength (Пустые строки)	10. Format (Формат поля)
2. Caption (Подпись поля)	11. Indexed (Индексированное поле)
3. DataType (Тип данных)	12. InputMask (Маска ввода)
4. DecimalPlaces (Число десятичных знаков)	13. Primary (Ключевое поле)
5. DefaultValue (Значение по умолчанию)	14. Required (Обязательное поле)
6. Description (Описание объекта)	15. SortOrder (Порядок сортировки)
7. FieldName (Имя поля)	16. ValidationRule (Условие на значение)
8. FieldSize (Размер поля)	17. ValidationText (Сообщение об ошибке)
9. Unique (Уникальный индекс)	18. IgnoreNulls (Пропуск пустых полей)

**Примечание:** Свойство “DataType” применимо только к таблицам, а свойство “Description” – к таблицам, полям таблиц и индексам. Свойство “FrozenColumns” применимо только к таблицам в режиме таблицы. Остальные свойства применимы к полям таблиц и индексам и могут быть определены в режиме конструктора таблиц.

## Свойство “Тип данных” (DataType)

Свойство “Тип данных” (DataType) для каждого столбца создаваемой таблицы определяет тип данных, хранящихся в поле таблицы. Допустимы следующие основные значения свойства “Тип данных” (таблица 2).

Таблица 2 – Типы данных

Значение	Описание
Текстовый (Text)	Текст, максимальная длина которого равняется 255 символам или определяется значением свойства “Размер поля” (FieldSize) (используется по умолчанию)
Поле МЕМО (Memo)	Текст, максимальная длина которого равняется 64 000 байтам. Поля этого типа не могут быть индексированными
Числовой (Number)	Любые числа. Для получения более подробных сведений смотрите описание свойства “Размер поля”
Дата/время (Date/Time)	Даты и время, относящиеся к годам с 100 по 9999, включительно
Денежный (Currency)	Числа, хранящиеся с точностью до 15 знаков в целой и до 4 знаков в дробной части
Счетчик (AutoNumber)	Число, автоматически увеличиваемое при добавлении в таблицу каждой новой записи. Поля этого типа не подлежат изменению
Логический (Yes/Now)	Логические значения, а также поля, которые могут содержать одно из двух возможных значений. Поля этого типа не могут быть индексированными
Поле объекта OLE (OLE Object)	Объект (например, электронная таблица Microsoft Excel или рисунок Microsoft Draw) созданный другим приложением. Максимальный размер – примерно один гигабайт. Поля этого типа не могут быть индексированными

Это свойство может быть определено только с помощью столбца “Тип данных” в окне конструктора таблиц. Оно отсутствует в бланке свойств; его нельзя определить с помощью макроса или Visual Basic.

Для каждого поля таблицы необходимо обязательно задать тип данных. В каждом поле могут храниться данные только одного типа.

**Внимание!** Изменение типа поля после ввода данных в таблицу может привести к потере данных.

## Свойство “Размер поля” (FieldSize)

Задаёт максимальный размер данных, которые могут быть помещены в данное поле.

Это свойство может быть определено только в бланке свойств поля. Рекомендуется использовать как можно меньшее значение свойства “Размер поля”, поскольку обработка данных меньшего размера выполняется быстрее и требует меньше памяти.

**Внимание!** Преобразование большего значения свойства “Размер поля” к меньшему в таблице, которая уже содержит данные, может привести к потере данных.

Невозможно отменить изменения в макете таблицы после его сохранения в режиме *Конструктора*.

### Свойство “Пустые строки” (AllowZeroLength)

Область применения – поля таблиц (текстовые и МЕМО). Указывает, можно ли вводить в данное поле строки нулевой длины (“”). Допустимы следующие значения свойства “Пустые строки” (таблица 3).

Таблица 3 - Значения свойства “Пустые строки”

Значение	Описание
Да	В данное поле можно вводить строки нулевой длины
Нет	В данное поле нельзя вводить строки нулевой длины (используется по умолчанию)

Для того чтобы ввести в поле строку нулевой длины, введите две кавычки(“”).

Значения свойств “Пустые строки” и “Обязательное поле” (Required) можно использовать для различения несуществующих данных (хранящихся в виде строк нулевой длины) и данных, которые существуют, но неизвестны (хранящихся в виде пустых (Null) значений). Например, база данных может содержать сведения о клиентах, у которых нет факса. Если для поля “Факс” свойство “Пустые строки” имеет значение “Да”, свойство “Обязательное поле” имеет значение “Нет”, а значение свойства “Значение по умолчанию” (DefaultValue) оставлено пустым, то оператор может ввести две кавычки, чтобы указать, что у данного клиента нет факса. Если же поле “Факс” оставлено пустым (номер факса не был введен вообще или был удален), значит, возможно, у данного клиента есть факс, однако, его номер неизвестен.

Свойства “Пустые строки” и “Обязательное поле” не зависят друг от друга. Свойство “Обязательное поле” всего лишь указывает, можно ли вводить в данное поле пустые (Null) значения. Если свойство “Пустые строки” имеет значение “Да”, значит, в данное поле можно вводить строки нулевой длины, независимо от значения свойства “Обязательное поле”.

### Свойство “Подпись” (Caption)

Указывает текст, который отображается в связанной с полем подписи и используется в качестве заголовка столбца в режиме таблицы.

Длина подписи поля, формы или кнопки может достигать 255 символов. Подпись поля можно определить в окне конструктора таблиц или в окне запроса (в списке “Свойства поля”). Обычно подписи используются для отображения полезных сведений. Если подпись связанного поля не определена, то в качестве нее используется имя базового поля.

### Свойство “Индексированное поле” (Indexed)

Определяет индекс по одному полю. Допустимы следующие значения свойства “Индексированное поле” (таблица 4).

Таблица 4 - Значения свойства “Индексированное поле”

Значение	Описание
Нет	Не создает индекс по данному полю (используется по умолчанию)

Да (Допускаются совпадения)	Создает индекс по данному полю
Да (Совпадения не допускаются)	Создает уникальный индекс по данному полю

Это свойство можно определить в бланке свойств в окне конструктора таблиц (в списке “Свойства поля”). Кроме того, можно нажать кнопку “Индексы” на панели инструментов. На экране появится окно “Индексы”. После добавления индекса по одному полю в окно “Индексы” свойство “Индексированное поле” автоматически примет значение “Да”.

Используйте свойство “Индексированное поле” для ускорения выполнения поиска и сортировки записей по одному полю таблицы. Индексированное поле может содержать как уникальные, так и повторяющиеся значения. Пользователь может создать сколько угодно индексов. Индексы создаются при сохранении макета таблицы и автоматически обновляются при вводе и изменении записей. Пользователь может в любое время добавить новые или удалить ненужные индексы в окне конструктора таблиц. Если ключ таблицы состоит из одного поля, то MS Access автоматически устанавливает значение “Да (Совпадения не допускаются)” свойства “Индексированное поле” для данного поля.

Однако индексы замедляют изменение, ввод и удаление данных, поэтому не рекомендуется создавать избыточные индексы.

**Примечание:** МЕМО, логические и OLE-поля не могут быть индексированными. Для создания составных индексов следует использовать окно “Индексы”.

### Свойство “Пропуск пустых полей” (IgnoreNulls)

Указывает, следует ли включать в индекс поля, содержащие пустые (Null) значения.

Выполнение поиска в определенном поле таблицы и сортировку записей по этому полю можно ускорить, создав индекс для этого поля. Если индексированное поле может содержать пустые (Null) значения, то можно установить значение “Да” свойства объекта типа Index “Пропуск пустых полей”, тем самым уменьшив объем индекса.

### Связывание таблиц

Для реляционных СУБД данные разных категорий хранятся в разных таблицах. Это позволяет исключить избыточность информации. Различают несколько типов реляционной связи: “один-к-одному” (1:1) и “один-ко-многим” (1:M)

На практике наиболее часто встречается связь “один-ко-многим” (1:M). При таком типе связи каждой записи главной таблицы могут быть поставлены в соответствие одна или несколько записей, так называемой, подчиненной таблицы.

Для создания связи необходимо определить первичный ключ, как для главной так и для подчиненной таблиц. Определение первичного ключа для подчиненной таблицы хотя и не является обязательным, но значительно увеличивает скорость работы. Для подчиненной таблицы надо определить поле внешнего ключа, тип данных и размер, которые совпадают с полем первичного ключа главной таблицы. Внешние ключи отличаются от первичных тем, что для них допускаются наличие одинаковых значений полей.

MS Access автоматически следит за обеспечением целостности данных. Если редактировать запись в подчиненной таблице, то эта запись может быть сохранена

лишь в том случае, если значение связующего поля присутствует в главной таблице. При редактировании главной таблицы можно удалить запись лишь в том случае, если эта запись не связана с записями подчиненной таблицы.

Таблицы связываются в режиме формирования схемы БД, при этом для установления связи 1 : М необходимо установить мышь на первичный ключ главной таблицы и потом протащить линию связи до подчиненной таблицы. Характеристики связи задаются дополнительно.

## Дополнительные режимы работы с базой данных

При практической эксплуатации базы данных, созданной с использованием MS Access, возникает необходимость общего копирования БД, восстановления БД после сбоев, сжатия БД и наконец репликации – режима получения текущей копии с возможностью эксплуатации ее на переносном или изолированном компьютере с последующей синхронизацией обеих копий БД.

Сжатие БД необходимо, когда очень интенсивно модифицируется БД либо в процессе разработки новых объектов (таблиц, форм, отчетов, запросов, модулей) либо в процессе добавления и удаления данных. В этих случаях все удаляемые объекты и данные удаляются логически, а физически остаются внутри файла \*.accdb, который разрастается значительно и более сложно администрируется. Именно для этого и используется режим сжатия. Операция сжатия проводится над закрытой БД, рекомендуется сначала присвоить сжатой БД новое имя и только после благополучного завершения операции уничтожить старую копию и переименовать сжатую БД. Для проведения операции сжатия необходимо нажать кнопку Microsoft Office и выбрать команду *Управление*, а затем команду *Сжать и восстановить базу данных*. При этом режим *Восстановления базы* используется после нарушения целостности файла БД.

Кроме того, в меню РАБОТА С БАЗАМИ ДАННЫХ находятся разделы, связанные с различными настройками. К ним относятся:

- Разделение БД на 2 взаимосвязанные части, в одной из которых находятся все таблицы, а в другой объекты их обработки: формы, запросы, отчеты, модули. Такое разделение обеспечивает наиболее эффективный способ эксплуатации системы, так как модификация объектов обработки может производиться отдельно и оба файла связываются только с использованием специального режима надстройки *Диспетчера связанных таблиц*.
- *Диспетчер кнопочных форм*, который обеспечивает быструю и гибкую разработку внешнего интерфейса системы, определяющего перечень основных режимов работы и их иерархию.
- *SQL Server*, позволяющий переместить части или всю базу данных в базу данных Microsoft SQL Server.
- Шифрование базы данных с использованием пароля.

Подсоединение таблиц к базе данных возможно осуществить через команды меню ВНЕШНИЕ ДАННЫЕ. При этом можно импортировать таблицы целиком из других баз данных, созданных в MS Access и даже из других баз данных в иных форматах, однако при импорте таблица преобразуется и копируется целиком в текущую БД.



## 2. ЗАДАНИЕ ПО РАБОТЕ

Создать базу данных в MS Access (определить, сколько и какие таблицы должна содержать БД). **Индивидуальные задания находятся в разделе «Варианты заданий для лабораторных работ».** При создании таблиц использовать метод построения таблиц с помощью режимов *Таблица* и *Конструктор*. При создании таблиц учесть все ограничения, которые заданы в постановке задачи. Предусмотреть необходимые связи между таблицами, определить все поля данных (атрибуты или столбцы) таблицы. Проверить работу всех введенных ограничений, для чего ввести данные во все созданные таблицы в количестве 2 – 3х строк. Задать схему БД. Установить требуемые связи и создать требуемую схему БД. Применить необходимые методы поддержки целостности при описании связей.

## 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Создать на диске X в директории (STUDENT – N группы) свой каталог.
2. Создать в своем каталоге новую базу данных, присвоив ей новое имя, отличное от стандартно предлагаемых – db1, db2 .
3. Создать с помощью режимов *Таблица* и *Конструктор* таблицы из схемы БД. При создании учесть ограничения, определенные ранее.
4. Проверить работу всех введенных ограничений, для чего ввести данные во все созданные таблицы в количестве 2 – 3х строк.
5. Установить требуемые связи между таблицами и создать требуемую схему БД.

## 4. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Название и цель выполнения работы.
3. Описание своей предметной области и обоснование принятых решений по таблицам: их перечень, свойства столбцов таблиц и свойства самих таблиц, типы данных, ограничения.

### Контрольные вопросы для самостоятельной проработки

1. Какие файлы будут соответствовать созданной Вами базе данных, их назначение?
2. Что означает *Сжать базу*, где эта команда находится, для чего ее применяют, как выполнение этой команды отразится на информации, которая содержится в системе, что происходит с полями типа “Счетчик” при использовании данной команды?
3. Что означает команда *Восстановить*, где она находится, когда требуется выполнить данную команду, что происходит с базой при выполнении данной команды?
4. Что означает команда *Зашифровать*, где она находится, что дает эта команда, как можно ее применять?
5. Что такое объект типа “Таблица”, как его можно создать, как задать свойства объекта “Таблица”, отличные от свойств его отдельных полей?

6. Что означает команда экспортирования объекта типа “Таблица”, как она выполняется, какие и откуда таблицы можно экспортировать?
7. Что означает команда импортирования объекта типа “Таблица”, как она выполняется, какие и откуда таблицы можно импортировать?
8. Какие свойства необходимо определять при описании полей данных некоторой таблицы?
9. Чем отличается свойство “Подпись” для элемента данных в таблице от графы “Описание” в конструкторе таблиц, где и как применяются эти параметры?
10. Как влияют друг на друга свойства “Формат” и “Маска ввода”?
11. Чем отличается свойство “Пустые строки” для символьных и числовых данных?
12. Что такое “Ключ” для объекта типа “Таблица” и как его можно задать, если он состоит из одного или нескольких полей таблицы? Может ли быть задано для таблицы несколько ключей?
13. Что такое связи между таблицами, какого типа связи между таблицами поддерживает MS Access?
14. Как можно задать связи между таблицами и где эти связи будут зафиксированы? Какие изменения можно внести в связи между таблицами?
15. Как можно удалить таблицу из БД в MS Access, всегда ли это допустимо, если нет, то какие таблицы не могут быть удалены и почему?
16. Что такое основная таблица и подчиненная таблица, как они связаны друг с другом?
17. Могут ли быть у одной основной таблицы несколько подчиненных? Если могут – приведите примеры, если нет, то почему?
18. Могут ли быть у одной подчиненной таблицы несколько основных, с ней связанных, если да – приведите примеры, если нет, то аргументируйте свой ответ?
19. Что такое поддержка каскадного удаления и каскадного обновления при описании связей?

## **Лабораторная работа №2**

### **СОЗДАНИЕ ЗАПРОСОВ В СРЕДЕ MS ACCESS**

*Цель работы:* изучение процесса проектирования запроса с помощью стандартной панели инструментов окна запроса в режиме *Конструктора*.

#### **1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

В данной работе рассматриваются способы создания объектов базы данных типа «Запрос» и типы этих объектов.

#### **Запросы**

Запросы обеспечивают быстрый и эффективный доступ к данным, хранящимся в таблице.

При создании таблицы с помощью MS Access можно заметить, что программа не разрешает использовать вычисляемые поля. Не существует также возможности выполнить сортировку не по первичному ключу.

Благодаря запросам можно выполнить сортировку или вычислить выражения. Например, свести вместе данные из связанных таблиц.

Запрос может быть сформулирован в среде ACCESS на двух языках

- **QBE** (Query By Example) - язык запросов по примеру (образцу), предназначенный для пользователей - не программистов.
- **SQL** (Structured Query Language) - структурированный язык запросов, используемый пользователями - программистами в программах приложений на VBA (Visual Basic for Application). Язык SQL имеет большие возможности по сравнению с языком запросов QBE. Это означает, что все, что можно сформулировать на языке QBE, можно сформулировать и на SQL, но не наоборот. Например, язык QBE не позволяет создавать и редактировать структуру таблицы и пр.

В процессе создания запроса на одном языке запросов ACCESS автоматически формулирует запрос и на другом (если это возможно).

Исполняя запрос, получаем ответ в виде так называемого динамического набора данных, который не сохраняется при сохранении запроса, однако он может быть использован как источник данных (наравне с таблицами) для построения другого запроса. Динамический набор данных можно сохранить, например, в таблице, если выполнить запрос на создание таблицы или запрос на добавление данных, а также использовать в формах и отчетах.

Работа с запросом возможна в следующих режимах:

- *Режим конструктора* - используется при создании и редактировании запроса на языке QBE в специальном окне (бланке) запроса.
- *Режим SQL* - позволяет писать запросы на языке SQL, используя окно встроенного редактора текстов.
- *Режим таблицы* - позволяет просматривать в виде таблицы динамический набор данных, который является ответом на запрос.

Переключение из одного режима в другой осуществляется с помощью кнопки «Режим».

### **Типы запросов**

- *на выборку* (отбор) данных (простой запрос): SQL- оператор SELECT ... WHERE
- *с параметром* (диалоговая форма запроса): PARAMETERS
- *на создание таблицы*: SELECT ... INTO
- *на изменение данных*:
  - ◆ на добавление записей: INSERT ... INTO
  - ◆ на удаление записей: DELETE
  - ◆ на обновление (корректировку) полей записи: UPDATE
- *перекрестный* (ответ имеет вид перекрестной таблицы): TRANSFORM
- *подчиненные* запросы: SELECT, вложенный в инструкции SQL: SELECT, SELECT ... INTO, INSERT ... INTO, DELETE, UPDATE или в другой подчиненный запрос
- *управляющий* запрос (**формулируется только на SQL**)
  - ◆ создание структуры таблицы: CREATE TABLE
  - ◆ модификация структуры таблицы: ALTER TABLE
  - ◆ удаление таблицы: DROP TABLE

♦ создание/удаление индекса: CREATE/DROP INDEX

- *запрос-объединение*, являющийся объединением запросов (**только на SQL**): UNION
- *запрос к серверу* (**только на SQL**).

При создании запроса любого типа рекомендуется придерживаться следующей процедуры:

- сформировать, если возможно, запрос на **выборку** данных на языке QBE (в режиме конструктора или с использованием мастера запросов),
- исполнить запрос и оценить правильность полученного динамического набора данных,
- трансформировать запрос в нужный тип запроса (например, на создание таблицы) выбором соответствующего типа запроса на вкладке КОНСТРУКТОР,
- отредактировать, если необходимо, текст запроса на языке SQL (в режиме SQL).
- сохранить запрос, если необходимо.

Запрос может быть однотабличным, ответ на который может быть получен из одной таблицы или запроса, и многотабличным.

## Создание запроса

Для создания нового запроса необходимо на вкладке СОЗДАНИЕ в группе *Другие* щелкнуть *Мастер запросов*. В диалоговом окне *Новый запрос* выбрать вариант создания запроса:

- ♦ *Простой запрос* позволяет создать с помощью мастера запрос на выборку (отбор) данных.
- ♦ *Перекрестный запрос* позволяет с помощью мастера создать многотабличный запрос, ответ на который представляется в виде перекрестной таблицы, отображающей некоторую статистическую информацию. Для этого мастер предлагает набор статистических функций. Имена строк и столбцов таблицы задаются пользователем.
- ♦ *Повторяющиеся записи*. Этот способ позволяет с помощью мастера отобрать в динамический набор данных с заданными полями повторяющиеся записи из таблицы - источника.
- ♦ *Записи без подчиненных*. Этот способ позволяет построить мастеру многотабличный запрос, с помощью которого в динамический набор данных отбираются записи из родительской таблицы, не имеющие связанных по полю связи записей в дочерних таблицах.

Далее необходимо просмотреть динамический набор данных, получающийся после исполнения сформулированного запроса.

Потом сохранить, если необходимо, запрос с заданным пользователем именем. При этом динамический набор данных (ответ) не сохраняется, а сохраняется формулировка запроса. Сохраненный запрос можно использовать как источник данных наравне с таблицами при создании других запросов, а также форм и отчетов.

Исполнить запрос можно, находясь в режиме конструктора (кнопка «Выполнить» на панели инструментов, на которой изображен восклицательный знак), или после сохранения запроса, находясь в окне БД.

### Заполнение QBE-бланка запроса

При выборе на вкладке СОЗДАНИЕ в группе *Другие* *Новый объект:запрос* откроется окно бланка запроса и окно *Добавление таблицы*, с помощью которого надо перенести в верхнюю часть бланка требуемые для запроса таблицы и заполнить нижнюю часть бланка запроса. Для этого нужно:

- отобразить в строке *Поля* внизу бланка поля, которые необходимо использовать для получения ответа. Это можно сделать следующими способами:

- ♦ двумя кликами на имени поля в таблице, вынесенной в верхнюю часть бланка. При этом имя таблицы автоматически заносится в соответствующую строку внизу бланка,

- ♦ выделить имя поля в таблице в верхней части бланка, захватить мышью, перетащить в нужный столбец бланка в строку *Поле*,

- ♦ выбрать нужное поле из списка в строке *Поле* внизу бланка запроса

**Добавление нескольких полей** одновременно:

⇒ выделить несколько полей в списке полей в таблице в верхней части бланка (выделение смежных, т. е. расположенных рядом, полей производится при нажатой клавише Shift, а не смежных - при нажатой клавише Ctrl),

⇒ любое выделенное поле перетащить в соответствующий столбец в строку *Поле* внизу бланка запроса,

**Добавление всех полей** таблицы:

⇒ одним кликом на символе “звездочка”, расположенном вначале списка полей каждой таблицы, добавленной в верхнюю часть бланка,

⇒ перетащить мышью этот символ из нужной таблицы в соответствующий столбец строки *Поле*. В этом случае в динамический набор данных будут включены все поля таблицы;

- можно потребовать отсортировать данные динамического набора по одному или нескольким смежным полям. Если поля не смежные, то нужно выполнить перестановку столбцов;

- включить/выключить (если не нужно выводить на экран значения этого поля в динамическом наборе данных) флажок *Вывод на экран*;

- написать в соответствующих столбцах условия для отбора данных, если в этом есть необходимость. Условие отбора может быть записано в одном столбце бланка или в нескольких. В последнем случае запрос будет булевым с использованием операции AND. Чтобы построить булевский запрос с использованием операции OR, необходимо воспользоваться строкой *или* бланка запроса. По такому же принципу можно создать булевский запрос с использованием обеих операций.

### Примеры условий

- ♦ **2** (равно двум для числового поля, знак = можно не писать); **< >2**(не равно двум); **Москва** (равно символьной константе **Москва**, кавычки можно не ставить, ACCESS сделает это автоматически, преобразуя эту константу к виду “Москва”)

- ♦ **Between 10 and 50** (этим оператором можно отобрать значения от 10 до 50 включительно)

- ♦ **In( 1;2;5;12)**-функция, обозначающая принадлежность множеству значений 1,2,5,12

♦ **Null** (пусто)- не введено никакого значения. ACCESS в этом случае автоматически добавит оператор **Is** так, чтобы условие было записано в виде **IsNull**.

♦ **Not IsNull** (не пусто), т. е. значение введено

♦ **<#01.10.99#** (символ # можно не писать, ACCESS это сделает автоматически) - означает ранее указанной даты 01.10.99

♦ **>=Date() AND <=Date() + 6**, что означает позже сегодняшней даты (ее выдаст функция **Date()** и не позже шести дней после сегодняшней даты

♦ при задании условий можно использовать символы-заменители (шаблоны) вместе с оператором **Like**: \* - любая группа символов и ? - любой одиночный символ. Например, вывести все фамилии, начинающиеся на букву А, можно, задав условие отбора в виде **Like A\*** ( после нажатия клавиши **Enter** ACCESS преобразует это условие к виду **Like "A\*"**)

♦ отбор данных можно производить и в поле **Memo**. В этом случае осуществляется сканирование по заданной символьной константе всего текста поля **Memo**.

Для задания условия отбора можно воспользоваться построителем выражений.

### ***Работа с построителем выражений***

Вызов построителя выражений можно осуществить при помощи кнопки *Построитель* на панели инструментов либо командой *Построить* из контекстного меню.

Открыв окно построителя, в его поле ввода можно писать выражения с помощью клавиатуры, а можно условие формировать с помощью кнопок построителя, выбирая операнды из списков.

**Пример 1.** Пусть требуется ввести условие отбора данных в виде:

**>= Date() And <= Date() + 6**

Это выражение можно написать в окне ввода построителя, а можно действовать так:

- щелкнуть по кнопке **>**, а затем по кнопке **=** построителя
- вставить в выражение функцию **Date()**. Для этого:
  - ♦ открыть папку *Функции* в левом окне построителя
  - ♦ открыть папку *Дата/Время* в среднем окне построителя
  - ♦ выбрать функцию **Date()** в правом окне построителя
- аналогично открыть список логических функций в папке *Операторы* и вставить в выражение функцию **And**
- щелкнуть по кнопке **<**, а затем по кнопке **=** построителя
- вставить еще раз функцию **Date()**
- выбрать кнопку **+** построителя, затем набрать цифру **6** и нажать ОК.

Если при наборе выражения делается ошибка, то построитель автоматически может вставить слово, заключенное в угловые скобки (например, « выражение»), чтобы сообщить, какого типа элемент пропущен.

Удаление слова, вставленного построителем, осуществляется выделением слова и нажатием клавиши **Del**.

## Правила построения выражений

Выражения строятся из операндов, знаков операций (операторов) и функций.

### Операнды:

1) *Идентификаторы* (имена объектов ACCESS):

- полей (в таблице, запросе, форме, отчете)
- элементов управления в текущем или в другом объекте
- таблиц или запросов
- свойства

Полное имя объекта задается с использованием следующих символов:

[ ] - выделяет имя объекта, состоящее более, чем из одного слова (т. е. имеющее пробелы в имени), например [номер детали]

! - разделяет имена объектов, например [Поставщики]![номер поставщика]. Здесь Поставщики - имя таблицы, а номер поставщика - имя поля этой таблицы.

. (точка) - отделяет имена объектов от свойств, например

**Forms![Поставщики]![Отбор].OnClick**

Здесь Forms - служебное слово, указывающее тип объекта, имя которого указано далее, [Поставщики] - имя формы, [Отбор] - имя кнопки (элемента управления) на форме, а **OnClick** - имя свойства кнопки (Нажатие кнопки).

2) *Литералы*:

- текстовые - заключаются в двойные кавычки, например, "Москва"
- типа даты - заключаются в символы решетки, например, #31.01.99#
- числовые - записываются буквально, например, 5245

3) *Константы* (специальные литералы):

- "" - пустая строка соответствует текстовому выражению, которое ничего не содержит
- **Null** - пустое (не введенное) значение
- Истина (**True**)/ Ложь (**False**) - логические значения

### Операторы:

1) *Арифметические* (+ - \* / ^ \ Mod). Здесь символ \ означает деление без остатка (возвращает только целую часть частного), Mod - вычисляет остаток от деления

2) *Текстовые* (+ &). И тот и другой символ используются для слияния (конкатенации) строк текста

3) *Отношения или сравнения* (> >= = <> (не равно) < <= **Between...And**). **Between...And** означает в указанном диапазоне, например, **Between 5 And 10** означает в диапазоне от 5 до 10 включительно.

4) *Логические* (**And**, **Or**, **Not** и пр.)

5) *Присваивания* (=)

6) *Like*, содержащий символы подстановки:

\* - любая группа символов

? - один любой символ

Например, **Like "A\*"**, что означает начинающиеся на букву А.

**Функции:**

- *Пользовательские* (созданные пользователем)
- *Встроенные* (некоторые из них):
  - ♦ Математические: **Sin()**, **Sqr()**, **Abs()** и пр.
  - ♦ Текстовые (для работы с текстом): **Ltrim()**, **Rtrim()**, **Trim()** - удаляют ненужные пробелы в начале, в конце, в начале и в конце строки и пр.
  - ♦ Дата/ Время: **Year([Поставки]![Дата поставки])** - выделяет в виде четырех цифр года из поля Дата поставки таблицы Поставки
  - ♦ Статистические: **Max()**, **Sum()**, **Avg()** и пр.
  - ♦ Проверки: **IsNull()** - проверяет, имеет ли поле пустое (не введенное) значение, например **IsNull([Имя поставщика])** и пр.
  - ♦ Управления: **Iif()**, например **Iif([Цена]>100; "Дорогая"; "Дешевая")**, что означает: если условие **[Цена]>100** - истина, то функция возвращает значение **"Дорогая"**, если - ложь, то функция возвращает значение **"Дешевая"** и пр.

**Запросы с параметрами**

Не всегда можно решить на этапе создания запроса, какие именно значения должен отыскивать MS Access, а иногда требуется гибкое условие отбора, которое может изменяться в зависимости от текущей ситуации. Поэтому вместо условия отбора, непосредственно вводимого в бланк QBE, можно включить в запрос параметр, и в этом случае перед выполнением запроса MS Access каждый раз будет запрашивать конкретные условия отбора, например, **Введите город**.

Чтобы установить параметр необходимо ввести в строку "Условие отбора" имя или фразу, заключенную в квадратные скобки ([ ]). То, что заключено внутри квадратных скобок, MS Access рассматривает как имя параметра. Это имя выводится в диалоговом окне при выполнении запроса, поэтому разумно в качестве имени параметра использовать содержательную фразу. В одном запросе можно задать несколько параметров; при этом имя каждого параметра должно быть уникальным и информативным.

По умолчанию MS Access преобразует введенные в запрос параметры в текстовый тип данных. Однако, так как текстовые поля нельзя сравнивать с полями дат, тип данных для параметров необходимо установить точно. Для этого на вкладке **КОНСТРУКТОР** в группе *Показать или скрыть* необходимо выбрать *Параметры*, в результате чего откроется окно диалога "Параметры запроса". В этом окне вводится параметр без квадратных скобок и выбирается необходимый тип данных.

Теперь для выполнения запроса сначала необходимо ввести значение параметра, а результат запроса будет содержать только те поля, которые удовлетворяют заданному значению параметра.

**Итоговые запросы**

Итоговые запросы значительно отличаются от обычных. В них можно использовать два типа полей. Существуют поля, по которым осуществляется группировка данных, и поля, для которых проводятся вычисления. Дополнительно можно включить еще поля для определения условий запроса. Группирование данных позволяет получать статистическую, итоговую и др. информацию по отдельным группам данных. Группировать данные можно по одному или нескольким полям. При



этом в одну группу включаются данные с одинаковыми значениями в поле группирования. Например, все записи, сгруппированные по номеру поставщика, образуют столько групп, сколько имеется разных значений в поле **номер поставщика**.

Для того, чтобы составить итоговый запрос, находясь в режиме *Конструктора*, следует выбрать пиктограмму *Итоги*. В результате чего в бланке запроса появится строка "Групповая операция". Затем для соответствующего поля из списка необходимо выбрать нужную функцию:

**Группировка** - означает, что данное поле является полем группирования (устанавливается по умолчанию);

**Sum** - суммирование всех значений поля по группе данных;

**Avg** - среднее значение поля по группе;

**Min** или **Max** - минимальное или максимальное значение поля в группе;

**Count** - количество записей, входящих в группу;

**StDev** -среднеквадратическое отклонение;

**Var** – дисперсия;

**First** или **Last** - первое или последнее значение поля в группе;

**Выражение** - позволяет ввести выражение вместо имени поля в строке **Поле** бланка запроса;

**Условие** - показывает, что это поле используется только для задания условия отбора данных в запросе. Это поле не отображается в динамическом наборе данных.

Если группировка производится по нескольким полям, то слово **Группировка** должно быть в каждом из них.

## Перекрестные запросы

MS Access поддерживает специальный тип итоговых запросов, называемый перекрестным запросом. Такой запрос позволяет увидеть вычисляемые значения в виде перекрестной таблицы, напоминающей электронную таблицу.

Представьте себе таблицу товаров, которая содержит три поля данных: "Тип", "Поставщик" и "Цена". Для каждой категории товаров требуется определить суммы цен товаров с указанием поставщика.

Поместите в окно *Конструктора запроса* таблицу "Товары". Затем включите строку "Групповая операция". После этого поместите поле данных "Тип" в бланк запроса. Для этого поля выберите из списка значение "Группировка".

Во второй столбец бланка запроса поместите поле данных "Поставщик". Для этого столбца также выберите значение "Группировка".

Последним в бланк запроса поместите поле "Цена". Для этого поля выберите значение "Sum".

При выполнении запроса его результат отображается в режиме таблицы, в первом столбце которой будут все типы товаров, во втором – для каждого типа будут приведены номера поставщиков, а в последнем столбце – суммы цен товара данного типа и данного поставщика.

Теперь необходимо итоговый запрос преобразовать в перекрестный запрос.

Для этого на вкладке **КОНСТРУКТОР** в группе *Тип запроса* выберите *Тип запроса:перекрестный*.

Выбор данной команды приводит к замене в бланке запроса строки "Вывод на экран" на новую строку "Перекрестная таблица". Все остальные параметры остаются без изменения. Поле "Тип" исходной таблицы будет использовано в качестве

заголовка строк. Поэтому из поля списка "Перекрестная таблица" выберите элемент "Заголовок строк". Для столбца "Поставщик" выберите из списка "Перекрестная таблица" элемент "Заголовок столбцов".

Сумму цен товаров необходимо поместить в ячейках перекрестной таблицы. Поэтому для этого поля из списка "Перекрестная таблица" следует выбрать элемент "Значение".

При выполнении запроса его результат отображается в режиме таблицы, в которой номера категорий служат в качестве заголовков строк. Кроме этого, для каждого поставщика сумма цен товаров в перекрестной таблице размещена в отдельном столбце.

**Примечание:** При создании перекрестной таблицы вы можете выбрать только одно поле в качестве содержимого таблицы и только одно поле в качестве заголовка столбцов. В качестве заголовков строк можно указать и несколько полей исходной таблицы.

## Запросы на удаление

Все выше перечисленные запросы относятся к запросам-выборкам, которые позволяют только выбирать и просматривать данные из таблиц. Следующие запросы являются запросами-действиями, позволяющими модифицировать хранимую информацию.

С помощью запроса на удаление осуществляется удаление определенных данных из таблиц. Прежде чем сконструировать запрос на удаление, следует подумать, какие записи необходимо удалить. Самым надежным методом является предварительное создание запроса-выбора.

В режиме *Конструктора запроса* выберите таблицу, в которой позднее будет проводиться удаление. Затем определите критерии и посмотрите результат выполнения запроса, отображаемый в режиме таблицы. Быстрое переключение между режимами конструктора и выполнения обычно осуществляется с помощью одноименных пиктограмм, расположенных на панели инструментов.

Лишь в том случае, если в результате запроса присутствуют только подлежащие удалению записи, следует преобразовать запрос-выборку в запрос на удаление.

Для этого на вкладке **КОНСТРУКТОР** в группе *Тип запроса* выберите *Тип запроса:удаление*. После этого MS Access выключает в бланке запроса строки "Сортировка" и "Вывод на экран". Для запросов на удаление эти строки не нужны. Кроме того, для всех колонок бланка запроса в новой строке "Удаление" отображается текст "Условие". Однако критерии при этом не изменяются.

Если выполнить запрос выбором пиктограммы "ВЫПОЛНИТЬ", MS Access не отобразит на экране результат выполнения запроса. Вы можете прервать процесс удаления нажатием кнопки "Отмена". Однако нажатие ОК приводит к безвозвратному удалению записей, удовлетворяющих критериям запроса.

Если удаление выполняется из дочерней таблицы, то после исполнения запроса из таблицы будут удалены все записи, удовлетворяющие заданному условию. Если удаление указано из родительской таблицы и при этом ранее было установлено **Каскадное удаление связанных записей**, то удаление будет произведено не только из родительской, но и из всех дочерних таблиц, которые имеют записи, связанные с удаляемыми записями родительской таблицы.

## Запросы-обновления

Запрос-обновление позволяет производить обновление некоторой хранимой информации, своевременность которой истекла на текущий момент.

Сначала необходимо создать новый запрос выбора и ввести критерии для выбора изменяемых записей. Только в том случае, если результат запроса отвечает заданным требованиям, следует преобразовать его в запрос замены.

Для этого на вкладке КОНСТРУКТОР в группе *Тип запроса* выберите *Тип запроса:обновление*. При этом будут выключены строки "Вывод на экран" и "Сортировка" в бланке запроса. Обе эти строки не нужны для запроса замены. Вместо них включается новая строка "Обновление". В эту строку следует ввести новые выражения для заменяемых записей. Обратите внимание, что тип данных выражения должен совпадать с типом данных поля исходной таблицы.

## 2. ЗАДАНИЕ ПО РАБОТЕ

Получить у преподавателя вариант заданий к лабораторной работе №2. **Индивидуальные задания находятся в разделе «Варианты заданий для лабораторных работ».** Определить тип для каждого запроса и спроектировать запрос, пользуясь QBE-бланком.

## 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Подготовить отчет по работе.
2. С помощью режима *Конструктор запросов* спроектировать заданные запросы по определенному преподавателем варианту и предъявить их преподавателю.

## 4. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Название и цель выполнения работы.
3. Описание и виды сконструированных запросов.

### Контрольные вопросы для самостоятельной проработки

1. Что такое запрос и какие типы запросов поддерживает MS Access?
2. Где находится шаблон вывода при конструировании запроса-выборки?
3. Как задать параметры при создании запроса-выборки с параметрами?
4. Что такое запрос-действие и какие бывают запросы-действия?
5. Как можно добавить или удалить таблицу при конструировании запроса?
6. Что такое агрегированные операции и как они применяются в запросах?
7. Как определить, из каких таблиц взяты поля в шаблон вывода при конструировании запроса?
8. Что такое перекрестный запрос и как он конструируется?
9. Что такое SQL и как этот язык используется при формировании запросов?
10. Можно ли вводить данные в таблицы через запросы, если можно, то в каких случаях, каковы ограничения, накладываемые на взаимосвязанные таблицы, на ввод данных через запросы в таблицы?
11. Что такое поля группировки, можно ли использовать одновременно несколько полей группировки в одном запросе и каков смысл получаемого результата?

12. Какие связи между таблицами могут быть установлены в запросе и в каких случаях связи устанавливаются автоматически, а когда их можно установить вручную?

### Лабораторная работа №3

#### СОЗДАНИЕ ОБЪЕКТОВ ТИПА "ФОРМА" В СРЕДЕ MS ACCESS

*Цель работы:* изучение процесса создания объектов типа форм в режиме проектирования форм с использованием *Мастера*. Редактирование управляющих элементов в форме, добавление, удаление, изменение созданных ранее элементов управления формы. Задание нестандартных свойств элементам управления в форме, добавление вычисляемых полей в формах.

#### 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

В данной работе рассматриваются способы создания объектов базы данных типа «Форма» и редактирования управляющих элементов форм.

##### Формы

Формы являются основным средством создания интерфейса пользователя, который обеспечивает наиболее удобный способ представления, просмотра, редактирования данных и управления ходом выполнения приложения.

Преимущества формы для ввода и редактирования данных состоит в простоте и наглядности, так как записи таблицы или запроса представлены в форме в удобном виде. В режиме *Формы* можно в полной мере воспользоваться возможностями, предоставляемые графической средой Windows. Кроме того, можно создавать необходимые для решения задачи элементы формы. В формах пользователь самостоятельно определяет сколько места должно быть определено для представления конкретной информации. С их помощью значительно упрощается внесение изменений, добавление и удаление данных из базы.

Конструирование формы можно возложить на мастеров MS Access, или выполнить вручную с помощью конструктора форм, или сначала сконструировать с помощью мастеров MS Access, а затем подправить форму вручную.

Форма может быть однотабличной (если источником данных служит одна таблица или запрос), многотабличной (построенной на основе нескольких взаимосвязанных таблиц) или вообще без источника данных (чаще всего используется для реализации пользовательского кнопочного меню).

##### Мастера для создания форм

MS Access содержит большой набор инструментов для автоматического создания стандартных элементов интерфейса. Для создания новой формы сначала следует открыть окно базы данных. Затем необходимо выбрать на вкладке СОЗДАНИЕ в группе *Форма* пиктограмму *Другие формы* и среди перечисленных вариантов выбрать *Мастер форм*.

MS Access открывает диалоговое окно "Создание формы". В поле следует указать таблицу или запрос, данные из которых необходимо представить в форме. В

этом же диалоговом окне мастера необходимо определить поля данных, которые будут помещены в формы. Поля можно выбирать по отдельности и помещать в список "Порядок полей в форме" нажатием кнопки >. При нажатии кнопки >> мастер помещает в этот список все имеющиеся поля. Список определяет не только, какие поля данных будут присутствовать в форме, но и их очередность.

После чего следует нажать кнопку *Далее*. MS Access открывает окно со списком доступных мастеров по разработке форм. Выберите элемент "В один столбец" и нажмите кнопку ОК. В этом случае данные в форме будут размещены в один столбец.

Продолжите процесс создания формы, нажав кнопку "Далее". Если больше никаких установок делать не требуется, нажмите кнопку "Готово". В этом случае для всех остальных установок мастер использует значения по умолчанию.

MS Access предлагает пользователю еще несколько мастеров по созданию форм, например, мастер табличной формы создает формы, данные в которых представлены в виде таблицы. Мастер составных форм можно использовать для создания форм, в которых необходимо отображать информацию из двух связанных таблиц. Мастер простых форм предлагает самый быстрый способ подготовки форм и может быть запущен с помощью кнопки "Форма", расположенной на панели управления. Он создает стандартную форму за один шаг. При запуске мастера простых форм посредством пиктограммы форма будет основана на таблице или запросе, выбранных в данный момент в окне базы данных.

## Режим конструктора

Для редактирования формы используется режим *Конструктора*. Если форма уже открыта, следует выбрать в группе *Режимы* среди перечисленных вариантов *Конструктор*.

При выводе формы в режиме *Конструктора* автоматически выводится вкладка КОНСТРУКТОР, основной группой здесь является *Элементы управления*. Она позволяет добавлять элементы управления в формы.

Для выбранных элементов можно изменить шрифт, размер и другие свойства. Для этого используйте списки и кнопки, расположенные в этой группе. Расположение элемента формы можно изменить, используя технику буксирования.

В форме MS Access присутствуют три области:

- область заголовка (размещается заголовок и примечания формы),
- область колонтитулов (отображаются верхний и нижний колонтитулы),
- область данных (непосредственно отображаются данные таблицы).

Содержимое областей заголовка и колонтитулов не изменяется при перемещении по записям, в отличие от области данных. Различие между областями заголовка, колонтитулов и данных становится очевидным при распечатке формы.

Для каждой области формы можно установить свой цвет. Для этого следует сначала указать на соответствующую область, а затем выбрать требуемый цвет на палитре цветов (контекстное меню).

Для наглядности отображения в форме данных таблиц и запросов содержимое полей можно снабдить подписями. Подпись в этом случае будет пояснением к данным, отображаемым в поле.

При перемещении одного из этих элементов перемещается и другой. Для того чтобы с помощью буксировки по отдельности переместить подпись или поле, следует "ухватиться" за большой квадрат, находящейся в левом верхнем углу соответствующего элемента.

Для изменения размера элемента окна формы следует переместить в нужную позицию один из маркеров выделения.

Цвет, размер и местоположение являются только частью семейства свойств элементов формы. Полный список всех свойств элементов можно получить, открыв окно свойств путем выбора одноименной команды в контекстном меню. Если выделено несколько элементов окна, можно сразу изменить их общие свойства.

## Создание элементов формы

Создание элементов формы осуществляется в режиме *Конструктора*. На панели элементов находятся кнопки, соответствующие доступным элементам управления, которые могут присутствовать в форме, среди них можно выделить следующие кнопки:

- «Вкладка»
- "Выключатель",
- "Группа переключателей",
- "Вставить диаграмму",
- "Кнопка",
- "Линия",
- "Свободная рамка объекта",
- "Переключатель",
- "Подпись",
- "Подчиненная форма/отчет",
- "Поле",
- "Поле со списком",
- "Прямоугольник",
- "Присоединенная рамка объекта",
- "Список",
- "Флажок".

Каждая пиктограмма этой панели предназначена для создания элемента определенного типа. Назначение пиктограмм однозначно определяется их наименованиями. Каждый из элементов имеет набор свойств, причем свойства элементов различного типа могут не совпадать. Например, линия не предназначена для отображения текста, поэтому в списке ее свойств отсутствует тип и размер шрифта. Тип элемента можно определить с помощью окна свойств. MS Access отображает его в качестве заголовка окна.

### **Поле**

"Поле" является основным элементом отображения данных в форме. С помощью "Поля" можно в форме отобразить не только данные из таблиц и запросов, но и некоторые дополнительные данные, например текущую дату. Для этого необходимо создать элемент типа "Поле" в форме и определить в качестве свойства "Данные" для этого поля значение функции Now().

MS Access автоматически создает поле с подписью. Если этого не происходит, то можно воспользоваться отдельным элементом управления "Подпись". Он может быть использован для изображения любых надписей на форме.

## **Флажки, переключатели и выключатели**

Если список выводимых значений содержит лишь несколько значений, которые не подлежат изменению, для упрощения ввода и отображения данных в форме можно использовать выключатели, флажки и переключатели.

Конструктор форм позволяет создать для логических полей такие элементы, как "Флажок", "Переключатель" и "Выключатель". Если элемент отмечен крестиком, то значение логического поля равно Да, Истина или 1. В противном случае полю присваивается значение Нет, Ложь или 0.

Чтобы создать необходимый элемент управления, следует выбрать соответствующую пиктограмму группы *Элементы управления* и нажать мышью на пустое пространство в форме. Затем в качестве значения свойства "Данные" (контекстное меню) необходимо ввести имя поля исходной таблицы или запроса. Кроме этого, можно изменить подпись для созданного элемента.

Если в числовом поле данные могут принимать только определенные значения, для выбора этих значений можно использовать группы переключателей. Прежде всего, выберите пиктограмму "Группа переключателей" и нарисуйте рамку. Задайте в качестве значения свойства "Данные" имя числового поля, значение которого определяется с помощью группы переключателей.

Затем разместите в группе несколько переключателей. Для каждого переключателя группы в поле "Значение параметра" введите требуемое значение. Это значение будет введено в поле таблицы при выборе соответствующего переключателя.

Каждому элементу управления, включенному в группу, присваивается числовое значение, которое заносится в базовое поле таблицы или запроса при выборе этого элемента.

Каждый переключатель может быть снабжен смысловой подписью.

## **Списки**

Списки и поля со списком очень удобны при вводе данных для связанных таблиц, в этом случае обычно легче выбирать значение из списка, чем вводить его по памяти. Кроме того, список вариантов гарантирует, что в поле будут вводиться допустимые значения. Список вариантов может быть подключен к данным или содержать фиксированные значения, введенные при создании элемента управления.

Для определения списка следует перейти в режим *Конструктора*, выбрать пиктограмму *Список* группы *Элементы управления* и нажать мышью на пустое пространство в форме. Откроется мастер списков, далее необходимо следовать инструкциям мастера.

MS Access автоматически создает новый список и полю "Данные" присваивает значение выбранного поля. Свойства "Тип источника строк" и "Источник строк" определяют происхождение данных, которые будут отображаться в списке. Другие свойства определяют внешний вид списка.

## **Поле со списком**

Создание полей со списком осуществляется аналогичным образом. При этом в распоряжение предоставляются точно такие же свойства. Поля со списком занимают на форме меньше места, а список их значений выводится на экран только по указанию пользователя. Списки занимают больше места, однако в них всегда отображается несколько возможных для ввода значений.

Элемент управления является как бы объединением поля и списка. Значение этого элемента управления может быть введено с клавиатуры или выбрано из предварительно подготовленного списка, который может быть сформирован из некоторой таблицы или запроса.

В режиме списка нижняя часть поля со списком (список) не изображается до тех пор, пока не будет нажата кнопка раскрытия списка (со стрелкой).

**Примечания:** Значение свойства "Ограничиться списком" (LimitToList) определяет, можно ли вводить в поле со списком значения, которых нет в списке.

### **Элемент управления кнопка**

Этот элемент управления предназначен для запуска макроса, вызова функции MS Access Basic или выполнения процедуры обработки событий. Например, форма может содержать кнопку, при нажатии которой открывается другая форма.

Если мастера по созданию элементов управления включены, можно создать кнопку с помощью мастера. Для этого необходимо на вкладке *Конструктор* в группе *Элементы управления* выбрать команду *Кнопка*. В сетке конструктора щелкнуть то место, куда требуется вставить кнопку. При этом будет запущен мастер создания кнопок. Далее необходимо следовать указаниям мастера. Мастер создает кнопку и внедряет макрос в свойство «Нажатие кнопки» кнопки. Этот макрос содержит команды для выполнения задач, выбранных в мастере.

Для размещения текста или рисунка на поверхности кнопки следует определить ее свойство "Подпись поля"(Caption) или "Рисунок"(Picture) соответственно.

Для того чтобы указать, что данная кнопка вызывает выполнение стандартного действия или является кнопкой Отмена, следует определить ее свойства "По умолчанию"(Default) и "Отмена"(Cancel) соответственно.

### **Рисунки и другие объекты**

В форму можно включить простые графические объекты, такие как линии и прямоугольники. Для вычерчивания линий или прямоугольников следует выбрать соответствующую пиктограмму в группе *Элементы управления*, а затем нажать мышью на пустое пространство в форме.

Кроме того, MS Access поддерживает Windows-технология OLE (Object Linking and Embedding – размещение и связывание объектов). Благодаря этому Вы можете встроить в свои формы объекты, созданные любой из Windows-программ, поддерживающих режим OLE-сервера. В качестве таких объектов могут быть использованы рисунки, диаграммы и т.п.

Для размещения объекта в форме следует выбрать пиктограмму *Свободная рамка объекта* в группе *Элементы управления*. MS Access открывает диалоговое окно "Microsoft Office Access", содержащее список средств, с помощью которых можно встроить объект в форму, а также позволяющее встроить в форму уже готовые объекты. Выберите соответствующее средство или объект и нажмите ОК. Например, чтобы создать диаграмму, можно выбрать "Диаграмма Microsoft Graph". В результате будет запущена программа Microsoft Graph, с помощью которой можно создать требуемый объект. Объект будет сохранен вместе с формой.

Готовые рисунки можно вставить в форму также при помощи команды *Рисунок* в группе *Элементы управления*.

Рисунки и другие объекты, встраиваемые в формы можно связывать с формой



целиком, либо с каждой записью формы.

### **Вычисляемые элементы управления**

Элементы формы могут быть связанными и несвязанными. Связанные элементы привязаны к полю исходной таблицы или запроса. Несвязанные элементы, как правило, отображают результат вычисления.

Откройте форму в режиме *Конструктора* и выберите пиктограмму *Поле* в группе *Элементы управления*. Затем укажите мышью место в пределах формы, в результате чего MS Access создаст новое поле и подпись. Откройте окно свойств поля и для свойства "Данные" введите выражение, задающее его значение.

Так как MS Access не может самостоятельно определить тип вычисляемых данных при создании формы, необходимо самостоятельно установить значение свойства "Формат поля".

Область примечаний формы является наиболее подходящим местом для отображения итоговых значений.

Отображаемая в форме информация не ограничивается таблицей или запросом. Например, Вы можете воспользоваться уже знакомой Вам функцией Date(), определяющей системную дату. Кроме того, при помощи несвязанных элементов управления возможно выполнение программ, написанных на встроенном языке программирования Visual Basic.

### **Построитель выражений**

Для создания полей, информация в которых отсутствует в таблицах или запросах, используется построитель выражений.

Создайте форму для исходной таблицы, поместив в нее все необходимые поля. Далее откройте форму в режиме *Конструктора* и разместите в ней новое поле. Вызовите окно свойств, перейдите в поле "Данные". Справа, рядом со строкой ввода, находится кнопка вызова построителя выражений. Нажатие этой кнопки приводит к открытию окна построителя выражений. Диалоговое окно построителя выражений состоит из двух областей.

Работа с построителем выражений подробно описана в методических указаниях к лабораторной работе №2 (файл **лаб\_работа\_2.docx**).

### **Подчиненная форма**

Часто требуется визуально представить отношение 1 : М между элементами данных. Для этого используется режим подчиненной формы. Этот режим позволяет создать подчиненную форму, источником данных для которой служит дочерняя таблица (таблица со стороны "многие" в схеме данных) в окне основной формы, источником данных для которой служит родительская таблица (со стороны "один" в схеме данных). При этом для текущей записи основной формы в подчиненной форме будут отображаться только связанные с ней записи, что создает максимальные удобства для ввода и редактирования записей в обеих формах.

Например, при представлении издательства мы хотим видеть список книг, изданных в данном издательстве. Таблица с данными об издательствах связана с таблицей "Книги". Связь осуществляется посредством поля "Код издательства", присутствующего в обеих таблицах.

Ниже описана процедура, которая создает новую комбинацию формы и подчиненной формы при помощи мастера форм.

На вкладке *Создание* в группе *Формы* щелкните *Другие формы*, а затем в

списке щелкните пункт *Мастер форм*. На первой странице мастера в раскрывающемся списке *Таблицы и запросы* выберите таблицу или запрос. Например, чтобы создать форму «Издательство», которая выводит в подчиненной форме список книг, изданных в данном издательстве, выберите пункт **Таблица: Издательство** (сторона «один» отношения «один-ко-многим»). Выберите поля таблицы, которые требуется включить в форму.

На той же странице мастера в раскрывающемся списке *Таблицы и запросы* выберите другую таблицу или запрос. В данном примере выберите таблицу «Книги» (сторона «многие» отношения «один-ко-многим») и укажите поля таблицы, которые требуется включить в форму.

Если до запуска мастера отношения установлены правильно, то после нажатия кнопки **Далее** будет задан вопрос **Выберите вид представления данных**. В мастере схематически отображается структура формы. Прямоугольник в нижней части формы обозначает подчиненную форму. Установите переключатель в нижней части страницы мастера в положение **Подчиненные формы** и нажмите кнопку **Далее**.

На последней странице введите заголовки форм. Имена автоматически присваиваются формам на основе введенных заголовков, а надписи подчиненных форм — на основе введенных заголовков подчиненных форм. После нажатия кнопки **Готово** будет создано две формы: одна для главной формы, содержащей элемент управления подчиненной формы, а другая — для самой подчиненной формы.

Откройте форму в режиме просмотра и ввода данных.

В верхней части формы расположены поля таблицы "Издательства", ниже — таблица, образованная полями таблицы "Книги". Особенность заключается в том, что в форме отображаются только те из записей таблицы "Книги", которые связаны с текущей записью таблицы "Издательства".

При редактировании данных MS Access автоматически поддерживает целостность данных. Поэтому запись об издательстве можно удалить лишь после того, когда будут удалены все записи о соответствующих ему книгах.

При вводе информации о новых книгах MS Access автоматически связывает записи с текущим издательством. Поэтому пользователю не нужно самому заполнять поле "Код издательства". Это поле можно удалить из подчиненной формы.

Составная форма может быть построена и иным путем. Для этого сначала разрабатывается подчиненная форма в виде ленточной формы. А потом разрабатывается основная форма, а подчиненная форма встраивается путем простого перетаскивания готовой подчиненной формы и размещения ее внутри основной.

## Свойства формы

Каждая форма имеет свойства, значения которых определяют ее вид и поведение, также MS Access представляет более 40 свойств событий форм, которые могут запускать макросы (или процедуры Visual Basic). Ниже приведен перечень этих событий:

- Открытие и закрытие форм.
- Изменение данных.
- Обнаружение смены фокуса.
- События работы с клавиатурой и мышью.
- Печать.
- Включение пользовательского меню формы.
- Отслеживание ошибок.

- Истечение промежутка времени.  
Разобраться со свойствами формы Вам необходимо самостоятельно.

## 2. ЗАДАНИЕ ПО РАБОТЕ

Получить у преподавателя вариант заданий к лабораторной работе №3. **Индивидуальные задания находятся в разделе «Варианты заданий для лабораторных работ».** При создании форм использовать *Мастер форм*, а для редактирования *Конструктор*.

## 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Получить у преподавателя индивидуальное задание.
2. Подготовить отчет, в который включить проекты проектируемых форм.
3. Выполнить работу и предъявить преподавателю результаты работы.

## 4. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Название и цель выполнения работы.
3. Виды сконструированных форм.

### Контрольные вопросы для самостоятельной проработки

1. Что такое объект типа форма, какие типы форм предлагает Вам создать мастер и чем они отличаются одна от другой?
2. Что такое элемент управления в форме, какие в MS Access используются элементы управления в форме?
3. Что такое список и поле со списком в форме, чем они отличаются, каковы свойства этих элементов управления?
4. Что такое элемент типа переключатель, как он работает, как и зачем объединяют эти элементы в группы?
5. Что такое подпись элемента управления и как ее можно изменить, переместить, убрать?
6. Какие разделы объекта типа форма используются при разработке экранных форм и что они означают?
7. Где задаются свойства объекта типа форма?
8. Чем отличается свойство объекта, встроенного в форму от свойства самой формы?
9. Из каких областей состоит форма и каково их назначение? Как можно изменить цвет формы целиком или по частям?
10. Можно ли сделать цвет фона объекта типа поле таким же как цвет фона формы, если цвет формы отличен от стандартного серого, если можно, то как?
11. Чем является элемент встроенная форма по отношению к основной форме? Какие свойства элемента типа встроенная форма используются для синхронизации работы основной и встроенной формы?
12. Что такое имя формы и какая надпись появляется в заголовке окна формы? Как можно изменить эту надпись? Какие стандартные системные элементы управления

появляются в окне формы, с какими свойствами формы эти элементы связаны и как их можно изменить?

13. Как можно заменить действия стандартных системных кнопок: закрытие, разворачивание, сворачивание?
14. Что такое область номера записи, каким образом можно изменить ее внешний вид, можно ли убрать эту область и какими элементами управления ее можно заменить?

## **Лабораторная работа №4**

### **РАЗРАБОТКА ОТЧЕТОВ В СРЕДЕ MS ACCESS**

*Цель работы:* изучение процесса создания объектов типа отчет; редактирование полей отчета, группировка, задание вычисляемых значений полям отчета; работа с областями отчета.

#### **1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

В данной работе рассматриваются способы создания объектов базы данных типа «Отчет», виды этих объектов и их оформление в режиме *Конструктора*.

#### **Отчеты**

Отчеты позволяют представить и распечатать данные в соответствии с требованиями пользователя. Причем возможности оформления данных для вывода на печать почти настолько же гибки, как и возможности отображения их на экране.

Отчеты очень похожи на формы. Так, например, режимы конструктора форм и отчетов почти идентичны. Это также касается панели инструментов, панели элементов, а также средств, предназначенных для размещения и работы с элементами управления.

При работе над отчетом Вы можете использовать те же области, что и при создании форм: области заголовка и примечания, области верхнего и нижнего колонтитулов, а также область данных. Кроме того, Вы можете включить в отчеты до четырех областей для полей, по которым осуществляется группировка записей.

MS Access печатает верхний колонтитул вверху каждой печатной страницы. Нижний колонтитул размещается в конце каждой печатной страницы. В отличие от них, заголовок отчета печатается один раз в начале, а примечание отчета – один раз в конце отчета. Содержимое области данных печатается один раз для каждой записи исходной таблицы или запроса. Кроме того, пользователь может определить несколько полей, по которым будет проводиться группировка записей исходной таблицы или запроса. Тогда для каждого поля, по которому проводится группировка данных, MS Access в качестве областей отчета формирует заголовок группы и примечание группы.

Однако между формами и отчетами имеется существенное различие – отчеты предназначены исключительно для вывода данных на печать. В них отсутствует необходимость наличия управляющих элементов для ввода данных. Поэтому в отчетах можно отказаться от использования списков, полей со списком и флажков.

Так как отчеты предназначены только для вывода данных на печать, для них нельзя установить режим таблицы или формы. Для отчета можно выбрать только режим конструктора и предварительного просмотра. При обработке больших таблиц

или запросов иногда прибегают к просмотру отчетов, осуществляемому с помощью команды *Файл/Печать/Предварительный просмотр*. Как правило, данная операция требует меньше времени по сравнению с созданием отчета, однако в этом случае в ваше распоряжение предоставляется минимальный набор средств.

## Создание отчетов

Наиболее быстрый способ создания отчета, всего за один шаг, заключается в выборе таблицы или запроса в окне базы данных, а затем – пиктограммы *Отчет* на вкладке СОЗДАНИЕ в группе *Отчеты*.

Приложение Access создаст отчет и отобразит его в режиме макета. После просмотра отчет можно сохранить, а затем закрыть и его, и источник записей — таблицу или запрос. В следующий раз при открытии отчета программа Access отобразит в нем самые последние данные из источника записей.

Еще один простой путь создания отчета состоит в использовании интегрированных в MS Access мастеров отчетов. В мастере отчетов предоставляется больше возможностей относительно выбора полей для включения в отчет. При этом можно указать способ группировки и сортировки данных, а также включить в отчет поля из нескольких таблиц или запросов, если отношения между этими таблицами и запросами заданы заранее.

Для создания отчета таким образом на вкладке СОЗДАНИЕ в группе *Отчеты* щелкните *Мастер отчетов*. Следуйте указаниям на страницах мастера отчетов. На последней странице нажмите кнопку **Готово**. Если нужно добавить в отчет поля из нескольких таблиц или запросов, не нажимайте в мастере отчетов кнопку **Далее** или **Готово** после выбора полей из первой таблицы или запроса. Повторите процедуру выбора для другой таблицы или запроса, а также выберите щелчком мыши другие поля, которые нужно добавить в отчет. Затем для продолжения нажмите кнопку **Далее** или **Готово**. Предварительный просмотр отчета позволяет увидеть, как будет выглядеть отчет при печати. Можно также увеличить масштаб для просмотра деталей.

Еще один мастер - мастер наклеек помогает создавать наклейки большинства стандартных размеров. На вкладке СОЗДАНИЕ в группе *Отчеты* нажмите кнопку *Наклейки*. Следуйте указаниям на страницах мастера наклеек. В приложении Access наклейки отображаются в режиме предварительного просмотра, дающем представление о том, как они будут выглядеть в напечатанном виде. При помощи ползунка в строке состояния можно увеличить масштаб, чтобы увидеть детали.

Если по каким-либо причинам для построения отчета не подходят ни средство «Отчет», ни мастер отчетов, можно создать отчет с нуля при помощи средства «Пустой отчет». Это очень быстрый способ, особенно если в отчете должно содержаться небольшое количество полей. Чтобы воспользоваться средством «Пустой отчет», выполните следующие действия. На вкладке СОЗДАНИЕ в группе *Отчеты* щелкните *Пустой отчет*. В режиме макета появится пустой отчет, а в правой части окна Access — область «Список полей». По очереди перетаскивайте каждое из нужных полей в отчет. С помощью инструментов, представленных в группе *Элементы управления* на вкладке **ФОРМАТ**, можно добавить в отчет эмблему компании, заголовок, номера страниц, дату и время.

## Отчеты с группировкой данных

Отчеты MS Access представляют мощные средства, позволяющие сортировать и группировать данные и проводить итоговые расчеты самыми разнообразными

способами.

После запуска мастера отчетов в первом диалоговом окне надо выбрать поля исходной таблицы или запроса, которые должны быть представлены в отчете.

Во втором и третьем диалоговых окнах можно задать до четырех полей, по которым будет осуществляться группировка данных. Например, если вы создадите отчет для таблицы товаров, то здесь можно указать поля "Тип" и "Цена". В этом случае мастер создает отчет, в котором данные исходной таблицы объединены по категориям и ценам.

Кроме этого, если в отчете представлены числовые поля, для каждой группы мастер автоматически вычисляет сумму значений, содержащихся в этих полях. Так, например, можно создать отчет, в котором товары сгруппированы по категориям и ценам и для каждой категории указана сумма цен товаров.

Если поля группировки не заданы, то в следующем диалоговом окне можно указать порядок сортировки данных исходной таблицы или запроса. Если же поля группировки определены, то данное диалоговое окно содержит только те поля, которые не были выбраны в предыдущем окне, и в нем можно задать последовательность сортировки данных внутри группы. В следующих диалоговых окнах выбирается стиль оформления отчета и ориентация бумаги.

Если вы хотите самостоятельно внести изменения в групповой отчет, в режиме *Конструктора* в группе *Группировка и итоги* следует выбрать пиктограмму *Группировка*. После чего MS Access открывает одноименное окно, в котором можно выбрать поля для группировки отчета.

Задание управляющего элемента в одной из областей отчета автоматически определяет его функцию. Например, если в области примечаний группы "Тип" определить управляющий элемент с выражением: =Sum( [Цена] ) в результате будет вычислена сумма цен на товары по каждой категории.

## Конструктор отчетов

Оформление отчета осуществляется в режиме конструктора. Переход из окна базы данных в режим конструктора осуществляется нажатием кнопки *Конструктор*.

Режим конструктора для отчетов очень похож на аналогичный режим для форм. Даже панель элементов для создания новых управляющих элементов та же.

По умолчанию MS Access различает в отчете пять областей. Заголовок и примечание отчета печатаются один раз в начале и в конце отчета. Верхний и нижний колонтитулы страницы выводятся соответственно вверху и внизу на каждой отпечатанной странице.

Наиболее важной является область данных отчета. MS Access распечатывает ее один раз для каждой записи исходной таблицы или запроса.

Каждая из пяти областей отчета имеет собственные свойства. В контекстном меню выберите *Свойства*, чтобы MS Access открыл окно свойств. В строке заголовка окна свойств отображается текст: "Раздел" и имя соответствующей области.

С помощью свойства "Конец страницы" вы определяете, должен ли осуществляться переход к новой странице перед началом области. Интересной является также свойство "Не разрывать". Если установить для данного свойства значение "Да", MS Access попытается напечатать управляющие элементы области на одной странице.

Однако, если на текущей странице не достаточно места для всех управляющих элементов области, тогда MS Access печатает все управляющие элементы на следующей странице. Свойство "Вывод на экран" определяет, должна ли

отображаться на экране данная область отчета. В отчетах некоторые из управляющих элементов содержат свойства "Расширение" и "Сжатие". Аналогичные свойства в формах отсутствуют.

### Использование вычисляемых значений

MS Access предоставляет десятки встроенных функций, которые можно использовать для обработки данных или для добавления информации в отчет. Одним из видов информации, которая наиболее часто включается в отчет, является информация о дате подготовки отчета или номера текущих страниц. Для дат MS Access предоставляет две встроенные функции Date() и Now(), возвращающие текущие системные даты. Чтобы добавить текущую дату в отчет, необходимо создать несвязанное поле и установить в его свойстве "Данные" значение =Date() (или Now()).

Чтобы добавить номер страницы, используйте свойство отчета "Страница" (Page), посредством создания несвязанного поля и установки его свойства "Данные" равным "Страница" (Page).

### Главные и подчиненные отчеты

Помимо внедрения в другие формы подчиненных форм существует возможность вставлять подчиненные отчеты (или подчиненные формы) в другие отчеты.

На первом шаге создайте стандартный отчет для подчиненной таблицы. Сохраните отчет под именем "Подчиненный". На втором шаге создайте отчет для главной таблицы. Сохраните данный отчет под именем "Главный".

Перейдите в окно базы данных. Затем в режиме *Конструктора* буксируйте подчиненный отчет из окна базы данных в область данных главного отчета. В результате MS Access автоматически размещает управляющий элемент для подчиненного отчета.

MS Access автоматически установил связь между таблицами на основании отношения, определенного в модели данных. В этом можно убедиться с помощью конструктора отчета.

Отметьте, что MS Access ввел для свойств управляющего элемента "Подчиненные поля" и "Основные поля" наименование связующего поля. Посредством этих двух свойств устанавливается связь между главным и подчиненным отчетами. При объединении главного и подчиненного отчетов для двух таблиц, которые не были связаны предварительно, вы должны самостоятельно определить значения этих свойств.

При просмотре отчета становятся видны некоторые недостатки размещения управляющих элементов в отчете, которые можно легко устранить. Так, например, подчиненный отчет имеет собственную нумерацию страниц и заголовок. Поскольку в данном случае подчиненный отчет встроен в главный отчет, оба элемента оказываются лишними.

Откройте подчиненный отчет в режиме *Конструктора*. Удалите заголовок, нумерацию страниц и все другие управляющие элементы, которые больше не понадобятся. Access будет учитывать изменения, выполненные в подчиненном и в главном отчетах.

## 2. ЗАДАНИЕ ПО РАБОТЕ

Получить у преподавателя вариант заданий к лабораторной работе №4. **Индивидуальные задания находятся в разделе «Варианты заданий для лабораторных работ».**

При создании отчетов использовать *Мастер отчетов*, а для редактирования *Конструктор*.

## 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Подготовить отчет по работе.
2. Спроектировать заданные отчеты по определенному преподавателем варианту и предъявить их преподавателю.

## 4. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Название и цель выполнения работы.
3. Виды сконструированных отчетов.

### Контрольные вопросы для самостоятельной проработки

1. Какие типы *Мастеров* для создания отчетов имеются в MS Access?
2. Что представляет из себя бланк отчета и как его разделы располагаются на странице при печати?
3. Как производится группировка данных в отчете?
4. Как включить в отчет заголовок и примечание группы, для чего они используются?
5. Как ввести в отчет пропущенное поле данного и как задать его свойства?
6. Как ввести в отчет вычисляемое поле?
7. Где располагаются итоговые данные по группам и по всему отчету?
8. Как добавляется в отчет описательная текстовая информация?
9. Что означает свойство раздела “Не разрывать”?
10. Что означает свойство раздела “Вывод на экран”?
11. Каким образом установить размер, формат и шрифт поля?
12. Как выровнять группу данных по горизонтали, по вертикали?
13. Каким образом вводятся в отчет текущая дата распечатки и номера страниц?

## Лабораторная работа №5

### РАБОТА С МАКРОСАМИ В СРЕДЕ MS ACCESS

*Цель работы:* изучение принципов использования макрокоманд в MS Access.



## 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

В данной работе рассматриваются макрокоманды и способы создания объектов базы данных типа «Макрос».

### Макросы

Макрос - это объект БД. Язык макросов является средством программирования, которое позволяет реализовать задачи пользователя, выполняя необходимые действия над объектами БД и их элементами.

Макрос - это программа, состоящая из последовательности макрокоманд. Макрокоманда - это инструкция, ориентированная на выполнение определенного действия. Например, макрокомандой можно открыть форму, отчет, запустить на выполнение запрос, применить фильтр, присвоить значение и пр. Язык макросов обеспечивает возможность решения большинства задач пользователя, не прибегая к программированию на языке Visual Basic.

Макросы определяют набор методов обработки объектов и их элементов. В ACCESS имеются средства, обеспечивающие взаимодействие макросов с объектами и их элементами на основе событий. События наступают при выполнении определенных действий, к которым, прежде всего, относятся действия пользователя. Последнее обстоятельство позволяет управлять выполнением программы извне. Пользователь, выполняя различные действия, например в формах (нажатие кнопки мыши, ввод и редактирование данных, выбор элемента списка и пр.) инициирует выполнение макросов, автоматизирующих решение связанных с действиями пользователя задач. Такой подход существенно отличает программирование на языке макросов от программирования в традиционном понимании, при котором только программа управляет процессом.

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

### Создание макросов

Макросы могут быть простые и групповые. Простой макрос - это последовательность макрокоманд. Групповой макрос - это последовательность простых макросов. При этом каждый простой макрос внутри группового должен иметь свое имя. Поскольку язык макросов является языком программирования, то он должен иметь средства для описания всех типов вычислительных процессов: линейных, разветвляющихся и циклических.

Чтобы создать макрос, необходимо выбрать пиктограмму *Макрос* на вкладке СОЗДАНИЕ в группе *Другие*. При этом откроется окно для конструирования макроса, а также появится панель инструментов конструктора макросов.

Для создания любого макроса окно конструктора макросов должно иметь минимум два столбца: *Макрокоманда* и *Примечание*. В примечаниях пользователь пишет обычно поясняющую информацию, которая в принципе может отсутствовать. В каждой строке пользователь из раскрывающегося списка выбирает имя нужной макрокоманды и указывает для нее необходимые аргументы. Некоторые аргументы задаются по умолчанию. Пользователь может им задать другие значения (обычно

выбрать из раскрывающегося списка). Такие аргументы, как имя объекта, можно перетащить мышью из окна БД. Если создается групповой макрос, то необходим третий столбец **Имя макроса**, который можно открыть соответствующей кнопкой на панели инструментов конструктора макросов. Если макрос реализует разветвляющийся процесс, то еще необходим и столбец **Условий** (кнопка **Условия** на панели инструментов конструктора макросов).

Запустить макрос можно из окна объектов БД или при помощи кнопки **Выполнить** на панели инструментов конструктора макросов. Также запустить макрос можно из другого макроса с использованием макрокоманды **ЗапускМакроса**.

Для отладки макроса используют режим запуска макроса по шагам, при котором макрос приостанавливает свою работу после выполнения каждой макрокоманды, так что пользователь может оценить правильность решения задачи. Установку этого режима можно сделать, находясь в режиме конструктора макроса. Далее, как обычно, запустить макрос на выполнение. Макрос будет выполняться по шагам.

После выполнения каждого шага можно продолжить выполнение следующего шага кнопкой **Продолжить** или закончить выполнение макроса, прервав его работу кнопкой **Остановить все макросы**.

### ***Пример простого линейного макроса***

Создать макрос, открывающий форму **Детали** в виде таблицы и отображающий только гайки, т. е. нужно открыть отфильтрованные данные. Затем форму закрыть.

Макрос, решающий поставленную задачу, будет таким:

Макрокоманда	Примечание
<b>ОткрытьФорму</b> (Имя формы: Детали, Режим: Таблица, Условие отбора: [имяд]="гайка")	Задание условия отбора позволяет отфильтровать нужные записи.
<b>Сообщение</b> (Сообщение: после просмотра нажмите ОК)	Для приостановки выполнения макроса. Выполнение макроса продолжится после того, как пользователь нажмет кнопку <b>ОК</b> в окне сообщений.
<b>Закрыть</b> (Тип объекта: Форма, Имя объекта: Детали)	

### ***Пример автоматически выполняемого макроса***

Создать макрос, при помощи которого автоматически откроется нужная форма при открытии базы данных.

Этот макрос будет называться **AutoExec** и выглядеть так:

Макрокоманда	Примечание
<b>ОткрытьФорму</b> (Имя формы: Кнопочная форма)	Задание той формы, которая должна открываться автоматически

При следующем открытии базы данных автоматически будет открываться форма "Кнопочная форма".

Если в исключительном случае потребуется подавить выполнение AutoExec-макроса, при открытии базы данных следует удерживать нажатой клавишу [Shift].

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

## Краткая характеристика макрокоманд

MS Access имеет около 50 макрокоманд, которые могут быть разбиты на группы (отобразить большее или меньшее количество макрокоманд можно при помощи кнопки **Показать все действия** на панели инструментов конструктора макроса).

### 1) Данные в формах и отчетах

- Отбор данных

**ПрименитьФильтр.** Позволяет отфильтровать данные в таблице, форме или отчете путем задания в аргументах макрокоманды имени фильтра (сохраненного как запрос), имени запроса или условия отбора.

- Перемещение по данным

**СледующаяЗапись**

**НайтиЗапись.** Позволяет отыскивать запись по заданному образцу поиска.

**КЭлементуУправления.** Позволяет переместить фокус на указанный элемент управления в форме **НаСтраницу**

**НаЗапись.** Позволяет установить указатель записи на указанную запись (новую, следующую, предыдущую первую, последнюю или конкретную)

### 2) Выполнение

- Выполнение команды

**ВыполнитьКоманду.** Позволяет выполнить одну из команд из предложенного списка команд.

- Выполнение макроса, процедуры или запроса

**ОткрытьЗапрос**

**ЗапускПрограммы.** Позволяет запустить на выполнение пользовательскую функцию с заданным именем и аргументами, которая создана пользователем в модуле на языке VBA.

**ЗапускМакроса**

**ЗапускЗапросаSQL**

- Выполнение другого приложения

**ЗапускПриложения.** Здесь имеется в виду любое приложение (в аргументах макрокоманды указывается полная спецификация исполняемого файла)

- Прерывание выполнения

**ОтменитьСобытие.** Вызывает отмену события, которое привело к запуску макроса, содержащего эту макрокоманду. Например, если событие **До обновления** запускает макрос проверки условия, и условие нарушается, то эта макрокоманда приведет к отмене обновления.

**Выход** (из MS Access)

**ОстановитьВсеМакросы**

**ОстановитьМакрос** (исполняемый)

### 3) Импорт/Экспорт

- Передача объектов

**ВывестиВФормате.** Позволяет вывести данные объекта БД (таблицы, формы и др.) в формате MS Excel, RTF или текста в формате MS DOS.

**ОтправитьОбъект** ( по электронной почте)

- Преобразование данных

**ПреобразоватьБазуДанных.** Позволяет импортировать данные из другой БД в текущую, экспортировать данные из текущей БД в другую БД или присоединять таблицы из другой БД к текущей.

**ПреобразоватьЭлектроннуюТаблицу** (то же, что и в предыдущей макрокоманде, но применительно к таблицам Excel).

**ПреобразоватьТекст** (то же, что и в предыдущей макрокоманде, но применительно к текстовым файлам).

### 4) Работа с объектами БД

- Копирование или переименование объекта

**КопироватьОбъект**

Удаление объекта

**УдалитьОбъект**

- Изменение размеров или положение окна

**Развернуть** (распахнуть во весь экран)

**Свернуть** (до значка)

**СдвигРазмер** (задаются новые координаты левого верхнего угла окна объекта и могут задаваться его размеры, Здесь можно указывать не все аргументы).

**Восстановить** (до прежних размеров)

- Открытие или закрытие объектов

**Закрыть** (тип и имя закрываемого объекта указывается в качестве аргументов)

**ОткрытьТаблицу**

**ОткрытьФорму**

**ОткрытьЗапрос**

**ОткрытьОтчет**

**ОткрытьМодуль**

- Печать объекта

**Печать**

- Выделение объекта

**ВыделитьОбъект** (эта макрокоманда должна предшествовать таким макрокомандам как **КопироватьОбъект**, **ПереименоватьОбъект**, **УдалитьОбъект** и др.)

- Определение значения поля, элемента управления или свойства

**ЗадатьСвойство.** Задаёт значение поля или другого ЭУ или свойства в форме или отчете.

- Обновление данных или экрана

**ОбновитьОбъект.** Как правило, ACCESS откладывает обновление изображаемых на экране ЭУ до тех пор, пока не завершит выполнение более важных задач. Эта же макрокоманда выполняет обновление указанного в аргументах объекта немедленно. Макрокоманда часто используется, например, после изменений значений нескольких ЭУ макрокомандой **ЗадатьЗначение**. Эти изменения могут быть не отражены на экране немедленно, особенно в том случае, если другие ЭУ (например,

вычисляемые) зависят от значений измененных макрокомандой **ЗадатьЗначение** элементов. В этом случае, чтобы увидеть све сделанные изменения можно объект закрыть, а затем снова открыть, что неудобно, а можно исполнить макрокоманду **ОбновитьОбъект**. Однако при этом не происходит повторного просмотра источника объекта. Повторный просмотр источника с перевыполнением всех используемых запросов происходит при выполнении макрокоманды **Обновление**.

### **Обновление**

**ПоказатьВсеЗаписи** отменяет фильтры.

#### 5) Другие вопросы

- Создание пользовательских меню для формы

**ДобавитьМеню** используется для создания пользовательских меню по типу системного меню ACCESS.

- Вывод сведений на экран

**ВыводНаЭкран** включает или выключает вывод результатов макроса при его выполнении. Монопольные окна диалога (например, сообщения об ошибках) выводятся всегда.

### **ПесочныеЧасы**

**Сообщение** выводит заданное пользователем сообщение в специальном окне. Часто используется для приостановки выполнения макроса на время выдачи сообщения и реакции на него пользователя.

**УстановитьСообщения** используется для запрещения вывода системных сообщений, например подтверждений об удалении, изменении данных и пр.

- Генерация комбинаций клавиш

**КомандыКлавиатуры** используется для имитации нажатия клавиш. Полезна при создании демонстрационных и рекламных программных продуктов.

- Вывод на экран или удаление с экрана ПИ

### **ПанельИнструментов**

- Выдача звукового сигнала

### **Сигнал**

## **Групповые макросы**

Для решения однотипных задач, например, открытие/закрытие таблиц, форм или отчетов, а также исполнение запросов рекомендуется создавать групповые макросы. Каждому групповому макросу при его сохранении дается имя. Каждому простому макросу, входящему в состав группового, также дается имя. Ссылка на простой макрос внутри группового дается в виде:

<имя группового макроса>.<имя простого макроса>

### **Пример группового макроса**

Создать форму для просмотра двух таблиц **Поставщики** и **Детали**.

Для этого на форме создадим две кнопки так, чтобы при помощи одной из них открывалась одна таблица, а при помощи второй - другая таблица.

Последовательность действий:

- Создать форму без источника данных, выбрав способ создания формы **Конструктор**. Откроется пустой планшет формы только с областью данных.

- Добавить область заголовка и примечания формы: В области заголовка формы написать текст **Просмотр таблиц**.
- Создать (при выключенной кнопке **Мастера** на панели элементов конструктора форм) две кнопки, на одной из которых сделать надпись **Поставщики**, а на другой - **Детали**. Открыть окно свойств для первой кнопки и свойству *Имя элемента управления* (вкладка *Другие*) зададим значение **Поставщики**, а такому же свойству для второй кнопки - **Детали**.
- Создать групповой макрос для открытия таблиц **Поставщики** и **Детали**.

Имя макроса	Макрокоманда	Примечание
<b>Открыть Поставщики</b>	<b>ОткрытьТаблицу</b> (Имя таблицы: Поставщики, Режим данных: Только чтение)	Первый простой макрос, открывающий таблицу <b>Поставщики</b> только для чтения.
<b>Открыть Детали</b>	<b>ОткрытьТаблицу</b> (Имя таблицы: Детали, Режим данных: Только чтение)	Второй простой макрос, открывающий таблицу <b>Детали</b> только для чтения.

- Сохранить групповой макрос, дав ему имя **Открыть**
- Вернуться в окно формы, которая открыта в режиме конструктора форм.
- Связать кнопку с надписью **Поставщики** с простым макросом **Открыть Поставщики**, созданного внутри группового макроса по имени **Открыть**. Для этого:
  - ♦ открыть для кнопки **Поставщики** окно свойств
  - ♦ событию **Нажатие** кнопки (вкладка **События**) задать значение (выбрать из списка): **Открыть Поставщики**.
- Аналогично связать кнопку **Детали** с макросом **Открыть Детали**.
- Перейти в окно свойств формы и установить следующие значения свойствам макета:
  - Полосы прокрутки: отсутствуют
  - Область выделения: нет
  - Поле номера записи: нет
  - Разделительные линии: нет
- Сохранить форму, дав ей имя **Просмотр таблиц**
- Перейти в режим формы и проверить функционирование кнопок.

## Реализация ветвлений в макросах

Разветвляющиеся вычислительные процессы бывают двух видов: односторонние и двусторонние. В односторонних процессах вычисления определены только в одной ветви, а в двусторонних - в обеих ветвях. Соответственно в макросах с односторонним ветвлением определен набор макрокоманд только в одной ветви, а с двусторонним ветвлением - в обеих ветвях. Ветвление осуществляется в зависимости от значения некоторого условия, которое может быть истинно или ложно. Поэтому при конструировании макроса с ветвлением используется еще столбец условий для записи условия ветвления. Макросы, исполняемые в одной ветви, обозначаются троеточием в столбце условий.

### **Пример макроса с односторонним ветвлением**

Пусть при открытии таблицы **Детали** кнопкой **Детали** в форме **Просмотр таблиц**, рассмотренной ранее, будет запрашиваться пароль. Если введен правильный пароль, таблица будет открыта, если неправильный, то будет выдано сообщение на повторный ввод пароля. Для этого нужно выполнить следующую последовательность действий.

- Придумать пароль, например **фффф**.
- Создать форму (без источника данных) по имени **фпароль**, в которой будет запрашиваться пароль. Такая форма будет содержать надпись **Введите пароль** и единственный элемент управления **Поле**. Задайте в окне свойств для поля имя, например, **ппароль** (вкладка **Другие**). Задайте свойству **Маска ввода** (вкладка **Данные**) значение **Пароль** (введите из списка и, если средство не установлено, то напишите с помощью клавиатуры). В этом случае при вводе пароля вместо вводимых символов будет отображаться символ \*. Сохраните форму и дайте ей имя **пароль**.
- Отредактировать групповой макрос **Открыть** следующим образом:
  - ◆ открыть макрос **Открыть** в режиме конструктора
  - ◆ дописать строку, в которой в столбце **Имя макроса** написать **Открыть пароль**, а в столбце **Макрокоманда** выбрать из списка макрокоманду **ОткрытьФорму**, указав в аргументах имя открываемой формы **фпароль**
  - ◆ закрыть групповой макрос, сохранив результаты редакции.
- Связать событие **Нажатие** кнопки для кнопки **Детали** в форме **Просмотр таблиц** с макросом **Открыть.Открыть пароль**. Тогда при попытке открыть таблицу **Детали** сначала откроется форма **пароль**, которая запросит ввод пароля.
- Создать макрос по имени **мпароль**, который будет анализировать правильность ввода пароля:

Условие	Макрокоманда	Примечание
Not Forms![фпароль]![ппароль] =“фффф”	<b>Сообщение</b> (Сообщение: неверно, повторите ввод пароля)	
Forms![фпароль]![ппароль] =“фффф”	<b>ОткрытьТаблицу</b> (Имя таблицы: Детали, Режим данных: Только чтение)	Пароль введен верно. Выполняются все макро- команды с троеточием в столбце <b>Условие</b> .
	<b>СдвигРазмер</b> (От верхнего края: 4см)	
	<b>Сообщение</b> (Сообщение: ОК после просмотра)	Приостановка выполнения макроса для просмотра данных.
	<b>Закрыть</b> (Тип объекта: Форма, Имя объекта: фпароль)	
	<b>Закрыть</b> (Тип объекта: Таблица, Имя объекта: Детали)	

- \* Связать событие После обновления поля ппароль в форме **фпароль** с макросом **мпароль**. Закрывать форму **фпароль**, сохранив результаты редакции.
- \* Открыть форму **Просмотр таблиц** в режиме формы и убедиться в том, что таблица **Детали** открывается только при правильном вводе пароля.

### *Пример макроса с двусторонним ветвлением*

Требуется при просмотре деталей в форме **Детали** сделать так, чтобы на форме появлялась надпись **Дорогая**, если цена детали превышает, например, 200 руб. В остальных случаях надпись появляться не должна. Эту задачу можно решить следующим способом.

- Открыть форму **Детали** в режиме конструктора форм и создать на форме элемент управления **Надпись**, написать текст **Дорогая**. Сделать надпись невидимой (задать свойству **Вывод на экран** (вкладка **Макет**) значение **нет**
- Создать условный макрос по имени **Сигнал**:

Условие	Макрокоманда	Примечание
Forms![Детали]![цена]>200	<b>ЗадатьЗначение</b> (Элемент: Forms![Детали]![Дорогая].Visible Выражение: Истина)	
Not Forms![Детали]![цена]>200	<b>ЗадатьЗначение</b> (Элемент: Forms![Детали]![Дорогая].Visible Выражение: Ложь)	

Здесь **Visible** - свойство элемента управления **Надпись**. Это свойство можно найти в построителе выражений в окне свойств. Константы **Истина** и **Ложь** также можно найти в построителе выражений в папке **Константы**. Вместо этих констант можно использовать другие, допустимые значения для логических выражений (да, нет или -1,0).

- Связать событие **Текущая запись** формы **Детали** с макросом **Сигнал**.
- Открыть форму **Детали** в режиме формы и, переходя от записи к записи, убедиться в правильном решении поставленной задачи.

## Реализация циклов в макросах

Для реализации циклов в макросах используется макрокоманда **ЗапускМакроса**. Рассмотрим примеры.

### *Пример макроса для запуска сообщений*

Выдать на экран три раза одно и то же сообщение. Для решения этой задачи можно три раза использовать макрокоманду **Сообщение**, а можно это сделать с помощью цикла:

- Создать макрос, содержащий единственную макрокоманду **Сообщение**, в аргументах которой указать повторяемое сообщение. Сохранить макрос, задав ему имя, например, **Сообщить**.



- Создать другой макрос по имени **Запуск сообщения** следующего вида:

Макрокоманда	Примечание
<b>ЗапускМакроса</b> (Имя макроса: Сообщить, Число повторов: 3)	

- Исполнить макрос **Запуск сообщения**.

***Пример макроса для корректировки таблицы***

Поставщики из города Санкт-Петербург переехали с Самару. Выполнить соответствующую корректировку таблицы **Поставщики**.

Рассмотрим способ решения этой задачи с использованием циклов в макросах.

- Создать макрос **Замена**, который будет выполняться в цикле:

Макрокоманда	Примечание
<b>НайтиЗапись</b> (Образец поиска: СПб, Совпадение: Поля целиком, Область поиска: Вниз)	
<b>ЗадатьЗначение</b> (Элемент: Forms![Поставщики]![город], Выражение: "Самара")	

- Создать макрос **Переехал**:

Макрокоманда	Примечание
<b>ОткрытьФорму</b> (Имя формы: Поставщики)	
<b>КЭлементуУправления</b> (Имя элемента: город)	Перенос фокуса на поле город в форме <b>Поставщики</b> .
<b>ЗапускМакроса</b> (Имя макроса: Замена, Число повторов: =DCount("*";"Поставщики"))	
<b>Сообщение</b> (Сообщение: Готово)	Фиксирует окончание корректировки.
<b>Закрыть</b> (Тип объекта: Форма, Имя объекта: Поставщики)	
<b>ОткрытьТаблицу</b> (Имя таблицы: Поставщики)	Для контроля правильности корректировки.
<b>СдвигРазмер</b> (От левого края: 1 см., От верхнего края: 5 см.)	
<b>Закрыть</b> (Тип объекта: Таблица, Имя объекта: Поставщики)	

- Запустить макрос на выполнение и убедиться в правильности решения поставленной задачи.

Макросы (за исключением встроенных макросов) могут быть преобразованы в код Visual Basic для приложений (VBA) с помощью пиктограммы *Преобразовать макросы* в группе *Макрос* на вкладке РАБОТА С БАЗОЙ ДАННЫХ.

## **2. ЗАДАНИЕ ПО РАБОТЕ**

Получить у преподавателя вариант заданий к лабораторной работе №5. **Индивидуальные задания находятся в разделе «Варианты заданий для лабораторных работ».**

Выбрать необходимые макрокоманды и создать макросы.

## **3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

1. Подготовить отчет по работе.
2. Создать макросы в соответствии с определенным преподавателем вариантом и предъявить их преподавателю.

## **4. СОДЕРЖАНИЕ ОТЧЕТА**

1. Титульный лист.
2. Название и цель выполнения работы.
3. Виды разработанных макросов и результаты их выполнения.

### **Контрольные вопросы для самостоятельной проработки**

1. В каких случаях используются макросы?
2. Как привязать выполнение макроса к одному из событий формы?
3. Какие существуют способы создания макросов?
4. Как привязать выполнение макроса к элементу управления "Кнопка"?
5. Что такое макрокоманда? Как определяется последовательность выполнения макрокоманд?
6. Что такое аргументы макрокоманды? Где они задаются?
7. Где определяются условия выполнения макрокоманд?
8. Каким образом объединить макросы в одну группу?
9. Как осуществляется автоматизация вычислений с помощью макросов?
10. Как осуществляется замена команд меню с помощью макросов?
11. Как создать сообщение, используя макросы?
12. Где осуществляется пошаговое выполнение макроса?
13. Какие функции выполняет макрос AutoExec?

## **ВАРИАНТЫ ЗАДАНИЙ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ**

### **ВАРИАНТ 1**

#### **Информационная система “Книжный магазин”**

##### **Лабораторная работа № 1**

Вы проектируете информационную систему для магазина самообслуживания, который занимается продажей книг и иной печатной продукции (газеты, журналы, календари, настольные игры). Продукция в магазин поступает как от отдельных фирм (перекупщики, распространители), так и напрямую из издательств. Со всеми поставщиками заключены договора (на определенный срок). Проектируемая информационная система должна вести учет: сотрудников, поставщиков и договоров, поступившей и проданной продукции. Продукция поступает партиями, согласно договорам. Каждый договор имеет уникальный номер. С одним поставщиком может быть заключено несколько договоров, в одном договоре может быть несколько наименований товаров (т.е. договор имеет спецификацию договора, в которой указывается перечень поставляемых товаров). Ведется учет как поступившей, так и проданной продукции. Сотрудники (продавцы и консультанты) работают посменно, причем производится разделение, в каком отделе работает сотрудник, продукция также продается по разделам (компьютерная литература, детективы и т.д.). Один сотрудник в разные дни может работать в разных отделах.

Ограничения:

- на реализацию принимается продукция выпуска не позже позапрошлого года (от текущего).
- работать в магазине могут только совершеннолетние сотрудники (старше 18 лет (возраст с учетом до дня)).
- стоимость продукции не может быть отрицательной.

##### **Лабораторная работа №2**

#### **Вариант 1.1**

1. Получить информацию о поставщике продукции по номеру договора (номер договора - параметр).
2. Получить информацию о поставщиках - издателях, которые не поставляли ни одной игры.
3. Подсчитать количество сотрудников, работавших в заданный день (дата - параметр), в заданном отделе (отдел - параметр).
4. Добавить информацию о новом товаре.
5. Удалить информацию о договоре (номер договора - параметр), если он еще не выполнен.

#### **Вариант 1.2**

1. Получить информацию о том, какие сотрудники работают в определенный день (дата - параметр запроса).
2. Получить информацию о сотрудниках – продавцах, продавших хотя бы один календарь.
3. Подсчитать стоимость продукции, проданной за последнюю неделю (считая от текущей даты).
4. Увеличить стоимость заданной продукции (номер продукции - параметр), увеличив ее на 10 процентов.

5. Удалить информацию о сотруднике (номер сотрудника - параметр), если он в дальнейшем (начиная с текущей даты) не указан в расписании работы сотрудников.

### **Вариант 1.3**

1. Найти, какая продукция была продана в указанный день (дата – параметр запроса).
2. Получить информацию о всей проданной продукции типа - журнал.
3. Подсчитать стоимость продукции, которую поставил определенный поставщик (номер поставщика - параметр) на текущую дату.
4. Добавить информацию о новом сотруднике.
5. Удалить информацию о заданном поставщике, если он не поставлял продукцию.

## **Лабораторная работа №3**

**Во всех разработанных формах удалить системные кнопки и кнопки перехода по записям, заменить их на свои кнопки. Каждая разработанная форма (если она не подчиненная) должна иметь заголовок, отражающий назначение формы.**

### **Вариант 1.1**

1. Разработать форму для просмотра и ввода данных о поставщиках товара.
2. Разработать форму для просмотра и ввода данных о договорах.
3. Разработать форму для спецификации договора и сделать ее подчиненной для формы из пункта 2. Добавить поле в котором подсчитывается стоимость товаров по договору. Поле “Название товара” сделать полем со списком
4. Добавить в форму из пункта 1 кнопку, при нажатии на которую открывалась бы форма из пункта 3 (только для выбранных записей).
5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Вариант 1.2**

1. Разработать форму для просмотра и ввода данных о сотрудниках фирмы. Поле должность сотрудника сделать полем со списком.
2. Разработать форму для просмотра и ввода данных о расписании работы сотрудников. Поле Отдел в котором работает сотрудник сделать полем со списком. Поле ФИО сотрудника сделать полем со списком.
3. Разработать форму для просмотра и ввода данных о проданной продукции. Добавить поле, в котором подсчитывается стоимость проданной продукции.
4. Сделать форму из пункта 1 подчиненной для формы из пункта 2. Добавить в полученную форму (основную) кнопку, при нажатии на которую, открывается форма из пункта 3 для выбранной записи.
5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Вариант 1.3**

1. Разработать форму для просмотра и ввода данных о товарах. Поле год выпуска сделать полем со списком. Поле тип продукции сделать полем со списком. Поле Отдел (к какому отделу относится товар) сделать полем со списком.

2. Разработать форму договор и подчиненную к ней форму спецификация договора.

3. Добавить в форму из пункта 2 (подчиненную) кнопку при нажатии на которую, открывается форма из пункта 1 для указанных записей

4. Добавить в форму из пункта 3 поле, в котором подсчитывается стоимость товара по выбранному договору.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

#### **Лабораторная работа №4**

##### **Вариант 1.1**

1. Создать отчет о заданном поставщике.

2. Создать отчет, отображающий информацию о договорах с подчиненным отчетом спецификация договора. В отчете должна подсчитываться общая стоимость товаров по договору.

3. Создать отчет отображающий информацию о заданном поставщике и договорах, заключенным с ним, сделав отчет из пункта 2 подчиненным отчету из пункта 1. В отчете должна подсчитываться общая стоимость всех договоров для данного поставщика.

##### **Вариант 1.2**

1. Создать отчет о расписании работы заданного сотрудника на текущий месяц.

2. Создать отчет о сотрудниках.

3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1. В отчете должно подсчитываться количество дней, которое сотрудник должен отработать за текущий месяц.

##### **Вариант 1.3**

1. Создать отчет о сотрудниках.

2. Создать отчет о расписании работы определенного сотрудника.

3. Создать отчет о проданных сотрудниками товарах.

4. Сделать отчет из пункта 3 подчиненным отчету из пункта 2.

5. Сделать отчет из пункта 4 подчиненным отчету из пункта 1.

6. В получившемся отчете должна быть информация: стоимость проданной продукции сотрудником за день; общая стоимость проданной сотрудником продукции.

#### **Лабораторная работа №5**

##### **Вариант 1.1**

1. При помощи макроса показать список сотрудников – продавцов, продавших за месяц товаров на сумму, меньшую заданной.

2. Вывести данные объекта БД из пункта 1 в формате MS Excel.

3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

##### **Вариант 1.2**

1. При помощи макроса показать продукцию, которуюставляет поставщик с заданным номером.

2. С помощью макроса произвести переименование одного из отделов книжного магазина.

3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

### **Вариант 1.3**

1. При помощи макроса показать присутствующий в магазине товар (поступил и не все продали) с указанием количества товара в наличии.

2. Вывести данные объекта БД из пункта 1 в формате RTF.

3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

## **ВАРИАНТ 2**

### **Информационная система «Курсы иностранных языков»**

#### **Лабораторная работа №1**

Вы проектируете информационную систему для курсов иностранных языков. Данные курсы предоставляют желающим услуги по обучению нескольким иностранным языкам. Процесс обучения иностранному языку разбит на несколько этапов (уровней), для каждого языка свое количество. На каждом уровне проводится определенное количество контрольных работ. По окончании каждого уровня проводится экзамен с выставлением оценки и выдачей свидетельства о сдаче экзамена по уровню. Услуги предоставляются платно. Каждый уровень имеет свою стоимость. Оплата обучения может быть произведена как за наличный, так и за безналичный расчет. Оплата обучения может быть произведена по частям. Обучение желающих производится в группах. Одновременно один обучаемый может изучать несколько иностранных языков, но изучать одновременно несколько уровней одного языка он не может. С одной группой занимается только один преподаватель, закрепленный за данной группой. Один преподаватель может вести несколько групп (не обязательно одного языка). Занятия проводятся в учебных классах. Длительность занятий в разных группах разная.

Реализовать ограничения: один уровень длится ровно 3 месяца; обучаемые должны быть старше 10 лет.

#### **Лабораторная работа №2**

### **Вариант 2.1**

1. Получить расписание занятий для конкретной группы (номер группы – параметр запроса).

2. Найти всех преподавателей способных преподавать английский язык.

3. Найти слушателей, которые заплатили за обучение, но общая сумма их оплаты меньше стоимости обучения по курсу на момент начала занятий (до дня начала занятий включительно).

4. Добавить в группу нового слушателя.

5. Удалить информацию о слушателе, удаление разрешить только в случае, если он прослушал полный курс и получил оценку за экзамен (ФИО слушателя – параметр).

### **Вариант 2.2**

1. Получить расписание занятий для конкретного преподавателя (ФИО преподавателя – параметр запроса).
2. Найти всех слушателей, когда-либо занимавшихся немецким языком.
3. Найти слушателей обучавшихся одному (и только одному) языку на нескольких различных уровнях.
4. Если студент уже обучался ранее на курсах этому языку, то сумму его оплаты за обучение уменьшить на 10 процентов.
5. Удалить информацию о преподавателе, удаление разрешить только в случае, если преподаватель не ведет занятий и больше ему занятий уже не назначено.

### **Вариант 2.3**

1. Найти перечень всех групп, когда-либо занимавшихся заданным языком (язык – параметр запроса).
2. Найти всех слушателей, не оплативших занятия.
3. Подсчитать количество студентов в каждой группе на день начала занятий оплативших свое обучение полностью (т. е. общая сумма оплаты больше или равна плате за обучение) и сумму переплаты либо недоплаты группой занятий (разница между суммарной стоимостью курса и оплатой всей группы) и общее количество студентов в группе.
4. Поменять одного преподавателя у группы на другого, назначение разрешить лишь в случае, если это время у преподавателя свободно.
5. Удалить информацию о группе из расписания занятий в случае, если группа окончила занятия.

### **Лабораторная работа №3**

**Во всех разработанных формах удалить системные кнопки и кнопки перехода по записям, заменить их на свои кнопки. Каждая разработанная форма (если она не подчиненная) должна иметь заголовок, отражающий назначение формы.**

#### **Вариант 2.1**

1. Разработать форму для просмотра и ввода данных о слушателе.
2. Разработать форму для просмотра и ввода данных о группах. Поле ФИО слушателя сделать полем со списком. Поле вид оплаты (наличный или безналичный расчет) сделать полем со списком.
3. Разработать форму для просмотра и ввода данных о сдаче слушателем контрольных работ и экзамена. Поле Вид контроля знаний (КР1, КР2, ..., экзамен) сделать полем со списком.
4. Сделать форму из пункта 1 основной, а форму из пункта 3 подчиненной, добавить в получившуюся форму кнопку, при нажатии на которую открывается форма из пункта 2 для выбранной записи. Добавить в форму из пункта 2 поле, в котором бы вычислялась суммарная оплата слушателем курса и поле стоимость курса.
5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

#### **Вариант 2.2**

1. Разработать форму для просмотра и ввода данных о преподавателе. Поле язык сделать полем со списком.

2. Разработать форму для просмотра и ввода данных о расписании занятий. Поля ФИО преподавателя, номер группы, язык преподавания, день недели, номер класса сделать полями со списком.

3. Создать форму для просмотра и ввода данных о группе. Поле язык, уровень и длительность занятий сделать полем со списком.

4. Добавить в форму из пункта 2 кнопку, при нажатии на которую, отрывалась бы форма из пункта 3 для выбранной записи. Сделать полученную форму подчиненной форме из пункта 1.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Вариант 2.3**

1. Разработать форму для просмотра и ввода данных о слушателях.

2. Разработать форму для просмотра и ввода расписания занятий. Поля ФИО преподавателя, номер группы, язык преподавания, день недели, номер класса сделать полями со списком.

3. Разработать форму для просмотра и ввода данных о преподавателях. Поле язык сделать полем со списком.

4. Сделать форму из пунктов 1 и 3 подчиненными форме из пункта 2, добавить поле, в котором подсчитывается количество студентов в группе.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

## **Лабораторная работа №4**

### **Вариант 2.1**

1. Создать отчет, в котором отображается расписание занятий текущих групп (группы в которых занятия уже начались, но еще не закончились).

2. Создать отчет об успеваемости заданного студента (ФИО студента - параметр).

3. Добавить в отчет из пункта 2 информацию об оплате данным слушателем курса (стоимость курса и суммарная оплата слушателем).

### **Вариант 2.2**

1. Создать отчет о заданном преподавателе (ФИО преподавателя - параметр).

2. Создать отчет о расписании занятий. Упорядочить информацию по дням недели и по времени начала занятий.

3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1, отчет должен иметь следующий вид: сначала идет общая информация о преподавателе, затем идет список языков, которые данный преподаватель может преподавать, а уже для каждого языка добавляется подчиненный отчет о расписании занятий для этого языка.

### **Вариант 2.3**

1. Создать отчет, в котором отображается информация о слушателях. Упорядочить информацию по ФИО студентов (по алфавиту).

2. Создать отчет о группах. Упорядочить информацию по возрастанию номеров групп.

3. Создать отчет о группах и слушателях в группах, но только о тех группах и тех слушателях, которые одновременно изучают несколько языков. (Основной отчет о слушателях, подчиненный – о группах).



## **Лабораторная работа №5**

### **Вариант 2.1**

1. При помощи макроса показать список слушателей в группах, которые одновременно изучают несколько языков.
2. Вывести данные объекта БД из пункта 1 в формате PDF.
3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

### **Вариант 2.2**

1. При помощи макроса вывести информацию о преподавателях конкретного языка с указанием расписания их занятий
2. Осуществить замену преподавателя у группы с помощью макроса.
3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

### **Вариант 2.3**

1. При помощи макроса показать список всех студентов и вывести предупреждение в случае неполной оплаты студентами курса обучения.
2. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных.
3. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

## **ВАРИАНТ 3**

### **Информационная система «Приемная комиссия»**

#### **Лабораторная работа № 1**

Вы проектируете информационную систему для приемной комиссии университета. Абитуриенты, поступающие в университет, представляют информацию о себе (заполняют заявление) и указывают факультет и специальность на которой они хотят обучаться. Один абитуриент имеет право одновременно поступать только на одну специальность. На каждую специальность есть свой набор вступительных экзаменов. Абитуриенты для сдачи экзаменов объединяются в группы. В одной группе могут находиться абитуриенты, поступающие на различные специальности, но при этом сдающие одинаковые экзамены. Один преподаватель может принимать экзамены по различным предметам. Один экзамен могут принимать несколько преподавателей.

#### **Лабораторная работа №2**

### **Вариант 3.1**

1. Найти абитуриентов, поступающих на заданный факультет (номер факультета – параметр запроса).
2. Найти все специальности (с указанием кафедры и факультета) для которых вступительным экзаменом является математика.
3. Для заданной специальности подсчитать среднюю оценку за каждый экзамен.
4. Назначить преподавателя на прием экзамена, при этом, если на этот экзамен еще не было назначено ни одного преподавателя, то не добавлять.

5. Удалить заданного преподавателя (ФИО – параметр запроса), если он не назначен ни на один экзамен.

### **Вариант 3.2**

1. Найти дисциплины, которые должны сдавать в качестве экзаменов абитуриенты, поступающие на заданную специальность (номер специальности – параметр запроса).

2. Найти всех абитуриентов, поступающих на специальность 2201.

3. Найти все группы абитуриентов, сдающие больше 3-х экзаменов, при этом каждый экзамен принимает комиссия из 3-х преподавателей.

4. Поменять местами (поменять даты) два экзамена у группы, но замену можно разрешить только в случае, если оба экзамена еще не пошли.

5. Удалить экзамен, если он еще не был проведен.

### **Вариант 3.3**

1. Найти преподавателей, принимающих экзамены у заданной группы абитуриентов (номер группы – параметр запроса).

2. Найти всех преподавателей, которые могут принимать экзамен по экономике.

3. Найти все группы, в которых сдают экзамены абитуриенты, поступающие на разные специальности, при этом абитуриенты, поступающие на определенную специальность находятся также в нескольких разных группах.

4. Перевести абитуриента на другую специальность, но перевод разрешить только в случае если у специальностей вступительные экзамены совпадают.

5. Удалить данные о специальности, если на нее не было подано ни одного заявления.

## **Лабораторная работа №3**

**Во всех разработанных формах удалить системные кнопки и кнопки перехода по записям, заменить их на свои кнопки. Каждая разработанная форма (если она не подчиненная) должна иметь заголовок, отражающий назначение формы.**

### **Вариант 3.1**

1. Разработать форму для просмотра и ввода данных о заданном абитуриенте. Поля номер специальности и номер факультета сделать полями со списком.

2. Разработать форму для просмотра и ввода данных о расписании экзаменов. Поля дисциплина, преподаватель, аудитория, группа сделать полями со списком.

3. Разработать форму для просмотра и ввода данных о сданных абитуриентом экзаменах. Добавить в форму поля, содержащие средние оценки по данным экзаменам среди абитуриентов, поступающих на эту же специальность.

4. Сделать форму из пункта 3 подчиненной форме из пункта 1. Добавить в получившуюся форму кнопку при нажатии на которую открывается форма из пункта 2 для указанной записи.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Вариант 3.2**

1. Разработать форму для просмотра и ввода данных о преподавателях.

2. Разработать форму для просмотра и ввода данных о расписании экзаменов. Поля преподаватель, аудитория, группа, дисциплина сделать полями со списком.

3. Разработать форму для просмотра и ввода данных об оценках за экзамен поставленных заданным преподавателем. Добавить поле, содержащее информацию о средней оценке за экзамен по дисциплине, поставленной данным преподавателем.

4. Сделать форму из пункта 2 подчиненной форме из пункта 1, добавить в получившуюся форму кнопку при нажатии на которую открывается форма из пункта 3 для указанной записи.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Вариант 3.3**

1. Разработать форму для просмотра и ввода данных о расписании экзаменов. Поля дисциплина, преподаватель. Группа, аудитория сделать полями со списком.

2. Разработать форму для просмотра и ввода данных о группах абитуриентов.

3. Разработать форму для просмотра и ввода данных о специальностях и дисциплинах, по которым должны сдаваться вступительные экзамены.

4. сделать форму из пункта 1 подчиненной форме из пункта 2, добавить в получившуюся форму кнопку, при нажатии на которую открывается форма из пункта 3. В получившуюся форму добавить поле в котором подсчитывается количество вступительных экзаменов у выбранной группы.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

## **Лабораторная работа №4**

### **Вариант 3.1**

1. Создать отчет о группах абитуриентов. Добавить информацию о количестве абитуриентов в группе. Добавить информацию о количестве абитуриентов определенной специальности в группе.

2. Создать отчет об абитуриентах.

3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1. Упорядочить информацию по возрастанию номеров групп, внутри группы по номеру специальности, а для отдельной специальности по ФИО абитуриента.

### **Вариант 3.2**

1. Создать отчет о преподавателях. Упорядочить информацию по ФИО преподавателя.

2. Создать отчет о дисциплинах. Упорядочить информацию по названию дисциплины.

3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1. Добавить информацию о количестве дисциплин, по которым преподаватель может принимать экзамен.

### **Вариант 3.3**

1. Создать отчет об оценках за экзамены. Упорядочить информацию по ФИО студента и по названию дисциплины. Добавить информацию о средних баллах за экзамен по специальности.

2. Создать отчет об абитуриентах. Упорядочить информацию по номеру специальности, а затем по ФИО студента.

3. Сделать отчет из пункта 1 подчиненным отчету из пункта 2. Добавить информацию о среднем балле студента за экзамены и об общем среднем балле по специальности (средний балл – средняя оценка за вступительные экзамены).

### **Лабораторная работа №5**

#### **Вариант 3.1**

1. При помощи макроса вывести информацию о преподавателях с указанием расписания экзаменов и количества дисциплин, по которым преподаватель может принимать экзамен.

2. Осуществить замену преподавателя на экзамене с помощью макроса.

3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

#### **Вариант 3.2**

1. С помощью макроса представить список абитуриентов по заданной специальности, сдавших все экзамены на положительные оценки. Добавить информацию о среднем балле студента за экзамены.

2. Вывести данные объекта БД из пункта 1 в формате MS Excel.

3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

#### **Вариант 3.3**

1. При помощи макроса привести список дисциплин, которые должны сдавать в качестве экзаменов абитуриенты, поступающие на заданную специальность, и общий средний балл по специальности (средний балл – средняя оценка за вступительные экзамены).

2. Перевести абитуриента на другую специальность с помощью макроса.

3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

## **ВАРИАНТ 4**

### **Информационная система «Домашняя бухгалтерия»**

#### **Лабораторная работа № 1**

Вы разрабатываете информационную систему для домашней бухгалтерии. Домашняя бухгалтерия предназначена для учета личных денежных средств и семейного бюджета. Информационная система ориентирована на пользователей, не имеющих специальной бухгалтерской подготовки, и позволяет учитывать доходы / расходы, вести учет денег взятых / отданных в долг, обеспечивает контроль возврата долгов, планирование предстоящих доходов / расходов. Учет можно вести в нескольких валютах (по выбору). Домашнюю бухгалтерию могут использовать несколько пользователей – все записи ведутся независимо друг от друга. Для удобства все расходы разделены по областям, например транспорт, питание и т.д., а уже затем в соответствующей области выделены пункты.

## Лабораторная работа №2

### Вариант 4.1

1. Найти все статьи расходов в заданной области расходов (область расходов – параметр запроса).
2. Найти все пункты расходов на транспорт у заданного пользователя (имя пользователя - параметр запроса).
3. Подсчитать доход каждого пользователя за текущий месяц, но только в том случае, если пользователь на текущий момент в этом месяце уже получил зарплату.
4. Зарегистрировать нового пользователя, но только в том случае, если еще не было пользователя с таким именем.
5. Удалить область расходов, удаление разрешить только в случае, если данной областью расходов никто не пользовался.

### Вариант 4.2

1. Найти все доходы заданного пользователя (имя пользователя – параметр запроса).
2. Получить информацию об области дохода – зарплата у заданного пользователя (имя пользователя параметр запроса).
3. Подсчитать расходы каждого пользователя по областям за последний месяц.
4. Все расходы в долларах перевести в рубли, т.е. заменить название валюты и изменить сумму расхода согласно введенному курсу (курс валюты – параметр запроса).
5. Удалить информацию о доходах заданного пользователя за текущий месяц (имя пользователя – параметр запроса).

### Вариант 4.3

1. Найти все планируемые расходы заданного пользователя (имя пользователя – параметр запроса).
2. Получить информацию о всех расходах заданного пользователя за текущий месяц (имя пользователя – параметр запроса).
3. Подсчитать расходы и доходы каждого пользователя, а также взятые и отданные в долг суммы за последние 7 дней.
4. Выдать указному знакомому в долг указанным пользователем указанную сумму (параметры запроса), при этом выдачу в долг разрешить, только если должник вернул предыдущий долг, который он брал (если брал).
5. Удалить информацию о выданных в долг средствах заданному должнику заданным пользователем (параметры запроса).

## Лабораторная работа №3

**Во всех разработанных формах удалить системные кнопки и кнопки перехода по записям, заменить их на свои кнопки. Каждая разработанная форма (если она не подчиненная) должна иметь заголовок, отражающий назначение формы.**

### Вариант 4.1

1. Разработать форму для просмотра и ввода данных о пользователях информационной системы.
2. Разработать составную форму для просмотра и ввода данных об областях и статьях расхода по областям. Поля область расхода и статья расхода в области сделать полями со списком.
3. Разработать форму для просмотра и ввода данных о расходах пользователей системы. Поля область расхода и статья расхода сделать полями со

списком. При этом после выбора области расхода, в поле статья расхода должны отображаться только пункты, соответствующие данной области.

4. Сделать форму из пункта 1 основной, а форму из пункта 3 подчиненной, добавить в получившуюся форму кнопку, при нажатии на которую открывается форма из пункта 2. Добавить поле, в котором подсчитывается общая сумма расходов за последний месяц для указанного пользователя.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

#### **Вариант 4.2**

1. Разработать форму для просмотра и ввода данных об указанном пользователе системы (пользователь - параметр).

2. Разработать форму для просмотра и ввода данных о расходах пользователей. Поле область расходов сделать полем со списком, поле статья расхода сделать полем со списком, при этом после выбора области расхода, в поле статья расхода должны отображаться только пункты, соответствующие данной области.

3. Разработать форму для просмотра и ввода данных о доходах пользователей.

4. Сделать форму из пункта 1 основной, а формы из пунктов 2 и 3 подчиненными, добавить поля общая сумма расходов, сумма расходов за текущий месяц, общая сумма доходов, сумма доходов за текущий месяц.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

#### **Вариант 4.3**

1. Разработать форму для просмотра и ввода данных о должнике (параметр).

2. Разработать форму для просмотра и ввода данных о выдаче денежных средств пользователями в долг.

3. Разработать форму для просмотра и ввода данных о возврате должниками денежных средств.

4. Сделать форму из пункта 1 основной, а формы из пунктов 2 и 3 подчиненными. Добавить в получившуюся форму поля общая сумма долга, общая сумма возвращенных долгов.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Лабораторная работа №4**

#### **Вариант 4.1**

1. Создать отчет, содержащий информацию о пользователях, дававших в долг.

2. Создать отчет о долгах, упорядочить информацию по должникам (по алфавиту).

3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1. Добавить информацию о сумме долга каждого должника, об общей сумме выданной в долг данным пользователем и об общей сумме выданной в долг всеми пользователями.

#### **Вариант 4.2**

1. Создать отчет, содержащий информацию о пользователях системы.
2. Создать отчет о расходах по областям за последний месяц, отчет должен содержать информацию об общей сумме расходов за последний месяц. Упорядочить информацию по областям расходов (по алфавиту).
3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1, добавить информацию об общей сумме расходов за последний месяц всех пользователей системы.

#### **Вариант 4.3**

1. Создать отчет, содержащий информацию о заданном пользователе системы (параметр).
2. Создать отчет о расходах. Добавить информацию об общей сумме расходов.
3. Создать отчет о доходах, добавить информацию об общей сумме доходов.
4. Сделать отчет из пункта 1 основным, а отчеты из пунктов 2 и 3 - подчиненными. Добавить информацию о разнице между доходами и расходами пользователя.

### **Лабораторная работа №5**

#### **Вариант 4.1**

1. С помощью макроса вывести суммарные расходы и доходы каждого пользователя, а также взятые и отданные в долг суммы, и вывести предупреждение в случае превышения расходов над доходами.
2. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных.
3. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

#### **Вариант 4.2**

1. С помощью макроса перевести все расходы только в одну валюту и представить общую сумму всех расходов согласно курсу этой валюты.
2. Вывести данные объекта БД из пункта 1 в формате PDF.
3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

#### **Вариант 4.3**

1. При помощи макроса представить план расходов пользователя по областям на следующий месяц, исходя из расходов по предыдущим месяцам (полным), планируемые суммы вычислять как среднее по предыдущим месяцам.
2. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных.
3. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

## **ВАРИАНТ 5**

### **Информационная система «Домашняя библиотека»**

#### **Лабораторная работа № 1**

Вы разрабатываете информационную систему для домашней библиотеки компьютерных дисков и книг. В вашей библиотеке присутствуют как компьютерные диски различных жанров типов и форматов (CD и DVD, mp3 и mp4, диски с программами, энциклопедии и т. д.), так и книги (учебная и художественная литература). У вас есть знакомые у которых вы можете взять какие-нибудь диски или книги займы, также вы сами можете дать им в долг часть своей библиотеки. Разрабатываемая информационная система должна отслеживать перемещение вашей собственности, а так же отслеживать ваши долги. Хранится информация не только о названии диска или книги, но и более подробная информация (например, если это фильм, то в каком формате, на каком диске, какой жанр фильма, длительность фильма, какие известные актеры там снимались...).

#### **Лабораторная работа №2**

##### **Вариант 5.1**

1. Найти все диски заданного жанра (жанр – параметр запроса).
2. Получить информацию о всех мультфильмах записанных на DVD.
3. Найти все диски и книги, которые вы взяли в долг, и их срок возврата уже прошел, а вы их не вернули.
4. Зарегистрировать выдачу диска в долг.
5. Удалить информацию о знакомом, удаление возможно только в случае если ни вы ему, ни он вам ничего не должны.

##### **Вариант 5.2**

1. Найти все книги заданного жанра (жанр – параметр запроса).
2. Получить информацию о всех детективах изданных после 1998 года.
3. Найти все жанры фильмов, все диски которых на данный момент находятся у вас дома (никому не отданы в долг).
4. Зарегистрировать возврат всех дисков взятых у вас в долг определенным знакомым (параметр запроса).
5. Удалить информацию о диске, но только в случае если это ваш диск.

##### **Вариант 5.3**

1. Получить информацию о знакомом (ФИО знакомого – параметр запроса).
2. Найти всех знакомых, кто взял у вас книги в долг и еще не вернул.
3. Подсчитать количество дисков по жанрам и типам, находящихся на данный момент у вас дома, но считать только свои диски, диски, отданные в долг также не считать.
4. Зарегистрировать взятие вами книги в долг у знакомого.
5. Удалить информацию о книге, но только если она находится у вас дома и она ваша.

#### **Лабораторная работа №3**

**Во всех разработанных формах удалить системные кнопки и кнопки перехода по записям, заменить их на свои кнопки. Каждая разработанная форма (если она не подчиненная) должна иметь заголовок, отражающий назначение формы.**

##### **Вариант 5.1**

1. Разработать форму для просмотра и ввода данных о дисках. Поля тип и жанр сделать полями со списком.



2. Разработать форму для просмотра и ввода данных о знакомых, которым одалживаем диски.

3. Разработать форму для просмотра и ввода данных о выдаче дисков в долг. Поле ФИО знакомого сделать полем со списком.

4. Сделать форму из пункта 1 основной, а форму из пункта 3 – подчиненной, добавить в получившуюся форму кнопку при нажатии на которую открывается форма из пункта 2 для отобранных записей. Добавить поле в котором подсчитывается количество дисков, выданных в долг выбранному знакомому, которые он еще не вернул.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Вариант 5.2**

1. Разработать форму для просмотра и ввода данных о книгах. Поле жанр сделать полем со списком.

2. Разработать форму для просмотра и ввода данных о знакомых, которые дают книги в долг.

3. Разработать форму для просмотра и ввода данных о взятии и возврате книг владельцам. Поле ФИО знакомого сделать полем со списком.

4. Сделать форму из пункта 3 подчиненной форме из пункта 1, в получившуюся форму добавить кнопку при нажатии на которую открывается форма из пункта 2. Добавить поле, в котором подсчитывается количество книг, которые мы взяли в долг, но еще не вернули у данного владельца.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Вариант 5.3**

1. Разработать форму для просмотра и ввода данных о дисках, содержащих программы.

2. Разработать форму для просмотра и ввода данных о выдаче дисков в долг и их возврате.

3. Разработать форму для просмотра и ввода данных о взятии и возврате дисков у знакомых.

4. Сделать форму из пункта 1 основной, а формы из пунктов 2 и 3 подчиненными. Добавить поля, в которых подсчитывается количество дисков с программами отданных в долг (их еще не вернули) и количество дисков с программами взятых в долг (которые мы еще не отдали владельцам).

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

## **Лабораторная работа №4**

### **Вариант 5.1**

1. Разработать отчет о знакомых, взявших диски в долг и не вернувших их в срок.

2. Разработать отчет, содержащий информацию о дисках.

3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1, добавить информацию о количестве у знакомого дисков на руках (взятых и не возвращенных в срок).

#### **Вариант 5.2**

1. Разработать отчет, содержащий информацию о движении дисков по знакомым за последний месяц.

2. Разработать отчет о дисках.

3. Сделать отчет из пункта 1 подчиненным отчету из пункта 2. В отчете должна содержаться об общем количестве дисков.

#### **Вариант 5.3**

1. Разработать отчет, содержащий информацию о книгах, которые мы взяли в долг.

2. Разработать информацию о знакомых, которые давали нам книги в долг.

3. Сделать отчет из пункта 1 подчиненным отчету из пункта 2. Добавить информацию об общем количестве книг взятых в долг и не возвращенных владельцам.

### **Лабораторная работа №5**

#### **Вариант 5.1**

1. При помощи макроса показать список всех книг и дисков с указанием их общего количества, находящихся в настоящий момент у конкретного знакомого.

2. Осуществить замену формата у дисков определенного жанра с помощью макроса.

3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

#### **Вариант 5.2**

1. При помощи макроса найти всех знакомых, кто взял у вас книги в долг. Вывести сообщение в случае, если срок возврата книг уже прошел, а их не вернули.

2. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных.

3. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

#### **Вариант 5.3**

1. Создать макрос, с помощью которого можно подсчитывать статистическую информацию: сколько всего книг и дисков имеется, сколько из них выдано, сколько должны уже были вернуть, но не вернули.

2. Вывести данные объекта БД из пункта 1 в формате MS Excel.

3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных.

### **ВАРИАНТ 6**

#### **Информационная система «Гостиница»**

#### **Лабораторная работа № 1**

Вы разрабатываете информационную систему для гостиницы обеспечивающую выполнение задач учета и планирования загрузки помещений. Также система должна вести учет персонала, расписание его работы. Должна существовать возможность бронирования мет, однако бронирование осуществляется с предоплатой и не более

чем за 30 дней. Оплата проживания и предоплата может осуществляться как за наличный, так и за безналичный расчет.

### **Лабораторная работа №2**

#### **Вариант 6.1**

1. Получить информацию о постояльцах, проживающих на данный момент времени в определенном номере (номер – параметр запроса).
2. Получить информацию о горничных, работающих в гостинице.
3. Найти всех постояльцев, проживавших в гостинице больше одного раза в одном и том же номере.
4. Изменить стоимость номера (увеличив на 10 процентов), учитывая это изменение изменить все счета выставляемые клиентам, начиная с текущей даты (номер – параметр запроса).
5. Удалить информацию о постояльце, данные можно удалить только в случае, если постоялец уже съехал и оплатил свое проживание.

#### **Вариант 6.2**

1. Получить информацию о персонале, обслуживающем определенный номер (номер – параметр запроса).
2. Получить информацию о администраторах, работающих в гостинице.
3. Найти всех постояльцев, которые раньше уже проживали в гостинице, а сейчас забронировали номер и при этом все время один и тот же.
4. Забронировать одноместный номер в гостинице (номер можно забронировать, только если он свободен).
5. Удалить информацию о брони номера. Информацию можно удалить только в случае если бронь номера еще не была оплачена.

#### **Вариант 6.3**

1. Получить информацию о постояльце (ФИО - параметр запроса.): когда в каких номер и как долго он проживал.
2. Получить информацию о постояльцах, оплативших свое пребывание по безналичному расчету.
3. Подсчитать стоимость проживания заданного постояльца (ФИО – параметр запроса), длительность проживания вычислять в календарных сутках.
4. Перевести все номера с двух горничных на одну (ФИО горничных – параметр запроса).
5. Удалить данные о номере, данные можно удалить только в случае если сейчас в нем никто не проживает и номер не забронирован на будущее.

### **Лабораторная работа №3**

**Во всех разработанных формах удалить системные кнопки и кнопки перехода по записям, заменить их на свои кнопки. Каждая разработанная форма (если она не подчиненная) должна иметь заголовок, отражающий назначение формы.**

#### **Вариант 6.1**

1. Разработать форму для просмотра и ввода данных о постояльце. Поле пол сделать полем со списком. Поля дата и месяц рождения сделать полями со списком.
2. Разработать форму для просмотра и ввода данных о номерах. Поле номер сделать полем со списком.
3. Разработать форму для просмотра и ввода данных о бронировании номера. Поле ФИО постояльца сделать полем со списком.

4. Сделать форму из пункта 3 подчиненной форме из пункта 2, добавить в получившуюся форму кнопку при нажатии на которую, открывается форма из пункта 1.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

#### **Вариант 6.2**

1. Разработать форму для просмотра и ввода данных о служащих гостиницы.

2. Разработать форму для просмотра и ввода данных о номерах гостиницы.

3. Разработать форму для просмотра и ввода данных о расписании работы и ответственности служащих за номера гостиницы.

4. Сделать форму из пункта 1 основной, а форму из пункта 3 - подчиненной. Добавить в получившуюся форму кнопку, при нажатии на которую открывается форма из пункта 2, для указанных записей.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

#### **Вариант 6.3**

1. Разработать форму для просмотра и ввода данных о занятых на текущий момент номерах гостиницы. Поле номер сделать полем со списком.

2. Разработать форму для просмотра и ввода данных о постояльцах гостиницы.

3. Разработать форму для просмотра и ввода данных о выписке постояльцев из номера. Добавить поле, в котором вычисляется стоимость проживания постояльца.

4. Сделать форму из пункта 1 основной, а форму из пункта 3 – подчиненной. Добавить в получившуюся форму кнопку, при нажатии на которую, открывается форма из пункта 2.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Лабораторная работа №4**

#### **Вариант 6.1**

1. Создать отчет, содержащий информацию о номерах гостиницы.

2. Создать отчет о занятости номеров за последний месяц.

3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1. Добавить информацию о прибыли, которую принес гостинице каждый номер и о общей выручке гостиницы.

#### **Вариант 6.2**

1. Создать отчет о постояльцах гостиницы (о всех, даже уже уехавших).

2. Создать отчет о проживании постояльцев в номерах. Упорядочить информацию по номерам (по возрастанию номеров) и по датам проживания.

3. Сделать отчет из пункта 1 основным, а отчет из пункта 2 - подчиненным. Добавить информацию о суммарном количестве дней, проведенных каждым постояльцем в гостинице и об общей сумме оплаты за проживание каждым постояльцем.

#### **Вариант 6.3**

1. Создать отчет о сотрудниках гостиницы и номерах, за которые они ответственны (если ответственные).
2. Создать отчет о номерах гостиницы. Упорядочить информацию по номерам.
3. Сделать отчет из пункта 1 основным, а отчет из пункта 2 – подчиненным. Добавить информацию о количестве номеров, за которые ответственен сотрудник.

### **Лабораторная работа №5**

#### **Вариант 6.1**

1. С помощью макроса показать информацию о постояльцах, проживающих на данный момент времени в определенном номере. Вывести сообщение в случае незанятости номера.
2. Вывести данные объекта БД из пункта 1 в формате MS Excel.
3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

#### **Вариант 6.2**

1. Создать макрос, с помощью которого можно вычислить стоимость проживания постояльца в гостинице (стоимость проживания вычисляется как количество дней, умноженное на стоимость места в номере). При этом проживающему представляется скидка в 15 процентов от основной стоимости в случае, если посетитель не достиг 18-летнего возраста, но старше 7 лет. Если постоялец младше 7 лет, то он оплачивает лишь 50 процентов от основной стоимости проживания.
2. Вывести данные объекта БД из пункта 1 в формате MS Excel.
3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных.

#### **Вариант 6.3**

1. Создать макрос, с помощью которого можно вычислить упущенную прибыль гостиницы за последний месяц, т.е. найти свободные номера и вычислить общую стоимость их простоя.
2. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных.
3. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

### **ВАРИАНТ 7**

#### **Информационная система «Туристическое агентство»**

##### **Лабораторная работа № 1**

Вы разрабатываете информационную систему для туристического агентства. Существуют заранее определенные маршруты. Маршрут может включать в себя несколько стран. Клиенты формируются в группы, и затем уже группа отправляется по маршруту. В агентстве работает несколько сотрудников. Оплата от клиентов принимается как за наличный, так и по безналичному расчету, в нескольких валютах (рубли, доллары, евро). Стоимость маршрута указывается в рублях.

##### **Лабораторная работа №2**

#### **Вариант 7.1**

1. Найти клиентов, едущих в определенную страну (название страны – параметр запроса).

2. Найти сотрудников, продававших путевки в текущем месяце.
3. Подсчитать количество человек в каждой группе, которые формируются для поездок в Испанию (они еще не уехали).
4. Пересчитать суммы, заплаченные клиентами (тех, что заплатили в евро) за путевки из евро в рубли, согласно введенному курсу (курс - параметр запроса).
5. Удалить данные о клиенте. Информацию можно удалить, только если клиент либо еще не успел заплатить за поездку и не успел ехать, либо если он уже вернулся из оплаченной поездки.

#### **Вариант 7.2**

1. Найти все предлагаемые туры, посещающие определенную страну (название страны – параметр запроса).
2. Найти клиентов, взявших путевки в Турцию.
3. Найти клиентов, которые ранее уже пользовались услугами данного агентства более двух раз за прошлый год.
4. Добавить в группу нового клиента.
5. Удалить информацию о маршруте. Информацию можно удалить, если на этот маршрут нет ни одной заказанной путевки.

#### **Вариант 7.3**

1. Найти сотрудников, продававших туры в определенную страну (название страны – параметр запроса).
2. Найти группы клиентов, которые еще не отправились путешествовать.
3. Найти все группы среди формирующихся (еще не уехали), в которых не более 3 человек.
4. Перевести клиента из одной группы в другую (если есть), отправляющуюся позже по тому же маршруту, при условии, что группа еще не отправилась.
5. Удалить информацию о сотруднике, при условии, что он не продал ни одной путевки.

### **Лабораторная работа №3**

**Во всех разработанных формах удалить системные кнопки и кнопки перехода по записям, заменить их на свои кнопки. Каждая разработанная форма (если она не подчиненная) должна иметь заголовок, отражающий назначение формы.**

#### **Вариант 7.1**

1. Разработать форму для просмотра и ввода данных о маршруте.
2. Разработать форму для просмотра и ввода данных о составе маршрута.
3. Разработать форму для просмотра и ввода данных о клиентах.
4. Сделать форму из пункта 1 основной, а форму из пункта 2 – подчиненной. В получившейся форме добавить кнопку, при нажатии на которую открывается форма из пункта 3 для указанных записей. Добавить поле, в котором указывается, какую сумму получила фирма за этот маршрут со всех клиентов, ездивших по нему.
5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

#### **Вариант 7.2**

1. Разработать форму для просмотра и ввода данных о клиентах фирмы.
2. Разработать форму для просмотра и ввода данных о путевках, по которым съездили клиенты. Указать дату поездки, ее стоимость.

3. Разработать форму для просмотра и ввода данных о маршрутах, предлагаемых фирмой.

4. Сделать форму из пункта 1 основной, а форму из пункта 2 – подчиненной, добавить кнопку, при нажатии на которую открывается форма из пункта 3 для указанных записей. Добавить в получившуюся форму поле, в котором подсчитывается, сколько денег клиент в сумме заплатил за все поездки.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

### **Вариант 7.3**

1. Разработать форму для просмотра и ввода данных о сотрудниках фирмы.

2. Разработать форму для просмотра и ввода данных о путевках, проданных фирмой.

3. Разработать форму для просмотра и ввода данных о маршрутах, по которым предлагает путевки фирма.

4. Сделать форму из пункта 1 основой, а форму из пункта 2 – подчиненной. Добавить в получившуюся форму кнопку, при нажатии на которую открывается форма из пункта 3 для указанных записей. Добавить поле в котором подсчитывается сумма, на которую сотрудник продал путевок.

5. Разработать кнопочную форму (с помощью мастера кнопочных форм), которая бы загружалась при загрузке базы данных и предлагала пользователю следующие действия: выполнить запросы, созданные в предыдущей лабораторной работе, открыть для работы форму из пункта 4, окончание работы с базой.

## **Лабораторная работа №4**

### **Вариант 7.1**

1. Создать отчет о сотрудниках. Упорядочить информацию по ФИО сотрудника.

2. Создать отчет о проданных путевках. Упорядочить информацию по маршрутам.

3. Сделать отчет из пункта 2 подчиненным отчету из пункта 1. Добавить информацию о сумме, на которую каждый сотрудник продал путевки. Добавить информацию об общей сумме, которую заплатили фирме.

### **Вариант 7.2**

1. Создать отчет о клиентах.

2. Создать отчет о купленных путевках. Упорядочить информацию по дате отправления.

3. Сделать отчет из пункта 1 основным, а отчет из пункта 2 – подчиненным. Добавить информацию о сумме, которую каждый клиент заплатил фирме.

### **Вариант 7.3**

1. Создать отчет о группах и маршрутах, по которым поедут группы. Упорядочить информацию по номерам групп.

2. Создать отчет о клиентах. Упорядочить информацию по ФИО клиента.

3. Сделать отчет из пункта 1 основным, а отчет из пункта 2 – подчиненным. Добавить информацию о количестве человек в группе.

## **Лабораторная работа №5**

### **Вариант 7.1**

1. При помощи макроса вывести список клиентов с указанием групп и маршрутов. Добавить информацию о сумме, которую каждый клиент заплатил фирме.

2. С помощью макроса произвести замену страны в маршрутах.
3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

### **Вариант 7.2**

1. Создать макрос, с помощью которого можно проверить рентабельность поездок групп: если в сформированной группе не более 3 человек, то поездку признать нерентабельной.
2. С помощью макроса показать маршруты, по которым отправляются такие группы.
3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

### **Вариант 7.3**

1. При помощи макроса вывести список сотрудников, продававших туры в определенную страну. Добавить информацию о сумме, на которую каждый сотрудник продал путевки.
2. Вывести данные объекта БД из пункта 1 в формате MS Excel.
3. Осуществить при помощи макроса возможность автоматического открытия кнопочной формы из пункта 5 лабораторной работы №3 при запуске базы данных. В кнопочной форме осуществить запрашивание пароля при нажатии кнопки, открывающей форму из пункта 4 лабораторной работы №3.

## **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. [681.518 (075) ПЗ3] Пирогов В.Ю. Информационные системы и базы данных: организация и проектирование. – СПб.: БХВ-Петербург, 2009. – 528 с. Количество экз. в библиотечке – 15.
2. [004.4 X20] Харитонов И., Рудикова Л. Microsoft Office Access 2007. - СПб.: БХВ - Петербург, 2008. - 1268 с. Количество экз. в библиотечке – 5.
3. [004.6 (075) С56] Советов Б. Я., Цехановский В.В., Чертовской В.Д. Базы данных : теория и практика: учебник. - 2-е изд.. - М.: Юрайт, 2012. - 464 с. Количество экз. в библиотечке – 60.
4. [004.6 Т83] Туманов В. Е. Основы проектирования реляционных баз данных: учебное пособие. - М.: Интернет-Ун-т Информ. Технологий: Бином, 2011. - 420 с. Количество экз. в библиотечке – 10.
5. [004.451 М65] Грох М. Microsoft Office Access 2007. Библия пользователя. - М. и др.: Диалектика, 2009. - 1200 с. Количество экз. в библиотечке – 3.



## СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА №1	
Создание базы данных и объектов типа "Таблица" в среде MS ACCESS.	
Определение связей между таблицами, задание схемы базы данных.....	3
ЛАБОРАТОРНАЯ РАБОТА №2	
Создание запросов в среде MS ACCESS.....	10
ЛАБОРАТОРНАЯ РАБОТА №3	
Создание объектов типа "Форма" в среде MS ACCESS.....	20
ЛАБОРАТОРНАЯ РАБОТА №4	
Разработка отчетов в среде MS ACCESS.....	28
ЛАБОРАТОРНАЯ РАБОТА №5	
Работа с макросами в среде MS ACCESS.....	32
ВАРИАНТЫ ЗАДАНИЙ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ.....	43
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	64