

Государственный комитет Российской Федерации
по связи и информатизации
Сибирский государственный университет
телекоммуникаций и информатики

**Л.Ф. Лебеденко
С.Г. Лапова**

БАЗЫ ДАННЫХ

Методические указания

Новосибирск
2006

Л.Ф. Лебедеко, С.Г. Лапова

В методических указаниях описан цикл лабораторных работ и задания для курсового проектирования по дисциплине "Базы данных" для студентов факультета ИВТ

Кафедра телекоммуникационных систем и
вычислительных средств

Талб. - 4

Рецензент: А.В. Борисов

Для специальностей 201200, 220400

Утверждено редакционно - издательским советом СибГУТИ
в качестве методических указаний

© Сибирский государственный университет
телекоммуникаций и информатики, 2006 год

ВВЕДЕНИЕ

Настоящие методические указания предназначены для использования в процессе лабораторного практикума и курсового проектирования по курсу "Базы данных" студентами специальностей 201200, 220400.

В результате выполнения описанного в указаниях цикла лабораторных работ студенты должны научиться работать в системе управления базами данных Access и программировать на языке Visual Basic for Application, а также в ходе курсового проектирования научиться разрабатывать автоматизированные информационные системы.

Лабораторная работа №1

Создание таблицы базы данных

Цель работы: Приобрести навыки создания таблиц базы данных в различных режимах Microsoft Access.

Краткая теория

Access хранит все таблицы базы данных в одном файле. Прежде, чем приступить к созданию таблиц базы данных, необходимо создать файл БД. Тип файла БД имеет расширение .mdb. Это расширение является зарегистрированным в Windows и связывается с программой Access.

Создание таблицы БД состоит из двух этапов: определения структуры (состава и имен полей, типа данных каждого поля, размера поля, ключей и т.д.) и заполнения таблицы данными.

В данной работе рассматривается три способа создания БД:

- в режиме таблицы;
- с помощью мастера таблиц;
- в режиме конструктора таблиц.

Порядок выполнения работы:

1. Создайте новую таблицу в режиме таблицы. Такой режим позволяет пользователю создать таблицу, не определяя предварительно ее структуру. После выбора этого режима сразу открывается пустая таблица, в которую можно ввести данные. При сохранении Access автоматически присвоит соответствующий тип данных каждому полю, т. е. создаст структуру таблицы.

Для выполнения задания выполните следующие действия:

- Загрузите Microsoft Access и в окне диалога **Создание** дважды щелкните на значке шаблона **Новая база данных**.
- В окне диалога **Файл новой БД** укажите имя файла БД и щелкните на кнопке **Создать**. Выберите вариант создания **Таблицы**. В результате на экране появится таблица, содержащая 20 столбцов и 30 строк.
- Переименуйте стандартные имена полей в соответствии с заданием. Для этого дважды щелкните на заголовке соответствующего поля (клетка должна выделиться цветом) и введите новое название.
- Заполните таблицу пятью произвольными записями и покажите результат преподавателю.
- Выйдите из режима создания таблицы без сохранения результатов.

2. Создайте новую таблицу БД, используя мастер таблиц. В этом режиме автоматически создается новая таблица по одному из шаблонов. Предлагается

более 40 образцов таблиц. Каждая таблица шаблона содержит набор полей, из которых выбираются те, которые соответствуют по типу данных полям результирующей таблицы. Включаемые в таблицу поля при необходимости могут быть переименованы.

Для создания таблицы БД с помощью Мастера

- В окне БД в режиме **Таблица** щелкните на ярлычке **Создать** и выберите вариант создания **Мастер таблиц**.

- Ознакомьтесь с набором таблиц делового и личного применения.

- В списке "**Образцы таблиц**" выделите таблицу, которая больше подходит по назначению к заданию, и переместите нужные вам поля в список "**Поля новой таблицы**". Чтобы переименовать любое из добавленных полей, выделите его и воспользуйтесь кнопкой **Переименовать поле**. Тип данных при этом сохраняется. После завершения формирования списка полей, щелкните по кнопке **Далее**.

- В текстовом поле следующего окна задайте имя вашей новой таблицы и выберите автоматический способ задания ключевого поля. Щелкните по кнопке **Далее**.

- Выберите режим "**Непосредственный ввод данных в таблицу**" или "**Ввод данных в таблицу с помощью формы, создаваемой мастером**". Щелкните по кнопке **Готово**.

- Введите в БД несколько записей и покажите результаты работы преподавателю.

- Выйдите из таблицы без сохранения данных.

3. Создайте новую таблицу в режиме конструктора. В этом режиме создание таблицы начинается с определения ее структуры. Режим конструктора позволяет разработчику произвольно задавать параметры всех элементов структуры таблицы.

Для работы в режиме конструктора

- В окне БД щелкните на вкладке **Таблицы**, а затем на кнопке **Создать**. В окне диалога **Новая таблица** выберите вариант создания **Конструктор**.

- В столбце **Имя поля** введите название нового поля, нажмите клавишу <Таб> и выберите для него тип данных из списка столбца **Тип данных**. Для поля "**Пол**" в списке столбца **Тип данных** выберите **Мастер подстановок**. Мастер строит для данного поля список значений на основе введенных пользователем фиксированных значений или на основе полей другой таблицы с данными для подстановки. Для поля "**Военнообязан**" выберите логический тип данных.

- На вкладке **Общие** укажите размер поля (число символов), формат поля и т. д. Для поля "**Телефон**" создайте маску ввода. Для добавления пропущенных полей щелкните по кнопке **Добавить строки** на панели инструментов конструктора таблиц. Для ускорения ввода данных в таблицу отдельным полям

можно назначить значение по умолчанию. Это значение будет появляться в соответствующем поле каждой новой записи и его можно заменить другим значением. Для поля "*Средний балл*" задайте ограничение на вводимые значения, при нарушении которых запрещается ввод и выводится текст, заданный свойством *Сообщение об ошибке*.

- Выйдите из режима конструктора с сохранением структуры БД, без указания ключевых полей.
- Откройте созданную БД и заполните ее двадцатью произвольными записями. Вводимые в поля таблицы значения должны соответствовать типам данных и накладываемым на них ограничениям. Проверьте действие ограничений на вводимые значения. Результаты работы покажите преподавателю.
- Завершите работу с Microsoft Access.

Задание для создания таблицы

Список полей БД "Студент": Номер, ФИО, Пол, Дата рождения, Домашний адрес, Факультет, Телефон_деканата, ФИО_декана, Курс, Группа, Количество_человек_в_группе, Средний балл, Размер_стипендии, Военнообязан.

Лабораторная работа №2

Редактирование, сортировка и отбор данных с помощью фильтра

Цель работы: Приобрести навыки работы с макетом таблицы, редактированием и сортировкой данных в табличном режиме.

Краткая теория

Для удобства работы с таблицей можно изменять ее представление на экране (менять ширину столбцов, строк, высоту строки, цвет текста, линий сетки или фона, оформление, фиксировать столбцы на экране или скрывать отдельные столбцы и т. д.), т. е. изменять макет таблицы. При этом используются панель инструментов **Форматы в режиме таблицы** и пункт меню **Формат**. Отдельные операции по изменению макета удобнее и быстрее выполняются непосредственно в таблице с помощью мыши.

В режиме таблицы может быть произведена корректировка данных. При этом вносимые в таблицу изменения не должны противоречить свойствам полей: типам данных, ограничениям на вводимые значения.

Для просмотра и корректировки записей БД, удовлетворяющим определенным условиям отбора, предусмотрены сортировка и фильтрация записей.

Порядок выполнения работы:

1. Откройте базу данных, созданную в лабораторной работе №1, в режиме таблицы и настройте параметры макета:

- Измените ширину столбцов. Для этого установите указатель мыши на линию, разделяющую названия столбцов (курсор превращается в планочку со стрелками), и перетащите границу на нужное расстояние. Можно удалить столбец с экрана, совместив, таким образом, его правую и левую границы.

- Аналогично измените высоту строк. Курсор в этом случае устанавливается в области маркировки записи (серая полоска слева), на границе между записями. Обратите внимание, что при этом изменяется высота всех строк таблицы.

- Переместите столбец "*Пол*" в конец таблицы. Для этого поместите указатель мыши на заголовок столбца так, чтобы он принял вид вертикальной стрелки и, не отпуская клавишу, перетащите столбец на новое место. Восстановите исходный порядок расположения полей.

- Закрепите первые два столбца таблицы на экране, воспользовавшись соответствующей командой меню **Формат**.

- Скройте столбец "*Военнообязан*" командой **Скрыть столбцы** меню **Формат**. Отобразите скрытые столбцы на экране.

2. Отредактируйте данные в столбце "*ФИО*". Для этого сделайте столбец текущим и выберите команду **Замена** меню **Правка**. Образец поиска и замены выберите произвольно. Внимательно проанализируйте и правильно задайте условия поиска.

3. Отсортируйте данные по значению поля "*ФИО*" в порядке убывания, щелкнув по кнопке **Сортировка** (курсор при этом должен находиться в поле сортировки). Отмените сортировку командой **Удалить фильтр** меню **Записи**. Сортировку можно выполнить также командой **Сортировка** меню **Записи**.

4. Отсортируйте записи таблицы по двум полям. В качестве первого ключа сортировки выберите столбец "*Факультет*", в качестве второго ключа - поле "*ФИО*". Сортировка по нескольким полям выполняется с помощью расширенного фильтра меню **Записи!Фильтр!Расширенный фильтр**. В появившемся окне диалога выберите из списка имена полей и порядок сортировки. Щелкните на панели инструментов по кнопке **Применение фильтра** и проанализируйте результаты.

5. Выведите сведения о мужчинах с помощью фильтра по выделенному. Это простейший способ задания условий отбора записей, при котором в таблице выделяется значение поля или его части. После выделения условия отбора в поле "*Пол*", воспользуйтесь командой меню **Записи!Фильтр!Фильтр по выделенному** или нажмите соответствующую кнопку на панели инструментов. Добавьте еще одно произвольное условие отбора в поле "*Год рождения*" для уже отобранных записей и еще раз воспользуйтесь **фильтром по выделенному**. Каждое новое выделение накладывает дополнительное условие на отбор записей существующим фильтром. Для отмены действия фильтра удобно пользоваться соответствующей кнопкой панели инструментов.

6. Воспользуйтесь для отбора записей контекстным меню. Для этого в поле "*ФИО*" щелкните правой клавишей мыши и введите в текстовое поле маску ввода для фамилий, начинающихся на букву "К". После просмотра полученных сведений отмените фильтр.

7. Повторите п.6, но теперь не выходя из текстового поля, нажмите кнопку <ТАБ>. Далее правой клавишей мыши щелкните по полю "*Группа*" и в текстовом поле введите условия отбора сведений о студентах групп специальности "Сотовая связь". После просмотра результатов удалите фильтр.

8. Выделите сведения о мужчинах, родившихся , например, в 1986 году. Для этого воспользуйтесь расширенным фильтром, введя условия отбора в бланк конструктора. Примените, а затем отмените фильтр.

9. Удалите две последние записи таблицы, а затем попробуйте добавить новую запись. Проследите за значением поля "Счетчик". Удалите введенную запись. Сохраните таблицу и закройте БД, а затем сожмите ее, воспользовавшись служебными программами меню **Сервис**. Задайте имя файла сжатой версии БД и сохраните его. Откройте новый файл и добавьте в конец таблицы новую запись. Как изменился счетчик?

10. Сохраните БД и завершите сеанс работы с Microsoft Access.

Лабораторная работа № 3

Создание простых запросов

Цель работы: Приобрести навыки создания различных видов запросов на основе таблиц баз данных Microsoft Access в режиме мастера и конструктора.

Краткая теория

Простой запрос позволяет выбрать необходимые данные таблицы, произвести вычисления и добавление новых полей в таблицу, получить результат в виде новой таблицы. Посредством запроса можно произвести обновление данных в таблице, добавление и удаление записей. Последовательное выполнение ряда запросов позволяет решать достаточно сложные задачи, не прибегая к программированию. Простейшие запросы (запрос на выборку, поиска повторяющихся записей и перекрестный) могут быть созданы с помощью мастеров Access. Мастер запросов ускоряет процесс создания запроса. В бланке запроса, подготовленном мастером, не формируются условия отбора и вычисляемые поля. При необходимости такой запрос можно отредактировать в режиме конструктора, который предоставляет большие возможности по созданию запросов.

Рассмотрите различные варианты построения запросов.

Порядок выполнения работы:

1. Создайте простой запрос на выборку с помощью Мастера. Для этого
 - Откройте свою базу данных, на вкладке **Запросы** щелкните на кнопке **Создать** и выберите вариант **Простой запрос**.
 - В раскрывшемся списке выберите имя нужной таблицы.
 - Дважды щелкните на полях, которые должны отображаться в запросе.
 - Ознакомьтесь с полученными результатами.

2. С помощью мастера создайте новый итоговый запрос, содержащий только сведения о количестве студентов на каждом факультете. Для этого при построении запроса на выборку (п.1) щелкните на переключателе **Итоговый** и по кнопке **Итоги**. Укажите, какие итоговые значения необходимо вычислить. Ознакомьтесь с результатом запроса.

3. Постройте запрос для поиска повторяющихся записей в полях таблицы БД. Такой запрос выводит только те записи, для которых есть хотя бы еще одна запись в таблице с одинаковым значением в выбранных полях. Для построения запроса

- В окне диалога **Новый запрос** выберите из списка **Повторяющиеся записи**.
- Дважды щелкните на имени поля (полей), по которому будет определяться дублирование.
- Задайте имя запроса и щелкните по кнопке **Далее**.
- Просмотрите результаты запроса.

4. Подсчитайте средний балл, на каждом курсе каждого факультета. Результат оформите в виде перекрестного запроса. Перекрестный запрос представляет собой сводную таблицу, в которой по строке и столбцу указываются заголовки полей, по которым группируются данные. На пересечении столбцов и строк указывается заголовок вычисляемого поля. При этом значения вычисляемого поля группируются по полям, используемых в качестве заголовков, и для получения групп значений применяется одна из выбранных статистических функций. При построении запроса

- В окне диалога **Новый запрос** выберите из списка **Перекрестный запрос**.
- В первом окне Мастера выберите имя таблицы.
- Продумайте, какие поля будете использовать в качестве строк и столбцов.
- Дважды щелкните на имени поля (полей), значения которого будут использоваться в качестве заголовков строк.
- Дважды щелкните на имени поля (полей), значения которого будут использоваться в качестве заголовков столбцов.
- В списке полей выберите поле, по которому будет определяться значение на основании выбранной статистической функции.
- В последнем окне мастера задайте имя запроса и щелкните по кнопке **Готово**.

5. Выберите из таблицы сведения о военнообязанных мужчинах, получающих стипендию. Для этого создайте с помощью Конструктора запрос на выборку

- В окне диалога **Новый запрос** выберите из списка **Конструктор**.
- Выберите имя таблицы, по которой будет построен запрос, и щелкните по кнопке **Добавить**.
- Перетащите в нижнюю область окна в строку **Поле** из списка полей таблицы те поля, которые будут отображаться в запросе, или выберите эти поля непосредственно из списка в строке **Поле**.
- Назначьте порядок сортировки и установите флажки вывода на экран.
- В ячейке строки **Условие отбора** поля, по которому осуществляется отбор данных, введите критерий отбора.
- Запустите запрос, щелкнув по кнопке на панели инструментов.

6. С помощью запроса на обновление измените данные в исходной таблице. Например, нужно увеличить для всех военнообязанных размер стипендии на 10%. Для этого

- В окне диалога **Новый запрос** выберите из списка **Конструктор**.
- Выберите имя таблицы, по которой будет построен запрос, и щелкните по кнопке **Добавить**.
- С помощью пункта меню **Запрос** или значка **Тип запроса** на панели инструментов выберите тип запроса **Обновление**.

- В список полей в нижней части окна включите поле, по которому будет осуществляться отбор данных для обновления и название поля, значения которого должны быть изменены.
- В поле "*Военнообязан*" в строке *Условие отбора* введите условие отбора записей.
- Для поля, подлежащего обновлению, введите в соответствующую ячейку нужное выражение (если в выражениях участвуют имена полей, то они заключаются в квадратные скобки).
- Кнопкой на панели инструментов запустите запрос на обновление. Подтвердите необходимость изменения данных в таблице.
- Переключитесь в режим таблицы после выполнения запроса (кнопка на панели инструментов) и просмотрите обновленное содержимое поля.
- Закройте и сохраните запрос.
- Убедитесь, что после выполнения запроса данные в исходной таблице изменились.

7. Аналогично запросу на обновление сконструируйте запрос на создание новой таблицы. В новую таблицу включите сведения о студентах 1984-1985 года рождения. Выберите соответствующий тип запроса, список полей, отображаемых в новой таблице, и условие отбора записей в новую таблицу. Для ввода выражения в ячейку *Условие отбора* используйте построитель выражения, значок которого находится на панели инструментов.

8. Сконструируйте еще один запрос действия - запрос на удаление. Критерий отбора записей для удаления выберите произвольно. Запрос на удаление первоначально создается как запрос на выборку. После выбора типа запроса (см. п.6), в бланке запроса в нижней части окна появится строка *Удаление*. С помощью мыши переместите в бланк запроса символ звездочки (*) из списка полей таблицы, записи которой требуется удалить. В строке *Удаление* в столбце этого поля появится значение *Из*. В строке *Поле* выберите имя того поля (полей), для которого устанавливается условие отбора. Ведите в соответствующую ячейку условие отбора. Для предварительного просмотра удаляемых записей нажмите кнопку ***Вид*** на панели инструментов. Запустите и сохраните запрос.

9. Получите информацию о студентах факультета посредством параметрического запроса. При этом наименование факультета будет запрашиваться через диалоговое окно перед выполнением запроса. Основой такого запроса является запрос на выборку, но в условие отбора в этом случае вместо конкретного значения поля введите в квадратных скобках название параметра, например, [ВВЕДИТЕ НАИМЕНОВАНИЕ ФАКУЛЬТЕТА]. Сохраните запрос и проверьте его работу. Добавьте в условие отбора еще один параметр и снова запустите запрос.

10. Вычислите стипендиальный фонд каждой группы. Для этого

- постройте простой запрос, содержащий только два нужных поля: поле, по которому производится группировка и поле, для которого выполняется групповая операция. Использование групповых операций позволяет выделить группы записей с одинаковыми значениями в указанных полях и использовать для других полей этих групп определенную статистическую функцию.
- На панели инструментов щелкните по значку **Групповые операции** или выполните команду **Групповые операции** меню **Вид**.
- Далее, в бланке конструктора в соответствующей ячейке *Групповая операция* выберите вариант *Группировка*.
- В столбце поля, над которым производится вычисление, замените слово "Группировка" на нужную статистическую функцию, выбрав ее из списка, предлагаемого строителем выражения.
- Для изменения имени поля установите курсор мыши на заголовке поля "Размер стипендии" и нажмите правую клавишу. В контекстно-зависимом меню выберите команду **Свойство!Подпись** и введите новый заголовок "Стипендиальный фонд"
- Запустите запрос на выполнение и сохраните его. Результат запроса с использованием групповых операций содержит по одной записи из каждой группы.

11. Включите в простой запрос, созданный в п.1, вычисляемое поле, в котором рассчитывается размер стипендии с учетом районного коэффициента. Вычисляемое поле, включенное в запрос, позволяет получить новое поле с результатами вычислений только в таблице запроса и не создает полей в таблицах БД. Для построения запроса

- Откройте простой запрос в режиме конструктора.
- Если поле "Размер стипендии" отсутствует в бланке конструктора, добавьте его из списка полей таблицы.
- В пустой ячейке строки *Поле* введите выражение для расчета значений поля "К_выдаче" в виде имя_нового_поля: [имя_поля]<выражение>. Для построения выражения воспользуйтесь строителем выражений.
- Запустите запрос.
- Просмотрите результаты запроса, сохраните его и завершите работу с Microsoft Access.

Конструирование многотабличных запросов

Цель работы: Приобрести навыки конструирования различных видов запросов на основе нескольких взаимосвязанных таблиц баз данных Microsoft Access.

Краткая теория

База данных СУБД Access является *реляционной базой данных* и состоит из взаимосвязанных реляционных таблиц. Логические связи каждой пары таблиц реализуются за счет одинаковых полей в связываемых таблицах - ключа связи. Ключом связи всегда является уникальный ключ главной таблицы. Значения уникального (первичного) ключа не могут повторяться в записях таблицы. Строки таблицы однозначно идентифицируются значением ключа. Ключ может быть простым или составным. Связи между таблицами дают возможность совместного использования данных из разных таблиц. Появляется возможность конструирования многотабличных форм, запросов и отчетов, обеспечивается требование целостности данных.

Реляционная база данных состоит из нормализованных таблиц, соответствующих следующим требованиям нормализации:

- Информационный объект (таблица) должен содержать уникальный идентификатор (ключ).
- Все описательные поля должны быть взаимонезависимыми.
- Все поля, входящие в составной ключ, должны быть также взаимонезависимы.
- Каждое описательное поле должно функционально - полно зависеть от ключа, т.е. каждому значению ключа соответствует только одно значение описательного поля.
- При составном ключе описательные поля должны зависеть целиком от всей совокупности полей, образующих ключ.
- Каждое описательное поле не может зависеть от ключа транзитивно, т.е. через другое промежуточное поле.

Выполнение требований нормализации обеспечивает построение реляционной БД без дублирования, в которой обеспечивается однократный ввод данных при первоначальной загрузке и корректировках.

В процессе создания БД сначала осуществляется конструирование таблиц, а затем создается схема данных, в которой фиксируются существующие логические связи между таблицами. Схема данных является графическим образом БД. Если поле, по которому устанавливается связь, является уникальным ключом как в главной таблице, так и в подчиненной, Access устанавливает связь один-к-одному (1:1). Такие таблицы легко могут быть объединены в одну. Если поле связи, является уникальным ключом в главной таблице, а в подчиненной таблице является не ключевым или входит в составной ключ, Access устанавливает связь один-ко-многим (1:M) от главной таблицы к

подчиненной. При выборе в качестве поля связи в главной таблице не ключевого поля Access сообщает, что тип отношения не может быть определен. В этом случае между таблицами возможно только связи - объединения.

Если модель данных разработана в соответствии с требованиями нормализации, в схеме данных могут быть заданы параметры обеспечения целостности базы данных, а именно

- В подчиненную таблицу не может быть добавлена запись с несуществующим в главной таблице значением ключа связи.
- В главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице.
- Изменение значение ключа связи главной таблицы должно приводить к изменению соответствующих значений в записях подчиненной таблице.

Установление между двумя таблицами связи типа 1:1 или 1:M и задание параметров целостности данных возможно только при следующих условиях:

- Связываемые поля имеют одинаковый тип данных, при этом имена полей могут быть различны.
- Таблицы хранятся в одной БД.
- Главная таблица связывается с подчиненной по первичному простому или составному ключу главной таблицы.

Access не позволяет создавать связи с параметрами целостности, если ранее введенные в таблицы данные не отвечают требованиям целостности. При вводе некорректных данных в связанные таблицы Access выводит сообщение.

Если для выбранной связи обеспечивается поддержание целостности, можно задать *режим каскадного обновления и удаления связанных записей*.

В режиме каскадного обновления, например, при изменении значения в поле связи главной таблицы Access автоматически изменит значения в соответствующем поле в подчиненных записях.

Если таблица с данными содержит поля с повторяющимися значениями (не нормализована), то расходуется дополнительная память, не может быть обеспечен однократный ввод и корректировка данных в этих полях, велика вероятность возникновения ошибок. С помощью Мастера Access позволяет автоматически или вручную выполнить анализ таблицы и разделить ее на несколько взаимосвязанных таблиц, в которых данные не будут дублироваться. Для полученных таблиц Мастер анализа таблиц автоматически создает схему данных, в которой устанавливаются связи 1:M и определяются параметры обеспечения целостности. После процедуры анализа для исходной таблицы будут получены нормализованные взаимосвязанные таблицы.

Однако Мастер анализа таблиц далеко не во всех случаях может правильно выделить повторяющиеся данные в отдельную таблицу. Предусматривается участие пользователя в корректировке предложений мастера по разделению таблицы, особенно если в таблице имеется несколько полей с повторяющимися значениями.

Порядок выполнения работы:

1. Выполните нормализацию таблицы, созданной в лабораторной работе №1. Для этого

- Откройте свою БД.
 - Проанализируйте и разбейте таблицу БД в соответствии с правилами нормализации. Запишите в тетрадь структуры и схему данных для новых взаимосвязанных таблиц.
 - Для выполнения автоматизированной нормализации таблиц воспользуйтесь командой меню **Сервис!Анализ!Таблица**. В третьем окне диалога выберите имя таблицы, для которой будет осуществляться анализ.
 - В следующем окне диалога выберете автоматический способ формирования таблиц. *Далее* проверьте правильность распределение полей по таблицам, созданных Мастером, сравните со своим вариантом нормализации и внесите коррективы в список полей взаимосвязанных таблиц путем перетаскивания полей из одной таблицы в другую, а при необходимости создайте еще одну таблицу или поменяйте ключевые поля.
 - Переименуйте таблицы, дважды щелкнув мышью на имени таблицы.
 - Окончательно определите уникальные ключи таблиц.
 - В последнем окне на вопрос о необходимости создать запрос, который бы объединил полученные таблицы в одну, соответствующие исходной, выберите *"Нет"*.
 - Откройте таблицы и ознакомьтесь с их содержимым.
 - Просмотрите структуру полученных таблиц в режиме конструктора.
- Обратите внимание, что Мастер анализа в подчиненной таблице по умолчанию определяет свойство **Тип элемента управления** закладки **Подстановка** равным значению **Поле со списком**. В качестве источника строк этого поля мастер выбирает запись главной таблицы со всеми полями. При просмотре в режиме таблицы в этом поле содержатся значения всех полей из соответствующей записи главной таблицы. Измените в строке **Тип элемента управления** значение **Поле со списком** на значение **Поле**, а в общих свойствах введите новое значение в строке **Подпись поля**. Посмотрите на результаты внесенных изменений в табличном режиме.
- На панели инструментов нажмите кнопку **Схема данных**, а затем на панели окна схемы данных щелкните по кнопке **Показать все связи**. Эту же команду можно выполнить через главное меню **Сервис!Схема данных**. Проанализируйте полученные данные.
 - Щелкните правой мышью на линии связи между двумя таблицами и удалите связь.
 - Восстановите удаленную связь. Для этого выделите в главной таблице уникальное ключевое поле, по которой устанавливается связь, а затем при нажатой кнопке мыши протащите курсор в соответствующее поле подчиненной таблицы.

- В открывшемся окне **Связи** установите параметры целостности, каскадное обновление связанных полей и каскадное удаление связанных записей. Эти же параметры можно задать командой меню **Связи!Изменить связь** или через контекстное меню **Изменить связь**.
 - Закройте окно схемы данных.
2. Создайте два варианта запроса на объединение по связанным таблицам, в который включите поля каждой из таблиц. Для построения первого варианта запроса воспользуйтесь Мастером создания запросов, для второго варианта - Конструктором.
 3. Создайте запрос на обновление данных в связанных таблицах. Поля, требующие обновления и поля, по которым задаются условия отбора, выберите произвольно, но так чтобы они принадлежали разным таблицам.
 4. Удалите записи из связанных таблиц, согласно произвольно выбранному критерию.
 5. Предварительно просмотрите в режиме таблицы список удаляемых записей, а затем запустите запрос. Удалять записи можно непосредственно из таблицы.
 6. Завершите работу с БД Access.

Лабораторная работа № 5

Проектирование простых форм

Цель работы: Приобрести навыки разработки и использования однотабличных форм.

Краткая теория

Формы являются основным диалоговым средством работы пользователя. Форма на основе одной таблицы может быть построена как самостоятельная для загрузки, просмотра и корректировки таблицы, так и в качестве вспомогательной для включения в составную форму.

Однотабличная форма может быть создана с помощью Мастера или в режиме Конструктора. В последнем случае вначале создается пустая форма и дальнейшее конструирование осуществляется разработчиком. Конструктор форм Access позволяет оформлять *Область заголовка*, *Область данных* и *Область примечаний формы*. Области формы заполняются различными графическими объектами. Форма в целом и ее отдельные элементы обладают свойствами, которые можно изменять с помощью команды **Свойства** меню **Вид** или командой **Свойства** контекстно - зависимого меню. В форму могут добавляться кнопки управления.

Порядок выполнения работы:

1. Создайте простую форму в режиме автоформы. Для этого
 - Откройте базу данных.
 - В окне **Базы данных** выберите щелкните на вкладке **Форма** и выберите команду **Создать**.
 - В окне **Новая форма** выберите в качестве источника данных имя таблицы. Создайте три возможных варианта **Автоформы**, сохраните их в базе данных и просмотрите полученные формы.
2. Воспользуйтесь для построения формы Мастером форм. Следуйте указаниям диалогового окна Мастера. В качестве источника данных выберите таблицу, созданную в лабораторной работе № 1. Сохраните, а затем откройте форму и добавьте в нее данные.
3. Создайте форму в режиме Конструктора:
 - В окне **Новая форма** выберите источник данных для формы и пункт **Конструктор**.
 - Щелкните на кнопке **Список полей** панели инструментов. Добавьте в область данных новой формы поля таблицы методом перетаскивания их из списка.
 - Вставьте заголовок формы. Для этого щелкните правой клавишей мыши в области данных и выберите пункт контекстно-зависимого меню **Заголовок/Примечание**. Расширить область заголовка формы можно, если установить курсор мыши на границу области заголовка и области данных и перетащить эту границу на нужное расстояние. С помощью панели элементов управления введите **Надпись** в поле заголовка.

Любой элемент может быть помечен щелчком мыши внутри его рамки. Пометить группу элементов можно, используя клавишу **Shift**. Изменение размеров помеченного элемента производится путем размещения курсора в соответствующих точках рамки, когда появляется двунаправленная стрелка. Перемещение возможно при появлении изображения руки. Причем при установке курсора мыши на любой границе рамки курсор отображается в виде раскрытой ладони и возможно перемещение элемента за пределы области. При установке курсора в левом верхнем углу курсор отображается в виде указательного пальца и перемещение возможно только в пределах области.
 - Осуществите выравнивание элементов формы. При редактировании связанных элементов - **Поле** и **Надпись** следует иметь в виду, что независимое перемещение поля и его надписи возможно только, когда курсор примет вид указательного пальца.
 - Выделяя элементы формы, настройте их свойства с помощью контекстного меню. Добавьте в форму графический объект. Для этого воспользуйтесь меню **Вставка**, выберите в графическом приложении любой графический объект или его фрагмент и вставьте рисунок в нужном месте

формы. С помощью мыши можете изменить размеры и местоположение объекта. В контекстном меню настройте **Свойства** графического объекта. При необходимости поменяйте значение **Фрагмент Макета** рисунка.

- Сохраните и просмотрите созданную форму.

4. Откройте в режиме конструктора одну из форм, созданную в п.1 и установите режим добавления записей. Для этого в свойстве формы **Ввод данных** установите значение *Да*.

5. Откройте форму и введите запись.

6. Закончите сеанс работы с базой данных.

Лабораторная работа №6

Создание многотабличных форм

Цель работы: Приобрести навыки создания форм, содержащих данные из нескольких связанных таблиц в режиме Мастера и Конструктора форм.

Краткая теория

Составная многотабличная форма создается для работы с несколькими взаимосвязанными таблицами. Многотабличная форма может состоять из основной части и одной или нескольких подчиненных включаемых форм. Подчиненная форма может быть построена на основе как подчиненной, так и главной таблицы относительно таблицы - источника основной части формы.

Многотабличная форма может не иметь включаемых форм. В этом случае в форме кроме полей таблицы, на основе которой она строится, добавляются поля из связанной с ней таблицы, являющейся главной относительно основной таблицы формы.

Многотабличная форма может быть создана в режиме Конструктора или с помощью Мастера. Мастер форм предоставляет возможность выбрать включаемые в форму поля из нескольких взаимосвязанных таблиц, а также запросов, и на их основе создает многотабличную форму, используя различные способы ее построения:

- явное включение подчиненной формы;
- вызов связанной формы по кнопке;

- многотабличная форма без подчиненных или связанных форм.

В первом случае подчиненная форма строится только для таблицы, которая является подчиненной, по отношению к таблице - источнику основной части. Подчиненная форма отображает данные из всех записей подчиненной таблицы, которые связаны с записью главной таблицы, отображаемой в основной части формы. Составная форма может включать одну или несколько подчиненных форм стандартного вида с выбранными полями.

Второй вариант построения формы - это многотабличная форма, в которой вместо подчиненной формы включается только кнопка. Нажатием этой кнопки вызывается на экран связанная форма. При этом содержимое связанной формы синхронизировано с текущей записью формы. Такой способ удобен для сложных перегруженных форм.

Составная многотабличная форма может не включать подчиненных и связанных форм. Такая форма создается, если необходимо отображать записи подчиненной таблицы, дополненной полями из одной или нескольких главных таблиц. Форма отображает поля из записи подчиненной таблицы и поля из единственной связанной с ней записи главной таблицы.

Порядок выполнения работы:

1. Откройте базу данных и проверьте схему связей таблиц, созданных в лабораторной работе №2.
2. Создайте сложную форму в режиме мастера форм. Для этого
 - в окне базы данных выберите вкладку **Формы/Создать**.
 - В диалоговом окне **Новая форма** выберите пункт **Мастер форм**.
 - В нижней части диалогового окна выберите в качестве источника данных главную таблицу.
 - В первом диалоговом окне мастера выберите поля, которые будут отображаться в главной форме. Не закрывая окно, щелкните на списке **Таблицы/Запрос** и выберите таблицу, содержащие данные для подчиненной формы.
 - Выберите поля, которые будут отображаться в подчиненной форме и щелкните по кнопке **Далее**.
 - В следующем диалоговом окне мастера убедитесь в том, что в списке данных, предназначенных для просмотра в главной форме, выделена главная таблица. Для непосредственного включения подчиненной формы переключатель **Подчиненные формы** должен быть включен.
 - В очередном окне выберите способ вывода подчиненной формы **Ленточный** или **Табличный**.
 - **Далее** выберите нужный стиль формы, а затем в последнем окне введите имена главной и подчиненной формы.
 - Просмотрите полученные результаты.

- Аналогично создайте второй вариант многотабличной формы, включающий кнопку для вызова связанной формы. В этом случае нужно выбрать вариант **Связанные формы**.
 - Выберите в качестве таблицы - источника подчиненную таблицу, создайте форму с помощью мастера. В этом случае многотабличная форма будет лишь включать поля из главной таблицы. Автоматически будет установлен тип формы **Одиночная форма**. Откройте форму и проанализируйте результаты.
3. Создайте сложную форму в режиме **Конструктора**, используя в качестве источника данных запрос на объединение таблиц, созданный в лабораторной работе №4. Сохраните форму и просмотрите полученные результаты.
4. Включите одну или более подчиненных форм в одну форму в режиме **Конструктора**. Для этого
- Создайте формы для главной и подчиненных таблицы, используя навыки создания простых форм, полученные в лабораторной работе №5.
 - Затем откройте главную форму в режиме **Конструктора** и перетащите подчиненные формы из **Окна базы данных** в главную форму.
 - Просмотрите полученные результаты.
 - Убедитесь в правильности установки связей. Для этого щелкните в **Конструкторе** на подчиненной форме, а затем на кнопке **Вид** панели инструментов. При этом свойство **Подчиненные поля** должны содержать имя поля в подчиненной форме для связи записей главной формы с записями подчиненной. Свойство **Основные поля** должно содержать имя поля в главной форме, с которым устанавливается связь.
 - Сохраните форму и просмотрите полученные результаты.
 - Используя возможности контекстного меню, в режиме **Конструктора**, внесите изменения в форму и сохраните ее.
5. Создайте еще одну простую форму для главной таблицы в режиме **Конструктора**, в которую добавьте элемент управления **Кнопка**, позволяющую просматривать связанные записи двух связанных форм. Для этого
- В первом диалоговом окне мастера в области **Категории** выберите пункт **Работа с формой**, а в области **Действия** - **Открытие формы**.
 - **Далее** выберите связанную форму, которую будете открывать с помощью кнопки.
 - В следующем окне диалога включите переключатель **Открыть форму для отображенных записей**.
 - **Далее** в раскрывшемся списке полей обеих таблиц, выберите по одному полю, используемому для синхронизации форм. Установите связь между выбранными полями щелчком на кнопке с двунаправленной стрелкой и нажмите кнопку **Далее**.

- В последнем диалоговом окне мастера выберите рисунок для кнопки или текст надписи и щелкните на кнопке **Готово**.
 - Откройте форму и просмотрите результаты.
6. Отредактируйте подчиненную форму в режиме **Конструктора**:
- Еще раз откройте в режиме конструктора многотабличную форму, созданную в п.5.
 - Дважды щелкните мышью в области подчиненной формы.
 - В свойствах **Формы** откройте вкладку **Макет** и в свойстве **Полосы прокрутки** задайте значение **Отсутствуют**, а в строке **Поле номера** записи установите значение **Нет**.
 - Добавьте в форму еще одну кнопку, закрывающую форму.
 - Откройте форму и ознакомьтесь с результатами.
 -
7. Осуществите расчет итоговых значений в подчиненной форме. Для этого
- Создайте подчиненную форму в режиме конструктора.
 - В разделе примечаний добавьте элемент управления **Поле**.
 - В свойстве **Данные** этого элемента введите выражение для расчета итогового значения произвольно выбранного поля (например, =sum[Размер_стипендии]).
 - На вкладке **Другие** диалогового окна свойств элемента управления введите имя для данного элемента управления, например, *Сумма*.
 - Установите в свойстве **Вывод на экран** значение **Нет**.
 - Сохраните подчиненную форму.
 - Создайте вновь или откройте главную форму в режиме конструктора.
 - Добавьте в форму элемент управления **Поле**, в котором будет отображаться результат итогового расчета.
 - В свойстве **Данные** этого элемента управления введите выражение, которое обеспечит связь со скрытым элементом управления подчиненной формы в виде:
=[имя_подчиненной_формы].Form![Имя_элемента_управления]
 - В свойстве **Формат поля** задайте нужный формат. В свойстве **Вывод на экран** установите значение **Да**.
 - Сохраните главную форму и запустите ее.
8. Закончите сеанс работы с базой данных.

Лабораторная работа №7

Создание панели управления приложения

Цели работы: Приобрести навыки создания главной кнопочной формы с помощью Диспетчера кнопочных форм.

Краткая теория

Для организации эффективной работы пользователя все разрозненные компоненты приложения обычно группируют по функциональному назначению. Для объединения объектов в едином диалоговом приложении создаются так называемые *кнопочные формы*. Кнопки такой формы обеспечивают вызов других кнопочных форм или отдельных объектов - отчетов, форм, макросов, модулей. Сюда же помещаются кнопки для возврата к кнопочным формам предыдущих уровней и выхода из Access. Вызов кнопочной формы при открытии БД позволяет сразу же начать работу в среде приложения. Элементами кнопочных форм являются кнопки с подписями. Подпись задается разработчиком и должна по возможности точно указывать действия, которые будут выполняться при ее нажатии. Выбор кнопки на панели управления инициализирует событие *нажатие кнопки*.

В Access имеется средство автоматизированной разработки формы по управлению приложением - Диспетчер кнопочных форм.

Порядок выполнения работы:

1. Создайте главную кнопочную форму базы данных, состоящую из двух страниц. Для этого
 - Откройте базу данных.
 - Выберите команду меню **Сервис ! Надстройка! Диспетчер кнопочных форм** и подтвердите создание кнопочной формы.

В окне диспетчера кнопочных форм появится строка **Главная кнопочная форма**. В этом окне диспетчера формируется список кнопочных форм разных уровней.
2. Создайте две кнопочные формы для работы по редактированию элементов приложения и отбору данных:
 - В окне диспетчера нажмите кнопку **Создать**. В окне **Создание** в поле **Имя страницы кнопочной формы** введите имя первой кнопочной формы **Работа с формами**. После нажатия кнопки **ОК** в окне диспетчера появится строка **Работа с формами**. Аналогично создайте страницу **Поиск**.
 - Оставьте стартовой (по умолчанию) главную кнопочную форму. Эта форма будет находиться на верхнем уровне иерархии взаимосвязанных кнопочных форм.
3. Сформируйте элементы главной кнопочной формы:
 - В окне диспетчера кнопочных форм щелкните по кнопке **Изменить**.
 - В окне **Изменение страницы кнопочной формы** введите новое название главной кнопочной формы.

- Сформируйте в главной кнопочной форме кнопку вызова подчиненной кнопочной формы. Для этого выберите команду **Создать** и в окне **Изменение элемента кнопочной формы** в строке **Команда** выберите **Переход к кнопочной форме**, а в строке **Кнопочная форма** выберите форму, к которой должна обеспечивать переход данная кнопка. Откройте список созданных кнопочных форм и выберите кнопочную форму **Работа с формами**. В строке **Текст** введите подпись для этой кнопки, например, *Ввод и редактирование*. Щелкните по кнопке **ОК**.

Аналогично создайте кнопку вызова для следующей формы, а для завершения работы с приложением создайте кнопку **Выход из приложения**. Закончив создание страницы кнопочной формы, нажмите кнопку **Заккрыть**. Это позволит вернуться к списку всех кнопочных форм приложения.

4. В подчиненных кнопочных формах создайте кнопку для возврата к главной кнопочной форме, заполнив поля в окне **Изменение элемента кнопочной формы**. Таким образом, установлены все необходимые связи между кнопочными формами. Вернитесь в окно диспетчера кнопочных форм.
5. Сформируйте элементы кнопочной формы **Работа с формами** :
 - В окне диспетчера кнопочных форм выберите кнопочную форму **Работа с формами**.
 - Для создания элементов в этой форме щелкните по кнопке **Изменить**.
 - В окне **Изменение страницы кнопочной формы** нажмите кнопку **Создать**.
 - В следующем окне **Изменение элемента кнопочной формы** выберите команду **Открытие формы в режиме редактирования**. При этом появится строка с наименованием **Форма**.
 - Выберите имя формы, созданной в лабораторной работе №5.
 - Введите подпись кнопки и щелкните на **ОК**.
 - В диалоговом окне **Изменение страницы кнопочной формы** еще раз нажмите кнопку **Создать**.
 - В окне **Изменение элемента кнопочной формы** выберите команду **Открытие формы в режиме добавления**.
 - Выберите имя формы.
 - Введите подпись кнопки и щелкните по **ОК**.
 - Вернитесь в окно диспетчера кнопочных форм и закройте окно.

Элементы для работы с формой **Поиск** создадим позже.
 - Запустите форму на выполнение.
6. В режиме конструктора измените свойства главной кнопочной формы, поменяйте ее название, добавьте слева рисунок, измените шрифт. Закройте форму.


7. Создайте макрос, позволяющий осуществить поиск данных по критерию, и подключите его к соответствующей кнопке в главной кнопочной форме.
Для этого
- В окне базы данных выберите вкладку **Макросы** и щелкните по кнопке **Создать**.
 - В окне Конструктора, в столбце **Макрокоманда** выберите из списка нужное действие. Например, создайте макрос поиска записи по образцу. Для этого в список макросов необходимо включить макрокоманду открытия формы и макрокоманду поиска записей. Образец поиска записи введите произвольно. Внимательно изучите и выберите аргументы макрокоманды.
 - Щелкните по кнопке закрытия окна и сохраните макрос.
 - Еще раз воспользуйтесь диспетчером кнопочных форм. Добавьте команду запуска макроса при нажатии кнопки **Поиск**. Введите соответствующий текст и имя макроса.
 - Сохраните и запустите главную кнопочную форму.
8. Добавьте в список макрокоманд макрос **Вывод на экран** сообщения в случае, если запись не найдена. Для этого
- Выберите режим Конструктора и в меню **Вид** установите флажок **Условия**.
 - В строке **Условие** нового макроса введите проверку совпадения поля с условием поиска. Пользуйтесь построителем выражений. Для проверки условия используйте оператор *If*. Если условие выполняется, из списка встроенных функций выберите функцию **Обработка ошибок** ERROR(0), в противном случае функцию **Сообщения** MsgBox(“значение не найдено”).
 - Сохраните макрос и откройте кнопочную форму.
 - Проверьте правильность ее работы для обоих случаев результатов поиска.
9. Измените параметры запуска меню **Сервис** так, чтобы при открытии базы данных на экране появлялась Главная кнопочная форма. В раскрывающемся списке **Форма** выберите главную кнопочную форму.
10. Закройте, а затем вновь откройте базу данных. Ознакомьтесь с результатами.
11. Закончите сеанс работы с Microsoft Access.

Лабораторная работа №8

Проектирование простых отчетов

Цель работы: Научиться проектировать отчеты по отдельным таблицам, вычислять итоги по данным, группировать и сортировать данные в отчетах.

Порядок выполнения:

1. Откройте базу данных.
2. Пользователю доступны два способа проектирования отчетов: с помощью мастера по созданию отчетов и с помощью конструктора отчетов. Создайте два автоотчета по запросам, созданным ранее: один в столбец, второй - ленточный. Сохраните каждый из них под своим именем.
3. Создайте отчет с помощью мастера отчетов. Для этого:
 - в разделе *Отчеты* окна базы данных нажмите кнопку **Создать**.
 - выберите опцию **Мастер отчетов** и нажмите кнопку ОК.
 - в следующем диалоговом окне мастера последовательно выберите несколько полей, которые будут включены в отчет и перейдите в следующее окно.
 - в следующем окне задаются поля для группировки (группировка задается по полям, в которых повторяются значения. Например: группы для студентов или отделы для сотрудников). Для данного отчета с помощью кнопки  отмените группировку и перейдите в следующее окно.
 - в следующем окне задайте порядок сортировки по фамилиям.
 - в следующих двух окнах выберите вид макета отчета и стиль вывода заголовков в создаваемом отчете.
 - Затем задайте имя созданному отчету и завершите его создание.
4. Просмотрите созданный отчет.
5. Выполните доработку созданного отчета с помощью конструктора. Для этого в окне базы данных, то имя отчета следует маркировать и затем щелкнуть на кнопке **Конструктор**. Затем внесите следующие изменения в проект отчета:
 - В область заголовка введите заголовок отчета.
 - В область примечания отчета вставьте формулу для вычисления, например суммы по зарплате или стипендии. Для этого вставьте в поле примечания свободное поле, в свойствах выберите закладку **данные** и с помощью построителя выражений задайте формулу $=SUM([имя_поля])$. Заголовок к полю введите "ИТОГО:".
6. Создайте еще один отчет с помощью мастера отчетов, в котором задайте поле для группировки, отсортируйте данные в группе и, нажав кнопку *Итоги*

для выполнения расчетов по группам, выберите, какие итоговые значения вы хотите получить. Задайте стиль и имя для созданного отчета.

7. Просмотрев отчет, доработайте его с помощью конструктора.

8. Создайте отчет с помощью конструктора с нуля. Для этого:

- В окне базы данных щелкните кнопку **Создать**, выберите в качестве источника данных таблицу или запрос и выберите режим **Конструктор**. Появится проект отчета содержащий область данных и колонтитулы.
- Откройте список полей из панели инструментов и вставьте нужные в область данных. Заголовки к полям разместите в верхнем колонтитуле, а сами поля в одной строке области данных. Ширину этой области уменьшить до размеров этой строки.
- В область нижнего колонтитула вставьте нумерацию страниц отчета. Для чего вставьте свободное поле и задайте формулу, например **=page**.
- Для создания заголовка и итоговых строк в меню **ВИД** щелкните команду **заголовок/примечание отчета**. В появившейся области заголовка введи заголовок, выбери для него стиль и шрифт. Вставьте в область заголовка знак фирмы, созданный ранее, как рисунок в графическом редакторе.
- В область примечания отчета задайте формулу для подсчета общего количества сотрудников или студентов, используя встроенную функцию **Count()**.
- Для группировки данных в отчете в меню **ВИД** щелкните команду **Сортировка и группировка**. В появившемся окне задайте поле, относительно которого будет выполняться группировка. Здесь же можно задать условие сортировки. Обратите внимание на свойства групп. В проекте отчета должны появиться области "заголовок группы" и "примечание группы".
- В область заголовка группы вставьте поле по которому выполняется группировка, а в область примечания группы - формулу для подсчета количества человек относящихся к каждой группе.
- Разделите строки и столбцы в отчете горизонтальными и вертикальными линиями.
- Сохраните проект отчета под именем и просмотрите его.

9. Создайте отчет по нескольким таблицам с помощью мастера отчетов.

10. Доработайте этот отчет с помощью конструктора.

11. Закончить работы с базой данных.

Лабораторная работа №9

Основы программирования на Visual Basic

Цель работы: Научиться использовать Visual Basic для:

- организации комплексного ввода дополнительных данных во время выполнения программы.
- использования переменных, без которых невозможно решить некоторых задачи.
- использования условного цикла или цикла с заданным числом повторений.

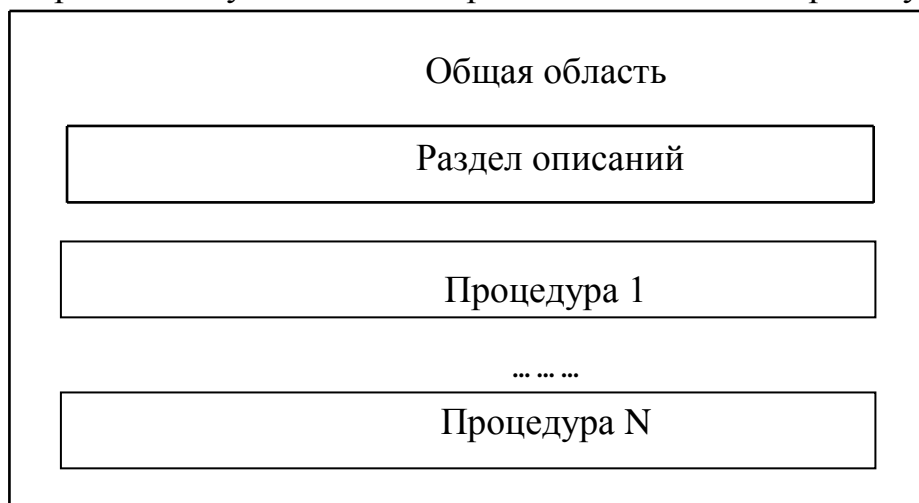
Основная теория:

Основой программы на Visual Basic является процедуры, состоящие из инструкций, которые выполняют необходимые действия и вычисления. Процедуры хранятся в модулях. Модули подразделяются на стандартные и модули форм и отчетов.

Стандартные модули представлены в окне базы данных. Вызов этих процедур может осуществляться из процедур обработки событий, процедур других стандартных модулей, макросов и просто из выражений.

Модули форм и отчетов создается автоматически при создании формы или отчета и является частью их описания.

Стандартный модуль состоит из раздела описаний и процедур:



В стандартном модуле могут храниться обычные процедуры типа Sub и Function. Sub-процедура - это подпрограмма, которая не возвращает значения. Function - процедура-функция, которая возвращает значение. При объявлении требуется соблюдать правила описания процедур.

Формат описания для процедуры Sub

Sub имя ([аргумент])

инструкции

end sub

Необязательный аргумент позволяет передавать в процедуру требуемые значения. Процедура **Sub** вызывается инструкцией: **Call имя**. Синтаксис

описания функции очень похож на синтаксис процедуры Sub, но есть и отличия:

```
Function имя ([ аргумент]) [ as тип]
    инструкции
    имя = выражение
end function
```

Параметр **as тип** позволяет явно задать тип данных, который возвращает функция. Если он опущен, то по умолчанию возвращается значение типа **Variant**. Если в разделе описания модуля помещена инструкция **Option Explicit**, то неявное описание запрещено. Функцию можно вызвать из выражения.

Константы и переменные могут быть описаны как в разделе описания модуля, тогда они доступны во всех процедурах модуля, так и внутри процедур, тогда они действуют только в ее пределах.

Формат описания переменных:

```
Dim Имя переменной [ as тип данных]
```

Основные типы данных, которые используются при описании: **Boolean, Byte, Integer, Long, Currency, Single, Double, Date, String, Variant, Object**. Если тип данных не указан, по умолчанию переменная получает тип **Variant**. Данные этого типа интерпретируются в зависимости от операции, в которой они используются, т.е. тип может меняться.

Формат описания констант:

```
Const Имя константы = значение
```

Константы может определять пользователь и можно использовать стандартные константы, определенные системой. Первые две буквы в имени встроенных констант являются префиксом. Константы из библиотеки VBA имеют префикс "vb". Константы библиотеки Access имеют префикс "as".

В выражениях могут использоваться стандартные функции языка. Перечислим некоторые из них: математические: Sin(x), Sqr(x), Exp(x), Log(x), Tan(x), Int(x) и пр.; преобразования: Str(s), Chr(x), Val(s); системные функции: InputBox- функция ввода данных через системное окно, MsgBox - функция выдачи сообщений в системное окно. Эти функции удобны для организации ввода - вывода в процессе работы программы.

Формат функции ввода:

```
Переменная=InputBox(Приглашение[, заголовок][, начальное значение])
```

Функция InputBox возвращает значение типа String. Например:

```
Dim c as string
```

```
Dim g as integer
```

```
c=InputBox("Введи наименование товара:")
```

```
g=Val(InputBox("Введи его цену:",0))
```

Формат функции вывода сообщений:

```
[Переменная=]MsgBox(Текст[,опция][,Заголовок])
```

Опция OP определяет внешний вид окна. Значение OP=OP1 + OP2

OP1-определяет вид пиктограммы выводимой в окне. Значения этой опции приведена в следующей таблице:

Константа	Значение	Описание
VbCritical	16	Используется значок <i>Критическое сообщение</i>
VbQuestion	32	Используется значок <i>Предупреждающий запрос</i>
VbExclamation	48	Используется значок <i>Предупреждение</i>
VbInformation	64	Используется значок <i>Информационное сообщение</i>

Например:

x=MsgBox("Возможна ошибка!",VbExclamation)

или

x=MsgBox("Возможна ошибка!",48)

OP2 - опция определяет набор кнопок выводимых в окне. Значения этой опции приводится в следующей таблице:

Константа	Значение	Описание
VbOKOnly	0	Отображается только кнопка ОК
VbOKCancel	1	Отображаются кнопки ОК и <i>Отмена</i>
VbAbortRetryIgnore	2	Отображаются кнопки <i>Прервать,Повторить</i> и <i>Пропустить</i>
VbYesNoCancel	3	Отображаются кнопки <i>Да, Нет</i> и <i>Отмена</i>
VbYesNo	4	Отображаются кнопки <i>Да, Нет</i>
VbRetryCancel	5	Отображаются кнопки <i>Повторить</i> и <i>отмена</i>

Например:

x=Msgbox("Функция="+str(f),VbOKOnly,"Результаты расчета")

x=MsgBox("Будете еще вводить?", VbYesNo+VbQuestion, "Запрос")

или

x=MsgBox("Будете еще вводить?,36,"Запрос")

Функция MsgBox возвращает значение, которое зависит от того, какая кнопка была нажата в окне. Значения, возвращаемые функцией приведены в следующей таблице:

Константа	Значение	Описание
VbOk	1	Ok
VbCancel	2	Отмена
VbAbort	3	Прервать
VbRetry	4	Повторить
VbIgnore	5	Пропустить
VbYes	6	Да
VbNo	7	Нет

Например:

```
x=MsgBox("Распечатать документ?",36)
```

```
if x=6 then Call PrintRep
```

Каждая инструкция в процедуре записывается с новой строки. Чтобы записать несколько инструкций в одной строке их надо разделять символом двоеточие. Если инструкция слишком длинна, то в конце строки вводят символ продолжения строки - подчеркивание (_) - ф затем нажимают клавишу <Enter> и продолжают инструкцию на следующей строке.

В языке Visual Basic Имеется целый набор инструкций, управляющих порядком выполнения команд. Рассмотрим некоторые из них.

1) Инструкция If ...Then ... Else

Формат:

```
if условие Then  
    [инструкции]  
[else  
    инструкции]  
End if
```

Например, разработаем процедуру вычисления функции $f=y/c$, где $y=5$, а c - вводится с клавиатуры. Если будет введено $c=0$, должно выводиться сообщение.

```
Sub prim()  
Dim f,c As Single  
Dim y As Integer  
y=5  
c=Val(InputBox("Введи C:"))  
if c=0 Then  
    MsgBox("Функция не существует!")  
else  
    f=y/c  
    MsgBox("функция="+str(f))  
end if  
end sub
```

2) Инструкция Select Case

Формат:

```
Select Case выражение  
    Case списокВыражений-1  
        инструкции  
    ...  
    Case списокВыражений-n
```

инструкции
[Case else
инструкции]

End Select

Например, необходимо определить значение скидки, обработав значение переменной UnOrder следующим образом:

```
Select Case UnOrder
  Case 1
    Discount=0
  Case 2, 3
    Discount=0.1
  Case 4 To 6
    Discount=0.2
  Case Is>6
    Discount=0.5
End Select
```

3) Цикл For ...Next

Формат:

For счетчик=начало **To** конец [**Step** шаг]
инструкции

Next

Счетчик - любая числовая переменная.

Например, вычислим функцию $f=a*x^2$, где $a=3$, а переменная $x=[1,5]$, шаг=1

```
Sub prim2()
  Dim a,x,f As Integer
  a=3
  For x=1 To 5
    f=a*x^2
    MsgBox("X="+str(x)+" F="+str(f))
  Next
End Sub
```

4) Цикл While ... Wend

Формат:

While условие
инструкции

Wend

Реализуем предыдущий пример, но таблица выводится по-другому:

```
Tabvivoda="" ' Переменная, в которой формируется таблица для
вывода
x=1
While x<=5
  f=a*x^2
```

```

    Tabvivoda=Tabvivoda+"X="+str(x)+" F="+str(f))+Chr(13)
    x=x+1
Wend
R=MsgBox(Tabvivoda, 0, "ТАБЛИЦА РАСЧЕТОВ")

```

5) Цикл Do ... Loop

Эта инструкция имеет четыре варианта синтаксиса:

- Синтаксис 1:
 Do While условие нахождения в цикле
 инструкции
 Loop
- Синтаксис 2:
 Do Until условие выхода из цикла
 инструкции
 Loop
- Синтаксис 3:
 Do
 инструкции
 Loop While условие нахождения в цикле
- Синтаксис 4:
 Do
 инструкции
 Loop Until условие выхода из цикла

Инструкция применяется аналогично While ...Wend. Например, вычислим сумму значений последовательности, вводимой с клавиатуры:

```

Sub Prim3()
Dim a,s As Single ' a- элемент последовательности,_
s- вычисляемая сумма
Dim rez As Integer
s=0
Do
a=Val(InputBox("Введи элемент последовательности:"))
s=s+a
rez=MsgBox("Будешь ещё вводить?", VbYesNo, "Нажми кнопку")
Loop Until rez=VbNo
rez=MsgBox("Сумма="+Str(s) , 0 , "Результаты расчета")
End Sub

```

Порядок выполнения:

1. Откройте базу данных.

2. Создайте в модуле процедуру **Sub** для вычисления таблицы значений функции $F=a|x^3|+a^2 \cdot x$ для $a=3$, x изменяется на интервале от -4 до 4 с шагом 2. Для этого:
- в окне базы данных перейти в закладку **Модули** и щелкнуть кнопку **Создать**. Access откроет окно проектирования модуля.
 - Вызовите команду *процедура* из меню **Вставка**. В появившемся диалоговом окне нажмите селективную кнопку **Подпрограмма** и введите имя создаваемой процедуры. Введите текст программы для решения поставленной задачи.
 - проведите компиляцию загруженного модуля. Если при компиляции будут выявлены ошибки, то их надо исправить.
 - После успешной компиляции выполните созданную процедуру и просмотрите результаты.
3. Подключите созданную процедуру к кнопке в форме. Для этого:
- перейдите в закладку **Формы** и создайте в режиме **Конструктор** новую форму.
 - В ней создайте кнопку. В окне **создания кнопки** не выбирайте действий, связанных с ее нажатием, щелкните кнопку **Отмена**.
 - В свойствах для созданной кнопки перейдите в закладку **События** и для опции **Нажатие кнопки** выберите значение **Процедура обработки событий**.
 - В текст процедуры обработки событий вставьте вызов процедуры Sub.
 - проверьте действие кнопки.
4. Скорректируйте созданную процедуру для расчета трех таблиц значений функции **f** при разных параметрах **a**=-2,4,7. Для этого организуйте внешний цикл ввода параметра **a**, отладьте скорректированную процедуру. Рекомендуется вывод таблиц организовать в окно **Msgbox** в виде таблиц целиком.
5. Создайте в модуле функцию для обработки последовательности вводимой с клавиатуры. Задание выберите в соответствии с вариантом.
6. В созданную ранее форму вставьте поле и организуйте вывод результата созданной функции в это поле.

Варианты задания

1. Определить максимальный элемент среди отрицательных.
2. Вычислить среднее арифметическое отрицательных чисел.
3. Определить количество чисел, значение которых кратно 3.
4. Вычислить сумму положительных четных чисел.
5. Определить минимальный элемент среди положительных.
6. Определить количество отрицательных чисел.
7. Определить среднее арифметическое положительных чисел.
8. Определить сколько четных чисел в последовательности.
9. Определить сумму максимального и минимального элементов последовательности.

10. Вычислить сумму чисел, значение которых кратно 5.

Лабораторная работа №10

Разработка средств вывода отчетов

Цель работы: научиться направлять вывод отчетов на разные устройства с помощью специальных форм.

Порядок выполнения

1. Откройте базу данных.

2. В разрабатываемых автоматизированных системах возникает необходимость просматривать и выводить на печать отчеты различного содержания. Используя группы переключателей в форме можно с легкостью создать программу, которая позволяет выбирать отчеты и направление их вывода. Для этого:

■ Создайте форму выбора отчета:

- 1) Для этого в окне базы данных раскройте вкладку **Формы** и нажмите кнопку *Создать*. Войдите в режим конструктора, не задавая таблицу или запрос, являющийся источником данных для формы.
- 2) Создайте группу переключателей, используя мастера групп, т.е. сначала создается группа, а в ней переключатели для каждого отчета. Введите имена отчетов в качестве надписей переключателей.
- 3) Создайте другую группу переключателей для печати и просмотра (можно еще и для вывода в файл).
- 4) Создайте кнопки **ОК** и **Отмена**.

Возможный внешний вид полученной в результате вышеприведенных действий формы показан ниже:

The screenshot shows a Windows application window titled "Выбор отчета". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main area of the window is divided into two vertical panels. The left panel is titled "Отчеты" and contains two radio button options: "Список студентов" and "По учебным". The right panel is titled "Вывод" and contains two radio button options: "Просмотр" and "Печать". To the right of these panels, there is a button labeled "ОК".



- Создайте процедуру обработки события *Нажатие кнопки* для кнопки **ОК**. Для этого в режиме конструктора перейдите к тексту программы, например, вызвав команда **Вид / Программа**. Процедуру нажатия кнопки **ОК** скорректируйте приблизительно следующим образом:

```
Private Sub Кнопка10_Click()  
Dim Nreport, Mode As Integer  
Dim strdocname As String  
Nreport = Группа2.Value  
Select Case Nreport  
Case 1  
    strdocname = "отчет1"  
Case 2  
    strdocname = "отчет2"  
Case 3  
    strdocname = "отчет3"  
End Select  
Mode= Группа10  
Select Case Mode  
Case 1    'просмотр  
    DoCmd.OpenReport strdocname, acPreview  
Case 2    'печать  
    DoCmd.OpenReport strdocname, acNormal  
End Select  
End Sub
```

При создании программы обратите внимание на номера групп в форме (можно было предварительно имя группам задать идентификаторами).

3. Сохранить программы в форме и проверить работу этой формы.

Лабораторная работа №11

Решение типовых задач с использованием Visual Basic

Цель работы: Научиться использовать Visual Basic для:

- использования переменных для решения некоторых типовых задач.
- управления объектами приложения Access.

Основная теория:

К классу объектов и семейств Microsoft Access относятся: семейство Forms, семейство Reports, объект DoCmd, объект Screen, составляющие объект Application.

Специальный объект DoCmd поддерживает методы, которые позволяют запускать большинство макрокоманд Access из программ Visual Basic. С помощью макрокоманд выполняются такие действия как закрытие окон, открытие форм и присвоение значений элементам управления. Ссылка на методы объекта DoCmd записывается в следующем виде:

[Application.]DoCmd.имя метода[аргументы...]

Наиболее часто используются методы OpenForm и OpenReport объекта DoCmd. Синтаксис метода OpenForm:

DoCmd.OpenForm имяФормы [, режим] [, имяФайла] [, условиеWhere] [, режимДанных] [, режимОкна] [, аргументыОткрытия]

Порядок выполнения

1. Откройте базу данных.
2. Организуйте поиск информации о студенте или сотруднике по фамилии. Для этого:
 - создайте **макрос** для поиска записи с фамилией, например, ИВАНОВ. Этот макрос должен содержать команды: 1) **ОткрытьФорму** для просмотра информации. 2) **НайтиЗапись** с образцом поиска "ИВАНОВ". 3) **ВыводНаЭкран** сообщения об отсутствии записи при условии, если запись с такой фамилией отсутствует (использовать функцию If, например : `If(Forms!имя_формы!ФИО = "ИВАНОВ", msgbox("yes"), msgbox("no"))`).
 - создайте в одной из форм **кнопку** и свяжите ее с созданным макросом. Проверьте работоспособность этой кнопки.

- Для поиска информации о **любом** студенте или сотруднике откройте форму с созданной кнопкой в режиме **конструктора**. Преобразуйте макрос в программу, выбрав команду **Сервис / Макросы / Преобразовать макросы**. Для упрощения текста программы отключите режим **Добавить программу обработки ошибок**. Откройте текст процедуры связанной с нажатием кнопки, используя команду **Вид / Программа** и выберите в качестве объекта просмотра кнопку, которая была связана с макросом.
- Скорректируйте программу, описав переменную строкового типа и организовав ввод фамилии с помощью функции InputBox. Замените в процедуре фамилию "ИВАНОВ" на имя переменной. Скорректировать фрагмент программы:

```
If (If(Forms!таблица1!FIO = имя_переменной, MsgBox("yes"), MsgBox("no")))_
Then
    DoCmd.Echo True, ""
End If
```

следующим образом:

```
result= If(Forms!таблица1!FIO = имя_переменной, "yes", "no")
if result="no" then MsgBox("Информации нет")
```

Не забудьте описать переменную result. Сохраните изменения в форме. Проверьте работу кнопки.

3. Организуйте проверку отсутствия записей при открытии формы для просмотра информации по заданному условию отбора. Для этого:
 - Создайте макрос для открытия нужной формы, задав в аргументах макрокоманды **ОткрытьФорму** имя формы и условие отбора, например, стипендия >100. Сохраните созданный макрос.
 - Подключите макрос к кнопке, созданной в другой форме и преобразуйте его в программу.
 - Проверьте работу кнопки. Если записей с заданным условием нет, то форма открывается с пустой записью.
 - Для организации выдачи сообщения можно использовать свойство RecordsetClone, в котором хранится ссылка на набор записей для формы, а в самом наборе имеется свойство RecordCount, содержащее число записей в результирующем множестве. Если значение свойства RecordCount равно 0, то должно выводиться сообщение. Для проверки значения этого свойства запишите инструкцию:

```
If Forms!имя_формы.RecordsetClone.RecordCount = 0 Then
    DoCmd.close acForm, имя_формы
```

```

MsgBox "записей, соответствующих заданному условию, не
найденно", _
VbInformation, "Нет данных"

End If

```

■ проверьте работу кнопки.

4. Откройте форму для просмотра информации о студентах или сотрудниках имеющих стипендию или зарплату выше средней. Для этого:

■ Откройте текст программы, созданной в предыдущем пункте и скорректируйте его следующим образом:

1) Вставьте в начало процедуры описание нескольких переменных: одну переменную - типа Object для организации цикла просмотра всех записей в таблице, и несколько переменных для вычисления средней стипендии или зарплаты.

2) После описаний с помощью объекта DoCmd откройте форму, например:
DoCmd.OpenForm "имя_формы"

3) Организуйте цикл просмотра записей таблицы следующим образом:
For Each c In Forms!имя_формы
kol=kol+1
summa = summa + Forms!имя_формы!стипендия
DoCmd.GoToRecord , "", acNext
Next

sred=summa/kol ' Вычисление средней стипендии

4) Далее в программе следует открытие формы с параметром **условиеWhere** задающим условие отбора: "[стипендия]>100". Скорректируйте условие отбора следующим образом: "[стипендия]>"&str(sred).

■ Сохраните программу , откомпилируйте ее и проверьте работу кнопки.

5. Продемонстрировать результаты преподавателю и закончить работу с Access.

ЗАДАНИЕ ДЛЯ КУРСОВОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ "БАЗЫ ДАННЫХ"

Разработать базу данных (БД) в соответствии с индивидуальным заданием. Процесс разработки должен включать следующие этапы.

1. Анализ предметной области

На данном этапе проектирования проводится постановка задачи, в процессе которой осуществляется анализ концептуальных требований,

предъявляемых к проектированию БД. Концептуальные требования выдвигаются разработчиком проекта (студентом) и утверждаются после собеседования преподавателем.

После определения перечня концептуальных требований и ограничений предметной области проводится выбор объектов предметной области и составляется универсальное отношение для проектируемой БД. Для полученного отношения проводится анализ диапазона значений каждого атрибута, и накладываются ограничения целостности.

2. Нормализация отношения

На этом этапе проектирования в универсальном отношении выявляются функциональные зависимости между атрибутами отношения, избыточность данных и возможные аномалии. На основании полученной информации проводится нормализация универсального отношения методом проекции с целью приведения отношения к совокупности отношений, находящихся в нормальной форме Бойса-Кодда.

3. Программное проектирование

На третьем этапе на основе информации, полученной в результате нормализации универсального отношения, разрабатывается структура таблиц БД, выбираются характеристики полей таблиц, устанавливаются связи между таблицами, создаются необходимые запросы, формы и отчеты, разрабатываются процедуры на языке Visual Basic.

СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЕ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ

1. Задание для курсовой работы
2. Описание концептуальных требований к БД.
3. Описание объектов, включенных в БД с обоснованием выбранных атрибутов.
4. Схема универсального отношения БД.
5. Описание ограничений целостности для атрибутов универсального отношения.
6. Описание функциональных зависимостей между атрибутами отношения, случаев избыточности данных и возможных аномалий.
7. Описание процесса нормализации универсального отношения с приведением всех промежуточных отношений.
8. Структура таблиц, входящих в БД.
9. Структура запросов, описание форм и отчетов.
10. Описание алгоритмов функционирования БД с иллюстрациями в виде распечаток копий экранов.
11. Листинги разработанных процедур.
12. Инструкции для работы с БД для конечного пользователя.