



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО  
ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»  
В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ**

**(Институт технологий (филиал) ДГТУ в г. Волгодонске)**



**УТВЕРЖДАЮ**  
Директор  
И.В. Столяр  
«26» апреля 2021 г.

**Лабораторный практикум**

**по дисциплине**

**«Информатика и информационно-коммуникационные технологии»**

**для обучающихся по направлению подготовки**

**15.03.05 Конструкторско-технологическое обеспечение**

**машиностроительных производств**

**профиль Технология машиностроения**

**2020 года набора**

Волгодонск  
2021

## **Лист согласования**

Лабораторный практикум по дисциплине «Информатика и информационно-коммуникационные технологии» составлен в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности) 15.03.05 Конструкторско-технологическое обеспечение машиностроительных производств

Рассмотрены и одобрены на заседании кафедры «ТСиИТ» протокол  
№ 10 от «26» апреля 2021 г.

## Содержание

Лабораторная работа № 1. Создание простейшей базы данных	4
Лабораторная работа № 2. Создание экранных форм для просмотра, ввода и редактирования	7
Лабораторная работа № 3. Создание кнопочной формы	11
Лабораторная работа № 4. Импорт и экспорт данных	16
Лабораторная работа № 5. Определение связей между таблицами и создание подчинённых форм	18
Лабораторная работа № 6. Поддержка целостности базы данных, имеющей связи «многие ко многим»	22
Лабораторная работа № 7. Формы для просмотра и ввода данных, имеющих связи «многие ко многим»	24
Лабораторная работа № 8. Формы для ввода связей «многие ко многим»	28
Лабораторная работа № 9 Главная (управляющая) форма информационной системы «Кафедра»	30
Лабораторная работа № 10. Создание запросов с помощью конструктора Access	33

## Лабораторная работа № 1. Создание простейшей базы данных

Цель работы: научиться создавать и использовать базу данных, состоящую из одной таблицы, практически познакомиться со всеми типами данных и некоторыми средствами поддержки целостности базы данных, используемыми в Access.

**Создание базы данных.** В Access базу данных можно создать двумя способами:

- при открытии выбрать в открывшемся окне переключатель «Новая база данных»,
- выбрать в меню Access пункт «Файл». Далее последовательно выбрать: пункт меню Создать ® вкладка Общие ® значок База данных.

Присвойте создаваемой базе данных имя «Кафедра.mdb» и поместите её в свой каталог.

**Создание таблицы.** После открытия или создания базы данных появляется окно базы данных с названием базы в верхнем левом углу. В расположенном слева списке объектов, выберите «Таблицы». Создайте в режиме конструктора таблицу «преподаватели».

Таблица должна состоять из девяти колонок (полей). Имена и характеристики полей приведены в табл. 1.

Список допустимых значений и ограничения, накладываемые на значения, позволяют значительно уменьшить количество ошибок при вводе данных, т.е. способствуют поддержанию *целостности* базы данных. Кроме того, список допустимых значений ускоряет ввод данных пользователем.

Ограничение на значение задаётся в «свойствах поля» в нижней части окна конструктора таблиц при выборе вкладки «Общие», а список допустимых значений – при выборе вкладки «Подстановка». При задании списка допустимых значений необходимо выбрать следующие значения свойств:

- для свойства «Тип элемента управления» – поле со списком или список,
- для свойства «Тип источника строк» – список значений.
- для свойства «Ограничиться списком» – да.

Таблица 1

Имя поля	Характеристика			
	Тип данных	Размер поля	Ограничение на значение	Список допустимых значений
Код_преп	Счётчик	Длинное целое		
ФИО	Текстовый	40		
Дата_рожд.	Дата/время		>#01.01.1910#	
Зарплата	Числовой	Длинное целое	<100000	
Должность	Текстовый	16		ассистент; ст.преподаватель; доцент; профессор
Степень	Текстовый	6		к.т.н.; к.э.н.; д.т.н.; д.э.н.
Звание	Текстовый	9		доцент; профессор
Биография	Поле МЕМО			
Фотография	Поле объекта OLE			

Значения перечисляются через точку с запятой в свойстве «Источник строк».

**Ввод данных.** Перейдите в режим таблицы. Для этого либо в меню, либо на панели инструментов выберите Вид ® Режим таблицы.

Присвойте таблице имя «преподаватели». Подтвердите создание ключевого поля.

Введите 5 записей с произвольными данными о преподавателях. При переходе к вводу следующей записи только что введённая запись автоматически сохраняется.

Обратите внимание на то, что счётчик (поле *Код\_преп*) автоматически увеличивается на единицу после ввода очередной записи. Значение счётчика невозможно исправить. Хранящееся в записи значение счётчика после удаления этой записи не используется. Максимальное значение счётчика в любой момент времени равно общему количеству введённых записей, включая удалённые.

Тип «Поле объекта OLE» служит для вставки в запись документов Word, электронных таблиц Excel и других объектов, или для связи с этими объектами. Для вставки нужно щёлкнуть правой кнопкой мышки по полю и в появившемся меню выбрать «Вставить объект». В появившемся окне выбрать переключатель «Создать из файла», щёлкнуть по кнопке «Обзор» и выбрать файл с рисунком из каталога, указанного преподавателем.

В поле МЕМО может находиться текст, имеющий длину до 64000 символов.

## Лабораторная работа № 2. Создание экранных форм для просмотра, ввода и редактирования

Цель работы: научиться создавать удобные экранные формы для трёх основных видов работ в информационной системе: просмотра, ввода и редактирования, научиться пользоваться этими формами, изучить различные права доступа к данным.

Кроме перечисленных видов работ к основным относится ещё и удаление, рассматриваемое в лабораторной работе № 3.

Пользователям информационных систем предоставляются разные права доступа к данным. Одним разрешается только просмотр, другим – просмотр и редактирование и т. д. Для удобства работы с данными и обеспечения прав собственности разрабатывают три вида форм: для *просмотра, ввода и редактирования*.

В формах для просмотра запрещено изменять данные, поэтому в них не нужно отображать списки допустимых значений. Так как при просмотре желательно видеть сразу много записей, то формы этого вида открывают в виде таблиц. Форму для редактирования удобно открывать сразу на записи, в которую нужно внести изменение, и отображать в ней списки допустимых значений .

Форма для ввода в момент начала ввода новой записи содержит пустые поля. Если набор вариантов вводимых данных небольшой, то следует пользоваться списками допустимых значений.

**Создание формы для просмотра.** В окне базы данных выберите объект «Формы» и пункт меню «Создать». Откроется окно «Новая форма». Выберите в верхней части окна пункт меню «Конструктор» и в нижней части окна укажите таблицу «преподаватели». Нажмите «ОК». Должны появиться 3 окна: форма, список полей и панель элементов. Список полей и панель элементов можно открыть через пункт меню Access «Вид» в верхней части экрана.

Перетащите мышкой из списка полей в форму поля «ФИО», «Дата\_рожд.», «Зарплата», «Биография» и «Фотография». Эти поля таблицы отобразятся в форме также в виде полей. Для поля «Фотография» будет создан элемент «Присоединённая рамка объекта».

При перетаскивании в форму поля «Должность» Access создаёт элемент «список», в котором будут показаны все допустимые значения должности. Так как при просмотре список не нужен, то для «Должности» нужно создать элемент «поле» вручную. Для этого щёлкните мышкой по элементу «поле» на панели элементов. Переместите указатель мышки в форму и раздвиньте поле до подходящих, на Ваш взгляд, размеров. Введите вместо «поле0» название поля. Теперь нужно привязать поле формы к полю «Должность» таблицы «преподаватели». Щёлкните правой кнопкой мышки по создаваемому полю в форме. В появившемся меню выберите пункт «Свойства». Появится окно свойств поля. Выберите вкладку «Данные» и в свойстве «Данные» выберите «Должность».

Аналогично создайте поля для «Степени» и «Звания».

Для ввода названия формы откройте раздел формы «Заголовок», выполнив пункты меню Access «Вид» ® «Заголовок/примечание формы». Название формы должно выглядеть примерно так:

## ПРЕПОДАВАТЕЛИ

(форма для просмотра)

В форме, предназначенной только для просмотра, необходимо запретить пользователю вносить какие-либо изменения в данные. Щёлкните правой кнопкой мышки по квадратику в левом верхнем углу формы и в раскрывшемся меню выберите пункт «Свойства». Должно открыться окно «Форма». Выберите вкладку «Данные». Установите для свойства «Тип набора записей» значение «Статический набор».

Отредактируйте по своему вкусу названия полей и формы.



Перейдите в режим формы и проверьте её работу. Форму можно просматривать в трёх режимах:

- простая форма,
- ленточная форма,
- таблица.

Чаще всего при просмотре используется режим «Таблица». Просмотрите форму во всех трёх режимах. Режимы просмотра устанавливаются Конструктором. В режиме конструктора откройте окно свойств формы и выберите вкладку «Макет». Подберите нужные значения свойств «Режим по умолчанию» и «Допустимый режим». Режим «Ленточная форма» не может использоваться при включении в форму подчинённых форм.

**Создание формы для редактирования.** Воспользуйтесь помощью мастера. В окне базы данных выберите объект «Формы» и затем – «Создание формы с помощью мастера». В появившемся окне «Создание формы» выберите таблицу «преподаватели» и все поля за исключением поля «код\_преп». Следуйте далее указаниям мастера. Форму назовите «преп\_ред». Если при просмотре Вам не понравится вид формы, то перейдите в режим конструктора и внесите изменения в свойства либо всей формы, либо отдельных элементов.

Для проверки работы формы внесите изменения в данные о преподавателях.

**Создание формы для ввода новых записей.** Форма для ввода отличается от формы для редактирования только значением свойства «Ввод данных».

Скопируйте форму «преп\_ред». Для этого в окне базы данных щёлкните правой клавишей мышки по названию формы «преп\_ред» и выберите «копировать», затем на любом свободном месте окна базы данных снова щёлкните правой клавишей мышки и выберите «вставить». Назовите форму «преп\_ввод». Откройте форму «преп\_ввод» в режиме конструктора и

измените значение свойства «Ввод данных» на «да». Значение «да» переводит форму в режим, в котором не видны уже имеющиеся в таблице записи, все поля формы пустые. После ввода данных очередной записи и перехода к следующей введённая запись автоматически заносится в базу и форма готова к вводу новой записи.

Удобно заполнять поля, когда они расположены друг под другом. Перетащите мышкой поля и их названия так, чтобы каждое поле со своим названием (кроме фотографии) располагалось в отдельной строке. Фотографию можно расположить справа от остальных полей.

Перейдите в режим формы и введите данные о двух-трёх преподавателях.

### Лабораторная работа № 3. Создание кнопочной формы

Цели работы:

- научиться создавать форму, с помощью которой множество объектов (формы, отчёты, запросы) объединяются в единую, управляемую пользователем систему;
- научиться создавать макросы и процедуры, вызываемые щелчком по кнопке или по полю;
- познакомиться с безопасными приёмами удаления записей из базы.

Вся работа пользователя с информационной системой организуется через главную, как правило, кнопочную форму. Создадим форму (рис. 1), из которой будут вызываться созданные ранее формы.

**Создание формы, не связанной с таблицей базы данных.** Создайте форму в режиме конструктора, не указывая источник данных.



Рис. 1. Пример формы

**Создание поля со списком для выбора записи.** Создайте поле со списком преподавателей. Из этого списка пользователь сможет выбрать преподавателя, сведения о котором нужно просмотреть. Выберите на панели

элементов поле со списком и поместите его в форму. Подтвердите, что поле со списком будет использовать значения из таблицы или запроса и щёлкните по кнопке «далее». Выберите таблицу «преподаватели» и снова щёлкните по кнопке «далее».

Для отображения в поле со списком нужно кроме поля «ФИО» выбрать ключевое поле «код\_преп» и показывать в форме при просмотре списка только «ФИО». Все значения ключевого поля по определению оригинальны. Люди с одинаковыми фамилиями и инициалами иногда встречаются. Если указать для отображения в списке только поле «ФИО», то Access (и другие СУБД) всегда будет выбирать из таблицы первую по порядку из двух записей, имеющих одинаковые значения поля «ФИО». Если в строке списка есть и «код\_преп» и «ФИО», то из таблицы будет выбрана запись, содержащая значение поля «код\_преп».

Для быстрого выбора из длинного списка можно в поле ввести первую букву нужной фамилии.

**Создание кнопки для просмотра данных обо всех преподавателях.** Если нужны данные о многих преподавателях, то целесообразно открыть форму для просмотра в режиме таблицы. Создадим кнопку и связанную с ней процедуру, открывающую форму в режиме таблицы.

Выберите на панели элементов кнопку и поместите её в форму. В открывшемся окне «Создание кнопок» выберите категорию «Работа с формой» и действия «Открыть форму». Далее выберите форму «преп\_просм», далее – переключатель «Открыть форму и показать все записи». далее – переключатель «Текст» и наберите в поле надпись на кнопке «Просмотр всех преподавателей». Задайте имя кнопки «откр\_таб\_преп». Созданная кнопка будет открывать «преп\_просм» в режиме формы.

Для того чтобы форма «преп\_просм» открывалась в режиме таблицы, внесите изменения в процедуру, которую Access автоматически создал вместе с кнопкой. Для этого в режиме конструктора откройте окно свойств кнопки «откр\_таб\_преп» и в нём вкладку «События». Щёлкните мышкой по

свойству «Нажатие мышки», а затем щёлкните по квадратной кнопке с многоточием, расположенной справа. Откроется окно Visual Basic. В процедуре «Откр\_таб\_преп\_Click()» в строчку

```
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

вставьте после первой запятой слово «acFormDS» (это параметр, задающий открытие формы в режиме таблицы). Строка примет вид:

```
DoCmd.OpenForm stDocName, acFormDS, , stLinkCriteria
```

Эта строка содержит команду на открытие формы. Значения «слов» в строке следующие:

- **DoCmd** – выполнить команду;
- **OpenForm** – команда «открыть форму»;
- **StDocName** – переменная, содержащая имя формы.

Через запятую перечислены параметры команды.

Закройте окно Visual Basic и проверьте работу кнопки.

**Создание кнопок для просмотра и редактирования данных об определённом преподавателе.** Создадим кнопку, с помощью которой будет открываться форма для просмотра данных об одном преподавателе. ФИО преподавателя выбирается из поля со списком.

Выберите на панели элементов кнопку и поместите её в форму. В открывшемся окне «Создание кнопок» выберите категорию «Работа с формой» и действия «Открыть форму». Далее выберите форму «преп\_просм», далее – переключатель «Открыть форму для отобранных записей».

В следующем окне нужно указать, что в открываемой форме «преп\_просм» будет показана запись из таблицы «преподаватели», содержащая значение поля «код\_преп» (поля из таблицы базы данных), равное выбранному из поля со списком в форме «кафедра».

Далее выберите переключатель «Текст» и наберите в поле надпись на кнопке «Просмотр одного преподавателя». Задайте имя кнопки «откр\_один\_преп».

Кнопка для редактирования данных о преподавателе создаётся аналогично. Отличие состоит только в названиях кнопки и открываемой формы. Назовите кнопку «Редактирование и удаление», так как в дальнейшем в форме для редактирования будет кнопка для удаления записи .

Кнопка для ввода записи создаётся аналогично двум предыдущим, только не нужно связывать поля и исправлять процедуру. Выберите переключатель «Открыть форму и показать все записи», надпись на кнопке «Ввод преподавателя». Задайте имя кнопки «ввод\_преп».

**Кнопка для удаления записи.** Чтобы уменьшить возможности ошибиться при удалении записи, необходимо:

- в момент нажатия кнопки удаления видеть удаляемую запись,
- после нажатия кнопки для удаления записи пользователь должен подтвердить приказ об удалении.

Создайте кнопку для удаления записи в форме для редактирования. Выберите категорию «Обработка записей» и действие «Удалить запись». На кнопке напишите «Удаление». Попробуйте удалить одну запись.

Внимание! При вводе новых записей номер удалённой записи не используется. В последовательности номеров образуется неустранимый разрыв.

На рис. 1 в форме «кафедра» изображена необязательная кнопка «Удаление». Для её работы используется следующая процедура:

```
Private Sub КнопкаУд_Click()  
    Dim N_record As String  
    Dim dbs As Database  
    a = MsgBox("Удалять?", vbYesNo)  
    If a = vbNo Then  
        Exit Sub
```

```

Else
    N_record = Me![ПолеСоСписком0]
    Set dbs = CurrentDb
    dbs.Execute "DELETE FROM преподаватели WHERE код_преп=" &
N_record
End If
End Sub

```

Процедурой *КнопкаУд\_Click()* удаляется из таблицы преподаватели запись, код (поле «код\_преп») которой равен значению кода преподавателя, выбранному из поля со списком в форме «кафедра» (Me![ПолеСоСписком0]). Следует отметить, что перед удалением записи пользователь должен видеть всё её содержимое. Поэтому лучше кнопку для удаления поместить только в форме для просмотра или в форме для редактирования.

*Самостоятельно* создайте кнопки для выхода из форм для просмотра, редактирования и ввода.

## Лабораторная работа № 4. Импорт и экспорт данных

Цель работы: научиться загружать в базу из внешних источников большие объёмы данных и сбрасывать из базы данные в формате других приложений.

В качестве приложения, с которым будет взаимодействовать Access, выберем электронные таблицы Excel. Откройте файл «группы.xls». Ознакомьтесь с листами «группы» и «факультеты».

**Импорт в новую таблицу.** Импортируйте из файла «группы.xls» в новую таблицу данные с листа «группы». Последовательность действий: Меню ® Файл ® Внешние данные ® Импорт. Далее выберите файл «группы.xls» и следуйте указаниям Access. Наименования полей оставьте такими же, как в таблице Excel, выберите автоматическое создание ключа, новую таблицу в базе данных назовите «Группы».

**Импорт в существующую таблицу.** Создайте в базе данных таблицу «факультет» следующей структуры:

Имя поля	Тип данных	Размер поля
Код_фак	Счётчик	
Факультет	Текст	50
Телефон	Текст	10
Декан	Текст	30
Секретарь	Текст	30

Сделайте поле «Код\_фак» ключевым. Введите первую запись следующего содержания:

- Факультет – Юридический,
- Телефон - 167,
- Декан - Семёнов А.Б.Э
- Секретарь – Никитина А.Г.

Импортируйте в таблицу «факультет» данные с листа ««факультеты»» из файла «группы.xls». Последовательность действий:



Меню ® Файл ® Внешние данные ® Импорт. Далее выберите файл «группы.xls» и следуйте указаниям Access. Наименования соответствующих полей в таблицах Excel и Access должны совпадать.

**Установка связи с таблицей в формате Excel.** Можно работать с таблицей Excel так же, как и с таблицами базы данных Access, если установить с ней связь.

Установите связь с таблицей Excel, расположенной на листе «Список» в файле (книге) «Сортировка1.xls». Последовательность действий: Меню ® Файл ® Внешние данные ® Связь с таблицами. Далее выберите файл «Сортировка1.xls» и следуйте указаниям Access.

**Экспорт данных из базы в таблицу Excel.** Экспортируем таблицу «Группы». Откройте таблицу «Группы» в Access. В меню Access выберите Файл ® Экспорт. В нижней части открывшегося окна выберите тип файла «Microsoft Excel» версии, установленной на Вашем компьютере, например, «Microsoft Excel 97-2000 (\*.xls)». Назовите файл «ГруппыАсс» и сохраните его в Вашем каталоге.

**Кнопки для экспорта и импорта данных.** В Access есть макрос «ПреобразоватьЭлектроннуюТаблицу», с помощью которого можно импортировать и экспортировать данные. Создайте форму для экспорта таблицы «Группы».

Создайте кнопку «экспорт в Excel». В окне «Создание кнопки» нажмите «Отмена». Вызовите свойства кнопки. Щёлкните по свойству «нажатие кнопки». Справа появятся две квадратные кнопочки. Щёлкните по кнопочке с многоточием и в появившемся меню выберите «Макросы» Дайте любое название макросу. Выберите макрокоманду «ПреобразоватьЭлектроннуюТаблицу» и присвойте ей необходимые значения аргументов (в нижней части окна). Файл с экспортированной таблицей поместите в свой каталог

Проверьте работу созданной формы.

## Лабораторная работа № 5. Определение связей между таблицами и создание подчинённых форм

Цель работы: научиться устанавливать связи между таблицами и строить подчинённые формы.

Между реальными объектами, данные о которых хранятся в базе, могут существовать логические связи. Например, многие группы студентов входят в состав одного факультета, и ни одна группа не может относиться сразу к двум факультетам. Говорят, что между факультетом и группой имеется связь «один ко многим».

**Отображение связи «один ко многим».** В режиме конструктора дополните структуру таблицы «Группы» полем «код\_ф», затем перейдите в режим просмотра таблицы и распределите группы по факультетам, то есть введите в поле «код\_ф» коды факультетов, хранящиеся в поле «код\_фак» таблицы «факультет».

**Подмена кода объекта именем.** Access позволяет во время просмотра данных в таблице «Группы» подменять код факультета его названием, взятым из таблицы «факультет». Для этого откройте таблицу «Группы» в режиме конструктора, выберите в свойствах поля «код\_ф» вкладку «Подстановка» и установите следующие значения свойств:

- для свойства «Тип элемента управления» - Поле со списком,
- для свойства «Тип источника строк» - Таблица или запрос,
- для свойства «Источник строк» - факультет,
- для свойства «Присоединённый столбец» - 1,
- для свойства «Число столбцов» - 2,
- для свойства «Ширина столбцов» – 0,
- для свойства «Ограничиться списком» – да.

Теперь перейдите в режим таблицы. В поле «код\_ф» должны появиться названия факультетов.

**Внесение связей между таблицами в схему данных.** В Access можно создать схему данных, в которой явно указываются связи между полями разных таблиц. Схема данных используется для поддержки целостности базы данных и, кроме того, упрощает построение форм и запросов.

Закройте таблицу «Группы». Выполните пункты меню Сервис ® Схема данных. В любом месте открывшегося окна щёлкните правой кнопкой мышки. В появившемся меню выберите пункт «Добавить таблицу» и добавьте таблицы «Группы» и «факультет».левой кнопкой мышки соедините поле «код\_фак» и поле «код\_ф» связываемых таблиц. Должно появиться окно «Изменение связей».

Установите флажок «Обеспечение целостности данных». Теперь, если не устанавливать два других флажка, Access при попытке удалить из таблицы «факультет» любую запись будет проверять, нет ли в таблице «Группы» кода удаляемой записи. Если такой код есть, то появится сообщение об ошибке. Другими словами, пока в базе данных указано, что на факультете есть хотя бы одна группа, удалить этот факультет нельзя.

Флажок «Каскадное удаление связанных записей» служит для автоматического удаления из таблицы «Группы» всех записей, связанных с записью, удаляемой из таблицы «факультет».

Флажок «Каскадное обновление связанных полей» служит для автоматического обновления в таблицы со стороны «многие» значения связанного поля, обновляемого в таблице со стороны «1». В нашем случае связанное поле – счётчик и оно не может обновляться.

После закрытия окна «Изменение связей» на схема данных появится связь «один ко многим» между таблицами «факультет» и «Группы» (рис. 2).

**Создание подчинённой формы.** Создайте новую форму. Назовите её «факультеты». Отобразите в ней все поля, кроме «код\_фак».



Рис. 2. Схема данных

Теперь в форме «факультеты» нужно создать подчинённую форму, в которой будут показываться данные о группах, входящих в состав показываемого в главной форме факультета. Раздвиньте границы формы «факультеты» так, чтобы в ней поместилась таблица с данными о группах. На панели элементов найдите элемент «Подчинённая форма/отчёт» и установите его в форме «факультеты».

Ном_групп	Специальность	Курс
124	теория вероятностей	1
217	теория вероятностей	2
321	теория вероятностей	3
451	анализ	2

Рис.3. Форма *факультет* с подчинённой формой

В окне мастера подчинённых форм установите переключатель «Имеющиеся таблицы и запросы» и щёлкните по кнопке «Далее». В следующем окне мастера подчинённых форм выберите таблицу «группы» и все поля, кроме «Код\_гр». Выберите переключатель «Выбор из списка» и оставьте предложенное название подчинённой формы. На рис. 3 показана форма «факультет» с подчинённой формой, в которой выведены группы,

входящие в состав просматриваемого в основной форме факультета.  
Перейдите в режим формы и просмотрите несколько записей.

Лабораторная работа № 6. Поддержка целостности базы данных,  
имеющей связи «многие ко многим»

Цель работы: научиться создавать вспомогательные таблицы для сведения одной связи «многие ко многим» к двум связям «один ко многим» и обеспечения целостности данных.

В данной работе нужно создать таблицу «дисциплины», вспомогательную таблицу «преп\_дис» и схему связей между ними и ввести данные в созданные таблицы.

Пример объектов, имеющих связи «многие ко многим» - преподаватели и предметы (дисциплины). Один преподаватель может вести несколько дисциплин, и одну дисциплину могут вести в разных группах разные преподаватели. Наиболее рациональный способ хранения данных о связях «многие ко многим» - это создание вспомогательной таблицы, в каждой записи которой хранятся код преподавателя и код предмета. Между основными таблицами и вспомогательной устанавливаются связи «один ко многим». Вспомогательной таблицы позволяет избежать избыточности данных в базе, уменьшить количество ошибок при вводе и упростить запросы к базе данных.

**Создание таблиц.** Создайте таблицу «дисциплины», имеющую следующую структуру:

№ п.п.	Имя поля	Тип данных	Размер поля
1	код_дис	Счётчик	Длинное целое
2	предмет	Текстовый	40
3	лекции	Числовой	Целое
4	практика	Числовой	Целое

Сделайте поле «код\_дис» ключевым. В полях «лекции» и «практика» хранится количество учебных часов, отводимых на лекции и практику соответственно. Введите в таблицу «дисциплины» несколько записей.

Создайте таблицу «преп\_дис», имеющую следующую структуру:

№ п.п	Имя поля	Тип данных	Размер поля
1	код_преп_дис	счётчик	Длинное целое
2	код_преп	Числовой	Длинное целое
3	код_дис	Числовой	Длинное целое

Введите в таблицу «преп\_дис» несколько записей, связывающих преподавателей и дисциплины. На рис. 4 показан пример, в котором преподаватель Андреев В.К. ведёт Pascal, информатику и Fortran, а Семёнова Е.М. ведёт информатику и Pascal.

**Внесение связей в схему данных** аналогично внесению связей между таблицами «Группы» и «факультет» в предыдущей лабораторной работе. Только с таблицей «преп\_дис» связаны две таблицы (рис. 5).

Лабораторная работа № 7. Формы для просмотра и ввода данных,  
имеющих связи «многие ко многим»

Цели работы – научиться создавать:

- подчинённые формы для просмотра данных из основных таблиц с использованием вспомогательных,
- экранные формы, с помощью которых при вводе пользователем данных, имеющих связи «многие ко многим», автоматически заполняется вспомогательная таблица (см. лаб. работу №6) со связями «один ко многим».

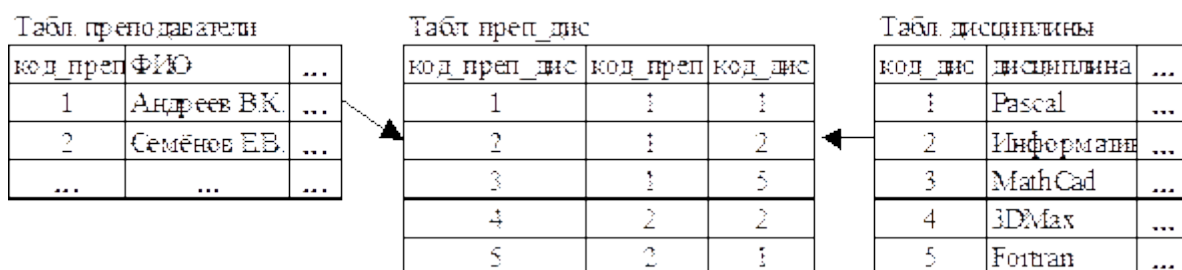


Рис. 4. Связывание данных с помощью вспомогательной таблицы

**Дополнение формы «преп\_просм» подчинённой формой с данными о предметах, которые ведёт преподаватель.** Откройте форму «преп\_просм» в режиме конструктора. На панели элементов найдите элемент «Подчинённая форма/отчёт» и установите его в открытой форме.

Далее описываются два способа создания подчинённой формы со связями «многие ко многим»:

- с помощью мастера подчинённых форм,
- с использованием запроса к базе данных.

. С помощью мастера создать подчинённую форму проще, но при отсутствии на компьютере мастера необходимо использовать запрос к базе данных.



**Создание подчинённой формы с помощью мастера.** В окне мастера подчинённых форм установите переключатель «Имеющиеся таблицы и запросы» и щёлкните по кнопке «Далее».

В следующем окне мастера подчинённых форм выберите

- таблицу «дисциплины» и в ней все поля,
- таблицу «преп\_дис» и в ней поля «код\_преп» и «код\_дис».

Перейдите в следующее окно и выберите переключатель «Выбор из списка» и оставьте предложенное название подчинённой формы.

Перейдите в режим формы и просмотрите несколько записей.



Рис. 5. Схема данных с использованием вспомогательной таблицы

**Создание подчинённой формы с использованием запроса к базе данных.** Сначала нужно создать запрос на выборку из базы данных. (Подробно методы создания запросов рассматриваются во второй части данного пособия.) Откройте окно базы данных. В левой части окна выберите пункт меню «Запросы». В верхней части окна выберите вкладку «Создать». Создайте запрос в режиме конструктора.

В окне «Добавление таблицы» выберите таблицы «преподаватели», «дисциплины» и «преп\_дис». Закройте окно «Добавление таблицы». окне «запрос на выборку» Перетащите мышкой поля из списков полей таблиц расположенных в верхней части окна «запрос на выборку», в нижнюю часть так , как это показано на рис. 6.

Выполните запрос с помощью пунктов меню Запрос ® Запуск. Результат запроса – новая (временная) таблица, состоящая из записей,

содержащих все допустимые комбинации значений полей из трёх исходных таблиц. Закройте окно запроса. Назовите запрос «преп\_дисц».

Откройте форму «преп\_просм» в режиме конструктора. Установите элемент «Подчинённая форма/отчёт» в форму «преп\_просм». Отмените использование мастера подчинённых форм (или закройте окно с сообщением о том, что мастер не установлен).

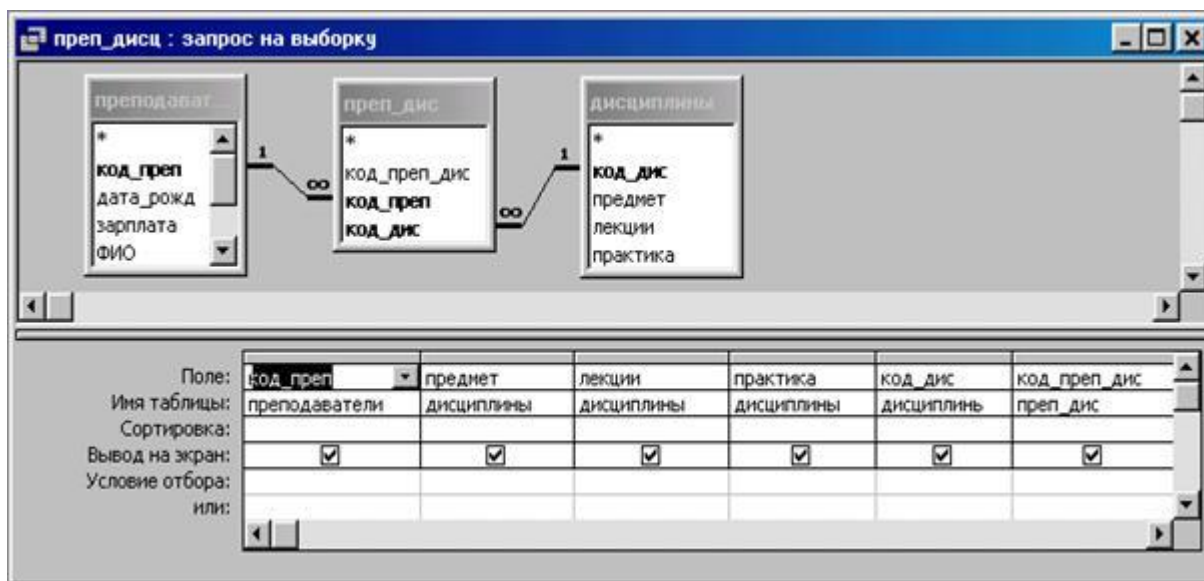


Рис. 6. Запрос, сформированный средствами Access


Подведите указатель мышки к квадратику в верхнем левом углу подчинённой (внедрённой) формы и щёлкните правой кнопкой. В появившемся окне выберите пункт «Свойства». Появится окно «Форма/отчёт». Выберите в нём вкладку «Данные». Установите для свойства объект-источник значение «Запрос.преп\_дисц». Перейдите в режим просмотра формы. Примерный вид формы показан на рис. 7.

**Дополнение формы «преп\_ред» подчинённой формой с данными о предметах, которые ведёт преподаватель.** Любым из двух способов, описанных выше в данной лаб. работе, вставьте в форму «преп\_ред» подчинённую форму.

преп\_просм : Форма

## ПРЕПОДАВАТЕЛИ (форма для просмотра)

дата рождения: 01.01.51  
 ФИО: Бородина О.А.  
 Должность: профессор  
 Зарплата: 5700  
 Степень: д.ф.н.  
 Звание: профессор



**Дисциплины**

	предмет	лекции	практика
▶	Линейная алгебра	34	34
▶	Теория множеств	68	34

**биография:**  
 С детства любила шляпки и математику

Запись: 1 из 2

Запись: 2 из 5

Закреть форму

Рис. 7. Форма для просмотра данных о преподавателях

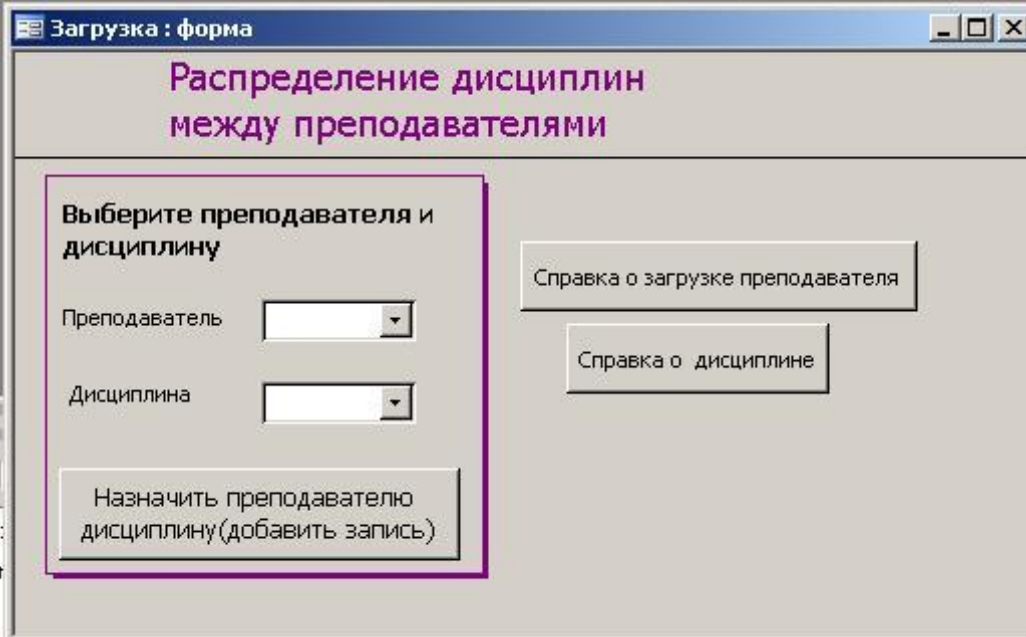
Данные о дисциплинах используются для формирования учебной нагрузки многих преподавателей, поэтому редактировать их нужно в специальной форме, а в подчинённой форме их редактирование нужно запретить. Для этого установите свойство подчинённой формы «доступ» в состояние «нет».

Самостоятельно создайте формы для ввода, просмотра и редактирования данных о дисциплинах, аналогичные формам для работы с данными о преподавателях. Эти формы понадобятся для построения информационной системы «Кафедра». Назовите формы «дис\_ввод», «дис\_просм» и «дис\_ред».

## Лабораторная работа № 8. Формы для ввода связей «многие ко МНОГИМ»

Цель работы: создать форму для автоматического заполнения вспомогательной таблицы, хранящей только коды записей двух таблиц со связей «многие ко многим».

Предполагается, что при эксплуатации создаваемой формы (рис. 8) пользователь, распределяющий нагрузку преподавателей, будет указывать преподавателя и ведомую им дисциплину, а СУБД Access, исходя из этих данных, будет создавать в таблице «преп\_дис» запись с кодами выбранных преподавателя и дисциплины.



The image shows a screenshot of a Microsoft Access form window. The title bar reads "Загрузка : форма". The main heading of the form is "Распределение дисциплин между преподавателями". On the left side, there is a section titled "Выберите преподавателя и дисциплину" which contains two dropdown menus: "Преподаватель" and "Дисциплина". Below these menus is a button labeled "Назначить преподавателю дисциплину(добавить запись)". To the right of the dropdown menus, there are two buttons: "Справка о загрузке преподавателя" and "Справка о дисциплине".

Рис.8. Форма для ввода учебной загрузки преподавателей

Кнопки «Справка о . . .» нужны для того, чтобы открыть формы для просмотра преподавателей и дисциплин и определить, достаточно ли загружен преподаватель и сколько преподавателей уже преподают данную дисциплину.

Создайте форму в режиме конструктора. Выберите таблицу «преп\_дис» в качестве источника записей. Укажите, что форма предназначена для ввода.

Создайте поле со списком для выбора преподавателя. Для этого выберите на панели элементов поле со списком и поместите его в форму. Подтвердите, что поле со списком будет использовать значения из таблицы или запроса и щёлкните по кнопке «далее». Выберите таблицу «преподаватели» и снова щёлкните по кнопке «далее».

Включите в поле со списком поля «код\_преп» и «ФИО». Подтвердите, что нужно скрыть ключевой столбец. Далее нужно указать, что выбранное из таблицы «преподаватели» значение поля «код\_преп» нужно сохранить в поле «код\_преп» создаваемой записи в таблице «преп\_дис».

Создайте поле со списком для выбора дисциплины. Все действия такие же, как и при создании предыдущего поля со списком. Только нужно выбрать таблицу «дисциплины» и поле «код\_дис».

Кнопки создаются так же, как и в лаб. работе № 3.

## Лабораторная работа № 9 Главная (управляющая) форма информационной системы «Кафедра»

Цель работы: объединить в единую систему с помощью управляющей формы все объекты созданные в лабораторных работах №№ 1, 2, 3, 6, 7, 8. Управляющая форма позволит пользоваться информационной системой, не зная ни структуры базы данных, ни названий форм.

В предыдущих лабораторных работах были созданы таблицы и формы для ввода, просмотра, редактирования и удаления данных о преподавателях и дисциплинах кафедры, а также данных об учебной загрузке преподавателей.

**Главная форма.** В лабораторной работе № 3 была разработана кнопочная форма (рис. 1). Необходимо дополнить её средствами для работы с дисциплинами и распределения загрузки преподавателей (рис.9).

**Поддержка целостности при вводе и редактировании данных.** Для предотвращения ввода по ошибке одних и тех же данных несколько раз в базе данных используются ключи. Ключом называются один или несколько столбцов (атрибутов) таких, что в таблице не найдётся ни одной пары строк с одинаковым набором данных в выбранных в качестве ключа столбцах.

В теории баз данных принято все ключи таблицы называть потенциальными. Один ключ выделяют по каким-либо соображениям и называют *первичным*, а остальные – *альтернативными*. В Access первичный ключ называют просто ключом. Для того, чтобы в Access установить контроль за уникальностью значений в каком-либо поле, нужно это поле назначить индексированным и указать, что совпадения в нём не допускаются.



Рис. 9. Главная форма информационной системы «Кафедра»

В таблице «дисциплины» два потенциальных ключа: «код\_дис» и «предмет». Поле «код\_дис» назначено (первичным) ключом. Поэтому необходимо сделать столбец «предмет» индексированным с запретом повторяющихся значений.

Откройте таблицу «дисциплины» в режиме конструктора. Установите курсор на поле «предмет» и выберите для свойства «индексированное поле» значение «Да (Совпадения не допускаются)».

В таблице «преп\_дис» нельзя допустить повторения пары значений «код\_преп» и «код\_дис». Нужно создать *составной ключ* из этих полей. Откройте таблицу «преп\_дис» в режиме просмотра и убедитесь в том что ни одному преподавателю не назначена дважды одна дисциплина. Перейдите в режим конструктора и выделите поля «код\_преп» и «код\_дис». Щёлкните по значку с изображением ключа на панели инструментов.

Теперь при попытке повторно ввести данные в таблицы «дисциплины» и «преп\_дис» Access будет сообщать об ошибке.

**Комплексная отладка информационной системы.** После создания информационной системы необходимо убедиться в правильности её работы.

В процессе отладки необходимо имитировать условия работы реального пользователя системы.

Проверьте правильность работы всех элементов главной формы.

Введите через форму для ввода данные о трёх-четырёх преподавателях.

Введите через форму для ввода данные о пяти-шести дисциплинах.

Через форму «загрузка» распределите учебную нагрузку между преподавателями. Убедитесь в том, что попытка дважды назначить одному преподавателю один и тот же предмет вызывает сообщение об ошибке.

Проверьте работу форм для просмотра и редактирования.

Удалите с помощью кнопок удаления в формах редактирования несколько записей о преподавателях и дисциплинах.

Исправьте обнаруженные ошибки, добавьте недостающие формы и элементы.



## Лабораторная работа № 10. Создание запросов с помощью конструктора Access

В большинстве запросов, рассматриваемых во II части, будут использоваться таблицы «Заказы», «Заказано» и «Товары», взятые из учебной базы данных «Борей». Импортируйте эти таблицы в свою базу данных. Для этого выберите последовательно пункты меню Access Файл ® Внешние данные ® Импорт.

Цель работы: научиться создавать с помощью конструктора одно- и многотабличные запросы на выборку записей с заданным набором полей и удовлетворяющие заданным условиям.

Во всех современных СУБД запросы пишутся на языке SQL (Structured Query Language – язык структурированных запросов). В ACCESS также есть возможность писать запросы на SQL, но разработчики ACCESS ориентируют пользователя на максимальное использование средств автоматизации для создания и ведения баз данных. К этим средствам относится конструктор запросов, с помощью которого можно быстро создать многие запросы.

**Запрос на просмотр всех данных одной таблицы.** (Этот запрос неявно генерируется СУБД Access при открытии таблицы в режиме «таблица».) Для создания запроса откройте окно базы данных. Выберите объект «Запросы». В верхней части окна выберите вкладку «Создать». Создайте запрос в режиме конструктора.

В окне «Добавление таблицы» выберите таблицу «Заказы». Закройте окно «Добавление таблицы». В окне «запрос на выборку» Перетащите символ «\*» (звёздочка) из списка полей таблицы «Заказы» в крайнее левое поле в нижней части окна. После этого запрос готов. Выполните его, выбрав в меню пункты Запрос ® Запуск.

Для лучшего понимания дальнейших упражнений внимательно просмотрите содержимое таблицы «Заказы».

Во всех современных СУБД запросы пишутся на языке SQL. Просмотрите созданный запрос в режиме SQL (пункты меню Вид ® Режим SQL). Запрос имеет вид

```
SELECT Заказы.*  
FROM Заказы;
```

Запрос читается так: выбрать (SELECT) все поля (Заказы.\*) из таблицы (FROM) Заказы.

**Запрос на просмотр всех записей с заданным набором полей.** Измените предыдущий запрос. Замените символ «\*» именем поля «Клиент». Добавьте в запрос поля «ДатаИсполнения», «СтоимостьДоставки» и «НазваниеПолучателя». Просмотрите созданный запрос в режиме таблицы и в режиме SQL.

**Выборка из таблицы записей, удовлетворяющих заданным условиям.** Нужно выбрать из таблицы «Заказы» все заказы, у которых стоимостью доставки не меньше 35 р. и меньше 40 р. Для этого внесите изменения в предыдущий запрос. В условиях отбора поля «СтоимостьДоставки» запишите «>=35 AND <40». *Здесь и далее кавычки «» в условие не входят.*

Выполните запрос. Должно быть отобрано 22 заказа (записи).

Просмотрите созданный запрос в режиме SQL. Обратите внимание на условие после ключевого слова WHERE.

Добавьте в запрос ещё одно условие. В условия отбора поля «НазваниеПолучателя» запишите «Like 'R\*'», означающее «выбрать названия получателей, начинающиеся на R».

**Многотабличные запросы.** Связанные между собой данные, хранящиеся в нескольких таблицах можно выбирать одним запросом.

В таблицах «Заказы» и «Заказано» хранятся данные о заказах. Таблицы связаны с помощью поля «КодЗаказа». В одном заказе заказывается несколько товаров. В таблице «Заказы» хранятся общие сведения о заказе, а в

таблице «Заказано» – сведения о заказанных товарах из этого заказа. Тип связи между таблицами «Заказы» и «Заказано» – один ко многим. Такая же связь установлена между таблицами «Товары» и «Заказано» через поле «КодТовара».

Создайте запрос на выборку всех марок товаров, заказанных клиентом ANTON. Для этого нужно в режиме конструктора запросов

- выбрать все три таблицы «Заказы», «Заказано» и «Товары»,
- из таблицы «Заказы» выбрать поле «КодКлиента», установить для него условие отбора «ANTON» и запретить его вывод на экран,
- из таблицы «Товары» выбрать поле «Марка» и сортировку по возрастанию,
- чтобы одна марка товара не была выбрана несколько раз, установить свойство запроса «уникальные значения» в положение «да».

В режиме SQL запрос, сгенерированный Access, выглядит следующим образом:

```
SELECT DISTINCT Товары.Марка
FROM Товары INNER JOIN (Заказы INNER JOIN Заказано ON Заказы.КодЗаказа = Заказано.КодЗаказа) ON Товары.КодТовара = Заказано.КодТовара
WHERE (((Заказы.КодКлиента)="ANTON"))
ORDER BY Товары.Марка;
```