

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
НАЦИОНАЛЬНАЯ МЕТАЛЛУРГИЧЕСКАЯ АКАДЕМИЯ УКРАИНЫ**

Г. Г. Швачич, Г. И. Рыжанкова, В. В. Кузьменко, В. П. Барвинов

**ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ В СУБД ACCESS
Часть 1**

Утверждено на заседании Ученого совета академии
как учебное пособие

Днепропетровск НМетАУ 2006

УДК 681.3.016(075)

Швачич Г. Г., Рыжанкова Г. И., Кузьменко В. В., Барвинов В. П. Проектирование баз данных в СУБД Access. Часть 1. Учеб. пособие. – Днепропетровск: НМетАУ, 2006.-54 с.

Содержит краткие теоретические основы проектирования реляционных СУБД. На основе СУБД Access продемонстрированы этапы проектирования реляционной БД – последовательное построение информационно-логической и физической модели базы данных. Предложенные примеры разработок позволят усвоить технологии проектирования БД и применить их при создании самостоятельных приложений.

Может быть использовано на занятиях в системе элективной подготовки, а также при изучении темы СУБД студентами разных специальностей в спецразделах дисциплины "Информатика и вычислительная техника".

Библиогр.: 7 наим.

Ответственный за выпуск Г. Г. Швачич, канд. техн. наук, проф.

Рецензенты: Н.К. Васильева, канд. физ.-мат. наук, доц. (ДГАУ)
И.Н. Пистунов, канд. техн. наук, доц. (НГУ)

© Национальная металлургическая академия
Украины, 2006

ВВЕДЕНИЕ

Создание базы данных является одним из последних этапов проектирования информационной системы, позволяющей получить гибкий доступ к интересующей пользователя предметной области знаний.

СУБД Access является насыщенным и мощным приложением, позволяющим создавать таблицы, редактировать данные и использовать запросы для поиска данных с минимальной затратой усилий. Специализированные программы-Мастера способны вместо программиста выполнить разработку форм и отчетов; Построитель выражений позволяет упростить процесс обработки данных, макросы – автоматизировать многие процессы без программирования.

Access включает в себя язык программирования VBA с достаточно мощным интерфейсом, что позволяет разработчикам создавать свои приложения без создания программ по работе с данными. Запросы, формы, отчеты и макросы могут выполнить большую часть работы, которая раньше требовала программирования.

Предлагаемая разработка предназначена для начинающих пользователей – материал изложен в доступной форме и подкреплен большим количеством примеров. Тем не менее, в ней представлены основные технологии баз данных, усвоение которых позволит проектировать базу данных любой сложности.

1. МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Этапы моделирования информационной системы (ИС) включают:

- создание концептуальной модели конкретной предметной области;
- создание логической модели данных;
- создание физической модели данных, представляющей собой базу данных.

Проектирование базы данных начинается с очерчивания интересующей исследователя предметной области и построения концептуальной модели, которая отражает взаимосвязи между реальными объектами и процессами в этой предметной области.

Информация о предметной области может включать сведения о персонале, заработной плате, изучаемых курсах, результатах сессии,

финансовых сделках, историях болезней и т.д.

Концептуальная модель затем преобразуется в *модель данных*. Версия концептуальной модели, которая может быть обеспечена конкретной СУБД, называется *логической* моделью. Логическая (внешняя) модель данных может быть реляционной, иерархической или сетевой.

Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания в среде хранения. Логическая модель отображается в физическую с учетом типов данных поддерживаемых конкретной СУБД (FoxPro, Access, Oracle и т.д.).

Физическая модель определяет способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне. В реализации для выбранной СУБД эта модель называется внутренней моделью системы или *базой данных*.

Анализ предметной области и создание концептуальной модели предметной области

Рассмотрим предметную область, связанную с процессом контроля обучения. Сформулируем постановку задачи, приведем упрощенную схему взаимосвязей компонентов и процессов этой области.

В качестве примера предметной области рассмотрим часть работы деканата, который формирует процесс организации занятий по ряду дисциплин, и затем организует контроль успеваемости по выставленным студентам оценкам.

При анализе процесса обучения и контроля необходимо:

1. Выявить наиболее важные для данной предметной области компоненты. В нашем случае это будут:

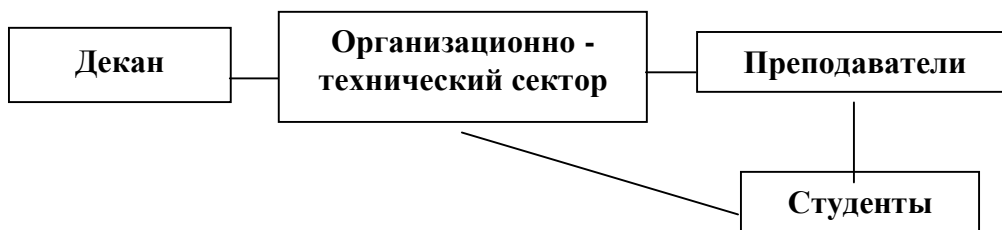
- преподаватели;
- студенты;
- дисциплины;
- оценочные ведомости.

2. Получить список основных процессов обучения и контроля. Для нашего примера в такой список можно включить следующие пункты:

- преподавание;
- изучение (студентами преподаваемых дисциплин);
- контроль обучения.

3. Представить проблемы телекоммуникаций и организации совместной работы персонала. В рассматриваемом примере допустим, что все операции выполняются в пределах одного здания, а организация совместного использования данных основана на возможностях локальной сети и сервера БД.

4. Определить организационную структуру подразделения. Представим схематически организационную структуру в виде :



Организационная структура подразделения

5. Определить периодичность осуществляемых процессов контроля, что позволит правильно расставить акценты в будущей прикладной программе. В нашем случае примем такую временную последовательность выполняемых процессов: итоговый контроль – один раз в конце семестра.

6. Определить мотивацию деканата:

- обеспечение условий для успешной деятельности персонала;
- получение студентами необходимого уровня знаний;
- повышение качества обучения.

Ответы на шесть пунктов позволяют подойти к построению концептуальной модели в виде взаимосвязей между компонентами и процессами. В практике проектирования информационных систем такие схемы получили название ER-диаграмм (Entity-relationship diagram (ERD) – диаграмма "Сущность-связь").



Диаграмма взаимосвязей между компонентами и процессами
Перечень вводимой информации:

- фамилия, имя, отчество студента;
- номер группы;
- дата рождения;
- оценки;
- название дисциплины;
- фамилия, имя, отчество преподавателя;
- дата рождения;
- должность;
- телефон;
- заработная плата.

2. СОЗДАНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ

Определение объектов (сущностей)

Для создания логической модели проанализируем перечень вводимой информации и концептуальную модель, с целью выяснить:

- какие нам понадобятся объекты;
- какими будут атрибуты этих объектов.

Объектом называется элемент информационной системы, информацию о котором мы сохраняем в базе данных. В реляционной теории баз данных объект называется сущностью.

Объект может быть реальным (например, человек, какой-либо предмет или населенный пункт) и абстрактным (например, событие или изучаемый студентами курс). Каждый объект обладает определенным набором свойств, но в базе данных информационной системы запоминаются только необходимые свойства, которые используются в ИС и необходимы для выдачи отчетов.

Исходя из поставленной задачи, выделим следующие сущности:

- сотрудники;
- студенты;
- дисциплины;
- оценки.

Объекты в реляционной модели данных представляются *отношениями*, отображенными в виде *таблиц*, в столбцах которых размещаются свойства объектов, называемые атрибутами.

Атрибут – это информационное отображение свойств объекта. Каждый

объект характеризуется рядом основных атрибутов. Каждый атрибут в модели должен иметь уникальное имя – идентификатор. Атрибут часто называют элементом данных, полем данных или просто *полем*.

В строках таблиц расположены записи.

Запись данных – это совокупность значений связанных элементов данных. Каждая запись одной таблицы состоит из конечного и одинакового числа полей, причем конкретное поле каждой записи одной таблицы может содержать данные только одного типа.

Информацию о некоторой предметной области можно представить с помощью нескольких объектов, каждый из которых описывается несколькими элементами данных.

Между атрибутами некоторых сущностей существует связь – своего рода функциональная зависимость между сущностями, позволяющая видеть многокомпонентную информационную систему как единое целое.

Для отражения и организации таких связей между таблицами используется понятие ключевого элемента.

Если известно значение, которое принимает ключевой элемент данных объекта, мы можем идентифицировать значения, которые принимают другие элементы данных этого же объекта.

Атрибут (или группа атрибутов), которые единственным образом идентифицируют каждую строку в таблице, называется первичным ключом.

Альтернативный (вторичный) ключ – это атрибут (или группа атрибутов), не совпадающий с первичным ключом и уникально идентифицирующий экземпляр объекта.

Например, для объекта *Студент*, который имеет атрибуты "Код студента", "Фамилия", "Имя" и "Отчество", группа атрибутов "Фамилия", "Имя", "Отчество" может являться альтернативным ключом по отношению к атрибуту "Код студента" (в предположении, что в группе не учатся полные тезки).

Для поддержания ссылочной целостности данных используется механизм внешних ключей. Смысл этого механизма состоит в том, что некоторому атрибуту (или группе атрибутов) одного отношения назначается ссылка на первичный ключ другого отношения и тем самым закрепляются связи подчиненности между этими отношениями. При этом *отношение, на первичный*

ключ которого ссылается внешний ключ другого отношения, называется главным отношением; отношение, от которого исходит ссылка, называется подчиненным отношением. После назначения такой ссылки СУБД имеет возможность автоматически отслеживать факты "нарушения" связей между отношениями. СУБД сгенерирует ошибку:

- при попытке вставить в подчиненную таблицу запись, для внешнего ключа которой не существует соответствия в главной таблице (например, там нет еще записи с таким первичным ключом);
- при попытке удалить из главной таблицы запись, на первичный ключ которой имеется хотя бы одна ссылка из подчиненной таблицы;
- при попытке изменить первичный ключ записи главной таблицы, на которую имеется хотя бы одна ссылка из подчиненной таблицы.

Поддержание непротиворечивости функциональных зависимостей между сущностями называется ссылочной целостностью.

Ссылочная целостность – это обеспечение соответствия значения внешнего ключа экземпляра дочерней сущности значениям первичного ключа родительской сущности.

Ссылочная целостность может контролироваться при всех операциях, изменяющих данные.

Определение взаимосвязей между сущностями

СУБД различает взаимосвязи таких типов: "один-к-одному", "один-ко-многим" и "многие-ко-многим".

Взаимосвязь "один-к-одному" (между двумя типами объектов)

Связь "один к одному" представляет собой простейший вид связи данных, когда первичный ключ таблицы является в то же время внешним ключом, ссылающимся на первичный ключ другой таблицы. Такую связь бывает удобно устанавливать тогда, когда невыгодно держать разные по размеру (или по другим критериям) данные в одной таблице. Например, можно выделить данные с подробным описанием изделия в отдельную таблицу с установлением связи "один к одному" для того чтобы не занимать оперативную память, если эти данные используются сравнительно редко.

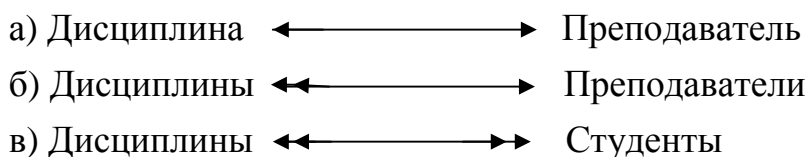
Допустим, что один предмет читает только один преподаватель. В этом случае между объектами *Преподаватель* и *Дисциплина* устанавливается

взаимосвязь "один-к-одному", обозначаемая одинарными стрелками, а).

Между данными, хранящимися в этих объектах, будет существовать взаимосвязь, в которой каждая запись в одном объекте будет однозначно указывать на запись в другом объекте. Ни в одном, ни в другом объекте не может существовать записи, не связанной с какой-либо записью в другом объекте.

Если один предмет читает несколько преподавателей, то между объектами *Дисциплина* и *Преподаватель* устанавливается связь один-ко-многим, б).

Студент изучает несколько дисциплин, и каждая дисциплина изучается множеством студентов; это порождает связь многие-ко-многим, в).



Взаимосвязи между двумя объектами: а) "один-к-одному";

б) "один-ко-многим"; в) "многие-ко-многим"

Взаимосвязь "один-ко-многим" (между двумя типами объектов)

Связь "один-ко-многим" в большинстве случаев отражает реальную взаимосвязь сущностей в предметной области. Она реализуется парой "внешний ключ – первичный ключ", т.е. когда определен внешний ключ, ссылающийся на первичный ключ другой таблицы.

Именно эта связь описывает широко распространенный механизм классификаторов. Пусть имеется справочная таблица, содержащая названия, имена и некие коды, причем, первичным ключом является код. В таблице, собирающей информацию – назовем ее информационной таблицей – определяется внешний ключ, ссылающийся на первичный ключ классификатора. После этого в нее заносится не название из классификатора, а код. Такая система становится устойчивой от изменения названий в классификаторах. Имеются способы быстрой "подмены" в отображаемой таблице кодов на их названия.

В этом случае одной записи данных первого объекта (его называют родительским или основным) будет соответствовать несколько записей второго

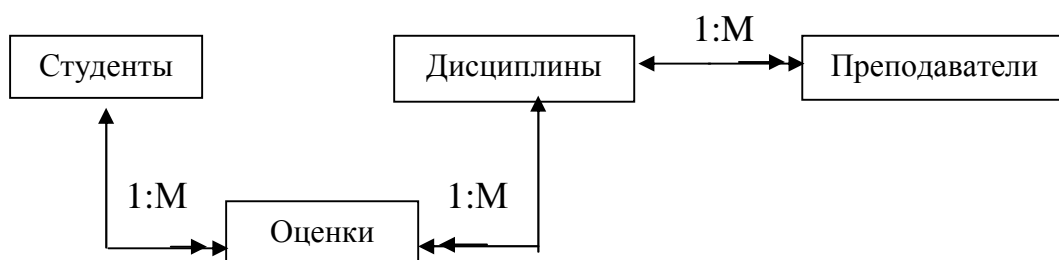
объекта (дочернего или подчиненного). Взаимосвязь "один-ко-многим" очень распространена при разработке реляционных баз данных. В качестве родительского объекта часто выступает справочник, а в дочернем хранятся уникальные ключи для доступа к записям справочника.

В нашем примере в качестве такого справочника можно представить объект *Дисциплины*, в котором хранятся сведения обо всех дисциплинах. При обращении к записи для определенной дисциплины, нам доступен список всех преподавателей, которые читают этот предмет и сведения о которых хранятся в объекте *Преподаватели*.

Взаимосвязь "многие-ко-многим" (между двумя типами объектов)

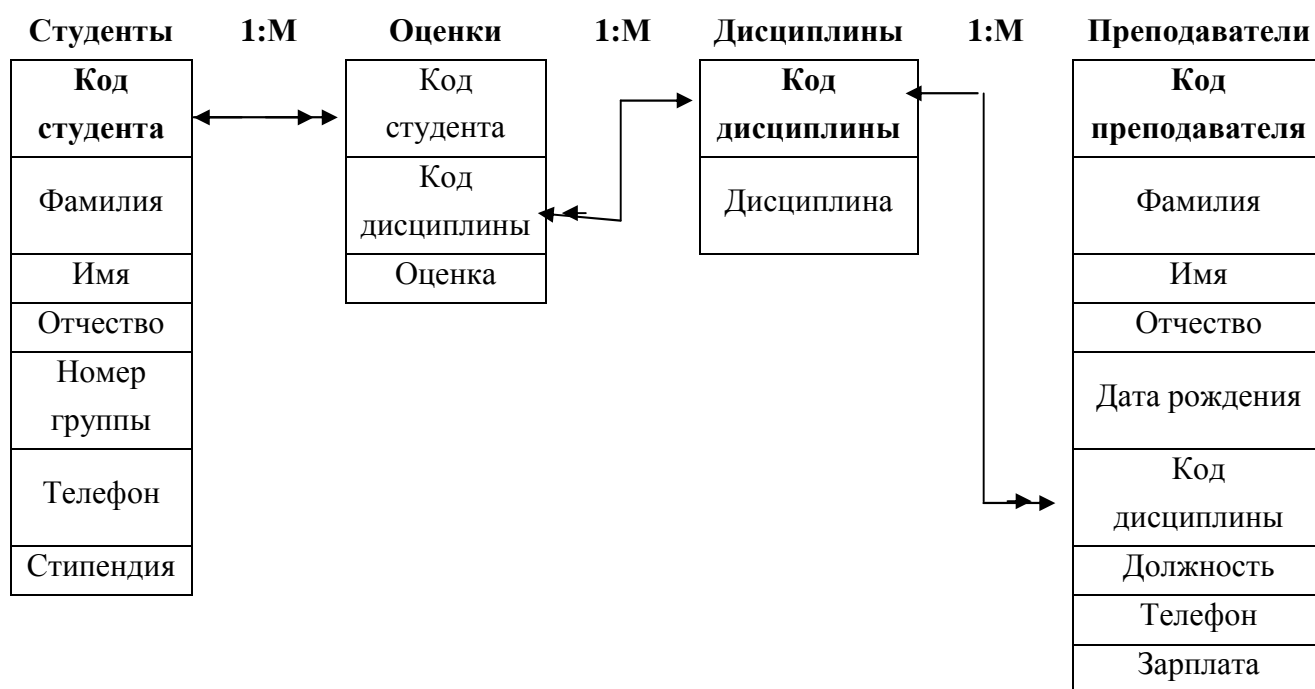
Связь "многие-ко-многим" в явном виде в реляционных базах данных не поддерживается, но имеются приемы косвенной реализации такой связи. Один из наиболее распространенных способов заключается во введении дополнительной таблицы, строки которой состоят из внешних ключей, ссылающихся на первичные ключи двух таблиц.

В реляционных СУБД для контроля целостности данных с возможностью каскадного обновления и удаления данных необходимо создать вспомогательный объект связи, который состоит из ключевых реквизитов связываемых объектов и который может быть дополнен описательными реквизитами. В нашем случае таким новым объектом для связи служит объект *Оценки*, реквизитами которого являются код студента, код дисциплины и оценки. Каждый студент имеет оценки по нескольким дисциплинам, поэтому связь между объектами *Студенты* и *Оценки* будет "один-ко-многим"(1:M). Каждую дисциплину сдает множество студентов, поэтому связь между объектами *Дисциплины* и *Оценки* также будет "один-ко-многим"(1:M). В результате получаем следующую информационно-логическую модель базы данных



Информационно-логическая модель реляционной базы данных

В реляционной базе данных в качестве объектов рассматриваются отношения, которые можно представить в виде таблиц. Таблицы между собой связываются посредством общих полей, т.е. одинаковых по форматам и, как правило, по названию, имеющихя в обеих таблицах. В соответствии с этим логическая модель базы данных **Деканат** может иметь вид (жирным начертанием выделены ключевые поля):



Логическая модель базы данных **Деканат**.

Приведение модели к требуемому уровню нормальной формы

Приведение модели к требуемому уровню нормальной формы является основой построения оптимальной структуры таблиц и связей в реляционной БД.

Введение нормализации отношений при разработке информационной модели обеспечивает её минимальный объем и максимальное быстродействие.

Нормализация позволяет разбить большие отношения, содержащие, как правило, большую избыточность информации, на более мелкие логические единицы, группирующие только данные, объединенные "по природе".

Исходные данные, представленные в виде двумерной таблицы, являются первой нормальной формой реляционной модели данных.

Первый этап нормализации заключается в образовании двумерной

таблицы, содержащей все необходимые атрибуты информационной модели, и в выделении ключевых атрибутов. Очевидно, что полученная весьма внушительная таблица будет содержать очень разнородную информацию. В этом случае будут наблюдаться аномалии включения, обновления и удаления данных, так как при выполнении этих действий нам придется уделять внимание данным (вводить или заботиться о том, чтобы они не были стерты), которые не имеют к текущим действиям никакого отношения.

Отношение задано во второй нормальной форме, если оно является отношением в первой нормальной форме и каждый атрибут, не являющийся первичным атрибутом в этом отношении, полностью зависит от любого возможного ключа этого отношения

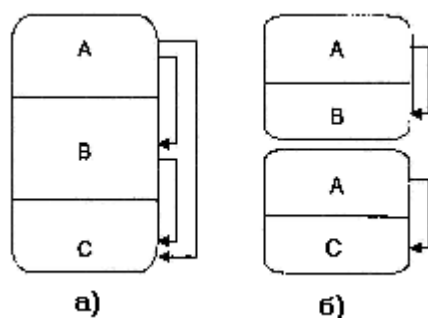
Приведение отношений ко второй нормальной форме заключается в обеспечении полной функциональной зависимости всех атрибутов от ключа за счет разбиения одной таблицы на ряд таблиц, в которых все имеющиеся атрибуты будут иметь полную функциональную зависимость от ключа этой таблицы. В процессе приведения модели ко второй нормальной форме в основном исключаются аномалии дублирования данных.

Отношение задано в третьей нормальной форме, если оно задано во второй нормальной форме и каждый атрибут этого отношения, не являющийся первичным, не транзитивно зависит от каждого возможного ключа этого отношения.

Транзитивная зависимость выявляет дублирование данных в одном отношении. Если А, В и С – три атрибута одного отношения и А зависит от В, а В от С. В этом случае зависимость $A \rightarrow C$ называется транзитивной зависимостью. Если отношение является содержит транзитивную зависимость сущностей, то преобразование его в третью нормальную форму происходит за счет разделения одного отношения на два.

Сформулируем основные правила, которым нужно следовать при проектировании базы данных:

- Исключайте повторяющиеся группы – для каждого набора связанных атрибутов создайте отдельную таблицу и снабдите ее первичным



Приведение транзитивного отношения к третьей нормальной форме: а) отношение с транзитивной зависимостью;

б) отношения без транзитивной зависимости

ключом. Выполнение этого правила автоматически приведет ко второй нормальной форме.

- Представьте, что в вашем списке дисциплин вы указываете фамилии преподавателей. Если в написании фамилии возникнет ошибка – то для СУБД это уже другой преподаватель. Поэтому лучше вести список преподавателей в отдельной таблице, а в списке дисциплин использовать только присвоенные им уникальные идентификаторы.
- Исключайте избыточные данные: если атрибут зависит только от части составного ключа, переместите атрибут в отдельную таблицу. Это правило помогает избежать потери одних данных при удалении каких-то других. Везде, где возможно использование идентификаторов вместо описания, выносите в отдельную таблицу список идентификаторов с пояснениями к ним.
- Исключайте столбцы, которые не зависят от ключа: если атрибуты не вносят свою лепту в описание ключа, переместите их в отдельную таблицу.

В основном изменения в модели связаны с введением искусственных атрибутов, которые в виде кодов участвуют в отношениях вместо естественных атрибутов. К необходимости введения в модель искусственных атрибутов мы приходим в процессе нормализации.

Анализ разработанной логической модели базы данных **Деканат** показывает, что она соответствует требуемым уровням нормальных форм.

3. ИНСТРУМЕНТАЛЬНЫЕ КОМПОНЕНТЫ СУБД ACCESS

Окно базы данных Access - контейнер, содержащий все объекты, которые формирует база данных. Окно имеет закладки с названиями различных типов объектов: *Таблицы, Запросы, Формы, Отчеты, Макросы, Модули*.

Сведения о каждом источнике сохраняются в отдельной таблице; при работе с несколькими таблицами между ними устанавливаются связи.

Для просмотра, ввода или изменения данных прямо в таблице применяются формы. Форма позволяет отображать данные, используя стандартный или пользовательский макет.

Для отбора и поиска данных, удовлетворяющих определенным условиям, создается запрос. Запрос позволяет также обновить или удалить несколько записей, выполнить встроенные или специальные вычисления.

Для анализа и распечатки данных используется отчет, позволяющий группировать данные и вычислять итоги.

Для выполнения различных действий с этими объектами можно использовать кнопки <Создать>, <Открыть>, <Конструктор>, находящиеся с правой стороны окна базы данных:

- Создать – создается новый объект отображаемого типа;
- Открыть – открывается выделенный объект для просмотра и ввода данных;
- Конструктор – позволяет изменить структуру выделенного объекта.

Особенности окна Access:

- большая часть операций выполняется из окна базы данных; система использует меню гораздо реже, чем другие приложения Microsoft Office;
- закладки: *Отчеты, Макросы, Модули* имеют вместо кнопки <Открыть> другие кнопки, но кнопки <Создать> и <Конструктор> у них те же;
- после открытия окна для отображения объекта можно переключиться между режимами отображения или изменения структуры данных (используя инструмент или пункт меню **Вид**).

Общие сведения о таблицах

Таблица – главный объект Access. В режиме таблицы добавляются, просматриваются и редактируются табличные данные. Можно также проверить орфографию, распечатать табличные данные, отфильтровать и отсортировать

записи, изменить внешний вид или структуру таблицы.

Каждый объект БД должен быть сначала сконструирован, а потом заполнен. До использования таблицы необходимо создать ее структуру, указав список полей и их тип. Это можно сделать в режиме конструктора, который можно включить, используя инструмент **Вид** в левой части панели инструментов.

Настройка таблицы в режиме конструктора включает:

- добавление, перемещение и переименование поля;
- определение типа данных;
- установку свойств полей;
- определение ключевого поля.

Работа с таблицей в режиме таблицы включает:

- редактирование, проверку орфографии, печать;
- добавление и удаление записей;
- сортировку, отбор и поиск записей;
- изменение макета таблицы (размеров строк и полей, цвета фона, сетки, символов и т.д.);
- вставку, удаление и переименование столбца.

Использование запросов

Запросы позволяют формировать выборки из таблиц, определяя поля и записи, выводимые на экран, объединять несколько таблиц в реляционных базах данных, модифицировать данные в таблицах. Самым простым и важным типом запроса является запрос на выборку, который позволяет указать:

- какие поля, из каких таблиц и в каком порядке должны отображаться в запросе, и тем самым определить структуру будущих записей;
- ввести критерий для отбора отображаемых записей;
- определить порядок сортировки этих записей.

Как и окно "Таблица", окно "Запрос" может отображаться как в режиме таблицы, так и в режиме конструктора. Панель инструментов в окне "Запрос" включает также кнопку <Режим SQL>, позволяющий отобразить лежащий в основе запроса код программы.

Для каждого поля могут быть заданы условия отбора, позволяющие определить, какие записи из общего набора запроса будут отображаться. Можно задать опцию выбора для отображаемых в запросе полей, которые

будут выведены на экран при просмотре запроса в виде таблицы. Запросы выборки можно использовать вместо таблиц при создании форм и отчетов.

Назначение форм

Формы позволяют управлять процессом отображения данных на экране. Формат таблиц позволяет видеть множество записей одновременно, но для длинных записей отображение всей информации обычно невозможно.

Форма, как правило, отображает одну запись одновременно. В форму можно включить не все поля таблицы, есть возможность расположить поля на экране визуальнее более информативно. В форму можно ввести дополнительные информационные элементы, делающие ее привлекательной и функциональной - например, фотографии, картинки, элементы управления в виде кнопок и т.д.

Формы используются для различных целей:

- для ввода данных в таблицу;
- для открытия других форм и отчетов (кнопочная форма);
- для выбора, предварительного просмотра и печати отчета (специальное окно диалога).

Инструмент **Вид**, расположенный в левой части панели инструментов, позволяет задать один из трех режимов функционирования формы: режим формы, режим таблицы или конструктора.

Режим таблицы позволяет просматривать поля, включенные в форму.

Окно базы данных в режиме формы представляется макетом, содержащим:

- поясняющий текст для полей;
- графические элементы (линии, кнопки, и др.);
- вычисленные значения;
- поля, данные для которых берутся из полей базовой таблицы.

Для создания связи между формой и записями таблицы, являющейся источником данных для формы, используются графические элементы, называемые элементами управления. Чаще всего для отображения и ввода данных используются поля.

Отчеты

Отчеты предназначены специально для печати данных. Поэтому при отображении отчета вместо кнопки окна базы данных <Открыть> используется

кнопка <Просмотр>. Источником данных для отчета может служить базовая таблица, запрос или инструкции на SQL. Для создания связи между отчетом и его исходными данными применяются элементы управления: поля (содержащие имена или числа), надписи для заголовков, декоративные линии для графического оформления отчета. Окно отчета в режиме конструктора очень похоже на окно формы в этом же режиме.

Данные отчета не редактируются, как это делается в таблицах, запросах или формах. Окно предварительного просмотра отображает данные такими, какими они будут выглядеть при печати.

Отчеты обычно конструируются с использованием Мастера отчетов. Работа типового мастера представляет собой серию шагов, снабженных инструкциями и элементами выбора, позволяющими задать объектам определенные характеристики и оценить визуально в малых окнах сделанный выбор. Каждый шаг представляется очередным окном диалога, содержащим навигационные кнопки <Назад>, <Далее (Готово) >, <Отмена>.

Автоформы. Автоотчеты

Создавать новые объекты можно также с использованием инструмента **Новый объект**. Инструмент содержит список, элементы которого доступны в режиме *Создать* из окна базы данных. Но есть две опции, позволяющие автоматически создать форму или отчет, основанные на выделенном запросе или таблице – это Автоформа и Автоотчет.

4. СОЗДАНИЕ БАЗЫ ДАННЫХ

Создать базу данных можно двумя различными способами:

- создать пустую базу данных, а затем таблицы и другие необходимые объекты;

- использовать один из шаблонов базы данных, и, воспользовавшись мастером баз данных, сделать его отвечающим предъявляемым требованиям.

Если выбрать **Создать базу данных** из меню **Файл**, возникнет окно диалога "Создание" с двумя закладками: *Общие* и *Базы данных*. Закладка *Общие* имеет значок "Новая база данных"; закладка *Базы данных* позволяет выбрать один из шаблонов, разработанных для специальных целей и содержащих особого вида таблицы, формы и отчеты (Основные фонды, Заказы,

Личное имущество и т.д.)

При создании пустой базы данных Access выводит окно "Файл новой базы данных" для определения папки и имени новой базы данных. Задав папку и имя файла, щелкнем на кнопке <Создать>. Access автоматически добавит к имени файла расширение .mdb. Появится окно с новой, незаполненной базой данных. Если выбрать шаблон, Access отобразит окно Мастера баз данных, который поможет приспособить этот шаблон для целей пользователя.

Создание таблицы

Созданное окно новой пустой базы данных имеет по умолчанию в качестве выбранной закладки *Таблицы*. Если щелкнуть на кнопке <Создать>, Access отобразит окно диалога "Новая таблица" с выбором ряда режимов создания новой таблицы: Режим таблицы, Конструктор, Мастер таблиц, Импорт таблиц, Связь с таблицами.

Режим таблицы создает новую таблицу и отображает ее в режиме таблицы, так что в нее можно сразу же вводить данные. Вновь созданная таблица имеет 20 полей с именами "Поле 1", "Поле 2",..., "Поле 20". Можно переименовать поля, дважды щелкнув по заголовкам столбцов и отредактировав их имена. При первом сохранении таблицы Access определит тип данных в каждом поле, базируясь на введенной информации. Можно потом изменить описание этих полей.

Импорт таблиц и связь с таблицами позволяют создать специальные типы таблиц для работы с данными из других приложений.

Если надо быстро ввести данные, удобно создавать таблицу в режиме конструктора. При этом целесообразно до ввода данных разработать структуру таблицы – состав полей и тип данных каждого поля.

Мастер таблицы позволяет осуществить выбор таблицы из предлагаемого множества готовых таблиц, после чего в режиме конструктора произвести необходимые настройки.

Первый шаг мастера позволяет сделать выбор образца таблицы и определить, какие поля будут входить в создаваемую таблицу. Окно "Создание таблиц" содержит три поля списка: "Образцы Таблиц", "Образцы полей", "Поля новой таблицы". Нужные поля выбранной в качестве прототипа таблицы можно выделить и перебросить кнопкой ">" в поле списка "Поля новой таблицы". Ошибочно введенное поле можно выделить в списке "Поля новой таблицы" и

вернуть в список "Образцы полей" кнопкой "<". Для одновременного перемещения всех полей используются кнопки ">" и "<<".

Второй шаг мастера позволяет ввести имя таблицы и выбрать ключевое поле.

Третий шаг – создание самой таблицы. Можно модифицировать ее, отобразив в режиме конструктора, или сразу вводить в нее данные.

Окно таблицы в режиме конструктора

При конструировании таблицы задаются имена полей, а также тип данных для каждого из этих полей; для некоторых типов полей необходимо задать размер. Такой процесс конструирования таблицы называют определением структуры или разработкой таблицы.

Окно таблицы в режиме конструктора содержит три колонки: "Имя поля", "Тип данных", "Описание". В режиме конструктора каждому полю таблицы отводится строка. Перемещаться между столбцами "Имя поля", "Тип данных", "Описание" можно с помощью мыши, клавишей горизонтального перемещения курсора, клавишей Tab.

Имя поля вводится в столбце "Имя поля". Если переместить курсор в столбец "Тип данных", Access развернет список всех возможных типов данных (необходимо выбрать тип данных для каждого поля таблицы). Использование столбца "Описание" не обязательно, в него можно вводить справочную информацию, полезную во время ввода данных.

Нижняя секция окна используется для задания свойств указанного поля.

Панель в левой нижней части окна содержит две вкладки: *Общие* (Размер поля, Формат поля, Маска ввода, Начальное значение и т.д.) и *Подстановка* (Тип элемента управления, Тип источника строк, Источник строк и т.д.).

Панель в правой нижней части окна используется для вывода короткой контекстно-зависимой справки по выбранной характеристике свойства поля.

Сохранение таблицы

При первом сохранении Access выводит на экран окно "Сохранение", позволяющее задать имя таблицы, но не позволяющее задать диск и папку, т.к. они определены для файла базы данных, а таблица – только часть этого файла.

Типы данных в структуре таблицы

Для определения структуры таблицы необходимо ввести имя и тип данных для всех полей.

Типы данных:

- текстовый (до 255 символов);
- Мемо (до 65000 символов). Поле с данными типа Мемо имеет переменную длину, и размер его не задается;
- числовой (тип чисел и точность вычислений зависят от заданного размера числового поля - длинное целое, целое, с плавающей точкой, байт и др.);
- дата/время (в зависимости от формата, присвоенного полю, в поле вводятся календарные данные или значения времени);
- денежный (используется в денежных расчетах; вычислительная точность - 4 знака после запятой; могут использоваться с обозначением денежной единицы или без);
- счетчик (последовательный набор натуральных чисел; введенные счетчиком числа нельзя изменить);
- логический (Yes/No). Определяется одним из двух значений в зависимости от формата, заданного полю в панели "Свойства поля";
- поле объекта OLE. Содержит данные из других приложений;
- гиперссылка (содержит адрес объекта, документа или Web-страницы);
- мастер подстановок (создает поле, при помощи которого могут быть выбраны значения из списка).

По умолчанию длина данных текстового поля составляет 50 символов.

Для целочисленных полей чаще всего выбирают в качестве размера *Длинное целое*; тип *С плавающей точкой* – для вещественных переменных или для случаев, когда вычисления должны быть точными.

Если достаточная точность 4 знака, то можно использовать тип *Денежный*, даже если значения не являются денежными суммами (расчет для этого типа идет с фиксированной запятой, а не с плавающей, и поэтому производится быстрее). Можно использовать раскрывающийся список свойства *Формат поля* для определения, каким образом на экране будут представлены эти поля – со знаком денежной единицы или без него.

Поля первичного ключа

Ключевое поле однозначно идентифицирует каждую запись в таблице. Самым распространенным первичным ключом является последовательно пронумерованное поле. Хотя это и возможно, не стоит использовать поля с конкретной информацией в качестве первичных ключей: если данные по такому полю изменяются, то создаются проблемы с выводом данных и можно потерять часть информации. Но в случае, когда данные представляют собой уникальный код (например, *Номер страхования*), то в отдельных случаях это поле может быть использовано в качестве первичного ключа.

Чтобы создать ключевое поле, необходимо:

- установить в поле курсор;
- выбрать опцию **Ключевое поле** в меню **Правка** или щелкнуть на кнопке <Ключевое поле> панели инструментов.

Access может сама создать поле первичного ключа, если пользователь не создал его самостоятельно. При закрытии окна "Таблица" Access запрашивает подтверждение на создание поля первичного ключа. Если в таблице есть поле с типом данных *Счетчик*, то Access назначит его в качестве первичного ключа; если нет – Access добавляет в таблицу новое поле с типом данных *Счетчик* и присваивает ему имя "Код".

Установка режимов отображения таблицы. Способы переключения между ними

После создания таблицы ее имя появляется в окне базы данных и можно теперь ее открыть в одном из режимов – таблицы или конструктора, если:

- выделить имя таблицы и щелкнуть на кнопке <Открыть> или дважды щелкнуть на имени таблицы, то она откроется в режиме таблицы;
- выделить имя таблицы и щелкнуть по кнопке <Конструктор> – таблица откроется в режиме конструктора.

После открытия таблицы в левой части панели инструментов появится кнопка <Вид>, позволяющая переключать режимы отображения и функционирования (Режим таблицы – Конструктор).

Модификация структуры таблицы

В режиме таблицы возможны некоторые простые изменения в конструкции таблицы: добавление и удаление полей, переименование и перемещение полей, изменение размеров (высоты строки, ширины столбца).

Для выполнения большинства из этих действий удобно использовать меню быстрого доступа.

Если щелкнуть на заголовке столбца правой кнопкой мыши, станут доступными команды: **Ширина столбца, Добавить столбец, Удалить столбец, Переименовать столбец, Столбец подстановок.**

Команда **Добавить столбец** вставляет новый столбец слева от того, меню которого используется; по умолчанию ему будет присвоено имя "Поле 1" текстового типа (поле можно переименовать, а свойство поля изменить в режиме конструктора).

Ширину столбца можно изменить, перетаскивая правую границу заголовка или используя команду **Ширина столбца.**

Для перемещения столбца необходимо его выделить (щелчком левой кнопки мыши по заголовку), захватить курсором заголовок и переместить столбец на нужное место (индикатором нового положения служит черная вертикальная черта за заменяемым столбцом в направлении перемещения).

Если щелкнуть правой кнопкой мыши на поле маркировки записей, то строка будет выделена, и станут доступны команды: **Вырезать, Вставить (копию), Высота строки.**

В режиме конструктора:

если щелкнуть правой кнопкой мыши на поле маркировки строки, то она будет выделена и станут доступны команды: **Добавить строки, Удалить строки.** Переместить строку можно, захватив ее в поле маркировки левой кнопкой мыши и перетаскивая в нужном направлении; индикатором места вставки для перемещаемой строки служит черная горизонтальная полоса.

Высоту строк можно изменить, перемещая границу разделения маркированных строк.

О потере данных.

Если изменена структура таблицы так, что это может привести к потере данных, Access предупредит об этом при попытке сохранить изменения. Например:

- удаление поля, содержащего данные;

- уменьшение размера поля, содержащего данные (текст может сократиться и потерять урезанную часть, числовые данные округлятся с потерей точности);

- изменение типа данных (данные, которые не могут быть сохранены в новом типе данных, теряются).

Перед модификацией структуры желательно создать резервную копию таблицы. Для этого можно использовать следующую последовательность действий:

- выделить таблицу в окне базы данных;

- выполнить команды: **Правка, Копировать; Правка, Вставить**).

Появится окно "Вставка таблицы", запрашивающее новое имя и параметры вставки (Только структуру; Структуру и данные).

5. УПРАВЛЕНИЕ ОТОБРАЖЕНИЕМ ДАННЫХ

Секция *Свойства поля* (режим конструктора) имеет окно диалога с двумя вкладками *Общие* и *Подстановка*, позволяющими задать некоторые свойства поля, и окно комментариев к задаваемым свойствам, например:

- *Подпись* (Caption); свойство позволяет указать надпись, которая будет отображаться в качестве наименования поля в таблице и автоформе;

- *Условие на значение* (так, если для поля "Количество" необходимо установить предел нижней величины, можно ввести, например, условие ">5"). Ввод данных, не соответствующих условию, вызовет возражение на этапе сохранения данных. При вводе условия можно использовать Построитель выражений.

Учесть:

- характеристика *Размер поля* применяется к полям с текстовым типом данных, числовым и типом Счетчик, но в каждом случае работает по-разному;

- характеристика *Число десятичных знаков* относится только к полям с численным и денежным типом данных;

- характеристика *Формат поля* работает по-разному для полей с различным типом данных (форматы текста, чисел и дат, безусловно, различны).

Вкладка *Подстановка* панели *Свойства поля* позволяет соответствующей настройкой создать раскрывающиеся списки для выбора из них заменяющих код значений при вводе данных в режиме таблицы.

Характеристики текстовых полей.

Размер поля: определяет максимальное число символов, которые могут быть введены в это поле (1 - 255), 50 по умолчанию. Если необходимо ввести более 255 символов, используется поле Мемо.

Access не добавляет пробелы в конец полей с текстом. Поэтому не следует уменьшать их размеры для экономии дискового пространства.

Формат текстового поля.

Для форматирования текстовых полей используются символы:

- @ – требуется символ;
- & – не требуется символ;
- > – преобразование в заглавные символы;
- < – преобразование в прописные символы.

Остальные символы используются буквально и вставляются в данные. Например, для ввода текста с разделяющими дефисами можно использовать, например, шаблон @@-@@-@@@ и вводить данные подряд, опираясь на шаблон.

Учесь: формат поля определяет только способ представления данных и не является контролем ввода. Для контроля необходимо использовать маску ввода.

Характеристики числовых полей.

Так как размер числового поля определяет, какой вид чисел может содержаться в поле, а также диапазон значений этих чисел, то всегда следует задавать размер поля для данных этого типа. Как правило, для числовых полей устанавливают характеристику *Формат поля*, где указывают число десятичных знаков после запятой.

Размер поля: байт, целое, длинное целое, с плавающей точкой (4 байт), с плавающей точкой (8 байт), код репликации (создает всемирно принятое идентификационное число, используемое в качестве сложного типа данных).

По умолчанию данные числовых полей отображаются на экране так, как их ввел пользователь. Но можно присвоить полям один из стандартных форматов, выбираемых из списка *Формат поля*:

- основной (выбор по умолчанию);
- денежный (включает символ валюты после числа; отрицательные числа отображаются в круглых скобках; каждые три цифры целой части разделяет разделитель, определенный настройкой в Языках и Стандартах); дробная часть

содержит 4 знака;

- фиксированный (в дробной части - два знака после запятой, если характеристики числа десятичных знаков не изменены);
- процентный (отображает значения в виде процентов);
- экспоненциальный (научная система обозначений).

Число десятичных знаков: – список этой характеристики поля включает опцию Auto, позволяющую отобразить число знаков после запятой, указанное в характеристике *Формат*. Список также дает возможность выбрать число знаков после запятой от 1 до 15, переопределив установку характеристики поля *Формат*.

Тип данных Дата/Время:

возможен выбор стандартных форматов из окна списка характеристики поля *Формат*:

- для даты: Полный формат даты; Дата (краткий формат) – Время (длинный формат); Длинный формат (6 сентября 2004); средний (06-сент-2004); краткий (06.09.04);
- для времени: Длинный (2:45:26); Средний (2:45); Краткий (14:26).

Тип данных Денежный:

в этом поле хранятся данные с фиксированной десятичной точкой и форматируются так же, как и числовые. Размер поля с этим типом данных не задают.

Тип данных Счетчик:

по умолчанию характеристика *Размер поля* устанавливается как *Длинное целое*, а *Новые значения* - как *Последовательные* (можно выбрать для этой характеристики значения *Случайные*).

Логический тип данных:

может иметь три формата: Да/Нет, Истина/Ложь, Вкл/Выкл. По умолчанию это поле появляется в таблице в виде выключателя состояния (флажка).

Тип данных Гиперссылка.

В полях с этим типом данных Access хранит адреса Web-страниц и документов, находящихся в сети или на собственном компьютере, например: C:\Мои документы\Имя файла. Данные в этом поле отображаются голубым цветом и подчеркиваются.

Поля объектов OLE.

Как правило, в полях этого типа хранится графическая информация, например, фотографии сотрудников.

Мастер подстановок.

Поле подстановок удобно, если ввод данных в это поле осуществляется из некоторого фиксированного набора значений.

Если поле создано Мастером подстановок, то при заполнении таблицы значения поля можно выбирать из раскрывающегося списка всех значений поля(его называют столбцом подстановок).

Мастер подстановок можно вызвать, выделив его в качестве типа данных поля (это не изменит тип данных поля: он восстановится после конца работы мастера).

Первый шаг работы мастера требует указать способ создания списка:

- использовать ввод фиксированных значений;
- значения из таблицы (запроса).

Последующие шаги зависят от первого выбора. Если выбран режим ввода в список, то второй шаг требует указать, какое количество столбцов необходимо отобразить, и ввести значения в эти столбцы.

Следующий шаг – указать в секции выбора "Доступные поля", какой столбец содержит значения для ввода в таблицу (в приведенной таблице - столбец 1).

Последний шаг – задать подпись, используемую в описании раскрывающегося списка, сгенерированного мастером (лучшим вариантом обычно является имя поля, используемое по умолчанию).

После запуска Мастера подстановок поля закладки *Подстановка* заполняются автоматически. Характеристика *Ограничиться списком* дает возможность выбрать: довольствоваться только значениями из списка или можно вводить вместо них другие значения (Да/Нет).

Общие свойства различных типов данных

Маска ввода.

Маска ввода определяет формат, по которому должны вводиться данные. Другие символы могут быть использованы в маске ввода буквально и будут

вставлены в данные, которые вводятся. Например, маска 00-00-000 используется для того, чтобы вводимые цифры отображались с дефисами в

Символ	Значение
0	Требуется цифра
#	Можно вводить цифру, пробел, знаки +,-, но не обязательно;
L	Должна быть введена буква
A	Должна быть введена буква или цифра
C	Может быть введен любой символ или пробел, но не обязательно
< , >	Преобразование к регистрам

соответствующих местах (Access автоматически вставит "\"(слэш) перед каждым дефисом, чтобы показать, что последующий символ будет использован буквально).

Характеристика Подпись.

Все поля имеют характеристику *Подпись*, которая используется в качестве заголовка столбца поля в режиме таблицы. Если не указывается подпись, то вместо заголовка используется имя поля.

Значение по умолчанию.

Отображается как значение поля каждый раз, когда добавляется новая запись. Это поле можно редактировать.

Условие на значение.

Это выражение, которое оценивается, когда в поле заносятся или редактируются данные.

Обязательное поле.

Обязывает ввести данные в случае, если эта характеристика имеет значение Да.

Индексированное поле.

Позволяет создавать индекс, основанный на поле, с целью ускорения доступа к данным.

Работа с данными в режиме таблицы

Редактор Access.

Отличия от редактирования в других приложениях:

- клавиши перемещения курсора используются для быстрого перемещения между полями;

- по умолчанию данные в полях всегда выделяются при использовании клавиатуры для перемещения по таблице;

- для помещения текстового курсора в подсвеченный (при использовании клавиатуры) текст поля, необходимо нажать F2, а затем стрелками горизонтального перемещения перемещаться по тексту;

- поместить курсор в нужное поле можно, щелкнув левой кнопкой мыши в этом поле; двойной щелчок выделяет данные.

Добавление и редактирование данных.

В поле маркера записи (слева от полей таблицы) может находиться один из символов:

- * - пустая запись, добавляемая в конец таблицы и не сохраняющаяся при закрытии таблицы;

- Ø - стрелка, указывающая на текущую запись;

- ! - карандаш; при вводе или редактировании данных стрелка меняется на карандаш – признак несохраненных изменений; нажатие <Esc> отменяет изменения, сделанные в текущей записи. Изменения автоматически сохраняются при переводе подсветки на другую запись или нажатии <Shift>+<Enter>.

Можно выделить запись (сделать ее текущей), щелкнув на поле маркировки записей или использовать меню **Правка: Правка, Выделить запись**. При этом все поля записи будут подсвечены.

Добавление и модификация записей

Чтобы добавить в таблицу новую запись, необходимо переместить подсветку в пустую строку, которую Access автоматически добавляет в конец таблицы (строка помечена *). Для позиционирования курсора на новой записи необходимо:

- щелкнуть на кнопке <Новая запись> в правой части панели инструментов прикладной программы;

- щелкнуть на кнопке <Новая запись> в правой части панели инструментов, находящейся в нижней части панели "Таблица";

- нажать <Ctrl>+<+>.

Если желательно добавлять новые записи без отображения существующих, используют команду **Записи, Ввод данных**. Access выведет на

экран только строку для ввода новой записи. Чтобы вернуться к полному отображению таблицы, необходимо ввести команду **Записи, Удалить фильтр**.

Для удаления записи необходимо запись сначала выделить, а затем нажать клавишу или щелкнуть на инструменте **Удалить** (справа в инструментальной панели).

Для ускорения режима ввода может быть использована команда **Специальная вставка**. Команда создает новую запись, используя буфер обмена: выделяется запись-образец, копируется в буфер, копия вставляется в конец таблицы командой **Правка, Специальная вставка**; после чего запись может быть отредактирована.

Для быстрого перемещения между записями следует использовать инструменты **Запись**, расположенные в нижней части окна "Таблица".

Форматирование ячеек

Access имеет несколько команд установки внешнего вида ячеек таблицы. Команда **Формат ячейки** выводит на экран окно диалога "Вид сетки", опции которого позволяют:

- выбрать цвет линий сетки и фона ячеек (секция **Линии сеток**);
- придать ячейкам приподнятый, утопленный или обычный вид (секция **Оформление**).

Сохранение установок таблицы

Реализуется командой **Файл, Сохранить**. Если закрыть окно таблицы без сохранения изменений, то Access даст запрос для сохранения макета или структуры (данные сохраняются в момент ввода или модификации автоматически).

Сортировка данных

Для удобства просмотра можно сортировать записи в таблице в определенной последовательности, выбрав поля, используемые для сортировки. Например, в таблице *Преподаватели* записи можно отсортировать в порядке убывания стажа преподавателей, используя кнопки сортировки на панели инструментов (или команды меню **Записи, Сортировка, Сортировка по возрастианию (Сортировка по убыванию)**). Для выбора поля достаточно поместить курсор в любую его запись. В режиме таблицы можно выделить сразу два или несколько соседних

столбцов, а затем выполнить по ним сортировку. По умолчанию в Access сортировка записей начинается с крайнего левого выделенного столбца. При этом записи таблицы будут отсортированы сначала по крайнему левому выделенному столбцу, затем (для одинаковых значений в первом сортируемом столбце) – по второму и т.д. Если необходимо восстановить порядок отображения записей, используйте команду **Записи, Удалить фильтр**.

Примечание. Современные СУБД, такие, как Access, не сортируют таблицу физически. Средства сортировки данных (а также фильтрации, поиска и замены) реализованы в Access как автоматически создаваемые запросы. Записи таблицы всегда располагаются в файле базы данных в том порядке, в котором они были добавлены в таблицу.

Отбор данных с помощью фильтра

Фильтр – это набор условий, применяемых для отбора подмножества записей. В Access существуют фильтры четырех типов: фильтр по выделенному фрагменту, обычный фильтр, расширенный фильтр и фильтр по вводу.

Фильтр по выделенному фрагменту — это способ быстрого отбора записей по выделенному образцу. Например, вам нужно просмотреть в таблице записи только о доцентах. Выделите слово *Доцент* в любой из записей. Щелкните по кнопке <Фильтр по выделенному> (или выполните команду **Записи, Фильтр, Фильтр по выделенному**), и Access выберет только те записи, для которых значение в столбце "Должность" равно *Доцент*. Обратите внимание: в строке состояния окна таблицы присутствует слово *Фильтр*, а кнопка <Применить фильтр> (третья кнопка фильтрования, на которой изображена воронка) затенена; это означает, что используется фильтр. При отключении этой кнопки все фильтры будут сняты. Установки фильтра не пропадут: он просто будет отключен.

Фильтр по выделенному может собирать вместе критерии выбора при каждом использовании кнопки <Фильтр по выделенному>. Например, вы поместили курсор в столбец "Должность" в запись, в которой значение поля равно *Доцент*, и щелкнули по кнопке <Фильтр по выделенному>. Вы

увидите только записи о всех доцентах. Если затем вы поместите курсор в столбец "Дисциплина" и выделите слово *Информатика* (конечно, если такая дисциплина присутствует в вашей таблице), а затем щелкните по кнопке <Фильтр по выделенному> – появятся только записи о доцентах, которые преподают информатику. Если вы хотите просмотреть значения, которые не удовлетворяют этому критерию, например, просмотреть всех преподавателей, кроме доцентов, читающих информатику, щелкните правой кнопкой (курсor мыши должен находиться внутри таблицы) и выберите команду **Исключить выделенное**. Будут выбраны все записи, кроме недавно выбранных (инверсный выбор).

Фильтрация данных в Access производится также с помощью кнопки <Изменить фильтр> (команда меню **Записи, Фильтр, Изменить фильтр**). После нажатия второй кнопки от таблицы остается одна запись. Каждое поле становится полем со списком (когда в нем находится курсор), в котором можно выбрать из списка все значения для данного поля. После щелчка, по кнопке <Применить фильтр> будут выбраны записи, соответствующие измененному фильтру. Еще более сложные условия фильтрации можно задать командой меню **Записи, Фильтр, Расширенный фильтр**.

6. ВВОД И ПРОСМОТР ДАННЫХ ПОСРЕДСТВОМ ФОРМЫ

Формы обеспечивают гибкий способ ввода, редактирования, просмотра и удаления данных и фактически являются шаблонами, управляющими отображением информации. Форма позволяет отображать одновременно все поля одной или нескольких записей. Оптимально построенная форма может вмещать несколько десятков полей на одном экране, а если полей намного больше, то для каждой записи можно создать многостраничную форму. Можно создать форму-меню для вызова других форм, таблиц, запросов или отчетов. В форме каждое поле можно разместить в точно заданном месте, выбрать для него цвет или заливку и добавить элементы управления текстом для эффективного ввода данных.

При вводе данных можно не только помещать вычисляемые поля в форму, но и добавлять расширенные правила проверки корректности ввода и элементы управления (например, переключатели, флажки, раскрывающиеся

списки). Линии, рамки, цвета и фоновые изображения улучшают внешний вид данных, облегчают восприятие формы и повышают продуктивность работы. В дополнение к этому OLE-объекты (такие, как рисунки и графики) можно увидеть только в форме или в отчете.

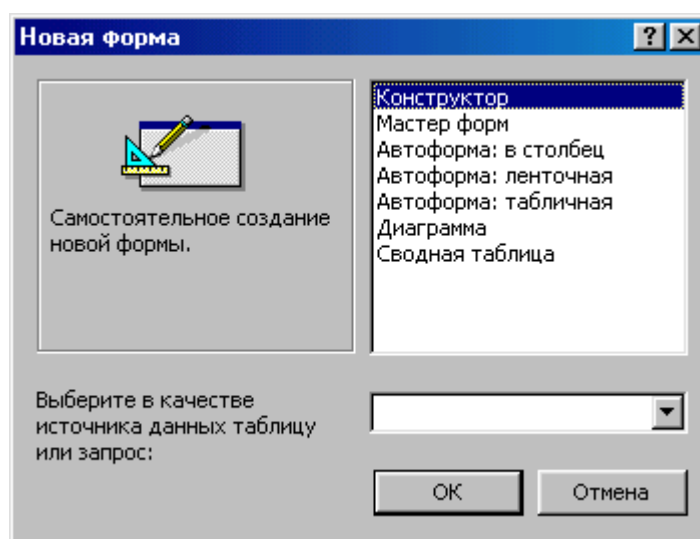
Создать форму можно несколькими способами. Если вы в режиме базы данных откроете вкладку **Формы** и щелкните по кнопке **<Создать>**, то откроется окно, в котором указаны способы создания формы.

Конструктор позволит вам создать форму самостоятельно, но это для начинающих пользователей довольно сложно.

Мастер форм дает возможность автоматически создать форму на основе выбранных полей. Access в режиме диалога выясняет у пользователя, какую форму он хочет получить, и создает ее автоматически. Если вас в этой форме что-либо не удовлетворяет, вы можете исправить это в режиме конструктора.

Автоформы являются частными случаями мастера форм, т.е. они автоматически создают заданные виды форм практически без участия пользователя. Это может быть удобно, когда базовая таблица одна, содержит немного полей и вам нужно быстро создать простую форму.

Диаграмма создает форму со встроенной диаграммой, а *Сводная таблица* – со сводной таблицей Excel.



Окно выбора вариантов построения формы

Наиболее удобным и гибким способом создания форм является Мастер

форм. В этом режиме вы можете выбрать поля таблицы для отображения в форме, стиль и цвет оформления фона и ячеек, а также вид формы. Мастер форм предлагает четыре вида формы представления данных: в один столбец, ленточная, табличная и выровненная.

Форма в один столбец

Ленточная форма и табличная форма похожи друг на друга, но табличная форма фактически повторяет вид уже знакомой вам таблицы и в ней ограничены возможности редактирования. В ленточной форме доступны все виды инструментов и способы оформления фона и ячеек с данными. Выровненный вид формы похож на ленточную форму, но отличается тем, что на экран выводятся все поля с одной записью, и если полей много, то они располагаются рядами – один под другим.

Ленточная форма

Если форма разрабатывается на основе нескольких таблиц или запросов, то Access предложит вам создать либо подчиненные, либо

связанные формы в ленточном или в табличном виде.

Код	Фамилия	Имя	Отчество	Номер группы	Телефон	Стипендия
1	Арбузов	Николай	Николаевич	151	260-15-63	<input checked="" type="checkbox"/>
2	Киршин	Петр	Валерьевич	151	110-67-82	<input checked="" type="checkbox"/>
3	Кривинский	Сергей	Николаевич	151	172-97-21	<input type="checkbox"/>
4	Крылова	Елена	Петровна	151	130-31-87	<input checked="" type="checkbox"/>
5	Кульчий	Григорий	Викторович	151	269-53-75	<input checked="" type="checkbox"/>
6	Патрикеев	Олег	Борисович	152	234-11-63	<input type="checkbox"/>
7	Перлов	Кирилл	Николаевич	152	312-21-33	<input type="checkbox"/>
8	Соколова	Наталия	Петровна	152	166-87-24	<input type="checkbox"/>
9	Степанская	Ольга	Витальевна	152	293-43-77	<input checked="" type="checkbox"/>

Табличная форма

Перемещаться по форме можно, просто щелкая по нужному полю и внося изменения или дополнения в данные. Для манипулирования записями (операции поиска, замены, сортировки и фильтрации данных) в режиме формы используются те же способы, что и в режиме таблицы.

7. ТЕХНОЛОГИЯ СОЗДАНИЯ БАЗЫ ДАННЫХ И РАБОТА С НЕЙ

Лабораторная работа №1. Создание базы данных

Задание 1. Создание базы данных и таблицы.

1. Создайте новую(пустую) базу данных.
2. Создайте таблицу **Преподаватели**.
3. Определите поля таблицы в соответствии с таблицей 1.
4. Сохраните созданную таблицу.

Технология работы

1. Для создания новой базы данных:

- загрузите Access, в появившемся окне выберите пункт **Новая база данных**;
- в окне "Файл новой базы данных" задайте имя вашей базы (пункт **Имя файла**) и выберите папку (пункт **Папка**), где ваша база данных будет находиться. По умолчанию Access предлагает вам имя базы db1, а тип файла – Базы данных Access(mdb). Имя задайте **Преподаватели**, а

тип файла оставьте прежним, так как другие типы файлов нужны в специальных случаях;

- щелкните по кнопке <Создать>.

Таблица 1. Структура таблицы **Преподаватели**

Имя поля	Тип данных	Размер поля
Код преподавателя	Счетчик	
Фамилия	Текстовый	15
Имя	Текстовый	15
Отчество	Текстовый	15
Дата рождения	Дата/время	Краткий
Должность	Текстовый	9
Дисциплина	Текстовый	11
Телефон	Текстовый	9
Зарплата	Денежный '	

2. Для создания таблицы базы данных:

- в окне базы данных выберите вкладку *Таблицы*, а затем щелкните по кнопке <Создать>;

- в окне "Новая таблица" выберите пункт **Конструктор** и щелкните по кнопке <ОК>. В результате проделанных операций открывается окно таблицы в режиме конструктора, в котором следует определить поля таблицы.

3. Для определения полей таблицы:

- введите в строку столбца "Имя поля" имя первого поля – "Код преподавателя";

- в строке столбца "Тип данных" щелкните по кнопке списка и выберите тип данных *Счетчик*. Поля вкладки *Общие* оставьте такими, как предлагает Access.

Примечание. Заполнение строки столбца "Описание" необязательно и обычно используется для внесения дополнительных сведений о поле.

Для определения всех остальных полей таблицы базы данных **Преподаватели** выполните действия, аналогичные указанным выше. Имена,

типы и размеры полей определены в таблице 1.

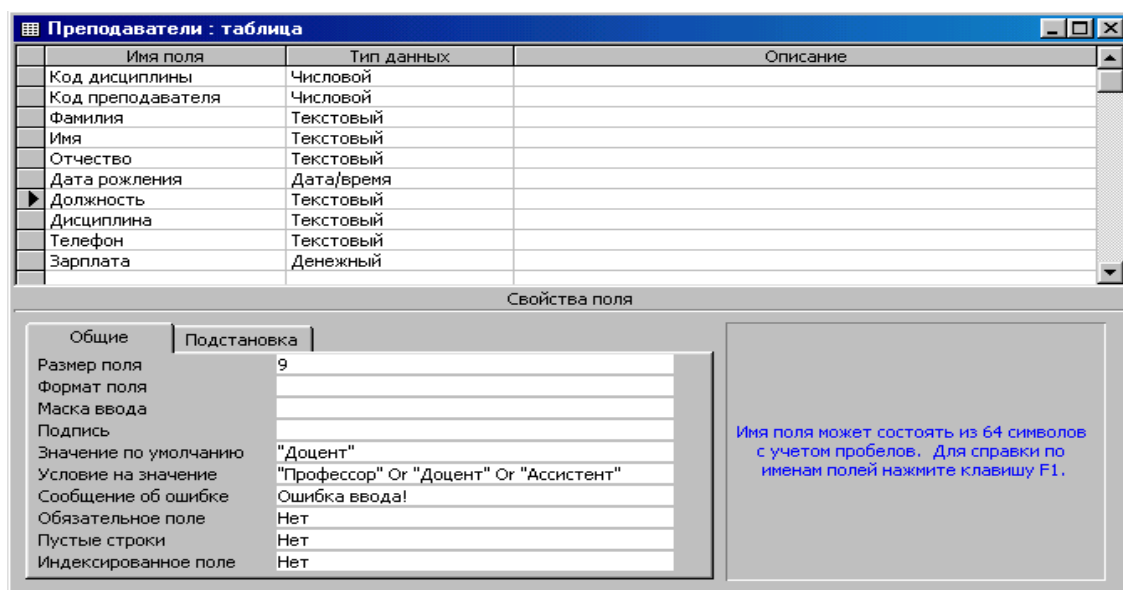


Рис. 1. Окно таблицы в режиме конструктора

Внимание! Обратите внимание на вкладку *Общие* в нижней части экрана. Советуем изменить данные в пункте *Размер поля*, а остальные пункты оставить по умолчанию (их функции рассмотрим далее). Например, для текстового типа данных Access предлагает по умолчанию длину 50 символов. Но вряд ли поле "Фамилия" будет содержать более 15 символов. Не бойтесь ошибиться – в дальнейшем можно скорректировать длину поля. Для числового типа Access предлагает *Длинное целое*, но ваши данные могут быть либо небольшие целые числа (в диапазоне от -32768 до 32767) – тогда надо выбрать *Целое*, либо дробные числа – тогда надо выбрать *С плавающей точкой*. Для выбора необходимого параметра надо щелкнуть по полю, а затем нажать появившуюся кнопку списка и выбрать необходимые данные. В результате таблица будет иметь более компактный вид, а объем базы данных уменьшится.

4. Для сохранения таблицы:

- выберите пункт меню **Файл, Сохранить**;
- в диалоговом окне "Сохранение" введите имя таблицы – **Преподаватели**;
- щелкните по кнопке <ОК>.

Примечание. В результате щелчка по кнопке <ОК> Access предложит вам

задать ключевое поле (поле первичного ключа), т.е. поле, однозначно идентифицирующее каждую запись. Для однотабличной базы данных это не столь актуально, как для многотабличной, поэтому щелкните по кнопке <Нет>.

Задание 2. Доработка структуры таблицы.

- 1 Введите ограничения на данные, вводимые в поле "Должность": в поле должны вводиться только слова *Профессор*, *Доцент* или *Ассистент*;
- 2 Задайте текст сообщения об ошибке, который будет появляться на экране при вводе неправильных данных в поле "Должность";
3. Задайте значение по умолчанию для поля "Должность" в виде слова *Доцент*.

Таблица 2

Код препод.	Фамилия	Имя	Отчество	Дата рожд.	Должность	Дисциплина	Телефон	Зарплата грн
1	Истомин	Ремир	Евгеньевич	23.10.54	Доцент	Информатика	110-44-68	890
2	Миронов	Павел	Юрьевич	25.07.40	Профессор	Экономика	312-21-40	1200
3	Гришин	Евгений	Сергеевич	05.12.67	Доцент	Математика	260-23-65	760
4	Сергеева	Ольга	Ивановна	12.02.72	Ассистент	Математика	234-85-69	450
5	Емец	Татьяна	Климовна	16.02.51	Доцент	Экономика	166-75-33	890
6	Игнатъева	Татьяна	Павловна	30.05.66	Доцент	Информатика	210-36-98	790
7	Миронов	Алексей	Николаевич	12.07.48	Доцент	Физика	166-75-33	890
8	Васильева	Рина	Петровна	4.06.62	Профессор	Информатика	234-58-00	1285
9	Корсун	Юрий	Андреевич	16.12.59	Ассистент	Экономика	173-45-26	470
10	Короткий	Никита	Николаевич	18.07.71	Доцент	Физика	235-41-27	890

4. Заполните таблицу данными в соответствии с таблицей 2 и проверьте реакцию системы на ввод неправильных данных в поле "Должность".
5. Измените ширину каждого поля таблицы в соответствии с шириной данных.

Технология работы

1. Для задания условия на значение для вводимых данных:
 - войдите в режим Конструктор для проектируемой таблицы. Если вы находитесь в окне базы данных, то выберите вкладку *Таблицы* и щелкните по кнопке <Конструктор>. Если вы находитесь в режиме таблицы, то щелкните по кнопке <Режим конструктора> на панели инструментов или выполните команду **Вид, Конструктор**;
 - в верхней части окна щелкните по полю "Должность";

- в нижней части окна щелкните по строке параметра *Условие на значение*;
 - щёлкните по кнопке <Построить> для определения условий на значение при помощи Построителя выражений;
 - в появившемся окне напишите слово *Профессор*, затем щелкните по кнопке <Ог>(эта кнопка выполняет функцию **Или**), напишите *Доцент*, снова щелкните по этой же кнопке, напишите *Ассистент* и щелкните по кнопке <ОК>. Таким образом вы ввели условие, при котором в поле "Должность" могут вводиться только указанные значения.
2. В строке *Сообщение об ошибке* введите предложение "Такой должности нет, правильно введите данные".
 3. В строке *Значение по умолчанию* введите слово "Доцент".
 4. Введите данные в таблицу в соответствии с таблицей 2. Попробуйте в поле "Должность" любой записи ввести слово *Лаборант*. Посмотрите, что получилось. На экране должно появиться сообщение: "Такой должности нет, правильно введите данные". Введите правильное слово.
 5. Для изменения ширины столбца таблицы в соответствии с шириной данных сделайте двойной щелчок в заголовке столбца.

Задание 3. Работа с таблицей Преподаватели.

1. Произведите поиск в таблице преподавателя Миронова.
2. Произведите замену данных: измените заработную плату ассистенту Сергеевой с 450 грн. на 670 грн.
3. Произведите сортировку данных в поле "Год рождения" по убыванию.
4. Произведите фильтрацию данных по полям "Должность" и "Дисциплина".
5. Просмотрите созданную таблицу, какой она будет выглядеть на листе бумаги при печати.

Технология работы

1. Для поиска в таблице преподавателя Миронова:
 - переведите курсор в первую строку поля "Фамилия";
 - выполните команду **Правка, Найти**;
 - в появившейся строке параметра *Образец* введите *Миронов*;

- в строке параметра *Просмотр* должно быть слово ВСЕ (имеется в виду искать по всем записям);
- в строке параметра *Совпадение* выберите из списка *С любой частью поля*;
- в строке параметра *Только в текущем поле* установите флажок (должна стоять "галочка");
- щелкните по кнопке <Найти>. Курсор перейдет на вторую запись и выделит слово *Мионов*;
- щелкните по кнопке <Найти далее>. Курсор перейдет на седьмую запись и также выделит слово *Мионов*;
- щелкните по кнопке <Закреть> для выхода из режима поиска.

2. Для замены заработной платы ассистенту Сергеевой с 450 грн. на 670 грн.:

- переведите курсор в первую строку поля "Зарплата";
- выполните команду **Правка, Заменить**;
- в появившемся окне в строке *Образец* введите 450;
- в строке *Заменить на* введите 670. Обратите внимание на остальные опции – вам надо вести поиск по всем записям данного поля;
- щелкните по кнопке <Найти далее>. Курсор перейдет на четвертую запись, но здесь не нужно менять данные, поэтому снова щелкните по кнопке <Найти далее>. Курсор перейдет на девятую запись – это то, что нам надо;
- щелкните по кнопке <Заменить>. Данные будут изменены;

Примечание. Чтобы заменить сразу все данные, надо воспользоваться кнопкой <Заменить все>.

- Щелкните по кнопке <Закреть>.

3. Для сортировки данных в поле "Год рождения" по убыванию:

- щелкните по любой записи поля "Год рождения";
- щелкните по кнопке <Сортировка> на панели управления или выполните команду **Записи, Сортировка, Сортировка по убыванию**. Все данные в таблице будут отсортированы в соответствии с убыванием значений в поле "Год рождения".

4. Для фильтрации данных по полям "Должность" и "Дисциплина":

- щелкните по записи *Доцент* поля "Должность";
- щелкните по кнопке <Фильтр по выделенному> или выполните

команду **Записи, Фильтр, Фильтр по выделенному**. В таблице останутся только записи о преподавателях-доцентах;

- щелкните по записи *Информатика* поля "Дисциплина";
- щелкните по кнопке <Фильтр по выделенному> или выполните команду **Записи, Фильтр, Фильтр по выделенному**. В таблице останутся только записи о преподавателях-доцентах кафедры информатики;

- для отмены фильтрации щелкните по кнопке <Удалить фильтр> на панели инструментов или выполните команду **Записи, Удалить фильтр**. В таблице появятся все данные.

5. Для просмотра созданной таблицы:

- щелкните по кнопке <Просмотр> или выполните команду **Файл, Предварительный просмотр**. Вы увидите таблицу как бы на листе бумаги;

- закройте окно просмотра.

Примечание. Если вы захотите изменить поля или ориентацию таблицы на листе бумаги, выполните команду **Файл, Параметры страницы**. В открывшемся окне можете изменять указанные параметры.

Лабораторная работа №2. Ввод и просмотр данных посредством формы

Задание. Создание и использование формы.

1. С помощью Мастера форм создайте форму Состав преподавателей (тип – форма в один столбец).
2. Найдите запись о доценте Гришине, находясь в режиме формы.
3. Измените зарплату ассистенту Корсун с 470 грн. на 590 грн.
4. Произведите сортировку данных в поле "Фамилия" по убыванию.
5. Произведите фильтрацию данных по полю "Должность".
6. Измените название поля "Дисциплина" на "Преподаваемая дисциплина".
7. Просмотрите форму с точки зрения того, как она будет выглядеть на листе бумаги.

Технология работы

1. Для создания формы **Состав преподавателей**:

- откройте вкладку *Формы* в окне базы данных; щелкните по кнопке <Создать>;
- в появившемся окне выберите (подведите курсор мыши и щелкните левой кнопкой) пункт **Мастер форм**;
- щелкните по значку списка в нижней части окна;
- выберите из появившегося списка таблицу "Преподаватели";
- щелкните по кнопке <ОК>;
- в появившемся окне выберите поля, которые будут присутствовать в форме. В данном примере присутствовать будут все поля, поэтому щелкните по кнопке <Все поля>;
- щелкните по кнопке <Далее>;
- в появившемся окне уже выбран вид *Форма в один столбец*, поэтому щелкните по кнопке <Далее> в появившемся окне выберите стиль оформления. После выбора стиля щелкните по кнопке <Далее>;
- в появившемся окне задайте имя формы, набрав на клавиатуре параметр *Состав преподавателей*. Остальные параметры в окне оставьте без изменений;
- щелкните по кнопке <Готово>. Перед вами откроется форма в один столбец. Столбец слева – это названия полей, столбец справа – данные первой записи (в нижней части окна в строке параметра *Запись* стоит цифра 1). Для перемещения по записям используйте кнопки счетчиков записей.

2. Для поиска преподавателя Миронова:

- переведите курсор в первую строку поля "Фамилия";
- выполните команду **Правка, Найти**;
- в появившемся окне в строке *Образец* введите фамилию Миронов;
- в строке параметра *Просмотр* должно быть слово *ВСЕ* (имеется в виду искать по всем записям);
- в строке параметра *Совпадение* выберите из списка параметр *С любой частью поля*;
- в строке параметра *Только в текущем поле* установите флажок;

- щелкните по кнопке <Найти>. Курсор перейдет на вторую запись и выделит слово Миронов;
 - щелкните по кнопке <Найти далее>. Курсор перейдет на седьмую запись и также выделит слово Миронов;
 - щелкните по кнопке <Заккрыть> для выхода из режима поиска.
3. Для замены зарплаты ассистенту Корсун с 470 грн. на 590 грн.:
- переведите курсор в первую строку поля "Зарплата";
 - выполните команду **Правка, Заменить**;
 - в появившемся окне в строке параметра *Образец* введите 470 ;
 - в строке параметра *Заменить на* введите 590. Обратите внимание на остальные опции – вам надо вести поиск по всем записям данного поля;
 - щелкните по кнопке <Найти далее>. Курсор перейдет на четвертую запись, но здесь не нужно менять данные, поэтому снова щелкните по кнопке <Найти далее>. Курсор перейдет на девятую запись – это то, что нам надо;
 - щелкните по кнопке <Заменить>. Данные будут изменены;
 - щелкните по кнопке <Заккрыть>.
4. Для сортировки данных в поле "Год рождения" по убыванию:
- щелкните по любой записи поля "Год рождения";
 - щелкните по кнопке <Сортировка> на панели управления или выполните команду **Записи, Сортировка, Сортировка по убыванию**. Все данные в таблице будут отсортированы в соответствии с убыванием значений в поле "Год рождения".
5. Для фильтрации данных по полю "Должность":
- щелкните по записи *Доцент* поля "Должность";
 - щелкните по кнопке <Фильтр по выделенному> или выполните команду **Записи, Фильтр, Фильтр по выделенному**. В форме останутся только записи о преподавателях – доцентах;
 - щелкните по записи *Информатика* поля "Дисциплина";
 - щелкните по кнопке <Фильтр по выделенному> или выполните команду **Записи, Фильтр, Фильтр по выделенному**. В форме останутся только записи о преподавателях-доцентах кафедры информатики;
 - для отмены фильтра щелкните по кнопке <Удалить фильтр>на

панели инструментов или выполните команду **Записи, Удалить фильтр**.

В таблице появятся все данные.

6. Измените название поля "Дисциплина" на "Преподаваемая дисциплина".

Для этого:

- перейдите в режим конструктора, щелкнув по кнопке <Режим конструктора> на панели инструментов или выполнив команду **Вид, Конструктор**;

- щелкните правой кнопкой мыши в поле "Дисциплина" (на названии поля – оно слева, а строка справа с именем *Дисциплина* – это ячейка для данных, свойства которых мы не будем менять). В появившемся меню выберите пункт *Свойства*. На экране откроется окно свойств поля "Дисциплина";

- щелкните по строке с именем *Подпись*, т.е. там, где находится слово *Дисциплина*;

- сотрите слово *Дисциплина* и введите *Преподаваемая дисциплина*;

- для просмотра результата перейдите в режим формы, выполнив команду **Вид, Режим формы**.

7. Для просмотра созданной формы:

- щелкните по кнопке <Просмотр> или выполните команду **Файл, Предварительный просмотр**. Вы увидите форму как бы на листе бумаги;

- закройте окно просмотра.

Лабораторная работа №3. Формирование запросов и отчетов для однотабличной базы данных

Краткая справка о формировании запросов на выборку

Запросы являются мощным средством обработки данных, хранимых в таблицах Access. С помощью запросов можно просматривать, анализировать и изменять данные из нескольких таблиц. Они также используются в качестве источника данных для форм и отчетов. Запросы позволяют вычислять итоговые значения и выводить их в компактном формате, подобном формату электронной таблицы, а также выполнять вычисления над группами записей.

Запросы можно создавать самостоятельно и с помощью мастеров.

Мастера запросов автоматически выполняют основные действия в зависимости от ответов пользователя на поставленные вопросы. Самостоятельно разработать запросы можно в режиме конструктора.

В Access можно создавать следующие типы запросов:

- запрос на выборку;
- запрос с параметрами;
- перекрестный запрос;
- запрос на изменение (запрос на удаление, обновление и добавление записей, на создание таблицы);
- запросы SQL (запросы на объединение, запросы к серверу, управляющие запросы, подчиненные запросы)

Запрос на выборку используется наиболее часто. При его выполнении данные, удовлетворяющие условиям отбора, выбираются из одной или из нескольких таблиц и выводятся в определенном порядке. Например, можно вывести на экран данные о фамилиях доцентов, стаж которых более 15 лет (на основе таблицы "Преподаватели").

Запрос на выборку можно также использовать, чтобы сгруппировать записи для вычисления сумм, средних значений, пересчета и других действий. Например, используя запрос на выборку, можно получить данные о среднем стаже доцентов и профессоров (на основе таблицы **Преподаватели**).

Запрос с параметрами – это запрос, при выполнении которого в его диалоговом окне пользователю выдается приглашение ввести данные, на основе которых будет выполняться запрос. Например, часто требуются данные о том, какие дисциплины ведут преподаватели. Чтобы не создавать отдельные запросы по каждому преподавателю, можно создать один запрос с параметрами, где в качестве параметра будет использоваться фамилия преподавателя. При каждом вызове этого запроса вам будет предложено ввести фамилию преподавателя, а затем на экран будут выведены все поля, которые вы указали в запросе, например фамилия, имя, отчество преподавателя и читаемая им дисциплина.

Для создания нового запроса надо в окне базы данных выбрать вкладку *Запросы* и щелкнуть по кнопке <Создать>. Откроется окно "Новый запрос", в котором вы должны выбрать один из пяти пунктов: *Конструктор*, *Простой*

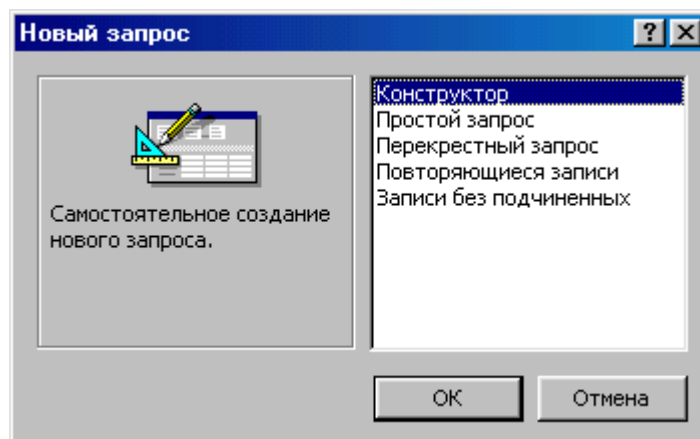


Рис.2. Окно выбора вариантов построения запросов

запрос, Перекрестный запрос, Повторяющиеся записи, Записи без подчиненных.

Конструктор позволит вам самостоятельно создать любой тип запроса, но этот режим рекомендуется пользователям, уже имеющим некоторый опыт создания запросов.

Простой запрос позволит создать с помощью Мастера запрос на выборку из определенных полей таблиц или других запросов. (Это наилучший способ создания запроса для начинающих пользователей.).

При выборе пункта *Повторяющиеся записи* будет создан запрос на поиск повторяющихся записей в простой таблице или в запросе.

При выборе пункта *Записи без подчиненных* – запрос на поиск записей, которым не соответствует ни одна запись в подчиненной таблице. Такой запрос используется для многотабличных баз данных.

При выполнении запроса на выборку Access извлекает записи из таблиц и формирует результирующий набор данных. Он выглядит, как таблица, хотя и не является ею. Результирующий набор данных является динамическим (или виртуальным) набором записей и не хранится в базе данных.

После закрытия запроса результирующий набор данных этого Запроса прекращает свое существование. Хотя сам по себе динамический набор данных больше не существует, помните, что данные, которые в нем содержались, остаются в базовых таблицах.

При сохранении запроса остается только структура запроса – перечень таблиц, список полей, порядок сортировки, ограничения на записи, тип запроса

и т.д. При сохранении в базе данных запрос, по сравнению с результирующим набором данных, имеет ряд преимуществ:

- на физическом носителе информации (обычно это жесткий диск) требуется меньший объем пространства
- запрос может использовать обновленные версии любых записей, измененных со времени последнего запуска запроса.

При каждом выполнении запрос обращается к базовым таблицам и снова создает результирующий набор данных. Поскольку сам по себе результирующий набор данных не сохраняется, запрос автоматически отображает любые изменения, происшедшие в базовых таблицах с момента последнего запуска этого запроса (даже в реальном времени в многопользовательской среде).

Для сохранения запроса следует выполнить следующие действия. Выполните команду **Файл, Сохранить** или щелкните по кнопке <Сохранить> на панели инструментов. Если вы впервые сохраняете запрос, введите новое имя запроса в диалоговом окне "Сохранение".

Задание 1. Формирование запросов на выборку на основе таблицы Преподаватели.

1. Создайте простой запрос на выборку, в котором будут отображаться фамилии, имена, отчества преподавателей и их должности.
 - Данные запроса отсортируйте по должностям.
 - Сохраните запрос.
2. Создайте запрос на выборку с параметром, в котором должны отображаться фамилии, имена, отчества преподавателей и преподаваемые ими дисциплины;
 - в качестве параметра запроса задайте фамилию преподавателя;
 - выполните этот запрос для преподавателя Гришина.
3. Создайте запрос с группировкой преподавателей по должностям и подсчетом средней зарплаты преподавателей в каждой группе.
4. Создайте запрос на выборку **Возраст преподавателя** с созданием нового поля "Возраст преподавателя"; запрос кроме нового поля должен содержать поля "Фамилия", "Имя", "Отчество", "Должность".
5. Создайте самостоятельно запрос с группировкой преподавателей по должностям и подсчетом количества преподавателей в каждой группе.

6. Создайте самостоятельно запрос с группировкой преподавателей по читаемым дисциплинам и подсчетом количества преподавателей в каждой группе.

Технология работы

1. Для создания простого запроса:

- в окне базы данных откройте вкладку *Запросы*;
- в открывшемся окне щелкните по кнопке <Создать>;
- из появившихся пунктов окна "Новый запрос" выберите *Простой запрос* и щелкните по кнопке <ОК>;
- в появившемся окне в строке *Таблицы/Запросы* выберите таблицу **Преподаватели** (т.к. других таблиц или запросов не было создано, она будет одна в раскрывающемся списке);
- в окне "Доступные поля" переведите выделение на параметр *Фамилия*;
- щелчком по кнопке переведите поле "Фамилия" окно "Выбранные поля";
- аналогично в окно "Выбранные поля" переведите поля "Имя", "Отчество", "Должность" (порядок важен — в таком порядке данные и будут выводиться);
- щелкните по кнопке <Далее>;
- в строке параметра *Задайте имя запроса* введите новое имя **Должности преподавателей**;
- щелкните по кнопке <Готово>. На экране появится таблица с результатами запроса.
- **Для сортировки данных:**
- щелкните в любой строке поля "Должность";
- отсортируйте данные по убыванию. Для этого щелкните по кнопке <Сортировка по убыванию> на панели инструментов или выполните команду *Записи, Сортировка, Сортировка по убыванию*.
- **Для сохранения запроса:**
- щелкните по кнопке <Сохранить> или выполните команду **Файл, Сохранить**;

- закройте окно запроса.
2. Для создания запроса на выборку с параметром:
- создайте запрос на выборку из следующих полей таблицы "Преподаватели": "Фамилия", "Имя", "Отчество", "Преподаваемая дисциплина".
 - задайте имя запросу **Преподаваемые дисциплины**;
 - щелкните по кнопке <Готово>. На экране появится таблица с результатами запроса;
 - перейдите в режиме конструктора, щелкнув по кнопке <Режим конструктора>(или выполнив команду **Вид, Конструктор**);
 - в строке параметра *Условия отбора* для поля "Фамилия" введите фразу (скобки вводить обязательно): [Введите фамилию преподавателя];
 - выполните запрос, щелкнув по кнопке <Запуск запроса> на панели инструментов или выполнив команду **Запрос, Запуск**;
 - в появившемся окне введите фамилию Гришин и щелкните по кнопке <ОК>. На экране появится таблица с данными о преподавателе Гришине – его имя, отчество и преподаваемая им дисциплина;
 - сохраните запрос;
 - закройте окно запроса.
3. Действуя в соответствии с п.1
- переведите в бланк запроса поля "Должность" и "Зарплата";
 - введите имя запроса – **Средние зарплаты по должностям**;
 - установите флажок в ячейке *Изменение макета запроса*;
 - нажмите кнопку <Готово>;
 - в бланке запроса в строке *Сортировка* поля "Должность" введите группировку(групповые операции). Это можно сделать с помощью контекстного меню или кнопкой <Групповые операции> панели инструментов;
 - в поле "Зарплата" в этой же строке из контекстного меню выберите функцию **Avg**(Average – среднее значение);
 - выполните запрос и сохраните его.
4. Действуя в соответствии с п.3
- переведите в бланк запроса поля "Фамилия", "Имя", "Отчество", "Должность";

- введите имя нового запроса;
- в бланке запроса установите курсор в первую строку пустого поля, следующего за полем "Отчество";
- нажмите кнопку <Построить> панели инструментов или выполните одноименную команду контекстного меню;
- в поле Построителя введите такую запись: `Возраст: DateDiff("уууу";[Дата рождения];Now())`.
- Запись можно выполнить автоматически. Для этого:
- сделайте двойной щелчок по пункту "Функции";
- раскройте папку "Встроенные функции";
- в окне категорий выберите категорию "Дата/Время";
- из списка функций выберите DateDiff;
- нажмите кнопку <Вставить>;
- удалите запись "Выражение";
- замените запись "Интервал" на "уууу" – этот параметр функции означает, что интервал между двумя датами будет измеряться в годах;
- замените запись "Дата1" на [Дата рождения];
- замените запись "Дата2" на Now();
- удалите записи "firstweekday" и "firstweek";
- щелкните на кнопке <Ок>.
- Окно Построителя закроется, и вы увидите новое поле, в первую строку которого скопирована запись, сделанная в Построителе;
- перейдите в режим таблицы, и вы увидите заполненным созданное поле "Возраст";
- закройте запрос.

Формирование отчетов

Отчет – это гибкое и эффективное средство для организации просмотра и распечатки итоговой информации. В отчете можно получить результаты сложных расчетов, статистических сравнений, а также поместить в него рисунки и диаграммы.

Пользователь имеет возможность разработать отчет самостоятельно или создать отчет с помощью мастера. Мастер по разработке отчетов выполняет всю рутинную работу и позволяет быстро разработать отчет. После вызова

Мастера выводятся диалоговые окна с приглашением ввести необходимые данные, и отчет создается на основании ответов пользователя. Мастер необходим даже для опытных пользователей, так как позволяет быстро разработать макет, служащий основой создаваемого отчета. После этого можно переключиться в режим конструктора и внести изменения в стандартный макет.

При работе с мастером в зависимости от того, какой отчет вы хотите создать (т.е. как вы отвечаете на вопросы мастера), Access предлагает вам различные варианты макетов отчета. Например, если вы создаете простой отчет без группировки данных, то вам предлагается три варианта макета: в столбец, табличный и выровненный. При этом в небольшом окне представляется вид этих макетов. Если вы задаете уровни группировки (т.е. признаки, по которым надо сгруппировать данные, например по должности), то вам предлагаются шесть видов макетов.

Основное различие между отчетами и формами заключается в их назначении. Если формы задуманы преимущественно для ввода данных, то отчеты – для просмотра данных (на экране либо на бумаге). В формах используются вычисляемые поля (обычно с помощью вычислений на основе полей в текущей записи). В отчетах вычисляемые поля (итоги) формируются на основе общей группы записей, страницы записей или всех записей отчета. Все, что можно сделать с формой (за исключением ввода данных), можно сделать и с отчетом. Действительно, форму можно сохранить в виде отчета, а затем изменить элементы управления формы в окне конструктора отчета.

Для создания отчета надо открыть вкладку Отчеты и щелкнуть по кнопке <Создать>. Откроется окно "Новый отчет", в котором приведены шесть пунктов меню, т.е. шесть способов создания отчета: конструктор, Мастер отчетов, Автоотчет в столбец, Автоотчет ленточный, Мастер диаграмм и Почтовые наклейки.

Конструктор позволит вам самостоятельно создать отчет, но это непросто даже для опытного пользователя.

Мастер отчетов автоматически создаст отчет на основе выбранных вами полей таблиц (запросов) и макетов отчетов. Этот способ создания отчетов является наиболее удобным как для начинающих, так и для опытных пользователей.

Автоотчет в столбец и Автоотчет ленточный – простейшие способы создания отчетов: достаточно указать только имя таблицы или запроса, на

основе которого будет создан отчет, а остальное сделает мастер отчетов.

Мастер диаграмм поможет создать отчет в виде диаграммы.

Почтовые наклейки создадут отчет, отформатированный для печати почтовых наклеек.

Задание 2. Построение отчета.

На основе таблицы **Преподаватели** создайте отчет с группированием данных по должностям.

Технология работы

Для создания отчета:

- откройте вкладку *Отчеты* и щелкните по кнопке <Создать>;
- в открывшемся окне выберите пункт *Мастер отчетов*;
- щелкните по значку раскрывающегося списка в нижней части окна;
- выберите из появившегося списка таблицу **Преподаватели**;
- щелкните по кнопке <ОК>. В появившемся окне выберите поля, которые будут присутствовать в отчете. В данном отчете будут присутствовать все поля таблицы.
- щелкните по кнопке <Далее>;
- в появившемся окне присутствует перечень полей. Требуется указать, по каким полям будет выполнена группировка. Переведите в правое окно поле "Должность". Таким образом вы задали группировку данных по должности;
- щелкните по кнопке <Далее>;
- сортировку в отчете выполнять не будем, поэтому в окне "Сортировка" щелкните по кнопке <Далее>;
- в появившемся окне выберите стиль оформления отчета;
- щелкните по кнопке <Далее>;
- в появившемся окне введите название отчета – **Преподаватели**;
- щелкните по кнопке <Готово>. На экране появится сформированный отчет;
- просмотрите, а затем закройте отчет.

ЛИТЕРАТУРА

1. Экономическая информатика. Учебник для вузов. Под ред. проф. В.В. Евдокимова. — СПб.: Питер, 1997. — 592 с.
2. Т.С. Карпова Базы данных: модели, разработка, реализация. Учебник. — М.;Х.; Минск ; СПб : "Питер", 2002. — 303 с.
3. Хансен Г., Хансен Дж. Базы данных: разработка и управление — М.: БИНОМ, 1999. — 704 с.
4. Дейт К.Дж. Введение в системы баз данных — К.; М.; СПб.: "Вильямс", 1999. — 848 с.
5. Праг К.Н., Ирвин М.Р. Библия пользователя Access для Windows 95 — К.: Диалектика, 1996. — 576 с.
6. Мартин Дж. Организация баз данных в вычислительных системах. — М.: Мир, 1980. — 458 с.
7. Атре Ш. Структурный подход к организации баз данных. — М.: ФиС, 1983. — 560 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	3
Анализ предметной области и создание концептуальной модели предметной области	4
2. СОЗДАНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ	6
Определение объектов (сущностей)	6
Определение взаимосвязей между сущностями.....	8
Взаимосвязь "один-к-одному" (между двумя типами объектов)	8
Взаимосвязь "один-ко-многим" (между двумя типами объектов)	9
Взаимосвязь "многие-ко-многим" (между двумя типами объектов)	10
Приведение модели к требуемому уровню нормальной формы	11
3. ИНСТРУМЕНТАЛЬНЫЕ КОМПОНЕНТЫ СУБД ACCESS	14
Общие сведения о таблицах	14
Использование запросов	15
Назначение форм	16
Отчеты	16
Автоформы. Автоотчеты	17
4. СОЗДАНИЕ БАЗЫ ДАННЫХ	17
Создание таблицы	18
Окно таблицы в режиме конструктора	19
Сохранение таблицы	19
Типы данных в структуре таблицы	20
Поля первичного ключа	21
Установка режимов отображения таблицы. Способы переключения	

между ними	21
Модификация структуры таблицы	22
5. УПРАВЛЕНИЕ ОТОБРАЖЕНИЕМ ДАННЫХ	23
Общие свойства различных типов данных	26
Работа с данными в режиме таблицы	27
Сортировка данных	29
Отбор данных с помощью фильтра	30
6. ВВОД И ПРОСМОТР ДАННЫХ ПОСРЕДСТВОМ ФОРМЫ	31
7. ТЕХНОЛОГИЯ СОЗДАНИЯ БАЗЫ ДАННЫХ И РАБОТА С НЕЙ	34
Лабораторная работа №1. Создание базы данных	34
Задание 1. Создание базы данных и таблицы	34
Технология работы	34
Задание 2. Доработка структуры таблицы	37
Технология работы	37
Задание 3. Работа с таблицей Преподаватели	38
Технология работы	38
Лабораторная работа №2. Ввод и просмотр данных посредством форм	40
Задание. Создание и использование формы	40
Технология работы	41
Лабораторная работа №3. Формирование запросов и отчетов для однотабличной базы данных	43
Краткая справка о формировании запросов на выборку	43
Задание 1. Формирование запросов на выборку на основе таблицы Преподаватели.....	46
Технология работы	47
Формирование отчетов	49
Задание 2. Построение отчета	51
Технология работы	51
ЛИТЕРАТУРА	52

Учебное издание

Швачич Геннадий Григорьевич
Рыжанкова Галина Ивановна
Кузьменко Вячеслав Витальевич
Барвинов Виталий Петрович

**ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ В СУБД ACCESS
ЧАСТЬ 1**

Учебное пособие

Тем. план 2006, поз. 168

Подписано к печати 03.05.06. Формат 60x84 ¹/₁₆. Бумага типогр. Печать
плоская. Уч.- изд. л.3,17. Усл. печ. л. 3,13. Тираж 100 экз. Заказ №

Национальная металлургическая академия Украины
49600, Днепропетровск – 5, пр. Гагарина, 4

Редакционно – издательский отдел НМетАУ