

Основы работы в Microsoft Office Access

Введение в базы данных и Microsoft Access

База данных (БД) — это организованная структура, предназначенная для хранения информации. Обычно БД представляются в виде совокупности взаимосвязанных файлов или таблиц, предназначенных для решения конкретной задачи.

С понятием БД тесно связано понятие **системы управления базой данных (СУБД)**. **СУБД** — это комплекс программных средств, предназначенных для создания структуры новой базы, наполнения ее содержимым, редактирования содержимого и визуализации информации.

СУБД Access входит в состав **Microsoft Office** и предназначена для работы с **реляционными БД**, т.е. представленными в табличной форме. В отличие от табличного процессора Excel, Access имеет более развитые средства для отбора данных из взаимосвязанных таблиц, формирования новых таблиц и отчетов.

Характерной особенностью баз данных, созданных в Access, является хранение создаваемых таблиц и средств для обработки данных в одном файле, имеющем расширение **.mdb**. Достоинством Access является возможность создания СУБД (т.е. программы управления) без программирования. Однако, для сложных СУБД применение программирования на встроенном языке **Visual Basic for Applications (VBA)** позволяет повысить эффективность системы управления.

Основным элементом БД является таблица. Столбцы таблицы БД называются **полями**, а **строки** — **записями**. Первым этапом создания таблицы БД является задание ее структуры, т.е. определение количества и типа полей. Вторым этапом является ввод и редактирование записей в таблицу. БД считается созданной, даже если она пустая.

Поля таблицы просто определяют ее структуру и групповые свойства данных, записываемых в ячейках. Рассмотрим основные свойства полей БД.

1. **Имя поля** — определяет как надо обращаться к данным поля (имена используются как заголовки таблиц).
2. **Тип поля** — определяет тип данных, которые могут содержаться в данном поле (текстовые, числовые, дата, Мемо, денежный, счетчик и др.).
3. **Размер поля** — определяет предельную длину данных, которые могут размещаться в поле.
4. **Формат поля** — способ форматирования данных в ячейках.

Типы таблиц и ключей в реляционных базах данных

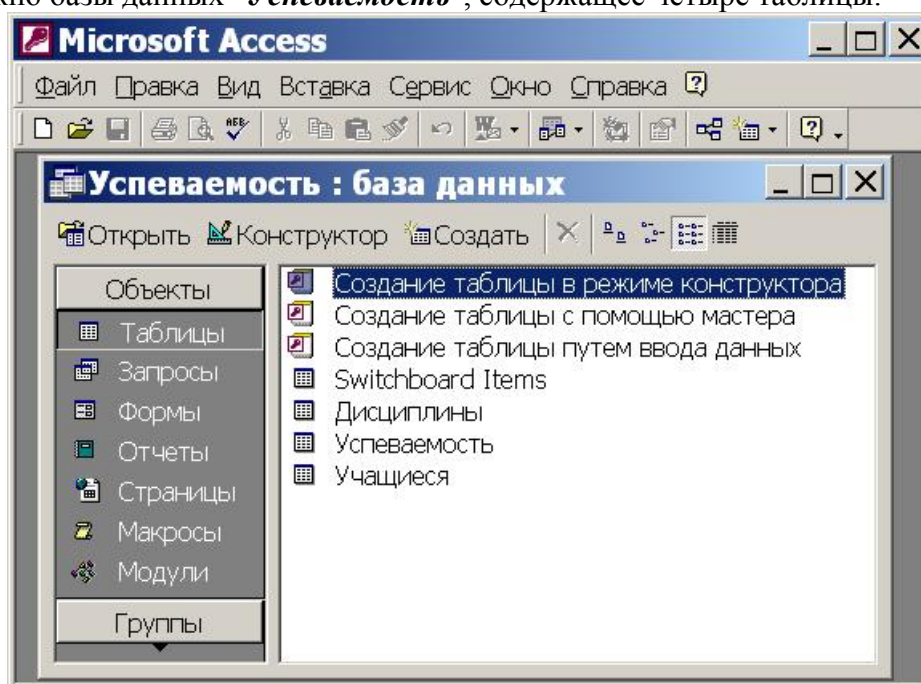
Реляционные базы данных характеризуются наличием некоторых типов таблиц и **ключей**, позволяющих определить отношения между таблицами. Для того чтобы понять принципы разработки реляционных баз данных, требуется дать определения различных типов реляционных ключей и таблиц:

- **Базовая таблица.** В реляционной базе данных базовой таблицей называется таблица, которая включает один или несколько столбцов свойств объекта и содержит **первичный ключ**, который однозначно определяет этот объект. Более того, базовая таблица должна содержать первичный ключ. Базовые таблицы часто называют **первичными**, поскольку они имеют **первичный ключ**.
- **Промежуточная таблица.** Таблица, не являющаяся базовой (т. к. она не объединяет свойства объекта или не содержит поле первичного ключа), которая используется для обеспечения связей между другими таблицами, называется **таблицей отношений**. Ключевые поля в таблицах отношений должны быть **внешними ключами**, связанными с первичными ключами базовой таблицы. Проще говоря, таблица отношений состоит только из внешних ключей и не содержит независимых элементов данных.

- **Первичный ключ.** Первичный ключ состоит из набора значений, которые однозначно определяют запись базовой таблицы. Любому значению первичного ключа должна соответствовать одна и только одна строка таблицы. Первичный ключ включает одно поле только в том случае, если это поле не содержит повторяющихся значений.
- **Составные ключи.** Если для выполнения условий, накладываемых на значения первичного ключа, заданный ключ включает несколько полей таблицы, то тогда он называется **составным**.
- **Внешние ключи.** Внешний ключ — это столбец, значения которого соответствуют значениям первичного ключа другой связанной таблицы.

Управление средой MS Access

После запуска **Access** на экран выводится окно базы данных со списком объектов. Основные объекты: таблицы, запросы, формы, отчеты, макросы, модули. С этими объектами можно работать в двух режимах: конструктора и воспроизведения. На рисунке показано окно базы данных "**Успеваемость**", содержащее четыре таблицы.



Основные объекты окна БД имеют следующее назначение:

- **таблица** — основное средство для хранения информации в БД;
- **запрос** — это инструмент для извлечения необходимой информации из исходных таблиц и представления ее в удобной форме;
- **форма** — это основное средство для ввода данных, управления СУБД и вывода результатов на экран монитора;
- **отчет** — это специальное средство для формирования выходных документов и вывода их на принтер;
- **макросы** в **Access** представляют собой совокупность внутренних команд, предназначенных для автоматизации работы с БД;
- **модули** являются программами, создаваемыми средствами языка VBA, и похожи на макросы в **Word** и **Excel**.

Таблицы и запросы связываются между собой с помощью **схемы данных**.

Таблицы, запросы, формы и отчеты БД можно создавать в двух режимах: вручную с помощью конструктора или при помощи Мастера. Выбор средства определяется конкретными обстоятельствами, однако следует заметить, что мастер быстро создает заготовку объекта, которую обычно требуется "дорабатывать" вручную.

Технология разработки СУБД содержит несколько этапов, основными из которых являются:

- проектирование структуры БД и связей между таблицами;
- разработка структуры отдельных таблиц и ввод данных в таблицы;

- разработка запросов;
- разработка схемы данных, реализующей запрооектированные связи между таблицами и запросами;
- разработка макросов и программных модулей для управления БД;
- разработка форм для реализации интерфейса управления БД;
- разработка отчетов для печати документов.

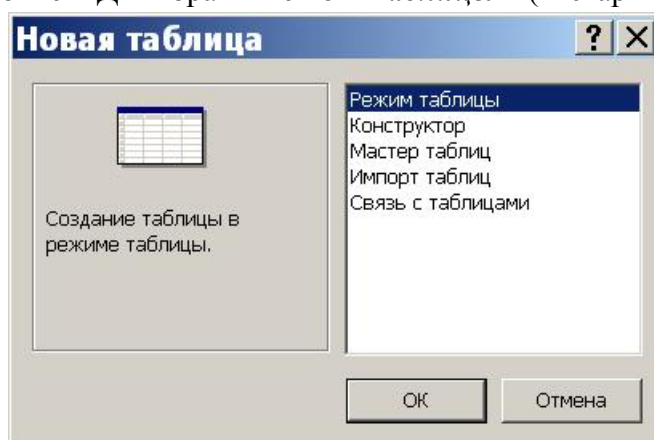
Приведенная последовательность этапов не является жесткой. Обычно разработчику СУБД приходится многократно возвращаться к одним и тем же этапам, постепенно уточняя проект.

Создание таблиц базы данных

Создание таблиц БД является первым шагом в разработке СУБД. Таблицы предназначены для хранения информации БД. Создание таблицы состоит из двух этапов: задание структуры таблицы; ввод записей в таблицу.

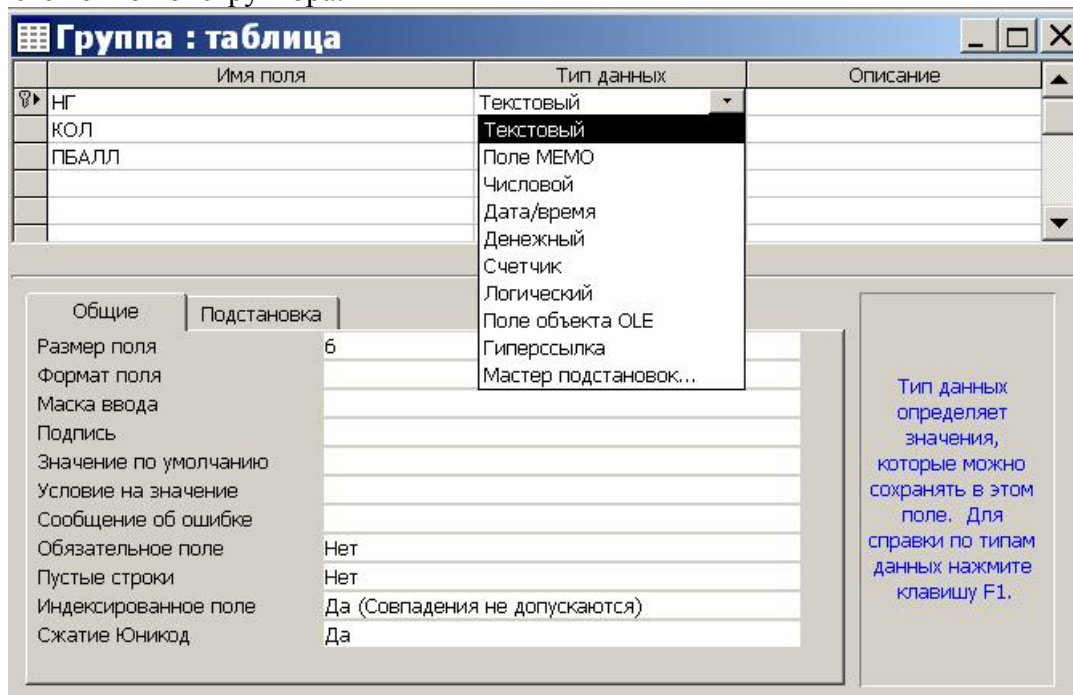
Для создания новой таблицы следует в окне БД выбрать меню **"Таблицы"** (в старых версиях Access для этой цели служит вкладка **"Таблицы"**) и нажать кнопку **"Создать"**. В окне **"Новая таблица"** необходимо выбрать способ создания таблицы.

Если у разработчика СУБД нет достаточного опыта, рекомендуется для создания таблицы выбрать режим конструктора.



Создание таблицы в режиме конструктора

При выборе режима конструктора появляется окно конструктора.



В поле **"Имя поля"** вводится имя поля таблицы, являющееся его идентификатором. Рекомендуется формировать имена полей так, чтобы имя было коротким, не содержащим специальных символов (запятых, пробелов и т.д.) и отражающим смысл поля. Необходимо различать идентификатор поля в структуре таблицы и заголовок поля в выходном

документе. Во втором случае заголовок поля должен в полной мере отражать смысл поля и обычно задается вручную при форматировании выходного документа.

В поле **"Тип данных"** выбирается один из типов, показанных в раскрытом списке на рисунке.


Поле **"Описание"** содержит комментарии к создаваемому полю таблицы. Его заполнение не является обязательным.

В нижней части окна, на вкладке **"Общие"** задаются свойства поля. Если щелкнуть кнопкой мыши по одной из строк таблицы свойств, справа появится подсказка о назначении этого свойства.

Вкладка **"Подстановка"** служит для организации подсказки при заполнении внешних ключей таблицы. Ключи, чаще всего, являются формальными идентификаторами записей в таблицах БД. Поэтому при заполнении внешних ключей у пользователя могут возникнуть затруднения, связанные с определением, какому ключу в базовой таблице соответствует запись в подчиненной таблице. Настройка свойств вкладки "Подстановка" позволяет превратить простое поле для внешнего ключа в поле со списком, содержащим полный список ключей базовой таблицы и соответствующие этим ключам поля – подсказки из базовой таблицы. После выбора в списке записи, в заполняемую таблицу помещается только ключ. Таким образом, пользователю не нужно помнить ключи и их ввод выполняется выбором из списка, а не вводом с клавиатуры. В дальнейшем поля подстановки наследуются формами, если в качестве источника данных формы выбрана таблица с такими полями.

На приведенном выше рисунке описана таблица **"Группа"**, состоящая из трех полей с идентификаторами **НГ**, **КОЛ** и **ПБАЛЛ**. Поле **НГ** является текстовым и содержит номера студенческих групп. Основное свойство этого поля — длина, не превышающая 6 символов. Поле **КОЛ** содержит количество студентов в группе и является числовым (целым). Поле **ПБАЛЛ** содержит средний балл, полученный студентами группы при поступлении в учебное заведение, является числовым, но в отличие от поля **КОЛ** — действительным, содержащим десятичную часть.

Важным действием на этапе разработки структуры таблицы является задание ключевых полей. Для задания простого ключа, состоящего из одного поля таблицы, достаточно в режиме конструктора установить курсор в любую позицию этого поля и нажать кнопку

"Ключевое поле"  на панели инструментов. На рисунке это поле **НГ**. Для задания составного ключа, состоящего из нескольких полей таблицы, необходимо выделить эти поля, щелкая мышью по кнопкам слева от соответствующих строк при нажатой клавише Ctrl, а затем нажать кнопку **"Ключевое поле"**. Признаком установки является появление рисунка ключа на кнопках слева от соответствующей строки конструктора.

Имя таблицы запрашивается при закрытии окна конструктора. После закрытия конструктора в окне БД появляется значок и имя созданной таблицы. Если выделить значок таблицы и щелкнуть по кнопке **"Открыть"**, то таблица будет открыта для ввода записей.

Создание таблицы путем ввода данных

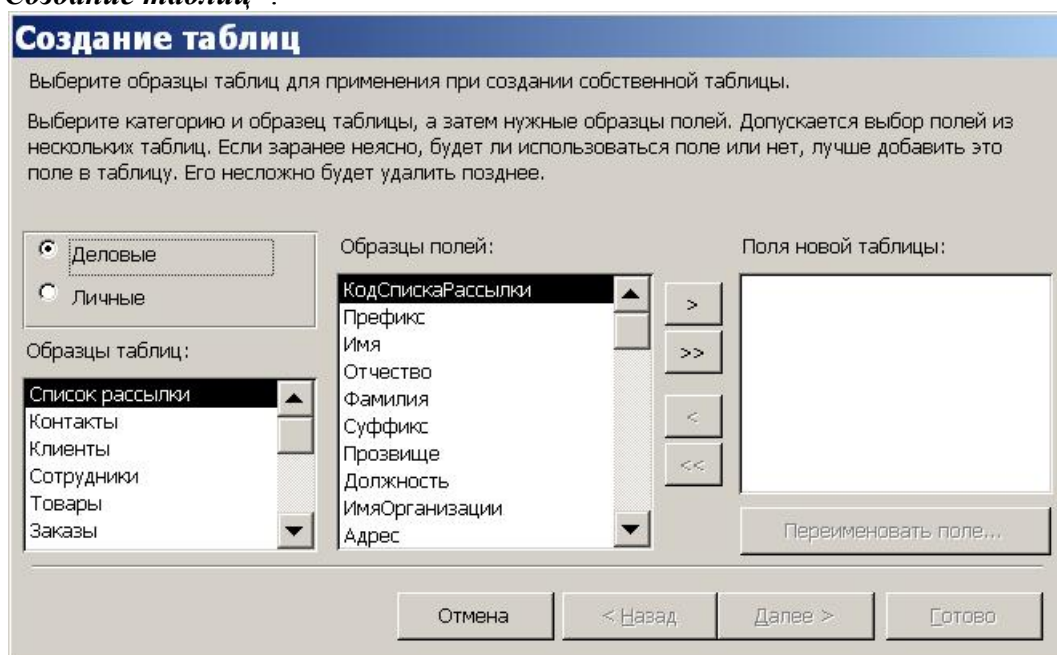
Создание таблицы путем ввода данных не предусматривает описания структуры таблицы. После выбора этого режима (режим таблицы) открывается пустая таблица, в которую можно ввести данные.



Любое поле в этой таблице можно изменить по желанию пользователя. Имена полей задаются непосредственно в заголовках. При сохранении этой таблицы Access проанализирует данные и автоматически присвоит соответствующий тип данных каждому полю, т.е. создаст структуру таблицы. При закрытии режима таблицы **Access** предложит создать ключевое поле. Если ответить "Да", то будет добавлено еще одно поле типа "Счетчик", которое и будет ключевым. Если ответить "Нет", то ключ можно задать позже, перейдя в режим конструктора.

Создание таблицы при помощи Мастера

Мастер таблиц автоматически создает таблицу по одному из шаблонов, предлагаемых в окне "Создание таблиц":



Мастер определяет ключ таблицы и создает связь новой таблицы с уже существующими. При этом ключ новой таблицы будет включен в таблицу, с которой устанавливается связь. По запросу пользователя мастер создает форму для ввода данных в таблицу.

Ввод данных в таблицу


Данные в таблицу можно ввести непосредственно в режиме ее открытия или через специально созданную форму. Второй способ предпочтителен, поскольку формы обеспечивают более удобный интерфейс и возможности контроля ввода.

Вводимые данные должны соответствовать типу данных и формату, определенным в структуре для каждого поля таблицы. При несоответствии **Access** выдает предупреждение и не разрешает продолжать ввод. Следует либо ввести данные требуемого формата, либо отменить ввод.

Определение связей между таблицами в базе данных Access

Схема данных

Схема данных является графическим образом БД. Она используется различными объектами Access для определения связей между несколькими таблицами. Например, при создании формы, содержащей данные из нескольких взаимосвязанных таблиц, схема данных обеспечивает автоматический согласованный доступ к полям этих таблиц. Она же обеспечивает целостность взаимосвязанных данных при корректировке таблиц.

После того, как созданы таблицы БД, можно установить связи между ними, обратившись к схеме данных. Запуск схемы данных выполняется через значок **"Схема данных"**  на панели инструментов Access.

Связь между таблицами устанавливает отношения между совпадающими значениями в ключевых полях, обычно между полями, имеющими одинаковые имена в обеих таблицах. В большинстве случаев с ключевым полем одной таблицы, являющимся уникальным идентификатором каждой записи, связывается **внешний ключ** другой таблицы.

Обязательным условием при установлении связи является совпадение связываемых полей по типу и формату.

Типы связей

Тип отношения в создаваемой **Microsoft Access** связи зависит от способа определения связываемых полей.

Отношение «один-ко-многим»

Отношение **«один-ко-многим»** является наиболее часто используемым типом связи между таблицами. В отношении **«один-ко-многим»** каждой записи в таблице А могут соответствовать несколько записей в таблице В, но запись в таблице В не может иметь более одной соответствующей ей записи в таблице А.

Отношение «один-к-одному»

При отношении **«один-к-одному»** запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот. Отношения этого типа используются не очень часто, поскольку большая часть сведений, связанных таким образом, может быть помещена в одну таблицу. Отношение **«один-к-одному»** может использоваться для разделения очень широких таблиц, для отделения части таблицы по соображениям защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

Отношение «многие-ко-многим»

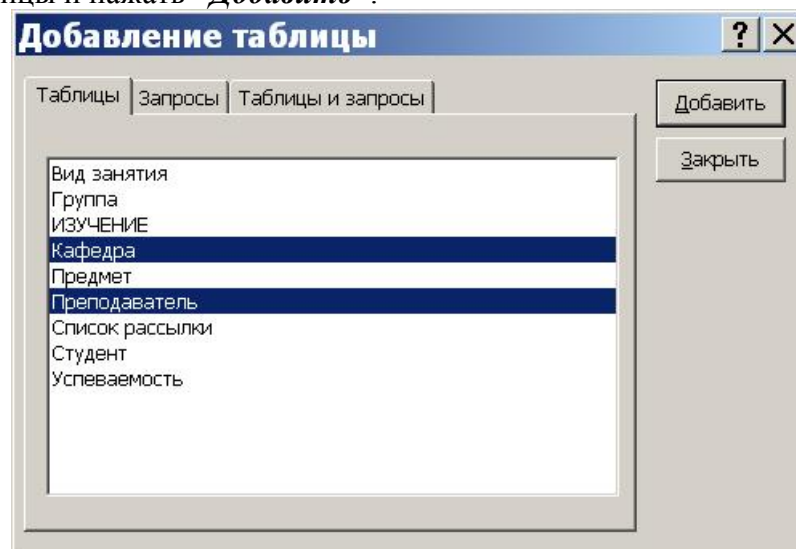
При отношении **«многие-ко-многим»** одной записи в таблице А могут соответствовать несколько записей в таблице В, а одной записи в таблице В несколько записей в таблице А. Этот тип связи возможен только с помощью третьей (связующей) таблицы, первичный ключ которой состоит из двух полей, которые являются внешними ключами таблиц А и В. Отношение **«многие-ко-многим»** по сути дела представляет собой два отношения **«один-ко-многим»** с третьей таблицей.

Определение связей между таблицами

Связь между таблицами определяется путем добавления связываемых таблиц в окно **«Схема данных»** с последующим перетаскиванием ключевого поля из одной таблицы в другую. Рассмотрим пример связывания таблиц.

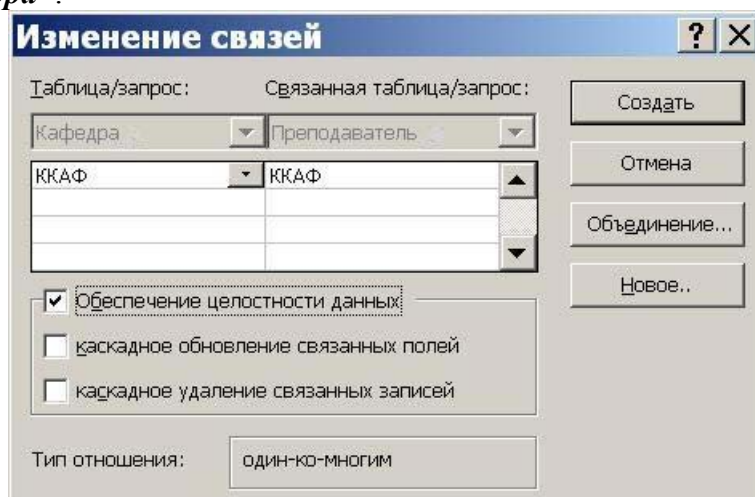
Предположим, что требуется установить связь между таблицами **"Кафедра"** и **"Преподаватель"** через поле **ККАФ** (код кафедры). В таблице **"Кафедра"** это поле является уникальным ключом, а в таблице **"Преподаватель"** — внешним ключом. Если схема данных создается заново, то при нажатии на кнопку **"Схема данных"** поверх окна

схемы данных появится окно **"Добавление таблицы"**. В этом окне следует выделить требуемые таблицы и нажать **"Добавить"**.



В результате в окно схемы данных будут добавлены графические образы двух таблиц:

Необходимо перетащить мышью поле **ККАФ** таблицы **"Кафедра"** на поле **ККАФ** таблицы **"Преподаватель"**. В открывшемся окне **"Изменение связей"** следует установить флажок **"Обеспечение целостности данных"**. В этом случае Access будет выдавать предупреждающие сообщения о неправильном вводе данных, если, например, в поле **ККАФ** подчиненной таблицы **"Преподаватель"** будет введено значение, отсутствующее в поле **ККАФ** базовой таблицы **"Кафедра"**.



Обратите внимание, что Access автоматически определил тип связи как "один-ко-многим".

Можно также установить флажки **"каскадное обновление связей"** и **"каскадное удаление связей"**. В этом случае Access автоматически скорректирует (удалит) записи в подчиненных таблицах, если будут изменены записи в базовой таблице.

После нажатия на кнопку **"Создать"**, образы таблиц будут соединены связями как показано на рисунке.

Ключевые в базовых таблицах выделяются жирным шрифтом.

Для установления



связей по составному ключу необходимо в окне **"Изменение связей"** в полях **"Таблица/Запрос"** и **"Связанная таблица/запрос"** вручную выбрать из списков пары связываемых полей. На рисунке показан пример связи по составному ключу.

Если перетащить поле, не являющееся ключевым и не имеющее уникального индекса, на другое поле, которое также не является ключевым и не имеет уникального индекса, создается **неопределенное отношение**. В **запросах**, содержащих таблицы с неопределенным отношением, **Microsoft Access** по умолчанию отображает линию объединения между таблицами, но условия целостности данных при этом не накладываются и нет гарантии уникальности записей в любой из таблиц.

Проектирование базы данных

Проектирование БД является очень важным этапом, от которого зависят последующие этапы разработки СУБД. Время, затраченное разработчиком на проектирование БД, обычно окупается высокой скоростью реализации проекта.

Перед созданием базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, иметь всю необходимую информацию для удовлетворения предполагаемых запросов пользователя и определить потребности в обработке данных.

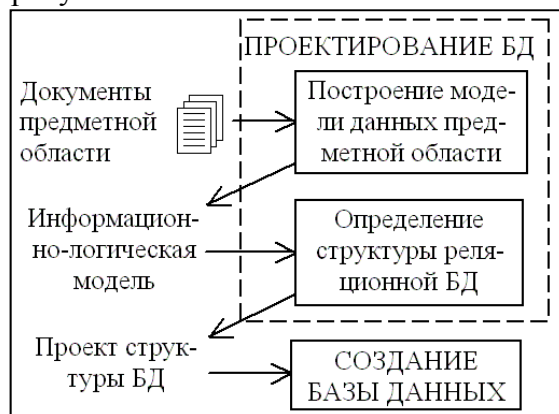
На основе такого описания на этапе проектирования базы данных осуществляется определение состава и структуры данных предметной области, которые должны находиться в базе данных и обеспечивать выполнение необходимых запросов и задач пользователя. Структура данных предметной области может отображаться информационно-логической моделью. На основе этой модели легко создается реляционная база данных.

Информационно-логическая модель отображает данные предметной области в виде совокупности информационных объектов и связей между ними. Эта модель представляет данные, подлежащие хранению в базе данных.

При разработке модели данных могут использоваться два подхода. В первом подходе сначала определяются основные задачи, для решения которых строится база, и выявляются потребности задач в данных. При втором подходе сразу устанавливаются типовые объекты предметной области. Наиболее рационально сочетание обоих подходов. Это связано с тем, что на начальном этапе, как правило, нет исчерпывающих сведений обо всех задачах. Использование такой технологии тем более оправдано, что гибкие средства создания реляционной базы данных в **Access** позволяют на любом этапе разработки внести изменения в базу данных и модифицировать ее структуру без ущерба для введенных ранее данных.

Основные этапы проектирования БД показаны на рисунке.

Информационный объект — это информационное описание некоторой сущности — реального объекта, процесса, явления или события. Информационный объект образуется совокупностью логически взаимосвязанных **реквизитов**, представляющих качественные и количественные характеристики некоторой сущности предметной области. Примерами информационных объектов могут быть — **ТОВАР, ПОСТАВЩИК, ЗАКАЗЧИК, ПОСТАВКА, ОТГРУЗКА, СОТРУДНИК, ОТДЕЛ, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ, КАФЕДРА** и т.п.



Информационные объекты выделяются на основе описания предметной области путем определения функциональных зависимостей между реквизитами. Совокупность реквизитов информационного объекта должна отвечать требованиям нормализации. Каждому информационному объекту нужно присвоить уникальное имя, например, СТУДЕНТ, ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ, КАФЕДРА.

Информационный объект имеет множество реализаций — экземпляров. Например, каждый экземпляр объекта СТУДЕНТ представляет конкретного студента. Экземпляр образуется совокупностью конкретных значений реквизитов и должен однозначно определяться (идентифицироваться) значением ключа информационного объекта, который состоит из одного или нескольких ключевых реквизитов. Таким образом, реквизиты подразделяются на **ключевые** и **описательные**. Последние являются функционально зависимыми от ключа.

Функциональная зависимость реквизитов имеет место в том случае, если одному значению ключа соответствует только одно значение описательного (зависимого) реквизита.

***Замечание.** При выявлении функциональных зависимостей реквизитов не рассматриваются арифметические зависимости (например, стоимость от количества), поскольку устанавливается только функциональная зависимость, определяющая связи описательных и ключевых реквизитов, и на основе которой выявляется реквизитный состав каждого информационного объекта.*

При графическом изображении модели данных каждый информационный объект представляется прямоугольником с обозначением его имени и идентификатора - ключа.

Реквизиты каждого информационного объекта должны отвечать **требованиям нормализации**:

- информационный объект должен содержать уникальный идентификатор (ключ). Ключ является **простым**, если он состоит из одного реквизита или **составным**, если из нескольких;
- все описательные реквизиты должны быть взаимонезависимы, т.е. между ними не может быть функциональных зависимостей;
- все реквизиты, входящие в составной ключ, должны быть также взаимонезависимы;
- каждый описательный реквизит должен функционально полно зависеть от ключа, т.е. каждому значению ключа соответствует только одно значение описательного реквизита;
- при составном ключе описательные реквизиты должны зависеть целиком от всей совокупности реквизитов, образующих ключ;
- каждый описательный реквизит не может зависеть от ключа транзитивно, т.е. через другой промежуточный реквизит.

***Замечание.** В случае транзитивной зависимости между реквизитами можно выполнить расщепление совокупности реквизитов с образованием двух информационных объектов вместо одного.*

Выполнение требований нормализации обеспечивает построение реляционной базы данных без дублирования данных и возможность поддержки целостности при внесении изменений.

Выделение информационных объектов предметной области

Процесс выделения информационных объектов предметной области, отвечающих требованиям нормализации, может производиться на основе интуитивного или формального подхода. Теоретические основы формального подхода были разработаны и полно изложены в монографиях по организации баз данных известного американского ученого Дж. Мартина. При интуитивном подходе легко могут быть выявлены информационные объекты, соответствующие реальным объектам. Однако, получаемая при этом информационно-логическая модель, как правило, требует дальнейших преобразований, в частности, преобразования много-многозначных (M:N) связей между

объектами. При таком подходе возможны существенные ошибки, если отсутствует достаточный опыт. Последующая проверка выполнения требований нормализации обычно приводит к необходимости уточнения информационных объектов.

Рассмотрим формальные правила, которые могут быть использованы для выделения информационных объектов, отвечающих требованиям нормализации:

- На основе описания предметной области выявить документы и их реквизиты, подлежащие хранению в базе данных.
- Определить функциональные зависимости между реквизитами.

Замечание. Функциональную зависимость реквизитов можно изобразить графически в виде линий со стрелками, идущих от ключевого реквизита к описательному (зависимому). Эти зависимости целесообразно отразить непосредственно в таблице, где представлен состав реквизитов, сгруппированных по документам.

- Выбрать все зависимые реквизиты и указать для них ключевые реквизиты (один или несколько).

Замечание. В случае транзитивной зависимости некоторые реквизиты являются одновременно зависимыми и ключевыми и соответственно представлены в группе зависимых и ключевых.

- Сгруппировать реквизиты, зависимые от одних и тех же ключевых реквизитов. Полученные группы зависимых реквизитов вместе с ключевыми реквизитами образуют информационные объекты.

После выделения информационных объектов надо дать окончательное их описание. В таком описании может быть представлена также семантика информационных объектов, то есть их смысловое определение. Затем можно осуществить контрольную проверку выполнения требований нормализации. При использовании приведенных правил нет необходимости отдельно преобразовывать транзитивные зависимости реквизитов. Совокупность получаемых при этом информационных объектов позволяет получить информационно-логическую модель, не требующую дальнейших преобразований для построения реляционной базы данных. Как правило, сразу оказываются выделенными объекты, выполняющие роль связки между объектами, находящимися в много-многозначных отношениях.

Пример проектирования БД "Учебный процесс"

Пусть требуется построить БД, содержащую информацию об учебном процессе текущего семестра. Необходимые данные хранятся в следующих документах:

- списки групп студентов;
- списки преподавателей кафедр;
- перечень изучаемых предметов;
- учебные программы;
- распределение нагрузки между преподавателями;
- экзаменационные ведомости.

В первую очередь, в имеющихся документах необходимо выявить реквизиты, подлежащие хранению в БД, определить функциональную зависимость между ними, выделить ключевые и описательные реквизиты и сгруппировать реквизиты, зависимые от выделенных ключевых реквизитов.

Вторым этапом является описание полученных информационных объектов. Удобной формой описания структуры информационных объектов являются таблицы. Для рассматриваемой задачи получено семь таблиц:

Таблица "Кафедра"

Содержание поля	Имя поля	Тип поля	Наличие ключа
Код кафедры	ККАФ	Счетчик	Простой ключ

Название кафедры	НКАФ	Текстовый	
Телефон кафедры	ТЕЛ	Текстовый	
Заведующий кафедрой	ЗАВ	Текстовый	
Фотография заведующего	ФОТО	OLE	

Таблица "Группа"

Содержание поля	Имя поля	Тип поля	Наличие ключа
Номер группы	НГ	Текстовый	Простой ключ
Количество студентов	КОЛ	Числовой	
балл успеваемости	СБАЛЛ	Числовой	

Таблица "Предмет"

Содержание поля	Имя поля	Тип поля	Наличие ключа
Код предмета	КП	Счетчик	Простой ключ
Название предмета	НП	Текстовый	
Всего учебных часов	ЧАСЫ	Числовой	
Часов лекций	ЛЕК	Числовой	
Часов практических занятий	ПР	Числовой	
Число семестров	ЧС	Числовой	
Программа курса	ПРОГ	МЕМО	

Таблица "Преподаватель"

Содержание поля	Имя поля	Тип поля	Наличие ключа
Табельный номер	ТАБН	Счетчик	Простой ключ

Фамилия, имя, отчество	ФИО	Числовой	
Ученая степень	СТ	Текстовый	
Ученое звание	ЗВ	Текстовый	
Код кафедры	ККАФ	Числовой	

Таблица "Студент"

Содержание поля	Имя поля	Тип поля	Наличие ключа
Номер группы	НГ	Текстовый	Составной ключ
Номер студента в группе	НС	Числовой	Составной ключ
Фамилия, имя, отчество	ФИО	Текстовый	
Год рождения	ГОДР	Дата	
Адрес	АДР	Текстовый	
Средний балл обучения	СБАЛЛ	Числовой	

Таблица "Изучение"

Содержание поля	Имя поля	Тип поля	Наличие ключа
Номер группы	НГ	Текстовый	Составной ключ
Код предмета	КП	Числовой	Составной ключ
Табельный номер преподавателя	ТАБН	Числовой	Составной ключ
Вид занятия	ВИДЗ	Текстовый	Составной ключ
Часов по данному виду	ЧАСЫ	Числовой	

Таблица "Успеваемость"

Содержание поля	Имя поля	Тип поля	Наличие ключа
-----------------	----------	----------	---------------

Номер студента	НС	Числовой	Составной ключ
Номер группы	НГ	Текстовый	Составной ключ
Код предмета	КП	Числовой	Составной ключ
Табельный номер преподавателя	ТАБН	Числовой	Составной ключ
Вид занятия	ВИДЗ	Текстовый	Составной ключ
Оценка	ОЦЕНКА	Числовой	

В рассмотренных таблицах добавлен столбец "Тип поля", являющийся характеристикой не информационного объекта, а таблицы БД. Он добавлен для иллюстрации особенностей реализации БД:

- связываемые поля должны быть одного типа;
- для ключевых полей в **Access** имеется специальный тип **счетчик**. Этот тип предусматривает автоматическое заполнение поля порядковыми номерами записей и является числовым типом в формате длинного целого. Поэтому внешние ключи этих полей тоже должны иметь формат длинного целого.

Реквизит НГ реализован как текстовое с максимальной длиной 6 символов, поскольку номер группы может содержать буквы и его можно использовать в качестве ключа.

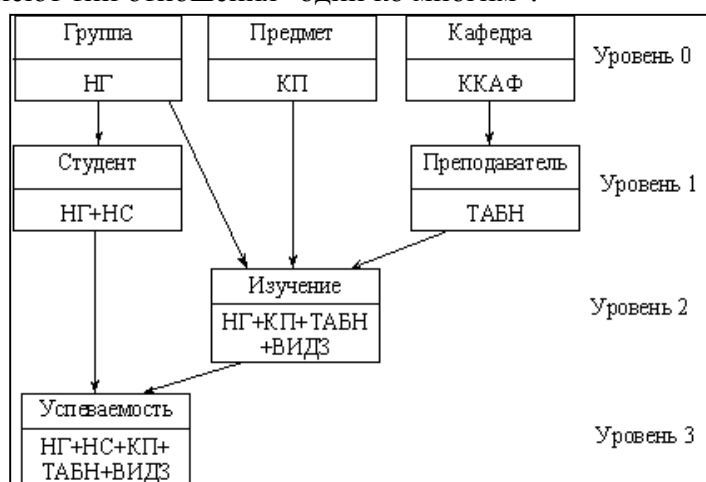
Для реквизита ФОТО в таблице "*Кафедра*" используется "**Поле объекта OLE**" для обеспечения возможности выводить фотографию.

Реквизиту ПРОГ таблицы "Предмет" соответствует тип поля МЕМО для вывода сравнительно большого текста, такого, как программа обучения по предмету.

Следующим этапом проектирования БД является определение связей между информационными объектами. Связи устанавливаются последовательно между парами объектов. В данной задаче все связи имеют тип отношения "один ко многим".

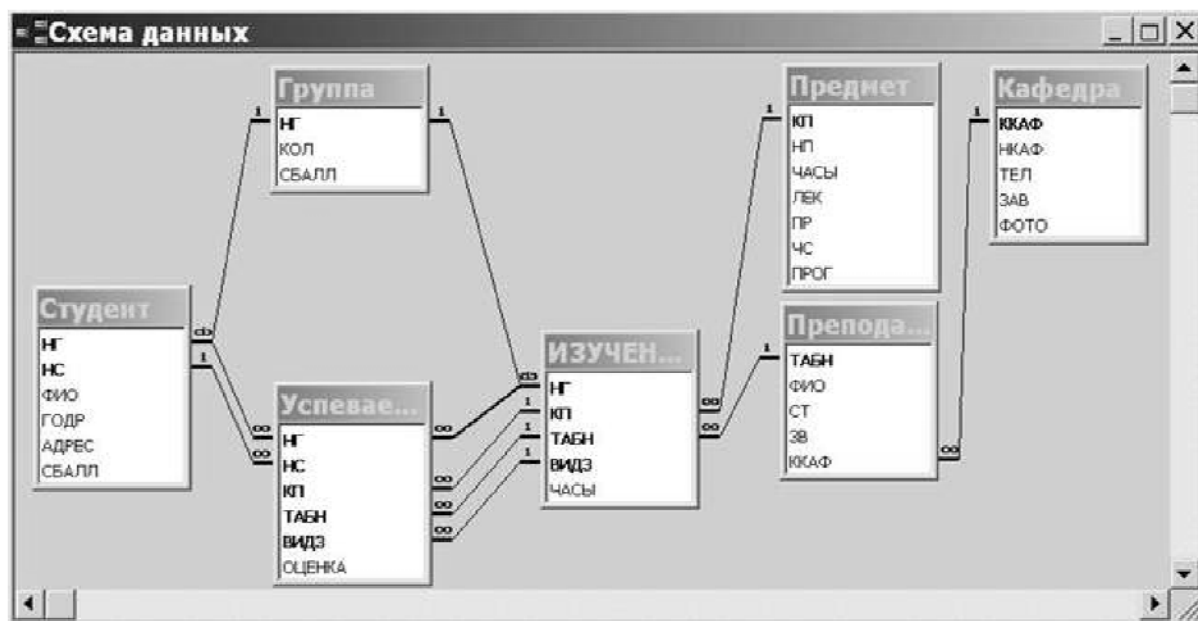
Информационно-логическая модель БД "Учебный процесс", построенная в соответствии с выявленными информационными объектами и связями, показана на рисунке:

Информационно-логическая модель приведена в каноническом виде, т.к. объекты размещены по уровням. На нулевом уровне размещаются объекты, не подчиненные никаким другим объектам. Уровень остальных объектов определяется наиболее длинным путем к объекту от нулевого уровня. Такое



размещение объектов дает представление об их иерархической подчиненности, делает модель более наглядной и облегчает понимание связей между объектами.

Используя информационно-логическую модель, на этапе реализации связей между таблицами получим следующую схему данных.



Разработка форм средствами Access

Access предоставляет широкие возможности по конструированию графического интерфейса пользователя для работы с БД. Формы являются важнейшим инструментом, позволяющим осуществить первоначальную загрузку записей в таблицы, выполнить их просмотр и редактирование. При этом работа пользователя с БД выполняется в привычном для него виде — в виде документа.

При наличии схемы данных формы помогают выполнить корректный ввод данных в систему взаимосвязанных таблиц. При этом реализуется важнейший аспект технологии работы с БД — однократный ввод данных.

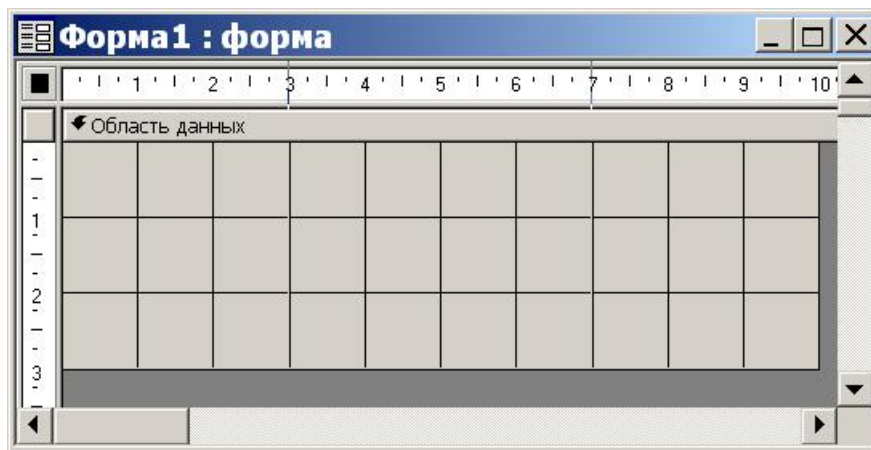
Для конструирования форм необходимо предварительно выполнить определенную последовательность действий по разработке СУБД:

- сконструировать таблицы БД;
- определить связи между таблицами и создать схему данных;
- определить эскиз экранной формы и состав размещаемых на ней объектов.


Конструирование форм обычно выполняют в режиме **Мастера** с последующей доработкой вручную в режиме конструктора. Мастер позволяет быстро разработать заготовку формы с необходимыми полями и связями, однако, он создает только типовые конструкции, вид которых может не устраивать пользователя. Переход в режим конструктора позволяет устранить недостатки оформления.

Разработка формы в режиме конструктора

После выбора режима конструктора в окне **"Новая форма"** требуется задать таблицу — источник данных для формы. После этого открывается окно конструктора форм, показанное на рисунке.



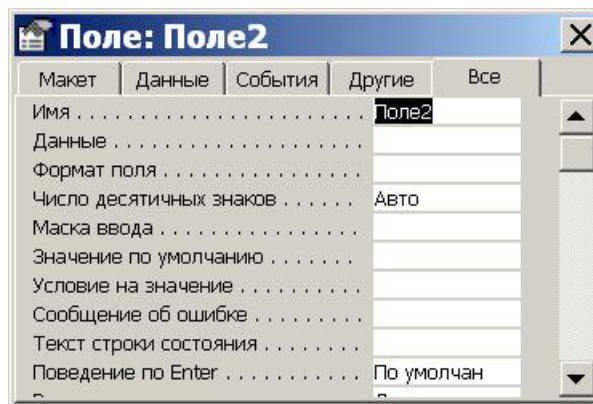
Область выделения формы служит для вызова контекстно-зависимого меню, через которое можно получить доступ к свойствам формы.

Для конструирования формы необходимо вызвать панель элементов нажатием кнопки  ("Панель элементов") на панели форматирования окна Access. Панель элементов имеет вид, показанный на рисунке.



Конструирование формы заключается в последовательном выборе объектов на панели элементов и установке их в рабочей области формы. Для каждого устанавливаемого элемента должны быть определены его свойства. Для вызова окна свойств элемента необходимо выделить его, вызвать правой кнопкой мыши контекстное меню и выбрать пункт "Свойства". Пример окна свойств показан на рисунке.

Если на панели элементов включена кнопка "Мастер элементов", то при установке некоторых из них (например, кнопки) запускается **Мастер**, который поможет определить главные свойства.



Пример создания однотабличной формы

Рассмотрим технологию создания однотабличной формы в режиме конструктора на примере формы для редактирования таблицы "Предмет" базы данных "Учебный процесс".

1. В окне БД выберем пункт "Формы", режим конструктора.
2. В окне "Новая форма" выберем источник данных — таблицу "Предмет".
3. Установим панель элементов нажатием соответствующей кнопки.
4. Установим на форме в один столбец 7 элементов "Поле". Заметьте, что элементы устанавливаются вместе с надписями, которые имеют формальный характер: "Поле0", "Поле2", ...
5. Для каждого из установленных элементов откройте окно свойств и задайте:
 - на вкладке "Данные" в поле "Данные" — поле — источник данных;
 - на вкладке "Другие" неформальное имя — идентификатор поля.

6. Замените формальные тексты в надписях на содержательные. Это можно сделать прямо



на форме щелчком мыши внутри области надписи.

7. Измените размеры полей, потянув мышью за маркеры. Большие маркеры в левых верхних углах элементов служат для перемещения только этих элементов. Если же "потянуть" мышью за рамку объекта, то надпись и поле будут перемещаться вместе. Поле ПРОГР сделайте большим, поскольку оно предназначено для вывода многострочной текстовой информации.

В результате после открытия форма должна иметь вид, показанный на рисунке.

Создание однотабличной формы в режиме Мастера

Рассмотрим технологию создания однотабличной формы в режиме мастера на том же примере.

1. В окне БД выберем пункт **"Формы"**, режим мастера.
2. В окне **"Новая форма"** выберем источник данных — таблицу **"Предмет"**.
3. На первом шаге Мастера переведем все поля из области **"Доступные"** в область **"Выбранные"**.
4. На втором шаге **Мастера** выберем внешний вид формы — в один столбец.
5. На третьем шаге выберем стиль формы — стандартный.
6. На четвертом шаге зададим имя формы — **"Предмет"** и нажмем **"Готово"**.
7. Для полученной заготовки формы перейдем в режим конструктора и заменим надписи на содержательные. В результате получим форму, аналогичную приведенной на рисунке выше. Следует заметить, что объем **"ручной"** работы при использовании Мастера существенно сократился.

Создание многотабличной формы в режиме мастера

Многотабличные формы имеют большое количество полей, надписей и других объектов. Поэтому разработка формы в режиме мастера особенно выгодна. Рассмотрим создание многотабличной формы, выводящей информацию об успеваемости в группе.

1. В окне БД выберем пункт **"Формы"**, режим мастера.
2. В окне **"Новая форма"** выберем источник данных — таблицу **"Изучение"**.
3. На первом шаге **Мастера** переведем все поля таблицы **"Изучение"** из области **"Доступные"** в область **"Выбранные"**. На этом же шаге добавим в область **"Выбранные"** следующие поля:
 - ИС, ОЦЕНКА из таблицы **"Успеваемость"**;
 - ФИО из таблицы **"Студент"**;
 - ФИО, СТ, ЗВ из таблицы **"Преподаватель"**;
 - НП, ЧАСЫ из таблицы **"Предмет"**.

4. На втором шаге Мастера выберем вид представления данных как показано на рисунке. Область справа, обведенная рамкой, будет оформлена как подчиненная форма, т.е. зависящая от главной и выводящая записи в зависимости от информации на главной форме.

Создание форм

Выберите вид представления данных:

- Предмет
- ИЗУЧЕНИЕ
- Преподаватель
- Успеваемость
- Студент

НГ, КП, ТАБН, ВИДЗ, ИЗУЧЕНИЕ_ЧАСЫ, Преподаватель_ФИО, СТ, ЗВ, НП, Предмет_ЧАСЫ

НС, ОЦЕНКА, Студент_ФИО

☒ Подчиненные формы ☐ Связанные формы

Отмена < Назад Далее > Готово

5. На третьем шаге выберем внешний вид формы — ленточный.
6. На четвертом шаге зададим стиль формы — "Стандартный".
7. На пятом шаге зададим имя формы и нажмем "Готово". В результате получим форму, показанную на рисунке.

ИЗУЧЕНИЕ

НГ M102 СТ д.т.н.

КП 1 ЗВ профессор

ТАБН 1 НП Информатика

ВИДЗ Лаб. раб. Предмет_ЧАСЫ 200

ИЗУЧЕНИЕ_ЧАСЫ 34

ФИО Андреев А.П.

Успеваемость

НС	ОЦЕНКА	ФИО
1	3	Дедикова Т.О.
2	3	Дронова И.Н.
3	4	Клемешов А.В.

Запись: 1 из 3

Запись: 2 из 24

8. Откроем форму в режиме конструктора и преобразуем к виду, показанному на рисунке.

ИЗУЧЕНИЕ

УСПЕВАЕМОСТЬ ГРУППЫ M102

Предмет: Информатика Преподаватель: Андреев А.П.

Вид занятия: Лаб. раб. ФИО: Андреев А.П.

Всего часов: 200 Ученая степень: д.т.н.

Часы занятия: 34 Должность: профессор

НС	Фамилия студента	ОЦЕНКА
1	Дедикова Т.О.	3
2	Дронова И.Н.	3
3	Клемешов А.В.	4

Запись: 1 из 3

Запись: 2 из 24

На форме, полученной после работы Мастера, сделаны следующие преобразования:

- поле с надписью НГ перенесено в область заголовка формы и надпись преобразована в "УСПЕВАЕМОСТЬ ГРУППЫ";
- поля с названием предмета, общее количество часов и реквизиты преподавателя сгруппированы и выделены инструментом "Прямоугольник";
- все поля со списком, унаследованные от таблиц-источников, преобразованы в простые поля. Для этого надо щелкнуть правой кнопкой мыши по полю со списком и в контекстном меню выбрать пункт "Преобразовать элемент в", затем выбрать "Поле";
- для других полей и подчиненной формы изменены размеры, надписи и положение.

Полученная многотабличная форма является составной, содержащей главную и подчиненную формы. Она позволяет просматривать информацию об успеваемости групп, изменяя значение счетчика записей на главной форме. При этом информация во всех полях составной формы будет изменяться согласованно.

Следует заметить, что многотабличная форма может не содержать подчиненных форм. С другой стороны, на главной форме может быть несколько подчиненных.

Разработка отчетов

Средства Access по разработке отчетов предназначены для создания макета отчета, по которому может быть осуществлен вывод данных из таблиц в виде выходного печатного документа. Эти средства позволяют конструировать отчет сложной структуры, обеспечивающий вывод взаимосвязанных данных из многих таблиц. При этом могут быть выполнены самые высокие требования к оформлению документа.

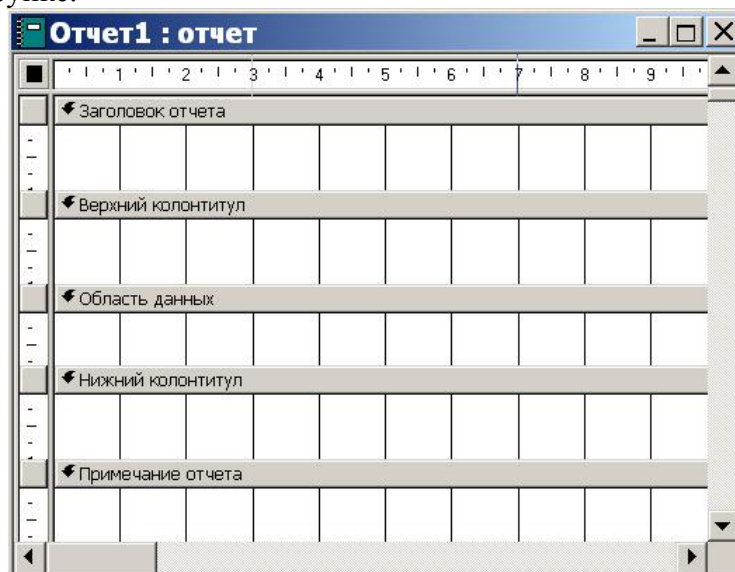
Перед началом конструирования отчета пользователь должен произвести подготовительную работу, в результате которой определяется требуемый **макет отчета**.

В процессе конструирования формируется состав и содержание разделов отчета, а также размещение в нем значений, выводимых из полей таблиц базы данных. Кроме того, оформляются заголовки, подписи реквизитов отчета, размещаются вычисляемые реквизиты.

Средства конструирования отчета позволяют группировать данные по нескольким уровням. Для каждого уровня могут производиться вычисления итогов, определяться заголовки и примечания по каждой группировке. При формировании отчета могут производиться разнообразные вычисления.

При необходимости вывода в отчете данных из многих таблиц в качестве основы для отчета может быть использован многотабличный запрос. На запрос могут быть возложены наиболее сложные виды выборки и предварительной обработки данных.

Отчет может создаваться с помощью мастера или в режиме конструктора отчетов. Достаточно часто сначала используется мастер отчетов, который позволяет быстро создать заготовку отчета, а затем заготовка дорабатывается в режиме конструктора. При создании отчета в режиме конструктора в окне отображаются пустые разделы отчета, показанные на рисунке.



Наличие этих разделов, а также их добавление или удаление определяется командами меню «**Вид/Колонтитулы**» и «**Вид/Заголовок/Примечание отчета**». Для этих же целей можно использовать соответствующие кнопки панели инструментов конструктора отчетов.

При разработке отчета разделы окна нужно заполнить элементами в соответствии с необходимым пользователю макетом отчета. В заголовок помещается текст из шапки макета отчета. В колонтитул обычно помещают заголовки, даты и номера страниц. При определении содержания этих разделов следует исходить из требований к оформлению отдельных страниц отчета. В области данных размещаются поля таблиц базы данных.

Для каждого элемента, а также раздела и отчета в целом могут быть уточнены свойства. Технология размещения элементов и определения их свойств практически такая же, как и в конструкторе форм.

Разработка отчета на основе запроса

Запрос является мощным и удобным средством выборки взаимосвязанных данных. Поэтому с помощью запроса можно подготовить данные для сложного отчета.

Рассмотрим технологию создания сложного отчета с использованием запроса на примере формирования бланка "Экзаменационная ведомость" группы по заданному предмету. Макет отчета показан на рисунке.

**ГОСУДАРСТВЕННЫЙ КОМИТЕТ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПО ВЫСШЕМУ ОБРАЗОВАНИЮ**

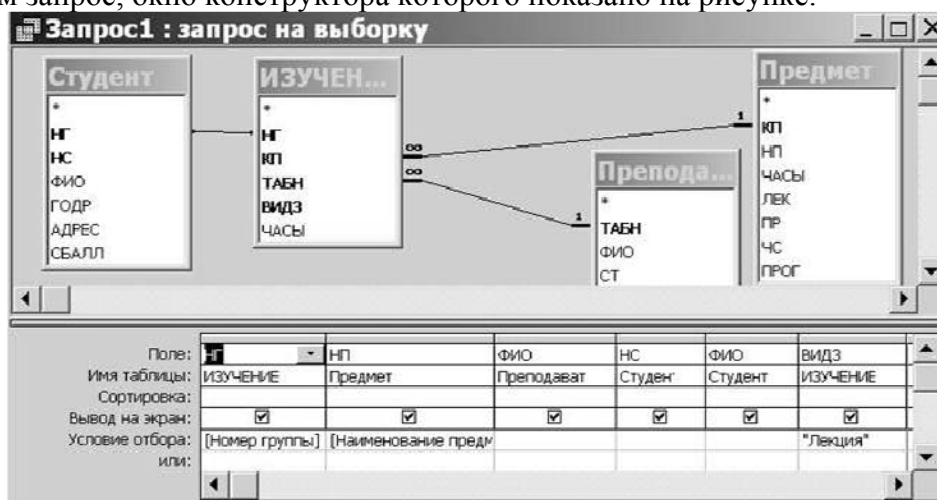
**Белгородский государственный университет
Экзаменационная ведомость**

Название предмета _____ Группа _____
Преподаватель _____
Дата _____

№ п/п	Фамилия И.О.	Оценка	Подпись преподавателя

Итого:
Отлично _____
Хорошо _____
Удовлетворительно _____
Неудовлетворительно _____
Не являлось _____

Организуем запрос, окно конструктора которого показано на рисунке.



При создании запроса связи между таблицами установятся автоматически. Связи таблиц ИЗУЧЕНИЕ, ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ определяются в соответствии со схемой БД. Между таблицами СТУДЕНТ и ИЗУЧЕНИЕ автоматически установится связь по одноименному полю ИГ. Эта связь является связью-объединением, которой нет в схеме данных базы. Заметим, что эти таблицы находятся в отношениях многие-ко-многим, т. к. один студент изучает много предметов и один предмет изучается многими студентами. Связь, установленная между таблицами СТУДЕНТ и ИЗУЧЕНИЕ, определяет операцию **симметричного объединения**. При этом записи из этих таблиц объединяются и добавляются в результат только в том случае, если связанные поля содержат одинаковые значения.

В запрос введено поле ВИДЗ для отбора только тех записей, которые связаны с видом занятия "Лекция", поскольку преподавателем, принимающим экзамен, обычно является лектор. Кроме того, в строку "Условие отбора" полей ИГ и ИП соответственно введены записи [Номер группы] и [Название предмета]. Это приведет к открытию окон для ввода параметров при открытии отчета.

- Запустим создание отчета в режиме мастера. В качестве источника данных выберем созданный запрос.
- На первом шаге мастера переведем из доступных в выбранные все поля запроса, кроме ВИДЗ.
- На втором шаге определяется группировка данных в отчете. В данном случае, способ группировки не очень важен, поскольку эта часть "доработки" заготовки макета будет выполнена вручную в режиме конструктора. Оставим группировку, как показано на рисунке.

Создание отчетов

Добавить уровни группировки?

НГ
 Преподаватель.ФИО
 НС
 Студент.ФИО

>
 <
 ↕
 Уровень
 ↕

НГ
 НГ, Преподаватель.ФИО, НС,
 Студент.ФИО

Группировка...
Отмена
< Назад
Далее >
Готово

- На третьем шаге выберем способ сортировки записей — по возрастанию номеров студентов НС.
- На четвертом шаге выберем вид макета отчета — **"структура 1"**.
- На пятом шаге выберем стиль отчета — **"строгий"**.
- На шестом шаге зададим имя отчета — **"Экзаменационная ведомость"** и нажмем **"Готово"**.
- Для редактирования откроем заготовку отчета в режиме конструктора. Рабочая область окна показана на рисунке.

Заголовок отчета									
Экзаменационная ведомость									
Верхний колонтитул									
Заголовок группы 'НГ'									
НГ				НГ					
		НС НГ		Преподаватель.ФИО		Студент.ФИО			
Область данных									
		НС		НГ		Преподаватель.ФИО		Студент.ФИО	
Нижний колонтитул									
=Now()					="Страница " & [Page] & " из " & [Pages]				
Примечание отчета									

- Преобразуем заготовку к следующему виду:

Заголовок отчета			
ГОСУДАРСТВЕННЫЙ КОМИТЕТ РОССИЙСКОЙ ФЕДЕРАЦИИ ПО ВЫСШЕМУ ОБРАЗОВАНИЮ			
БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ			
Экзаменационная ведомость			
Верхний колонтитул			
Название предмета	ИП	Группа	ИГ
Преподаватель	Преподаватель. ФИО		
Дата	=Now()		
Заголовок группы 'ИГ'			
№ п/п	Фамилия И.О.	Оценка	Подпись преподавателя
Область данных			
ИС	Студент. ФИО		
Нижний колонтитул			
Итого:			
Отлично _____			
Хорошо _____			
Удовлетворительно _____			
Неудовлетворительно _____			

На рисунке специально убрана сетка, чтобы состав элементов был заметнее. Большая часть надписей и полей заготовки перемещена на требуемые позиции, изменен размер этих элементов и размер шрифта. Недостающие элементы (линии и надписи) установлены при помощи панели элементов.

Пример экзаменационной ведомости для группы М102 и предмета "Информатика" показан на рисунке.

ГОСУДАРСТВЕННЫЙ КОМИТЕТ РОССИЙСКОЙ ФЕДЕРАЦИИ ПО ВЫСШЕМУ ОБРАЗОВАНИЮ			
БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ			
Экзаменационная ведомость			
Название предмета	Информатика	Группа	М102
Преподаватель	Андреев А.П.		
Дата	24 марта 2003 г.		

№ п/п	Фамилия И.О.	Оценка	Подпись преподавателя
1	Дедикова Т.О.		
2	Дронова И.Н.		
3	Клемшов А.В.		
4	Кононенко Р.М.		
5	Привалова Е.Л.		

Итого:

Отлично _____

Хорошо _____

Удовлетворительно _____

Неудовлетворительно _____

Контрольные вопросы

- 1). Что называется базой данных (БД)? Как представляются БД?
- 2). Что такое система управления базой данных (СУБД)?
- 3). Что представляет собой СУБД Access? В чем его отличие от табличного процессора Excel?
- 4). Какие БД называются реляционными?
- 5). Как СУБД Access хранит создаваемые таблицы и средства для обработки данных?
- 6). Каковы основные элементы БД?
- 7). В чем заключается первый этап создания таблицы БД? Второй этап?
- 8). Что определяют поля таблицы?
- 9). Каковы основные свойства полей?
- 10). Перечислите типы таблиц и ключей в реляционных БД?
- 11). Чем характеризуется базовая таблица?
- 12). Какие таблицы называются промежуточными?
- 13). Что такое первичный ключ?
- 14). В каких случаях используется составной ключ?
- 15). Что такое внешний ключ?
- 16). Каковы основные объекты окна базы данных, которое появляется после загрузки Access? Каково их назначение?
- 17). Как можно создавать таблицы, формы и отчеты?
- 18). Какие этапы включает в себя технология разработки СУБД?
- 19). Для чего предназначены таблицы БД?
- 20). Из скольких этапов состоит создание таблицы?
- 21). Как создать новую таблицу, работая в Access?
- 22). Какие поля следует заполнить при создании новой таблицы в режиме конструктора?
- 23). Как формируется имя поля? В чем его отличие от заголовка поля в выходном документе?
- 24). Каково назначение полей "Тип данных" и "Описание"?
- 25). Какие свойства поля можно задать? Каково их назначение?
- 26). Для чего служит вкладка "Подстановка"?
- 27). Как задаются простые ключи? Составные?
- 28). В чем особенность создания таблицы путем ввода данных?
- 29). Как создать таблицу при помощи Мастера?
- 30). Какие способы позволяют ввести данные в таблицу?
- 31). Что представляет собой схема данных БД?
- 32). Каково назначение схемы данных?
- 33). Как открыть схему данных в Access?
- 34). Как установить связь между таблицами?
- 35). Между какими полями таблиц устанавливается связь?
- 36). Каково обязательное условие при установлении связи?
- 37). Перечислите типы связей между таблицами. Охарактеризуйте их.
- 38). Как определить связи между таблицами?
- 39). От чего зависит выбор отношения в создаваемой Microsoft Access связи?
- 40). В каком случае создается отношение "один-ко-многим"? "Один-к-одному"? "Многие-ко-многим"?
- 41). В каком случае создается неопределенное отношение?
- 42). К каким последствиям приводит создание неопределенных отношений?
- 43). В каких формах может быть представлено описание предметной области на основе которого требуется разработать базу данных?
- 44). Почему нельзя напрямую использовать документы, описывающие предметную область и имеющие табличную форму, в качестве таблиц проектируемой базы данных?
- 45). Чем отличается информационный объект от документа, описывающего предметную область?
- 46). Чем отличается информационный объект от таблицы базы данных?

- 47). Что такое "реквизит информационного объекта"? Приведите примеры реквизитов для объектов, рассмотренных в примере.
- 48). Какие отношения возможны между связанными информационными объектами?
- 49). Что такое транзитивная зависимость реквизита? Приведите пример.
- 50). Что такое каноническая форма информационно-логической модели. Как эта форма отражена на схеме данных рассмотренного примера?
- 51). Приведите примеры выполнения требований нормализации в проекте базы данных "Учебный процесс".
- 52). Приведите примеры объектов, в которых не выполняются требования нормализации.
- 53). Каково основное назначение форм при разработке СУБД?
- 54). Перечислите способы создания форм. Какой способ является оптимальным с точки зрения трудоемкости?
- 55). Почему разработчики СУБД стремятся обеспечить ввод и редактирование данных через формы, а не напрямую в таблицах?
- 56). Какие типы (внешний вид) формы предлагает создать **Мастер форм** (проанализируйте на компьютере)?
- 57). Каково основное назначение отчетов?
- 58). Сравните назначение отчетов и назначение форм.
- 59). Какова оптимальная технология разработки отчетов с точки зрения трудоемкости?
 - 60). Как при работе с **Мастером отчетов** изменить группировку данных в отчете?

[1] Широков Л.А. «Информационные технологии в экономике». Учебное пособие.