**Конспект лекций по теме: «Базы данных»**

**Понятие базы данных**

**База данных (БД)** – это систематизированное хранилище информации, например, телефонный справочник, сведения о студентах и т.д.

До появления компьютеров эта информация хранилась в папках или картотеках. Это доставляло много неудобств, так как простая невнимательность сотрудника, вносившего изменения в такую базу данных, могла привести к появлению ложной информации. Весьма затруднителен был и поиск нужной информации. Нередко для получения справки приходилось перебирать сотни личных карточек.

Использование компьютеров позволило устранить многие проблемы, характерные для некомпьютерных БД. При правильном проектировании компьютерной БД добавление в нее новой информации и модификация уже существующих данных перестает быть трудной задачей.

**Функции БД:**

* хранение больших массивов информации;
* исключение или сведение к минимуму дублирования данных;
* установление и поддержка связей между данными;
* защита целостности данных (под защитой целостности данных здесь понимается набор средств, предотвращающих появление в БД некорректных данных) ;
* быстрый доступ к нужной информации;
* обеспечение секретности;
* простота внесения изменений;
* возможность одновременного доступа к информации для нескольких пользователей.

Чтобы обеспечить выполнение этих требований, данные в БД должны иметь определенную структуру. В зависимости от того, какой способ описания структуры данных используется при создании БД, различают сетевые, иерархические, реляционные и объектно-ориентированные БД . Наибольшее распространение получили *реляционные* БД.

**Реляционные БД**

Информация в таких БД хранится в виде двумерных таблиц. В каждой таблице содержатся сведения о наборе объектов определенного типа (людях, товарах и т.д.).

**Рисунок 1**. Пример таблицы *Access*

Строки таблицы называются *записями*. Запись – это компьютерный аналог той информации, которая обычно хранилась на карточке или бланке.

Столбцы таблицы называются *полями*. Поле – это компьютерный аналог графы карточки или бланка. В нем содержится информация о каком-либо свойстве описываемых объектов. Все записи состоят из одинакового набора полей.

Значения в каждом поле относятся к одному типу данных: числа, строки символов, даты. Пересечение отдельной записи и отдельного поля называется *ячейкой*, а сами данные в отдельной ячейке называются *значением в поле* или *элементом таблицы*.

В простейшем случае БД состоит из одной таблицы, но обычно она включает несколько взаимосвязанных таблиц. Связь (relation) между таблицами осуществляется через общие поля. Установление связи между таблицами в реляционной БД позволяет извлекать и объединять информацию сразу из нескольких таблиц.

**СУБД *Access***

Для взаимодействия пользователя с БД используются системы управления базами данных (СУБД). Одной из наиболее распространенных СУБД для персонального компьютера является СУБД *Access*, входящая в состав пакета *Microsoft Office*.

*Access* позволяет пользователю решать следующие задачи:

* создавать БД и вводить в нее данные;
* просматривать и редактировать содержимое таблиц;
* устанавливать связи между таблицами;
* обеспечить защиту целостности и секретность данных;
* выполнять различные запросы к данным;
* представлять информацию в виде форм и отчетов;
* вставлять в формы и отчеты рисунки и графики;
* осуществлять операции импорта и экспорта данных;
* публиковать БД на *web*-страницах в *Internet*;
* создавать собственные программы для работы с БД, содержащие меню, диалоговые окна и командные кнопки;
* обеспечивать многопользовательский режим доступа к информации,
* хранящейся в БД.

**Объекты *Access***

*Объект* – отдельные компоненты БД, которые используются для хранения и представления информации .

Основные объекты *Access*: таблицы, запросы, формы, отчеты, макросы и модули . Все они хранятся в общем файле с расширением *mdb* .

*Таблица* используется для хранения информации в БД.

*Запрос* позволяет выбрать нужные данные из одной или нескольких таблиц. С помощью запросов можно модифицировать существующие таблицы, а также создавать новые таблицы.

*Форма* используется для ввода данных в таблицу и для просмотра в заданном формате данных из таблицы или запроса. С ее помощью можно также запустить на выполнение макрос или процедуру.

*Отчет* предназначен для создания документа на основе данных из таблицы или запроса. Этот документ можно распечатать или включить в документ другого приложения, например, *Word* или *Excel*.

*Макрос* представляет собой описание стандартных действий, которые нужно выполнить в ответ на определенное событие. Например, можно определить макрос, который в ответ на выбор некоторого элемента в одной форме открывает другую форму.

*Модуль* позволяет автоматизировать выполнение сложных действий, которые нельзя описать с помощью макросов.

Для создания таких объектов, как таблицы, запросы, формы или отчеты можно использовать специальные средства – *мастера*.

**Запуск и завершение программы**

Обычно при установке пакета *Microsoft Office* ярлык помещается в подменю стартового меню *Windows*, поэтому запуск программы осуществляется из этого подменю. Запустить *Access* можно также, сделав двойной щелчок по ярлыку файла с расширением *mdb*, содержащего БД *Access*.

Завершить работу в *Access* можно одним из следующих способов: нажать сочетание клавиш *Alt*+*F4*, выбрать команду **Выход** из меню **Файл**, щелкнуть по кнопке **Закрыть** окна *Access*.

**Рисунок 2**. Окно программы *Microsoft Access*

**Структура окна *Access***

* *Строка заголовка* содержит системное меню и три кнопки, предназначенные для сворачивания , восстановления размеров и закрытия окна Access.
* *Строка состояния* является последней строкой экрана. В ней появляется информация о назначении выбранного пункта меню или кнопки на панели инструментов.
* *Строка меню* содержит имена раскрывающихся меню команд.
* *Панель инструментов* содержит набор кнопок, которые предназначены для быстрого выполнения определенной команды.

Внутри окна Access могут находиться другие окна, содержащие его различные объекты: таблицы, формы, запросы и т.д.

**Открытие и закрытие БД**

После загрузки *Access* появляется диалоговое окно и пользователю предлагается выбрать один из двух вариантов: создать новую БД или открыть существующую БД.

**Рисунок 3**. Окно, появляющееся при запуске *Access*

Для открытия БД, содержащейся в списке последних открывавшихся БД, нужно сделать двойной щелчок по соответствующему элементу списка. Если нужной БД в этом списке нет, то следует выбрать элемент списка *Другие файлы*..., найти в появившемся диалоговом окне *Открытие файла базы данных* файл, содержащий искомую БД, и щелкнуть по кнопке **ОК**.

Для открытия другой БД во время сеанса работы с *Access* нужно выбрать команду **Открыть** в меню **Файл** или щелкнуть по кнопке **Открыть БД**. В появившемся диалоговом окне нужный файл БД и щелкнуть по кнопке **ОК**. Открытой может быть только одна БД.

После открытия БД на экране появляется окно **База данных** (см. рис. 2), содержащее вкладки для каждого типа объектов *Access*. При щелчке по корешку какой-либо вкладки на экране появляется список имен имеющихся в БД объектов данного типа. При открытии БД первоначально активизируется вкладка **Таблица** и на экран выводится список таблиц. Чтобы открыть нужный объект, достаточно сделать двойной щелчок по его имени или щелкнуть по нему, а затем – по кнопке **Открыть**. Для создания новых объектов следует использовать кнопку **Создать**, а для модификации существующих объектов – кнопку **Конструктор**.

Каждый из открытых объектов *Access* появляется в собственном окне. Для переключения между окнами можно использовать комбинацию клавиш *Ctrl*+*F6* или выбрать нужное окно в меню **Окно**. Окно **База данных** в любой момент можно активизировать нажатием клавиши *F11*. Для закрытия активного окна достаточно нажать комбинацию клавиш *Ctrl*+*F4* или щелкнуть по кнопке **Закрыть**.

Кроме операции открытия с каждым объектом, в зависимости от конкретной ситуации, можно совершать те или иные действия (редактировать, копировать, сохранять и т.д.). Перечень допустимых действий содержится в *контекстном меню*, которое вызывается щелчком правой кнопки мыши по объекту в окне БД или по окну объекта, если он уже открыт.

Для закрытия БД нужно выбрать команду **Закрыть** в меню **Файл** или щелкнуть по кнопке **Закрыть** окна БД.

**Принципы проектирования БД**

Процесс проектирования БД включает следующие этапы:

1. **Определение назначения БД.**
2. **Определение структуры таблиц.**
3. **Определение полей.**
4. **Определение ключевых полей.**
5. **Определение связей между таблицами.**

Для создания БД нужно выбрать в меню **Файл** команду **Создать** или нажать кнопку **Создать базу данных** на панели инструментов. Будет открыто диалоговое окно **Создание**. В этом окне следует выбрать диск и каталог для сохранения БД и ввести имя файла новой БД. На экране появится окно новой «пустой» базы данных.

Быстро создать БД можно с помощью специального мастера. Для этого в окне **Создание** нужно щелкнуть по корешку вкладки **Базы данных** и выбрать из списка нужную БД .

**Создание таблиц**

Для создания таблицы нужно щелкнуть по корешку вкладки **Таблица** окна БД, а затем – по кнопке **Создать**. Access открывает окно **Новая таблица** и предлагает несколько способов создания таблицы.

1. **Режим таблицы**. В этом режиме появляется заготовка таблицы, со стандартными именами Поле1, Поле2 и т.д. После завершения ввода данных в таблицу нужно нажать кнопку **Сохранить**. *Access* предложит добавить в нее ключевое поле. Использование режима таблицы не позволяет установить нужные свойства полей. Для этого следует перейти в режим конструктора таблицы.
2. При создании таблицы в режиме коснтруктора открывается окно таблицы в режиме конструктора. В верхней части окна находится пустой бланк таблицы, содержащей три графы. Следует заполнить, по крайней мере, две из них: **Имя поля** и **Тип данных**.

Свойства полей:

* **Размер поля**. Задает максимальный размер данных, которые будут храниться в поле.
* **Формат поля**. Это свойство поля задает формат представления его значений при выводе на экран или печать.
* **Дробная часть**. Задает количество знаков в числах после запятой.
* **Маска ввода**. Устанавливает символы форматирования для автоматического заполнения полей во время ввода данных .
* **Подпись**. Задает альтернативное имя, которое будет появляться в качестве заголовка поля.
* **Значение по умолчанию**. Задает значение, автоматически вводящееся в поле при создании новой записи.
* **Условие на значение**. Задает логическое выражение, определяющее условие для ввода или редактирования данных.
* **Сообщение об ошибке**. Содержит текст сообщения, которое будет выведено на экран при вводе в поле данных, нарушающих условие на значение.
* **Обязательное поле**. Позволяет указать, обязателен ли ввод данных в это поле.
* **Пустые строки**. Если установить значение этого свойства равным *Да*, то для текстового поля будет разрешен ввод пустых строк .
* **Индексированное поле**. Создается индекс по данному полю. Индекс указывает местоположение записей таблицы на диске и помогает *Access* быстро находить нужные данные .

Для задания свойства поля нужно щелкнуть мышью по изменяемому свойству и выполнить одно из следующих действий : если в ячейке появится кнопка раскрытия списка, нажать эту кнопку и выбрать значение из списка ; если рядом с ячейкой появится кнопка **Построить**, то следует нажать эту кнопку и в появившемся окне построителя ввести нужное значение или выражение.

Для установки свойств таблиц (эти свойства относятся ко всей таблице и всем ее записям) следует щелкнуть правой кнопкой мыши в окне конструктора и выбрать в появившемся контекстном меню пункт **Свойства**. В появившемся диалоговом окне **Свойства** таблицы следует задать нужные свойства таблицы.

**Мастер таблиц**

Чтобы создать таблицу с помощью мастера, нужно в окне диалога **Создание таблицы** нажать кнопку **Мастер**. Мастер **Создание таблиц** предлагает на выбор различные образцы таблиц, каждая из которых содержит набор полей. Пользователь должен отобрать те поля, которые он собирается включить в свою таблицу. После этого мастер автоматически создает таблицу, предоставляя на завершающем этапе возможность пользователю создать ключевое поле и определить связи с уже существующими таблицами.

После завершения работы мастера можно перейти в режим конструктора и произвести необходимые изменения в структуре таблицы и свойствах ее полей.

**Импорт таблиц**

Таблицу можно создать, импортируя данные из файлов других форматов. Если выбран этот способ создания таблицы, то открывается окно диалога Импорт. Пользователь должен указать тип файла, из которого будут импортированы данные, и найти этот файл.

**Связывание таблиц**

Альтернативным способом использования в *Access* данных из других источников (баз данных, электронных таблиц или текстовых файлов) является установление связи с внешними данными – *связывание* данных. При импорте данных создается их копия в новой таблице текущей БД. После этого всякая связь с внешними данными теряется. Связывание позволяет использовать данные из внешнего файла, не импортируя их в *Access* . Поэтому в тех случаях, когда внешний файл часто изменяется или используется в режиме коллективного доступа, целесообразно вместо операции импорта использовать связывание .

Операция связывания производится при помощи мастера **Связь**, который начинает работу при выборе пункта **Связь с таблицами** контекстного меню окна БД.

Для связанных таблиц изменение некоторых свойств полей становится невозможным. В связанную таблицу нельзя добавить новое поле или удалить существующее поле, но можно добавить записи, а также изменить значения отдельных полей. Если связь установлена с таблицей *Access*, то доступна также операция удаления записей.

После того как в БД созданы основные таблицы, следует указать, как они связаны друг с другом. Эти связи Access будет использовать в запросах, формах и отчетах при отборе информации из нескольких таблиц. Задание связей между таблицами позволяет также обеспечить защиту целостности данных в БД.

Связь между двумя таблицами, одна из которых является *главной*, а другая *подчиненной*, устанавливает соответствие между записями этих таблиц. Для установления связи нужно, чтобы в главной таблице существовало поле или группа полей, совокупность значений которых однозначно определяла бы запись (была уникальной). Обычно в качестве поля (группы полей) связи выбирается ключевое поле таблицы, но достаточно, чтобы оно имело уникальный индекс. В соответствие ему ставится поле (группа полей) подчиненной таблицы, называемое внешним ключом. Связь между записями устанавливается по совпадению значений в полях связи. *Access* анализирует значение поля связи (ключа) любой записи главной таблицы и связывает ее со всеми записями подчиненной таблицы, имеющими такое же значение внешнего ключа.

**Связь типа «один-ко-многим»**

При таком типе связи каждой записи в главной таблице могут соответствовать одна, несколько или ни одной записи в подчиненной таблице, а каждая запись в подчиненной таблице не может иметь более одной соответствующей ей записи в главной таблице. Если запись в подчиненной таблице не имеет соответствующей ей записи в главной таблице, то значение поля связи в такой записи должно быть пусто.

**Связь типа «один-к-одному»**

При связи типа «один-к-одному» каждой записи в главной таблице может соответствовать не более одной записи в подчиненной таблице, и наоборот, каждая запись в подчиненной таблице не может иметь более одной соответствующей ей записи в главной таблице.

Этот тип связи применяется реже, так как такие данные могут быть помещены в одну таблицу. Связь типа «один-к-одному» обычно используют для разделения таблиц, имеющих много полей, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

**Связь типа «многие-ко-многим»**

При установлении связи между таблицами возможна ситуация, когда между ними нельзя установить отношение «главная-подчиненная» из-за того, что любой записи в одной таблице может соответствовать несколько записей из другой таблицы. Если, например, имеются две таблицы **Студенты** и **Преподаватели**, то между этими таблицами нельзя установить ни одну из описанных выше связей, поскольку, например, каждый студент сдавал экзамены нескольким преподавателям, а каждый преподаватель принимал экзамен у нескольких студентов .

*Access* непосредственно не поддерживает такой тип связи. Но ее можно реализовать в виде двух связей типа «один-ко-многим» с помощью третьей (связующей) таблицы. В

качестве такой связующей таблицы в приведенном выше примере используется таблица **Сессия**. Она связана как с таблицей **Студенты** по полю **Код студента**, так и с таблицей **Преподаватели** по полю **Код преподавателя**. После того как эти связи установлены, можно легко определить с помощью соответствующих запросов, у каких студентов принимал экзамены данный преподаватель или кому сдавал экзамены данный студент.

**Создание связей**

Для создания связей между таблицами нужно вернуться в окно БД, закрыть все открытые таблицы и щелкнуть по кнопке **Схема данных** панели инструментов либо вызвать щелчком правой кнопки мыши контекстное меню и выбрать в нем пункт **Схема данных**. Если связи в БД определяются впервые, то будет открыто пустое окно **Схема данных**. В это окно нужно добавить таблицы, между которыми устанавливается связь.

Для добавления таблиц следует вызвать щелчком правой кнопки мыши контекстное меню окна **Схема данных** и выбрать в нем пункт **Добавить таблицу**. Откроется диалоговое окно **Добавление таблицы**, содержащее список таблиц БД (см. рис. 4). Для добавления таблицы нужно щелкнуть по ее имени, а затем — по кнопке **Добавить**. После того как все таблицы отобраны, нужно закрыть это окно и вернуться в окно **Схема данных**.

**Рисунок 4**. Окно **Добавление таблицы**

Чтобы определить связь между таблицами, находящимися в окне **Схема данных**, следует перенести с помощью мыши поле связи главной таблицы и поместить его на поле связи подчиненной таблицы. Откроется диалоговое окно Связи (см. рис. 5).

**Рисунок 5**. Окно **Связи**

В левом столбце выводятся имена главной таблицы и ключа, используемого для связи, а в правом столбце – имена подчиненной таблицы и внешнего ключа.

Для изменения поля следует открыть список полей справа от его имени. Если связь производится по нескольким полям, то их имена можно добавить, используя пустые строки.

Обычно *Access* сам определяет тип создаваемой связи, проводя анализ полей, для которых определяется связь. Если только одно из полей является ключевым или имеет уникальный индекс, создается ·связь «один-ко- многим». Связь «один-к-одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.

Создавая связь, нужно настроить режим обеспечения целостности данных. Обеспечение целостности позволяет избежать ситуации, когда в подчиненной таблице имеются записи, не связанные с записями главной таблицы. Если этот режим включен, то *Access* не разрешит добавить в подчиненную таблицу запись, для которой не найдется связанной с ней записи из главной таблицы. Нельзя будет также удалить из главной таблицы запись, имеющую связанные с ней записи в подчиненной таблице.

Режим обеспечения целостности данных этой связи можно включить, если выполнены следующие условия:

* поле связи главной таблицы является первичным ключом или имеет уникальный индекс;
* связанные поля имеют один и тот же тип данных;
* обе связанные таблицы принадлежат одной базе данных *Access*.

Если для связи включен этот режим, то можно дополнительно указать, следует ли автоматически выполнять для связанных записей операции *каскадного обновления* и *каскадного удаления*. Если включить режим **Каскадное обновление связанных полей**, то при изменении значения ключа в главной таблице будут автоматически обновлены соответствующие значения в связанных записях подчиненной таблицы. При включении режима **Каскадное удаление связанных записей** при удалении записи из главной таблицы будут автоматически удалены связанные с ней записи в подчиненной таблице.

В том случае, когда эти режимы не включены, а режим обеспечения целостности данных включен, *Access* не позволит изменить значение в ключевом поле главной таблицы, а также удалить запись в главной таблице, если в подчиненной таблице имеются данные, связанные с этой записью.

После завершения операции создания связи нужно нажать кнопку **ОК**. Связь отображается в виде линии, соединяющей две таблицы. Если включен режим обеспечения целостности данных, то *Access* изобразит на конце линии, соответствующей главной таблице, цифру **1**. На другом конце линии, соответствующем подчиненной таблице, будет изображен символ бесконечности ∞ для связи типа «один-ко-многим» и цифра *1* для связи типа «один-к-одному».

Любую связь можно изменить или удалить. Для изменения связи нужно сделать двойной щелчок по линии связи, и отредактировать ее в открывшемся окне **Связи**. Чтобы удалить связь, следует щелкнуть по ней и нажать клавишу **Delete**.

**Формы**

Формы предназначены для просмотра, ввода, редактирования и управления данными. При использовании формы Access предоставляет пользователю существенно больше возможностей для работы с информацией по сравнению с ее стандартным представлением в режиме таблицы. Формы позволяют:

* Отобразить на экране всю запись целиком.
* Размещать и нередактируемые (вычисляемые) поля.
* Добавлять комментарии, рисунки, графики, изменять их внешний вид .
* Существенно упростить, а зачастую полностью автоматизировать ввод новых данных .
* Размещать на них различные кнопки (кнопочные формы), нажатие на которые приводит к открытию других форм, выполнению запросов, печати отчетов и т. д.
* Использовать их в качестве диалоговых окон и содержать собственное меню пользователя .

Основным источником данных для формы являются таблицы и запросы.

Построение формы – итеративный процесс. После создания макета формы нужно просмотреть его, чтобы убедиться в его пригодности. Если необходимо внести изменения, можно вернуться к корректировке макета.

Для создания макета формы нужно в окне БД перейти на вкладку **Формы** и нажать клавишу **Создать**. После выбора источника данных для формы, а также способа ее создания, при переходе к следующему этапу автоматически обновляются панели инструментов. Появляется панель инструментов **Конструктор форм**.

Создание, корректировка и просмотр формы осуществляется в различных режимах:

* в режиме конструктора форма создается и корректируется;
* в режиме формы или режиме таблицы форма используется для работы с данными;
* в режиме предварительного просмотра форма просматривается перед печатью.

Все эти режимы выбираются с помощью кнопки **Вид** инструментов **Конструктор форм** или через меню **Вид**.

**Структура формы**

Форма состоит из нескольких разделов (см. рис. 6), причем обязательным является только один из них – область данных. Также на форме присутствуют следующие разделы: заголовок и примечание формы, а также верхний и нижний колонтитулы. Информация в них вводится разработчиком формы. Эти разделы имеют следующие назначения:

**Заголовок формы** может содержать название формы, инструкции по работе с ней, а также другую информацию (текущую дату, время и т.д.).

**Верхний колонтитул** может содержать заголовки и любые другие сведения, помещаемые в верхней части каждой страницы формы при печати.

**Нижний колонтитул** также используется при печати и может содержать номер страницы, дату и другую информацию.

**Примечание формы** может содержать инструкции по использованию формы, кнопки и свободные элементы управления для ввода или вывода данных.

Форма может включать *подчиненные формы*, внутри которых могут отображаться три раздела: заголовок, область данных и примечание.

**Рисунок 6**. Структура формы

**Виды форм**

Различные виды форм создаются обычно мастером форм или пользователем с помощью конструктора. Кроме того, существуют мастер *диаграмм* и мастер *сводная таблица*.

Возможно создание трех видов автоформ: *в один столбец*, *табличной* и *ленточной формы*.

*Форма в один столбец* отображает поля, расположенные в один столбец. На экране отображается одна запись.

*Табличная форма* отображает данные в виде нескольких строк и столбцов.

*Ленточная форма* похожа на табличную форму и отличается лишь внешним оформлением.

В отличие от автоформ при использовании мастера форм поля в форму можно выбирать. Мастер может создать кроме вышеперечисленных видов форм *составную форму*. Она состоит из *главной* формы и *подчиненной*. В этих формах отображаются данные из разных таблиц, причем эти таблицы чаще всего связаны отношением «один-ко-многим». Данные главной формы отображаются в один столбец, а подчиненная форма обычно имеет табличный формат.

**Запросы**

Одним из основных инструментов обработки данных в СУБД являются запросы. В Microsoft Access имеется удобное для пользователя графическое средство формирова­ния запроса по образцу - QBE (Query By Example), с помощью которого легко может быть построен сложный запрос.

Запрос строится на основе одной или нескольких таблиц. При этом могут исполь­зоваться таблицы базы данных, а также другие запросы. Запрос QBE содержит схему данных, включающую используемые таблицы, и бланк запроса. При конструировании запроса достаточно, работая мышью, выделить и перетащить необходимые поля из таб­лиц, представленных в схеме данных запроса, в бланк запроса и ввести условия отбора записей.

С помощью запроса можно выполнить следующие виды обработки данных:

♦ выбрать записи, удовлетворяющие условиям отбора.

♦ включить в результирующую таблицу запроса заданные пользователем поля.

♦ произвести вычисления в каждой из полученных записей.

♦ сгруппировать записи с одинаковыми значениями в одном или нескольких полях для выполнения над ними групповых функций.

♦ произвести обновление полей в выбранном подмножестве записей.

♦ создать новую таблицу базы данных, используя данные из существующих таблиц.

♦ удалить выбранное подмножество записей из таблицы базы данных.