

## **Пример создания базы данных в Microsoft Access**

**Цель:** В результате выполнения лабораторной работы студент должен иметь представление о СУБД, освоить приемы работы с базами данных на начальном уровне на примере MS Access.

### **Теоретический минимум:**

1. Базы данных и СУБД.
2. Структура простейшей БД.
3. Свойства полей БД.
4. Типы данных.
5. Режим работы с БД.
6. Объекты БД.
7. Разработка схемы данных.
8. Пошаговое создание БД.
9. Информационный анализ предметной области и построение информационно-логической модели.
10. Создание экранных форм и их использование для ввода данных и просмотра таблиц.
11. Конструирование запросов.
12. Вывод результатов обработки данных с помощью отчетов.
13. Создание интерфейса для управления диалоговым приложением пользователя.
14. Диспетчер кнопочных форм.

### **Краткие теоретические сведения**

**Базы данных** - это организованная структура данных для хранения информации.

Такое понятие базы данных обусловлено тем, что современные системы управления базами данных (СУБД) позволяют размещать в своих структурах не только данные, но и методы (то есть программный код), с помощью которых происходит взаимодействие с потребителем или с другими программно-аппаратными комплексами. Если в базе нет никаких данных, то это всё равно полноценная база данных. В такой базе данных всё-таки есть информация - это структура базы. Она определяет методы занесения и хранения данных в базе.

С понятием базы данных тесно связано понятие системы управления базой данных.

***Система управления базами данных (СУБД)** - это комплекс программных средств, предназначенных для создания структуры новой базы, наполнения её содержимым, редактирования содержимого и визуализации информации.*

Под визуализацией информации понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода или передача по каналам связи.

В мире существует множество СУБД. Однако большинство из них опирается на единый устоявшийся комплекс основных понятий. Поэтому рассмотрим и обобщим понятия на примере только одной системы управления базами данных Microsoft Access, входящей в пакет Microsoft Office. Базы данных могут содержать различные объекты, но основными объектами являются её таблицы. Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в структуре базы данных являются соответственно поля и записи. Поля базы данных определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. Изменение состава полей исходной таблицы (или их свойств) приводит к изменению структуры базы данных, то есть создаётся новая база данных.

Любая система (структура) базы данных является моделью организации хранения данных (информации). Существуют следующие понятия типовых моделей баз данных:

- ✓ реляционная (табличная);
- ✓ иерархическая;
- ✓ сетевая;
- ✓ объектно-ориентированная.

Сегодня наиболее распространены реляционные модели.

В СУБД Microsoft Access реализуются следующие **основные свойства полей в структурах таблиц базы данных**:

✓ *Имя поля* (заголовок столбца таблицы) - определяет порядок обращения с данными этого поля при автоматических операциях с базой. *Имя поля* в таблице должно быть уникальным, содержать не более 64 символов и удовлетворять соглашениям об именах объектов *MS Access*. Следует избегать длинных имен.

✓ *Тип поля* - определяет тип данных, которые могут содержаться в данном поле.

✓ *Размер поля* - определяет предельную длину (в символах) данных, которые могут размещаться в данном поле. *Размер поля* – задает максимальный размер данных, хранимых в поле в соответствии с выбранным типом данных. Рекомендуется задавать минимально допустимый размер поля.

✓ *Формат поля* - определяет способ форматирования данных в ячейках, принадлежащих полю. *Формат поля* – задает вид отображения значений полей всех типов, кроме *Поля объекта OLE*. В Access определены в виде раскрывающихся списков встроенные стандартные форматы для полей типов: *Числовой*, *Дата/Время*, *Логический* и *Денежный*;

✓ *Маска ввода* (средство автоматизации ввода) - определяет форму, в которой вводятся данные в поле. *Маска ввода* – свойство предназначено для облегчения ввода и контроля вводимой информации. *Маска ввода* строится с помощью мастера, выбирается из списка (для некоторых типов полей) или самостоятельно пользователем на основе таблицы специальных символов, которую можно увидеть в справочной системе MS Access.

Примеры:

1. Для ввода номера телефона \ (000") "000\ -00\ -00 => (065) 123-45-67;

2. Для ввода почтового индекса "Index" 000000 => Index 123456;

3. Для ввода имен >L< => Иван, Сергей – первая буква прописная, остальные строчные.

Свойства *Формат поля* и *Маска ввода* задают шаблон, в котором данные вводятся в таблицу. Свойство *Формат поля* используется для данных *числового* и *денежного* типов и не определено для текстовых полей, полей МЕМО и гиперссылок. Свойство *Маска ввода* используется для отображения символьных констант и при вводе данных в пустые поля. Маска обеспечивает сохранение информации в определенном формате. Главное отличие свойств: *Формат поля* влияет на все содержимое поля, а *Маска ввода* – на часть.

✓ *Подпись* - определяет заголовок столбца таблицы для данного поля. Если подпись не указана, то в качестве заголовка столбца используется свойство *Имя поля*.

✓ *Значение по умолчанию* - то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных). *Значение по умолчанию* – свойство, присущее полям всех ти-

пов, за исключением полей типа *Счетчик* и *Объект OLE*. Значение этого свойства автоматически подставляется при создании новой записи. Например, можно использовать выражение *=Date()* – вывод по умолчанию текущей даты, символ *=* (равно) обязателен.

- ✓ *Условие на значение* - ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный или тип даты).

- ✓ *Сообщение об ошибке* - текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности выполняется автоматически, если задано свойство *Условие на значение*). *Условие на значение* и *Сообщение об ошибке* – эти свойства применимы ко всем типам полей, кроме типа *Счетчик* и *Объект OLE*. При переходе на другую запись проверяется указанное условие, если оно не выполняется, выдается сообщение об ошибке – значение второго свойства. Например, условие *">#01.01.2008# and <#01.01.2009#"* задает диапазон введенной даты. Константа типа *Date* обрамляется с обеих сторон знаком диеза *"#"*.

- ✓ *Обязательное поле* - свойство, определяющее обязательность заполнения данного поля при наполнении базы. *Обязательное поле* – свойство можно указать для всех полей, кроме поля *Счетчик* (оно задано по умолчанию). Возможные значения *"Да"*, *"Нет"* выбираются из списка. Если указано *"Да"*, то ввод значения поля обязателен.

- ✓ *Пустые строки* - свойство, разрешающее ввод пустых строчковых данных (от свойства *Обязательное поле* отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например, текстовым). Свойство *Пустые строки* – разрешает или запрещает ввод пустых строк, возможны два значения: *"Да"*, *"Нет"*. Значение *"Да"* обычно используется для текстовых полей, *МЕМО* и гиперссылок.

- ✓ *Индексированное поле* - если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. *Индексированное поле* – указывает, будет ли создан индекс по этому полю. Возможные значения: *"Нет"*, *"Да (Допускаются совпадения)"*, *"Да (Совпадения не допускаются)"*. *Индекс* – вспомогательная таблица, в которой хранится порядок следования записей,

упорядоченных по этому полю. Для первичного ключа таблицы указывается – "Да (Совпадения не допускаются)". Access автоматически индексирует таблицу по значению ключа. При создании запросов и отчетов может потребоваться создать дополнительные индексы по значениям других полей. Индексы позволяют ускорить поиск данных в больших таблицах. Каждая таблица может иметь до 32 индексов, 5 из которых могут быть составными, в составной индекс могут входить не более 10 полей. Чтобы создать дополнительный простой ключ, нужно в режиме Конструктора задать свойство Индексированное поле – "Да (Совпадения допускаются)". Чтобы создать дополнительный составной ключ, нужно в режиме Конструктора открыть окно Индексы, нажав соответствующую кнопку Индексы на панели инструментов. В этом окне выводятся все индексы таблицы. В поле Индекс ввести имя индекса (произвольное), в поле Имя поля выбрать из списка сначала одно поле, перейти на строку ниже, выбрать другое и т. д. Закрывать окно, подтвердить сохранение структуры таблицы.

- ✓ Сжатие Юникод – включает и отключает сжатие текстовых полей, полей МЕМО и гиперссылок, используя универсальный международный код. По умолчанию включено. Можно отключить, тем самым сэкономить память.

- ✓ Число десятичных знаков – указывает количество десятичных знаков после запятой в числовых и денежных типах полей; если выбрать значения свойства Авто, то число отображается так, как было введено в поле.

- ✓ Новые значения – свойство определено только для поля Счетчик и указывает, какие значения автоматически будут подставляться в поле при создании новой записи. Возможные значения: последовательные, случайные (числа). Принято по умолчанию первое значение и лучше использовать его.

- ✓ Поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных.

Система управления базами данных Microsoft Access поддерживает следующие **типы данных**:

- ✓ **Текстовый** - используется для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

✓ *Поле Мемо* - специальный тип данных для хранения больших объёмов текста (до 65535 символов), предназначено для хранения комментариев, не может быть ключевым полем, ввод данных в это поле можно осуществлять в режиме таблицы непосредственно в поле либо через область ввода, которая открывается при нажатии комбинации клавиш.

✓ *Числовой* - для хранения действительных чисел, длина поля зависит от значения свойства *Размер поля*.

✓ *Дата/время* - для хранения дат и текущего времени, с помощью свойства *Формат поля* задается способ представления значения.

✓ *Денежный* - предназначенный для проведения денежных расчетов с точностью до 15 знаков в целой и 4 знаков в дробной частях.

✓ *Счётчик* - уникальные целые числа, последовательно возрастающие на единицу, значения этого поля нельзя изменить или удалить (ввести); используется в качестве первичного ключа таблицы.

✓ *Логический* - хранит логические данные (только два значения *да/нет*, *истина/ложь*, *вкл./выкл.*).

✓ *Поле объекта OLE* - используется для вставки объекта другого приложения, например, рисунка, фотографии, звука с помощью меню *Вставка → Объект*.

✓ *Гиперссылка* - специальное поле для указания пути к файлу на жестком диске или адреса (универсальный указатель ресурса) Web-страницы в сети Интернет.

✓ *Мастер подстановки* - выбор этого типа данных запускает *Мастер подстановок*, обычно используется для внешних ключей, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

**Постановка задачи:** В базе данных сети Билайн регистрируют и учитывают разговоры абонентов сети по автоматической связи. Разговоры предоставляют в кредит. Оплату производят по счетам, высылаемым абонентам с периодичностью согласованной с абонентом (1 раз в месяц, раз в 2 месяца и т.п.). Стоимость разговора зависит от оператора вызываемого абонента, длительность разговора измеряется в секундах, и зависит от того является ли номер телефона личным или принадлежит юридическому лицу.

### Построение схемы данных:

Исходя из условий задачи, предлагается следующий список полей:

- ✓ Номер заказа.
- ✓ Номер телефона.
- ✓ Фамилия ИО.
- ✓ Дата рождения.
- ✓ Адрес.
- ✓ Личный/организация.
- ✓ Код (оператора).
- ✓ Стоимость 1 секунды.
- ✓ Длительность разговора.
- ✓ Дата.

### Определение типов полей.

*Номер заказа* – тип **счетчик**, т.к. требуется уникальный номер для каждого телефонного звонка. *Номер телефона, Фамилия ИО, Адрес* – **текстовый** тип (номер телефона будет вводиться по шаблону 000-00-00). *Личный/организация* – **логический** тип, т.к. достаточно двух значений (да – личный, нет – организация). *Код* (целое число), *Стоимость 1 секунды* (нецелое число), *Длительность разговора* (для удобства тоже целое число) – **числовой** тип. *Дата, Дата рождения* – тип **дата/время**.

### Распределение полей по таблицам.

При объединении полей в одну таблицу возникает нецелесообразный повтор следующей информации:

- при каждом звонке через определенного оператора будет повторяться информация о стоимости 1 секунды,
- при звонке с некоторого телефона будет повторяться информация о владельце.

Для учета этих условий можно оформить 3 таблицы:

1) *Справочник абонентов:* ( *Номер телефона, Фамилия, Имя, Отчество, Личный/организация, Дата рождения, Адрес*).

2) *Справочник кодов операторов и тарифов:* ( *Код, Стоимость* ).

3) *Регистрация разговоров:* ( *Номер заказа, Номер телефона, Код, Длительность разговора, Дата* ).

### **Связи между таблицами.**

*Номер телефона (Регистрация разговоров) – Номер телефона (Справочник абонентов). Тип связи: "один-ко-многим".*

*Код (Справочник кодов операторов и тарифов) – Код (Регистрация разговоров). Тип связи: "один-ко-многим".*

### **Технология работы:**

#### **Шаг 1. Создание базы данных "Билайн".**

Создать и сохранить файл базы данных в рабочей директории.

#### **Шаг 2. Создание таблиц.**

Создать таблицы при помощи конструктора таблиц. Задать необходимые параметры (типы данных, маски ввода, размеры полей, определить ключевые поля).

#### **Шаг 3. Установка связи между таблицами.**

Нажав кнопку *Схема данных* на панели инструментов, добавить таблицы в окно установки связей, организовать связи (Обозначить *Обеспечение целостности данных*).

#### **Шаг 4. Создание форм ввода информации (простые, сложные, кнопочные).**

#### **Шаг 5. Ввод данных в таблицы с помощью форм.**

#### **Шаг 6. Создание запросов.**

✓ **Создание запроса на выборку.** Например, выбрать данные одного абонента.

#### ✓ **Создание параметрического запроса.**

Параметрический запрос – это запрос на выборку, в котором условие отбора определяется во время его выполнения. Например, создать параметрический запрос, который выбирает звонки с одного номера, запрос на звонки определенного числа.

#### ✓ **Использование в запросе вычисляемого поля.**

В результате выполнения запроса с вычисляемым полем, Access позволяет не только выбирать из таблицы содержащуюся в ней информацию, но также производить вычисления и отображать результат вычисления в результирующей таблице. Таким обра-



зом, можно получить данные, отсутствующие в исходной таблице. Например:

1. Создать запрос из таблицы *Абоненты*, который будет рассчитывать возраст, исходя из их даты рождения. Для расчета возраста воспользоваться встроенными функциями Round() – округление до целого, и Date() – сегодняшняя дата.

2. Вычислить стоимость каждого разговора (Воспользоваться формулой: Стоимость: If ([Личный]=Да;[Длительность разговора]\*[Стоимость 1 секунды]; [Длительность разговора]\*[Стоимость 1 секунды]\*3).

✓ **Построение запроса на обновление.**

С помощью данного запроса можно обновлять данные в таблицах. Например, создадим запрос, который изменяет Фамилию абонента.

✓ **Построение запроса на добавление записей.**

Запрос на добавление переносит (добавляет) данные из одной таблицы в другую. Например, необходимо создать запрос на добавление нескольких записей в уже имеющуюся таблицу *Архив*.

✓ **Построение запроса на создание таблицы.**

Данный тип запроса создает новый объект *Таблица*.

✓ **Создание перекрёстного запроса.**

Перекрёстный запрос представляет собой сводную таблицу, в которой данные могут быть сгруппированы определённым образом, а также могут подсчитываться среднее, максимальное, минимальное и другие значения.

## **Шаг 7. Создание отчетов.**

В качестве примера создать извещения об оплате услуг. Изменить размер элемента управления в разделе *Заголовок отчета* или *Нижний колонтитул*. Перетащить с помощью мыши элементы управления в разделах с одного места на другое. Изменить порядок вывода столбцов. Выделить элемент управления в разделах верхнего и нижнего колонтитулов, изменить его форматирование.

## **Шаг 8. Печать отчетов.**