

Теоретическое занятие № 9
по ОДА Р4 «Информационное обеспечение профессиональной
деятельности»
Тема: «Системы управления базами данных»
для отделения Фармация

Интерфейс работы с базами данных Base

Base – это компонент LibreOffice, предназначенный для создания баз данных. Источники данных или база данных – это массив информации, организованный особым образом для обеспечения более легкого доступа, управления и обновления. Например, список имён и адресов – это источник данных, используя который можно подготовить стандартные письма для почтовой рассылки. Список складских запасов может быть источником данных, управляемым при помощи LibreOffice.

Если у вас есть информация, которую вы хотели бы упорядочить, Base поможет вам это сделать. Конечно, для организации информации можно использовать электронную таблицу, но поддержание в актуальном состоянии таблицы Calc часто может быть более сложным и трудоемким, чем использование базы данных Base.

Эта глава предназначена для людей, которые никогда не использовали Base или для тех, кто хочет ознакомиться с основами создания и использования баз данных. Терминология сведена к минимуму. В этой главе рассматриваются мастера, необходимые для создания частей базы данных и принципы, которые используют мастера. Глава также содержит ссылки на дополнительную информацию в *Base Handbook* или в *Руководстве по Base* (англоязычные версии которых находятся в процессе написания).

Base Handbook и Руководство по Base предназначены для людей, которые уже понимают основы работы в Base и хотят изучить принципы работы программы более глубоко.

Примечание

Чтобы использовать Base, необходимо установить Java Runtime Environment (JRE). Используйте меню Сервис ▸ Параметры ▸ LibreOffice ▸ Расширенные возможности, чтобы выбрать нужный JRE из уже установленных на компьютере версий.

Если JRE не установлен, то необходимо скачать и установить его. Для Windows посетите сайт www.java.com. Для Linux, скачайте JRE с указанного сайта, либо используйте `openjdk-7-jre`, доступный в репозитории вашего дистрибутива Linux. Пользователи Mac OS X могут установить JRE от Apple Inc.

Одна из вещей, которая иногда смущает людей – это терминология. В

частности, что за разница между понятиями база данных и источник данных? Base использует эти термины, как синонимы. Это потому, что база данных – это набор данных, которые могут существовать в различных формах. Они включают в себя текстовые файлы, электронные таблицы или файл, созданный программой управления базами данных.

Данные бесполезны, если мы не можем использовать информацию, которая в них содержится. Необходимо создать структуру для использования данных. Формулы и ссылки на ячейки в электронной таблице Calc используются для получения информации из данных, содержащихся в ячейках. Программы управления базами данных могут делать то же самое.

При работе в Base создаётся файл базы данных. Этот файл содержит данные и все структуры для получения информации из данных, созданные вами. Поскольку всё включено в один файл, то он называется встроенная база данных.

Base также может быть использован для подключения к другим базам данных. В этом случае Base создает отдельный файл для создания соединения. Он не является частью базы данных, так как существует вне файла базы. Для получения дополнительной информации обратитесь к *Base Handbook* или *Руководству по Base*.

Типы баз данных

Base может создавать и работать с двумя типами баз данных: плоскими и реляционными. В этой главе рассматриваются плоские базы данных. Реляционные базы данных подробно описаны в *Base Handbook* и в *Руководстве по Base*.

Плоская база данных содержит одну или несколько таблиц, каждая из которых содержит одно или несколько полей. Каждая таблица является полностью независимой от всех других таблиц в базе данных. Например, у вас есть база данных адресов с несколькими таблицами в ней. В них можно хранить ваши личные контакты, ваши деловые контакты и контакты вашего супруга. Некоторые контакты могут находиться в более чем одной таблице, при этом в разных таблицах информация об одном и том же человеке может отличаться.

Реляционная база данных содержит одну или несколько таблиц с одной или более связей, образованных между этими таблицами. Каждая связь определяется парой полей. Одно поле каждой пары относится к одной таблице, а второе поле принадлежит другой таблице. Хотя плоская база данных и может быть использована при множестве таблиц, но в ней отсутствует механизм для определения отношений между таблицами. В такой базе данных одинаковые данные должны быть введены по отдельности в одинаковых полях разных таблиц, при этом не должно быть ошибок при вводе данных. Хорошо разработанная реляционная база данных позволяет вводить данные только один раз, уменьшая возможные ошибки.

Плоская база данных

Рассмотрим адресную книгу. Как правило, данные в адресной книге, могут быть разделены на группы в зависимости от отношений между данными. Например, все имена помещены в одну группу. Другие группы могут включать фамилии, адреса, номера телефонов, дни рождения и так далее. Вероятно можно также разделить на отдельные группы номера телефонов по их местонахождению (дома, на работе, мобильный). Если эта информация находилась в электронной таблице, то вы скорее всего использовали отдельный столбец для каждой из этих групп. В базе данных этот столбец называется **поле**.

Другие отношения между данными могут быть использованы для дальнейшего определения структуры этих данных. В адресной книге одно значение в каждой из этих групп описывает конкретное лицо. В таблице будут столбцы, как описано выше. Мы переставим строки так, чтобы каждая из них содержала информацию о человеке.

Распределим данные в таблице. Каждый столбец содержит данные с одинаковыми свойствами. Каждая строка содержит данные, описывающие конкретную организацию или человека и называется **записью**. Структура строк и столбцов называется **таблицей**.

Обработка данных без какой-либо структуры, как мы уже упоминали выше – это не самое лучшее решение. Теперь, после создания таблицы, у нас есть плоская база данных и мы можем воспользоваться ей, чтобы получить информацию.

Мы будем использовать **запросы** для получения информации из базы данных. Запрос – это такой способ задавать вопросы базе данных и получить на них ответы. Если мы хотим передать информацию из базы данных кому-то другому, то мы передадим им **отчёт**, основанный на запросе, который мы создали (Отчёт также может быть создан прямо из таблицы).

Таким образом, плоская база данных состоит из **таблицы**, столбцы которой называются **полями**, а строки называются **записи**. Мы используем структуру таблицы, чтобы задавать вопросы и получать ответы, при помощи **запросов**. Для предоставления информации, полученной из запроса, другим людям, мы создадим **отчет**. Это термины, которые мы используем в базе данных: таблица, с ее полями и записями, запросы и отчеты.

Планирование базы данных

Совет

Перед созданием базы данных, спланируйте свои действия: подумайте, что вы хотите делать и почему. От планирования зависят результаты, которые вы получите при использовании базы данных.

Вы должны чётко представлять себе, что вы будете делать с данными, которые

у вас есть. Это представление включает в себя знание о том, как вы будете делить данные по столбцам, полям, строкам и записям. Этим вы определите, какой будет ваша таблица. *Какую информацию вы хотите получить из данных?* Этим вы определите, какие вопросы надо будет задавать базе данных. *Какую информацию вы хотите увидеть в отчете?* Итак, потребуются некоторое время, чтобы обдумать, какой результат вы хотите получить, прежде чем начать создавать базу данных.

В *Руководстве по Base* в *Главе 2. Планирование и проектирование баз данных* подробно освещается эта тема.

Мастер таблиц (смотрите раздел [Таблицы: использование мастера для создания таблиц](#)) содержит список предварительно настроенных таблиц для деловых и для личных целей. Мастер также содержит список полей для каждой из таких таблиц. Каждому из полей также уже заданы необходимые свойства.

В результате мастер выполняет большую часть планирования за вас. Тем не менее, необходимо будет изучить готовую таблицу и подумать, например, следует ли использовать все предлагаемые поля? Будут ли использоваться дополнительные поля, которых нет в готовой таблице из мастера? Какие свойства должны иметь эти поля?

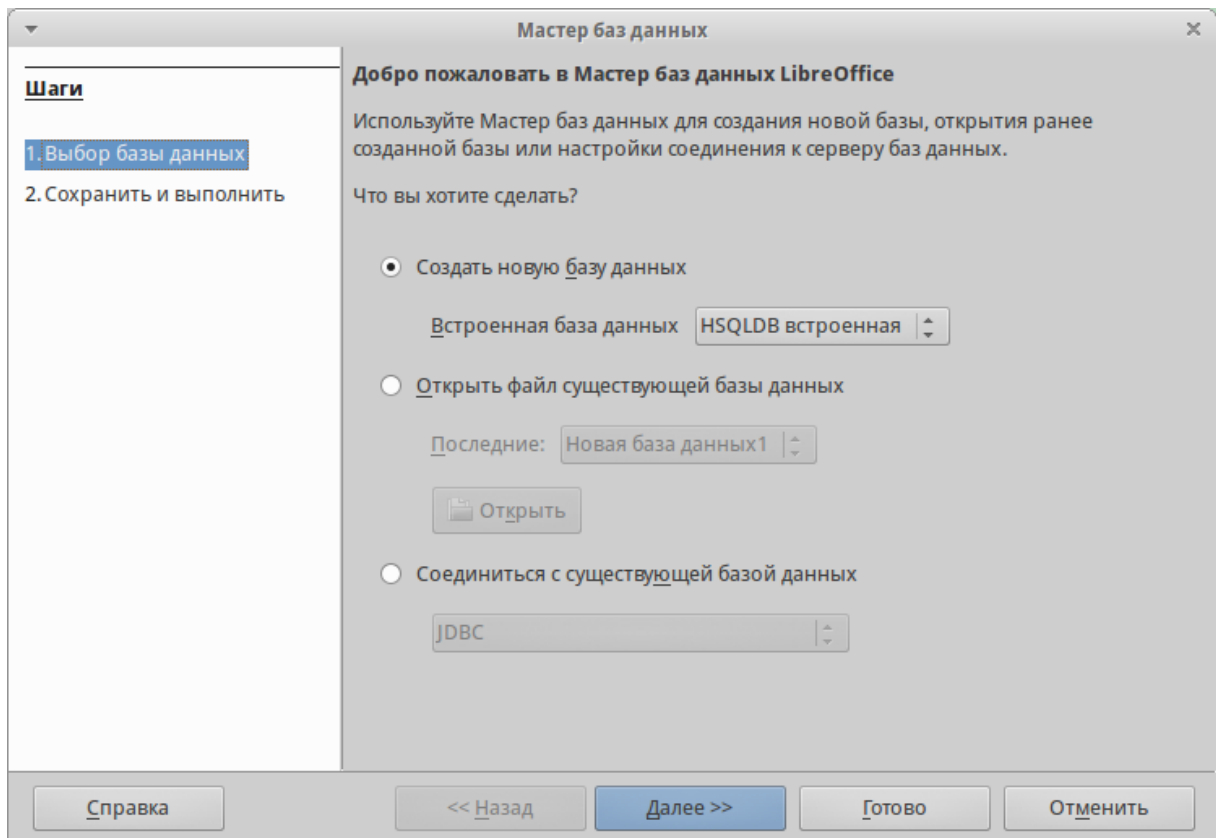
Таким образом, нужно понимать, что вы делаете, как и зачем. Попрактикуйтесь в создании объектов базы данных, прежде чем создавать их в реальной базе. Узнайте сначала на примере, как создаётся база данных. При необходимости делайте для себя заметки о принципах работы Base.

Создание нового файла базы данных

После планирования вашей базы данных, создайте новый файл базы данных. Чтобы это сделать, откройте Мастер баз данных одним из следующих способов:

- Выберите пункт меню *Файл* ▸ *Создать* ▸ *Базу данных*.
- Нажмите на треугольник справа от значка *Создать* на* *Стандартной панели инструментов** и выберите из списка *Базу данных*.
- Нажмите на значке *Новая база данных* на стартовом экране после запуска LibreOffice.

Первый шаг мастера баз данных состоит из трех вариантов. Используйте верхний вариант, чтобы *Создать новую базу данных*. Средний и нижний варианты используются для открытия существующей базы данных (смотрите [Открытие существующей базы данных](#)).



Создание новой базы данных, используя мастер

Создание базы данных в мастере баз выполняется в два шага:

1. На первом шаге **Выбор базы данных**:
 - a. Под вопросом «Что вы хотите сделать?», выберите вариант *Создать новую базу данных* (он выбран по умолчанию).
 - b. Нажмите кнопку *Далее* внизу окна мастера.

Примечание

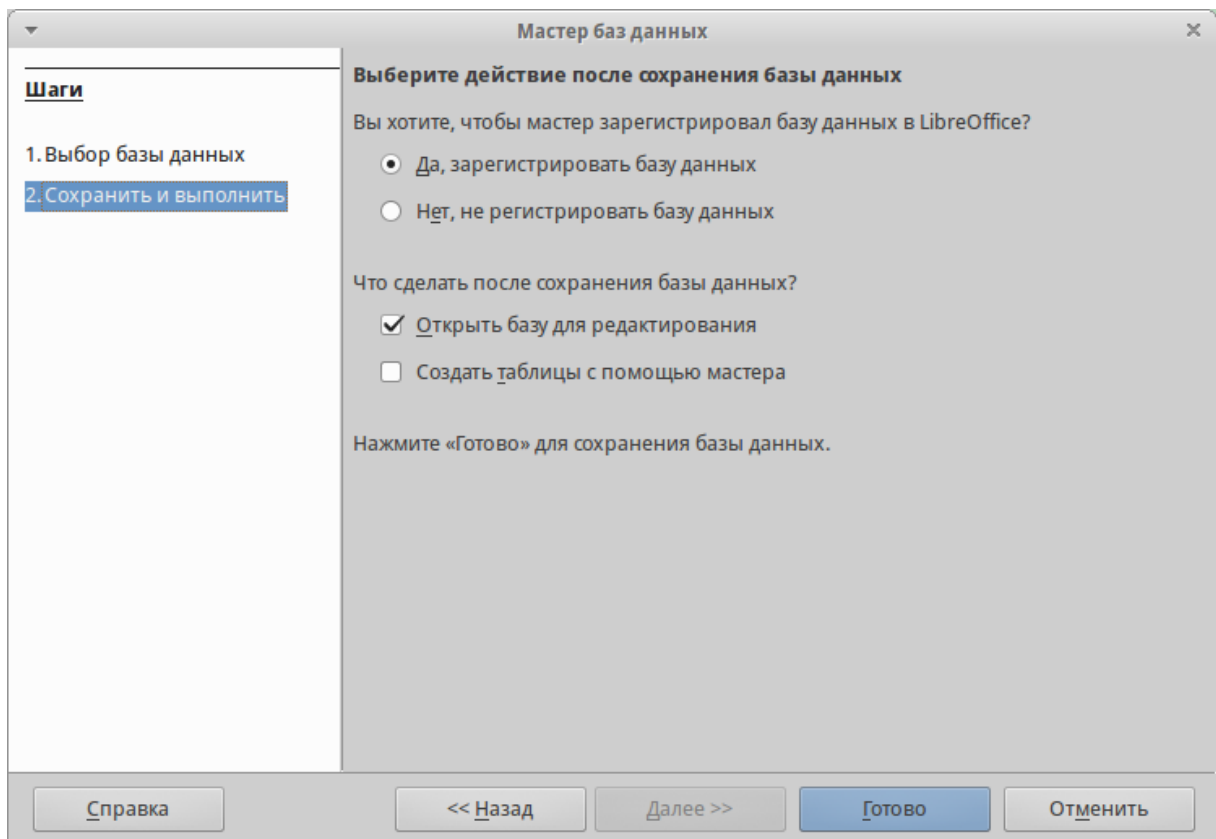
Начиная с версии 4.2.0, первая страница мастера баз данных получила новый вариант в выпадающем списке выбора типа встроенной базы данных. В момент написания этого текста вариант базы данных «Firebird встроенная» относится к экспериментальным возможностям, поэтому он не будет обсуждаться здесь.

Примечание

При создании новой базы данных вы должны зарегистрировать её. Регистрация указывает расположение базы данных в одном из конфигурационных файлов LibreOffice. После этого вы сможете получить доступ к базе данных из Writer или Calc.

2. На втором шаге *Сохранить и выполнить*:

- a. Выберите вариант *Да*, зарегистрировать базу данных (установлен по умолчанию).
- b. Выберите, что сделать после сохранения базы данных. В этом примере мы выберем вариант *Открыть базу для редактирования*.



Регистрация и открытие базы данных

Примечание

Если вы хотите создать таблицу с помощью мастера сразу после создания новой базы данных, то отметьте галочкой пункт *Создать таблицы с помощью мастера таблиц* в дополнение к пункту *Открыть базу для редактирования*. Если вы сделаете это, то откроется главное окно базы данных с открытым окном мастера таблиц.

Мы рассмотрим главное окно базы данных перед разделом по использованию мастера таблиц.

3. Нажмите кнопку *Готово* внизу окна мастера. Откроется стандартное окно *Сохранить как*. Задайте имя вашей базе данных и сохраните файл.

Таким образом мы создали и сохранили новый файл базы данных, который открылся в главном окне программы Base. Если хотите, то можете закрыть его сейчас и открыть свой файл позже.

Открытие существующей базы данных

Файл с новой базой данных можно открыть заново несколькими способами. Например, откройте каталог, где сохранили базу данных, в любом файловом браузере и дважды нажмите по файлу с базой мышкой. Или нажмите правой кнопкой мыши на файле и выберите пункт *Открыть* в контекстном меню. Также для открытия файла можно использовать мастер создания баз данных, как описано ниже.

Использование мастера баз данных для открытия существующей базы данных

Под вариантом *Открыть файл существующей базы данных* на первой странице мастера баз данных, в раскрывающемся списке прописаны все базы, использованные в программе Base ранее. После создания первой базы данных, её имя появится в качестве значения по умолчанию в этом списке. После того, как вы создадите или откроете другую базу данных, в этом списке появятся другие соответствующие имена баз. Вы можете использовать этот список для открытия существующей базы данных.

1. Откройте мастер создания базы данных так, как вы это делали, когда создавали первую базу данных.
2. Выберите вариант *Открыть файл существующей базы данных*:
 - Из раскрывающегося списка *Последние* выберите имя ранее использованной базы данных.

или

- Нажмите кнопку *Открыть* ниже, найдите вашу базу данных и выберите её.

3. Нажмите кнопку *Готово*.

Третий вариант на первом шаге мастера баз данных используется для подключения к базам данных, которые были созданы в иных СУБД. Это текстовые базы данных, электронные таблицы, базы данных MySQL, PostgreSQL, Oracle или Access.

Совет

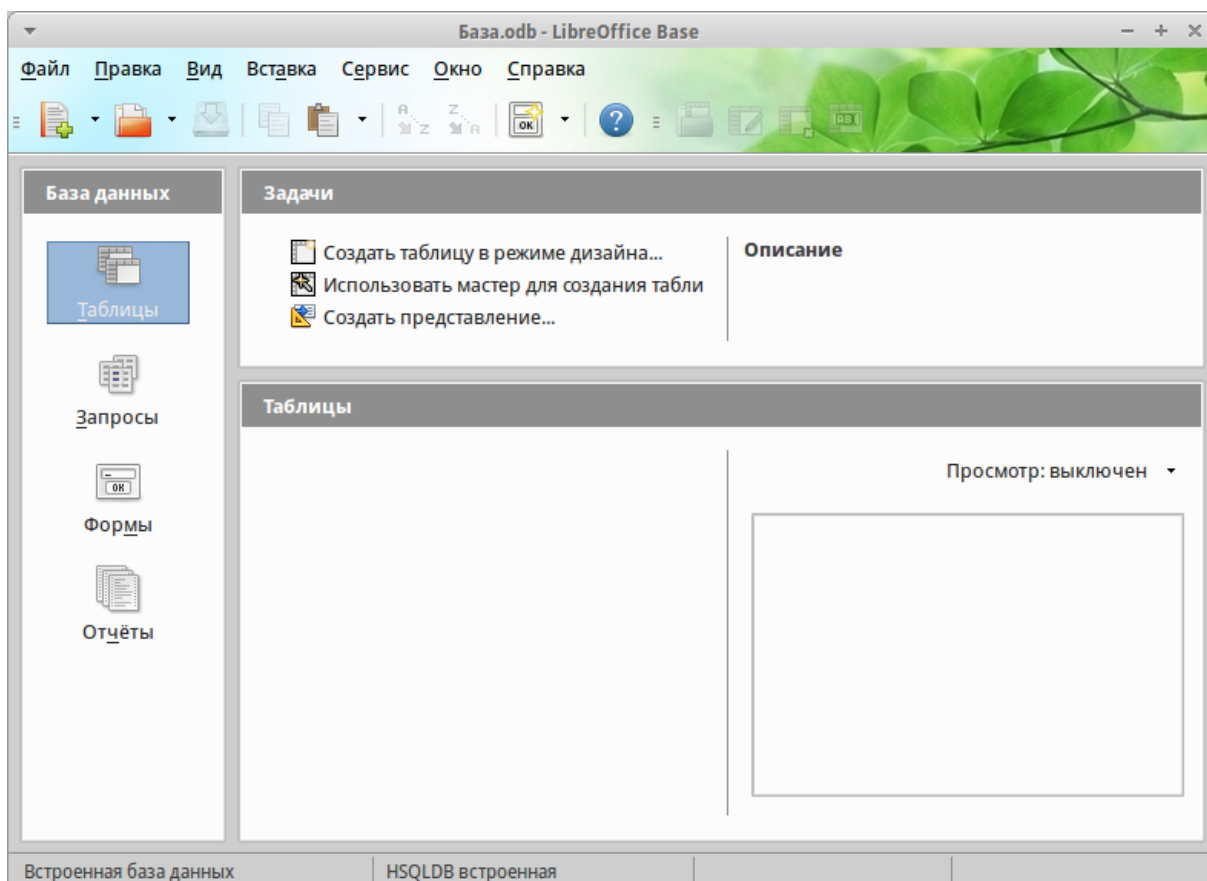
Такие базы данных и метод подключения к ним с помощью Base обсуждаются в *Главах 2 и 8 Руководства по Base*.

Главное окно Base и его части

Всё, что вы делаете с базой данных, начинается с главного окна программы Base. Для выполнения некоторых задач необходимо будет возвращаться к этому окну, поэтому его нужно хорошо изучить.

Главное окно открывается после того, как вы создали новую базу данных и сохранили её. Оно также открывается при открытии файла существующей базы данных.

Главное окно Base содержит три секции: *База данных*, *Задачи* и *Список*. Названия заголовков секций *База данных* и *Задачи* отображаются всегда. Название секции *Список* всегда будет отображаться иначе, в зависимости от того, какой значок выбран в разделе *База данных*.



Главное окно базы данных

Секция База данных

Эта секция состоит из колонки значков с левой стороны главного окна. Там находятся значки для каждой части базы данных. Первым шагом после открытия базы данных должен быть выбор, с какой частью базы данных вы будете работать. Ваш выбор повлияет на то, что будет отображаться в других секциях главного окна.

Секция Задачи

Эта секция главного окна содержит список задач, связанных с выбранным значком в секции *База данных*, которые могут быть выполнены. Когда в секции *База данных* выбран значок *Таблицы*, в секции *Задачи* появятся три пункта. Два из них помогут вам создать таблицу и один поможет вам создать представление. При выборе значка *Запросы*, в секции *Задачи* появятся три пункта, которые

помогут вам создать запрос. При выборе значка *Формы*, в секции *Задачи* появятся два пункта, которые помогут вам создать форму. При выборе значка *Отчёты*, в секции *Задачи* появятся два пункта, которые помогут вам создать отчет .

Правая сторона секции *Задачи* называется *Описание*. Эта область покажет вам информацию о каждой из задач. Нажмите любую из задач, чтобы увидеть её описание.

Примечание

В этой главе мы используем только мастера для создания таблиц, запросов, форм и отчетов. В *Руководстве по Base* подробно описаны остальные задачи и принципы работы Base.

Секция Список

Эта секция содержит список объектов для значка, выбранного в секции *База данных*. Нажмите на значок *Таблицы*, *Запросы*, *Формы* или *Отчеты* и в секции *Список* будет показан список таблиц, запросов, форм или отчетов соответственно. При этом названием для этой секции будет служить имя выбранного значка.

Множество разных операций может быть сделано над любым видимым пунктом списка, если нажать на нём правой кнопкой мыши и выбрать действие из появившегося контекстного меню. Например, стандартные действия *Копировать*, *Удалить*, *Переименовать*, *Изменить* и *Открыть*. Также контекстное меню содержит различные дополнительные команды, которые зависят от типа списка.

Справа от списка есть небольшая область с названием *Просмотр*, в котором показывается подробная информация о конкретном документе (таблица, запрос, форма или отчет), выбранном в списке. Тип просмотра задается из выпадающего списка, содержащего три варианта: *Выключен*, *Информация о документе* и *Документ*.

Если в секции *База данных* выбраны значки *Таблица* или *Запросы*, то в области *Просмотр* доступны только два варианта: *Нет* или *Документ*. Когда выбраны *Формы* или *Отчеты*, то доступны все три варианта. При варианте *Выключен* в области *Просмотр* ничего не отображается.

Выберите вариант *Информация о документе* для списка форм, чтобы увидеть, кто последним изменил форму, и когда она была изменена. Если форма никогда не изменялась, то никакой информации показано не будет.

Выберите вариант *Документ* для любого элемента списка и вы увидите снимок того, что вы выбрали. В большинстве случаев вы увидите только верхнюю левую часть элемента из-за ограничений по размеру. Для таблицы или запроса вы увидите только первые несколько столбцов и строк вместе с данными. Для

формы вы увидите её верхний левый угол (Если форма небольшая, то вы увидите её всю).

Совет

Просмотр документа не работает для отчетов. Вы увидите все три варианта в раскрывающемся списке, но вы не увидите ничего, если вы выберете вариант *Документ*.

Представления

Представление – это виртуальная таблица или встроенный в базу данных запрос, который вы можете создать, используя поля одной или более таблиц, уже созданных ранее. Представление позволит вам установить отношения между таблицами, используя выделенные поля, и увидеть результат. Структура представления записывается в специальный файл в одном каталоге с файлом базы данных. Запросы записываются в другом файле в другом месте.

Нажмите на значок *Таблицы* в секции *База данных*. Затем выберите в секции *Задачи* вариант *Создать представление*, откроется *Конструктор представлений*. Это диалоговое окно похоже на диалог *Конструктор запросов*. Обсуждение этих диалогов выходит за рамки данной главы. Оба они подробно обсуждаются в *Руководстве по Base. Глава 5 – Запросы*.

Объекты базы данных

Объектами базы данных являются таблицы, запросы, формы и отчеты. Чтобы сделать данные в базе данных полезными, при создании новой базы данных такие объекты также должны быть созданы.

В этой главе рассматривается использование мастеров для создания объектов. В качестве примера базы данных будет база «Мебель».

Таблицы: использование мастера для создания таблиц

Чтобы открыть мастер создания таблиц нажмите на значок *Таблицы* в секции *База данных*. В секции *Задачи* будут показаны три пункта. Выберите *Использовать мастер* для создания таблицы. Откроется первая страница мастера.

Мастер создания таблиц разделён на несколько шагов. Каждый шаг отображается на отдельной странице. В каждом шаге необходимо выполнить несколько действий. Основные шаги:

1. Выбор полей.
2. Выбор типа и формата полей.

3. Выбор первичного ключа.
4. Создание таблицы.

Примечание

На [Шаг 2: Выбор типа и формата полей](#) вы можете выбрать один из возможных типов поля, создать новые поля и установить их тип и формат, а также переименовывать поля.

Шаг 1: Выбор полей для вашей таблицы

Категория – две больших категории, которые определяют тип таблиц, которые вы можете использовать: *Деловые* и *Персональные*. Названия говорят сами за себя, выберите ту категорию, которая необходима для ваших целей.

Примеры таблиц – это выпадающий список таблиц, для каждой категории свой. Выберите таблицу из этого списка и её поля появятся в списке *Доступные поля*.

Используйте стрелки между списками полей, чтобы перемещать поля из списка *Доступные поля* в список *Выбранные поля* и обратно. Кнопка с одной стрелкой перемещает одно поле; кнопка с двойной стрелкой – все поля одновременно. Если вы хотите перемещать одновременно более одного поля (но не все поля), выделите нужные поля (нажимайте по ним мышкой с зажатой клавишей **Ctrl**), а затем нажмите кнопку с одной стрелкой.

Стрелки справа от списка *Выбранные поля* нужны для изменения порядка следования полей в списке *Выбранные поля*. Нажмите на поле, чтобы выделить его. Нажмите стрелку вверх, чтобы переместить его выше или нажмите стрелку вниз, чтобы переместить его вниз.

Выбор поля для таблицы

Мастер поможет создать таблицы для базы данных. После выбора категории и примера таблицы выберите поля, которые вы хотите включить в таблицу. Возможно включать поля из нескольких примеров таблиц.

Категория

Деловые Персональные

Примеры таблиц

Активы

Доступные поля

- Штрихкод
- Примечания
- ТекущаяСтоимость
- ДатаПокупки
- ДатаПродажи
- КодОтдела
- СрокАмортизации

Выбранные поля

- КодАктива
- КодКатегорииАктива
- Когда купили

1 2 3 4

Мастер создания таблиц. Шаг 1. Выбор полей

1 – Категории; **2** – Перемещение выбранных полей справа налево; **3** – Перемещение выбранных полей слева направо; **4** – Изменение порядка следования полей

Нажмите кнопку *Далее*, для перехода к *Шагу 2*.

Практическое упражнение: Создание таблицы «Мебель»

Откройте файл базы данных, в главном окне слева нажмите на значок *Таблицы*. В секции *Задачи* выберите пункт *Использовать мастер для создания таблицы*, чтобы открыть мастер.

Мастер содержит много предварительно настроенных вариантов таблиц, которые вы можете использовать в своей базе данных. Часть из них предназначены для коммерческих целей, а часть для личных. Первое, что нужно сделать, это просмотреть названия этих таблиц. Для этого вам необходимо выбрать вариант *Деловые* или *Персональные* и просмотреть выпадающий список таблиц под надписью *Примеры таблиц*. Отметьте для себя любые таблицы, которые покажутся вам полезными. Таким образом, если вы захотите использовать образец таблицы в своей базе данных, вы будете знать, где её искать.

Найдите в списке *Примеры таблиц* из категории *Персональные таблицы* с именем *ДомашнийИнвентарь* и выберите её.

Для этой таблицы доступны шестнадцать полей. Вы можете выбрать их все или только некоторые из них, в зависимости от информации, необходимой для базы данных.

Выбор используемых полей является частью планирования базы данных. Имеет смысл просмотреть доступные поля, чтобы увидеть, должны ли конкретные поля быть частью таблицы. Если вы считаете, что какое-либо из полей не нужно в базе данных, то не выбирайте его.

Что делать, если поле, которое нужно иметь в таблице, отсутствует в списке? Вы сможете создать его на следующем шаге мастера (шаг 2). Также можно выбрать похожее поле в шаге 1 и изменить его тип и формат в шаге 2.

Например: вы хотите включить в таблицу названия комнат, в которых есть мебель. Одно из 16 полей называется *КодКомнаты*. Вы можете выбрать это поле на 1 шаге. Тогда на странице шага 2 вы измените имя поля с *КодКомнаты* на *Комната*. Кроме того, вы можете изменить тип поля и его формат.

Для выполнения этого упражнения, выберите следующие поля: *ОцененнаяСтоимость*, *ДатаЗаказа*, *Описание*, *Застраховано*, *Элемент*, *Производитель*, *Модель*, *Примечания*, *МестоЗаказа*, *ПокупнаяЦена*, *КодКомнаты* и *КодИнвентаря*.

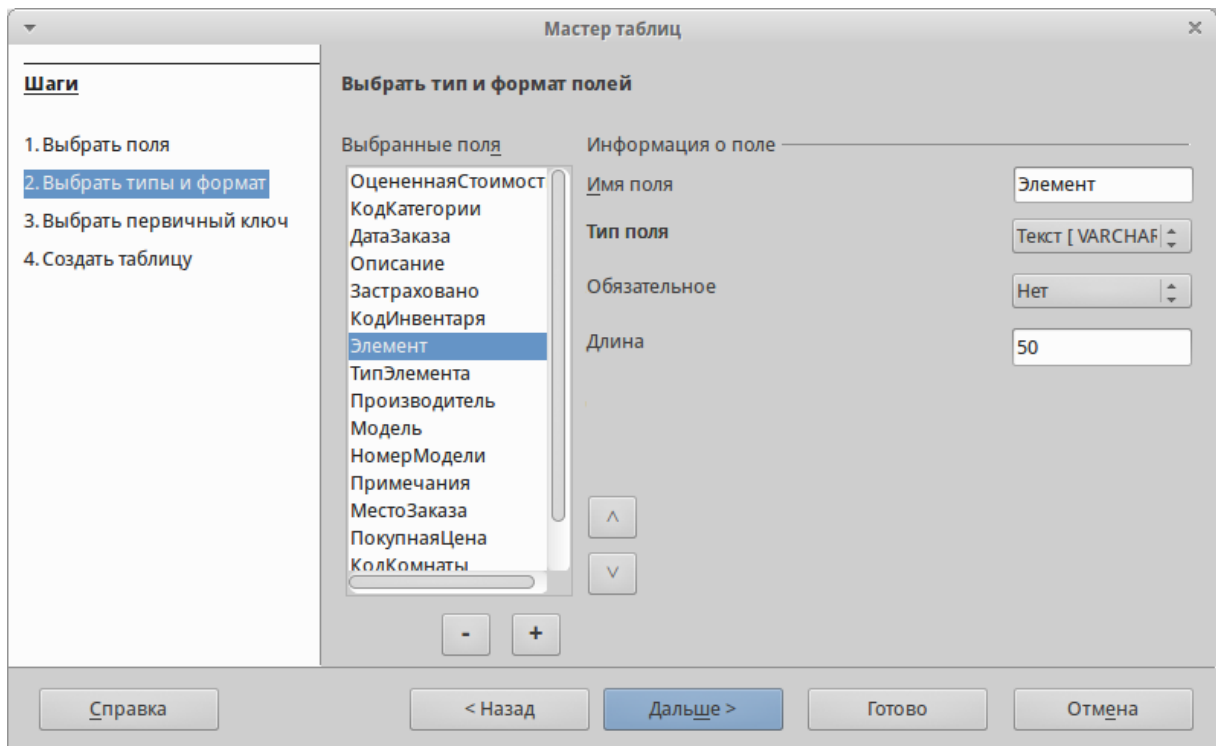
С помощью кнопок со стрелками переместите эти поля из списка *Доступные поля* в список *Выбранные поля*. Вы можете перемещать поля по одному или щелкать по нужным полям мышкой с зажатой клавишей **Ctrl**, чтобы выделить несколько полей и одновременно переместить их.

Сейчас начнётся следующая часть планирования. Подумайте о том, как вы хотите вводить данные в таблицу. Поля в настоящее время размещены в произвольном порядке. В таком же порядке нужно будет вводить данные. Вас это устраивает? Скорее всего, ответ будет отрицательным. Тогда какой порядок вас устроит? Возможно такой, какой показан на рисунке **Режимы отображения: одна страница, бок-о-бок, книга..** Такой порядок полей будет использоваться в нашем примере в остальных шагах мастера.

Чтобы завершить эту часть практического задания, нажмите кнопку *Дальше*. Упражнение будет продолжено в конце шага 2.

Шаг 2: Выбор типа и формата полей

Эта страница состоит из двух частей: список *Выбранные поля* и *Информация о поле*. Первая часть содержит тот же список полей, который был создан в предыдущем шаге мастера, включая заданный им там порядок. При выборе одного из полей в этом списке, информация о нём появится во второй части.



Отсортированный список выбранных полей и информация о поле «Элемент»

Список **Выбранные поля**

Есть несколько вещей, которые вы можете делать со списком *Выбранные поля*. В правом нижнем углу списка расположены две кнопки со стрелками. Ими можно изменять порядок полей в этом списке так же, как в первом шаге мастера.

Под списком расположены кнопка плюс (+) и кнопка минус (-). Используйте их, чтобы создать (+) новое поле или удалить (-) существующее поле.

Будьте осторожны при удалении поля. Если вы случайно удалите поле, которое не надо было удалять, вам придется использовать кнопку (+) плюс, чтобы добавить поле обратно в список. При этом вы должны будете заново задать информацию о поле вручную.

Информация о поле

Большинство типов полей, используемых в Base, аналогичны тем, которые используются в иных программах управления базами данных. Тем не менее, различные программы, скорее всего, имеют различные типы полей, которые не доступны в Base. Например, тип поля `MEDIUMINT`, который используется в MySQL. Его длина меньше, чем у типа Целое (Integer) и больше, чем у типа Короткое целое (Smallint), которые используются в Base.

Совет

Новичок ли вы в создании баз данных или уже знакомы с принципами

использования Base для их создания, вам всё равно придётся задавать *Информацию о поле* для всех ваших полей. По этой причине, все параметры полей, которые используются в Base описаны в *Руководстве по Base* в *Приложение I – Информация о полях Base*.

Параметр *Имя поля* позволяет вам изменить имя поля. Параметр *Тип поля* определяет основные характеристики поля, такие, как: текст, число, дата, время, логическое значение (например: да/нет, верно/неверно или мужчина/женщина) и «очень большие поля», включая изображения.

Используйте значение *Да* для параметра *Обязательное*, для любого поля, которое обязательно должно иметь значение. Проверьте заранее, что запись действительно должна всегда иметь некое значение, прежде, чем выбрать вариант *Да*.

Предупреждение

Если у вас в таблице есть поле, которое требует обязательную запись, то вы получите сообщение об ошибке, если не введёте туда значение. После появления ошибки вы не сможете вводить значения в другие поля, пока не введете значение в то поле, которое требует обязательную запись.

Параметр *Длина* определяет размер записи, который может поместиться в поле. Каждый из типов полей имеет конкретную максимальную длину. Крайне желательно проверять, какого максимального размера данные, которые будут заноситься в это поле, и задавать значение длины поля исходя из этого. При этом, возможно, потребуется изменить тип поля на позволяющий ввести больший размер данных, чем предустановленный тип. Опять же, тип поля, который имеет меньшую длину, может быть более подходящим.

Текстовые поля включают в себя типы *Текст (фикс.) [CHAR]*, *Текст [VARCHAR]*, *Текст [VARCHAR_IGNORECASE]* и *Памятка [LONGVARCHAR]*.

Тип поля *Текст (фикс.) [CHAR]* сохраняет записи фиксированной длины. Например, выбрана длина 10 и введённые значения были «кошка», «мышь» и «собака». Внутри программы они будут храниться в виде «кошка00000», «мышь000000» и «собака0000». Нули были добавлены программой, чтобы сделать длину каждого значения равной 10. Впоследствии, при чтении этого поля, нули игнорируются.

Тип поля *Текст [VARCHAR]* – является типом поля с переменной длиной. Сохраняются только реально введенные символы, длиной не больше заданного. Если этот тип поля будет иметь длину, равную 10, то приведённые выше примеры будут храниться в программе в виде «кошка», «мышь» и «собака» без всяких дополнительных символов. При использовании типа *VARCHAR* вместо *CHAR* используется меньше места для хранения данных. Это особенно полезно, когда значения поля могут значительно

изменяться по длине. Таким образом, значение длины, равное 50, может быть использовано для типа VARCHAR до тех пор, пока самое длинное вводимое значение меньше или равно 50. С длиной, равной 50, при типе поля VARCHAR, вышеприведенные примеры будут храниться, все ещё как, «кошка», «мышь» и «собака».

Выполните следующие действия, чтобы установить поля и их свойства для таблицы :

1. Чтобы проверить или изменить *Информацию о поле* для выделенного поля:
 - a. Нажмите по названию поля в списке *Выбранные поля*.
 - b. Сравните *Информацию о поле* с желаемой для этого поля.
 - c. Измените параметры, если это необходимо.
 - d. Повторите шаги 1a-1c для остальных полей.
2. Чтобы создать новое поле:
 - a. Нажмите на кнопку плюс (+).
 - b. Измените *Имя поля* на нужное вам.
 - c. Измените иную информацию в разделе *Информация о поле* на нужную.
 - d. Повторите шаги 2a-2c для добавляемых полей.
3. Чтобы удалить ненужные поля:
 - a. Нажмите по названию удаляемого поля в списке *Выбранные поля*.
 - b. Нажмите кнопку минус (-) внизу списка.
 - c. Повторите шаги 3a-3b, чтобы удалить все ненужные поля.

Практическое упражнение

В нашей таблице у поля *КодКомнаты* необходимо изменить имя, тип поля и длину. У полей *МестоЗаказа*, *ПокупнаяЦена*, *ДатаЗаказа* – изменить имя поля. Поля *ОцененнаяСтоимость* и *ПокупнаяЦена* требуют изменения числа десятичных знаков. Выполните изменения в следующем порядке: нажмите на поле в списке *Выбранные поля*, а затем измените часть *Информации о поле* так, как было сказано выше. Наконец, для поля *КодИнвентаря*, параметр *Автозначение* установите в значение *Да*. После внесения изменений используйте клавишу **Tab** или нажмите на одно из других полей в списке. Таким образом изменения будут сохранены.

Сначала нажмите на поле *КодКомнаты*. Измените *Имя поля* на *Комната*. Используйте клавишу **Tab**, чтобы переместиться на элемент *Тип поля*. Нажмите на выпадающий список, чтобы открыть его. Выберите тип **Текст [VARCHAR]**. И, наконец, задайте длину поля равной 50, если конечно у вас нет комнаты с именем длиннее, чем 50 символов. Нажмите клавишу **Tab** или нажмите на одну из других областей, чтобы сохранить изменения.

Нажмите на поле *МестоЗаказа*. Измените *Имя поля* на *МестоПокупки*.

Нажмите на поле *ПокупнаяЦена*. Измените *Имя поля* на *ЦенаПокупки*.

Нажмите на поле *ДатаЗаказа*. Измените *Имя поля* на *ДатаПокупки*.

Нажмите по полю *ОцененнаяСтоимость*. Измените значение параметра *Знаков после запятой* с 0 на 2. Теперь мы можем хранить значения в рублях и копейках. Для поля *ЦенаПокупки* аналогично измените значение параметра *Знаков после запятой* с 0 на 2.

Нажмите по полю *КодИнвентаря*. Измените значение параметра *Автозначение* с *Нет* на *Да*. Это делается для того, чтобы поле использовалось в качестве первичного ключа таблицы. (Смотрите *Шаг 3* ниже для получения дополнительной информации о первичных ключах).

Нажмите клавишу **Tab** или нажмите по другому полю для сохранения изменений.

Нажмите кнопку *Дальше*, чтобы перейти к шагу 3.

Примечание

При использовании мастера таблиц для создания таблицы, вам очень редко придется вносить какие-либо изменения в информацию о полях, так как предлагаемые значения, как правило соответствуют ожиданиям. Тем не менее, следует проверять значение элемента *Знаков после запятой* для полей, в которых будут храниться денежные суммы, так как мастер предлагает использовать по умолчанию значение 0 (то есть вы не сможете вводить копейки в суммах).

Шаг 3: Выбор первичного ключа

Прежде всего, что такое *первичный ключ* и зачем он нужен в таблице? Ключ состоит из одного или нескольких полей, которые имеют одну особенность: никакие два значения первичного ключа не являются одинаковыми. Целью первичного ключа является однозначная идентификация строк в таблице.

Для начинающих: любой первичный ключ используется в одном поле. Лучше, если это поле имеет тип *Целое [Integer]*, и параметр *Автозначение* для него установлен в значение *Да*. Если создать первичный ключ таким образом, то база данных будет автоматически присваивать значения этому полю, начиная с 0. Каждым новым значением для этого поля будет число, которое больше на какое-то значение (например на 1), чем предыдущее. Этим гарантируется уникальное значение в каждом поле.

Таблица состоит из строк и столбцов с данными. Если таблица содержит первичный ключ, то мы можем выбрать заданную строку, задав при поиске значение первичного ключа для этой строки. Каждый столбец таблицы содержит значения в конкретном поле. Поэтому, когда мы одновременно

указываем значение первичного ключа и имя поля, мы можем выбрать конкретную ячейку таблицы. Это то, что нам нужно, если мы хотим найти конкретную информацию, которая содержится в таблице: путь с указанием строки и столбца для нашего поиска (адресация).

Первичные ключи, содержащиеся в более, чем одном поле, требуют более бдительного отношения: две строки не должны совместно для пары полей иметь одинаковые значения. Рассмотрим таблицу ниже: две строки не идентичны, хотя *Поле1* имеет повторяющиеся значения, так же как и *Поле2*. Тем не менее, если рассматривать два поля совместно, то дублирования значений нет.

Некоторые типы таблиц не требуют наличия первичного ключа, но это может привести к потенциальным проблемам в случае, если вы что-то не учтёте. Вообще же, лучше всего иметь первичный ключ для каждой таблицы во избежание лишних проблем, тем более, что это не так сложно делается.

Пример значения первичного ключа из двух полей

Поле1	Поле2
1	0
0	1
0	0
1	1

Предупреждение

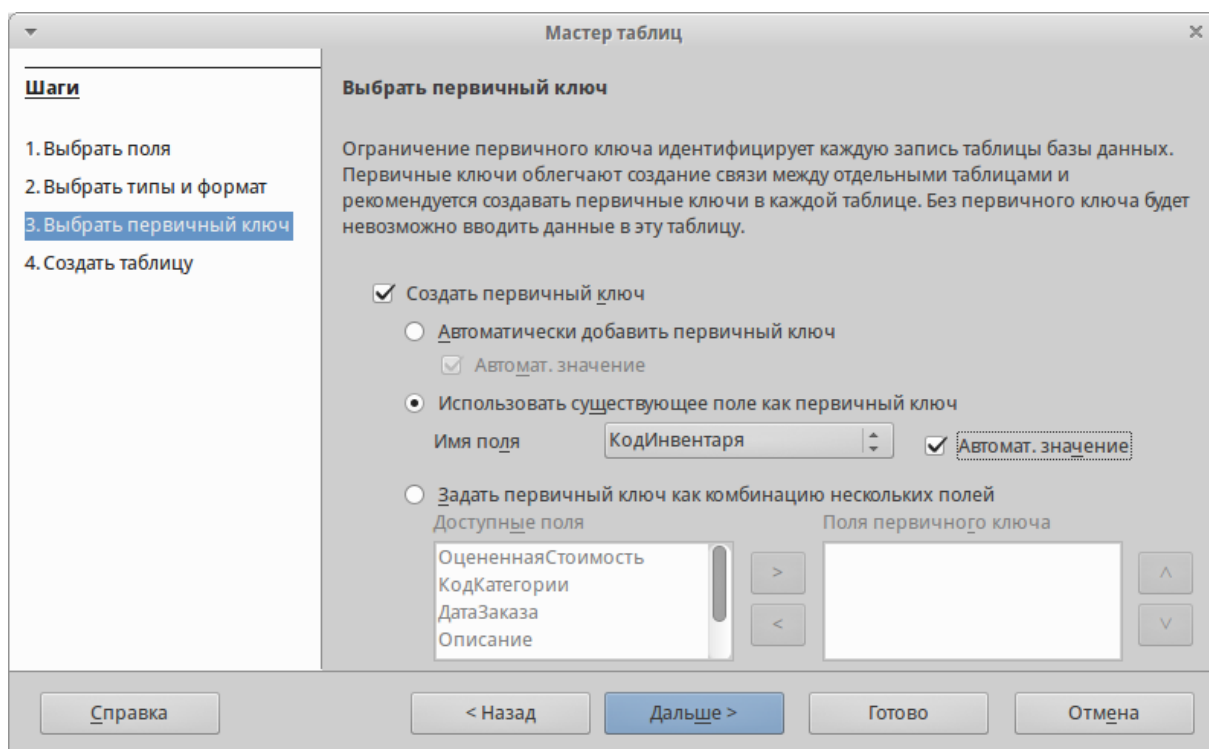
Если таблица создана без первичного ключа, вы не сможете вводить данные в таблицу. Это может быть неприятно. Прежде, чем вы сможете исправить эту ошибку, вы должны научиться использовать диалог *Конструктор таблиц*. (смотрите *Руководство по Base. Глава 3*). Поэтому при использовании мастера таблиц всегда проверяйте наличие первичного ключа в шаге 3: **Выбрать первичный ключ**.

После выбора опции *Создать* первичный ключ становятся доступны три варианта создания первичного ключа. Опция активна по умолчанию. У вас будет выбор из следующих вариантов: *Автоматически добавить первичный ключ*, *Использовать существующее поле как первичный ключ* и *Задать первичный ключ, как комбинацию нескольких полей*.

Первый вариант добавляет дополнительное поле «*ID*» в качестве первичного ключа таблицы с типом поля *Целое [Integer]*. Однако опция *Автомат.значение* установлена в значение *Нет*. Это означает, что необходимо будет вручную вводить каждое новое значение в поле «*ID*». Активируйте опцию, чтобы позволить Base самостоятельно задавать значения для первичного ключа.

Выбор второго варианта показан на рисунке ниже. *Имя поля* было выбрано из выпадающего списка с именами полей. Активируйте

опцию *Автомат.значение*, чтобы Base задавал значения для этого поля автоматически .



Мастер таблиц. Шаг 3

Третий вариант позволяет выбрать два или более полей в качестве полей первичного ключа. Если его выбрать, то можно будет выбрать поля из списка *Доступные поля* и переместить их в список *Поля первичного ключа*, используя стрелку направо между списками. Также можно будет изменять порядок полей в списке *Поля первичного ключа*, используя стрелки вверх или вниз справа от этого списка.

Если вы хотите поэкспериментировать с множественными полями первичного ключа, то выберите на первом шаге мастера таблиц категорию *Персональные* и образец таблицы *Адреса*. Выберите поля *Имя*, *Фамилия*, а также любые другие поля, которые вам нужны. После этого на третьем шаге выберите третий вариант *Задать первичный ключ*, как комбинацию нескольких полей. Переместите поля *Имя* и *Фамилия* из списка *Доступные поля* в список *Поля первичного ключа*. После добавления данных в эту таблицу убедитесь, что вы не используете одинаковую комбинацию имени и фамилии более одного раза.

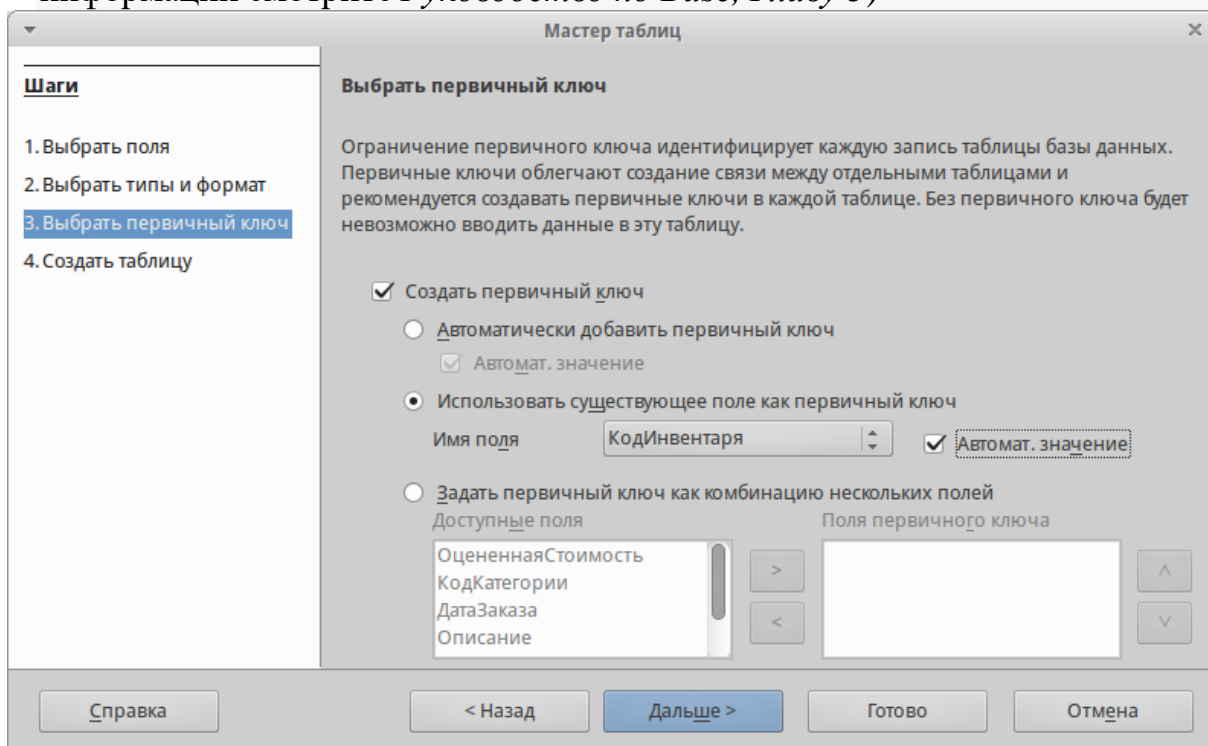
Практическое упражнение

Выберите поле *КодИнвентаря* в качестве первичного ключа таблицы и активируйте опцию *Автомат.значение*. Нажмите кнопку *Дальше*, чтобы перейти к шагу 4.

Шаг 4: Создание таблицы

Совет

Обычная практика присваивания имён полям или таблицам состоит в том, чтобы объединить два или несколько слов в одно слово (каждое слово при этом начинается с заглавной буквы). *ДомашнийИнвентарь* и *КодИнвентаря* – это два типичных примера. Использовать пробелы в именах, используемых в базе данных, можно, но такие имена могут потребовать в дальнейшем использования двойных кавычек при использовании языка SQL для создания, удаления или изменения таблиц или полей. (Для получения более подробной информации смотрите *Руководство по Base, Главу 3*)



Мастер таблиц. Шаг 4

Используйте эту страницу, чтобы задать название созданной таблице. Можно использовать предложенное мастером название, изменить его или придумать таблице своё название. Затем нужно выбрать из трёх вариантов того, что делать дальше: *Немедленно вставить данные*, *Модифицировать дизайн таблицы* или *Создать форму* на основе данной таблицы. После изменения названия таблицы выберите нужный вариант дальнейших действий и нажмите кнопку *Готово*, которая закроет мастер таблиц.

Если выбран первый вариант, то после нажатия на кнопку *Готово* откроется окно *Данные таблицы*. В нём можно добавлять данные в эту таблицу.

Если выбран второй вариант, то после нажатия на кнопку *Готово* откроется *Конструктор таблиц*. В нём можно изменить, удалить или добавить поля таблицы и их параметры. (Смотрите *Главу 3 – Таблицы в Руководстве по Base*)

Если выбран третий вариант, то после нажатия на кнопку *Готово* откроется

мастер создания форм. В нём вы сможете создать форму для вашей таблицы *Мебель* (смотрите раздел [Формы: использование Мастера создания форм](#)).

Практическое упражнение

Измените название вашей таблицы на *Мебель*. Затем выберите первый вариант действий из списка ниже *Немедленно вставить данные*. Нажмите кнопку *Готово*, чтобы закрыть мастер таблиц. Откроется окно *Данные таблицы* (рисунок [Строка состояния Writer](#)).

Ввод и удаление данных, используя окно *Данные таблицы*

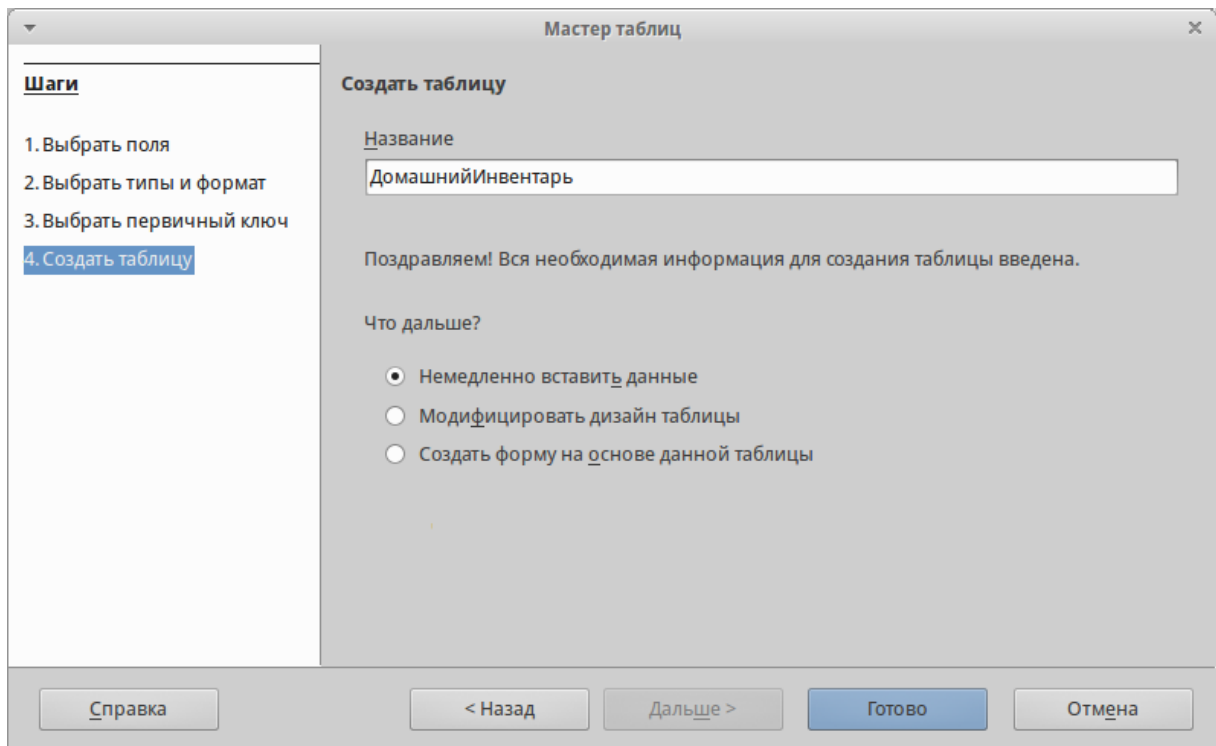
Данные могут быть введены в таблицу, используя окно *Данные таблицы*, или через форму, основанную на таблице. Здесь будут даны инструкции о вводе данных с помощью окна *Данные таблицы*. Ниже, в разделе [Формы: использование Мастера создания форм](#), будут даны инструкции, как вставить, изменить или удалить данные, используя форму.

Примечание

Этот раздел содержит базовые инструкции по добавлению и удалению данных в таблице. *Главы 3 и 4 Руководства по Base* описывают этот вопрос более подробно.

Чтобы иметь возможность вставлять или удалять данные в окне *Данные таблицы*, оно должно быть открыто. Это происходит автоматически при выборе варианта *Немедленно вставить данные* в мастере на последнем шаге создания таблицы. Или вы можете открыть любую таблицу в любой момент из главного окна базы данных, дважды нажав на нужной таблице в списке, чтобы отредактировать данные.

1. Нажмите по значку *Таблицы* в секции *База данных* главного окна программы.
2. Нажмите правой кнопкой мыши по имени нужной таблицы в списке таблиц.
3. Нажмите в контекстном меню пункт *Открыть*. Откроется окно *Данные таблицы*.



Окно *Данные таблицы*

Ввод данных в таблицу

Если таблица только создана, то она не содержит никаких данных, как это показано на рисунке выше. Она состоит из одной строки и нескольких столбцов, по одному для каждого поля, содержащегося в таблице (Таблица на рисунке разделена на две части из-за ширины таблицы). Заголовок каждого столбца – это имя поля.

Перед вводом данных в ячейки вспомним несколько фактов о строках и столбцах таблицы. Все данные, содержащиеся в одной строке относятся к одному элементу. Все данные, содержащиеся в одном столбце относятся к области, название которой вынесено в заголовок этого столбца.

Например, каждая строка на рисунке выше, содержит данные об отдельной единице, скажем, диване, который находится в гостиной. Данными о диване, вводимые в эту строку являются: его имя – поле *Элемент*; его местонахождение – поле *Комната*; информация о нём – поле *Описание*; кто его изготовил – поле *Производитель*; его модель – поле *Модель*; где он куплен – поле *МестоПокупки*; когда он куплен – *ДатаПокупки*; его цена при покупке – поле *ЦенаПокупки*; его стоимость в настоящее время – поле *ОцененнаяСтоимость*; факт страхования – поле *Застраховано*; другая информация – поле *Примечания*; значение первичного ключа – *КодИнвентаря*.

В качестве ещё одного примера возьмём ещё один диван. Это старый диван, который находится в другой комнате. Строка данных, которая относится к этому дивану будет иметь ряд отличий от строки, которая относится к первому

дивану (данные в поле *ДатаПокупки* будут являться одним из таких различий).

Из-за этих различий, необходимо позаботиться при вводе данных о том, чтобы все данные, вводимые в данной строке относились к одному элементу. В противном случае, данные, получаемые при запросе из таблицы будут содержать ошибки.

Теперь рассмотрим столбцы наших двух примеров о диванах. Столбец *Элемент* содержит в себе названия предметов. Столбец *Комната* содержит название места расположения предмета. Столбец *Описание* содержит некоторую информацию о них ... столбец *КодИнвентаря* содержит уникальные идентификационные номера.

Все эти нюансы должны быть учтены при заполнении таблицы данными. Данные, введенные в таблицу, должны быть размещены в правильных столбцах. В противном случае в результате обработки данных вы получите ошибки (не ошибки программы, а ошибки, связанные с неверным вводом данных!).

При открытии окна *Данные таблицы* курсор находится в первой ячейке строки. Окно готово для ввода данных в эту ячейку. После того, как вы ввели данные в ячейку, можете переместить курсор в другие ячейки, для ввода других данных.

Ввод данных в этом окне похож на ввод данных в ячейки электронной таблицы Calc. Курсор можно перемещать из одной ячейки в другую, используя клавиши курсора (в электронных таблицах, иногда нужно использовать клавишу Enter). Ввод данных только помещает его в ячейку, в которой находится курсор.

Есть важные различия. Ввод данных в ячейки таблицы базы данных не вводит данные в таблицу, как это происходит в электронной таблице Calc. Данные только отображаются в ячейке. Чтобы ввести их в ячейку, курсор должен быть перемещен нажатием на клавишу Enter, либо с помощью клавиш курсора. Наконец, использование клавиши Enter, когда курсор находится в последнем столбце строки, переместит курсор в первый столбец следующей строки.

Ввод данных в пустую таблицу: (курсор расположен в первом столбце первой строки)

1. Первая строка:

- a. Введите данные в первую ячейку.
- b. Переместите курсор в следующую ячейку. (Используйте клавишу Enter или клавишу курсора со стрелкой вправо)
- c. Повторите шаги a) и b) пока курсор не окажется в последней ячейке строки.
- d. Введите данные в последнюю ячейку.
- e. Переместите курсор в первую ячейку второй строки, нажав клавишу Enter.

2. Последующие строки: Повторяйте шаги а) – е) пункта 1).
3. Закройте окно *Данные таблицы*, появится диалог подтверждения сохранения изменений, подтвердите сохранение данных.

Ввод данных в любую пустую ячейку таблицы очень похож на описанные выше действия.

1. Используйте мышь, чтобы поместить курсор в пустую ячейку.
2. Введите в неё данные.
3. Переместите курсор в следующую ячейку (Используйте клавишу **Enter** или клавишу курсора со стрелкой вправо)
4. Закройте окно *Данные таблицы*, появится диалог подтверждения сохранения изменений, подтвердите сохранение данных.

При создании списков данных, которые вы хотите ввести в таблицу, вы должны рассмотреть формат списка. Рассмотрим, какие из следующих двух списков проще в использовании:

Это список данных, разделенных запятыми и точками с запятой:

Элемент;Комната;Описание;СерийныйНомер;Производитель;НомерМодели
 Клавиатура;Бухгалтерия;;1426894123;Cisao;СТК-720 НР
 Plavilion;Компьютерная;Компьютер;KQ946AA#AB;Heplett Wackard;a6503f

Приведенная ниже таблица показывает те же данные в табличном формате:

Элемент	Комната	Описание	СерийныйНомер	Производитель	НомерМодели
Клавиатура	Бухгалтерия		1426894123	Cisao	СТК-720
НР Plavilion	Компьютерная	Компьютер	KQ946AA#AB	Heplett Wackard	a6503f

Практическое упражнение

Введите эти данные в таблицу *Мебель*:

Пример данных (первые 5 полей)

Элемент	Комната	Описание	Производитель	Модель
Диван	Гостиная	Зелёный диван	ООО «Суровый диванчег»	«Челябинск»
Диван	Детская	Бежевый диван-книжка	ООО «Милая мебель»	«Толстая хрюшка»
Стол	Кухня	Кухонный стол круглый	ООО «Кухни и спальни»	«Классика»
Шкаф	Детская	Платяной шкаф с полками	ИП Обдиралов	«Зайка»

Пример данных (первые 5 полей)

Элемент	Комната	Описание	Производитель	Модель
Шкаф	Прихожая	Шкаф с зеркалами	ООО «Мебельщик-3»	«Красота-24»

Пример данных (последние 7 полей)

МестоПокупки	ДатаПокупки	ОценённаяСтоимость	ЦенаПокупки	Застраховано	Примечание	КодИнвентаря
Магазин «Сифон и Борода»	12.01.2012	4500,00	6500,00	Нет	Ещё вполне годный диван	0
Магазин «Пушистики»	30.10.2013	6700,00	7900,00	Нет	Диван-няшка	1
ИП Обдиралов	18.05.2010	6900,00	10000,00	Нет		2
ИП Обдиралов	30.10.2013	9000,00	12000,00	Нет	Куча места	3
Магазин «Сифон и Борода»	23.04.2011	7850,00	9500,00	Нет	Ляпота	4

Изменение данных в ячейке таблицы

Изменение данных производится в пять шагов:

1. Откройте таблицу.
2. Нажмите мышкой по нужной ячейке.
3. Измените данные на нужные (так же, как вы редактируете обычный текст).
4. Сохраните таблицу.
5. Закройте таблицу.

Удаление данных из таблицы

Удаление данных может быть двух типов: удаление данных только из одной ячейки или удаление данных из всех ячеек строки. Удаление данных из одной ячейки выполняется так же, как изменения данных в одной ячейке.

Удаление всех данных в строке таблицы выполняется в несколько шагов:

1. Откройте таблицу.
2. Нажмите по серому полю слева от строки с ненужными данными, которые нужно удалить.
3. Нажмите правой кнопкой мыши по серому полю и выберите в контекстном меню пункт *Удалить строку*. У вас запросят подтверждение удаления.
4. Нажмите *Да*, если вы действительно хотите удалить эту строку с данными. Нажмите *Нет*, если не хотите удалять.
5. Сохраните таблицу.
6. Закройте таблицу.

Предупреждение

Будьте очень внимательны при удалении строк с данными!

Если удалить не ту строку, то придется повторно вводить данные. Это займет какое-то время и будет не очень страшно, если есть где-нибудь копия данных.

Если копии данных нигде нет и нельзя получить данные где-то еще, данные будут навсегда утеряны!