

**НАЦІОНАЛЬНА АКАДЕМІЯ УПРАВЛІННЯ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ**

**МЕТОДИЧНІ МАТЕРІАЛИ
щодо виконання курсового проекту
з курсу “Основи баз даних та баз знань”**

(для студентів заочної форми навчання)

Київ - 2009

Методичні матеріали щодо виконання курсового проекту з курсу “Основи баз даних та баз знань” (для студентів заочної форми навчання) / Укл. Баклан І.В. (мова рос.) - К.: НАУ, 62 с.

Цей посібник має теоретичний та практичний матеріал, який дозволяє отримати уяву про роботу в СКБД OpenOffice.org Base. У посібнику викладені основні поняття баз даних. Розглянуті властивості та характеристики інтерфейсу та інструментарію СКБД OpenOffice.org Base. Дається матеріал для самостійного контролю.

Посібник розрахований на студентів заочної форми навчання спеціальності “Комп'ютерні науки”.

Оглавление

Предисловие	4
Введение	6
Глава 1. Основные понятия баз данных	11
1.1. Классификация баз данных.....	11
1.2. Объекты базы данных.....	15
Глава 2. Основы работы с OpenOffice.org Base	18
2.1. Этапы проектирования базы данных в OpenOffice.org Base.....	18
2.2. Основные принципы работы OpenOffice.org Base	19
2.3. Создание базы данных в OpenOffice.org Base.....	22
2.4. Открытие базы данных.....	24
2.5. Создание новой таблицы.....	25
2.6. Связи.....	31
2.7. Создание запросов.....	34
2.8. Создание форм.....	36
2.9. Создание отчетов.....	38
2.10. Пример создания базы данных	40
Вопросы для самоконтроля и задания для самостоятельного выполнения	51
Глоссарий	59
Список литературы	62

Предисловие

Одним из требований, предъявляемых к современным специалистам в любой сфере нашей жизни, является умение использовать передовые информационные технологии в профессиональной деятельности. Подготовка специалиста в области информационно-коммуникационных технологий должна включать как овладение теоретическими знаниями, так и получение практических навыков работы с современным программным обеспечением.

Сфера применения информационных технологий в системе образования очень обширна и включает в себя различные аспекты, от создания простейших текстовых документов до планирования и управления образовательным учреждением и системой образования в целом.

Данное учебное пособие предназначено для самостоятельного изучения. Прежде всего оно рассчитано на работников системы образования, но может использоваться при обучении студентов вузов, в системе повышения квалификации.

Учебное пособие написано на основе опыта преподавания информационных технологий в вузе и системе повышения квалификации. В ходе изучения пособия читатель знакомится с теоретическими знаниями по базам данных, получает практические навыки работы в СУБД, навыки по созданию баз данных, подготовки отчетов, созданию запросов и т.п.

Пособие состоит из введения, двух глав, глоссария, вопросов и заданий для самостоятельного выполнения, списка литературы. Во введении рассматриваются основные понятия баз данных, современные направления работы с данными, приводятся примеры основных СУБД.

Первая глава рассматривает теоретические основы баз данных. Здесь даются основные определения, приводятся классификации баз данных.

Во второй главе описываются основные принципы работы с СУБД OpenOffice.org Base. Рассматриваются вопросы создания таблиц, связей. На примерах показывается создание форм, отчетов, запросов. В заключительной части второй главы рассматривается пример пошагового создания базы данных.

Пособие содержит большое количество иллюстративного материала. В заключении приводится список контрольных вопросов, для закрепления практических навыков рекомендуется выполнить практические задания.

Неотъемлемой частью пособия является глоссарий, содержащий список терминов, список литературы, которую рекомендуется изучить.

Материалы могут быть использованы студентами направления “Компьютерные науки” специальности “Интеллектуальные системы принятия решений” при работе над курсовым проектом.

Введение

База данных — важнейшая составная часть информационных систем. Информационные системы предназначены для хранения и обработки больших объемов информации. Изначально такие системы существовали в письменном виде. Для этого использовались различные картотеки, папки, журналы, библиотечные каталоги и т.п. Базы данных — это реализованная с помощью компьютера информационная структура (модель), отражающая состояния объектов и их отношения [1].

В общем смысле термин «база данных» можно применить к любой совокупности связанной информации, объединенной вместе по определенному признаку. Примерами баз данных могут служить адресная книга, картотека, словарь, энциклопедия или кафедральный журнал. При этом в качестве базы данных рассматривается только набор данных, организованных определенным образом.

База данных предполагает наличие комплекса программных средств, обслуживающих эту базу данных и позволяющих использовать содержащуюся в ней информацию. Такие комплексы программ называются СУБД (*системы управления базой данных*). Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнение ее содержимым, редактирование содержимого и визуализации информации. Под *визуализацией информации* базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройства вывода или передачи по каналам связи.

В мире существует множество систем управления базами данных. Несмотря на то, что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий.

Основные функции СУБД:

1. **Определение данных** — определить, какая именно информация будет храниться в базе данных, задать свойства данных, их тип, а также указать, как эти данные связаны между собой.

2. **Обработка данных** — данные могут обрабатываться самыми разными способами. Можно выбирать любые поля, фильтровать и сортировать данные. Можно объединять данные со связанной информацией, вычислять итоговые значения.

3. **Управление данными** — можно указать, кому разрешено знакомиться с данными, корректировать их или добавлять новую информацию. Можно определить правило коллективного доступа.

СУБД поддерживают один из возможных типов моделей данных — сетевую, иерархическую или реляционную, которые являются важнейшими признаками классификации СУБД.

Основными средствами СУБД являются:

- средства задания структуры данных;
- средства конструирования экранных форм, предназначенных для ввода данных, просмотра и обработки в диалоговом режиме;
- средства создания запросов для выборки данных при заданных условиях, а также выполнения операций по их обработке;
- средства создания отчетов из базы данных для вывода на печать результатов обработки в удобном для пользователя виде;
- языковые средства — макросы, встроенный алгоритмический язык, язык запросов и т.п., которые используются для реализации нестандартных алгоритмов обработки данных, а также процедур обработки событий в задачах пользователя;
- средства создания приложений пользователя, позволяющие объединять различные операции работы с базой данных в единый технологический процесс.

Архитектура СУБД состоит из двух основных компонентов: языка описания данных и языка манипулирования данными. Язык описания данных (ЯОД) — средства описания данных в БД и связей между ними. Средствами этого языка описывается структура БД, форматы записей, пароли, защищающие данные.

Язык манипулирования данными (ЯМД) — язык для выполнения операций над данными, позволяющий менять их строение. Для разных СУБД реализация этих уровней языков может быть различной. В одних случаях ЯОД и ЯМД требует составления пользователем программы полностью «вручную», в других (что отражает современную тенденцию) СУБД присутствуют средства визуальной (зримой, наглядной) разработки программ. Для этого в современных СУБД имеются редакторы экранных форм, отчетов. «Кирпичиками» (инструментами) таких редакторов являются поля различных видов (поля ввода, поля вывода, вычисляемые поля), процедуры обработки различных типов (формы ввода, таблицы, отчеты, запросы). На основании созданных пользователем объектов программы-генераторы формируют программный код на языке конкретной машины или на промежуточном языке.

Обычно с базами данных работают две категории пользователей. Первая категория — проектировщики. Их задача состоит в разработке структуры таблиц базы данных и согласовании ее с заказчиком. Вторая категория исполнителей, работающих с базами данных, —

пользователи. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием.

Соответственно СУБД имеет два режима работы: *проектировочный* и *пользовательский*. Первый режим предназначен для создания или изменения структуры базы и ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

Выбор СУБД для практического приложения пользователем определяется многими факторами, к которым относятся:

- имеющееся техническое и базовое программное обеспечение, их конфигурация, оперативная и дисковая память;
- потребности разрабатываемых приложений пользователя;
- тип поддерживаемой модели данных, специфика предметной области и т.п.;
- требования к производительности при обработке данных;
- наличие в СУБД необходимых функциональных средств;
- наличие русифицированной версии СУБД;
- уровень квалификации пользователей и наличие в СУБД диалоговых средств разработки и работы с базой данных.

Кратко рассмотрим характеристики наиболее известных СУБД.

Paradox

Paradox был разработан компанией Ansa Software. В конце 80-х — начале 90-х годов Paradox, принадлежавший тогда компании Borland International, был весьма популярной СУБД, в том числе и в нашей стране, где он одно время занимал устойчивые позиции на рынке средств разработки настольных приложений с базами данных.

Принцип хранения данных в Paradox сходен с принципами хранения данных в dBase. Однако, в отличие от dBase, формат данных Paradox не является открытым, поэтому для доступа к данным этого формата требуются специальные библиотеки.

Отметим, однако, что отсутствие «открытости» формата данных имеет и свои достоинства. Так как в этой ситуации доступ к данным осуществляется только с помощью «знающих» этот формат библиотек, простое редактирование подобных данных по сравнению с данными открытых форматов типа dBase существенно затруднено. В этом случае возможны такие недоступные при использовании «открытых» форматов данных сервисы, как защита таблиц и отдельных полей паролем, хранение некоторых правил ссылочной целостности в самих таблицах — все эти сервисы предоставляются Paradox, начиная с первых версий этой СУБД.

Microsoft FoxPro и Visual FoxPro

FoxPro ведет свое происхождение от настольной СУБД FoxBase фирмы Fox Software. По сравнению с аналогичными версиями dBase, FoxBase и более поздняя версия этого продукта, получившая название FoxPro, предоставляли своим пользователям несколько более широкие возможности, такие, как использование деловой графики, генерация кода приложений, автоматическая генерация документации к приложениям и т.д.

Впоследствии этот продукт был приобретен компанией Microsoft. Его последние версии (начиная с версии 3.0, выпущенной в 1995 г.) получили название Visual FoxPro. В состав Visual FoxPro в течение нескольких последних лет входят средства переноса данных FoxPro в SQL Server и средства доступа к данным этого сервера из Visual FoxPro и созданных с его помощью приложений.

Одна из последних версий этого продукта — Visual FoxPro 6.0 — доступна и отдельно, и как составная часть Microsoft Visual Studio 6.0. Отличительной особенностью этой настольной СУБД от двух рассмотренных выше является интеграция этого продукта с технологиями Microsoft, в частности поддержка COM (Component Object Model — компонентная объектная модель, являющаяся основой функционирования 32-разрядных версий Windows и организации распределенных вычислений в этой операционной системе), интеграция с Microsoft SQL Server, возможности создания распределенных приложений, основанных на концепции Windows DNA (Distributed interNet Applications).

Microsoft Access

Первая версия СУБД Access появилась в начале 90-х годов. Это была первая настольная реляционная СУБД для 16-разрядной версии Windows. Популярность Access значительно возросла после включения этой СУБД в состав Microsoft Office.

В отличие от Visual FoxPro, фактически превратившегося в средство разработки приложений, Access ориентирован в первую очередь на пользователей Microsoft Office, в том числе и незнакомых с программированием. Это, в частности, проявилось в том, что вся информация, относящаяся к конкретной базе данных, а именно таблицы, индексы (естественно, поддерживаемые), правила ссылочной целостности, бизнес-правила, список пользователей, а также формы и отчеты хранятся в одном файле, что в целом удобно для начинающих пользователей.

Поддержка COM в Access выражается в возможности использовать элементы управления ActiveX в формах и Web-страницах,

созданных с помощью Access. В отличие от Visual FoxPro создание COM-серверов с помощью Access не предполагается.

Microsoft Access может быть использован, с одной стороны, в качестве настольной СУБД и составной части офисного пакета, а с другой стороны, в качестве клиента Microsoft SQL Server, позволяющего осуществлять его администрирование, манипуляцию его данными и создание приложений для этого сервера.

В Microsoft Access пользователь может выбрать, какой механизм доступа к данным следует применять: Microsoft Jet — стандартный набор библиотек доступа к данным или MSDE (в этом случае управление базой данных осуществляется с помощью отдельного процесса). Возможно преобразование имеющихся баз данных Access в базу данных MSDE из среды разработки Access.

OpenOffice.org Base

OpenOffice.org Base является некоммерческим аналогом СУБД Microsoft Access. Сама программа очень проста и приятна в использовании. Но за всей ее простотой есть много функций. Как уже сложилось у разработчиков OpenOffice.org, заголовки меню всех программ полностью повторяют их аналоги от Microsoft.

Мастер отчетов отображает нужный вам отчет в приложении Writer. Там же можно вручную отредактировать этот отчет, чего не может обеспечить Access. Следует учитывать, что при всей схожести OpenOffice.org Base не совместим с Microsoft Access.

Для домашнего использования или для использования программы на малых предприятиях OpenOffice.org Base — лучший бесплатный помощник. В программе небольшое количество функций, но самое основное и необходимое в ней есть. Поэтому с уверенностью можно сказать, что OpenOffice.org Base даже сегодня может составить здоровую конкуренцию Microsoft Access.

В данном учебном пособии мы рассмотрим именно приложение OpenOffice.org Base, так как создание баз данных в этом приложении не требует знания каких-либо языков программирования, а также, изучив OpenOffice.org Base, вы легко сможете освоить Microsoft Access и аналогичные ему продукты.

Глава 1. Основные понятия баз данных

1.1. Классификация баз данных

Рассмотрим классификацию баз данных. По технологии обработки данных базы данных подразделяются на **централизованные** и **распределенные**. **Централизованная база данных** хранится в памяти одной вычислительной системы. Эта вычислительная система может быть мэйнфреймом — тогда доступ к ней организуется с использованием терминалов — или файловым сервером локальной сети ПК. **Распределенная база данных** состоит из нескольких, возможно, пересекающихся или даже дублирующих друг друга частей, которые хранятся в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

База данных может быть в монопольном распоряжении пользователя одного персонального компьютера. В этом случае она размещается только на дисках данного компьютера и к информационной базе не обеспечивается одновременный доступ нескольких пользователей. При наличии сети персональных компьютеров открывается возможность хранить и использовать централизованные базы данных, размещаемые на машине-сервере, в многопользовательском режиме. Это позволяет классифицировать базы данных по способу доступа к данным. Согласно этой классификации, базы данных делятся на **базы данных с локальным доступом** и **базы данных с сетевым доступом**. Для всех современных баз данных можно организовать сетевой доступ с многопользовательским режимом работы.

Существуют разные подходы к организации баз данных. **Иерархические базы данных** — в основе данной модели лежит иерархическая модель данных. В этой модели имеется один главный объект и остальные — подчиненные — объекты, находящиеся на разных уровнях иерархии. Взаимосвязи объектов образуют иерархическое дерево с одним корневым объектом. Иерархическая БД состоит из упорядоченного набора нескольких экземпляров одного типа дерева. Автоматически поддерживается целостность ссылок между предками и потомками. Основное правило: никакой потомок не может существовать без своего родителя.

Другой подход к организации баз данных — **сетевые базы данных**. Сетевой подход к организации данных является расширением иерархического. В иерархических структурах запись-потомок должна

иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков.

К современным базам данных относят **реляционные системы**. Реляционные системы далеко не сразу получили широкое распространение. В то время как основные теоретические результаты в этой области были получены еще в 70-х годах и тогда же появились первые прототипы реляционных СУБД, долгое время считалось невозможным добиться эффективной реализации таких систем. Однако постепенное накопление методов и алгоритмов организации реляционных баз данных и управления ими привели к тому, что уже в середине 80-х годов реляционные системы практически вытеснили с мирового рынка ранние СУБД. Реляционная модель данных основывается на математических принципах, вытекающих непосредственно из теории множеств и логики предикатов. Эти принципы впервые были применены в области моделирования данных в конце 60-х годов доктором Е.Ф. Коддом, в то время работавшим в IBM, а впервые опубликованы в 1970 г.

Реляционные базы данных характеризуются тем, что информация в них хранится в одной или нескольких таблицах. Между таблицами могут существовать связи по ключевым значениям. При дальнейшем изложении под базой данных мы будем понимать реляционную компьютерную базу данных.

Каждая таблица состоит из строк и столбцов, которые в компьютерной базе данных называются записями и полями соответственно. Каждая запись содержит информацию об отдельном объекте системы: одном ученике в школе, одной книге в библиотеке и т.п. А каждое поле — это определенная характеристика (свойство, атрибут) объектов: фамилия ученика, год рождения, название книги, автор книги и т.п. Поля таблицы должны иметь несовпадающие имена. Каждое поле таблицы имеет определенный тип. Тип — это множество значений, которые поле может принимать, и множество операций, которые можно выполнять над этими значениями. Существуют четыре основных типа для полей баз данных: символьный, числовой, логический и дата. Например, для полей таблицы «Библиотека» могут быть установлены следующие типы (табл. 1):

Таблица 1

Поле	Тип
Автор, название, издательство	Символьный
Инв_номер, год_изд	Числовой
Дата_поступления	Дата

Чтобы автоматизировать процесс нужных записей в таблицах баз данных, необходимо указать поле или несколько полей, значения которых позволяют однозначно выбрать из всей совокупности записей в таблице именно ту запись, которая интересует запрашивающего информацию. Такое поле или группа полей называется **первичным ключом** или просто ключом таблицы. Значение первичного ключа должно быть уникальным. Для каждой таблицы реляционной базы данных должен быть определен первичный ключ. Например, в библиотечной базе данных таким ключом может быть выбран инвентарный номер книги, который не может совпадать у разных книг.

Первичный ключ не допускает значение Null. Если первичный ключ состоит из одного поля, он считается простым. В противном случае, то есть если ключ образован из нескольких полей, его называют составным.

Важным требованием, предъявляемым к таблицам реляционной модели, является нормализация данных, представленных таблицей. Первично нормализованная таблица содержит строки, в которых для каждого атрибута может быть только одно значение. Это соответствует требованиям недопустимости множественных и повторяющихся структур данных. Ненормализованной таблице обычно соответствует одна или несколько нормализованных таблиц.

Понятие нормальной формы было введено Эдгаром Коддом при создании реляционной модели баз данных. Основное назначение нормальных форм — приведение структуры базы данных к виду, обеспечивающему минимальную избыточность. Устранение избыточности производится за счет декомпозиции отношений (таблиц) таким образом, чтобы свести к минимуму функциональные зависимости между их атрибутами (полями).

Нормализация может применяться к таблице, первоначально отвечающей следующим требованиям:

- Таблица содержит нуль или более записей.
- Все записи таблицы имеют одно и то же множество полей, причем одноименные поля относятся к одинаковым типам данных.
- Таблица не может содержать двух полностью идентичных записей.

Приведем два примера первых двух нормальных форм. Первая нормальная форма (1NF). Таблица находится в первой нормальной форме, если каждый ее атрибут атомарен и все строки различны. Под выражением «атрибут атомарен» понимается, что атрибут может содержать только одно значение. Таким образом, не соответствуют 1NF таблицы, в полях которых могут храниться списки значений. Для приведения таблицы к 1NF обычно требуется разбить таблицу на

несколько отдельных таблиц, например, исходная, ненормализованная, таблица (табл. 2):

Таблица 2

Сотрудник	Номер телефона
Иванов И. П.	25-56-82
	39-89-34
Семенова А. Ю.	58-92-34

Таблица, приведенная к 1NF (табл. 3):

Таблица 3

Сотрудник	Номер телефона
Иванов И. П.	25-56-82
Иванов И. П.	39-89-34
Семенова А. Ю.	58-92-34

Теперь разберем пример второй нормальной формы (2NF). Таблица находится во второй нормальной форме, если она находится в первой нормальной форме, и при этом любой ее атрибут, не входящий в состав первичного ключа, функционально полно зависит от первичного ключа. Функционально полная зависимость означает, что атрибут функционально зависит от всего первичного ключа, но при этом не находится в функциональной зависимости от какой-либо его части. Пример приведения таблицы ко второй нормальной форме (табл. 4):

Таблица 4

Сотрудник	Должность	Зарплата
Иконников	Директор	30 000
Семенов	Завуч	25 000
Селезнева	Завуч	25 000

В результате приведения к 2NF получим две таблицы (табл. 5 и 6):

Таблица 5

Сотрудник	Должность
Иконников	Директор
Семенов	Завуч
Селезнева	Завуч

Таблица 6

Должность	Зарплата
Директор	30 000
Завуч	25 000

Обычно выделяют пять нормальных форм. Более подробно с понятием нормальной формы можно познакомиться, например, в [2].

1.2. Объекты базы данных

Если в базе нет никаких данных (пустая база), то это все равно полноценная база данных. Этот факт имеет методическое значение. Хотя данных в базе и нет, но информация в ней все-таки есть — это структура базы. Она определяет методы занесения данных и хранения их в базе. Простейший «не компьютерный» вариант базы данных — деловой ежедневник, в котором каждому календарному дню выделено по странице. Даже если в нем не записано ни строки, он не перестает быть ежедневником, поскольку имеет структуру, четко отличающую его от записных книжек, рабочих тетрадей и прочей писчебумажной продукции.

Базы данных могут содержать различные объекты. Основными объектами любой базы данных являются ее таблицы. Простейшая база данных имеет хотя бы одну таблицу. Соответственно структура простейшей базы данных тождественно равна структуре ее таблицы.

Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в простейшей базе данных являются поля и записи. Если записей в таблице пока нет, значит, ее структура образована только набором полей. Изменив состав полей базовой таблицы (или их свойства), мы изменяем структуру базы данных и соответственно получаем новую базу данных.

Рассмотрим объекты базы данных (на примере СУБД OpenOffice.org Base).

Таблицы

Таблицы — это основные объекты любой базы данных. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы хранят и структуру базы (поля, их типы и свойства). Таблица предназначена для хранения данных в виде записей (строк) и полей (столбцов). Обычно каждая таблица используется для хранения сведений по одному конкретному вопросу.

Запросы

Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции, как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразования данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое.

Формы

Если запросы — это специальные средства для отбора и анализа данных, то формы — это средства для ввода данных. Смысл их тот же — предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочее) для автоматизации ввода. Преимущества форм раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают графическими средствами так, чтобы она повторяла оформление бланка — это заметно упрощает работу наборщика, снижает его утомление и предотвращает появление печатных ошибок.

Отчеты

По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на принтер. В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов.

Базы данных — это тоже файлы, но работа с ними отличается от работы с файлами других типов, создаваемых прочими приложениями. Для базы данных предъявляются особые требования с точки зрения безопасности, поэтому в них реализован другой подход к сохранению данных.

Базы данных — это особые структуры. Информация, которая в них содержится, очень часто имеет общественную ценность. Нередко с одной и той же базой работают тысячи людей по всей стране. Поэтому целостность содержимого базы не может и не должна зависеть ни от конкретных действий некоего пользователя, забывшего сохранить файлы перед выключением компьютера, ни от перебоев в электросети.

Проблема безопасности баз данных решается тем, что в СУБД для сохранения информации используется двойной подход. В части операций, как обычно, участвует операционная система компьютера, но

некоторые операции сохранения происходят в обход операционной системы.

Глава 2. Основы работы с OpenOffice.org Base

2.1. Этапы проектирования базы данных в OpenOffice.org Base

Процесс создания базы данных можно представить в виде следующих основных этапов.

1. Определение цели создания базы данных. На первом этапе проектирования базы данных необходимо определить назначение базы данных, как она будет использоваться и какие сведения она должны содержать. Дается словесная и документальная характеристика данной предметной области. Зная это, можно определить, какие сведения будут храниться в таблицах и в полях таблиц. База данных должна отвечать требованиям конечного пользователя. Для этого необходимо определить темы, которые должна раскрывать база данных, отчеты, которые она должна выдавать.

2. Определение таблиц, которые должна содержать база данных. Определение необходимых в базе данных таблиц может оказаться самым непростым этапом процесса проектирования базы данных, поскольку результаты, которые должна выдавать база данных: отчеты, формы и т.п. — не всегда дают полное представление о структуре таблиц, по которым они создаются. Для проектирования таблиц вовсе не обязательно использовать СУБД. Разрабатываемая модель предметной области обычно представляется в виде графической схемы, начерченной на бумаге.

При разработке таблиц рекомендуется руководствоваться следующими основными принципами:

- сведения не должны дублироваться в таблице или между таблицами;
- данные, хранящиеся только в одной таблице, обновляются только в этой таблице. Это исключает возможность дублирования записей, содержащих разные сведения. Например, адрес и номер телефона каждого ученика достаточно сохранить один раз, в одной таблице;
- каждая таблица должна содержать информацию только на одну тему. Когда таблица содержит сведения только по одной теме, со сведениями по каждой теме можно работать независимо от остальных тем. Например, адрес читателя библиотеки хранится отдельно от списка книг, взятых этим читателем, что позволяет удалить список книг, сохранив сведения о читателе.

3. Определение необходимых в таблице полей. Каждая таблица содержит сведения по конкретной теме, а каждое поле в таблице содержит конкретный факт по теме таблицы. Например, таблица

сведений об ученике может содержать поля сведений об имени, отчестве, фамилии, дате рождения, адресе, номере телефона. При составлении схемы полей для каждой таблицы необходимо учитывать следующее:

- каждое поле должно быть связано с темой таблицы;
- не рекомендуется включать в таблицу данные, которые являются результатом выражения;
- данные следует разбить на наименьшие логические единицы (например, поля Имя и Фамилия, а не общее поле Имя)

4. Определение полей с уникальными значениями в каждой записи. Для связывания сведений, хранящихся в разных таблицах, например для связывания данных о читателе со всеми книгами, которые он брал, каждая таблица базы данных должна содержать поля или набор полей, однозначно определяющих каждую запись. Такое поле или набор полей называют первичным ключом.

5. Определение связей между таблицами. После разбиения сведений на таблицы и определения ключевых полей необходимо выбрать способ, которым СУБД будет объединять связанные сведения. Для этого необходимо определить связи между таблицами базы данных.

6. Усовершенствование структуры базы данных. После создания нужных таблиц, полей и связей необходимо еще раз просмотреть структуру базы данных и выявить возможные недочеты. Желательно это сделать до заполнения таблиц данными.

7. Ввод данных и создание других объектов базы данных. Если структуры таблиц отвечают поставленным требованиям, то можно вводить все данные. Затем можно создать все необходимые запросы, формы, отчеты.

2.2. Основные принципы работы OpenOffice.org Base

Программа OpenOffice.org Base является простой системой по внесению изменений, редактированию и обработке данных. База данных OpenOffice.org Base позволяет вставлять данные из внешних ресурсов. Источником может быть Adabas, JDBC, ADO, dBase, Текстовый файл, Документ электронной таблицы или Данные из адресной книги. Кроме того, можно использовать базу данных OpenOffice.org для подключения к внешним реляционным базам данных, например к базам данных MySQL или Oracle.

Рассмотрим далее, как можно использовать простые функции программы в повседневной работе. Для освоения навыков работы в программе еще раз вспомним структуру построения любой базы. Как правило, данные в базе хранятся в виде таблиц. Но если в обычном

табличном редакторе элементом таблицы является ячейка, то в таблице базы основой является поле.

Поле — это элемент таблицы, который содержит данные определенного рода, например фамилию ученика. В режиме таблицы для представления поля используется столбец или ячейка, в этом случае имя поля является заголовком столбца таблицы.

Запись — полный набор данных об определенном объекте. В режиме таблицы запись изображается как строка.

База данных может состоять из нескольких таблиц, связанных между собой по различным полям. Имеется возможность выбирать и анализировать данные по различным параметрам.

В нашем пособии рассматривается СУБД **OpenOffice.org Base**, входящая в состав пакета **OpenOffice.org 2.3**, включенного в Пакет свободного программного обеспечения для общеобразовательных учреждений Российской Федерации. Внешний интерфейс **OpenOffice.org Base** наследует общие принципы пакета **OpenOffice.org**, при этом возможны незначительные отличия в зависимости от версии **OpenOffice.org**.

Главное меню содержит команды **Файл, Правка, Вид, Вставка, Сервис, Окно, Справка**. Рассмотрим некоторые команды меню.

Пункт меню **Файл**.

Сохранить. Сохраняет текущий файл базы данных, запрос, форму или отчет. Для файла базы данных открывается диалоговое окно **сохранения файла**. Для прочих объектов открывается диалоговое окно **Сохранить**.

Сохранить как. Сохраняет текущий файл базы данных под другим именем. В диалоговом окне **сохранения файла** выберите путь и имя файла для сохранения.

Экспорт. Экспортирует выбранный отчет или форму в текстовый документ. Динамический отчет экспортируется как копия содержимого базы данных на момент экспорта.

Отправить. Открывает подменю. **Документ как сообщение электронной почты** — открывает заданное по умолчанию почтовое приложение для отправки нового сообщения. Текущий файл базы данных добавляется как вложение. Можно ввести тему, получателей и текст письма. **Отчет по электронной почте** — открывает заданное по умолчанию почтовое приложение для отправки нового сообщения. Выбранный отчет добавляется как вложение. Можно ввести тему, получателей и текст письма. Динамический отчет экспортируется как копия содержимого базы данных на момент экспорта. **Отчет в текстовый документ** — экспортирует выбранный отчет в текстовый

документ. Динамический отчет экспортируется как копия содержимого базы данных на момент экспорта.

Пункт меню **Правка**.

Копировать — копирует выделенный объект в буфер обмена.

Вставить — вставляет объект из буфера обмена. При необходимости можно вставлять формы и отчеты, включая вложенные папки, из одной базы данных в другую.

Вставить как — вставляет объект из буфера обмена. При необходимости можно вставлять формы и отчеты, включая вложенные папки, из одной базы данных в другую.

Изменить — открывает окно, где можно редактировать выбранную таблицу, запрос, форму или отчет.

Удалить — удаляет выбранную таблицу, запрос, форму или отчет.

Переименовать — переименование выбранного объекта. В зависимости от базы данных некоторые имена, символы или длина имени могут быть недопустимыми.

Открыть — открывает выбранный объект в том виде, в котором он сохранялся последний раз.

Создать как представление — преобразует выбранный запрос в представление. Исходный запрос остается в файле базы данных, а на сервере создается дополнительное представление. Для добавления представления к базе данных необходимо иметь разрешение на запись.

Мастер форм — запускает **мастер форм** для выбранной таблицы, запроса или представления.

Мастер отчетов — запускает **мастер отчетов** для выбранной таблицы, запроса или представления.

Выделить все — выделяет все записи, включая вложенные папки, в нижней части окна базы данных.

База данных — открывает подменю, содержащие команды **Свойства**, **Тип подключения**, **Дополнительные свойства**.

Пункт меню **Вид** содержит такие команды, как:

Объекты базы данных — открывает подменю **Формы** (выбирает контейнер для форм и отображает все формы в подробном представлении), **Отчеты** (выбирает контейнер для отчетов и отображает все отчеты в подробном представлении), **Запросы** (выбирает контейнер для запросов и отображает все запросы в подробном представлении), **Таблицы** (выбирает контейнер для таблиц и отображает все таблицы в подробном представлении).

Сортировка — открывает подменю, позволяющее сортировать **По возрастанию**, **По убыванию**.

Предварительный просмотр — открывается набор подменю, позволяющий отключать предварительный просмотр в окне базы данных, отображать информацию о документе формы или отчета и др.

Обновить таблицы — обновляет таблицы.

Пункт меню **Вставка**.

Форма — открывает новый текстовый документ в режиме формы.

Мастер отчетов — запускает мастер отчетов для выбранной таблицы, представления или запроса.

Запрос (конструктор) — открывает новый запрос в режиме конструктора.

Запрос (режим SQL) — открывает новый запрос в режиме SQL.

Конструктор таблиц — открывает конструктор таблиц.

Конструктор представлений — открывает новое представление в режиме конструктора.

Представление (простое) — открывает новое представление в режиме SQL.

Папка — открывает диалоговое окно, где можно сохранить новую папку в файл базы данных.

Пункт меню **Сервис**.

Связи — открывает **Конструктор связей** и проверяет, поддерживаются ли связи подключением базы данных.

Управление пользователями — открывает диалоговое окно **Управление пользователями**, если база данных поддерживает эту функцию.

Фильтр таблиц — открывает диалоговое окно **Фильтр таблиц**, где можно выбрать, какие из таблиц базы данных следует показать, а какие — скрыть. В списке **Фильтр** выберите таблицы, которые требуется отфильтровать. Если выбрать самую верхнюю таблицу в иерархии, будут выбраны все таблицы этой иерархии. Если выбрать в иерархии таблицу более низкого уровня, вышестоящие таблицы этой иерархии не выбираются.

SQL — открывает диалоговое окно "SQL", где можно вводить инструкции SQL.

2.3. Создание базы данных в OpenOffice.org Base

Чтобы создать новый файл базы данных, можно пойти двумя путями. Если уже открыт какой-либо документ OpenOffice.org, то достаточно выбрать команду меню **Файл — Создать — База данных**. Второй путь предполагает запуск программы **OpenOffice.org Base** через меню графической оболочки **KDE, Gnome**. Как правило, для этого необходимо открыть главное меню, затем **Офис** —

«OpenOffice.org Base». Во время работы часто возникают вопросы по выполнению тех или иных задач. **OpenOffice.org Base** имеет хорошую справочную систему, которая поможет ответить практически на любой вопрос. Для вызова справки выберете пункт **Вызов справки**.

Создание новой базы данных происходит с помощью мастера и состоит из нескольких этапов. На первом этапе в диалоговом окне предлагается выбрать, создавать новую базу данных или открыть уже существующую (рис. 1).

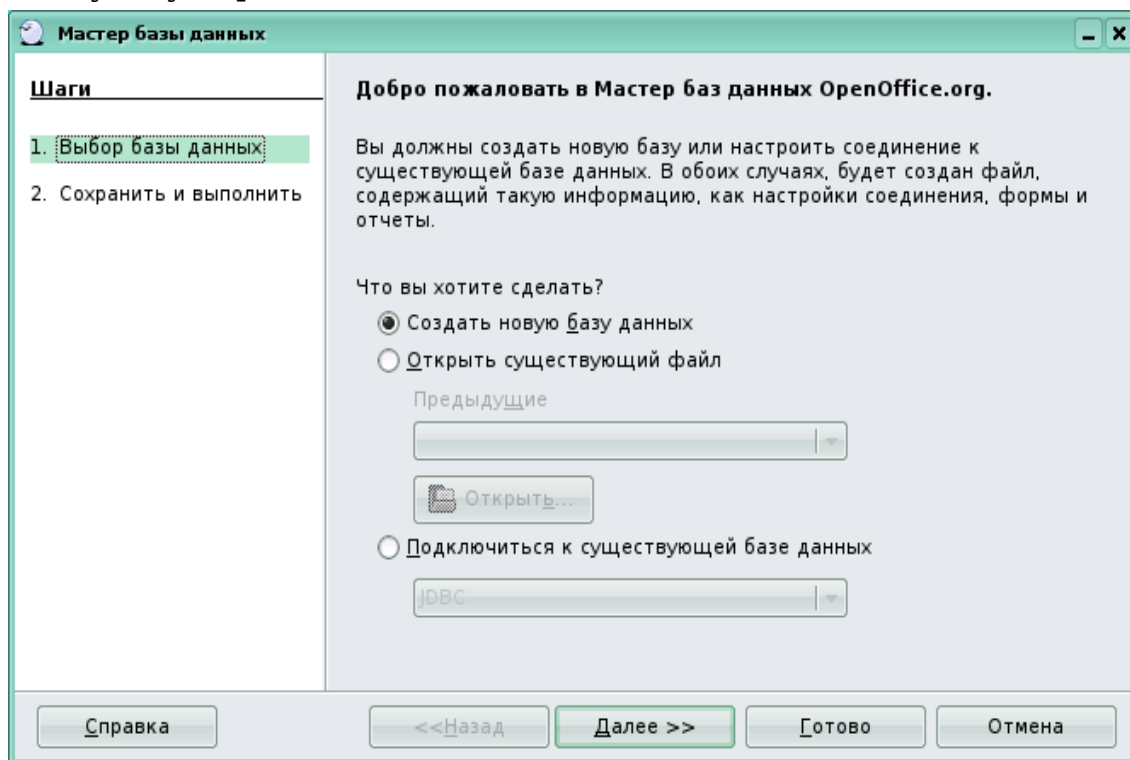


Рис. 1

Чтобы создать новую, необходимо выбрать **Создать новую базу данных** и нажать на кнопку **Далее**.

На втором этапе мастер предлагает два варианта: открыть базу данных для редактирования или сразу приступить к созданию таблиц (рис. 2). После выбора варианта необходимо нажать на кнопку **Готово**.

Чтобы открыть файл базы данных в процессе работы, выберите команду меню **Файл — Открыть**. Из списка **Тип файла** выберите для просмотра только **Документы базы данных**. Выберите документ базы данных и щелкните на кнопку **Открыть**.

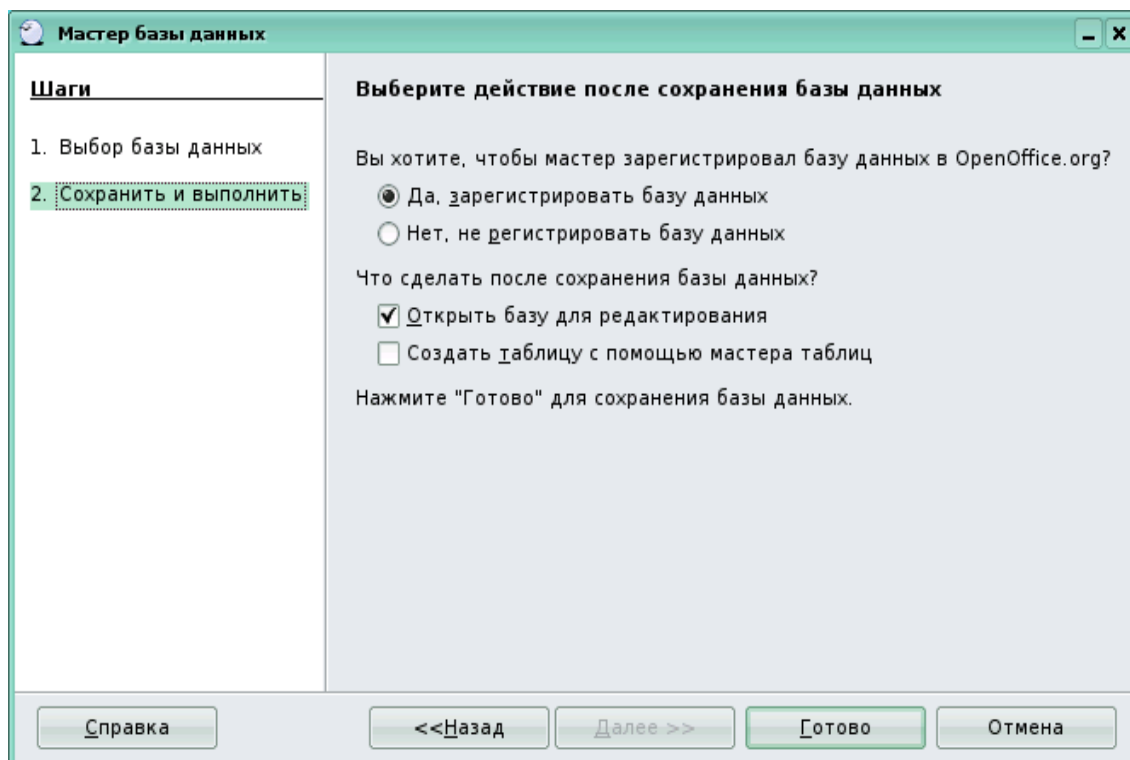


Рис. 2

2.4. Открытие базы данных

Создав базу данных (с нуля, на основе уже существующей базы данных), вы будете периодически открывать ее для внесения изменений или проведения различных операций. Для того чтобы открыть базу данных, выполните следующие действия.

1. Щелкните на кнопке **Открыть** на панели инструментов, и на экране появится предназначенное для этого диалоговое окно (рис. 3).
2. Укажите необходимую папку в списке **Папка** и щелкните на имени файла базы данных, имеющего расширение * .odb.
3. Щелкните по кнопке **ОК** (рис. 4).

После этого вы можете приступить к формированию таблиц базы данных.

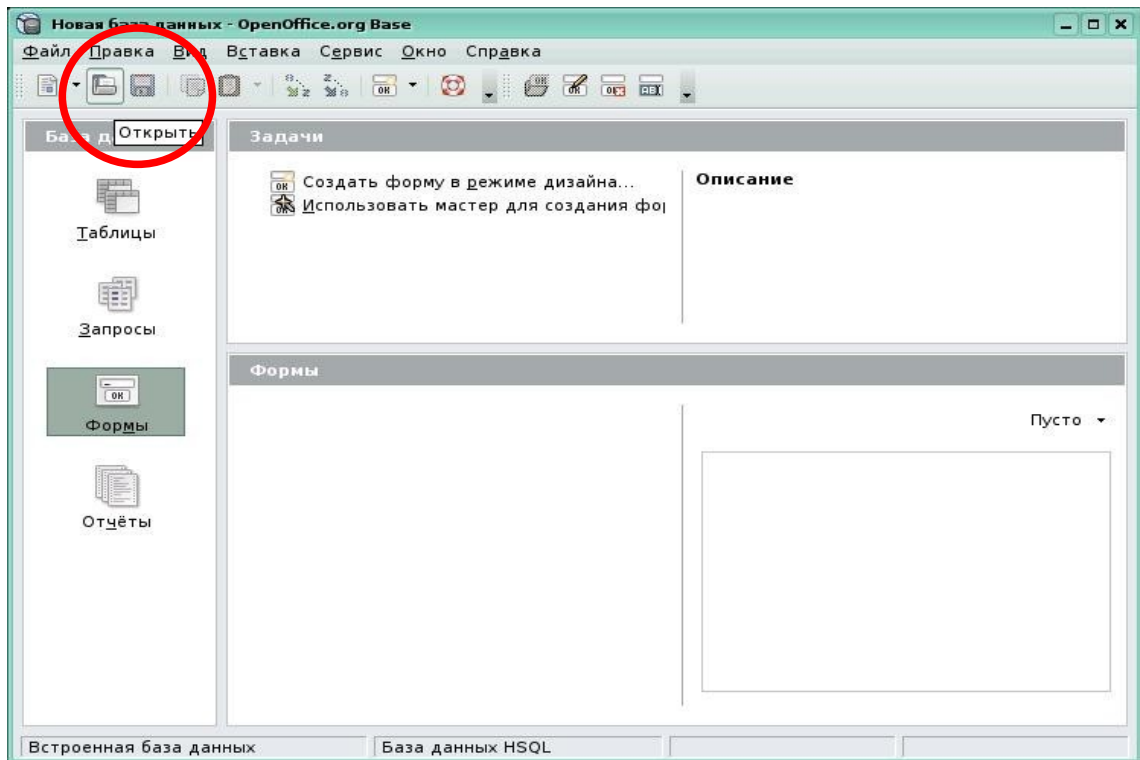


Рис. 3

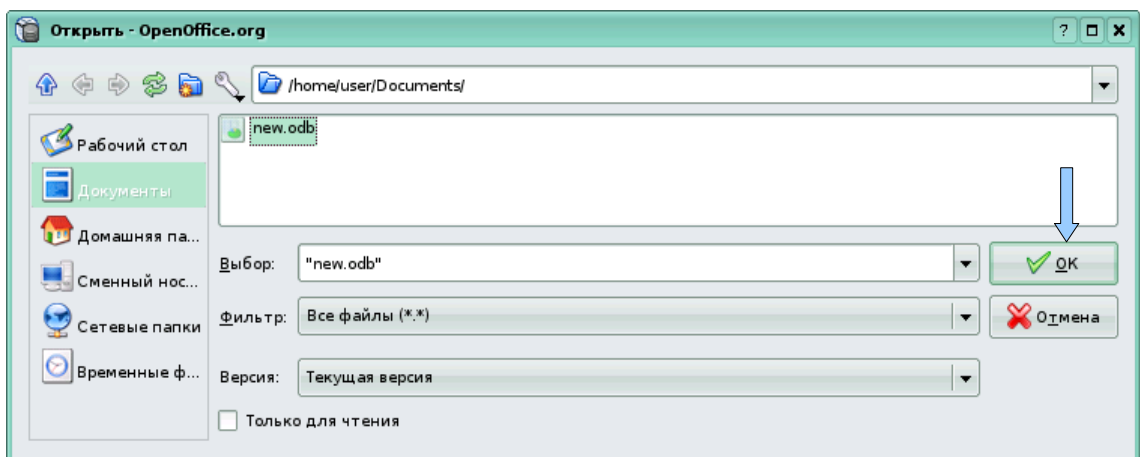


Рис. 4

2.5. Создание новой таблицы

После того как создан новый файл базы данных, редактор откроет основное рабочее окно базы (рис. 5). Здесь можно работать с таблицами, запросами, формами и пр. Создавать, редактировать, управлять ими. Причем практически каждую форму можно создавать в двух режимах — с помощью мастера (для начинающих) или с помощью режима дизайна (для опытных пользователей).

Чтобы воспользоваться помощью мастера таблиц, следует щелкнуть на кнопке **Таблицы** в окне базы данных. В программе OpenOffice.org Base предусмотрено три различных способа создания таблицы базы данных: **Создать таблицу в режиме дизайна** (вся работа по определению структуры таблицы и ее созданию ложится на разработчика), **Использовать мастер для создания таблицы** (позволяет частично автоматизировать процесс создания таблицы), **Создать представление**.

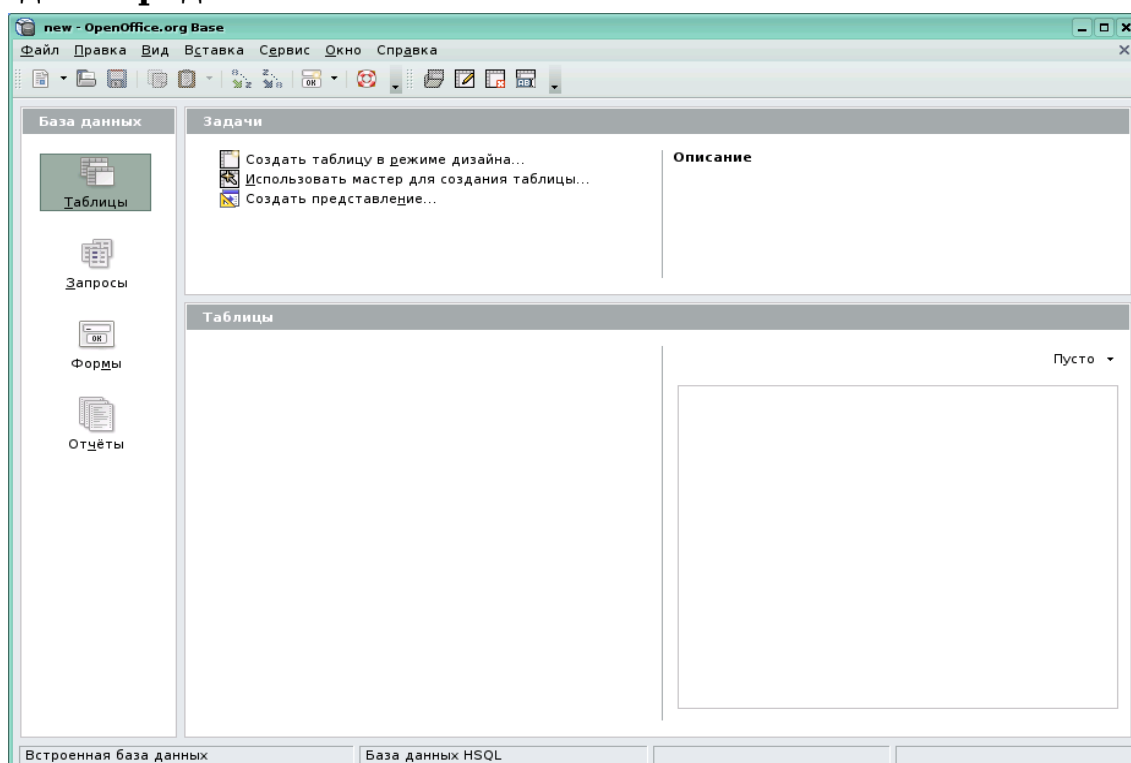


Рис. 5

Рассмотрим создание таблицы с помощью мастера. Для создания таблицы откройте файл базы данных, в которой требуется создать новую таблицу. Щелкните значок **Таблицы**, а справа щелчком левой кнопки мыши выберите **Использовать мастер для создания таблиц...** (рис. 6).

Мастер создания таблиц предлагает множество уже готовых шаблонов. Остается выбрать подходящий и использовать необходимые поля. Создание таблицы начинается с определения ее категории с помощью переключателей **Деловой**, **Персональный**. Затем в списке **Примеры таблиц** следует выбрать название наиболее подходящего образца. После щелчка по названию выбранного образца формируется соответствующее содержимое списка **Переменные поля**.

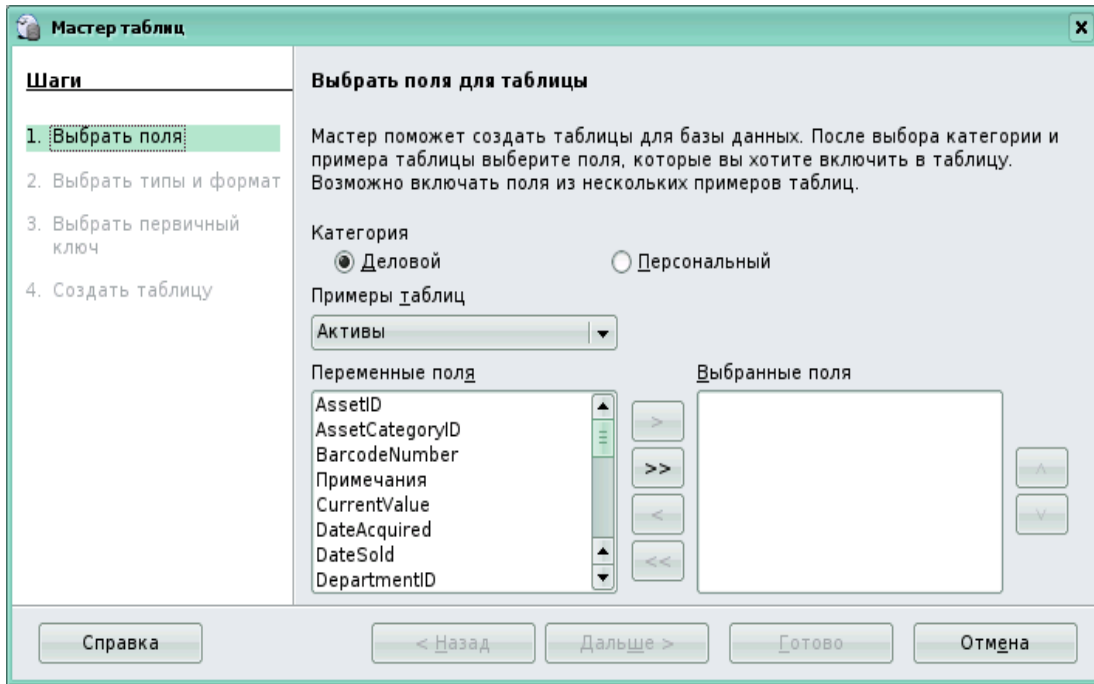



Рис. 6

Следует просмотреть этот список и отобрать те поля, которые предусмотрены в проекте таблицы. Чтобы поле попало в структуру формируемой таблицы, его нужно выделить, а затем щелкнуть по кнопке .

Поле, случайно попавшее в список, можно исключить из него. При желании любое поле, включенное в список выбранных полей, можно переименовать. Это можно сделать в окне следующего шага (рис. 7).

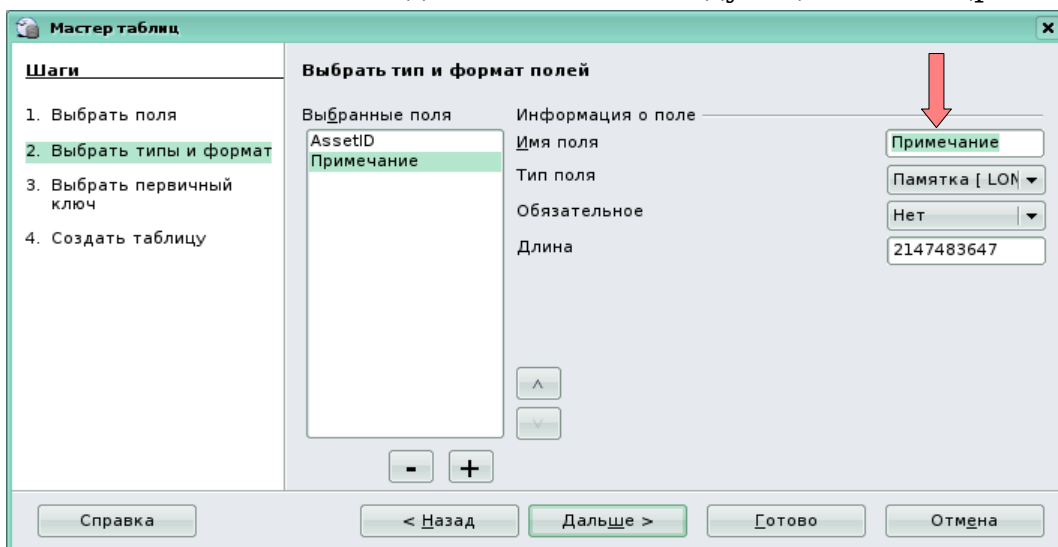


Рис. 7

На третьем шаге происходит задание первичного ключа. В последнем окне мастер предлагает три варианта дальнейших действий:

Немедленно вставить данные, Модифицировать дизайн таблицы, Создать форму на основе данной таблицы. Следует сделать выбор и щелкнуть на кнопке **Готово**. В случае выбора первого варианта происходит переключение в режим таблицы, в котором можно выполнить первичное заполнение базы. В случае выбора второго варианта происходит переключение в режим дизайна, в котором можно произвольным образом изменить структуру созданной таблицы. При выборе третьего варианта мастер таблиц автоматически создаст форму, обеспечивающую более удобный стиль заполнения базы.

Далее рассмотрим процесс создания новой таблицы в режиме дизайна. Чтобы создать новую таблицу в режиме дизайна, выберите **Создание таблицы в режиме дизайна**.

Теперь можно создать поля для таблицы в режиме дизайна. Создадим таблицу «Ученики», содержащую информацию об учениках. Наша таблица будет содержать такие поля: порядковый номер, имя, отчество, фамилия, дата рождения, возраст, класс, параллель, улица, дом, квартира, классный руководитель.

Введем новые поля в строки сверху вниз. Щелкните ячейку **Имя поля** и введите имя для каждого поля данных (рис. 8).

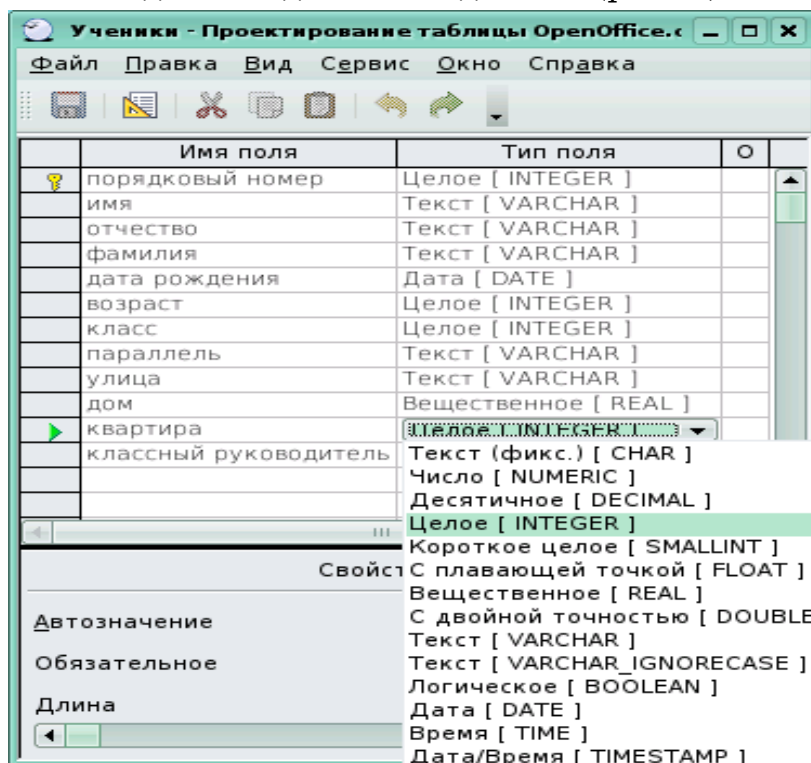


Рис. 8

В следующей ячейке справа определите **Тип поля**. Щелкнув в этой ячейке, выберите тип поля из списка (см. рис. 8).

В нашем примере поля будут иметь следующий тип:

порядковый номер — целое,
имя — текст,
отчество — текст,
фамилия — текст,
дата рождения — дата,
возраст — целое,
класс — целое,
параллель — текст,
улица — текст,
дом — вещественное,
квартира — целое,
классный руководитель - целое (номер).

В каждое поле можно вводить данные, соответствующие заданному типу. Например, нельзя ввести текст в числовое поле. Поля записок в формате dBase III являются ссылками на внутренне управляемые текстовые файлы, которые могут содержать до 64 Кбайт текста.

Для каждого поля можно ввести **Описание**. Текст описания будет возникать в виде всплывающей подсказки при наведении курсора на заголовок в представлении таблицы.

Ниже вводятся **Свойства поля** для каждого выбранного поля данных (рис. 9).

В зависимости от типа базы данных некоторые возможности ввода могут быть недоступны.

В поле **Значение по умолчанию** введите содержимое по умолчанию для каждой новой записи. Это содержимое потом можно будет изменить.

В поле **Обязательное** укажите, может ли поле оставаться пустым.

В поле **Длина** можно открыть список с вариантами выбора. После того как все поля будут заполнены, откройте таблицу для заполнения, для этого в окне базы данных щелкните два раза левой кнопкой мыши на названии таблицы.

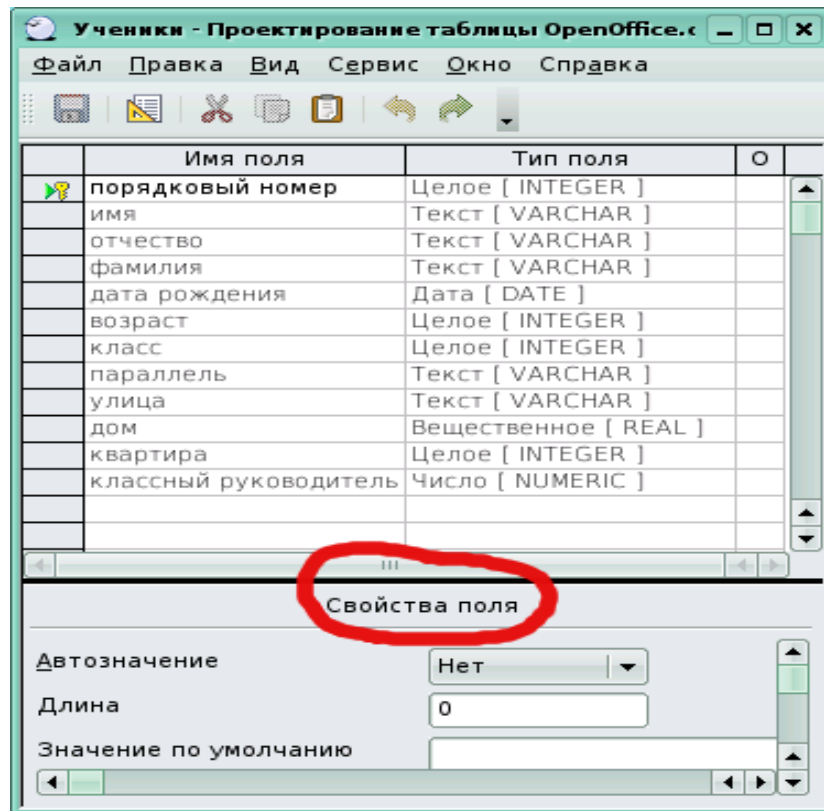


Рис. 9

Заполненная таблица выглядит следующим образом (рис. 10):

порядков...	имя	отчество	фамилия	дата рождения	возраст	класс	парал...	улица	дом	кварт...	классный руковод...
1	Иван	Аркадьевич	Белоногов	02.12.96	12	5	а	Ленина	17	114	1
2	Светлана	Андреевна	Сидорова	15.04.96	12	5	б	Парковая	104	26	2
3	Ирина	Ивановна	Аллаева	21.10.96	12	5	а	Садовая	12	68	1

Рис. 10

Записи в таблице можно отсортировать. Для этого:

1. Выделите тот столбец, по которому будет проводиться сортировка.
2. Выберите на панели инструментов кнопку " **По возрастанию** " или " **По убыванию** ".

В нашей таблице мы отсортировали по алфавиту поле «Фамилия», результат сортировки показан на рис. 11.

порядков...	имя	отчество	фамилия	дата рождения	возраст	класс	парал...	улица	дом	кварт...	классный руковод
3	Ирина	Ивановна	Аллаева	21.10.96	12	5	а	Садовая	12	68	1
1	Иван	Аркадьевич	Белоногов	02.12.96	12	5	а	Ленина	17	114	1
2	Светлана	Андреевна	Сидорова	15.04.96	12	5	б	Парковая	104	26	2

Рис. 11

2.6. Связи

При существовании связей между именем поля в одной таблице и именем поля в другой таблице эти связи можно использовать в запросе.

Если, к примеру, имеется электронная таблица «Ученики» и электронная таблица, в которой хранится информация о классных руководителях — «Учителя», то существует связь между полями данных. Теперь, чтобы создать запрос о классном руководителе ученика, необходимо получить данные из двух электронных таблиц. Для этого требуется указать в OpenOffice.org Base, какая связь существует между данными этих электронных таблиц.

Чтобы приступить к созданию связей, выберите пункт меню **Сервис — Связи** (рис. 12).

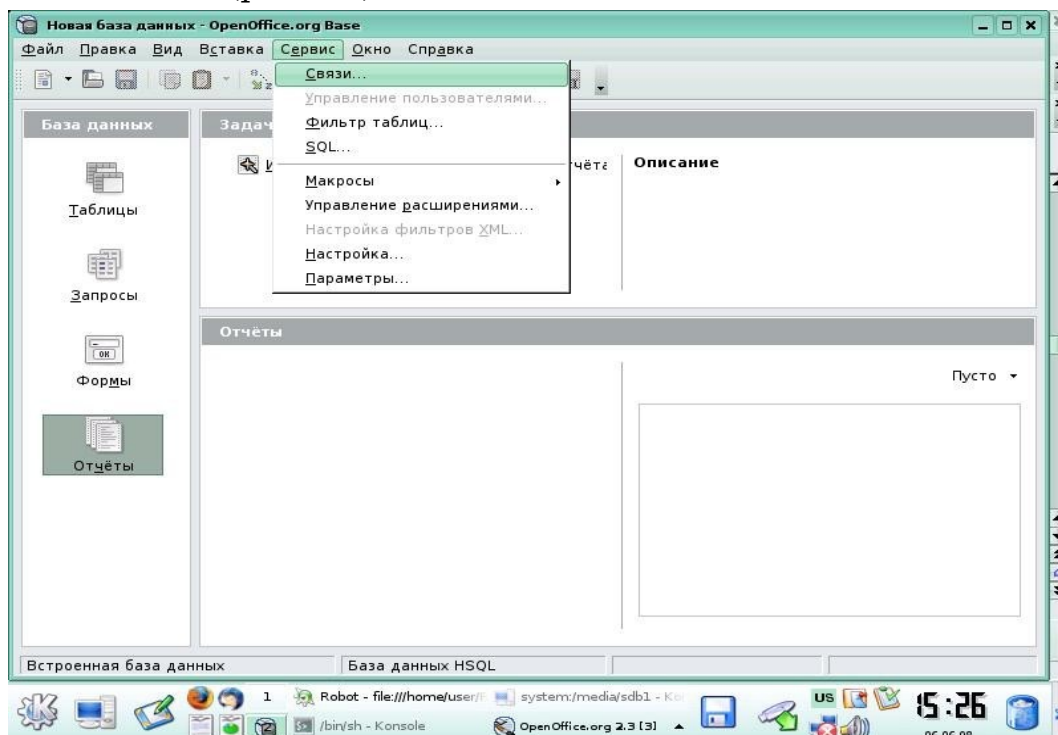


Рис. 12

Следует помнить, что создавать или изменять связи между открытыми таблицами нельзя. Поэтому вначале необходимо закрыть все ранее открытые таблицы базы данных и открыть окно базы данных.

Если в базе еще не определены связи между таблицами, автоматически откроется диалоговое окно **Добавить таблицы** (рис. 13).

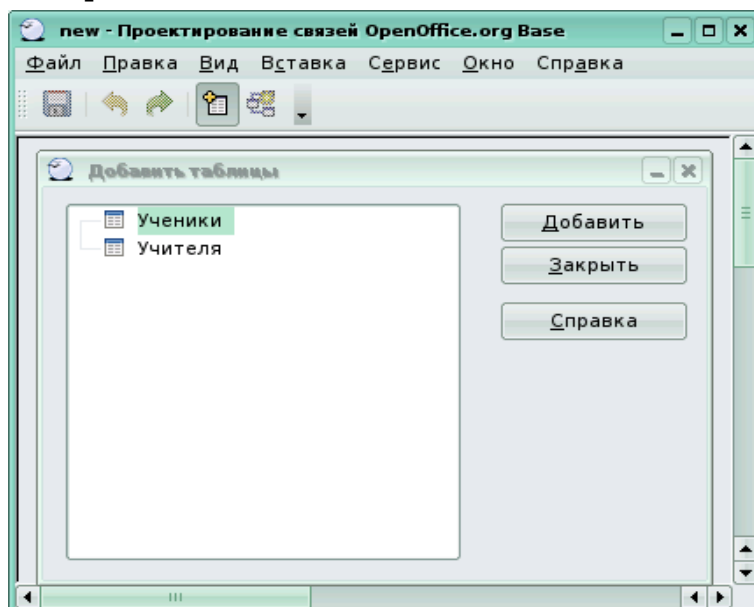


Рис. 13

В открытом окне **Добавить таблицы** следует выполнить двойные щелчки по названиям таблиц, которые участвуют в связи, а затем закрыть его. Кроме того, можно перетащить нужные таблицы, удерживая левую кнопку мыши. В нашем примере для наглядности добавим все таблицы базы данных.

Далее есть два варианта создания связи. Выбираем пункт меню **Вставка – Создать связь**. Появляется модальный диалог (рис. 14).

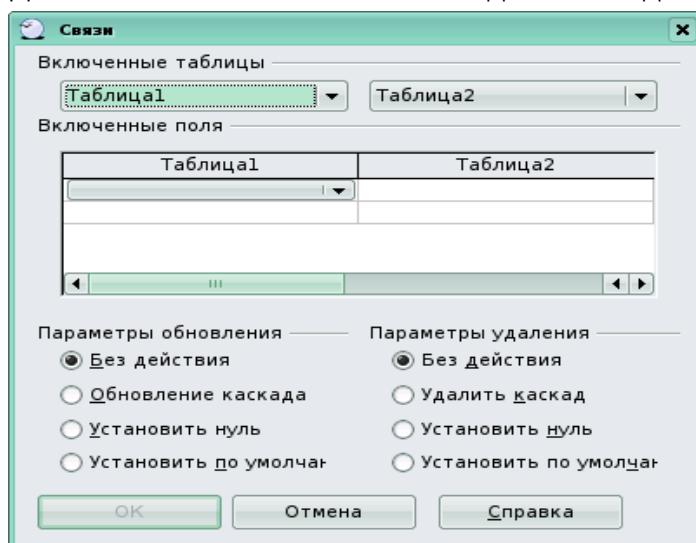


Рис. 14

В нем мы осуществляем все необходимые настройки и в результате получаем отображение всех таблиц и связей между ними (рис. 15).

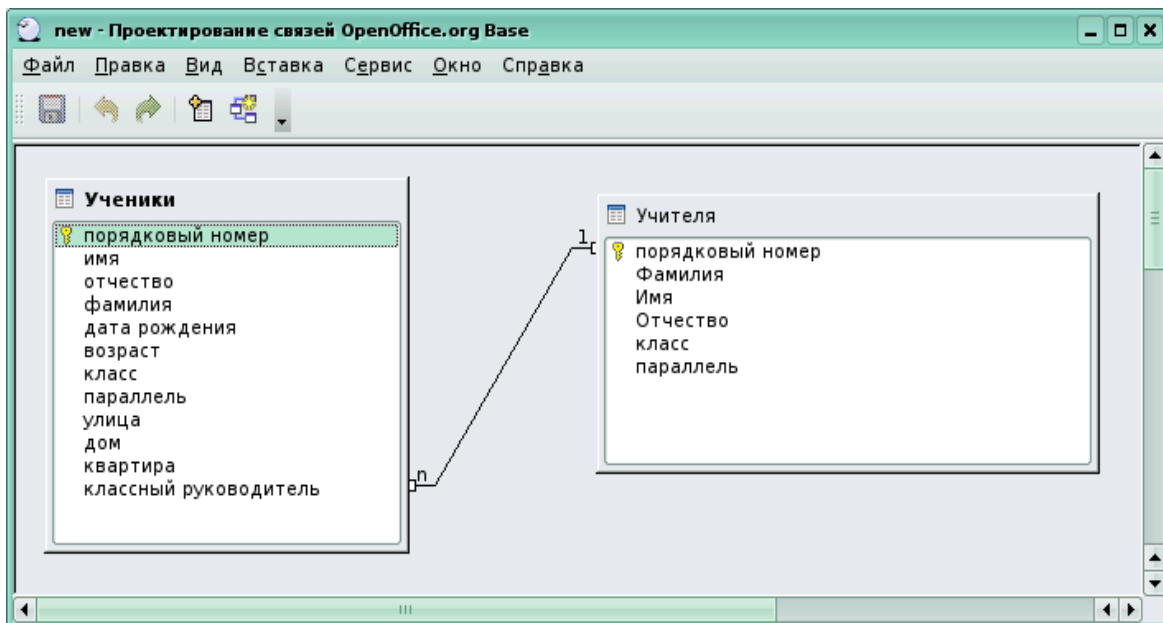


Рис. 15

В диалоговом окне (см. рис. 14) выбираются таблицы, определяются включенные поля, задаются параметры обновления (эти параметры вступают в силу при изменении поля первичного ключа) и удаления (эти параметры вступают в силу при удалении поля первичного ключа). Можно задать следующие параметры обновления:

- **Без действия** — указывается, что любое изменение первичного ключа не должно затрагивать остальные внешние ключевые поля.
- **Обновление каскадом** — происходит обновление всех внешних ключевых полей при изменении соответствующего первичного ключа (обновление каскадом).
- **Установить нуль** — если соответствующий первичный ключ изменен, этот параметр используется для задания значения "ПУСТО" всем внешним ключевым полям. Это значение указывает, что поле пустое.
- **Установить по умолчанию** — если соответствующий первичный ключ изменен, этот параметр используется для задания значения по умолчанию всем внешним ключевым полям. При создании соответствующей таблицы значение по умолчанию внешнего ключевого поля определяется при задании свойств данного поля.

Аналогичный смысл имеют параметры удаления.

Создать связь можно и следующим образом: щелкните имя поля в таблице (например, имя поля "класс"), удерживая кнопку мыши, и

перетащите имя поля на имя поля другой таблицы. Если отпустить кнопку мыши, появляется линия, соединяющая эти два поля в двух окнах.

Чтобы удалить связь между двумя таблицами, щелкните соединительную линию и нажмите клавишу **DELETE**, или выделите соединительный вектор с помощью клавиши **TAB**, вызовите контекстное меню, нажав клавиши **SHIFT+F10**, и выберите команду **Удалить**.

Связь между таблицами можно редактировать, для этого можно воспользоваться контекстным меню связи.

2.7. Создание запросов

Если вам часто приходится обращаться только к подмножеству своих данных, которые могут быть хорошо определены условием фильтра, рекомендуется составить запрос. По сути, это просто новое представление отфильтрованных данных. Открыв запрос, вы увидите текущие данные в виде таблицы, которую вы определили.

Создавать запросы можно в режиме **Мастер запросов** или в режиме **Дизайн запросов**. Для создания запроса в режиме **Мастер запросов** необходимо:

1. Открыть файл базы данных, в котором нужно создать новый запрос.
2. В левой области окна базы данных щелкнуть значок **Запросы**.
3. Установить флажок **Использовать мастер** для создания запросов.

Чтобы создать запрос в режиме **Дизайн запросов**, щелкните значок **Запросы** в документе базы данных, а затем щелкните **Создать запрос в режиме дизайна**.

Определение запроса выполняется в нижней области окна. Чтобы определить запрос, укажите имена полей базы данных, которые требуется включить, а также условия отображения полей. Чтобы переупорядочить столбцы в нижней области конструктора, перетащите заголовок столбца в новое расположение или выберите столбец и нажмите **CTRL** и клавишу со стрелкой. Создадим запрос, который будет выводить фамилии учеников старше 10 лет (рис. 16).

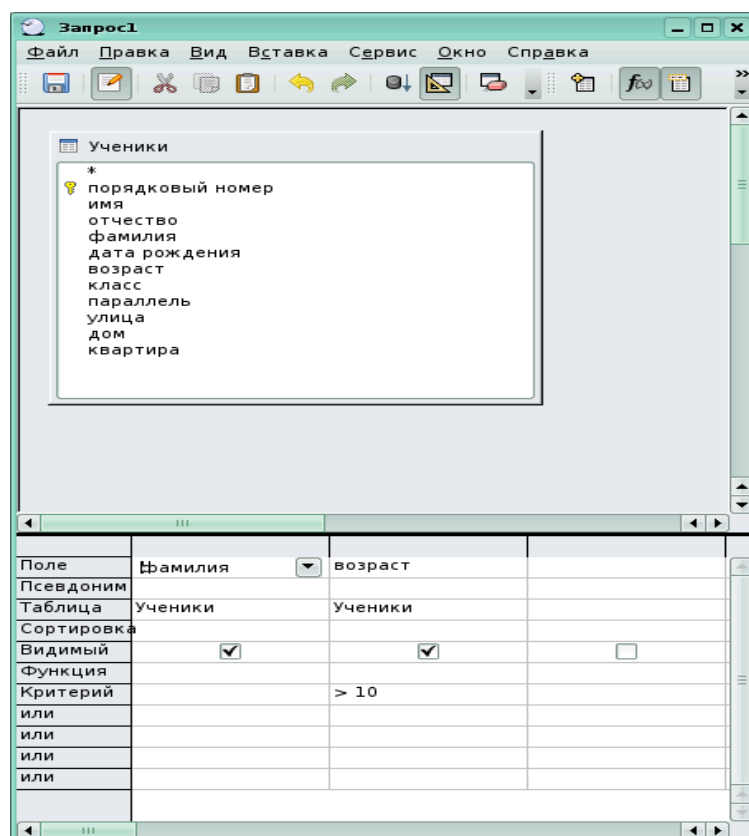


Рис. 16

Для формирования запроса выбираем поля **Фамилия** и **Возраст**. Указываем, что это поля видимые. Для поля **Возраст** вводим критерий выбора «>10». После этого закрываем окно запроса. При закрытии окна редактор предложит сохранить запрос, сохраняем его, указав имя запроса.

Чтобы проверить запрос, дважды щелкните его имя в документе базы данных.

Результат запроса отобразится в таблице, аналогичной представлению источника данных, также результаты запроса можно посмотреть не выходя из режима дизайна, для этого нужно выбрать пункт меню **Вид — Вид** или просто нажать клавишу **F4**. В нашем примере получились следующие результаты запроса (рис. 17):



Рис. 17

2.8. Создание форм

Рассмотрим создание новой формы с помощью мастера форм. Для создания новой формы в OpenOffice.org Base можно использовать

Мастер форм:

1. Откройте файл базы данных, в котором нужно создать новую форму.
2. В левой области окна базы данных щелкните значок **Формы**.
3. Выберите **Использовать мастер для создания формы**.

Мастер форм предложит вам следующие шаги:

Шаг 1. Выбор поля (рис. 18).

Выбираем таблицу или запрос, по которым мы будем создавать форму. В нашем примере для таблицы «Ученики» мы выбрали поля Фамилия и Класс.

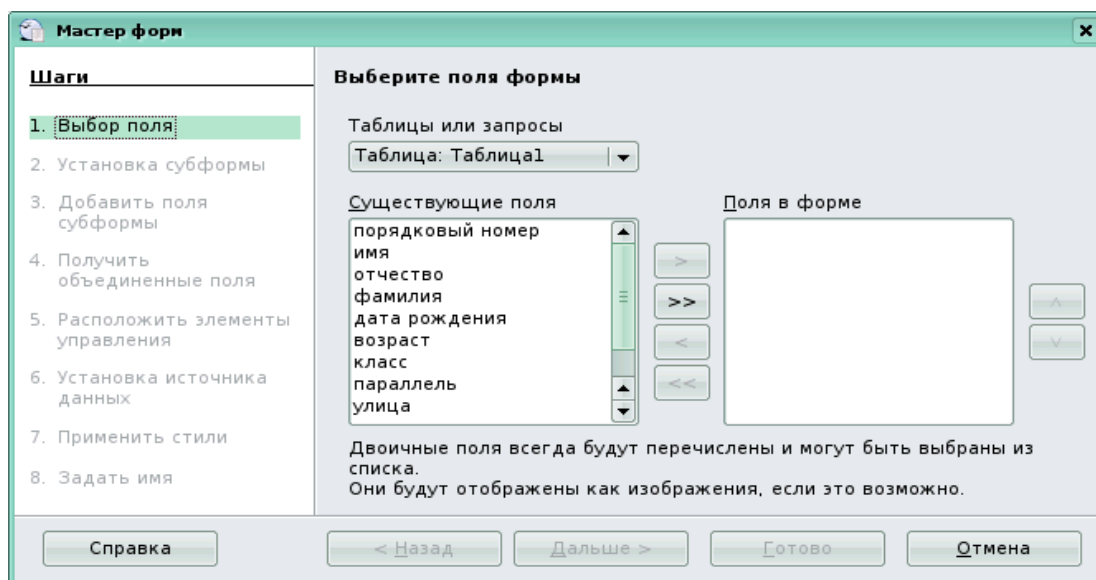


Рис. 18

Шаг 2. Установка субформы.

Эта форма является вложенной в другую форму. Используется для отображения данных из таблиц или запросов типа «один ко многим». Если вы выбрали **Добавить субформу**, вам нужно выполнить два дополнительных шага, в которых вы выбираете необходимые вам пункты, аналогично шагу 1. В нашем случае субформа не нужна, и мы переходим сразу к шагу 5.

Шаг 5. Расположить элементы управления.

На этом шаге выберите, каким образом будут расположены элементы на форме (рис. 19).

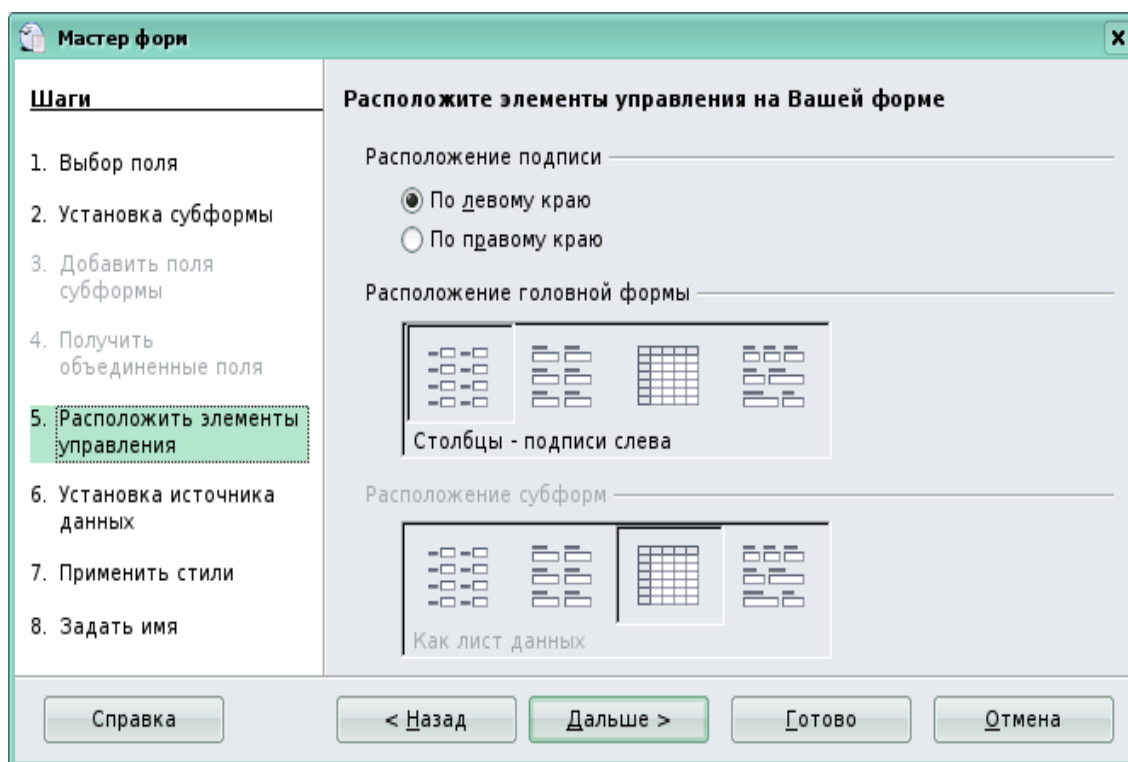


Рис. 19

Шаг 6. Установка источника данных.

На данном этапе производится выбор режима источника данных.

В зависимости от ваших потребностей форма может использоваться для отображения всех данных или только для ввода новых данных, можно запретить изменение существующих данных, добавление и удаление данных.

Мы выбираем вариант «только для ввода новых данных».

Шаг 7. Применить стили.

Выберите цвет формы и вид ее элементов.

Шаг 8. Задать имя.

Задайте имя формы и выберите действие после заполнения формы. В результате получаем следующую форму (рис. 20).

Возможно также создание новой формы вручную. Для этого необходимо:

1. Открыть файл базы данных, в котором нужно создать новую форму.
2. В левой области окна базы данных щелкнуть значок **Формы**.
3. Щелкнуть **Создание формы в режиме конструктора**.

Откроется новый текстовый документ. Чтобы вставить в форму элементы управления, используйте **Элементы управления форм**.

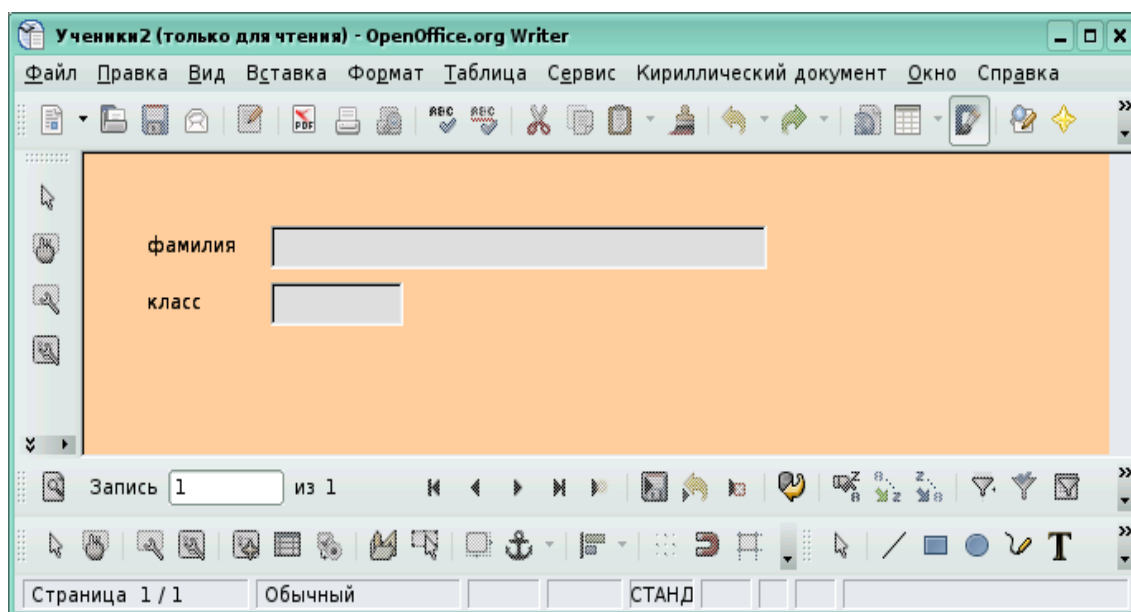


Рис. 20

2.9. Создание отчетов

Отчет является эффективным средством наглядного представления данных в печатном документе. Отчет — это текстовый документ, отображающий текущие данные или данные на время создания отчета. Разработчик имеет возможность эффективно управлять размером, внешним видом и структурой всех компонентов отчета.

Большинство отчетов связаны с одной или несколькими таблицами и/или запросами базы данных, поля которых являются источником формирования для включаемых в отчет записей. Отчет может включать любые комбинации полей из каждой таблицы или запроса, на основе которых он создается.

Для создания нового отчета в OpenOffice.org Base можно использовать **Мастер отчетов**:

1. Откройте файл базы данных, в которой необходимо создать новый отчет.
2. В левой области окна базы данных щелкните значок **Отчеты**.
3. Выберите задачу **Использовать мастер для создания отчета** (рис. 21).

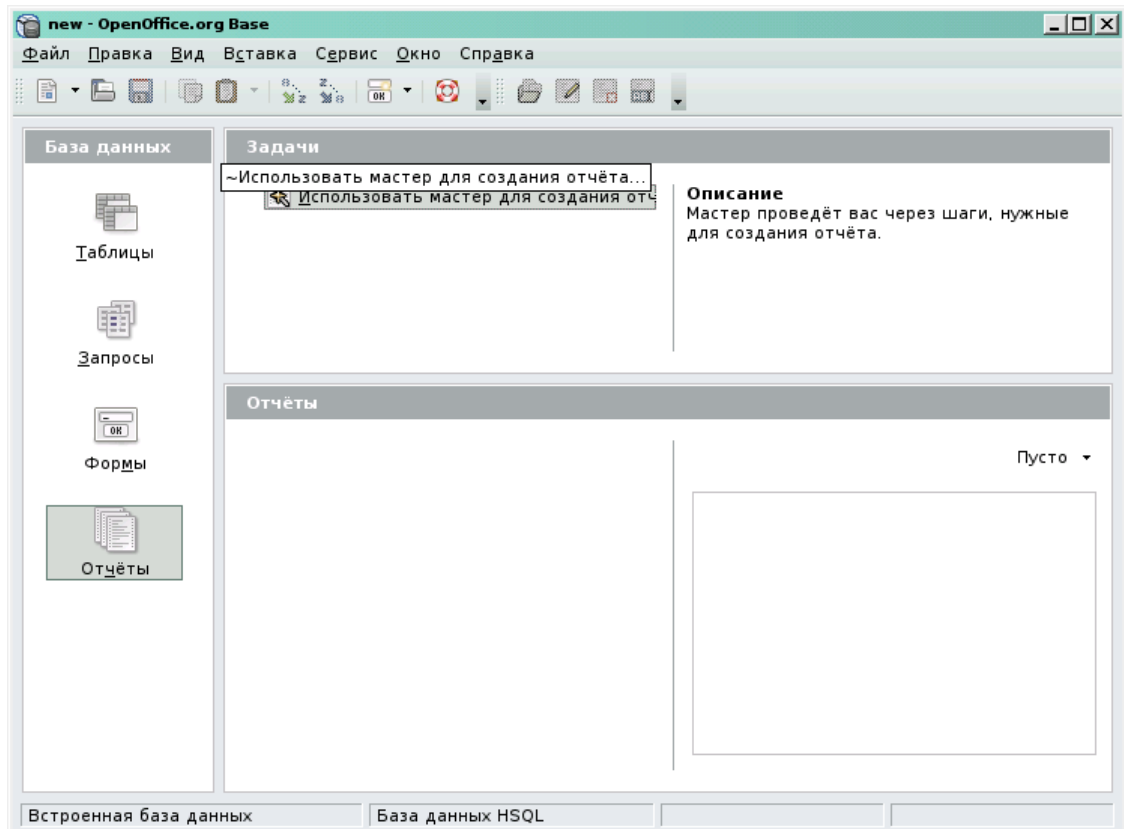


Рис. 21

В результате откроется окно первого шага мастера построения отчетов (рис. 22).

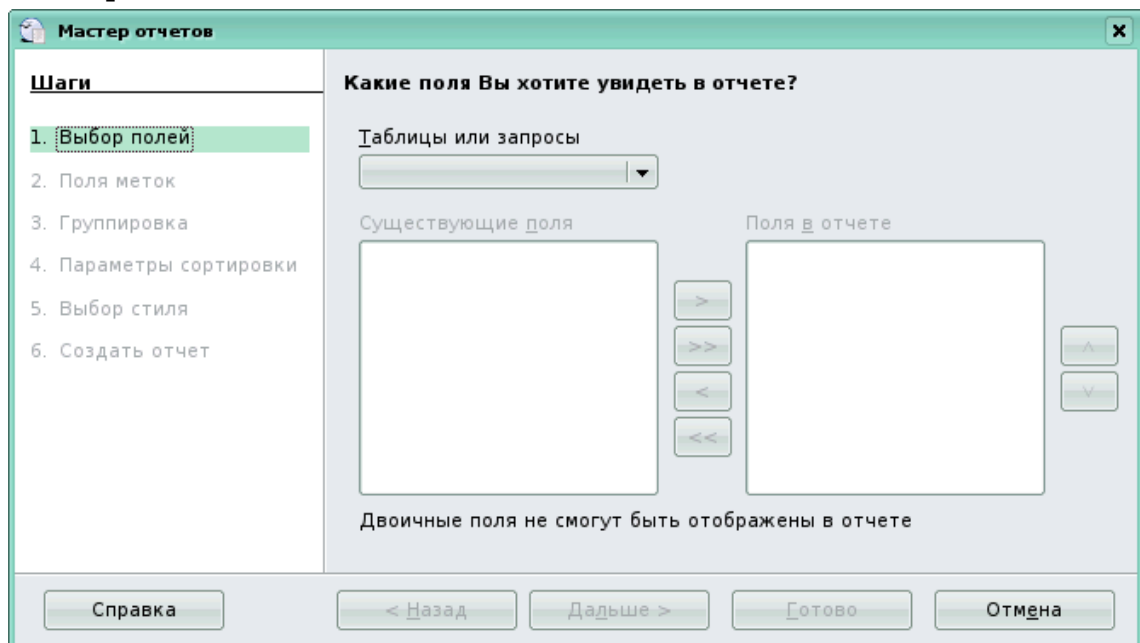


Рис. 22

Описанным ранее способом с помощью списка **Таблицы или запросы** в этом окне следует сформировать список **Поля в отчете** и с помощью кнопки **Далее** перейти к следующему шагу.

На втором шаге **Поля меток** указывается, что будут обозначать поля. В окне третьего шага можно осуществлять группировку записей в отчет. Записи отчета можно группировать на основе значений одного или нескольких полей. Выберите поля, по которым будет сгруппирован итоговый отчет. В отчете можно группировать до четырех полей. При группировке нескольких полей OpenOffice.org Base создает вложенные группы согласно их уровню группировки. В образце отчета отображается получаемая при этом структура отчета, в котором каждый следующий уровень сдвинут вправо по отношению к предыдущему уровню и обозначены поля, отображаемые на каждом уровне.

В окне четвертого шага **Параметры сортировки** мастер предоставляет возможность выбрать поля, по которым требуется сортировать отчет. Поля можно сортировать по четырем уровням, на каждом — по возрастанию или по убыванию. Сгруппированные поля можно сортировать только внутри соответствующих групп.

На следующем шаге **Разметка** выберите разметку из разных шаблонов и стилей и ориентацию страницы (книжную или альбомную).

Отчет можно создать как статический или как динамический отчет. При открытии динамического отчета он отображается с текущим содержимым. При открытии статического отчета всегда отображаются те данные, которые содержались в этом документе в момент его создания.

2.10. Пример создания базы данных

В заключение рассмотрим пример создания базы данных «Библиотека». На этом примере покажем все стадии создания базы, формирования отчетов и запросов. Рекомендуем читателю повторить рассматриваемый пример или создать свой.

В базе данных «Библиотека» будем хранить следующую информацию: название книги, автор, шифр книги, название издательства, адрес издательства, главный редактор, количество экземпляров книги.

Очевидно, что если поместить всю эту информацию в одну таблицу, мы столкнемся с проблемой избыточных данных. Для того чтобы избежать этого, информация разбивается на несколько таблиц, что позволяет оптимально использовать память, этот процесс называется нормализацией. В нашем случае создадим две таблицы: 1)

таблица «Книги», содержащая поля название книги, шифр, автор, название издательства, количество экземпляров, и так как нам необходимо ключевое поле, добавим поле номер книги; 2) таблица «Издательства», содержащая поля название издательства, адрес издательства, главный редактор, в данном случае ключевым полем выберем название издательства. Между этими таблицами будет связь «один ко многим», так как несколько книг могут издаваться в одном издательстве (таблица «Книги»), названия издательств не совпадают (таблица «Издательства»).

Теперь можно приступить к непосредственному созданию базы данных. Открываем приложение **OpenOffice.org Base**. Напомним, что процесс создания новой базы описан в п. 2.3 настоящего пособия. Назовем нашу базу «Библиотека».

После того как база создана, приступим к созданию таблиц. Создадим наши таблицы в режиме дизайна (см. п. 2.5). Первой создадим таблицу «Книги».

В колонку **Имя поля** записываем названия полей в удобной для нас очередности, в колонке **Тип поля** выбираем тип данных, которые будет содержать это поле (рис. 23).

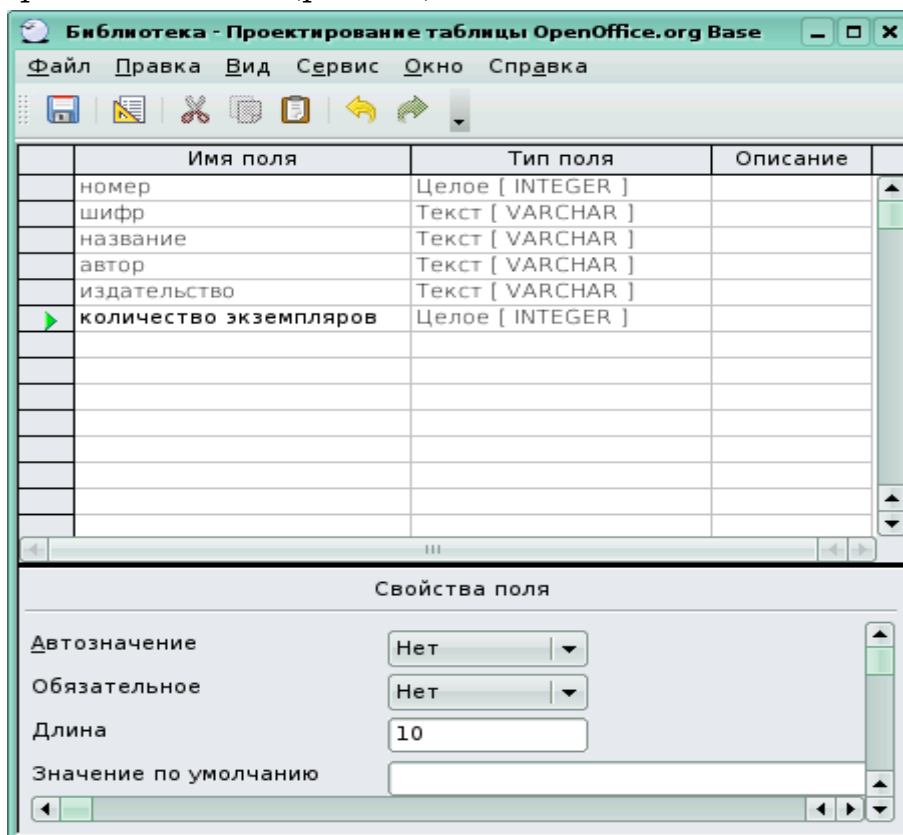


Рис. 23

Номер книги делаем целым числом, шифр — текстовым, так как он может содержать и буквы и цифры; название книги, автор и

название издательства тоже текстовые, а количество экземпляров — целое число.

После заполнения всех полей создаем первичный ключ. Напомним, что в данной таблице первичным ключом у нас будет номер. Щелкаем правой кнопкой мыши на прямоугольник слева от поля номер, в выпавшем контекстном меню выбираем пункт **Первичный ключ** (рис. 24).

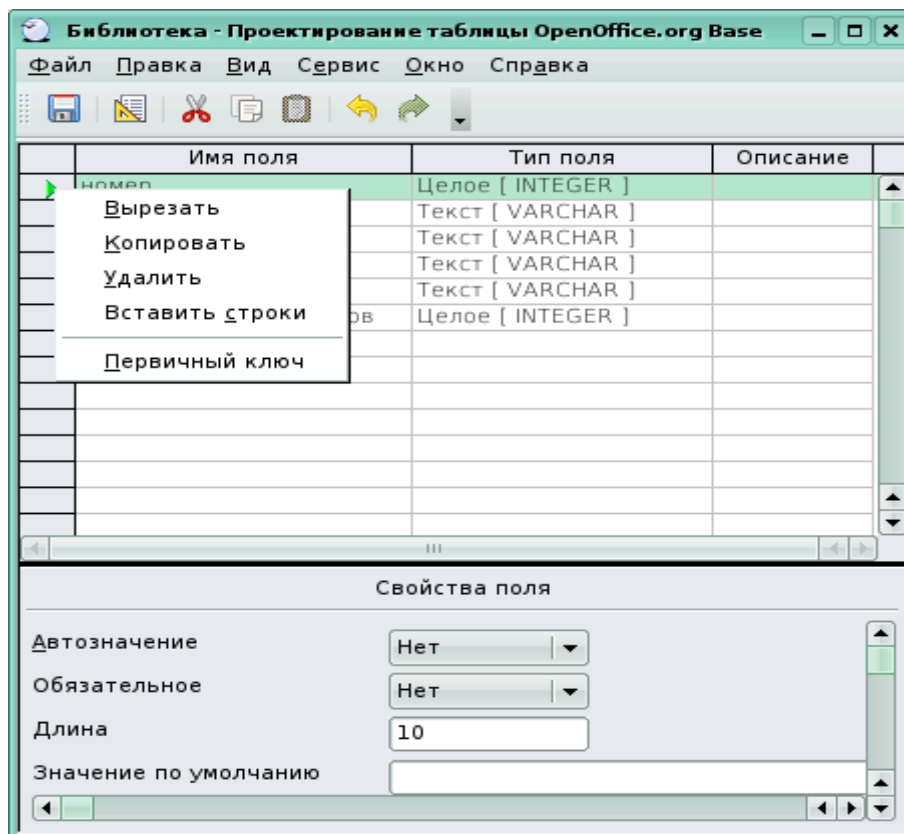


Рис. 24

Сохраняем изменения и закрываем окно редактирования. Аналогичным образом создаем таблицу «Издательства», которая имеет поля — название издательства, адрес издательства, главный редактор, с ключевым полем — название издательства.

Если мы посмотрим в главное окно базы данных, то увидим, что в правом нижнем разделе **Таблицы** появились наши таблицы «Книги», «Издательства» (рис. 25). Теперь заполним обе таблицы. Начнем с таблицы «Книги». Щелкнем на нее два раза левой кнопкой мыши, таблица откроется для заполнения, внесем в нее данные (рис. 26). Аналогичным образом заполним таблицу «Издательства» (рис. 27).

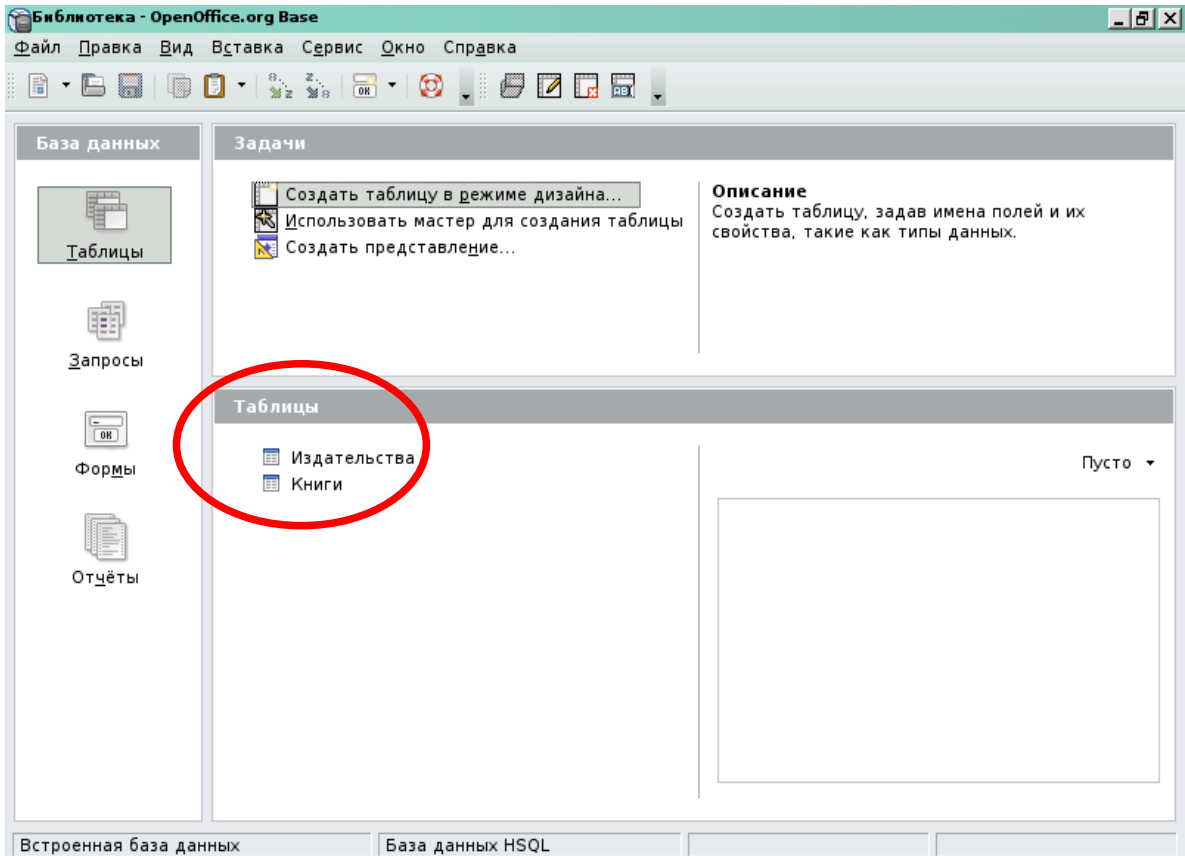


Рис. 25

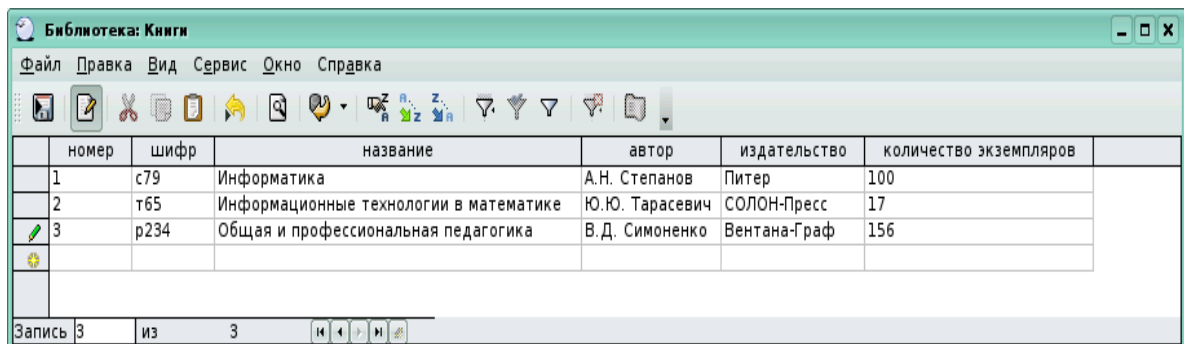


Рис. 26

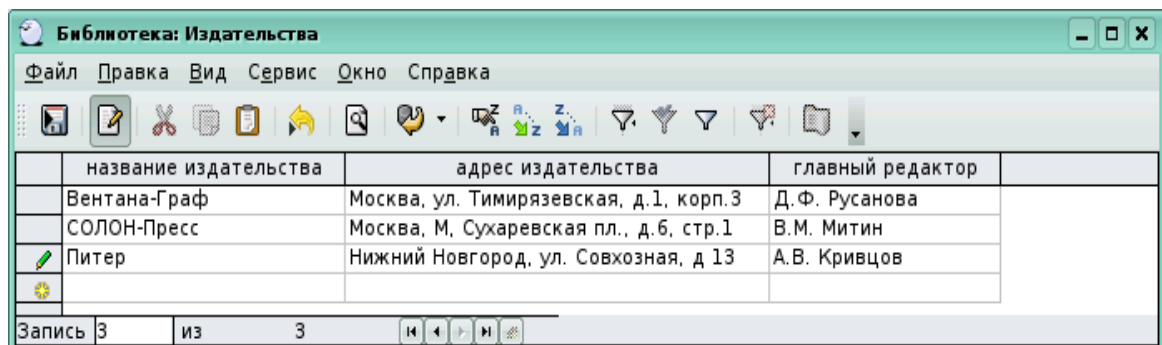


Рис. 27

После того как все таблицы созданы, приступаем к созданию связей. Этот этап подробно рассмотрен в п. 2.6 **Связи**, предлагаем читателю проработать его самостоятельно, результат представлен на рис. 28.

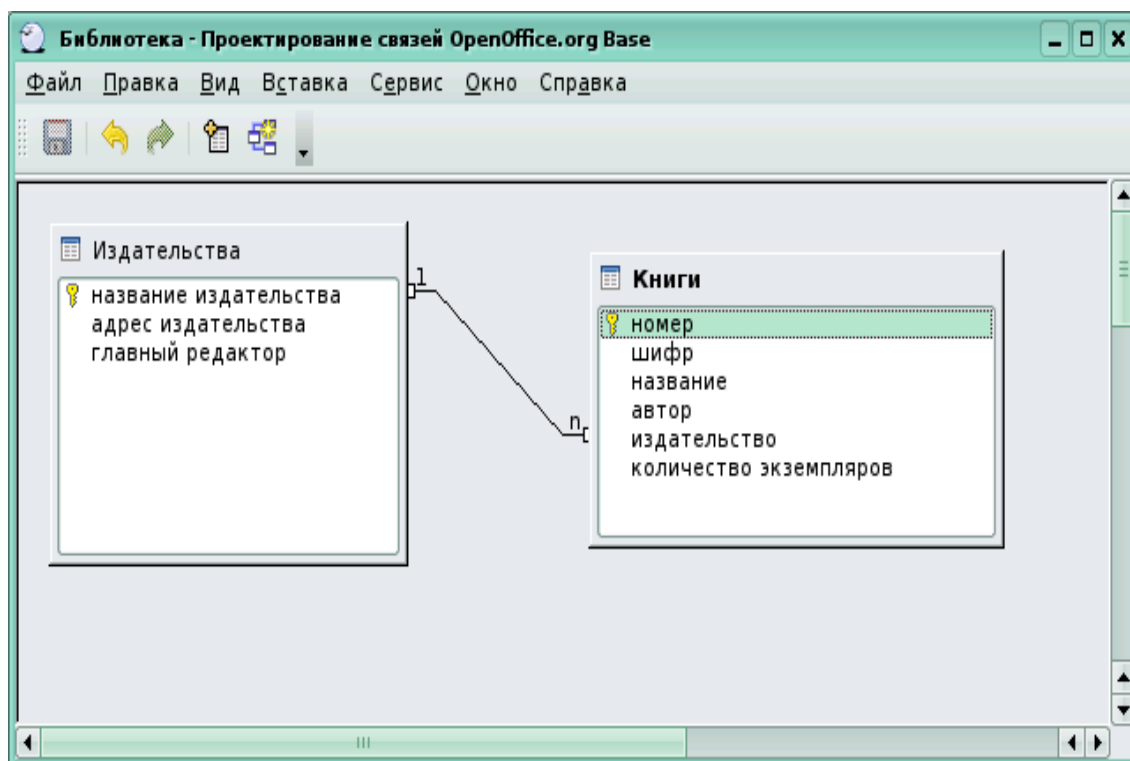


Рис. 28

Теперь приступим к созданию запросов. Для начала создадим запрос сортировки данных по таблице «Книги». Для этого заходим в режим дизайна запросов и добавляем туда таблицу «Книги», после чего выбираем интересующие нас поля и ставим сортировку «по возрастанию» для поля «название» (рис. 29).

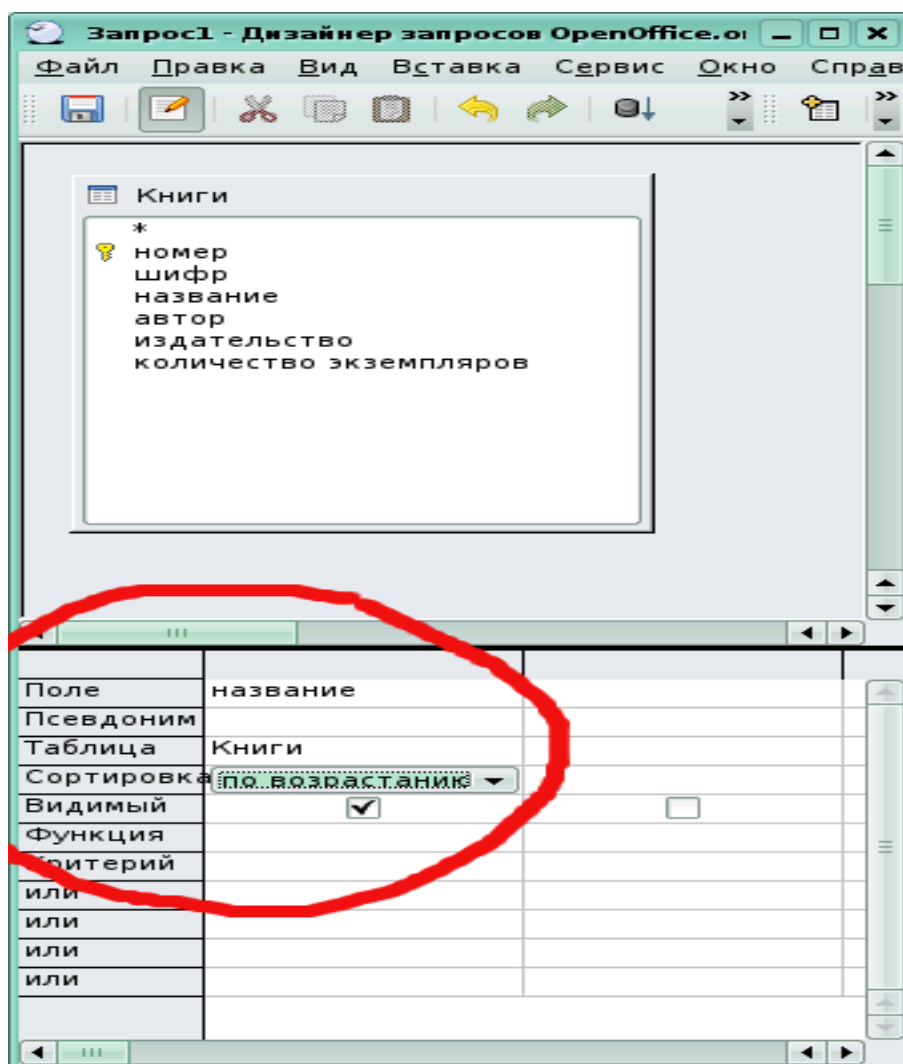


Рис. 29

Результаты данного запроса показаны на рис. 30.

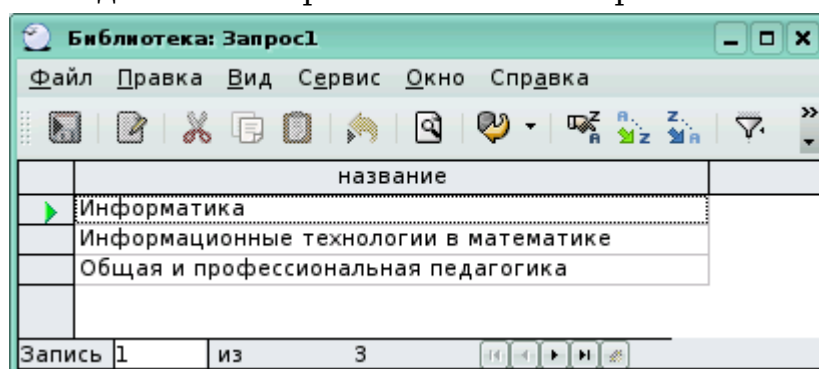


Рис. 30

Для того чтобы знать, сколько различных наименований книг хранится в библиотеке, создадим следующий запрос. В режиме дизайна добавляем таблицу «Книги», для подсчета книг можно воспользоваться полем «номер книги». Выбираем это поле, пишем псевдоним

«количество наименований книг» и выбираем функцию «счетчик (Count)» (рис. 31)

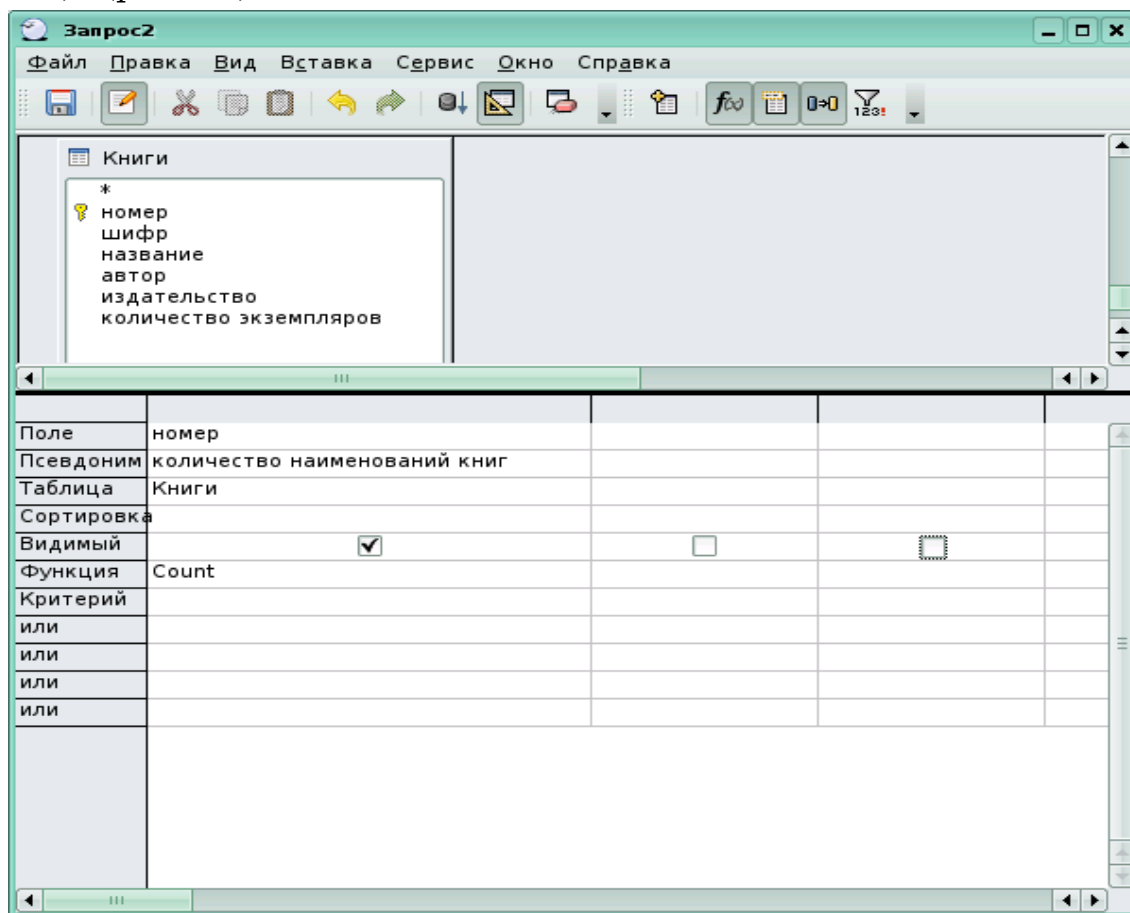


Рис. 31

При этом «Псевдоним» несет смысловую нагрузку. Результат запроса представлен на рис. 32.

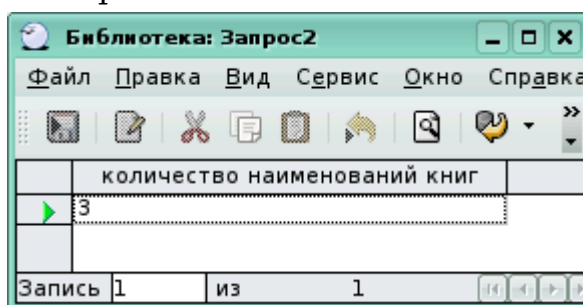


Рис. 32

Далее можно создать форму, формы могут создаваться на основе таблиц или запросов. Мы создадим форму, которая будет отображать все данные о книге с возможностью их добавления.

Открываем **Мастер форм** и приступаем к выполнению шагов.

Шаг 1. Помещаем в форму все поля таблицы «Книги» (рис. 33).

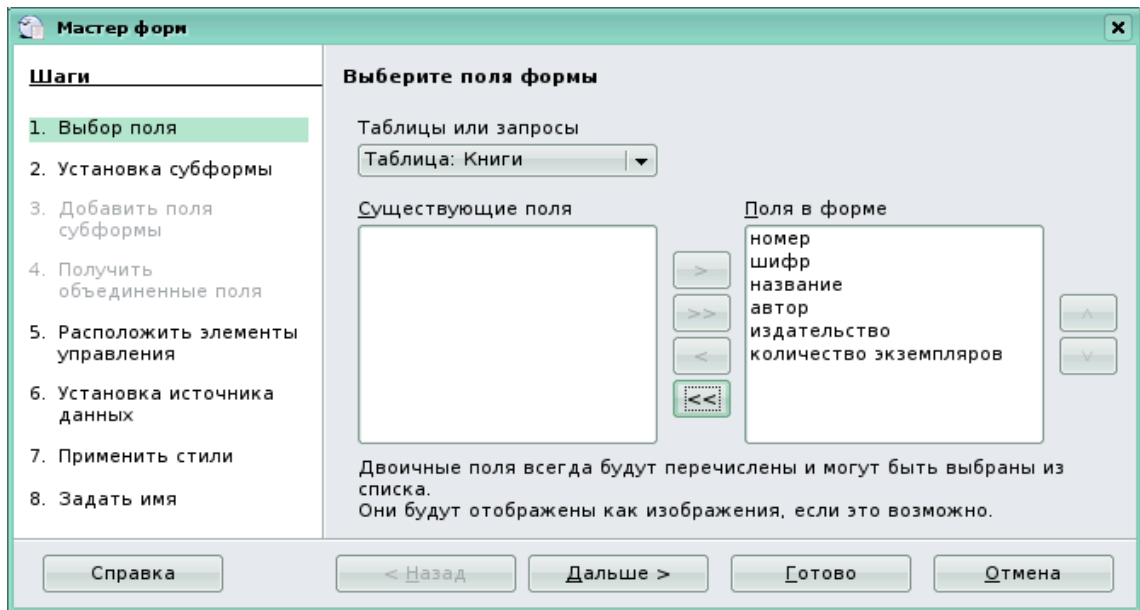


Рис. 33

Шаг 2. Выбираем добавление субформы.

Шаг 3. Выбираем поля для субформы (это будут поля «адрес» и «главный редактор» таблицы «Издательства») (рис. 34).

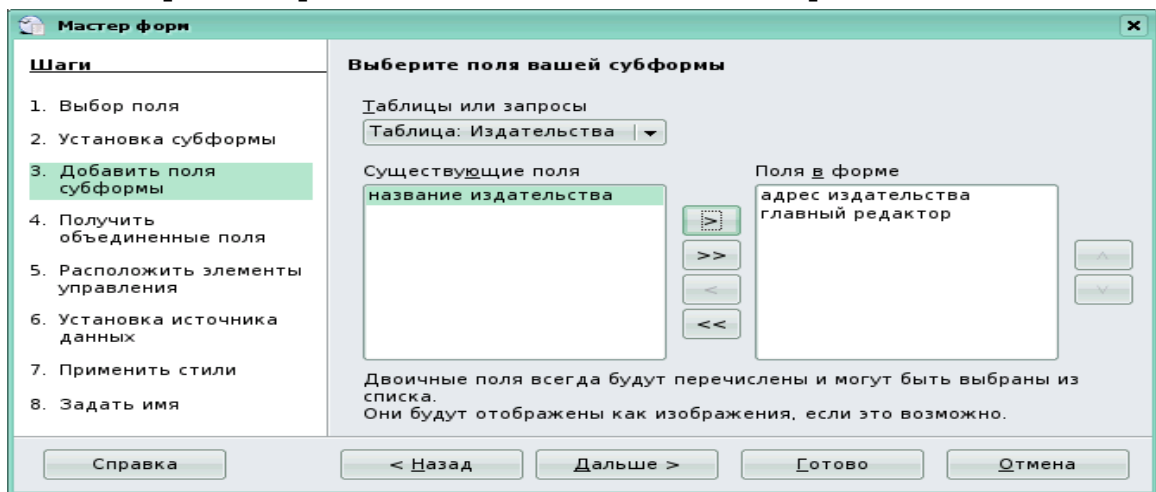


Рис. 34

Шаг 4. На этом шаге не меняем ничего.

Шаг 5. Выбираем расположение элементов на форме. В данном случае столбцы подписи расположим слева и на основной форме и на субформе.

Шаг 6. Здесь задаем условие, что в форме будут отображены все данные, но их нельзя изменять и удалять, можно только добавлять новые данные (рис. 35).

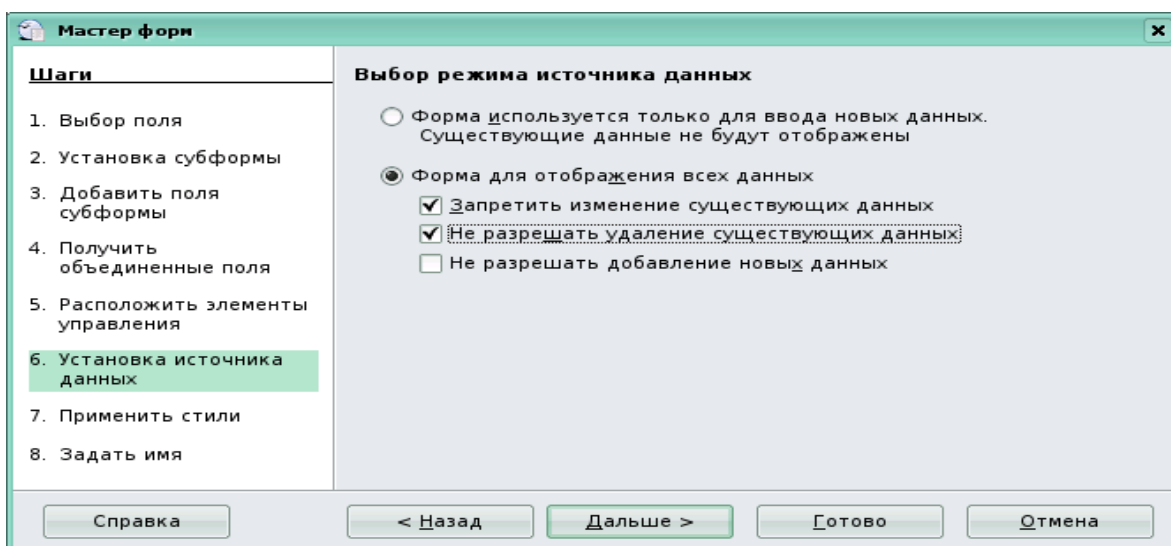


Рис. 35

Шаг 7. Выбираем оформление формы. Мы выбрали светло-синий фон и трехмерный вид поля.

Шаг 8. На заключительном шаге задаем имя формы как «информация о книгах» и отмечаем, что после заполнения формы хотим с ней работать (рис. 36).

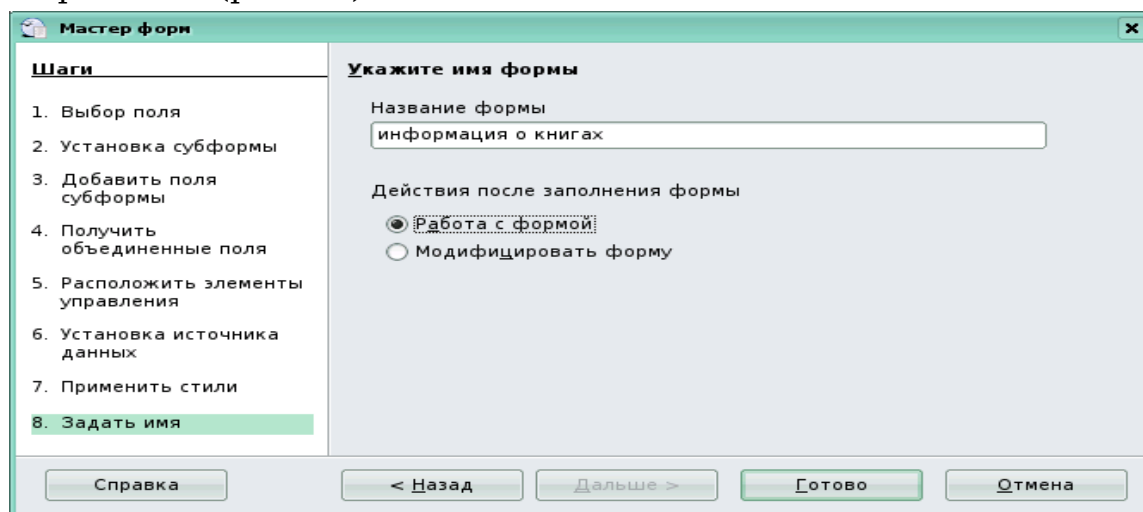


Рис. 36

После завершения последнего шага жмем кнопку «Готово», на этом создание формы закончено (рис. 37).

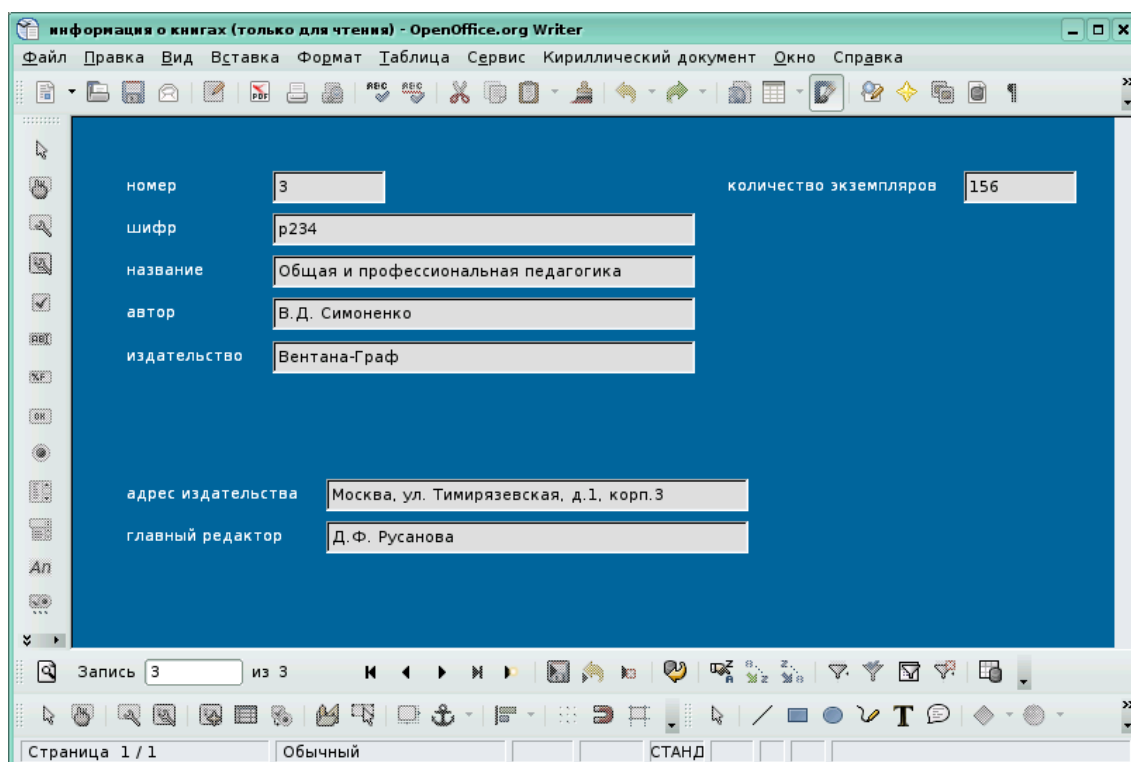


Рис. 37

Форма открывается в виде текстового документа в окне редактора OpenOffice.org Writer. Если читателя заинтересовала работа в этом редакторе, он может обратиться к пособию А.В. Литвиновой «Создание и редактирование текстов в среде OpenOffice.org», разработанному по пакету СПО.

Для создания отчета выбираем **Мастер отчетов**. На рис. 38 приведен пример отчета. Отчеты, построенные на основе таблиц, будут статическими, то есть будут отображать только ту информацию, которая содержалась в таблице во время их создания. Напомним, что есть возможность создать динамический отчет. При этом происходит сохранение отчета в виде шаблона. При открытии динамического отчета он отображается с текущим содержимым.

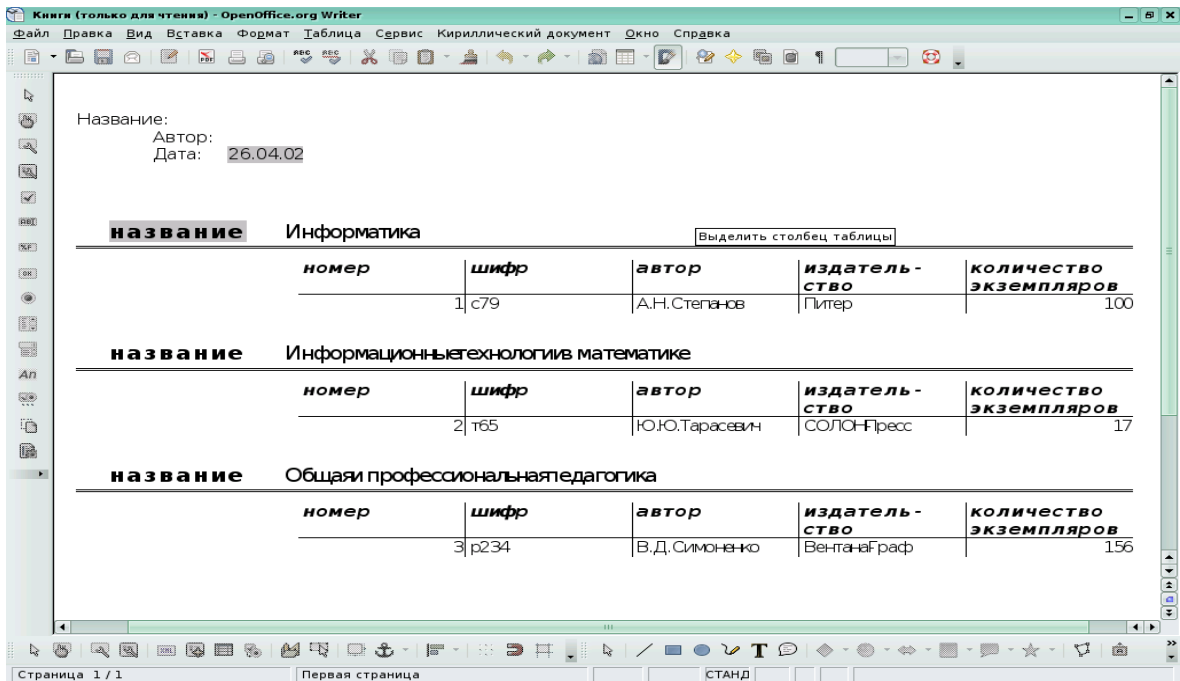


Рис. 38

На этом создание элементарной базы данных «Библиотека» закончено. Естественно, эта база является лишь «пробой пера», но она также является основой, опираясь на которую, можно построить более расширенную и более функциональную базу данных.

Таким образом, мы рассмотрели основные этапы создания базы данных и последующей работы с ней. Для более детального знакомства с процессом проектирования и создания баз данных предлагаем читателю изучить дополнительную литературу из списка.

Вопросы для самоконтроля и задания для самостоятельного выполнения

Вопросы:

1. *Что такое база данных? (выберите правильный вариант)*

- а) реализованная с помощью компьютера информационная структура (модель), отражающая состояния объектов и их отношения;
- б) минимальная именованная структурная единица данных;
- в) совокупность нормализованных отношений, логически взаимосвязанных и отражающих некоторую предметную область;
- г) универсальное программное средство, предназначенное для обработки информации.

2. *Что такое система управления базами данных? (выберите правильный вариант)*

- а) операции обработки данных;
- б) реализованная с помощью компьютера информационная структура (модель), отражающая состояния объектов и их отношения;
- в) совокупность взаимосвязанных структур данных и операций над этими структурами;
- г) комплекс программных и языковых средств, необходимых для создания и модификации базы данных, добавления, модификации, удаления, поиска и отбора информации, представления информации на экране и в печатном виде, разграничения прав доступа к информации, выполнения других операций с базой.

3. *Какой из типов моделей данных не поддерживают СУБД? (выберите неправильный вариант)*

- а) нелинейный;
- б) иерархический;
- в) сетевой;
- г) реляционный.

4. *Какие модели баз данных существуют? (выберите неправильный ответ)*

- а) сетевые;
- б) иерархические;
- в) нелинейные;
- г) реляционные.

5. *Какой программный продукт не является СУБД? (выберите правильный вариант)*

- а) OpenOffice.org Calc;
- б) OpenOffice.org Base;
- в) MS Access;
- г) MS FoxPro.

6. Какую базу данных называют реляционной? (выберите правильный вариант)

- а) совокупность взаимосвязанных структур данных;
- б) универсальное программное средство, предназначенное для обработки информации;
- в) комплекс программных и языковых средств, необходимых для создания и модификации базы данных, добавления, модификации, удаления, поиска и отбора информации, представления информации на экране и в печатном виде, разграничения прав доступа к информации, выполнения других операций с базой;
- г) совокупность нормализованных таблиц, между которыми могут существовать связи по ключевым значениям.

7. Что такое поле таблицы? (выберите правильный вариант)

- а) объекты, которые служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде;
- б) полный набор данных об определенном объекте;
- в) элемент таблицы, который содержит данные определенного рода;
- г) специальная структура, предназначенная только для вывода данных.

8. Какой тип полей не используются в OpenOffice.org Base? (выберите правильный вариант)

- а) символьный;
- б) числовой;
- в) время;
- г) процентный.

9. Что такое первичный ключ? (выберите правильный вариант)

- а) главный ключевой элемент (поле), однозначно идентифицирующий строку в таблице;
- б) совокупность взаимосвязанных структур данных;
- в) элементарная единица логической организации данных;
- г) поле, значение которого может повторяться в нескольких записях базы.

10. Для чего предназначены таблицы? (выберите правильный вариант)

- а) для хранения данных в виде записей (строк) и полей (столбцов);
- б) только для вывода данных;

- в) для ввода данных;
- г) для извлечения данных из таблиц и предоставления их пользователю в удобном виде.

11. Для чего предназначены формы? (выберите правильный вариант)

- а) только для вывода данных;
- б) для ввода данных;
- в) для извлечения данных из таблиц и предоставления их пользователю в удобном виде;
- г) для хранения данных в виде записей (строк) и полей (столбцов).

12. Для чего предназначены запросы? (выберите правильный вариант)

- а) для хранения данных в виде записей (строк) и полей (столбцов);
- б) для извлечения данных из таблиц и предоставления их пользователю в удобном виде;
- в) только для вывода данных;
- г) для ввода данных.

13. Для чего предназначены отчеты? (выберите правильный вариант)

- а) для извлечения данных из таблиц и предоставления их пользователю в удобном виде;
- б) только для вывода данных;
- в) только для ввода данных;
- г) для хранения данных в виде записей (строк) и полей (столбцов);

14. С какого этапа начинается процесс создания базы данных? (выберите правильный вариант)

- а) определение связей между таблицами;
- б) усовершенствование структуры базы данных;
- в) определение цели создания базы данных;
- г) определение полей с уникальными значениями в каждой записи.

15. Для чего организуется связь между таблицами? (выберите неправильный вариант)

- а) для создания форм;
- б) для создания отчетов;
- в) для внесения данных в таблицы;
- г) для создания запросов.

16. В каком режиме могут создаваться отчеты в СУБД OpenOffice.org Base? (выберите правильный вариант)

- а) в режиме Дизайна;

- б) в режиме Мастера создания отчетов;
- в) в режиме Дизайна и в режиме Мастера создания отчетов;
- г) нельзя создавать отчеты.

17. Какие режимы используются в OpenOffice.org Base для создания таблиц? (выберите правильный вариант)

- а) только режим Дизайна;
- б) только режим Мастера;
- в) режимы Мастер, Дизайн, создание представления;
- г) режимы Дизайн, создание представления.

18. Каким требованиям должна удовлетворять таблица, чтобы можно было провести ее нормализацию? (выберите неправильный вариант)

- а) таблица должна содержать нуль или более записей;
- б) все записи таблицы должны иметь одно и то же множество полей, причем одноименные поля должны относиться к одинаковым типам данных;
- в) таблица не должна содержать двух полностью идентичных записей;
- г) все записи таблицы должны иметь одно и то же множество полей, причем одноименные поля должны относиться к различным типам данных.

19. Сколько нормальных форм существует? (выберите правильный вариант)

- а) одна;
- б) три;
- в) пять;
- г) шесть.

20. Что такое первая нормальная форма? (выберите правильный вариант)

- а) каждый атрибут таблицы атомарен и все строки различны;
- б) все атрибуты таблицы являются простыми и каждый неключевой атрибут функционально полно зависит от ключа;
- в) каждый неключевой атрибут нетранзитивно зависит от первичного ключа;
- г) каждый атрибут таблицы атомарен и все строки одинаковы.

21. Может ли OpenOffice.org Base сохранить базу данных в формате Microsoft Access? (выберите правильный вариант)

- а) нет;

б) да.

22. Возможна ли сортировка данных в полях таблицы? (выберите правильный вариант)

а) да;

б) нет.

23. Что означает значение Null поля таблицы? (выберите правильный вариант)

а) значение поля таблицы, показывающее, что информация в данном поле может изменяться;

б) значение поля таблицы, показывающее, что информация в данном поле присутствует;

в) значение поля таблицы, показывающее, что информация в данном поле отсутствует;

г) значение поля таблицы, показывающее, что информация в данном поле может быть числовой.

Задания

Задание 1. Запустить «OpenOffice.org Base», открыть и познакомиться со справочной системой.

Задание 2. Создайте с помощью Мастера таблицу, содержащую поля: КодАдреса, Фамилия, Имя, Телефон, Страна. При создании таблицы с помощью Мастера предлагается более 40 образцов таблиц, предназначенных для использования в личных или деловых целях, каждая из которых содержит соответствующий набор полей. Остается выбрать образцы полей, после чего Мастер автоматически создает таблицу.

Задание 3. В таблицу, созданную при выполнении второго задания, добавьте в режиме дизайна поле Год рождения.

Задание 4. В таблице, созданной при выполнении второго задания, отсортируйте фамилии по алфавиту.

Задание 5. Создайте базу данных «Записная книжка», которая будет содержать в себе следующую информацию: фамилия, имя, отчество, город, адрес, код города, телефон, дата рождения. Разбейте информацию на несколько таблиц, создайте запрос, который будет сортировать фамилии по возрастанию, и запрос, который будет выводить всех людей, проживающих в вашем городе.

Задание 6. Создайте базу данных «Сотрудники», содержащую: ФИО, образование, должность, заработная плата, семейное положение, дети. Разбейте информацию на несколько таблиц, создайте запрос, который выводит 5 сотрудников с самой высокой зарплатой и форму для добавления новых сотрудников и удаления уволенных.

Задание 7. Создайте базу данных «Магазин», содержащую наименование товара, количество на складе, количество проданного, цена, отдел, в котором продается данный товар. Разбейте информацию на несколько таблиц, создайте запрос, который выводит товар, представленный на складе менее пяти единиц.

Задание 8. Создайте базу данных «Видеопрокат», содержащую название фильма, цену, режиссера, жанр, год создания, страну, сдана в прокат или нет. Разбейте информацию на несколько таблиц, с помощью запроса определите, какое количество кассет выдано на руки, создайте запрос, который выводит все фильмы в жанре «комедия».

Задание 9. Создайте таблицу «**Читатели**»

Ключ	Поле	Тип	Размер поля
	ФИО	текстовый	55
	Класс	текстовый	6
	Год рождения	дата	—
	Адрес	текстовый	30
	Телефон	числовой	70

Задание 10. Создайте таблицу «**Абонент**»

Ключ	Поле	Тип	Размер поля
	Номер	числовой	—
	Автор	текстовый	30
	Название книги	текстовый	—
	Год издания	числовой	—

Задание 11. Создайте таблицу «**Учет**»

Ключ	Поле	Тип	Размер поля
	Номер	числовой	—
	ФИО читателя	текстовый	28
	Дата выдачи	дата	—

Задание 12. Создайте форму «**Список читателей**», позволяющую вводить данные в таблицу «**Читатели**».

Задание 13. Установите связи между таблицами:

- а) «**Абонемент**» и «**Учет**», связав поля «**Номер**» и «**Номер**»;
- б) «**Учет**» и «**Читатели**», связав поля «**ФИО читателя**» и «**ФИО**».

Задание 14. Создайте составную форму «**Учет книжного фонда**». При создании формы выберите в качестве базовой таблицы «**Абонент**» и Мастер формы. В качестве основы для подчиненной формы возьмите таблицу «**Учет**».

Поля основной формы	Поля подчиненной формы
Номер, Автор, Название книги	ФИО читателя Дата выдачи

Задание 15. Используя созданные формы, внесите данные в следующие таблицы:

«**Читатели**» — 5 записей.

«**Абонент**» — 10 записей.

«**Учет**» — 4 записи.

Задание 16. Создайте запрос:

- а) дореволюционные издания (на экране отобразятся поля **«Автор»** и **«Название книги»** из таблицы **«Абонент»**)
- б) выданная литература (на экране поля **«Автор»** и **«Название книги»** — таблица **«Абонент»** и **«ФИО читателя»**, **«Дата выдачи»** — таблица **«Учет»**).

Задание 17. В таблице **«Абонент»** создайте новое поле **«Книга выдана»** — тип логический (да/нет). В форме **«Учет книжного фонда»** создайте выключатель, определяющий, выдана книга или находится в фонде. Разрабатывая группу, переключатель задайте для размещения значений (да/нет) поле таблицы **«Абонент»**, **«Книга выдана»**.

Глоссарий

Б

База данных (БД, database) — это реализованная с помощью компьютера информационная структура (модель), отражающая состояния объектов и их отношения.

З

Запросы — объекты, которые служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции, как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразования данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое.

К

Ключевой элемент таблицы (ключ) — такое ее поле (простой ключ) или строковое выражение, образованное из значений нескольких полей (составной ключ), по которому можно определить значения других полей для одной или нескольких записей таблицы. На практике для использования ключей создаются индексы — служебная информация, содержащая упорядоченные сведения о ключевых значениях. В реляционной теории и концептуальной модели понятие «ключ» применяется для атрибутов отношения или сущности.

О

Отчеты — по своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на принтер. В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов.

П

Первичный ключ — главный ключевой элемент, однозначно идентифицирующий строку в таблице. Могут также существовать

альтернативный и уникальный ключи, служащие также для идентификации строк в таблице.

Р

Распределенная база данных — база данных, которая состоит из нескольких, возможно, пересекающихся или даже дублирующих друг друга частей, которые хранятся в различных ЭВМ вычислительной сети.

Реляционная БД — основной тип современных баз данных. Состоит из таблиц, между которыми могут существовать связи по ключевым значениям.

С

Связь — функциональная зависимость между объектами. В реляционных базах данных между таблицами устанавливаются связи по ключам, один из которых в главной (родительской) таблице — первичный, второй — внешний ключ — во внешней (дочерней) таблице, как правило, первичным не является и образует связь "один ко многим" (1:N). В случае первичного внешнего ключа связь между таблицами имеет тип "один к одному" (1:1). Информация о связях сохраняется в базе данных.

Система управления базами данных (СУБД) — комплекс программных и языковых средств, необходимых для создания и модификации базы данных, добавления, модификации, удаления, поиска и отбора информации, представления информации на экране и в печатном виде, разграничения прав доступа к информации, выполнения других операций с базой.

Т

Таблица базы данных — регулярная структура, которая состоит из однотипных строк (записей), разбитых на столбцы (поля).

Таблицы — это основные объекты любой базы данных. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы хранят и структуру базы (поля, их типы и свойства).

Таблица предназначена для хранения данных в виде записей (строк) и полей (столбцов). Обычно каждая таблица используется для хранения сведений по одному конкретному вопросу.

Ф

Файл — совокупность связанных записей, хранящихся во внешней памяти компьютера и рассматриваемых как единое целое. Обычно файл однозначно идентифицируется указанием имени файла, его расширения и пути доступа к файлу.

Формы — средства для ввода данных. Смысл их тот же — предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочее) для автоматизации ввода. Преимущества форм раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают графическими средствами так, чтобы она повторяла оформление бланка — это заметно упрощает работу наборщика, снижает его утомление и предотвращает появление печатных ошибок.

N

Null — значение поля таблицы, показывающее, что информация в данном поле отсутствует. Разрешение на возможность существования значения Null может задаваться для отдельных полей таблицы.

Список литературы

Использованная:

1. Марков А.С. Базы данных. Введение в теорию и методологию. — М.: Финансы и статистика, 2006.
2. Справочная система OpenOffice.org 2.3
3. <http://computerlife.ru/>
4. www.OpenOffice.ru

Рекомендуемая:

1. Ульман Д., Уиндом Д. Введение в системы баз данных. — М.: Лори, 2000.
2. Хансен Г., Хансен Д. Базы данных: разработка и управление. — М.: БИНОМ, 1999.
3. Грабер М. Введение в SQL. — М.: Лори, 1996.
4. Дунаев С. Доступ к базам данных и техника работы в сети. — М.: Диалог, 1999.
5. Вендров А. CASE-технологии. Современные методы и средства проектирования информационных систем. — М.: Финансы и статистика, 1998.
6. Конноли Т., Бегг К., Страчан А. А. Базы данных. Проектирование, реализация и сопровождение. — К.: Вильямс, 2000.
7. Бирн Д. Л. Microsoft SQL Server. Руководство администратора. — М.: Лори, 1998.
8. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. — СПб.: Питер, 2001