

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Р.Е.АЛЕКСЕЕВА»

Институт радиоэлектроники и информационных технологий

Кафедра «Информационные радиосистемы»

**Приобретение практических навыков работы с
системой управления базами данных OpenOffice.org Base для Windows**

Методические указания к лабораторной работе
по дисциплине «Информационные технологии» для студентов
специальности 11.05.01 «Радиоэлектронные системы и комплексы»
дневной формы обучения

Нижний Новгород 2015

Составители Е.Н.Приблудова, В.А. Попов

УДК 621.325.5-181.4

Приобретение практических навыков работы с системой управления базами данных OpenOffice.org Base для Windows: метод. указания к лаб. работе по дисциплине «Информационные технологии» для студентов специальности 11.05.01 «Радиоэлектронные системы и комплексы» дневной формы обучения / НГТУ; Сост.: Е.Н.Приблудова, В.А. Попов. Н.Новгород, 2015, с. 27

Изложены краткие сведения о работе с системой управления базами данных OpenOffice.org Base для Windows. Сформулированы задания и порядок выполнения для лабораторной работы.

Подп. к печ. . Формат $60 \times 84 \frac{1}{16}$. Бумага газетная.

Печать офсетная. Печ.л. . Уч.-изд.л. . Тираж . Заказ .

Нижегородский государственный технический университет им.
Р.Е.Алексеева

Типография НГТУ. 603950, Н.Новгород, ул.Минина, 24.

©Нижегородский государственный технический
университет им. Р.Е.Алексеева, 2015

© Приблудова Е.Н., Попов В.А., 2015

1. Цель работы

Практически научить студентов основным приемам работы с системой управления базами данных для Windows, таким как:

- создание базовых таблиц;
- создание межтабличных связей;
- создание запросов;
- создание сложной формы, состоящей из основной и подчиненной форм;
- создание отчетов.

2. Краткие сведения

2.1 Запуск OpenOffice.org Base для Windows

Рассмотрим один из способов запуска системы управления базами данных OpenOffice.org Base. Нажав кнопку **Пуск**, получаем доступ к главному меню, а через это меню выполняем команду **Все программы** и выбираем необходимую программу - OpenOffice.org. Программа будет загружена, на панели задач появится новая кнопка **Мастер базы данных**, на экране будет открыто окно, в котором необходимо создать новую или открыть созданную базу данных (рис 1). Затем появится окно OpenOffice.org Base с вкладками: таблицы, формы, запросы, отчеты (рис.2).

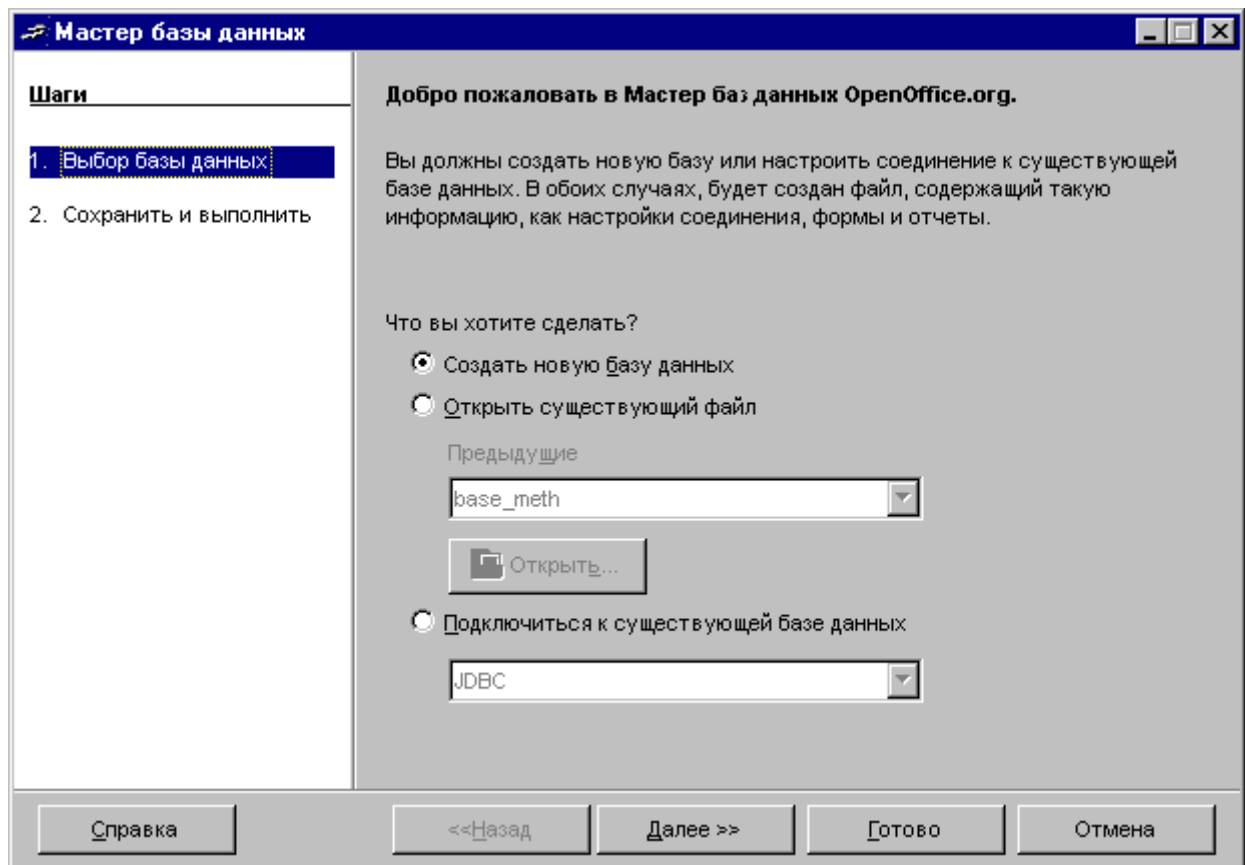


Рис. 1

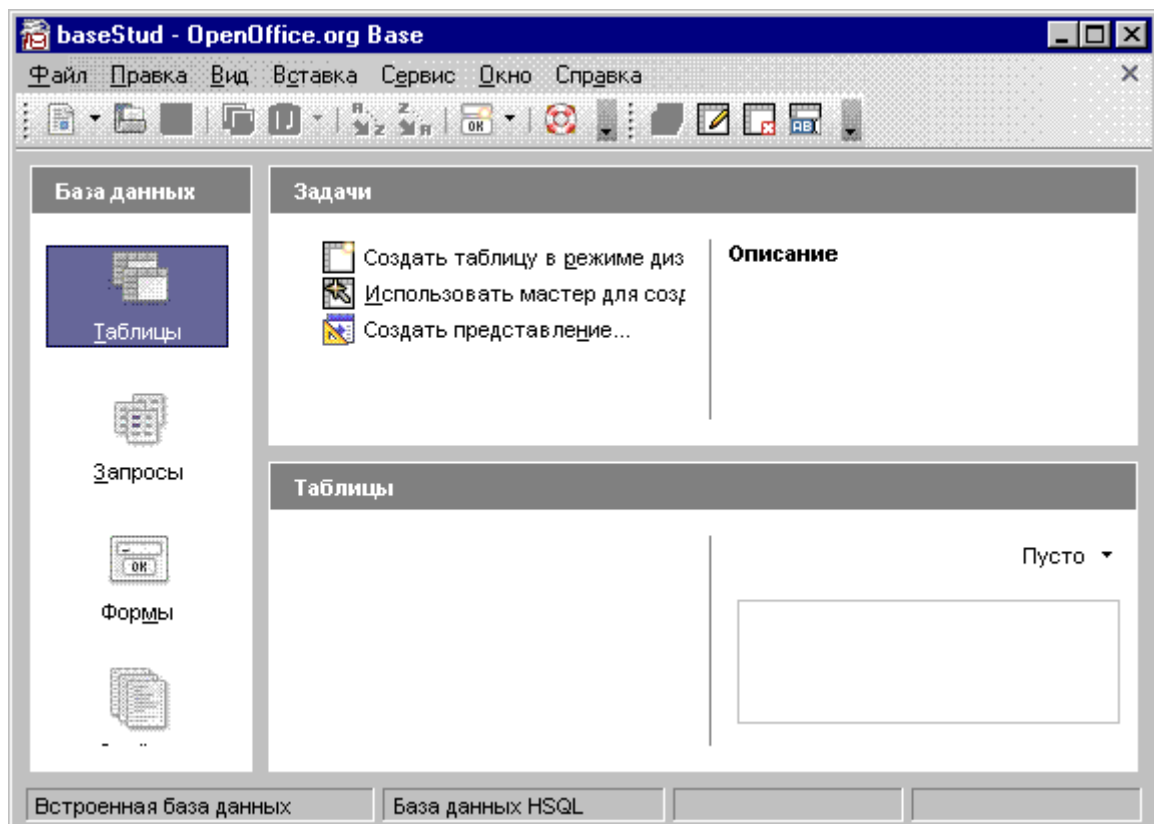


Рис. 2

2.2 Основные понятия OpenOffice.org Base

База данных (БД) – это совокупность сведений, относящихся к определенной теме или задаче.

Реляционные БД – базы данных с табличной формой организации информации. Реляционная БД состоит из одной или нескольких взаимосвязанных двумерных таблиц.

СУБД (Система Управления Базами Данных) – программное обеспечение для работы с базами данных. Большинство современных СУБД предназначены для работы с реляционными базами данных.

Данные – это все, что вы хотите сохранить и к чему намерены обращаться неоднократно. В OpenOffice.org Base данными могут быть тексты, числа, даты и картинки. Если, например, вы продаете книги, то можете хранить их названия, изображения обложек, координаты авторов, количество книг на складе, цены на них, даты продаж.

Запись – строка таблицы. Одна запись содержит информацию об отдельном объекте, описываемом в БД.

Поле – столбец таблицы. Поле содержит определенное свойство объекта. Каждое поле имеет имя. Внутри имени поля нельзя использовать пробелы. Для связки между словами можно ставить знак подчеркивания.

Первичный (главный) ключ БД – это поле, с помощью которого можно однозначно идентифицировать запись.

Тип поля определяет множество значений, которые может принимать данное поле в различных записях. В реляционных базах данных используется четыре основных типа полей: числовой, символьный, дата, логический. Логический тип соответствует полю, которое может принимать всего два значения: "да" – "нет" или "истина" – "ложь".

Таблица базы данных – это совокупность сведений, относящихся к определенной теме или категории объектов.

2.3 Свойства полей БД:

- **Название поля** - определяет, как следует обращаться к данным этого поля при автоматических операциях с базой;
- **Тип поля** – определяет тип данных, которые могут содержаться в данном поле;
- **Длина** – определяет предельную длину (в символах) данных, которые могут размещаться в данном поле;
- **Значение по умолчанию** – то значение, которое вводится в ячейки поля автоматически.

2.4 Заполнение и редактирование БД

После появления окна (рис. 2) необходимо выбрать вкладку **Таблица** и соответствующую ей задачу **Создать таблицу в режиме дизайна....** Далее задать имя, свойство поля и выбрать тип поля из заданного списка.

При создании таблицы целесообразно задать ключевое поле. Это поможет впоследствии, при организации связей между таблицами. Для задания ключевого поля достаточно щелкнуть на его имени правой кнопкой мыши и в открывшемся динамическом меню выбрать пункт **Первичный ключ** (рис. 3). Далее выбрать из меню **Файл** команду **Сохранить как....** Затем аналогично создать другие таблицы.

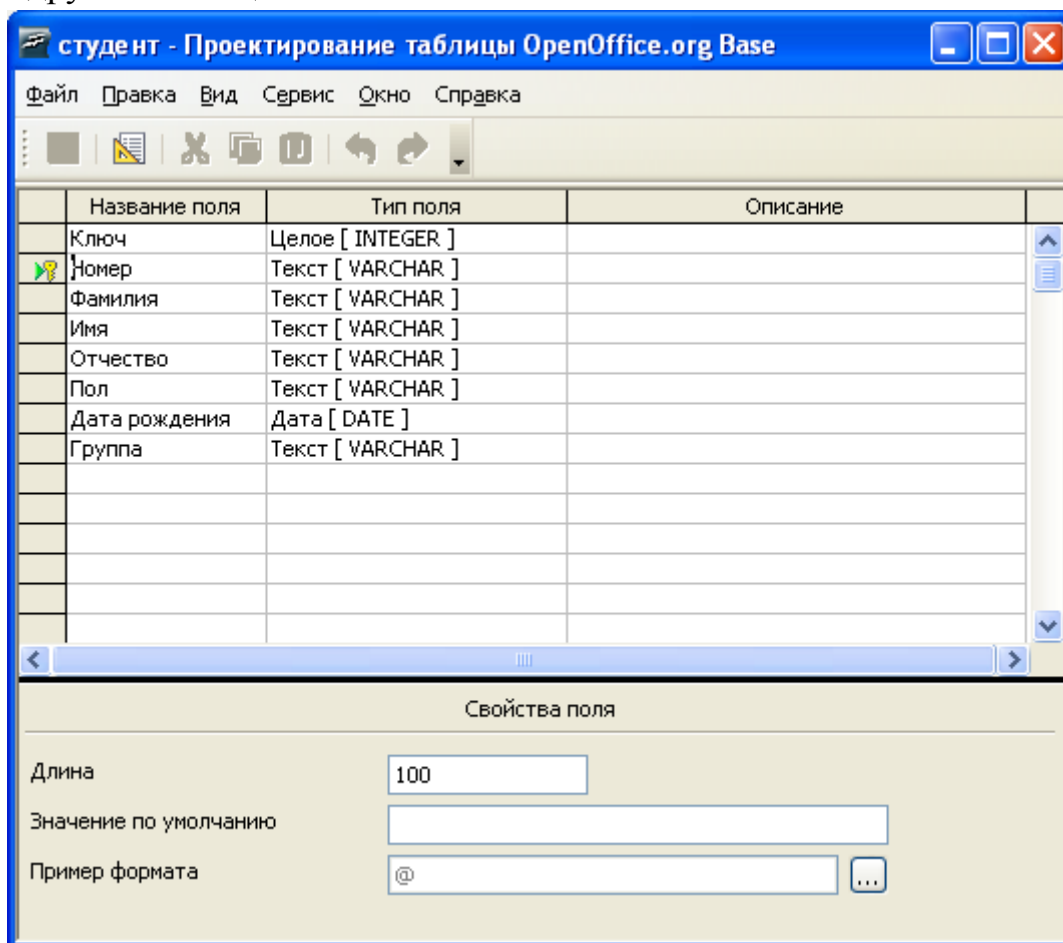


Рис. 3

Для ввода и редактирования записей БД в СУБД есть два режима: таблица и форма.

Режим таблицы позволяет просматривать и редактировать данные, организованные в виде строк и столбцов. Каждый столбец таблицы соответствует одному полю, а каждая строка – одной записи. В режиме таблицы на экран выводится сразу несколько записей БД. Его удобнее использовать для просмотра и удаления записей, а также для ввода информации в таблицы, состоящие из небольшого количества полей.

Режим формы удобно использовать для ввода данных в таблицы с большим числом полей. В этом режиме в большинстве СУБД поля таблицы можно расположить в любом месте экрана. В более простых СУБД существует одна стандартная форма, в которой каждое поле занимает одну или несколько строк. В этом режиме на экран обычно выводится только одна запись таблицы базы данных.

2.5 Извлечение информации из БД

Запрос – это средство извлечения информации из базы данных, отвечающей некоторым условиям, задаваемым пользователем.

Простое логическое выражение представляет собой либо операцию отношения ($>$, $<$, $=$, \neq , \geq , \leq), либо поле логического типа.

Сортировка – процесс упорядочения записей в таблице.

2.6 Проектирование БД

От этапа проектирования зависит дальнейший успех работы с базой. При неправильно спроектированной базе данных придется вносить изменения не только в ее структуру, но и во все уже созданные для этой базы приложения (формы, отчеты, запросы, программы и т.д.).

Первое, что нужно сделать при проектировании БД, - определить список данных, которые необходимо хранить в базе. Следующий шаг – сформировать структуру таблиц (одной или нескольких). База данных, созданная при проектировании, может обладать некоторыми недостатками. Например, может содержать избыточную информацию (например, повторение записей). Для избавления от недостатков база данных должна быть **нормализованной**. Обычно в результате нормализации получается многотабличная БД.


БД может считаться нормализованной, если выполнены следующие условия:

- каждая таблица имеет главный ключ;
- все поля каждой таблицы зависят от главного ключа целиком;
- в таблицах отсутствуют группы повторяющихся значений.

Для успешной работы с многотабличными базами данных обычно требуется установить между ними связи. При установке связей обычно пользуются терминами **базовая таблица** и **подчиненная**. Связь создается парой полей, одно из которых находится в базовой таблице, а другое – в подчиненной. **Базовая** – это таблица, которая участвует в связи своим **Первичным ключом**.

Если связанное поле подчиненной таблицы содержит только уникальные значения, то создается связь **один к одному**. Если это поле может содержать повторяющиеся значения, то создается связь **один ко многим**.

2.7 Пример создания базы данных "Результат сессии" (baseStud.odt)

После создания таблиц (п. 2.4) появятся таблицы с именами **студент**, **сессия**, **стипендия** (рис. 4). Для ввода записей (строк), например, таблицы **студент**, нужно нажать двойным щелчком мыши на ее имени и появится окно с таблицей, в которой заполнены поля. Для изменения имен, типов и свойств полей таблицы необходимо нажать кнопку  - **Правка** на панели инструментов (рис. 4).

Пусть одна таблица содержит данные о студентах – номер зачетной книжки, фамилию, имя, отчество, пол, дату рождения, номер группы (рис. 5). Другая - может быть посвящена результатам сессии (рис. 6). Третья – назначению стипендии (рис. 7).

Из рис. 5 видно, что каждое *поле* представляет собой *столбец* таблицы и содержит определенную категорию информации. Каждая *запись* представляет собой *строку* таблицы и содержит всю информацию о конкретном студенте.

Для того чтобы изменить высоту строк в таблице, используют разделители записей, расположенные с левой стороны таблицы. Для изменения ширины колонки пользуются разделителями полей, которые находятся в верхней части таблицы. Для перехода к следующей записи нажмите клавишу [Tab].

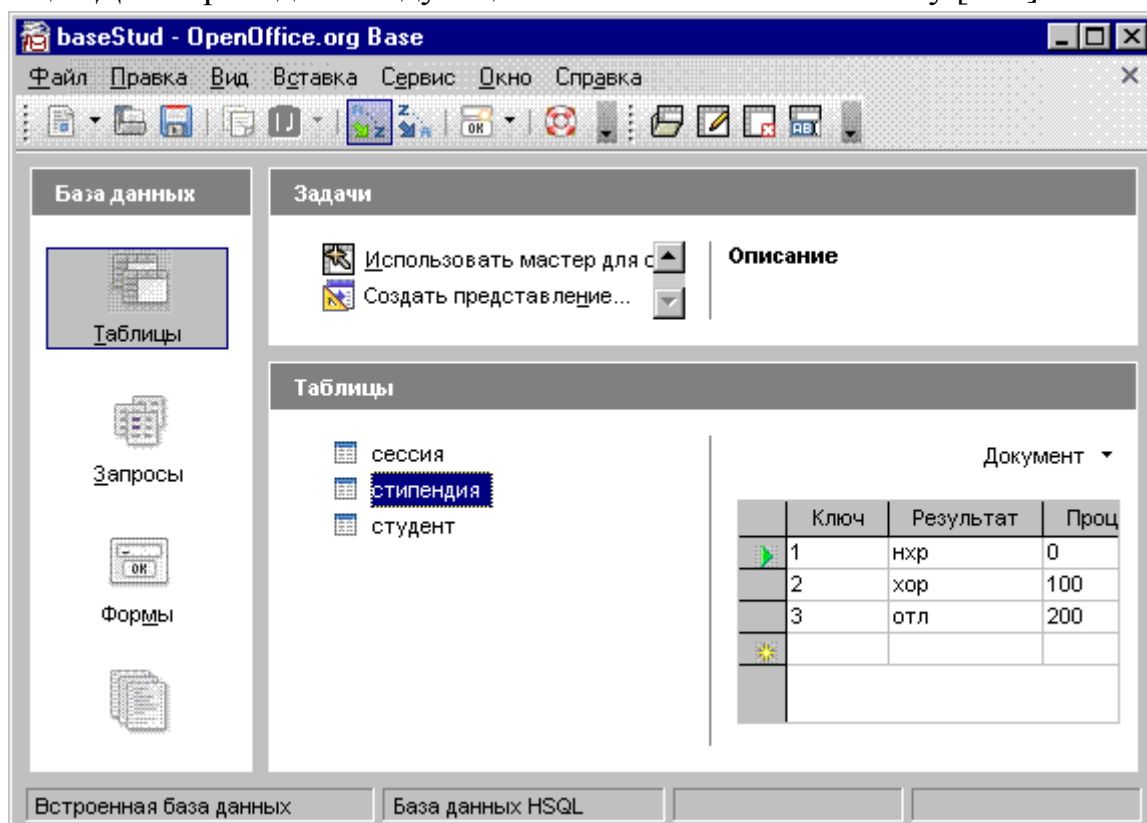


Рис.4

baseStud: студент

Файл Правка Вид Сервис Окно Справка

Ключ	Номер	Фамилия	Имя	Отчество	Пол	Дата рождения	Группа
1	16493	Липов	Андрей	Михайлович	м	01.01.91	111
2	16793	Сиротова	Анна	Владимировна	ж	23.03.90	112
3	16893	Анохин	Андрей	Борисович	м	14.07.91	112
4	16593	Борисова	Мария	Михайловна	ж	05.10.91	111
5	16693	Иванов	Алексей	Иванович	м	20.02.90	112
6	16393	Сидоров	Максим	Алексеевич	м	08.12.91	112
7	16293	Петров	Лев	Викторович	м	03.11.90	111
8	16193	Ляпкина	Ольга	Михайловна	ж	05.06.91	111
9	16093	Исакина	Ирина	Сергеевна	ж	11.03.91	112

Запись 1 из 9

Рис. 5

baseStud: сессия

Файл Правка Вид Сервис Окно Справка

Ключ	Номер	Оценка1	Оценка2	Оценка3	Оценка4	Результат
1	16493	4	5	5	5	хор
2	16793	5	5	5	5	отл
3	16893	4	4	5	4	хор
4	16593	4	4	4	4	хор
5	16693	4	3	3	4	нхр
6	16393	5	3	4	4	нхр
7	16293	3	3	3	3	нхр
8	16193	5	4	5	4	хор
9	16093	5	5	5	5	отл

Запись 1 из 9

Рис. 6

baseStud: стипендия

Файл Правка Вид Сервис Окно Справка

Ключ	Результат	Процент
1	нхр	0
2	хор	100
3	отл	200

Запись 1 из 3

Рис. 7

2.8 Установление связей между таблицами

OpenOffice.org Base имеет дело с базами родственных данных, благодаря чему вы можете одновременно работать с данными из нескольких таблиц. Сформировав таблицы базы данных и установив ключевое поле для каждой таблицы, можно создать между таблицами взаимоотношения. Они помогут **OpenOffice.org Base** связывать данные в каждом новом запросе, форме или отчете, включающем обе связанные таблицы.

В **OpenOffice.org Base** можно создать несколько типов связей. Рассмотрим два типа связи: отношение *один ко многим* и отношение *один к одному*. В первом случае, более распространенном, одна запись таблицы связана с множеством записей другой таблицы.

В случае менее распространенного отношения *один к одному* одной записи в первичной таблице соответствует только одна запись связанной таблицы. Например, фамилия, имя, отчество, пол, дата рождения, группа студента могут располагаться в одной таблице, а данные о результатах сессии - в другой. Между этими таблицами будет существовать отношение *один к одному*, так как каждой записи в одной из них соответствует единственная запись в другой.

2.9 Создание связи между двумя таблицами

Для создания связи между двумя таблицами, следует прежде убедиться, что в них есть совпадающие поля, выбрать базовую таблицу из двух (если речь идет о связи *один ко многим*).

Например, в двух таблицах **студент** (рис. 5) и **сессия** (рис. 6) количество записей одинаково, следовательно, здесь тип связи **один к одному**. Установив **Первичный ключ** в двух таблицах (п. 2.4), их можно связать.

Например, в двух других таблицах: **сессия** (рис. 6) и **стипендия** (рис. 7), количество записей разное, следовательно, здесь тип связи **один ко многим**. В этом случае необходимо выбрать из них базовую таблицу. **Базовая** таблица – это та таблица, в которой нет совпадающих записей в одинаковых полях двух таблиц (совпадающие поля – **Результат**). В данном примере **базовой** будет таблица **стипендия** и только в ней необходимо установить **Первичный ключ** напротив поля **Результат**. Установив **Первичный ключ** в таблице **стипендия**, можно установить тип связи **один ко многим**.

После того как связь установлена, нельзя изменять или удалять поля, на которых она основана, до тех пор, пока не удалите саму связь.

Прежде чем создать связь, необходимо открыть окно **Проектирование связей OpenOffice.org Base**. Для этого выберите команду **Связи...** из меню **Сервис**. Если данное окно открывается не впервые, окно появится точно в таком виде, в каком сохранили его в последний раз. При первом установлении связи окно будет пустым и сначала появится диалоговое окно **Добавление**

таблицы. Затем необходимо добавить все таблицы, которые необходимо связать.

В данном примере в окне **Проектирование связей OpenOffice.org Base** показана взаимосвязь между таблицами базы данных (рис. 8).

Для установления связи между таблицей **студент** и таблицей **сессия** необходимо сделать следующее.

1. Перетащите мышью поле **Номер** из списка полей таблицы **студент** в поле **Номер** из списка полей **сессия**. Отпустив клавишу мыши, появится связь **один к одному**. Теперь таблица **студент** связана с таблицей **сессия**. Видно линию, которая соединяет два совпадающих поля в этих таблицах (рис. 8). Взаимосвязь будет сохраняться до тех пор, пока ее не удалите, нажав на правую кнопку мыши и выбрав команду **Удалить**.
2. Закройте данное окно, при этом система спросит, сохранить ли изменения. Нажмите на кнопку **Да**, чтобы сохранить представление окна взаимосвязей.

Открыв в следующий раз окно **Проектирование связей OpenOffice.org Base**, увидите его таким, каким сохранили.

Для установления связи между таблицей **стипендия** и таблицей **сессия** необходимо перетащить мышью поле **Результат** из списка полей **стипендия** в поле **Результат** из списка полей **сессия**.

Далее действовать аналогичным образом, как и при установлении связи между предыдущими таблицами.

Замечание: Поля связанной таблицы, соответствующие ключевым полям первичной таблицы, могут иметь разные имена. Но они должны содержать совпадающие данные и характеризоваться одинаковым типом данных, а поля данных цифрового типа должны также иметь одинаковый формат.

Исключение составляет случай, когда ключевое поле первичной таблицы по типу данных является счетчиком. Тогда соответствующее поле связанной таблицы может быть тоже счетчиком или характеризоваться цифровым типом данных.

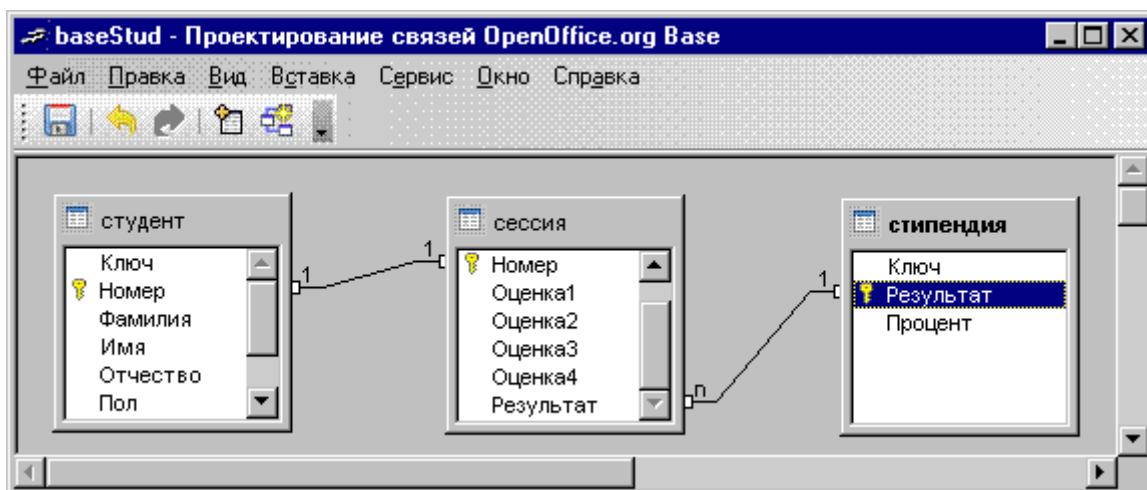


Рис. 8

2.10 Работа с запросами

Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. Запрос определяет группу записей, с которыми необходимо работать. Можно применить запрос для сортировки данных или для просмотра целого подмножества сведений из вашей базы данных. Например, вместо того, чтобы просматривать всю таблицу **студент**, можно просмотреть сведения только о студентах, получивших удовлетворительные оценки (рис. 11) или о студентах, получивших оценки хорошо или отлично (рис. 13).

Для создания запроса необходимо выбрать вкладку **Запрос** и соответствующую ей задачу **Создать запрос в режиме дизайна....** После данного выбора появится окно с возможностью добавления таблиц, по которым будет осуществлен запрос. Выбрав необходимые таблицы, в верхней области (рис. 9) откроются те таблицы, к которым обращен запрос. Сформировав структуру запроса, его закрывают, дают ему имя и в дальнейшем запускают двойным щелчком в окне **База данных**. Для исправления запроса необходимо через меню **Правка** выбрать команду **Правка....**

В нижней части бланка имеется специальная строка **Сортировка**. При щелчке на этой строке открывается кнопка раскрывающегося списка, в котором можно выбрать метод сортировки: по возрастанию или по убыванию.

В нижней части бланка имеется специальная строка **Критерий**. Для каждого поля в этой строке можно задать индивидуальное условие, например, на рис. 10 введено условие **0** в столбце **Процент**, на рис. 12 введено условие **>0**.

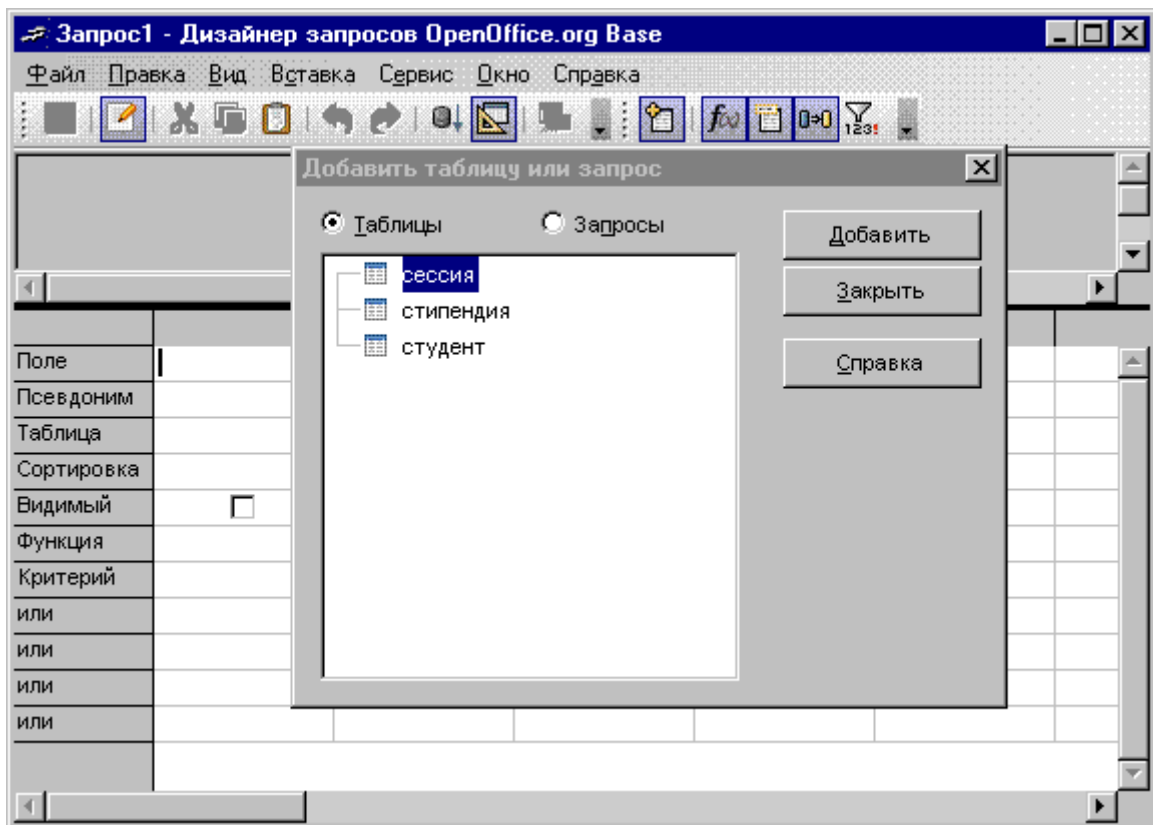


Рис. 9

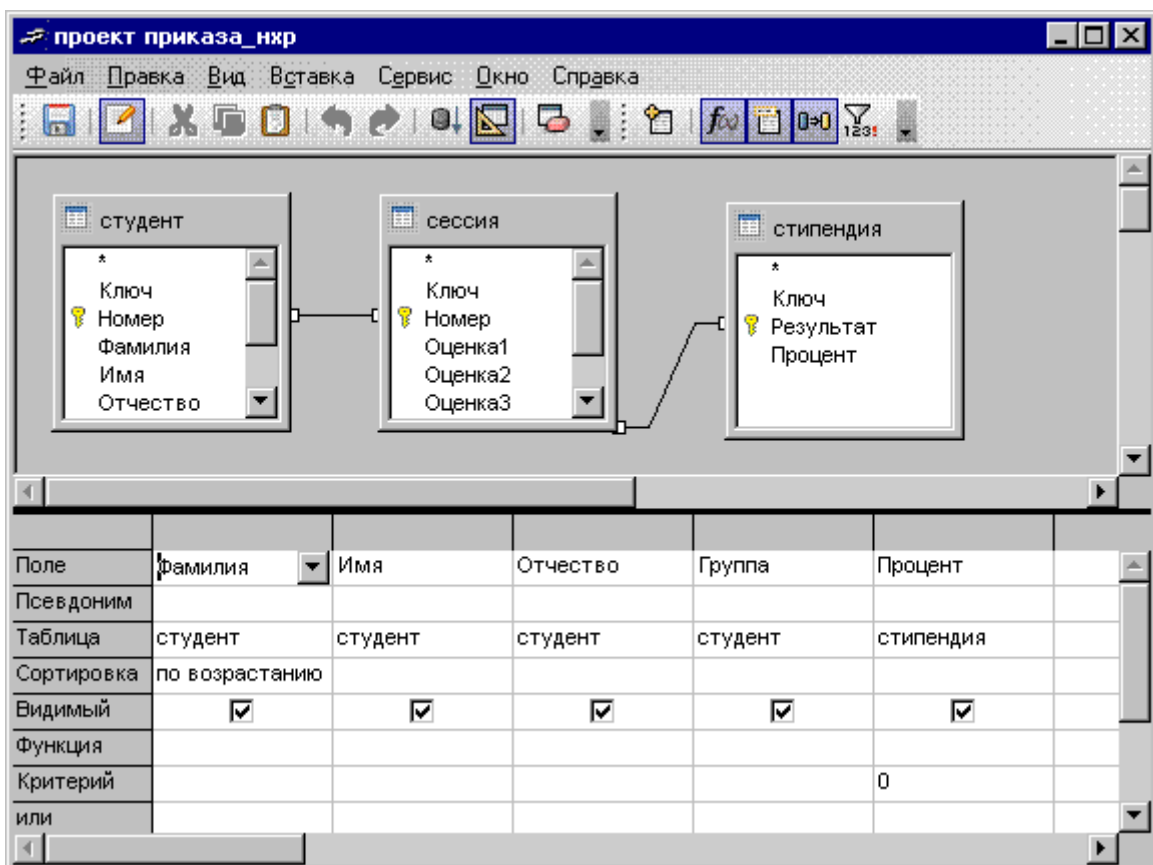


Рис. 10

Фамилия	Имя	Отчество	Группа	Процент
Иванов	Алексей	Иванович	112	0
Петров	Лев	Викторович	111	0
Сидоров	Максим	Алексеевич	112	0

Запись 1 из 3

Рис. 11

Поле	Фамилия	Имя	Отчество	Группа	Процент
Псевдоним					
Таблица	студент	студент	студент	студент	стипендия
Сортировка	по возрастанию				
Видимый	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Функция					
Критерий или					> 0

Рис. 12

Фамилия	Имя	Отчество	Группа	Процент
Анохин	Андрей	Борисович	112	100
Борисова	Мария	Михайловна	111	100
Исакина	Ирина	Сергеевна	112	200
Липов	Андрей	Михайлович	111	100
Ляпкина	Ольга	Михайловна	111	100
Сиротова	Анна	Владимировна	112	200

Запись 1 из 6

Рис. 13

2.11 Создание запросов с использованием языка SQL

SQL (Structured Query Language) – структурный язык запроса, который используется при работе с реляционными базами данных в современных СУБД.

Оператор **SELECT** определяет структуру данных источника передаваемых записей – имена таблиц и полей, содержащих исходные данные для загрузки в таблицу.

Оператор **FROM** позволяет указать имена исходных таблиц, участвующих в формировании выборки.

Оператор **WHERE** задает условия выполнения подзапроса. Оператор **WHERE** может содержать выражения, связанные логическими операторами, с помощью которых задаются условия выборки.

Для создания запроса в SQL необходимо:

- выбрать задачу **Создать запрос в SQL представлении...** в основном окне базы данных;
- ввести операторы языка SQL (например, **SELECT**, **WHERE**, **FROM**) и структуру данных источника в появившемся окне;
- сохранить запрос.

Для появления окна с результатом необходимо выбрать созданный запрос.

Для редактирования запроса необходимо выбрать его и затем через динамическое меню выбрать команду **Редактировать в режиме SQL...**

Пример 1. Создадим запрос, объединяющий две таблицы. Предположим необходимо создать запрос об оценках, полученных на экзаменах студентами двух групп с использованием языка запроса SQL на основе двух связанных таблиц **студент** и **сессия**. На рис. 14 представлен запрос на языке SQL, а на рис. 15 - результат запроса **Запрос_объединение_двух_таблиц**.

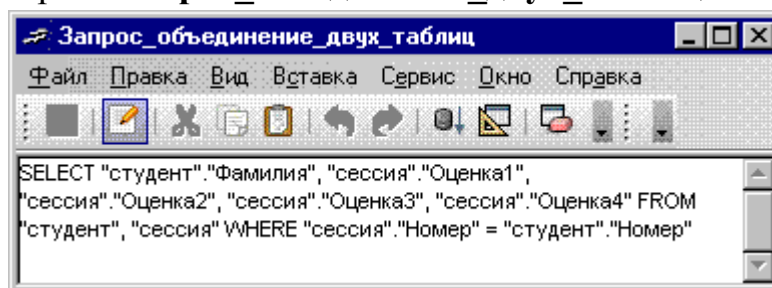


Рис. 14

Фамилия	Оценка1	Оценка2	Оценка3	Оценка4
Анохин	4	4	5	4
Борисова	4	4	4	4
Иванов	4	3	3	4
Исакина	5	5	5	5
Липов	4	5	5	5
Ляпкина	5	4	5	4
Петров	3	3	3	3
Сидоров	5	3	4	4
Сиротова	5	5	5	5

Рис. 15

Пример 2. Создадим запрос, объединяющий более двух таблиц. Предположим, что необходимо найти всех студентов в группе 111, которые будут получать стипендию на основе трех связанных таблиц: **студент**, **сессия**, **стипендия**. На рис. 16 представлен запрос на языке SQL, а на рис. 17 - результат запроса **Запрос_объединение_трех_таблиц**.

```

SELECT "студент"."Фамилия", "студент"."Группа",
"сессия"."Результат", "стипендия"."Процент"
FROM "студент", "сессия", "стипендия"
WHERE "сессия"."Номер" = "студент"."Номер"
AND "сессия"."Результат" =
"стипендия"."Результат"
AND "студент"."Группа" = '111'
AND "стипендия"."Процент" > 0

```

Рис. 16

Фамилия	Группа	Результат	Процент
Ляпкина	111	хор	100
Липов	111	хор	100
Борисова	111	хор	100

Рис. 17

Пример 3. С помощью языка SQL можно вкладывать запросы, т.е. в оператор **SELECT** может быть вложен другой оператор **SELECT**. Обычно внут-

ренний запрос генерирует значение, которое проверяется в предикате внешнего запроса. Чтобы оценить внешний (основной) запрос, SQL сначала должен оценить внутренний запрос (подзапрос) внутри предложения **WHERE**. Язык SQL делает это так, как и должен делать запрос, имеющий единственную цель – отыскать через таблицу **студент** строку, где поле **Фамилия** равно значению **Липов**, а затем извлечь значение поля **Номер** этой строки (рис. 18). Однако SQL не просто выдает значение, а помещает его в предикат основного запроса вместо самого подзапроса, так чтобы предикат прочитал, что **WHERE Номер = 16493**. На рис. 19 представлен результат запроса **Запрос_Липов**.

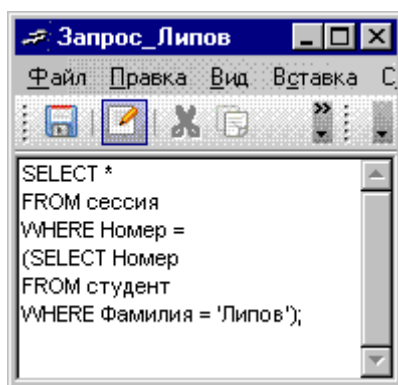


Рис. 18

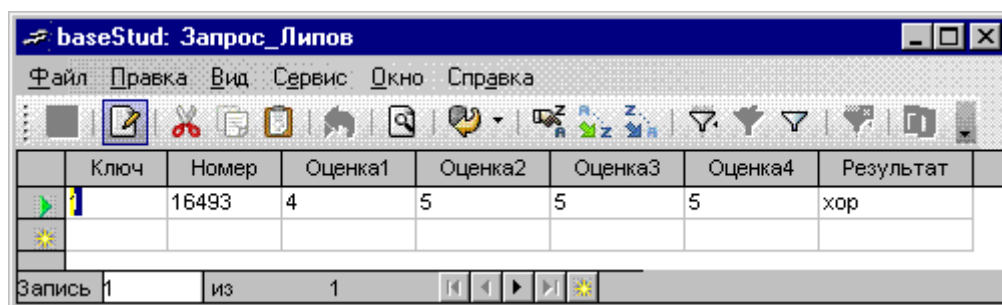


Рис. 19

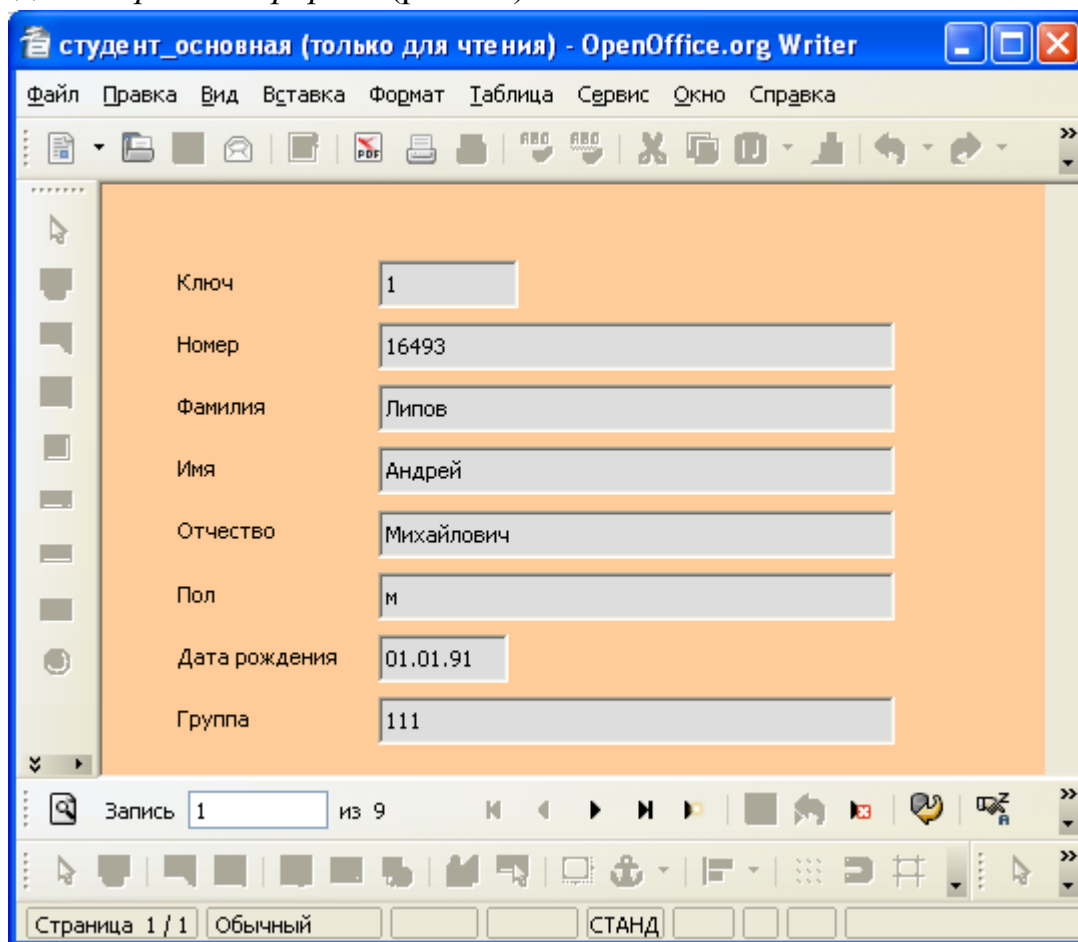
2.12 Работа с формами

С одной стороны, формы позволяют пользователям вводить данные в таблицы базы данных без непосредственного доступа к самим таблицам. С другой стороны, они позволяют выводить результаты работы запросов не в виде результирующих таблиц, а в виде красиво оформленных форм. Также возможен и комбинированный подход – это вопрос творчества.

Форма, которая основана на одном объекте, называется *простой формой*. Форма, основанная на полях из нескольких связанных таблиц, называется *сложной* и представляет собой композицию из нескольких форм.

Для создания *простой формы* необходимо выбрать вкладку **Форма** и соответствующую ей задачу **Использовать мастер для создания формы....** После выбора данной задачи появятся два окна: **OpenOffice.org Write** и **Мастер**

форм. В окне **Мастер форм** необходимо пройти только шаги: 1, 5-8 создания формы. После последнего шага в окне **OpenOffice.org Write** появится результат создания *простой формы* (рис. 20).



The screenshot shows the OpenOffice.org Writer window titled "студент_основная (только для чтения) - OpenOffice.org Writer". The menu bar includes "Файл", "Правка", "Вид", "Вставка", "Формат", "Таблица", "Сервис", "Окно", and "Справка". The toolbar contains various icons for file operations and editing. The main content area displays a form with the following fields:

Ключ	1
Номер	16493
Фамилия	Липов
Имя	Андрей
Отчество	Михайлович
Пол	м
Дата рождения	01.01.91
Группа	111

The status bar at the bottom shows "Запись 1 из 9", navigation buttons, and "Страница 1 / 1 Обычный СТАНД".

Рис. 20

Представление в форме связанных данных

Может возникнуть необходимость работать с формой, которая содержит информацию сразу из двух таблиц или из таблицы и запроса. В этом случае необходимо создать **форму с подчиненной формой**, которая представляет собой форму внутри другой формы. В большинстве случаев **подчиненная форма** связана с **основной**. Форма, которая должна быть подчиненной, может иметь вид таблицы.

Для создания *сложной формы* необходимо выбрать вкладку **Форма** и соответствующую ей задачу **Использовать мастер для создания формы...**. В окне **Мастер форм** необходимо пройти все шаги создания формы (рис. 21 – рис. 28). После последнего шага в окне **OpenOffice.org Write** появится результат создания *сложной формы* (рис. 29).

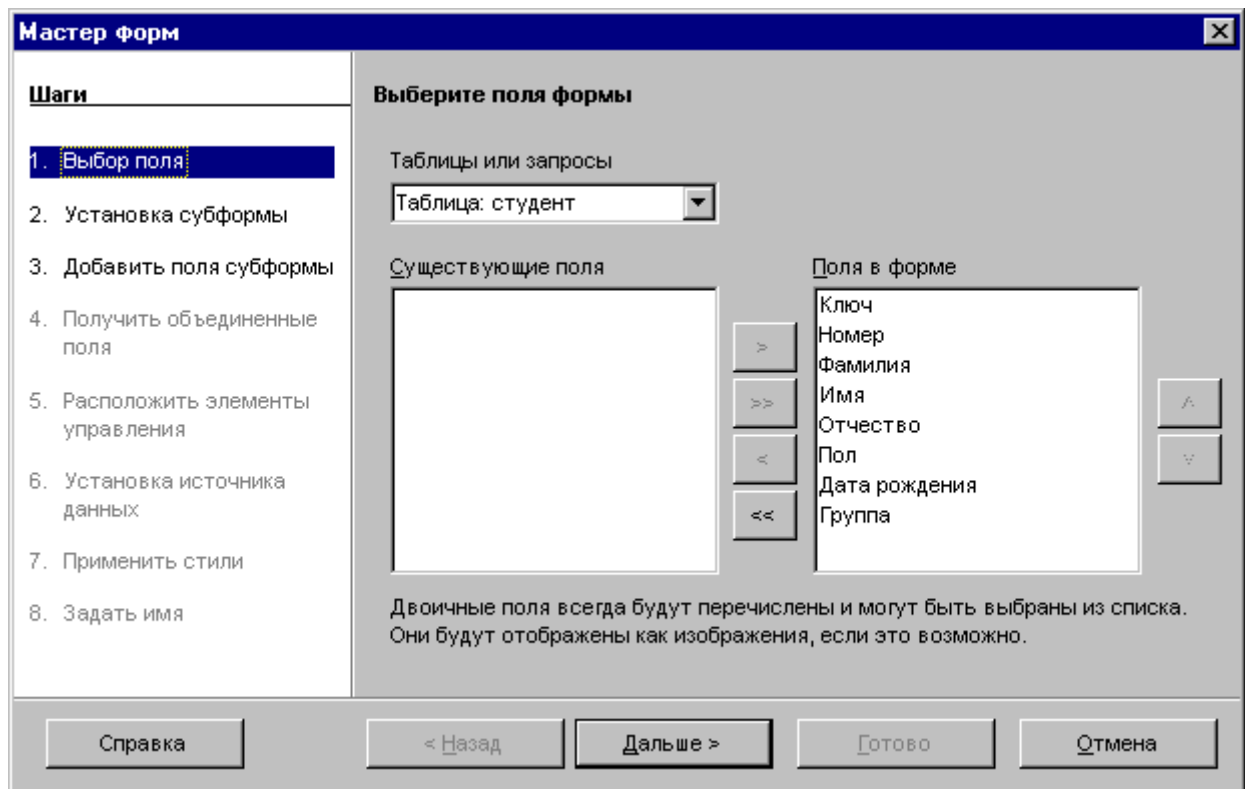


Рис. 21

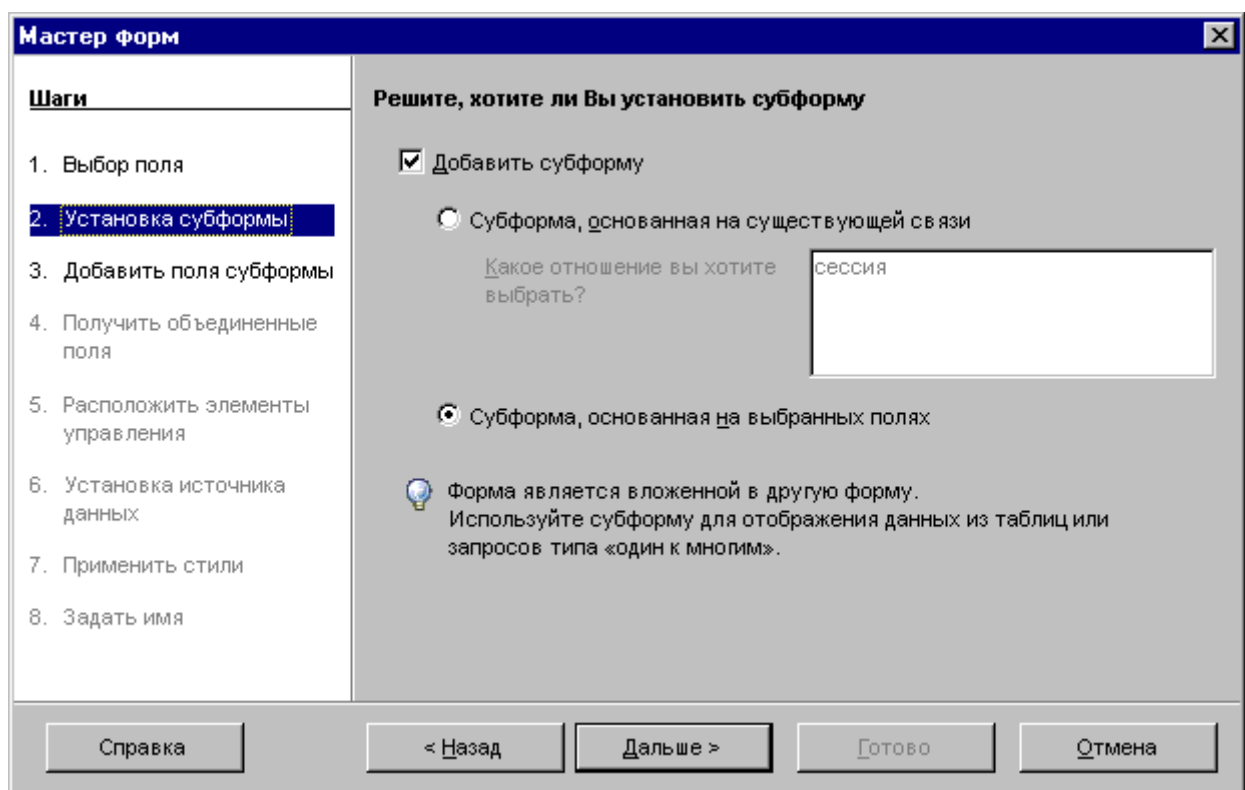


Рис. 22

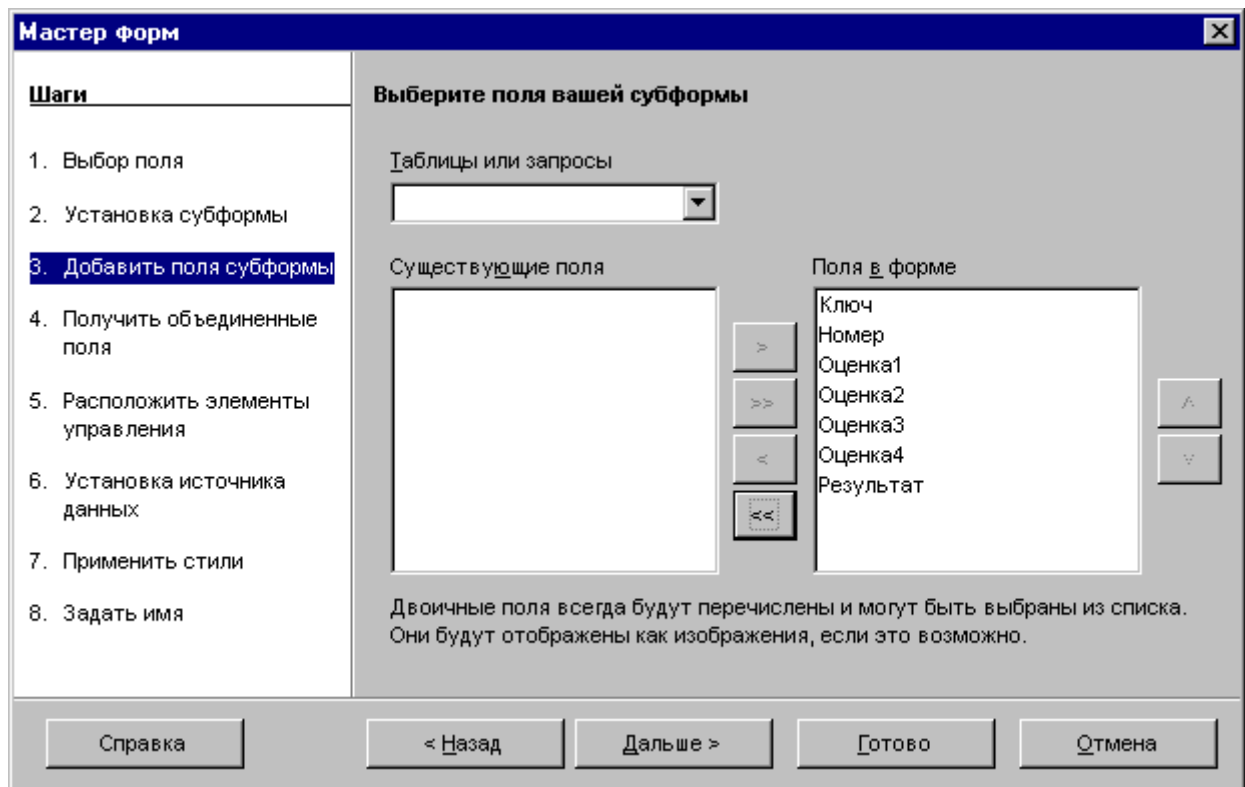


Рис. 23

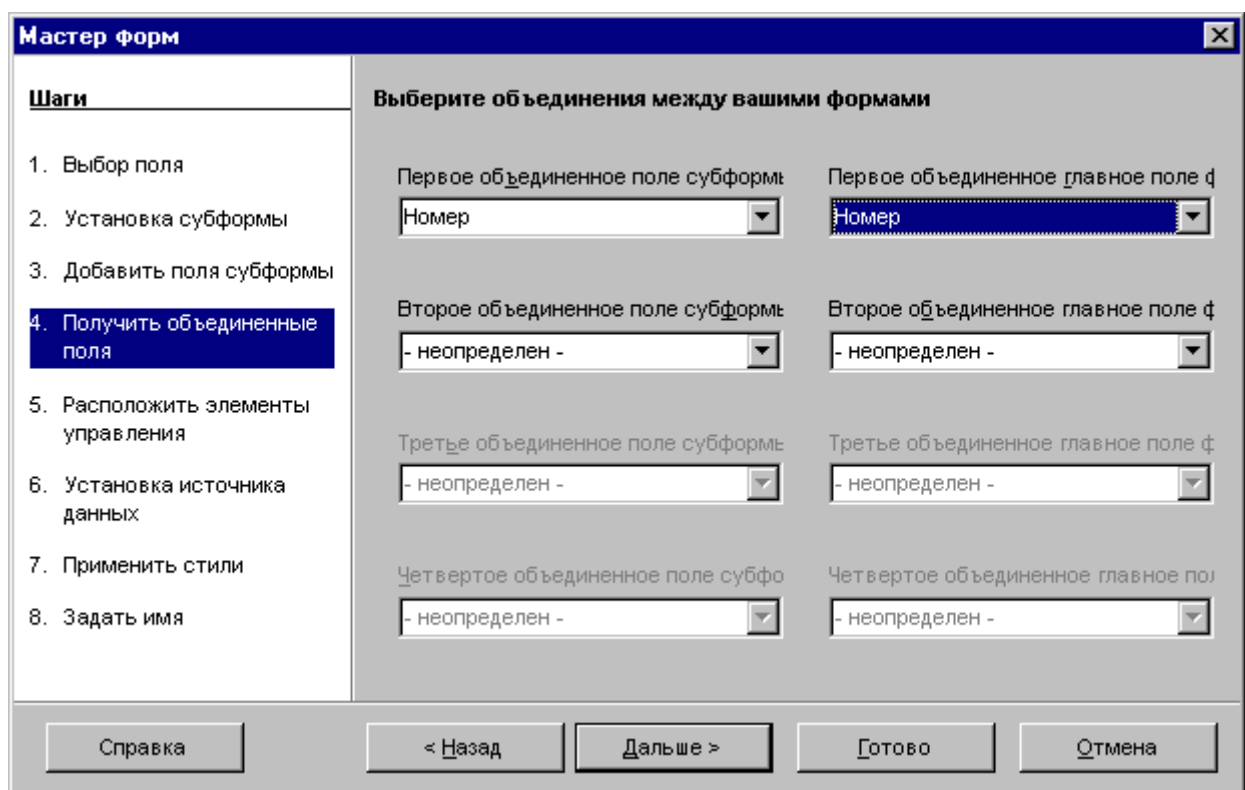


Рис. 24

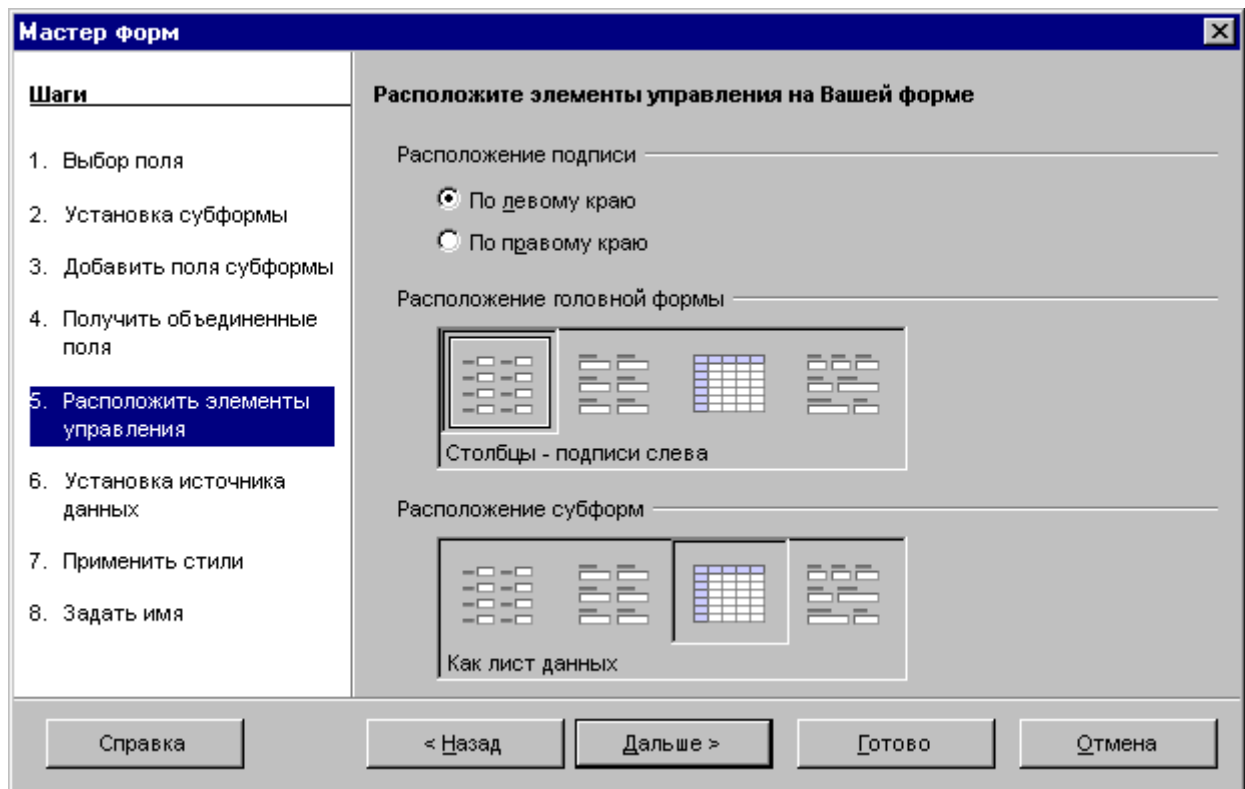


Рис. 25

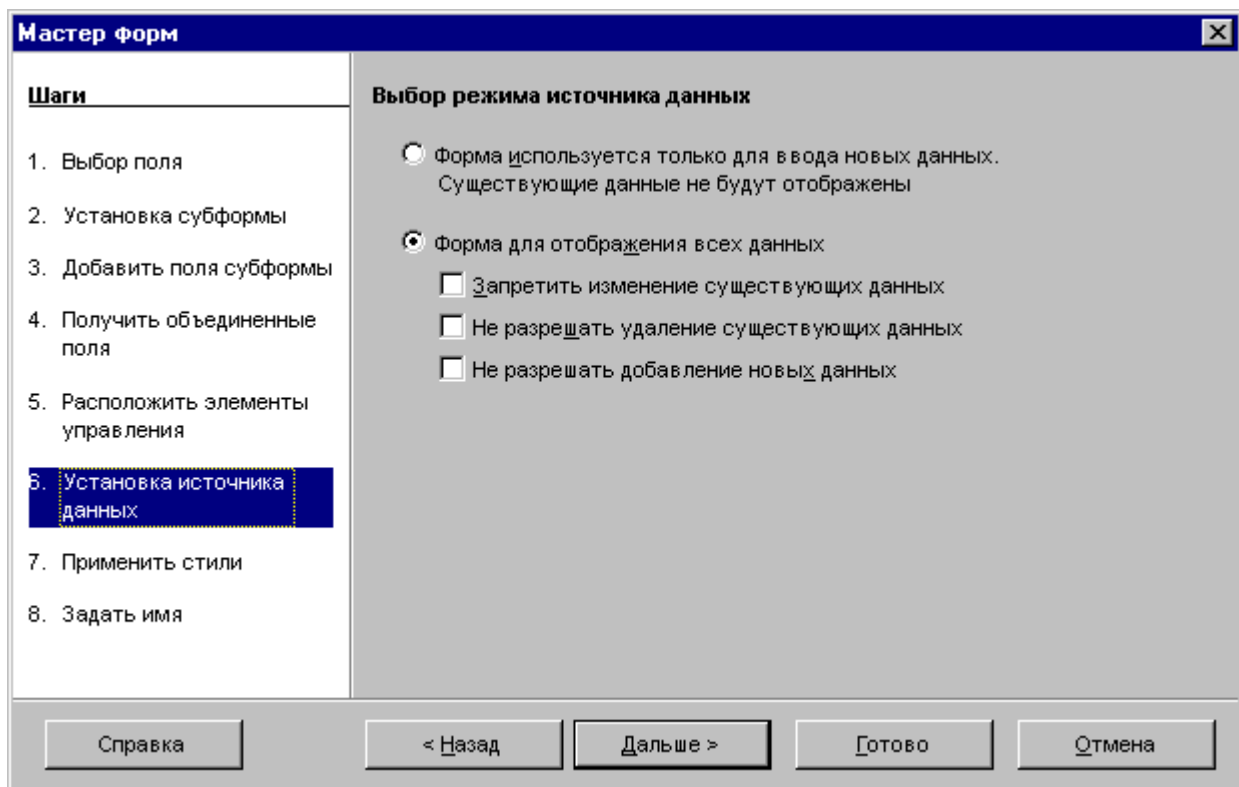


Рис. 26

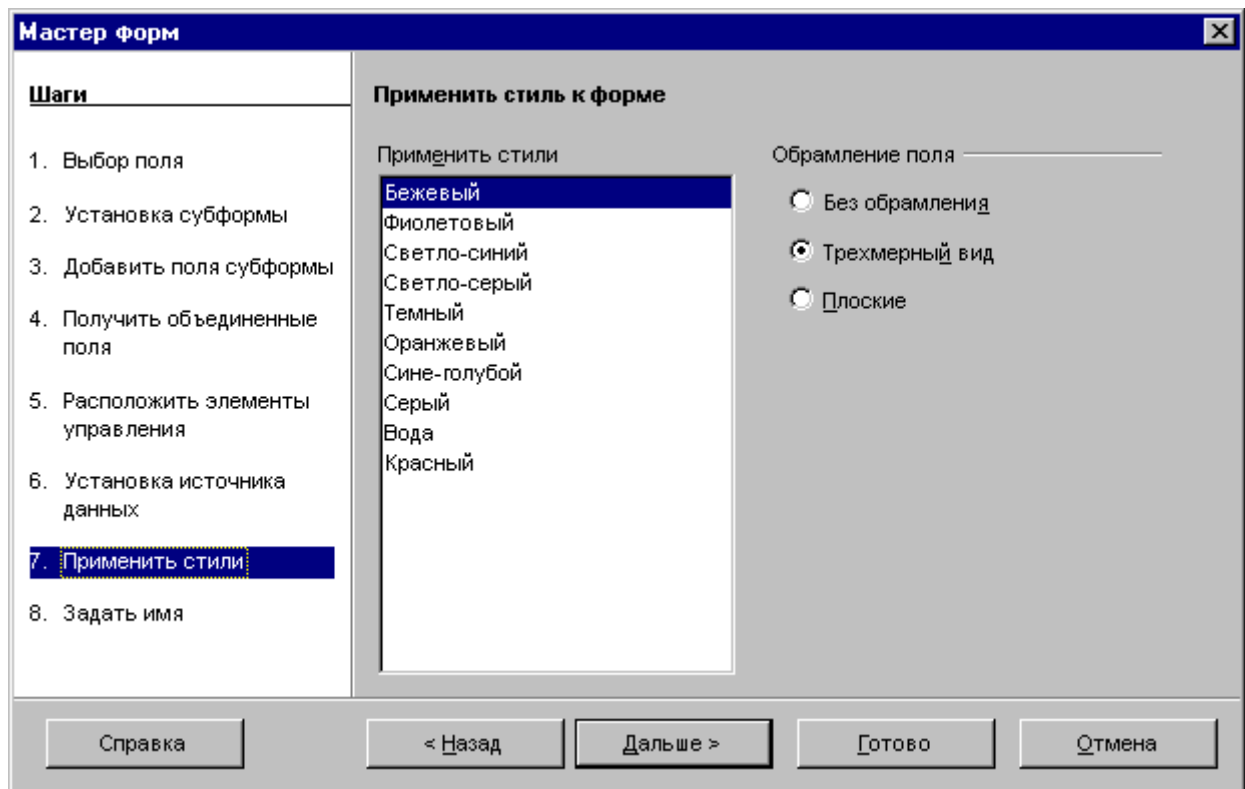


Рис. 27

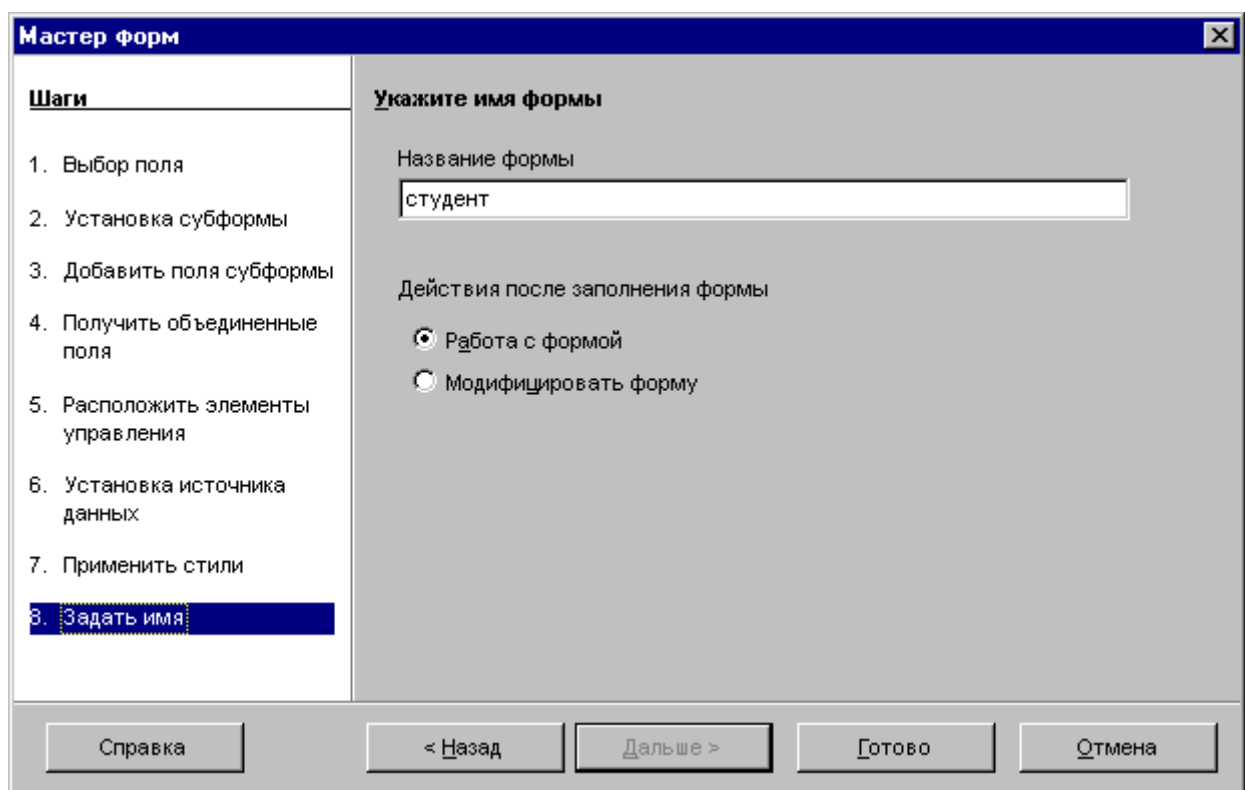


Рис. 28

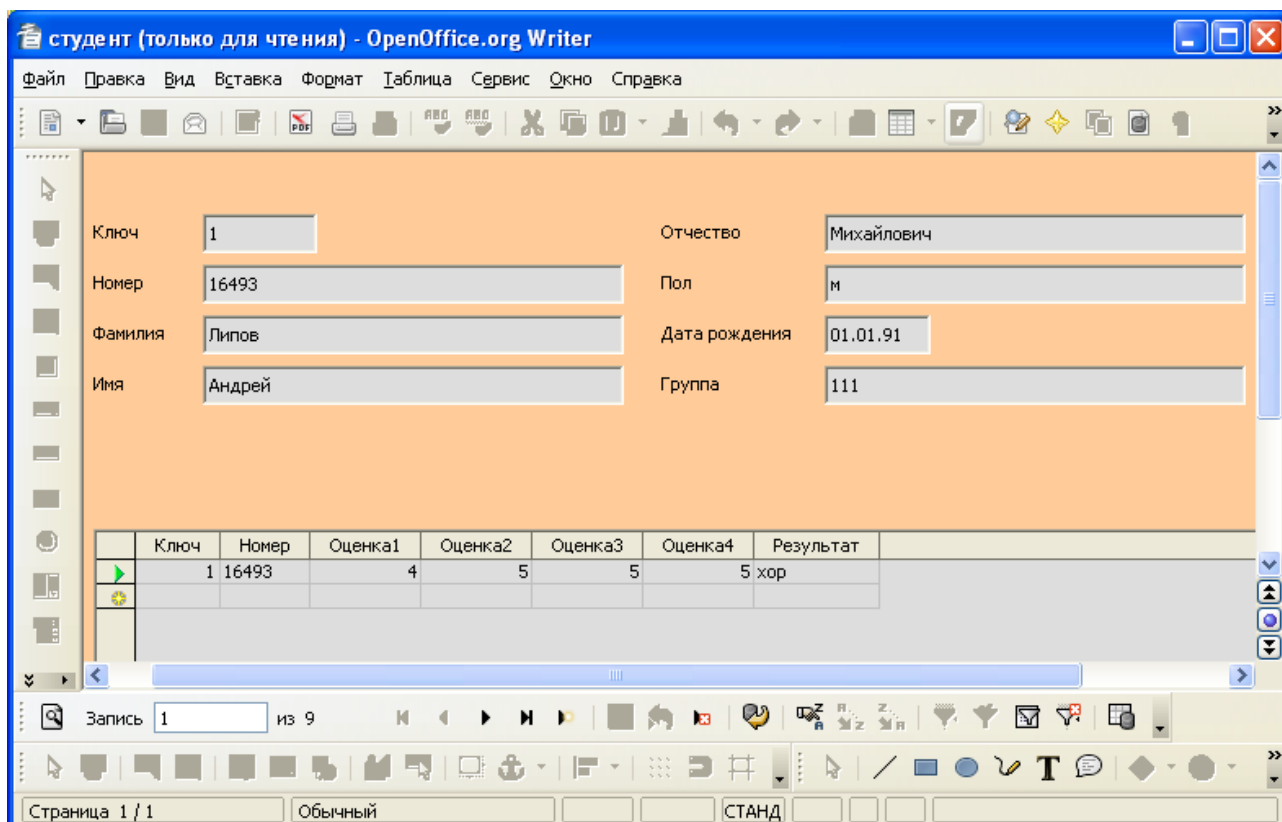


Рис. 29

После создания двух форм (рис. 20, 29) в основном окне базы данных появятся две формы: **студент** и **студент_основная** (рис. 30).

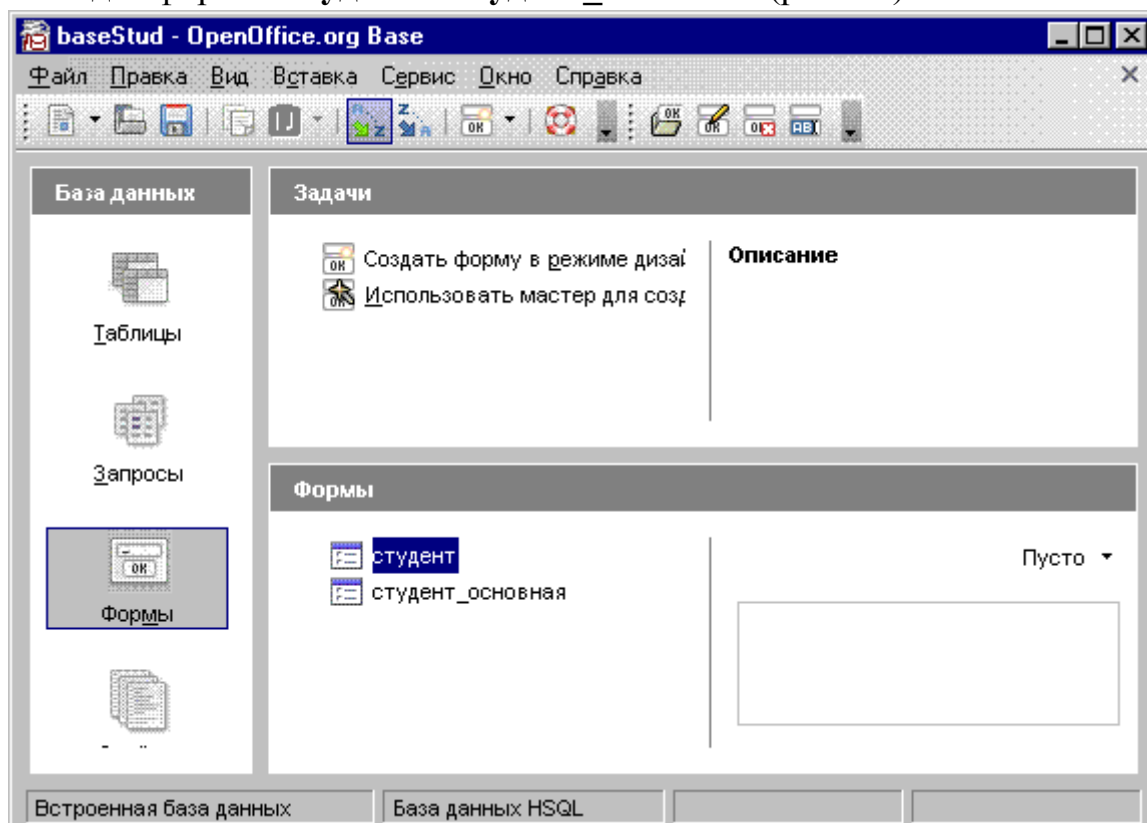


Рис. 30

2.13 Создание отчетов

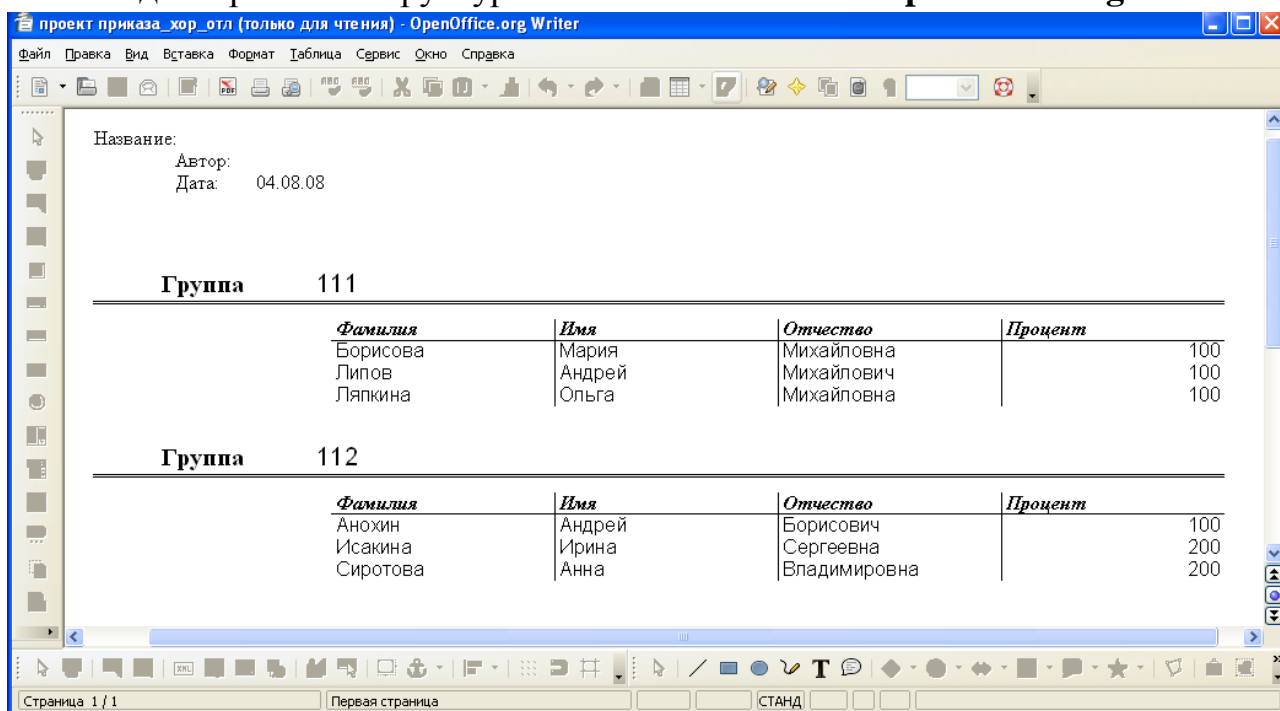
По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены для вывода на печатающее устройство. Отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов (верхний и нижний колонтитулы, номера страниц, служебная информация о времени создания отчета и т.д.).

Данные станут более информативными, если их сгруппировать по определенным категориям. Группа представляет собой совокупность родственных записей. Когда, например, просматриваем данные об успеваемости студентов, то лучше получить сведения, сгруппированные по номерам групп (рис. 31).

Кроме того, что отчет может представлять сгруппированные определенным образом данные, он еще может содержать вводную и суммарную информацию по каждой из групп.

Для создания отчета необходимо выбрать вкладку **Отчеты** и соответствующую ей задачу **Использовать мастер для создания отчета....** После выбора данной задачи появятся два окна: **OpenOffice.org Write** и **Мастер отчетов**. В окне **Мастер отчетов** необходимо пройти все шаги от 1 до 6. После последнего шага в окне **OpenOffice.org Write** появится результат создания отчета (рис. 31). Отчет был сделан на основании запроса обо всех студентах двух групп, получивших хорошие и отличные оценки.

Редактирование структуры отчета выполняют в **OpenOffice.org Write**.



The screenshot shows the OpenOffice.org Writer interface with a report titled "проект приказа_хор_отл (только для чтения) - OpenOffice.org Writer". The report contains two tables, one for Group 111 and one for Group 112. Each table has columns for Family Name, First Name, Patronymic, and Percentage.

Группа 111				
Фамилия	Имя	Отчество	Процент	
Борисова	Мария	Михайловна	100	
Липов	Андрей	Михайлович	100	
Ляпкина	Ольга	Михайловна	100	

Группа 112				
Фамилия	Имя	Отчество	Процент	
Анохин	Андрей	Борисович	100	
Исакина	Ирина	Сергеевна	200	
Сиротова	Анна	Владимировна	200	

Рис. 31

3. Задания и порядок выполнения

Варианты для заданий определяются преподавателем.

Задание № 1. Создание собственной базы данных и создание таблиц в ней

Задание № 2. Установление связей между таблицами

Самостоятельно определить базовую таблицу и тип связи между таблицами.

Задание № 3. Формирование запроса к многотабличным базам данных

Замечание: При формировании запросов к многотабличным базам данных нужно указывать, к какой таблице относится то или иное поле. При определении условия поиска обычно название таблицы отделяется от названия поля точкой. Формирование запросов с использованием полей из различных БД имеет смысл только в том случае, если между этими таблицами установлены связи.

Задание № 4. Создание форм и сложных форм

С помощью таблиц необходимо создать две формы (одна - основная, другая – подчиненная).

Задание № 5. Создание отчетов

Задание № 6. Создание запросов с использованием языка SQL (общее для всех)

На языке SQL необходимо написать команду выборки данных о студентах двух групп, получивших “4” и “5” (≥ 4). Необходимо использовать таблицы **студент** и **сессия** данного методического указания.

4. Контрольные вопросы

1. Какие этапы создания БД известны?
2. Как осуществить проектирование БД?
3. Что означает нормализованная БД?
4. Какие типы данных могут храниться в полях базы данных?
5. Каковы требования к связывающим таблицам?
6. Поясните, что означают связи: один к одному и один ко многим?
7. Для чего необходимо использовать **Связи**?
8. Какие известны виды запросов?
9. Какими возможностями обладают запросы?
10. Для чего необходим язык SQL?
11. Перечислите основные операторы языка SQL?
12. Из каких частей состоит сложная форма?
13. Что такое отчеты, и какими возможностями они обладают?

5. Список рекомендуемой литературы

1. Информатика. Базовый курс: Учебник для вузов / Симонович С.В. и др. – СПб.: Питер, 2011. – 640 с.
2. Костромин, В. Экспресс-Курс -- OpenOffice.org: открытый офис для Linux и Windows. / В.Костромин.- СПб.: БХВ, 2005. – 272 с.

