

ПРИМЕР РАЗРАБОТКИ БАЗЫ ДАННЫХ «ШКОЛА»

Построим пример базы данных учащихся школы средствами СУБД LibreOffice Base.

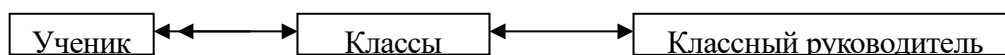
База данных учащихся должна являться частью общей базы данных информационной системы образовательного учреждения. Такая база данных содержит сведения об учащихся и их родителях, о сотрудниках, учебный план, электронные классные журналы, расписание уроков и факультативных занятий, распоряжения и отчеты, методические и образовательные ресурсы. Общая база данных является одним из компонентов информационной системы, который обеспечивает управление образовательным процессом, обучающимися, кадрами, ресурсами и предоставляет возможность интерактивной связи участникам этого процесса.

Рассмотрим упрощенный вариант школьной базы данных, причем обратим внимание только на ту ее часть, которая относится к учащимся.

Разработку БД начнем с проектирования. Так как LibreOffice Base использует реляционную модель данных, информацию расположим в трех таблицах — **Классы**, **Классный руководитель**, **Ученик**.

Так как мы рассматриваем небольшой пример, то естественно, что в таблицах отражена только небольшая часть реальной информации. Например, в таблице **Ученик** следовало бы внести не только фамилии учеников, но и имена, и другие личные данные.

Эти три таблицы должны образовать единую систему, поэтому они взаимосвязаны:



Здесь присутствуют два типа связи: "один к одному" и "один ко многим".

Таблицы **Классы** и **Классный руководитель** объединяет связь типа "один к одному", она на рисунке обозначена одинарной стрелкой. При этом типе связи две таблицы можно было бы объединить в одну таблицу, так как одной записи в одной таблице соответствует одна запись в другой таблице. Информация разделена на две таблицы, так как одна таблица содержала бы слишком много полей и с ней было бы неудобно работать. Эти таблицы находятся на одном уровне иерархии.

Таблицы **Классы** и **Ученик** объединяет связь типа "один ко многим", она обозначена двойной стрелкой. Например, в одном классе обучается несколько учеников, но каждый ученик может обучаться только в одном классе. Поэтому одной записи в таблице **Классы** соответствует несколько записей в таблице **Ученик**. Это связь между соседними уровнями иерархической структуры.

Приступим к созданию БД в приложении LibreOffice Base.

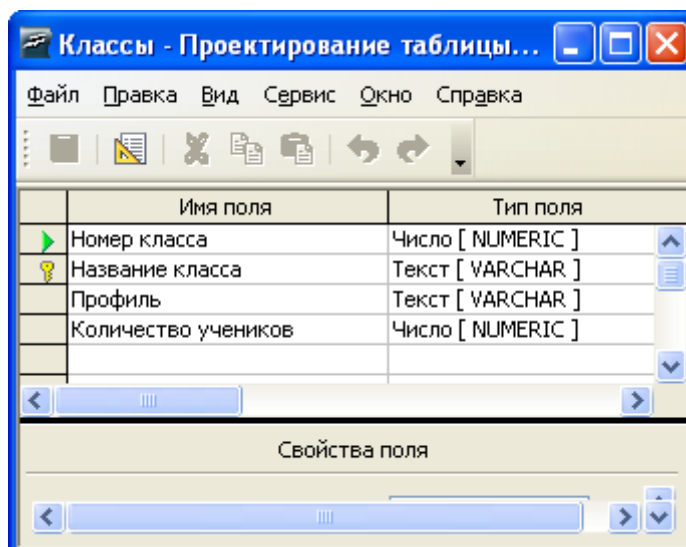
После запуска программы выбираем *Создать новую базу данных* и нажимаем кнопку *Далее*.

На втором шаге работы мастера подтверждаем *Открыть базу для редактирования*, нажав кнопку *Готово*. В окне *Сохранить как* выбираем свою папку на сетевом диске и сохраняем файл под именем **Школа**.

После создания файла новой БД следует построить структуру таблиц и заполнить их данными.

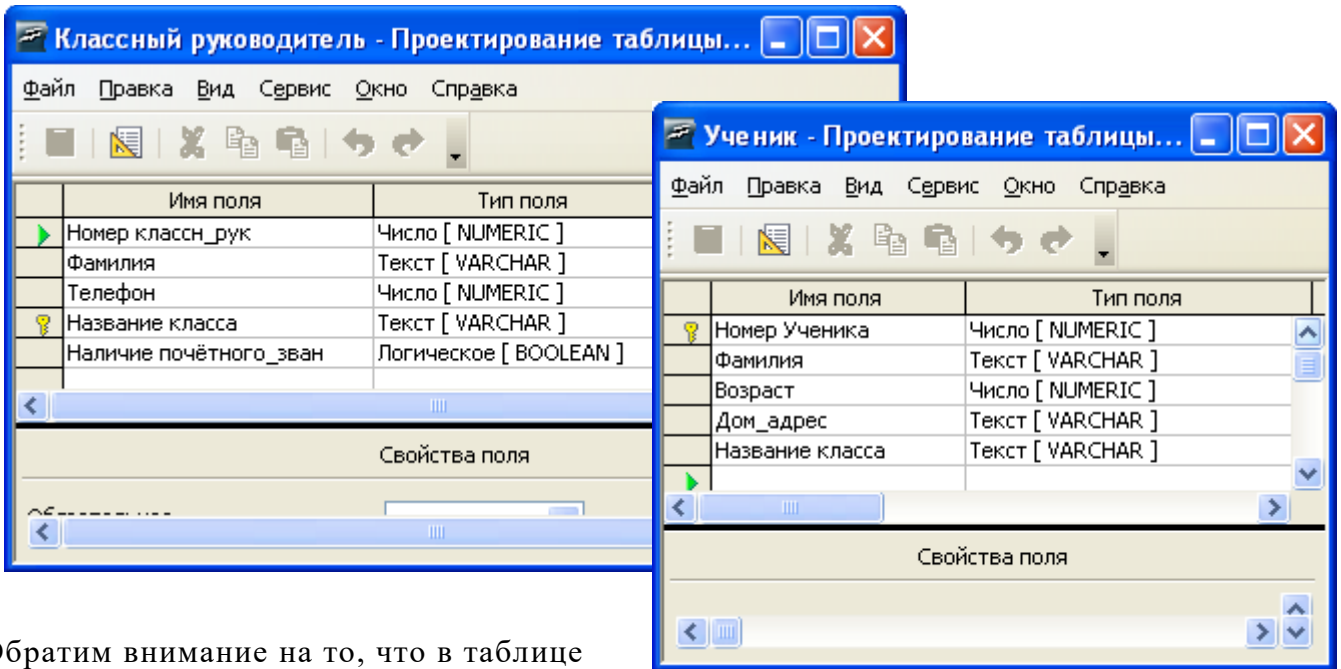
Для этого выделим вкладку *Таблицы*.

Структуру таблиц можно задать с помощью Конструктора (*Создать таблицу в режиме дизайна...*). На рисунке показано окно Конструктора при определении структуры таблицы **Классы**:



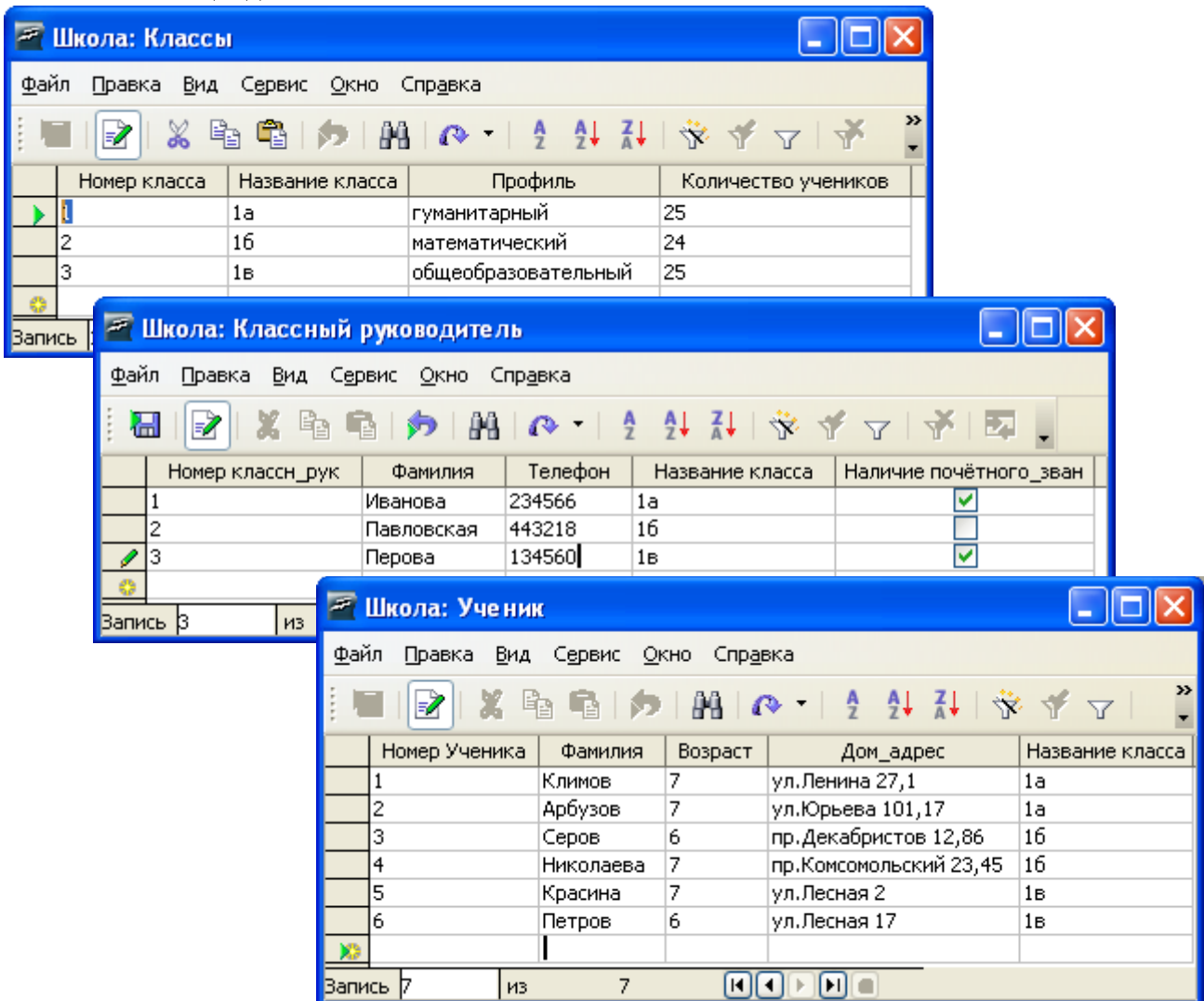
Изображение ключа указывает на ключевое поле. Установить первичный ключ можно, щелкнув правой клавишей мыши слева от имени поля и выбрав в меню соответствующий пункт.

Аналогично определим структуры таблиц *Классный руководитель* и *Ученик*:

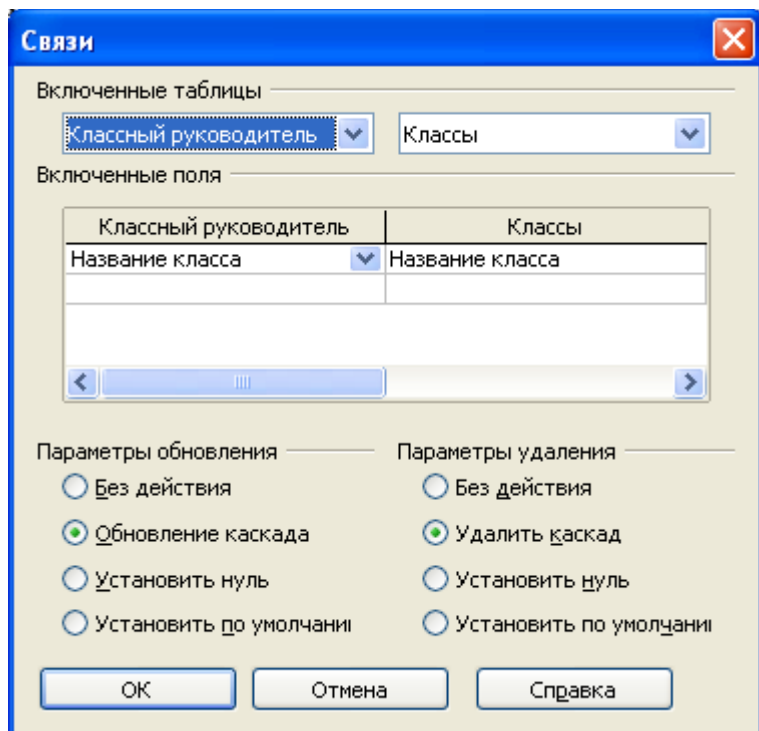


Обратим внимание на то, что в таблице *Классный руководитель* поле **Наличие почетного зван** имеет логический тип.

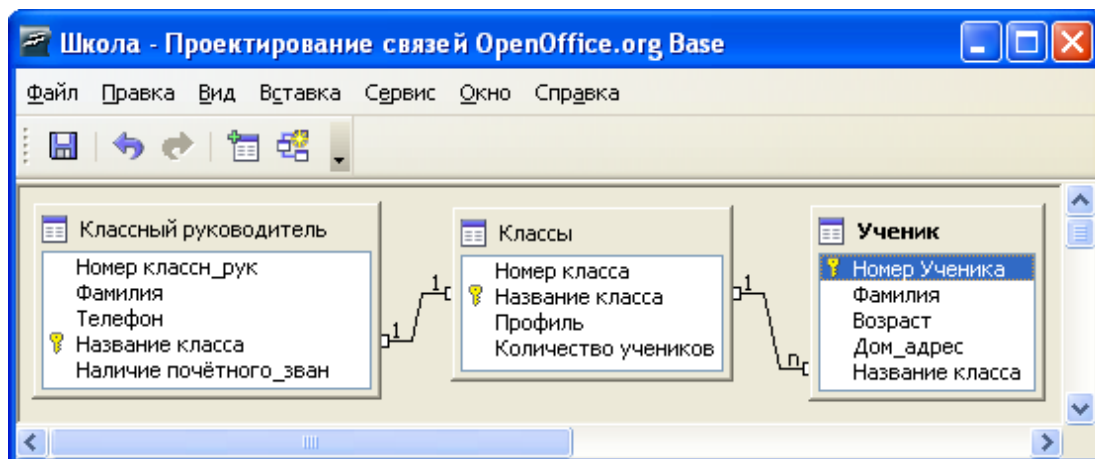
Заполним таблицы данными:



На следующем этапе определим связи между таблицами, выбрав пункты *Сервис, Связи..., Вставка, Создать связь...*



Программа обеспечивая целостность данных, препятствует удалению и изменению связанных данных. Изменение ключевого поля в первичной таблице приведёт к автоматическому изменению ключей в связанных записях. Удаление записей из первичной таблицы вызовет соответствующее удаление записей в связанной таблице. Для этого при создании связи следует установить **Каскадное обновление связанных данных** и **Каскадное удаление связанных данных**.



Здесь отражены два типа связи. Связь "один к одному" между таблицами **Классы** и **Классный руководитель** производится через общий главный ключ "Название класса".

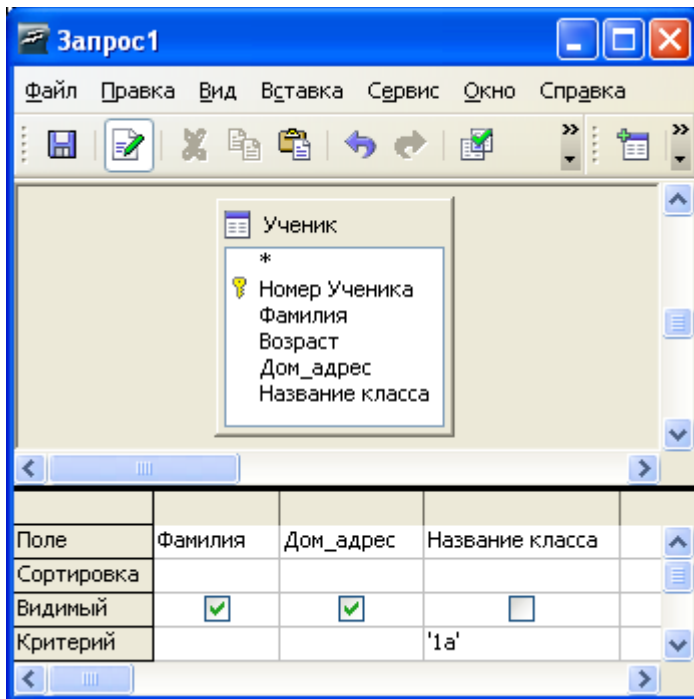
Связь "один ко многим" между таблицами **Классы** и **Ученик** создана с помощью главного ключа "Название класса" таблицы **Классы** и одноименного поля таблицы **Ученик**.

Для поиска информации в БД создают запросы.

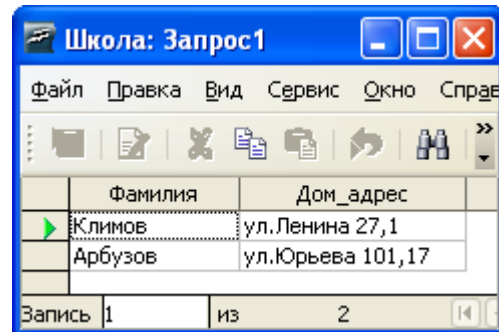
Запрос позволяет манипулировать данными БД: выбирать данные по некоторому условию, сортировать, обновлять, удалять и добавлять данные в БД. Простейшим является запрос на выборку. Результатом запроса на выборку будет таблица, которая содержит поля, удовлетворяющие условиям отбора.

В нашем примере *Запрос1* позволяет получить список учеников с домашними адресами, которые обучаются в классе 1а.

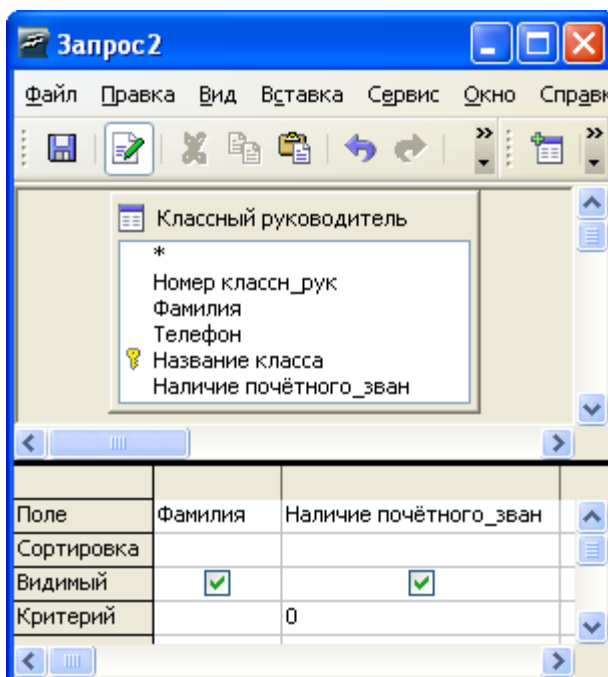
Создадим этот запрос в конструкторе (*Создать запрос в режиме дизайна...*), в качестве исходной таблицы укажем таблицу **Ученик**, в строке **Условие отбора (Критерий)** в поле **Название класса** зададим условие '1а':



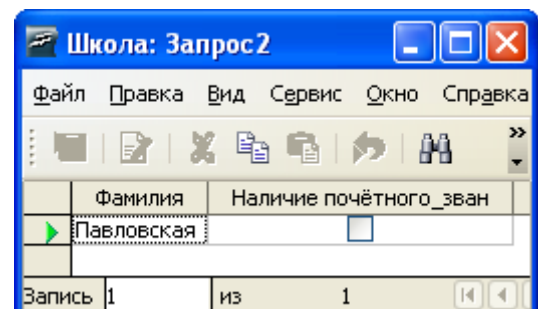
Далее получим результат выборки в таблице:



Второй запрос выводит фамилии классных руководителей, которые не имеют почётного звания. Он основан на таблице *Классный руководитель*. Поле **Наличие почётного зван** имеет логический тип, поэтому в **Условие отбора** внесена константа 0.

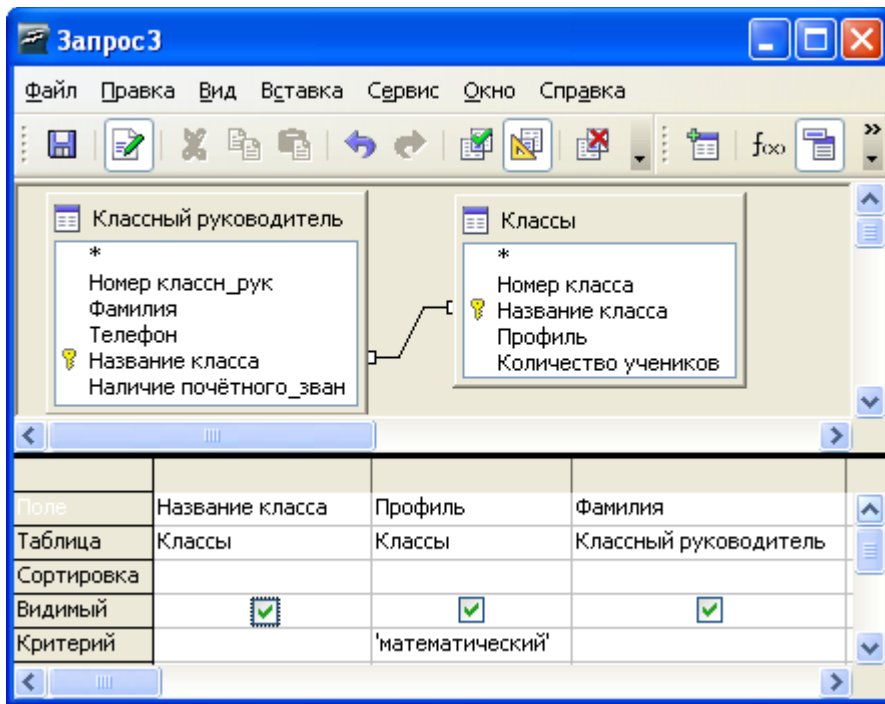


Результат выборки:



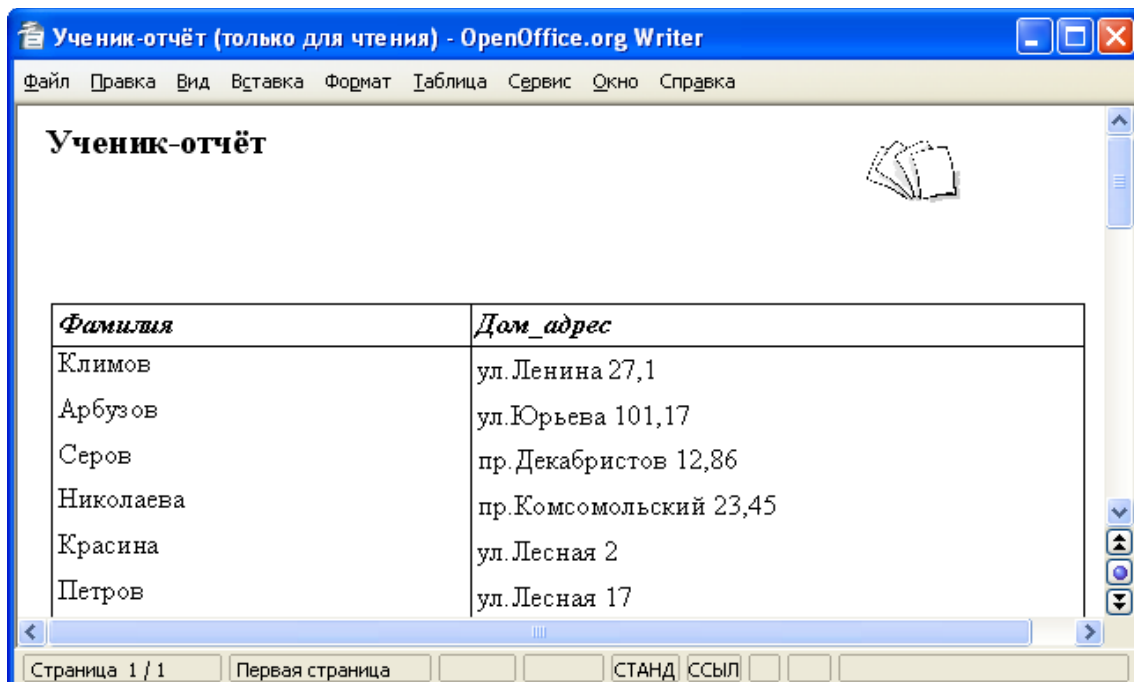
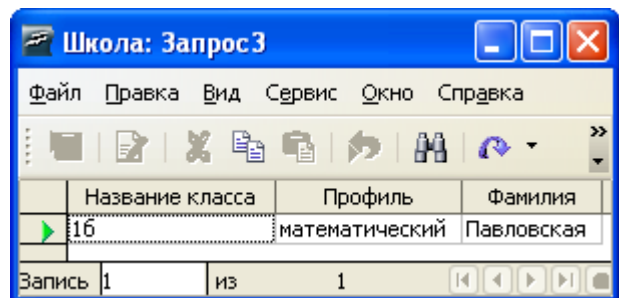
Запрос может быть многотабличным.

Запрос3 выводит поля **Название класса**, **Профиль** из таблицы *Классы*, и поле **Фамилия** из таблицы *Классный руководитель*, при этом поставлено условие, что класс имеет математический профиль.



Ответ можно просмотреть в режиме таблицы:

В заключение нашего примера подготовим отчет. Отчеты предназначены только для вывода информации на печать. Напечатаем фамилии учеников и их домашние адреса, основываясь на таблице **Ученик**. Выберем создание отчета с помощью мастера и, следуя его указаниям, получим следующий отчет:



Задание для самостоятельной работы:

Создайте формы для 3 таблиц и в этом режиме дополните БД *Школа* новыми записями!