

Министерство внутренних дел Российской Федерации
Тюменский институт повышения квалификации сотрудников МВД России

Д.Б. Панюшин, С.В. Щербич

ИНФОРМАЦИОННЫЕ СИСТЕМЫ:
НАЗНАЧЕНИЕ, СОЗДАНИЕ, ИСПОЛЬЗОВАНИЕ
В ДЕЯТЕЛЬНОСТИ ПОДРАЗДЕЛЕНИЙ
ОПЕРАТИВНО-РАЗЫСКНОЙ ИНФОРМАЦИИ

Учебное пособие

Тюмень
2016

УДК 343
ББК 67.99(2)91
П 16

Рекомендовано к изданию Редакционно-издательским советом
Тюменского института повышения квалификации
сотрудников МВД России

Рецензенты:

старший уполномоченный по ОВД 1 отдела УОРИ МВД России *М.В. Шавкунов*;
доцент кафедры информационного и технического обеспечения ОВД
Дальневосточного юридического института МВД России,
кандидат технических наук, доцент *П.Б. Скрипко*

Панюшин Д.Б., Щербич С.В.

П 16 Информационные системы: назначение, создание, использование в деятельности подразделений оперативно-разыскной информации: учебное пособие. Тюмень: Тюменский институт повышения квалификации сотрудников МВД России, 2016. 40 с.
ISBN 978-5-93160-222-6

Целью учебного пособия является систематизированное изложение информации по вопросам назначения и создания информационных систем, их использования в деятельности подразделений оперативно-разыскной информации. Рассматриваются основные понятия баз данных и этапы их проектирования. Дается характеристика моделей данных, приводятся их достоинства и недостатки. Описываются основы физического проектирования баз данных в ИСУБД CronosPro и СУБД Oracle с помощью пакета программ информационного обеспечения администратора системы «Портрет».

Учебное пособие составлено в соответствии с дополнительными профессиональными программами повышения квалификации сотрудников подразделений оперативно-разыскной информации системы МВД России.

Предназначено для сотрудников подразделений оперативно-разыскной информации системы МВД России, преподавателей образовательных организаций МВД России.

УДК 343
ББК 67.99(2)91

ISBN 978-5-93160-222-6

© ФГКУ ДПО «ТИПК МВД России», 2016

Содержание

Предисловие	4
1. Основные компоненты информационных систем.....	5
2. Базы данных.....	6
2.1. Основные понятия	6
2.2. Этапы проектирования базы данных	6
2.3 Предметная область баз данных, используемых в деятельности подразделений оперативно-разыскной информации.....	7
2.4. Построение инфологической модели предметной области баз данных, используемых в деятельности подразделений оперативно-разыскной информации	7
2.5. Модели данных.....	9
3. Физическое проектирование базы данных, основанной на сетевой модели данных, в ИСУБД CronosPro	16
3.1. ИСУБД CronosPro: основные понятия.....	16
3.2. Свойства полей ИСУБД CronosPro.....	16
3.3. Организация связи между базами данных в ИСУБД CronosPro.....	18
3.4. Проектирование банка данных в ИСУБД CronosPro	19
4. Физическое проектирование базы данных, основанной на реляционной модели данных, для СУБД «Oracle» с помощью пакета программ информационного обеспечения администратора системы «Портрет».....	25
Список литературы	38

Предисловие

Эффективное решение задач, стоящих перед оперативными подразделениями органов внутренних дел, невозможно без удовлетворения потребности каждого сотрудника в нужный момент получить систематизированную информацию по интересующему его вопросу. Удовлетворение этой потребности достигается за счет создания информационных систем.

В соответствии с Федеральным законом от 7 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» информационная система представляет собой совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств. Таким образом, основу информационных систем составляют базы данных, системы управления базами данных (далее – СУБД) и прикладные программные продукты, позволяющие работать с СУБД простым пользователям.

В связи с этим каждый сотрудник специализированных подразделений, выполняющих информационно-аналитическое обеспечение оперативно-розыскной деятельности, для результативного решения стоящих перед ним задач обязан иметь познания в области баз данных, СУБД и программных продуктов, используемых для работы с ними.

В учебном пособии рассмотрены основные понятия баз данных и этапы их проектирования. Дается характеристика моделей данных, приводятся их достоинства и недостатки. Описываются основы физического проектирования баз данных, основанных на сетевой модели данных, в ИСУБД CronosPro, а также основанных на реляционной модели данных – для СУБД «Oracle» с помощью пакета программ информационного обеспечения администратора системы «Портрет». Приведенные в учебном пособии примеры позволяют выявить особенности инфологической модели предметной области информационных систем, используемых подразделениями оперативно-розыскной информации.

Изложенный в учебном пособии материал предназначен для дополнительного профессионального обучения сотрудников подразделений оперативно-розыскной информации системы МВД России, позволяет им систематизировать уже имеющиеся у них знания по теории информационных систем и сформировать компетенции, необходимые для работы с программными продуктами, используемыми в служебной деятельности. Учебное пособие состоит из четырех разделов. По каждому разделу приводятся вопросы для самоконтроля.

Следует отметить, что хотя настоящее издание предназначено главным образом для сотрудников подразделений оперативно-розыскной информации системы МВД России, однако сфера его возможного практического использования не ограничивается подготовкой сотрудников только названной категории. Изучение представленного в пособии материала будет полезно также сотрудникам правоохранительных органов, планирующим использование информационных систем в своей служебной деятельности.

1. Основные компоненты информационных систем

Информационные системы включают два основных компонента: собственно базу данных и СУБД.

База данных предназначена для хранения данных, СУБД, в свою очередь, обеспечивает выполнение двух групп функций:

- предоставление доступа к базе данных прикладному программному обеспечению (или квалифицированным пользователям);
- управление хранением и обработкой данных в базе.

Таким образом, обращение к базе данных возможно только через СУБД.

Как правило, СУБД делятся на универсальные (предназначенные для различных сфер деятельности) и специализированные (созданные для конкретной сферы деятельности). Если интерфейс специализированных СУБД, как правило, интуитивно понятен заказчику, то работа напрямую с универсальной СУБД, требует от пользователя специальных познаний (например, знание языков описания и манипулирования данными). В связи с этим для облегчения работы пользователей создается прикладное программное обеспечение, позволяющее работать с СУБД, не обладая специальными познаниями (рис. 1.1).

Для решения разных задач, в информационной системе может использоваться несколько приложений. Так, например, информационная система ЦОРИ ГУ МВД России по Московской области помимо базы данных СУБД «Oracle» содержит несколько приложений, входящих в систему «Портрет-Поиск»: программа информационного обеспечения клиентского места системы «Портрет» и пакет программ информационного обеспечения администратора системы «Портрет»: «Редактор структуры», «Администратор баз», «Диспетчер пользователей», «Диспетчер репозитория» и др.

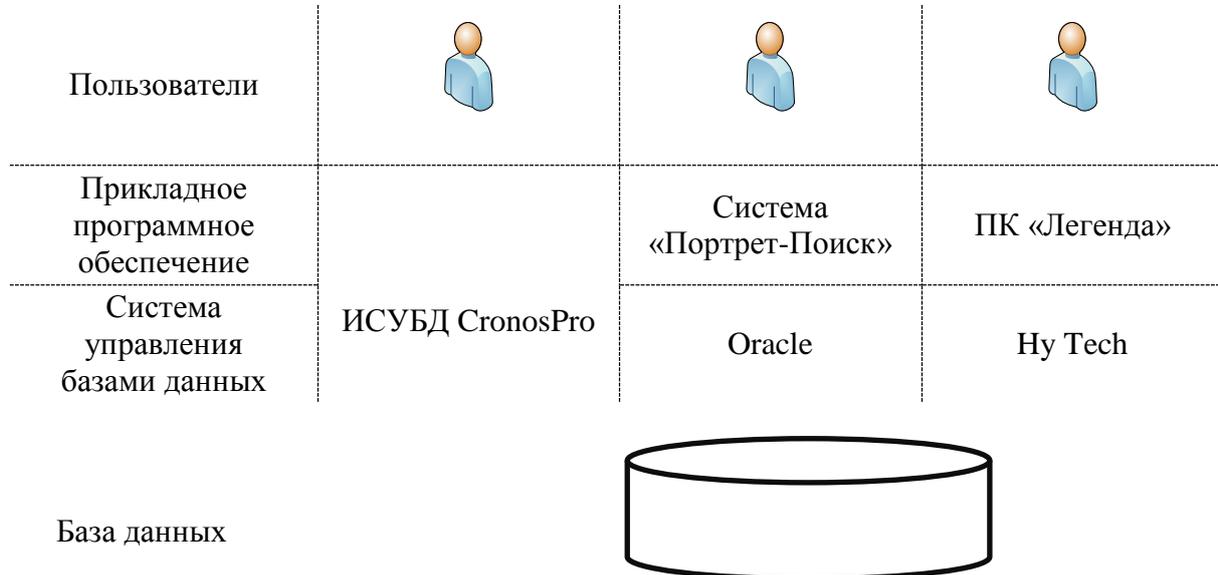


Рис.1.1. Компоненты информационной системы оперативно-разыскной информации.

Вопросы для самоконтроля

1. Дайте понятие информационной системы.
2. Какие компоненты входят в состав информационной системы?
3. Для чего предназначены СУБД?
4. Сформулируйте назначение прикладного программного обеспечения.

2. Базы данных

2.1. Основные понятия

Информация – сведения (сообщения, данные) независимо от формы их представления¹.

Данные – формализованная информация, готовая для последующей обработки, хранения и передачи средствами автоматизации профессиональной деятельности².

База данных – организованная совокупность данных, которые отображают состояние объектов какой-либо предметной области, отношения между этими объектами³.

Ограничение целостности – условие, которому должны соответствовать хранимые в базе данные.

Физическая независимость данных – неизменность логической структуры базы данных (сущностей, атрибутов, связей, ограничений целостности) при модификации способа ее хранения на физических носителях (изменение файловых систем или структур хранения, устройств хранения, индексов, методов шифрования или хеширования).

Логическая независимость данных – неизменность прикладных программ, используемых при работе с базой данных, при модификации ее логической структуры.

Нормализация отношений – процесс преобразования отношений базы данных к виду, отвечающему нормальным формам.

Нормальная форма – свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам при выборке или изменении данных.

Агрегация – процесс объединения элементов в одну систему.

2.2. Этапы проектирования базы данных

Создание любой базы данных должно начинаться с ее проектирования, которое включает в себя:

1. Изучение предметной области и информационных потребностей пользователей;
2. Создание инфологической модели предметной области;
3. Выбор модели данных и СУБД, с учетом созданной инфологической модели;
4. Описание инфологической модели на основе конкретной модели данных (логическое проектирование);
5. Физическое проектирование (создание модели базы данных для конкретной СУБД)⁴.

¹ Об информации, информационных технологиях и о защите информации: федер. закон от 7 июля 2006 г. № 149-ФЗ // URL: <http://pravo.gov.ru> (дата обращения: 10.08.2016).

² Информационные системы и технологии в профессиональной деятельности: учеб. пособие. Хабаровск: Дальневосточный юрид. ин-т МВД России, 2012. С. 32.

³ Там же.

⁴ В третьем разделе будет рассмотрено физическое проектирование базы данных на основе сетевой модели данных с использованием ИСУБД CronosPro и базы данных, основанной на реляционной модели данных, для СУБД Oracle с помощью пакета программ информационного обеспечения администратора системы «Портрет».

2.3. Предметная область баз данных, используемых в деятельности подразделений оперативно-разыскной информации

Под **предметной областью** следует понимать информацию об объектах, процессах и явлениях окружающего мира, которая с точки зрения потенциальных пользователей должна храниться и обрабатываться в информационной системе⁵. Исходя из того, что пользователями баз данных, используемых подразделениями оперативно-разыскной информации, являются оперативные подразделения органов внутренних дел, их предметная область является обширной и может включать в себя любую информацию, которая может понадобиться сотруднику оперативного подразделения при решении стоящих перед ним задач.

2.4. Построение инфологической модели предметной области баз данных, используемых в деятельности подразделений оперативно-разыскной информации

Объекты, процессы и явления, входящие в предметную область и представляющие интерес для пользователя, в теории баз данных называют *сущностями* (Entity) (ЛИЦО, УГОЛОВНОЕ ДЕЛО, ПРЕСТУПЛЕНИЕ и др.)⁶. При этом необходимо различать такие понятия, как *тип сущности* и *экземпляр сущности*. Понятие тип сущности относится к набору однородных объектов, выступающих как целое (ЛИЦА, УГОЛОВНЫЕ ДЕЛА и т.д.). Экземпляр сущности к конкретному объекту (УГОЛОВНОЕ ДЕЛО № 123456 и т.д.).

Каждая сущность обладает определенным набором свойств (*атрибутов*), которые позволяют описать ее. Например, атрибутами для сущности УГОЛОВНОЕ ДЕЛО являются НОМЕР ДЕЛА, ДАТА ЗАВЕДЕНИЯ ДЕЛА, ОРГАН и т.д. Говоря об атрибутах, также следует выделять тип и экземпляр. Тип атрибута ДАТА ЗАВЕДЕНИЯ имеет много экземпляров или значений (домен), однако каждому экземпляру сущности УГОЛОВНОЕ ДЕЛО присваивается только одно значение атрибута ДАТА ЗАВЕДЕНИЯ.

Экземпляры сущностей одного типа обладают одинаковыми наборами атрибутов, но должны отличаться значениями хотя бы одного атрибута, для того чтобы быть идентифицируемыми. Минимальный набор атрибутов, по значению которых можно идентифицировать экземпляр сущности, называется *ключом* или *первичным ключом*. Первичный ключ может состоять из единственного атрибута, значения которого уникальны для каждого экземпляра сущности. Так, например, в базе данных не должно быть двух абонентских номеров сотовой связи с одинаковым сочетанием цифр, поэтому абонентский номер является первичным ключом. Такой первичный ключ называют простым ключом. Если несколько экземпляров сущностей одного типа не отличаются значениями атрибутов, то для их идентификации может быть добавлен еще один атрибут – порядковый или системный номер, который будет уникальным у каждого экземпляра сущности одного типа.

Если тип сущности не имеет атрибута, содержащего уникальное значение, то первичный ключ может быть составлен из нескольких атрибутов, совокупность значений которых гарантирует уникальность. Так, атрибуты ГОРОД, УЛИЦА, ДОМ, КВАРТИРА не могут быть по отдельности первичными ключами сущности АДРЕС, так как могут оказаться одинаковыми у двух и более адресов. Однако не должно быть двух адресов с одинаковым значением атрибутов ГОРОД, УЛИЦА, ДОМ,

⁵ Информационные системы... С. 32.

⁶ В ЕСД ИС ОРИ содержится 21 тип сущности.

КВАРТИРА. Поэтому у сущности АДРЕС первичным ключом будет указанный выше набор атрибутов. Такой первичный ключ называется составным ключом.

Помимо атрибутов сущностей в базе данных должны содержаться *связи* (Relationships) между различными сущностями, что позволяет осуществлять отыскание одних сущностей по значению других⁷. Связь характеризуется количеством участников связи. Так, связь может быть одинарной (это связь между экземплярами одного типа сущности, ее еще называют рефлексивной), бинарной (между двумя сущностями), тернарной (между тремя сущностями) и т.д. На практике при моделировании предметной области наиболее часто используются бинарные связи.

Другой важной характеристикой связи между экземплярами сущностей является степень (кардинальность) этой связи. Степень связи характеризует число экземпляров сущности, которое может участвовать с каждой стороны связи. Для бинарных связей по степени связи с каждой ее стороны различают связи типов «один-к-одному» (1:1), «один-ко-многим» (1:m), «многие-к-одному» (m:1) и «многие-ко-многим» (m:n). Рассмотрим эти варианты.

Связь 1:1 (один-к-одному). В этой связи может участвовать не больше одного экземпляра каждой из связываемых сущностей.

Пример. Начальник отдела полиции является начальником не более одного отдела полиции, и в каждом отделе полиции имеется не более одного начальника.

Связи 1:m (один-ко-многим) и m:1 (многие-к-одному). Не более одного экземпляра одной сущности может быть связано с любым количеством экземпляров другой сущности.

Пример связи 1:m. В каждом отделе полиции могут работать несколько сотрудников оперативных подразделений, но каждый сотрудник оперативного подразделения работает только в одном отделе полиции.

Пример связи m:1. Каждое уголовное дело может быть возбуждено не более чем одним сотрудником, но каждый сотрудник может возбудить более чем одно уголовное дело.

Связь m:n (многие-ко-многим). Каждый из экземпляров обеих сущностей может быть связан с любым числом экземпляров другой сущности.

Пример. Каждое преступление может быть совершено любым количеством лиц, и каждое лицо может совершить любое количество преступлений.

Помимо количества участников и степени, связи могут характеризоваться обязательностью. Так, выделяют *обязательные* связи, когда каждый экземпляр сущности одного типа должен быть обязательно связан с каким-либо экземпляром сущности другого типа. Например, как правило, экземпляр сущности АДРЕС, должен быть связан с экземпляром другой сущности (ЛИЦО, ПРЕСТУПЛЕНИЕ и т.д.). *Необязательность* связи означает, что каждый экземпляр сущности одного типа может быть связан с каким-либо экземпляром сущности другого типа, а может быть и не связан.

Как у сущностей и атрибутов, у связей можно выделить тип связи и экземпляр. Тип связи характеризуется ее именем, количеством участников, степенью и обязательностью. Например, связь ПРОЖИВАЕТ сущностей ЛИЦО и АДРЕС является бинарной, степенью m:n и необязательной. Примером экземпляра связи будет конкретная связь экземпляра сущности ЛИЦО, с экземпляром сущности АДРЕС.

При построении инфологической модели необходимо учитывать информационные потребности будущих пользователей базы данных, что позволит уточнить круг необходимых сущностей, их атрибутов и связей. Например, если у сущности ЛИЦО отсутствует атрибут ОСОБЫЕ ПРИМЕТЫ, то пользователь в дальнейшем не сможет запросить информацию о лицах, имеющих шрам на лице.

⁷ В ЕСД ИС ОРИ содержится 12 типов связей.

Чаще всего инфологическая модель представляется в виде диаграммы сущностей-связей (entity-relationship) или ER-диаграммы. Несмотря на то, что при построении ER-диаграмм существуют некоторые классические обозначения (сущность представляется в виде четырехугольника, в который вписано ее название и атрибуты), разработчики программ, позволяющих рисовать ER-диаграммы, зачастую используют свою собственную систему обозначений⁸. Однако разобраться в различных способах записи ER-диаграмм обычно не сложно.

2.5. Модели данных

Исходя из того, что каждая СУБД поддерживает ту или иную модель данных, прежде чем определить СУБД, средствами которой будет описана инфологическая модель, необходимо определиться с моделью данных. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними. Модель данных определяет совокупность структур данных, операций их обработки и общих ограничений целостности.

По способу установления связей между данными различают *иерархическую, сетевую, реляционную, постреляционную, многомерную, объектно-ориентированную и объектно-реляционную модели данных*.

Иерархическая модель – представление базы данных в виде древовидной структуры, состоящей из объектов (данных) различных уровней. В иерархической структуре подчиненный элемент данных (потомок) всегда связан только с одним исходным (предок) (рис. 2.1).

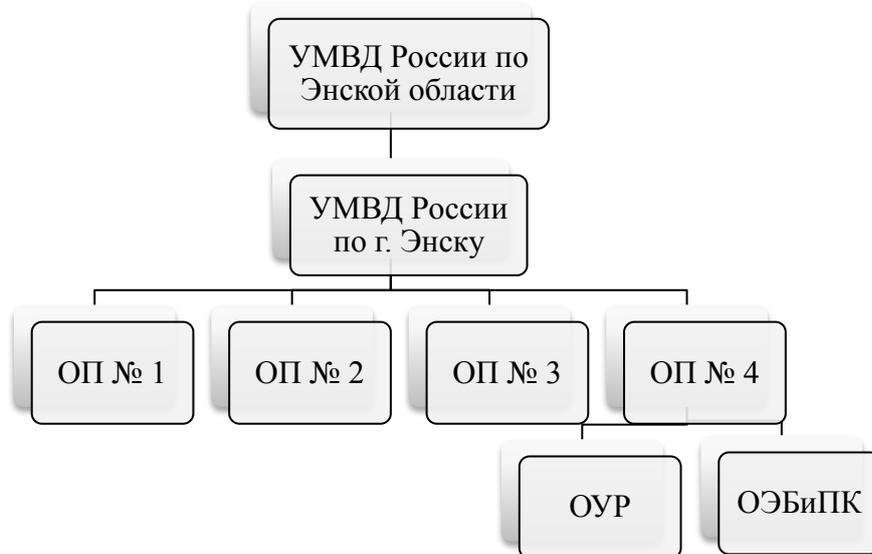


Рис. 2.1. Иерархическая модель данных

Достоинства иерархической модели: простота представления предметной области, наглядность, удобство анализа структур и простота их описания.

Недостатки иерархической модели: сложность добавления новых и удаления существующих сущностей, атрибутов и их связей; невозможность отображения от-

⁸ Например, в MS Visio связи обозначаются стрелками между сущностями и названия связей не пишутся. Название сущности выделяется цветом, поля, входящие в первичный ключ отделяется чертой от остальных атрибутов. Кроме того, слева от атрибута указывается, входит ли данный атрибут в первичный ключ, а также является ли атрибут внешним ключом.

ношений, отличающихся от иерархических; громоздкость описания и информационная избыточность.

Использование иерархической модели данных целесообразно, когда данные имеют естественную древовидную структуризацию, например, уголовный кодекс: разделы, главы, статьи, части. Для описания же предметной области, представляющей интерес для сотрудника оперативного подразделения, использование иерархической модели данных представляется не достаточно удобным.

Сетевая модель является расширением иерархической модели данных. Отличием сетевой модели данных является то, что в структуре данных у потомка может иметься любое число предков (рис. 2.2).

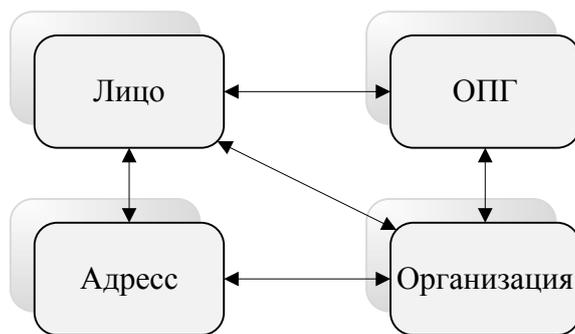


Рис. 2.2. Сетевая модель данных

По сравнению с иерархическими, сетевые модели обладают рядом существенных преимуществ: возможность отображения практически всего многообразия взаимоотношений объектов предметной области, непосредственный доступ к любой вершине сети (без указания других вершин), малая информационная избыточность. Вместе с тем в сетевой модели невозможно достичь полной независимости данных – с ростом объема информации сетевая структура становится весьма сложной для описания и анализа.

Реляционная модель. Термин «реляционный» означает, что такая модель данных основана на математическом понятии отношение (relation). В качестве неформального синонима термину «отношение» часто встречается слово «таблица». При использовании реляционной модели база данных представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об определенном типе сущности (ДОСТАВЛЕННЫЕ, СУДИМЫЕ, УГОЛОВНОЕ ДЕЛО и т.д.)⁹. Каждая строка таблицы (запись или кортеж) содержит данные об экземпляре сущности (например, лице, автомобиле, похищенной вещи), а столбцы таблицы содержат различные характеристики этих типов сущностей – атрибуты (например, ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ДАТА РОЖДЕНИЯ, МЕСТО РОЖДЕНИЯ). Атрибут может быть обязательным и необязательным. Значение обязательного атрибута должно быть определено в момент внесения данных в БД. Если атрибут необязательный, то для таких случаев предусмотрено специальное значение – NULL, которое можно интерпретировать как «неизвестное значение» (рис. 2.3).

⁹ Информационная система ЦОРИ ГУМВД России по Московской области насчитывает более 140 таблиц, содержащих информацию о типах сущностей и типах их связей.

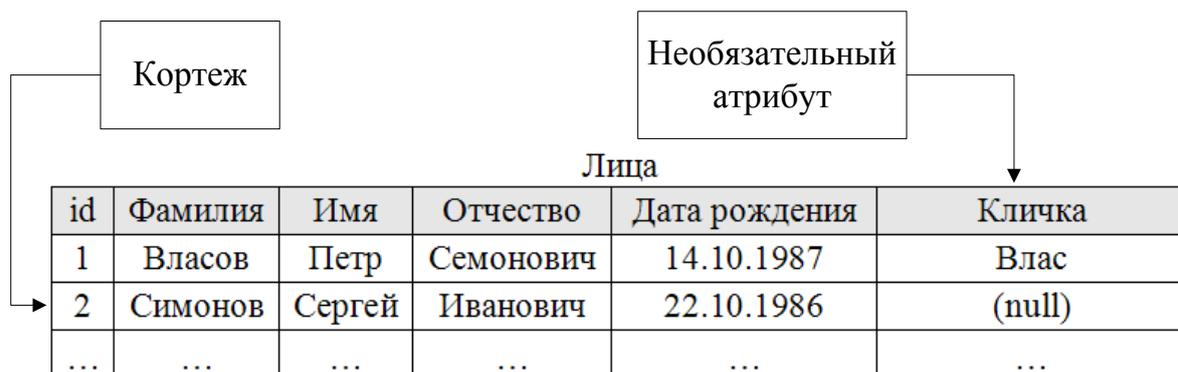


Рис. 2.3. Пример отношения «Лица» реляционной модели данных

Для связи отношений (таблиц) используются внешние ключи. Внешний (вторичный) ключ (Foreign key) – это атрибут отношения (подчиненной таблицы), который является копией первичного (Primary key) ключа другого отношения (главной таблицы) (рис. 2.4).

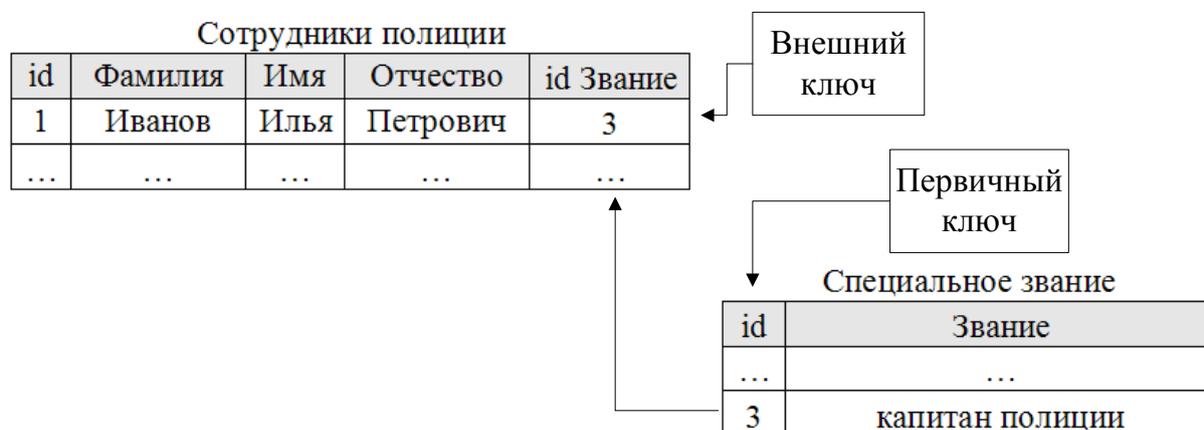


Рис. 2.4.1. Пример связи отношений один-ко-многим

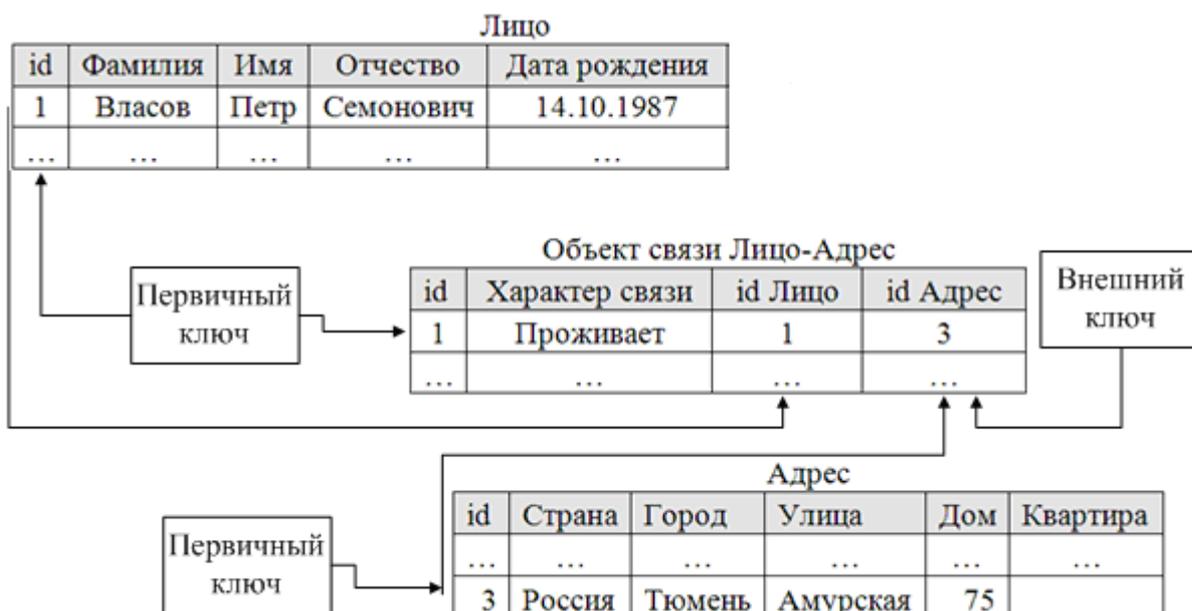


Рис. 2.4.2. Пример связи отношений «многие-ко-многим»

Таким образом, внешние ключи логически связывают экземпляры сущностей разных типов. Кроме того, внешний ключ может ссылаться на первичный ключ этой же таблицы (рис. 2.5). Это позволяет описывать одинарную связь – однотипных сущностей. Если связь необязательная, то значение внешнего ключа может быть неопределенным (Null).



Рис. 2.5. Пример одинарной связи в реляционной модели данных

Достоинствами реляционной модели данных являются:

- упрощенная схема представления данных – в виде таблицы;
- простота формирования базы данных;
- универсальность и удобство обработки данных, которая осуществляется с помощью декларативного языка запросов SQL (Structured Query Language).
- наличие методик, позволяющих описать предметную область.

В качестве недостатка реляционной модели можно отметить сложность описания иерархических и сетевых связей, необходимость проведения нормализации отношений, отсутствие механизмов навигации.

Постреляционная модель фактически является реляционной моделью, в которой допускаются некоторые ограничения. Так, например, она снимает ограничение неделимости данных, допуская многозначные поля, значения которых состоят из подзначений, и набор значений воспринимается как самостоятельная таблица, встроенная в главную таблицу. Так, при использовании постреляционной модели в отношении (таблице) может присутствовать множественное поле КОЛИЧЕСТВО ПРЕСТУПЛЕНИЙ, состоящее из полей, указывающих количество преступлений по месяцам года.

Таким образом, основным достоинством постреляционной модели является возможность представления совокупности связанных реляционных таблиц в виде одной постреляционной таблицы. А недостатком является сложность обеспечения целостности и непротиворечивости данных, хранимых в ней, что зачастую является определяющим.

Многомерная модель появилась в ответ на необходимость хранить информацию о состоянии сущности за различные периоды времени.

Например, значение атрибута КОЛИЧЕСТВО сущности КОЛИЧЕСТВО ДОСТАВЛЕННЫХ ЛИЦ зависит от таких параметров, как ПЕРИОД (месяц) и ОТДЕЛ ПОЛИЦИИ (табл. 2.1).

Реляционное и многомерное представление данных

Количество лиц, доставленных в отделы полиции г. Энска

Номер отдела	Месяц	Количество
ОП № 1	январь	18
ОП № 1	февраль	22
ОП № 1	март	25
ОП № 3	январь	15
ОП № 2	январь	13
ОП № 3	февраль	18
ОП № 2	февраль	15

а) Реляционное представление

Количество лиц, доставленных в отделы полиции г. Энска

Номер отдела	Количество		
	январь	февраль	март
ОП № 1	18	22	25
ОП № 2	13	15	
ОП № 3	15	18	

б) Многомерное представление (срез)

Наиболее подходящей моделью данных для данного случая является многомерная модель, используемая в технологии OLAP (OnLine Analytical Processing – оперативная аналитическая обработка).

Многомерные структуры представляются как гиперкубы данных. Каждая грань куба является размерностью. Основными понятиями, используемыми в многомерных моделях данных, являются измерение (dimension) и ячейка (cell).

Измерение – упорядоченный набор значений, принимаемых конкретным параметром, соответствующий одной из граней гиперкуба. Для нашего примера можно указать в качестве измерений: месяц – январь, февраль, март и т.д.; отдел полиции – ОП № 1, ОП № 2, ОП № 3 и количество доставленных.

Ячейка, или показатель – это поле, соответствующее атрибуту сущности, значение которого однозначно определяется фиксированным набором значений параметров (значениями измерений, например, январь, ОП № 1).

В многомерной модели данных определяется ряд дополнительных операций, среди которых можно выделить операции «формирование среза» и «агрегация».

При формировании среза пользователю по его запросу предоставляется некоторое подмножество гиперкуба, полученное в результате фиксации пользователем одного или нескольких значений параметров.

При агрегации допускается, например, объединение показателей за счет того, что параметры часто связаны в иерархии. Например, существует временная иерархия, связывающая секунды с минутами, минуты с часами, часы с сутками, сутки с неделями, недели с месяцами, месяцы с годами. Или возможна иерархия, связывающая отделы полиции с УМВД России по городу, УМВД России по городу с УМВД России по области. Часто одни и те же данные могут агрегироваться многими разными способами, т.е. одна и та же независимая переменная может принадлежать ко многим различным иерархиям.

Многомерная модель данных позволяет легко сравнивать данные при разных значениях параметров, строить графики зависимости значений конкретных атрибутов от значений определенных параметров (например, изменение атрибута по годам)

и т.п. Поэтому основное назначение технологии OLAP – обработка информации для проведения анализа и принятия решения.

К недостаткам многомерной модели можно отнести то, что из-за недостаточной научной обоснованности, зачастую возникают сложности при изменении схемы, например, при добавлении в базу новых типов данных или изменении ограничения. Таким образом, нельзя сказать, что многомерные модели являются универсальными, и нужно очень аккуратно подходить к их использованию.

Объектно-ориентированная модель. В конце 80-х годов XX века, когда впервые появились системы объектных баз данных, ряд ведущих специалистов в области объектного программирования заявили, что такие системы вскоре придут на смену реляционным системам. Однако сейчас можно сказать, что этого не произошло в силу того, что объектные системы имеют узкую область применения.

Объектно-ориентированная модель появилась благодаря объектно-ориентированным языкам программирования. Основная идея модели, состоит в избавлении пользователя от необходимости иметь дело с такими машинно-ориентированными конструкциями, как биты и байты (или даже записи и поля). Вместо этого пользователю предоставляется доступ к объектам и операциям над этими объектами, которые более соответствуют своим аналогам в реальном мире.

Например, используя традиционные термины реляционной модели, можно представить преступную группы, как кортеж ПРЕСТУПНАЯ ГРУППА с набором кортежей ЛИЦО, т.е. лиц, которые имеют «значения внешних ключей», «ссылающихся» на «значение первичного ключа» в кортеже ПРЕСТУПНАЯ ГРУППА. А в объектно-ориентированной модели будем иметь дело с объектом «преступная группа», который содержит соответствующее множество объектов «лицо». И вместо выполнения операции «вставки» нового «кортежа» в отношении ЛИЦО с соответствующим «значением внешнего ключа», «указывающего» на «значение первичного ключа» некоторого «кортежа» в отношении ПРЕСТУПНАЯ ГРУППА», пользователь должен иметь возможность непосредственно «включить» лицо (представленного объектом) в члены преступной группы (также представленной соответствующим объектом).

Объектно-ориентированная модель не имеет общей теории, при этом необходимо обозначить ее основные термины: неизменяемые объекты (значение в реляционной модели), изменяемые объекты (переменные), объектный класс (тип), метод (оператор), сообщение (вызов оператора).

Объектно-ориентированная модель данных состоит из объектов, каждый из которых должен принадлежать к определенному классу, то есть каждый объект – экземпляр класса. Объект-экземпляр класса принадлежит своему классу и имеет одного «родителя». При этом он сам может быть «родителем» для других объектов. Родовые отношения в БД образуют связанную иерархию объектов. Так, например, объект ПРЕСТУПНАЯ ГРУППА является родительским для объектов-экземпляров классов ЛИДЕР, и ЧЛЕН, которые в свою очередь являются родительскими для объектов-экземпляров классов ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО и т.д. (рис. 2.6). Структура и поведение объектов в объектной среде полностью определяются его классом, структура и поведение объектов одного класса одинаковы.

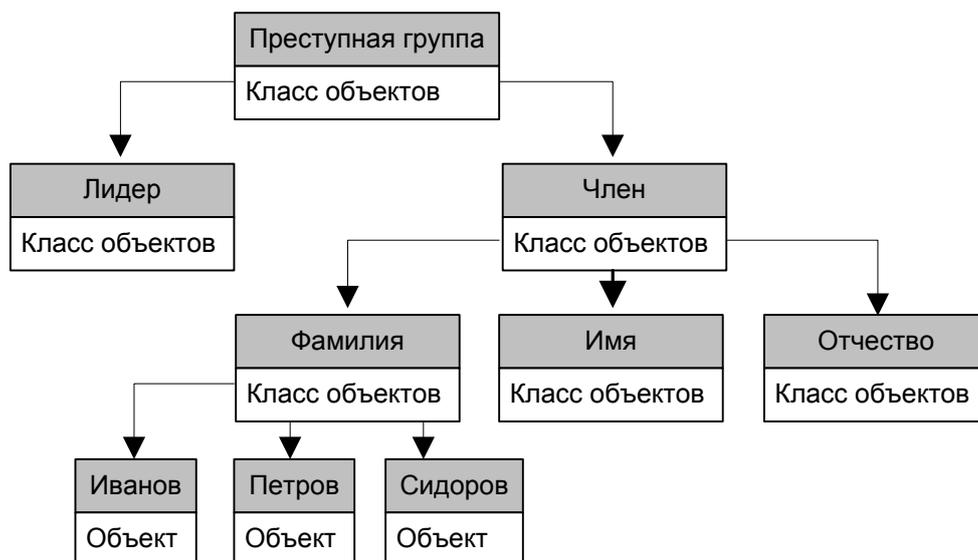


Рис. 2.5. Схема объектно-ориентированной модели

Среди недостатков объектно-ориентированной модели следует отметить отсутствие общепринятой теории, нацеленность на решение конкретных прикладных задач. Так, для манипулирования объектами заданы только заранее определенные методы, что делает затруднительным выполнение произвольных запросов (не предусмотренных этими методами).

Очень часто в специальной литературе выделяются также *объектно-реляционная модель*, которая объединяет в себе преимущества как реляционных, так и объектно-ориентированных моделей, однако, по мнению К. Дж. Дейта «объектно-реляционные системы – это подлинно реляционные системы, отличительной особенностью которых является то, что они позволяют пользователям определять их собственные типы. Реляционная модель не требует никаких дополнений для достижения объектно-реляционных функциональных возможностей»¹⁰.

Вопросы для самоконтроля

1. Дайте понятие следующим терминам: информация, данные, база данных, ограничение целостности, физическая независимость данных, логическая независимость данных, агрегация, нормализация отношений.
2. Назовите этапы проектирования базы данных.
3. Что такое предметная область?
4. Что такое инфологическая модель?
5. Назовите основные элементы инфологической модели.
6. Раскройте содержание понятия ER-диаграмма.
7. Дайте понятие типа и экземпляра сущности.
8. Какие характеристики присущи связи?
9. Сформулируйте понятие модели данных.
10. Охарактеризуйте иерархическую модель данных.
11. Охарактеризуйте сетевую модель данных.
12. Охарактеризуйте реляционную модель данных.
13. Охарактеризуйте многомерную модель данных.
14. Охарактеризуйте объектно-ориентированную модель данных.

¹⁰ Дейт К. Дж. Введение в системы баз данных / пер. с англ.; 8-е изд. М.: Вильямс, 2005. С. 1105.

3. Физическое проектирование базы данных, основанной на сетевой модели данных, в ИСУБД CronosPro

3.1. ИСУБД CronosPro: основные понятия

Банк данных – система специальным образом организованных данных о какой-либо предметной области, предназначенная для их накопления и использования. Данный термин является синонимом термину «база данных», используемому в первой главе.

База данных – относительно самостоятельная часть банка данных. Данный термин аналогичен термину отношение, используемому во второй главе (база данных может быть представлена ИСУБД CronosPro в виде таблицы). Кроме того, можно сказать о том, что название базы данных должно соответствовать названию сущности или связи, данные о которой содержатся в базе данных.

Словарная база данных (словарь) – список всех возможных значений некоторого поля, которое должно быть описано точно и однозначно (домен). Например, такое поле, как «Гражданство». При использовании словарной базы ввод значения данного поля сводится к простому выбору значения из списка, который выводится на экран. Такой список, т.е. словарная база, предварительно заполнен только теми значениями, которые следует использовать.

Запись базы данных – одна строка базы данных. Данный термин соответствует понятию кортежа в реляционных базах данных или экземпляру сущности.

Каждая запись базы данных в ИСУБД CronosPro имеет системный номер. Он присваивается записи автоматически и предназначен для того, чтобы система могла различать записи между собой даже в тех случаях, когда они содержат одинаковые значения¹¹. Пользователь видит (и иногда использует) системный номер (синоним – *ключ*), присвоенный каждой записи, однако не может его изменить.

Поле записи – элемент записи базы данных. Каждая запись базы данных содержит набор полей записи. Определить структуру базы данных, означает, в частности, решить то, какие поля будут входить в запись базы данных или какие атрибуты будут установлены для сущности.

Значение поля – это конкретные данные, хранимые в каждом поле (экземпляр атрибута). Многообразие значений поля является доменом.

3.2. Свойства полей ИСУБД CronosPro

1) *Тип данных поля*. Зависит от того, значения какого рода будут храниться в этом поле, а также от того, какие операции будут выполняться над этими значениями. Наиболее часто используется тип данных *Текстовый*, который применяется, например, для хранения данных о фамилии, имени и отчестве. Для хранения информации о дате рождения применяется тип данных *Дата*, для описания количественных характеристик – тип данных *Числовой*, и т.д. (табл. № 3.1).

¹¹ Одинаковые записи называются дублями и нужно стараться свести их количество к минимуму.

Тип поля в ИСУБД CronosPro версия 6.4

ПРОСТЫЕ ПОЛЯ		
Тип	Графическое обозначение	Допустимые значения поля
Текстовое		Любая последовательность алфавитно-числовых символов.
Числовое		Числовые значения (как целые, так и с плавающей запятой).
Дата		Значения даты в формате ДД.ММ.ГГГГ, где ДД – число (не более 31), ММ – месяц (не более 12), ГГГГ – год (не менее 1900)
Время		Значения времени в формате ЧЧ:ММ, где ЧЧ – часы (от 0 до 23), ММ – минуты (от 0 до 59)
Словарное		Значения задаются только из словаря (числовым кодом или его декодированным значением, которое автоматически преобразуется в числовой код)
Файл		Позволяет хранить файлы любых форматов, содержащие любые данные (изображения, видео, тексты и т.д.)
Внешний файл		По своему назначению аналогично полю типа ФАЙЛ. Отличие заключается в том, что в поле ВНЕШНИЙ ФАЙЛ хранится только ссылка на файл (т.е. адрес файла на диске или сетевом ресурсе). Сам файл в банк не помещается, что позволяет уменьшить размер банка данных.
СЛОЖНЫЕ ПОЛЯ (предназначены для установления связей между базами данных)		
Прямая ссылка		Содержит <i>прямую</i> ссылку на связанную базу. Значением такого поля являются имя (мнемокод) связанной базы и системный номер связанной записи.
Обратная ссылка		Содержит <i>обратную</i> ссылку из связанной базы. Значением такого поля являются имя (мнемокод) связанной базы и системный номер связанной записи.
Прямая-Обратная ссылка		Содержит <i>прямую-обратную</i> ссылку (ссылку между равноправными базами). Значением такого поля являются имя (мнемокод) связанной базы и системный номер связанной записи.
Связь по полю		Организует связь между базами посредством отношений с использованием связных (ключевых) полей.

Помимо перечисленных, в ИСУБД CronosPro предусмотрено два типа служебных полей (табл. № 3.2).

Служебные поля ИСУБД CronosPro

Тип	Графическое обозначение	Описание
Системный номер		Содержит числовое значение, однозначно отождествляющее запись в пределах банка. Поле «Системный номер» создается и заполняется системой автоматически и не может быть удалено или откорректировано пользователем
Уровень доступа		Значением данного поля является список групп доступа, которым доступна конкретная запись.

2) *Количество значений*, которые могут одновременно храниться в этом поле. Поле может быть *не множественным* (не кратным), если оно может одновременно содержать только одно значение, или *множественным* (кратным), если таких значений может быть несколько. Например, в базе данных есть поле «Особые приметы». Очевидно, что у лица может быть несколько особых примет (шрам, татуировки, ожог и т.д.) Таким образом, данное поле следует определить как множественное (кратное).

3) *Обязательность*. Поле может является *обязательным* или *необязательным* для заполнения. Например, если база данных содержит информацию о деле оперативного учета (ДОУ), то запись, в которой не указан вид ДОУ или его номер, является бессмысленной. Поэтому поля, имеющие важное значение, как правило, являются обязательными, а те поля, которые содержат дополнительную информацию, – необязательными.

3.3. Организация связи между базами данных в ИСУБД CronosPro

Запись одной базы данных может быть связана с записью другой базы данных. Для этого обе записи должны содержать сложные поля (ссылочные), ссылающиеся друг на друга. Очевидно, что для организации такой связи, структура базы данных должна предусматривать наличие сложного поля, поскольку для связи двух экземпляров разных сущностей, необходимо, чтобы типы сущностей, к которым они относятся, были связаны.

Например, в банке данных содержится две базы данных (ЛИЦО и АДРЕС), для создания связи между ними в базу данных ЛИЦО должно быть включено сложное поле (например, с именем ПРОЖИВАЕТ), содержащее ссылку на сложное поле с именем ЯВЛЯЕТСЯ АДРЕСОМ, базы данных АДРЕС.

Несмотря на то, что такой подход рекомендован¹² разработчиками ИСУБД CronosPro, он не позволяет дать точную характеристику связи между базами данных. Так, например, связь лица с адресом может происходить в силу того, что лицо там проживает, появлялся, ранее проживало, может находиться и т.д. Для отражения всех характеристик связи ЛИЦО-АДРЕС может использоваться дополнительная база данных, каждая запись которой содержит два сложных поля, связанные с базой данных АДРЕС и базой данных ЛИЦО (рис. 3.1).

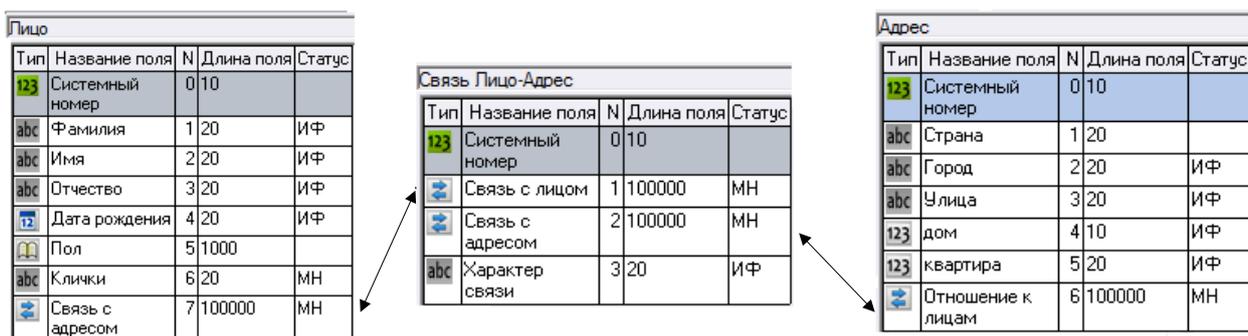


Рис. 3.1. Организация связи двух баз «Лицо» и «Адрес» через дополнительную базу «Связь Лицо-Адрес».

ИСУБД CronosPro позволяет связывать записи одной базы данных, например лицо, друг с другом. Однако, как и в предыдущем примере, для точной характеристики связи (родственники, друзья, соучастники и т.д.) целесообразно использовать дополнительную базу данных СВЯЗЬ ЛИЦО-ЛИЦО.

¹² Руководство пользователя CronosPro. URL: <http://cronos.ru/cronospro.html> (дата обращения: 10.08.2016).

3.4. Проектирование банка данных в ИСУБД CronosPro

Для ознакомления с проектированием банка данных в ИСУБД CronosPro создадим банк данных «Фанаты».

Данный банк данных должен содержать следующие базы данных: ЛИЦО, АДРЕС, СПОРТИВНЫЙ КЛУБ, ФАНАТСКАЯ ГРУППА, СВЯЗЬ ЛИЦО-ЛИЦО, СВЯЗЬ ЛИЦО-АДРЕС, СВЯЗЬ ЛИЦО-ФАНАТСКАЯ ГРУППА (рис. 3.2); словарные базы данных: ПОЛ, СТРАНЫ.

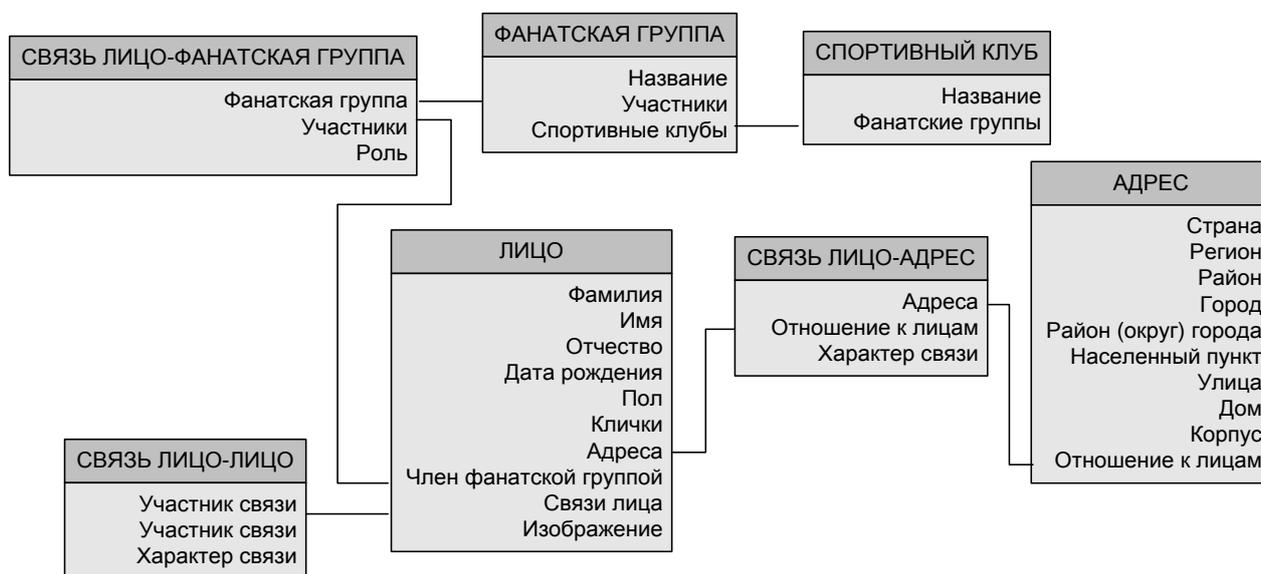


Рис. 3.2. Инфологическая модель банка данных «Фанаты»

Для создания нового банка данных необходимо перейти во вкладку «Банки», выбрать команду «Список банков» и в открывшемся окне выбрать команду «Новый» (рис. 3.3).

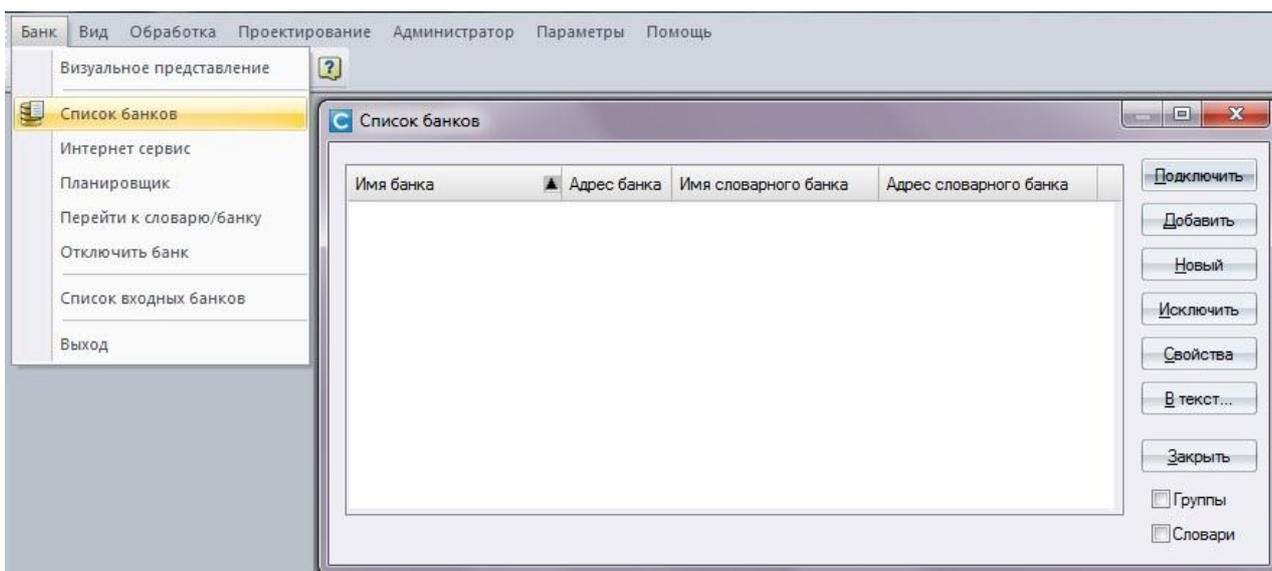


Рис. 3.3. Создание нового банка данных в ИСУБД CronosPro

В открывшемся окне «Создание нового банка» необходимо задать параметры (свойства) создаваемого банка данных: его имя, адрес хранения данных, тип банка и т.д. Поскольку мы определили, что наш банк данных будет содержать словари, то необходимо указать его имя в качестве имени словарного банка (рис. 3.4).

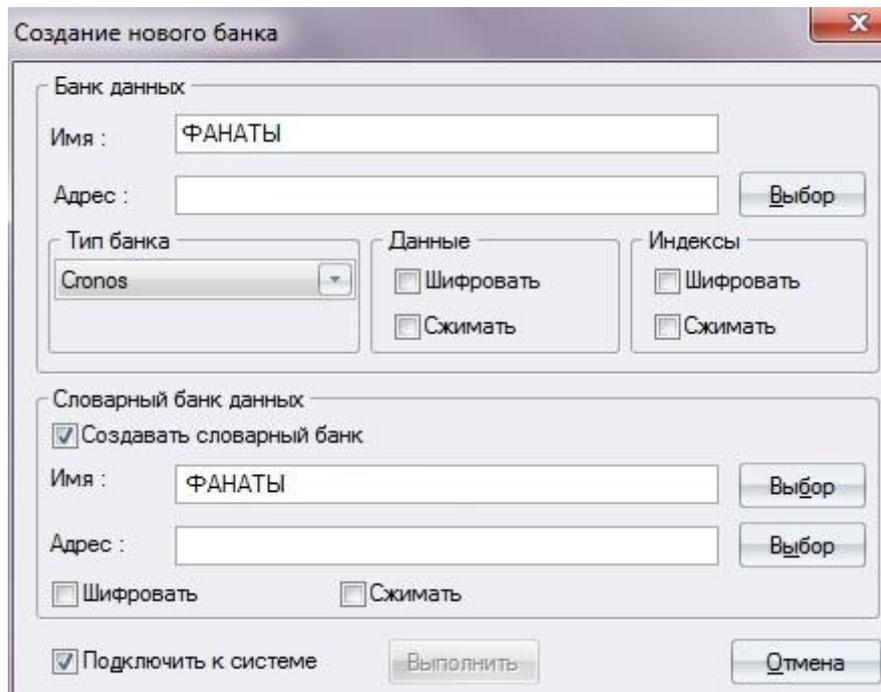


Рис. 3.4. Создание нового банка данных в ИСУБД CronosPro

Чтобы приступить к проектированию структуры вновь созданного банка данных, необходимо выполнить команду Главного меню **Проектирование** ⇒ **Структуры банка данных**. На экране появится окно диалога «Проектирование банка данных». Кнопка «Новая» служит для создания и описания свойств *новой* базы «Лицо» (рис. 3.5).

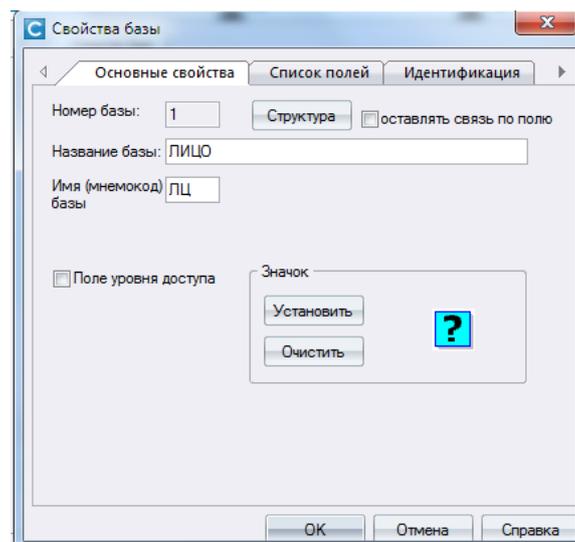


Рис. 3.5. Создание новой базы данных в ИСУБД CronosPro

Используя вкладку *Список полей* в окне свойств базы данных, описываем поля в создаваемой базе (рис. 3.6).

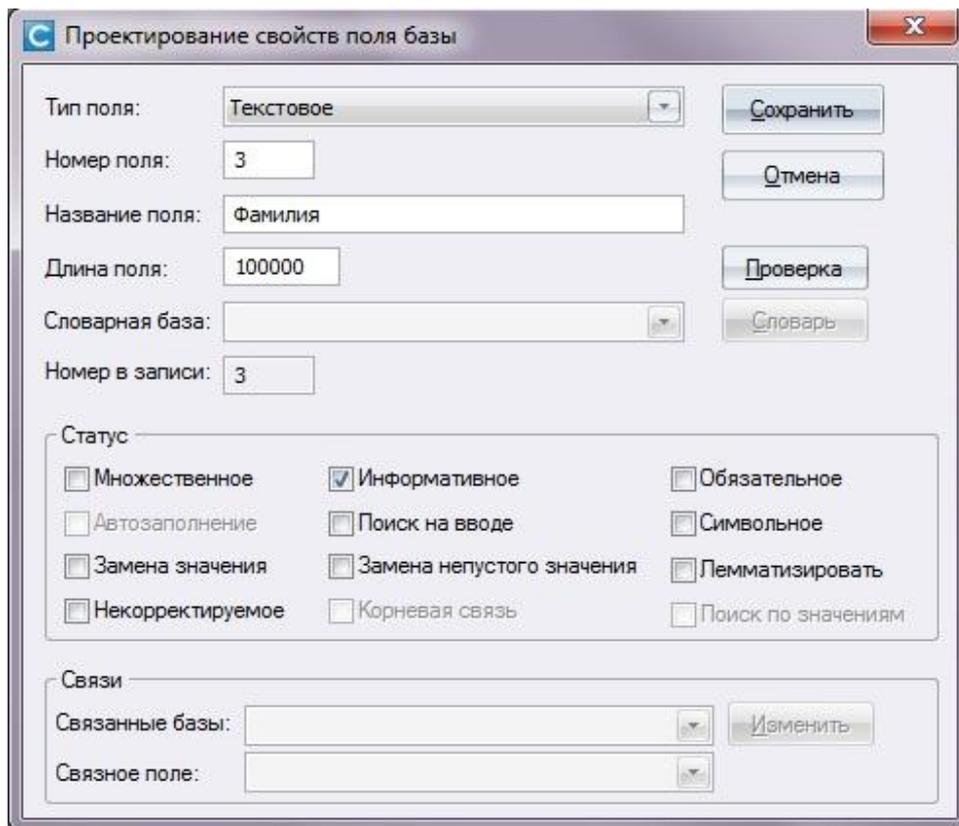


Рис. 3.6. Окно свойств поля базы данных в ИСУБД CronosPro

Очевидно, что поля ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, КЛИЧКИ будут полями типа *Текстовое*. Причем, поле КЛИЧКИ может быть множественным, о чем следует указать при проектировании свойств поля, поставив галочку в соответствующем окне. Длина поля (допустимое количество символов) указывается в соответствующей строке.

Поля ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО следует отметить как *Информативные*. В этом случае значение информативных полей записи базы данных ЛИЦО, при работе с банком данных будут показываться через пробел в полях ПРЯМАЯ-ОБРАТНАЯ ССЫЛКА, связанных с данной записью, записей других баз данных, например, СВЯЗЬ ЛИЦО-АДРЕС, СВЯЗЬ ЛИЦО-ФАНАТСКАЯ ГРУППА, СВЯЗЬ ЛИЦО-ЛИЦО.

Поле ДАТА РОЖДЕНИЯ в создаваемой базе данных будет иметь тип *Дата*.

Поля НАЦИОНАЛЬНОСТЬ и ПОЛ будут заполняться из словарной базы данных, для этого необходимо выбрать из предложенного списка имя соответствующего словаря. Этот список состоит из имен всех словарных баз, входящих в состав подключенного словарного банка (рис. 3.7).

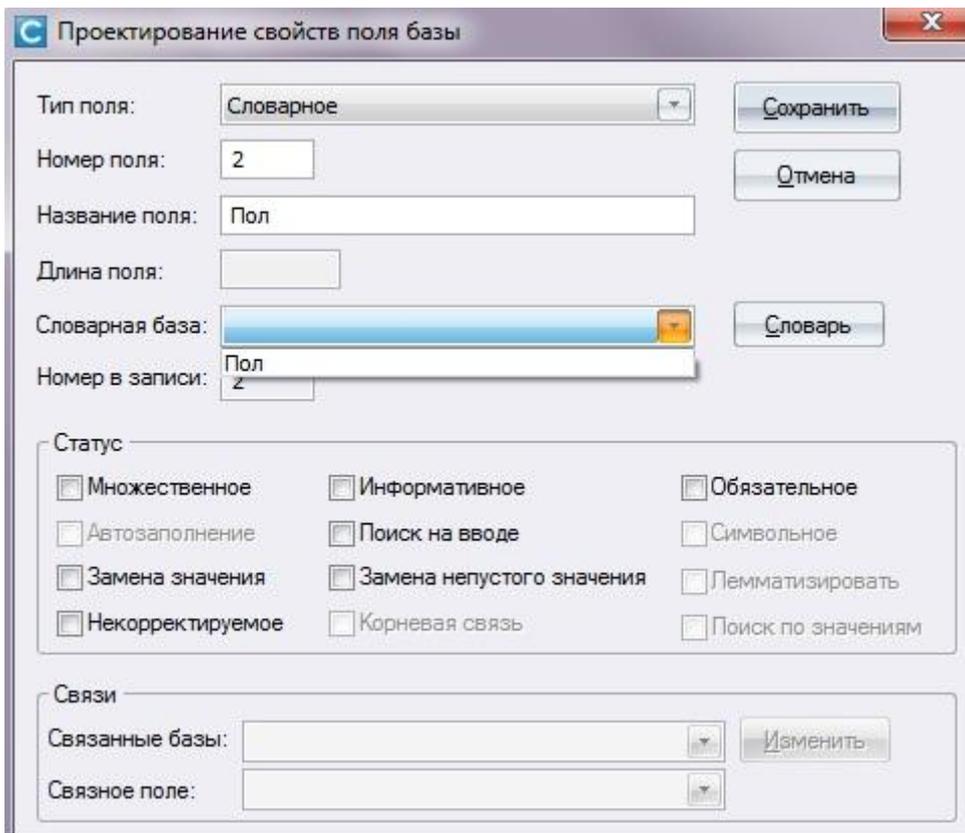


Рис. 3.7. Окно создания словарного поля «Пол»

Словарная база данных создается предварительно точно так, как обычная база данных (рис. 3.8).

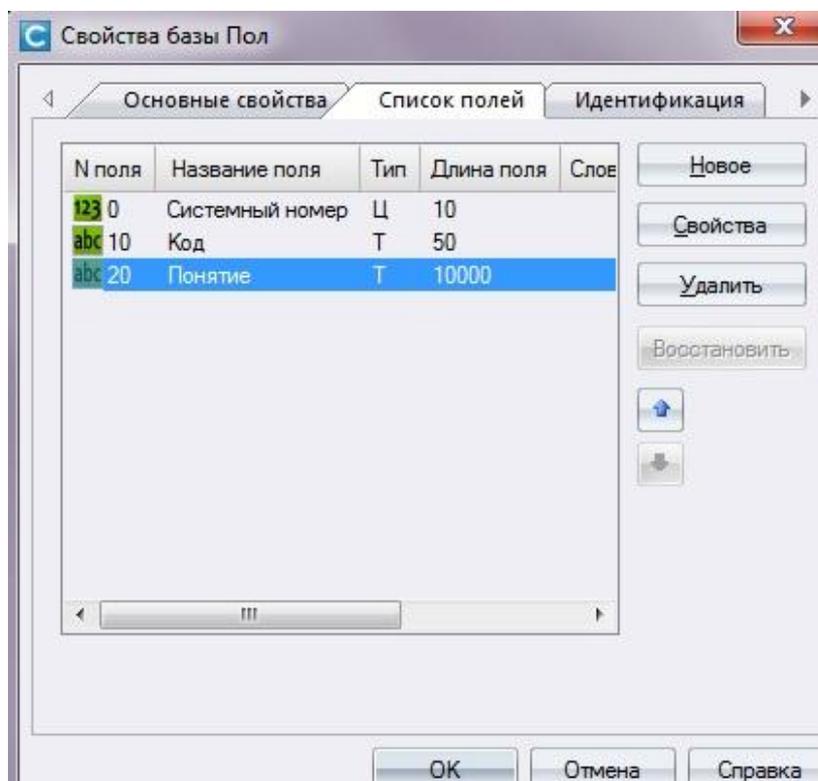


Рис. 3.8. Окно свойств словарной базы данных «Пол»

Поля ЧЛЕН ФАНАТСКОЙ ГРУППЫ, АДРЕС, СВЯЗИ ЛИЦА будут сложными полями типа *Прямая-Обратная ссылка*. Данные поля будут связаны с созданными в дальнейшем базами данных СВЯЗЬ ЛИЦО-АДРЕС, СВЯЗЬ ЛИЦО-ФАНАТСКАЯ ГРУППА, СВЯЗЬ ЛИЦО-ЛИЦО. После указания типа поля *Прямая-Обратная ссылка*, необходимо указать его имя и нажать на кнопку «Изменить» для выбора связанной базы и имеющегося в ней сложного поля, в нашем случае «Отношения к лицам» (рис. 3.9). После чего следует нажать кнопку «Установить».

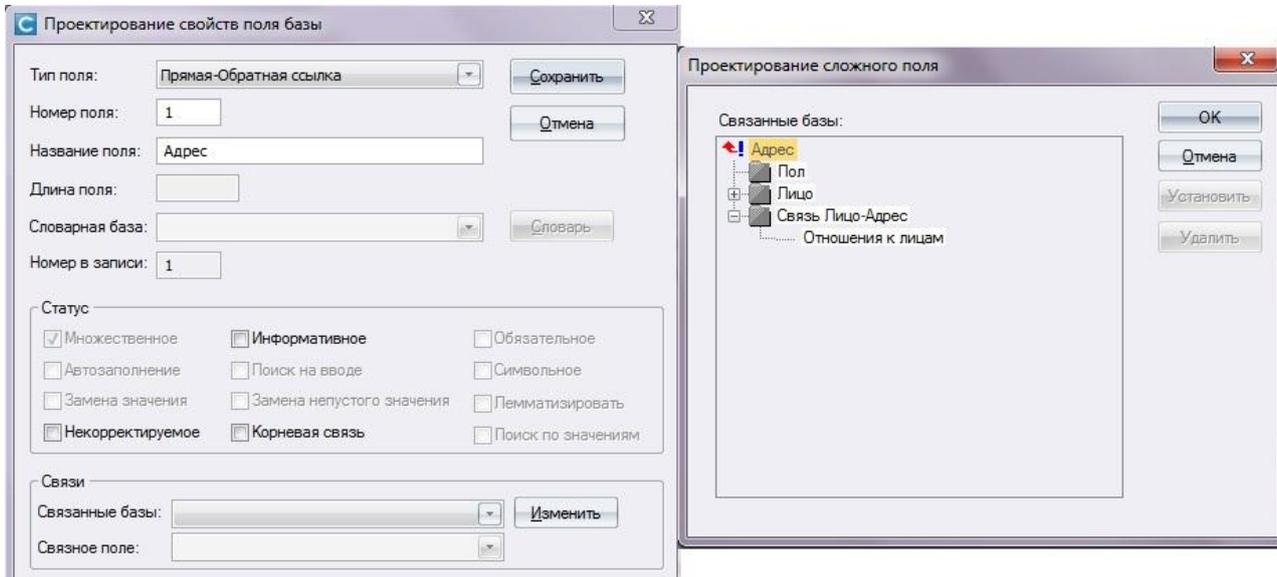


Рис. 3.9. Окна проектирования сложного поля «Адрес»

В результате выполнения вышеуказанных операций вновь созданная база будет иметь следующий вид (рис. 3.10).

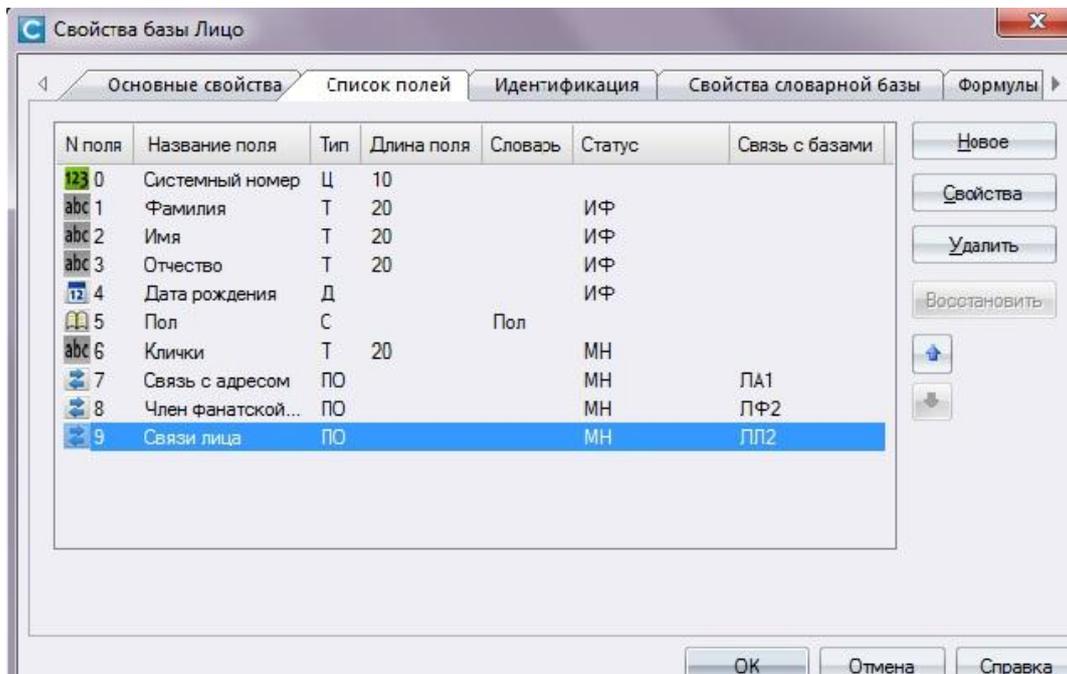


Рис. 3.10. Поля, содержащиеся в базе данных «Лицо»

Аналогичным образом проектируем другие базы данных, которые должны содержать банк данных ФАНАТЫ. После последовательного построения указанных баз данных банк данных ФАНАТЫ примет следующий вид (рис. 3.11).

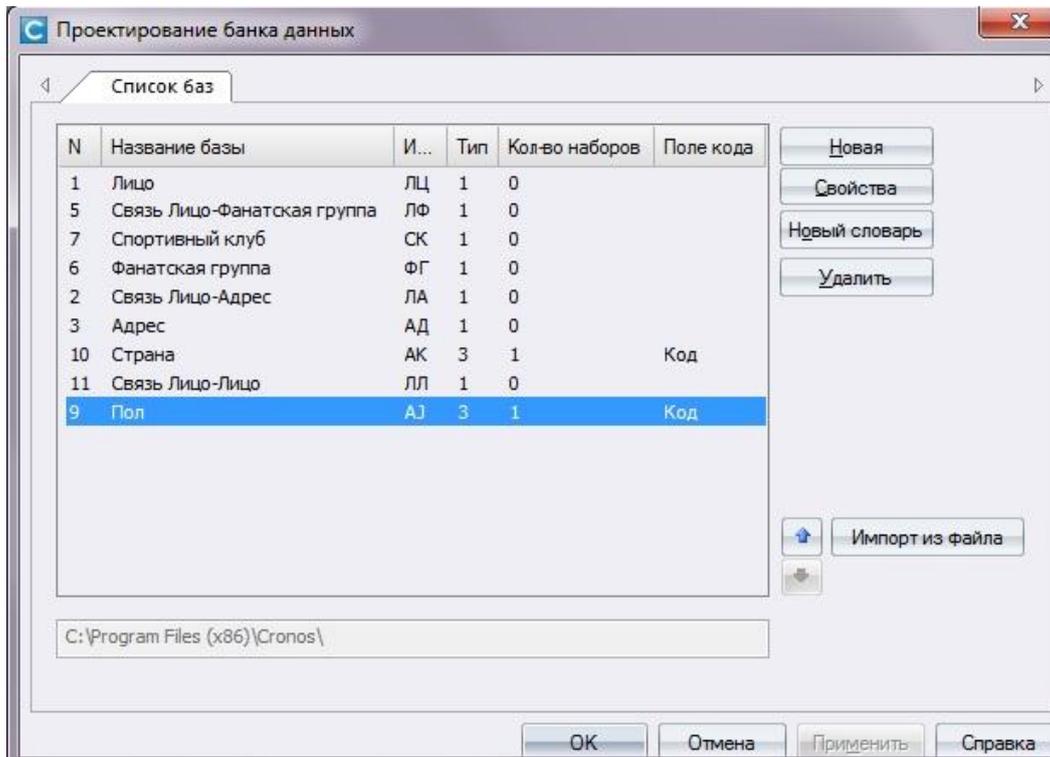


Рис. 3.11. Структура банка данных «Фанаты»

Вопросы для самоконтроля

1. Дайте определение следующим терминам: банк данных, база данных, словарь, запись базы данных, поле записи, значение поля. Соотнесите данные термины с терминами реляционной модели данных.
2. Назовите свойства полей ИСУБД CronosPro.
3. Как организована связь между двумя базами данных в ИСУБД CronosPro?
4. Назовите отличия словарной и обычной базы данных.

4. Физическое проектирование базы данных, основанной на реляционной модели данных, для СУБД «Oracle» с помощью пакета программ информационного обеспечения администратора системы «Портрет»

В данном разделе будет описан процесс создания учебной базы данных ФАНАТЫ, инфологическая модель которой приведена в третьем разделе. Для создания базы данных в «Портрете» необходимо запустить «Администратор баз».

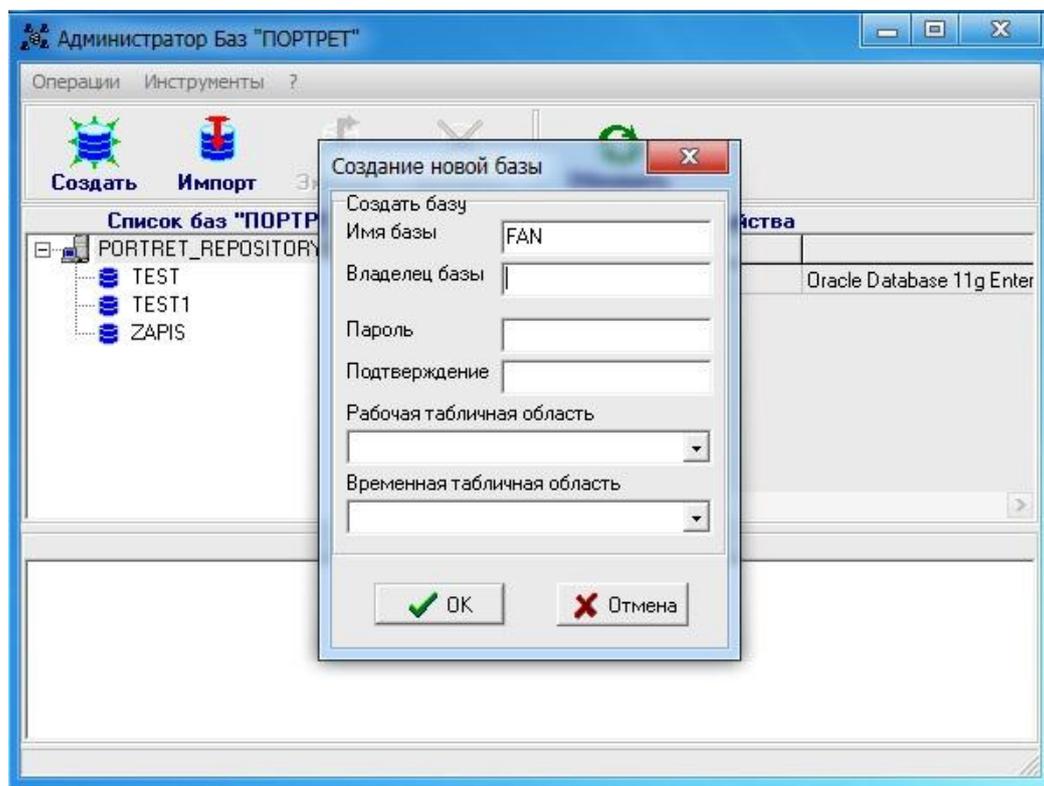


Рис. 4.1. Создание новой базы данных

После нажатия кнопки «Создать» опишем характеристики базы данных ФАНАТЫ (рис. 4.1). Рекомендуется, чтобы имя базы и владелец базы совпадали. Для примера, значение рабочей табличной области можно выбрать USERS, для временной табличной области – TEMP.

Создав базу данных, дальнейшую работу с ней будем осуществлять в приложении «Редактор структуры» (рис. 4.2), который позволит создать в ней объекты, характеризующиеся атрибутами, и установить между ними связи. Кроме того, объект может содержать набор форм, через которые пользователь в дальнейшем будет работать с базой.

Для создания структуры базы данных используются следующие кнопки панели инструментов:

-  – создать объект (Alt + O)
-  – создать атрибут (Alt + A)
-  – создать справочник (глобальный)
-  – новая форма (Alt + F)
-  – новая таблица (Alt + T)
-  – новый альбом (Alt + B)
-  – новая связь (Alt + L).

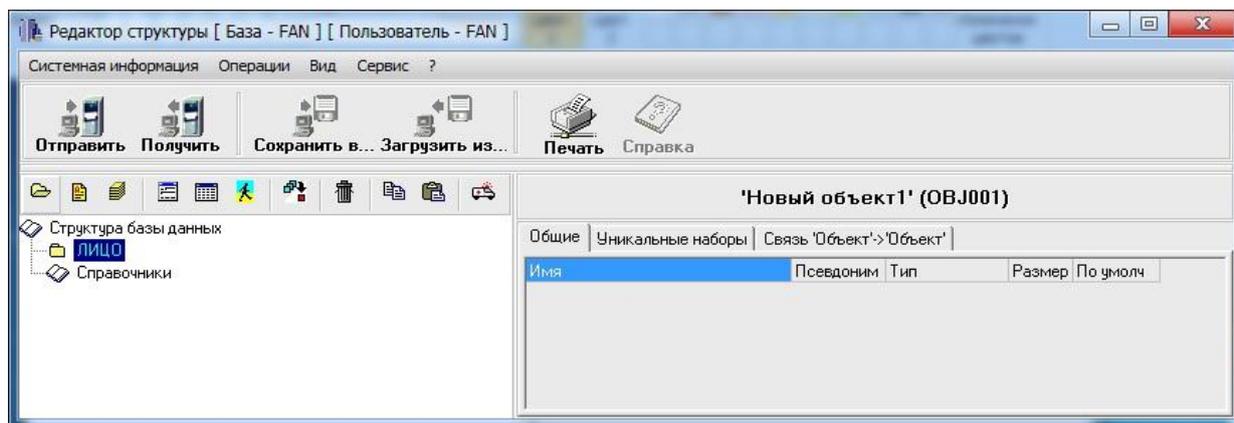


Рис. 4.2. Интерфейс приложения «Редактор структуры»

Для создания первого объекта ЛИЦО, необходимо нажать на кнопку «Создать объект» или нажать сочетание клавиш Alt + O.

При создании атрибутов и форм необходимо сначала мышкой выбрать объект, которому они будут принадлежать (рис. 4.3).

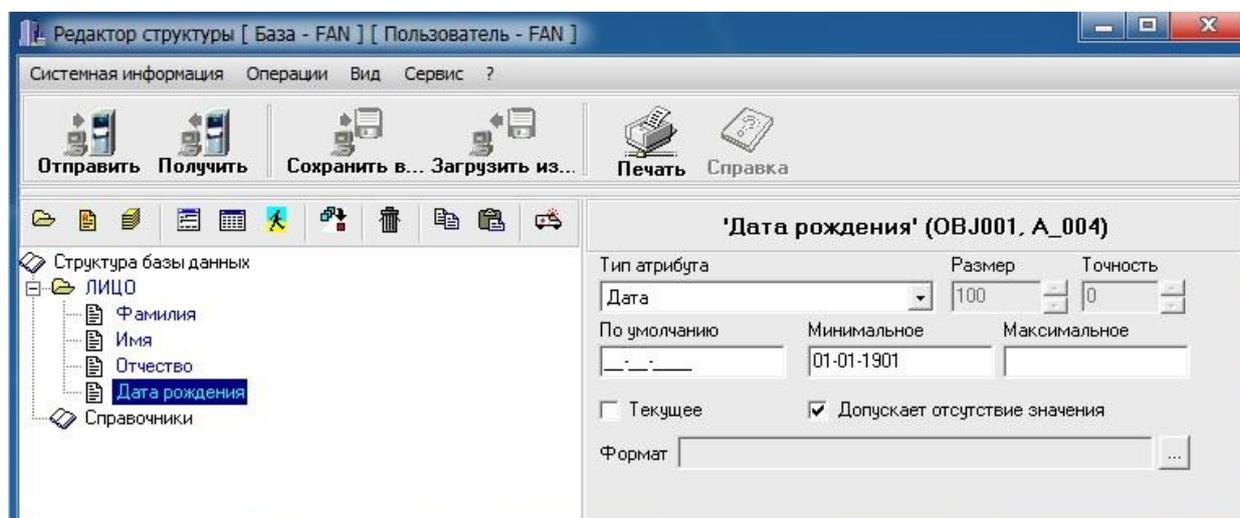


Рис. 4.3. Создание атрибутов объекта «Лицо»

Количество атрибутов может быть произвольным. Поле может быть текстовым, числовым, датой, хранить изображения и т.д.

Свойства атрибута отображаются на правой панели.

Тип атрибута можно выбрать из выпадающего списка «Тип атрибута». Атрибуты бывают следующих типов:

Число – подходит для представления чисел.

Дата – подходит для представления дат.

Строка (2000) – подходит для представления текстовых данных длиной до 2000 символов.

Строка неопределенной длины – подходит для представления текстовых данных очень большого размера.

Фото – подходит для представления изображений (форматы *.jpg, *.jpeg, *.tif, *.gif, *.bmp).

Время – подходит для представления времени.

Габитоскопия – специальный тип, предназначенный для хранения изображения с габитоскопическими точками.

Файл – для хранения файлов (максимальный размер файла 100 Мб).

Если поле допускает *отсутствие значения* (необязательное для заполнения), необходимо проставить «галочку» напротив «Допускает отсутствие значения», в противном случае – убрать.

Для текстовых и числовых полей можно задать *размер*, для текстовых полей это будет максимальная длина строки, а для чисел – число цифр в представлении числа.

Для полей типа ФОТО и ФАЙЛ можно задать ограничение на размер вводимых фотографий или файлов. Для фотографий исчисляется в Кб, для файлов – в Мб (диапазон размера файла от 1 до 100 Мб).

Для числовых полей с плавающей точкой после определения размера поля определяется его *точность* (количество знаков после запятой).

Для типов атрибута ВРЕМЯ, ЧИСЛО, СТРОКА можно определить значение по умолчанию, тогда при внесении данных, если для данного атрибута не указано значение, будет автоматически вноситься значение по умолчанию.

Для того чтобы в поле типа ДАТА автоматически вводилась текущая дата ввода, нужно поставить «галочку» напротив «Текущее», после чего в поле «Значение по умолчанию» (режим выбора свойств атрибута) появится TODAY.

Для атрибутов, имеющих тип ЧИСЛО, ДАТА и ВРЕМЯ предусмотрена возможность задание *ограничения на вводимые значения*. Для этого нужно ввести граничные значения в строки «Минимальное» и (или) «Максимальное».

Для всех типов, кроме ФОТО, ГАБИТОСКОПИЯ и ФАЙЛ можно определить набор действий, которые будут выполняться над значением поля при занесении в базу:

Для этого нажмите на кнопку «...» стоящую справа от строки «Формат» (рис. 4.4).

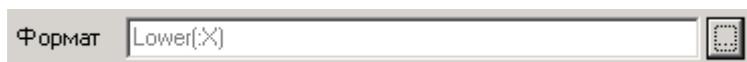


Рис. 4.4. Расположение кнопки в строке «Формат»

Появится окно «Новое значение», в нем нужно задать одну или несколько операций (рис. 4.5).

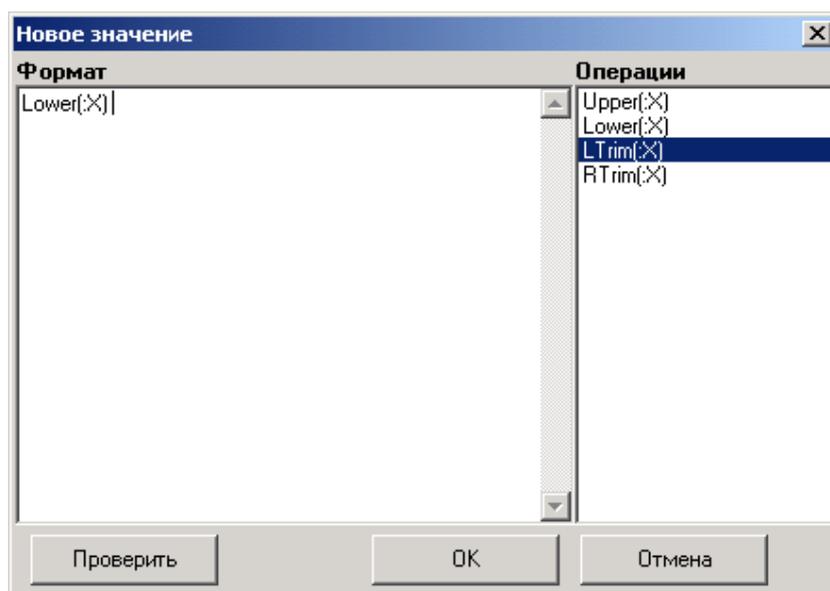


Рис. 4.5. Список функций

Можно выбрать некоторую функцию из списка, либо указать суперпозицию функций, также разрешено пользоваться всеми операторами, определенными синтаксисом SQL. Для проверки правильности задания выражения существует кнопка «Проверить».

Список функций содержит следующие операции (:X – входной параметр):

- Upper(:X) – приведение всех символов к верхнему регистру
- Lower(:X) – приведение всех символов к нижнему регистру
- LTrim(:X) – отрезает все пробелы с левой стороны
- RTrim(:X) – отрезает все пробелы с правой стороны
- LTrim(RTrim(:X)) – отрезает пробелы справа и слева.

Для атрибутов типа СТРОКА можно задать справочник. Максимальная длина справочного значения 200 символов.

Для использования одного справочника для нескольких объектов, рекомендуется создание глобального справочника.

Для создания глобального справочника нажмите кнопку «Новый справочник». Введите название для справочника. На правой панели с помощью кнопок добавления значения (подзначения) создайте справочник (рис. 4.6).

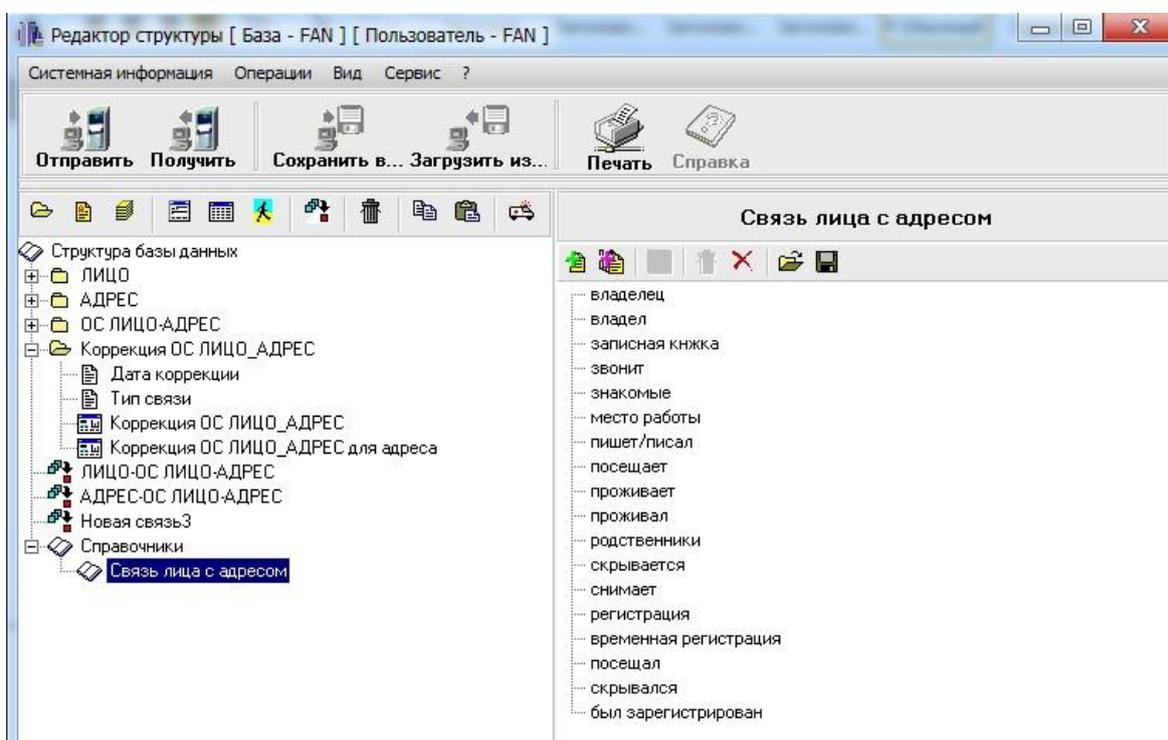


Рис. 4.6. Создание глобального справочника

Для того чтобы соединить такой «глобальный справочник» с атрибутом, выберите атрибут, сделайте тип атрибута *Глобальный*, выберите из выпадающего списка нужный «глобальный» справочник (рис. 4.7).

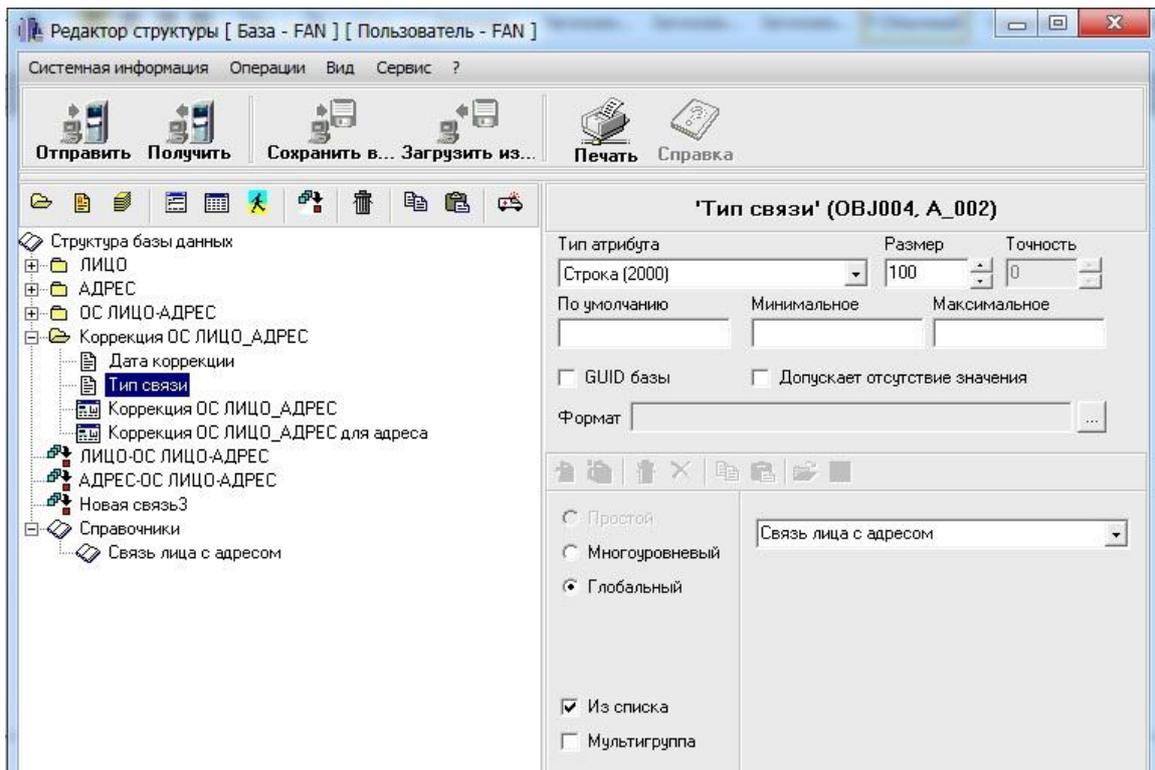


Рис. 4.7. Прикрепление глобального справочника к атрибуту

Если поле может принимать одновременно несколько значений (например, «кличка»), необходимо поставить «галочку» напротив «Мультигруппа».

Если нужно, чтобы пользователь мог только выбирать значения из справочника, и не вводил ничего самостоятельно, поставьте «галочку» напротив «Из списка».

Исключение дублирования данных при их занесении возможно путем определения уникальных наборов (например, если определить уникальный набор: ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ДАТА РОЖДЕНИЯ, МЕСТО РОЖДЕНИЯ, то в базе будет только один Иванов Иван Иванович 01.04.1970 г.р., уроженец г. Тюмени).

Можно определять несколько уникальных наборов для одного и того же объекта (рис. 4.8).

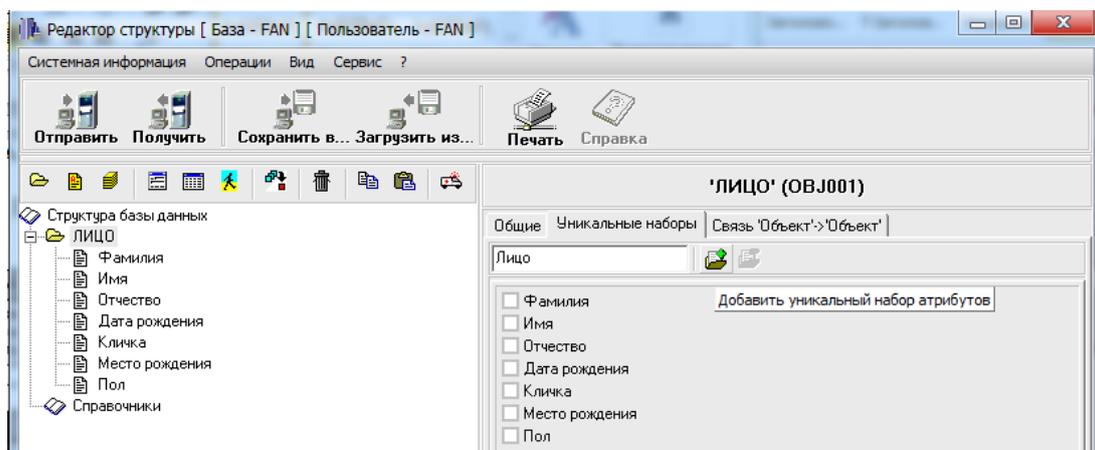


Рис. 4.8. Создание уникальных наборов

Для создания уникального набора выберите мышкой объект, для которого нужно завести этот самый уникальный набор, на правой панели выберете закладку «Уникальные наборы». Введите любое название для уникального набора. Нажмите

кнопку «Добавить уникальный набор атрибутов». Проставьте «галочки» напротив тех атрибутов, которые требуется включить в уникальный набор.

Как уже было сказано, объект может содержать экранные формы. Формы делятся на три типа: формы, таблицы и альбомы. Формы находятся в конце списка атрибутов объекта. Форма создается для текущего объекта.

Таблица – форма показывающая данные в табличном виде. Для ее создания выполните следующие действия:

– Поместив курсор мыши на значок объекта, щелкнуть правой кнопкой. Из списка меню выбирается «Таблица». Либо выбрать кнопку  на панели инструментов.

– Открыть форму и перетащить в нее требуемые атрибуты из окна «Атрибуты» (рис. 4.9).

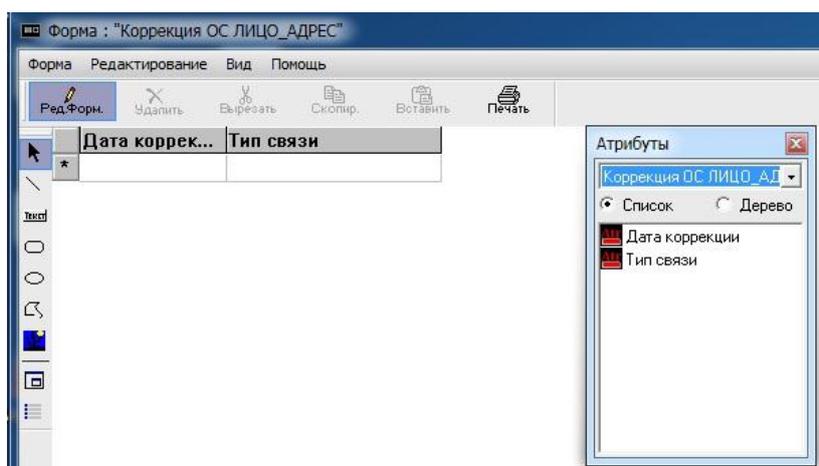


Рис. 4.9. Создание формы в виде таблицы

Альбом – форма, отображающая информацию в виде альбома. В списке меню выбрать «Альбом» либо нажать значок  на панели инструментов. Появится форма, в которую нужно перетащить нужные атрибуты из окна «Атрибуты». Простая форма содержит различные элементы, поля ввода, подформы и т.д. Форму можно создать либо из контекстного меню (щелкните правой клавишей мышки по целевому объекту), либо нажав кнопку  на панели инструментов, либо нажав «горячие» клавиши ALT + T.

При открытии экранной формы появляется окно «Атрибуты». Оно содержит все атрибуты (красные значки) объекта, для которого создается форма и связанные с ним другие объекты (зеленые значки). Для создания формы необходимо перетащить мышкой на форму нужные атрибуты (рис. 4.10).

На рисунке 4.10 на форму перенесены все атрибуты созданного нами объекта АДРЕС, а поля, входящие в уникальный набор, закрашены зеленым цветом.

Для отображения на форме информации о связанном объекте перетащите нужный связанный объект. При этом появится окно «Формы», спрашивающее какую форму от связанного объекта отобразить на форме (рис. 4.11).

В случае, когда количество полей достаточно велико, и они не вмещаются на видимую часть формы, рекомендуется условно разделить данные на закладки. Для того чтобы создать закладку: щелкните двойным щелчком мышки по свободному пространству формы, либо нажмите правую кнопку мышки на свободном пространстве формы и из контекстного меню выберете «Настройки». Появится окно «Form-Properties».

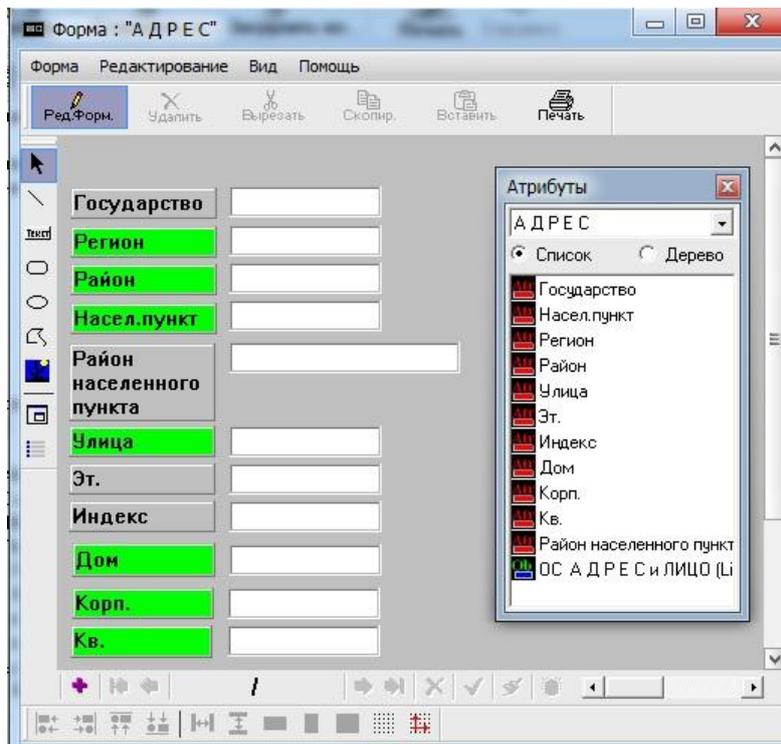


Рис. 4.10. Создание простой формы

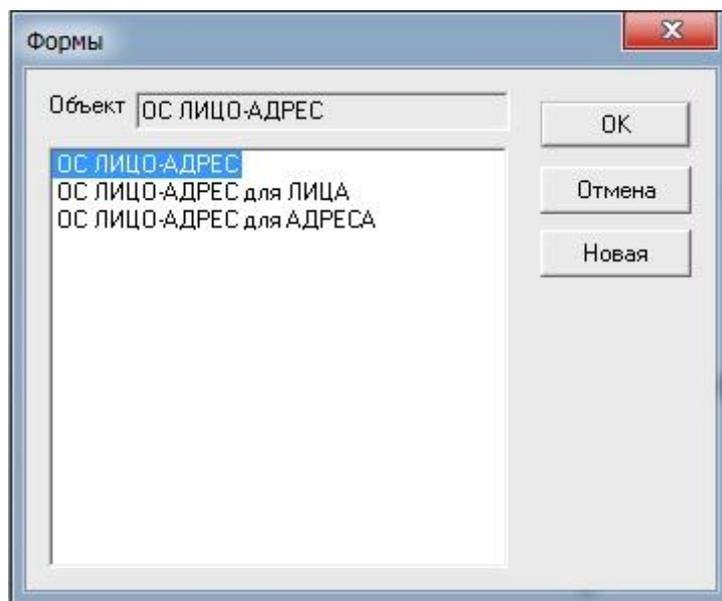


Рис. 4.11. Перенос на форму информации о связанном объекте

В нем перейдите на «Закладки», введите имя для новой закладки, нажмите кнопку «Добавить». Создайте нужное число закладок. Нажмите «Ок» (рис. 4.12).

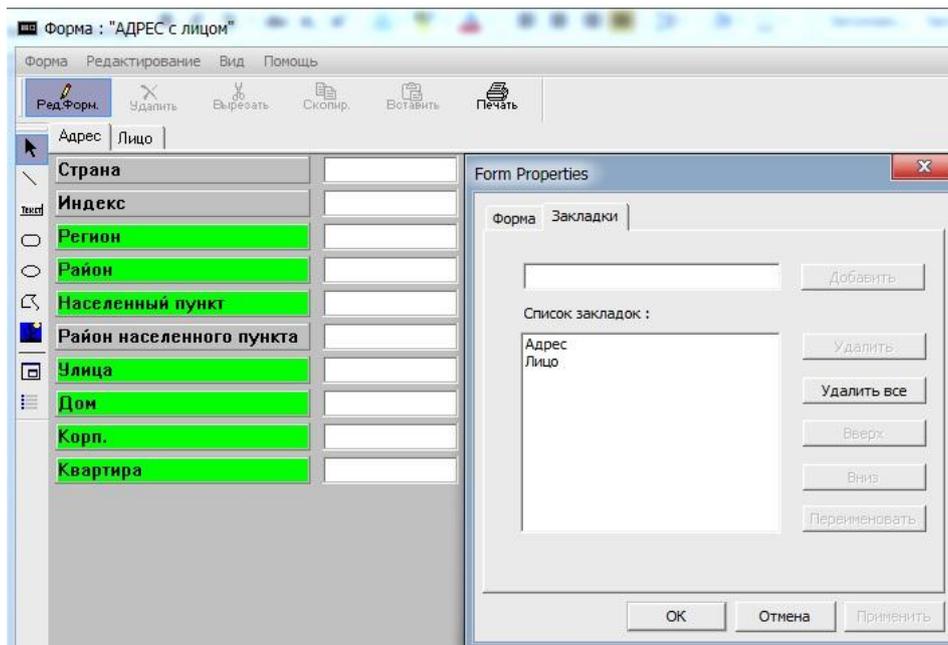


Рис. 4.12. Создание закладок для формы

Таким образом, нами созданы две закладки «Адрес» и «Лицо».

Между двумя объектами может быть установлена связь. При этом должен быть выделен *главный объект* и *подобъект*. В системе «Портрет» реализованы два типа связи: «один ко многим» и «многие со многими».

Создание связи «один ко многим» осуществляется путем создания связи между объектами.

- В главном меню программы выбрать «Создать связь» либо воспользоваться значком  панели инструментов.
- Ввести название связи.
- В списке «Главный объект» выбрать главный объект.
- В списке «Подчиненный объект» выбрать подчиненный объект.
- Пометить поле «Удаление связанных экземпляров объекта» (при этом при удалении экземпляра главного объекта будут удалены все связанные с ним экземпляры подчиненного объекта) (рис. 4.13).

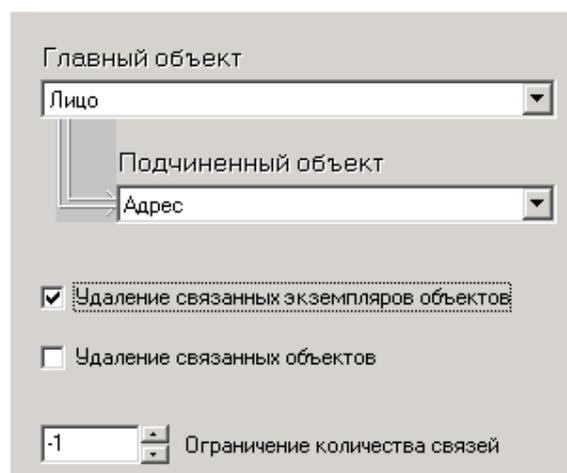


Рис. 4.13. Окно характеристики связи

Создание связи по типу «многие со многими» требует создания промежуточного (связующего) объекта, в который могут быть добавлены поля, описывающие тип и дату возникновения связи или какую-либо другую полезную информацию. Будем называть этот объект «объект связи» или сокращенно «ОС».

1. Создание ОС:

– Создаем объект связи и его название. При этом название должно быть информативным. Например, если это связующий объект объектов ЛИЦО и АДРЕС, то он может быть назван «ОС ЛИЦО-АДРЕС».

– Создать атрибут в объекте связи, например ХАРАКТЕР СВЯЗИ.

– Создать форму этого объекта, открыть ее и в режиме редактирования и перетащить в нее из окна «Атрибуты» атрибут ХАРАКТЕР СВЯЗИ.

2. Создание двух связей с ОС:

Создать две связи: от ЛИЦА – к ОС ЛИЦО-АДРЕС и от АДРЕСА – к ОС ЛИЦО-АДРЕС.

– Ввести название первой связи, например, ЛИЦО и ОС ЛИЦО-АДРЕС.

– В списке «Главный объект» выбрать главный объект (в данном случае ЛИЦО).

– В списке «Подчиненный объект» выбрать подчиненный объект (в данном случае ОС ЛИЦО-АДРЕС).

– Пометить поле «Удаление связанных экземпляров объекта».

– Точно также создаем вторую связь «АДРЕС и ОС ЛИЦО-АДРЕС»

(рис. 4.14).

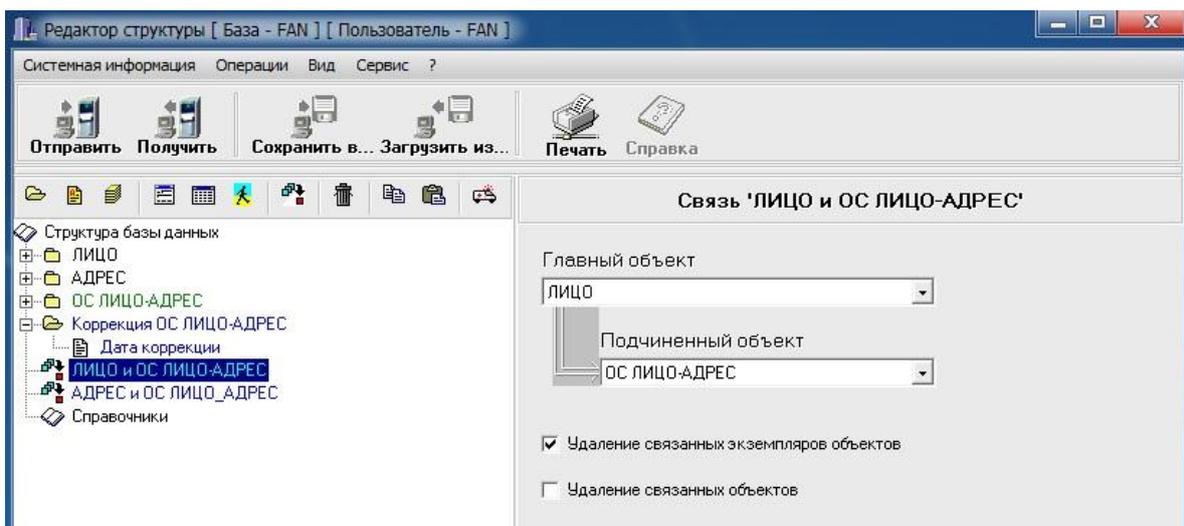


Рис. 4.14. Создание связи двух объектов через объект связи

3. Создание двух форм в ОС:

– В объекте связи ОС ЛИЦО-АДРЕС нужно создать еще две формы (в дальнейшем именно эти формы будут использоваться в многообъектных формах объектов ЛИЦО и АДРЕС).

– Делаем это так: копируем ранее созданную форму «ОС ЛИЦО-АДРЕС».

– Создаем вставлением новую форму, назвав ее, к примеру, «ОС ЛИЦО-АДРЕС для ЛИЦА».

– Открываем форму «ОС ЛИЦО-АДРЕС для ЛИЦА», переходим в режим редактирования и из окна «Атрибуты» перетаскиваем в нее объект АДРЕС.

– В списке экранных форм объекта АДРЕС выбираем нужную форму (для простоты лучше для начала использовать форму АДРЕС) и нажать «Ок». В результате этого в форме появится подформа, представляющая собой форму объекта АДРЕС.

– Точно также создаем форму «ОС ЛИЦО-АДРЕС для АДРЕСА».

4. Вставка форм из ОС ЛИЦО-АДРЕС в основные объекты (АДРЕС, ЛИЦО):

– Теперь, чтобы, например, в объекте ЛИЦО появился связанный объект АДРЕС, создаем в объекте ЛИЦО новую форму, например, ЛИЦО (ВСЕ ФОРМЫ) копированием формы ЛИЦО.

– Переходим в режим редактирования и из окна «Атрибуты» перетаскиваем в нее объект ОС ЛИЦО-АДРЕС.

– В списке экранных форм объекта «ОС ЛИЦО-АДРЕС» выбираем форму, предназначенную для связи с объектом АДРЕС (в нашем случае форма «ОС ЛИЦО-АДРЕС для ЛИЦА»), и нажимаем «Ок». В результате этого в форме появится подформа. Форма «ОС ЛИЦО-АДРЕС для ЛИЦА» также может быть вставлена в новую закладку.

– Аналогично создаем все для объекта АДРЕС.

Однако в рассматриваемом нами примере мы решили исходить из того, что характер связи между ЛИЦОМ и АДРЕСОМ будет периодически меняться. При этом ценность представляет не только характер связи в данный момент, но и в прошлом. Для отображения характера связи ЛИЦО-АДРЕС в зависимости от времени мы решили отказаться от указания атрибута ХАРАКТЕР СВЯЗИ в объекте ОС ЛИЦО-АДРЕС и создать дополнительный объект КОРРЕКЦИЯ ОС ЛИЦО-АДРЕС. Данный объект обладает атрибутами ДАТА КОРРЕКЦИИ и ТИП СВЯЗИ, содержит форму и таблицу, связан с объектом «ОС ЛИЦО-АДРЕС» (рис. 4.15).

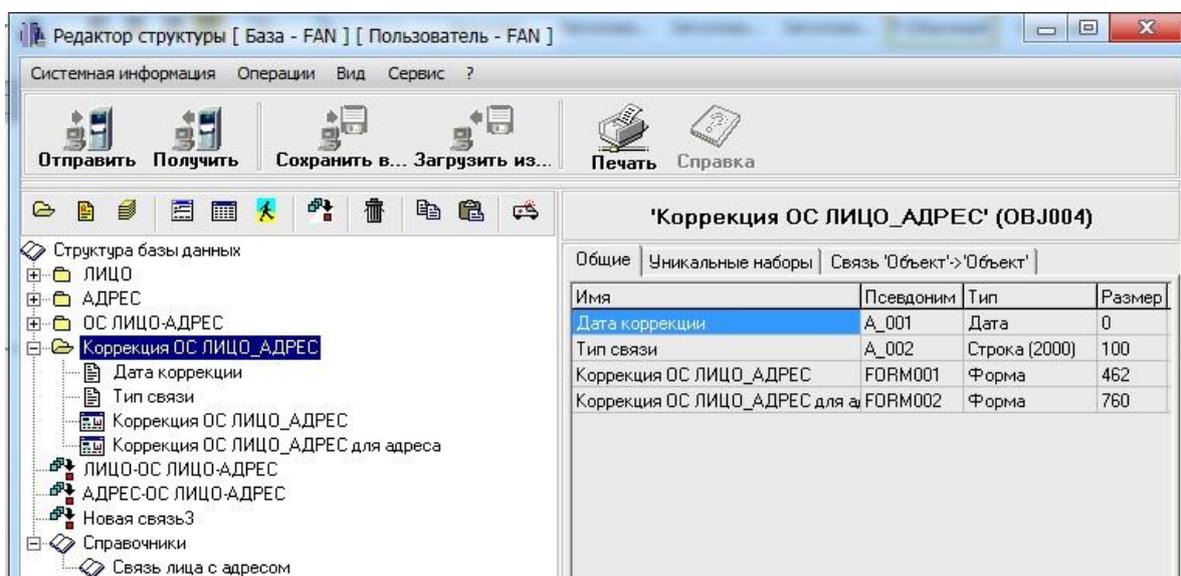


Рис. 4.15. Создание объекта коррекции связи

Теперь создадим форму ОС ЛИЦО-АДРЕС ДЛЯ ЛИЦА и поместим на нее формы КОРРЕКЦИЯ ОС ЛИЦО-АДРЕС, АДРЕС и таблицу КОРРЕКЦИЯ ОС ЛИЦО-АДРЕС (рис. 4.16).

Создание такой формы позволит оператору наблюдать изменения характера связи между объектами. (рис. 4.17).

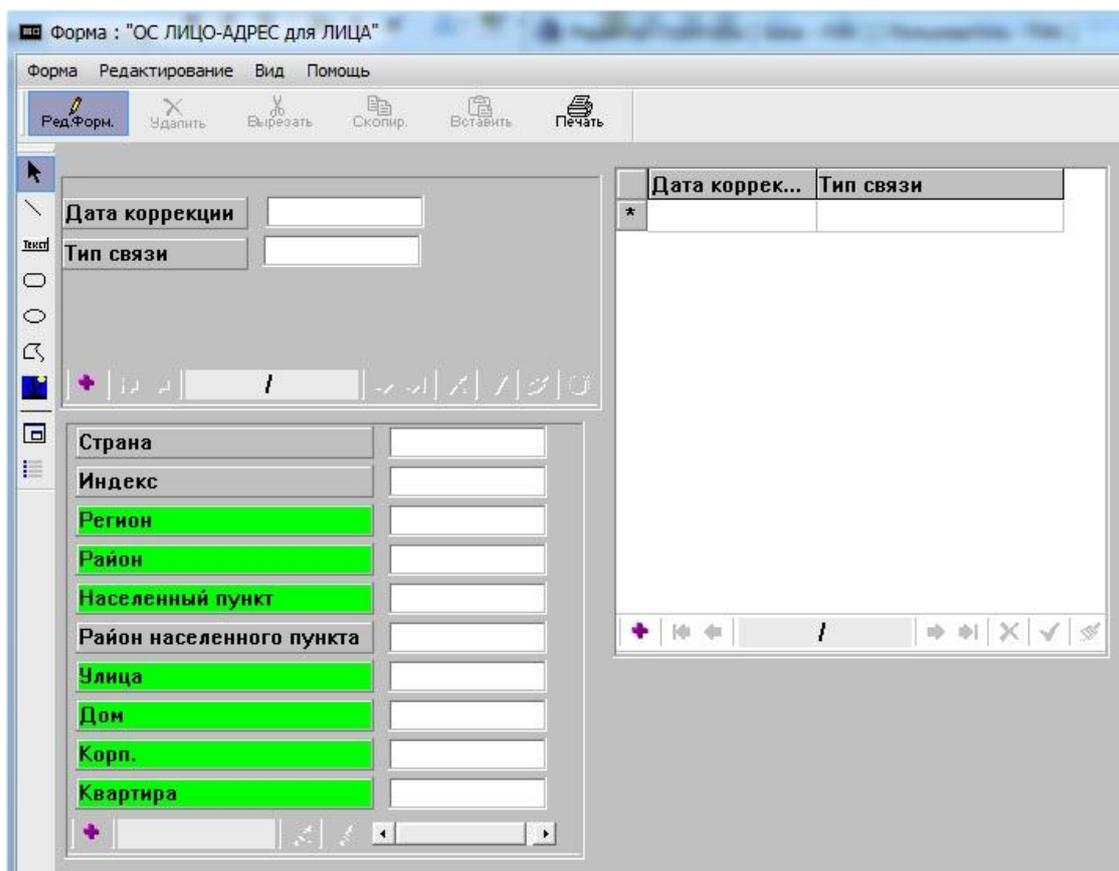


Рис. 4.16. Создания закладки «Адрес» в форме объекта «Лицо»

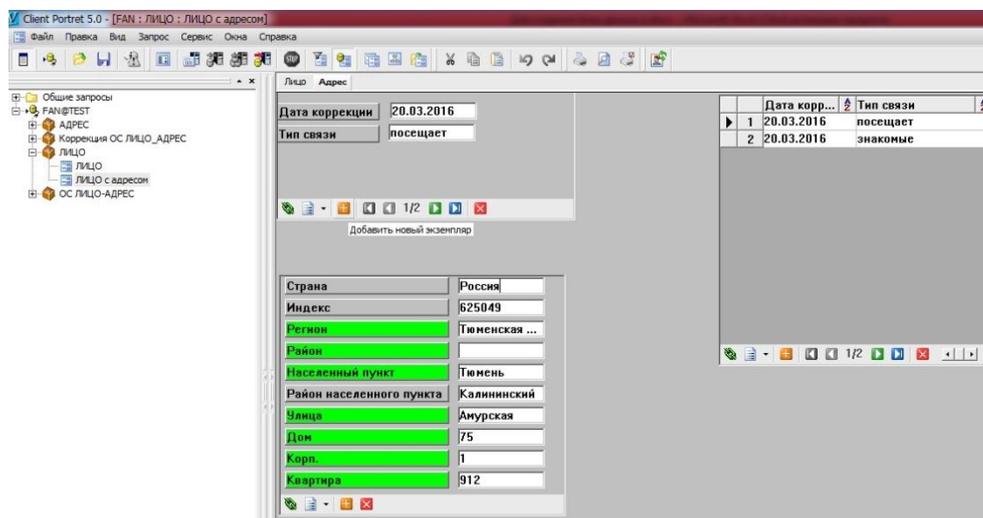


Рис. 4.17. Вид окна программы информационного обеспечения клиентского места системы «Портрет»

При создании базы данных ФАНАТЫ следует предусмотреть, что экземпляры объекта ЛИЦО могут быть связаны друг с другом. При этом характер их связи может меняться (были друзьями – стали врагами). Для создания такой связи необходимо:

1. Создать объект связи ОС ЛИЦО-ЛИЦО.
2. Создать две связи ЛИЦО – ОС ЛИЦО-ЛИЦО и ОС ЛИЦО-ЛИЦО – ЛИЦО, в каждой из которых главным объектом будет являться ЛИЦО, а подчиненным – ОС ЛИЦО-ЛИЦО.

3. Пометив объект связи ОС ЛИЦО-ЛИЦО, в правой части окна программы выбрать закладку «Связь ОБЪЕКТ⇒ОБЪЕКТ». В списке «Первая связь» выбрать ЛИЦО – ОС ЛИЦО-ЛИЦО. При этом в поле «Вторая связь» автоматически будет введено ОС ЛИЦО-ЛИЦО – ЛИЦО (рис. 4.18).

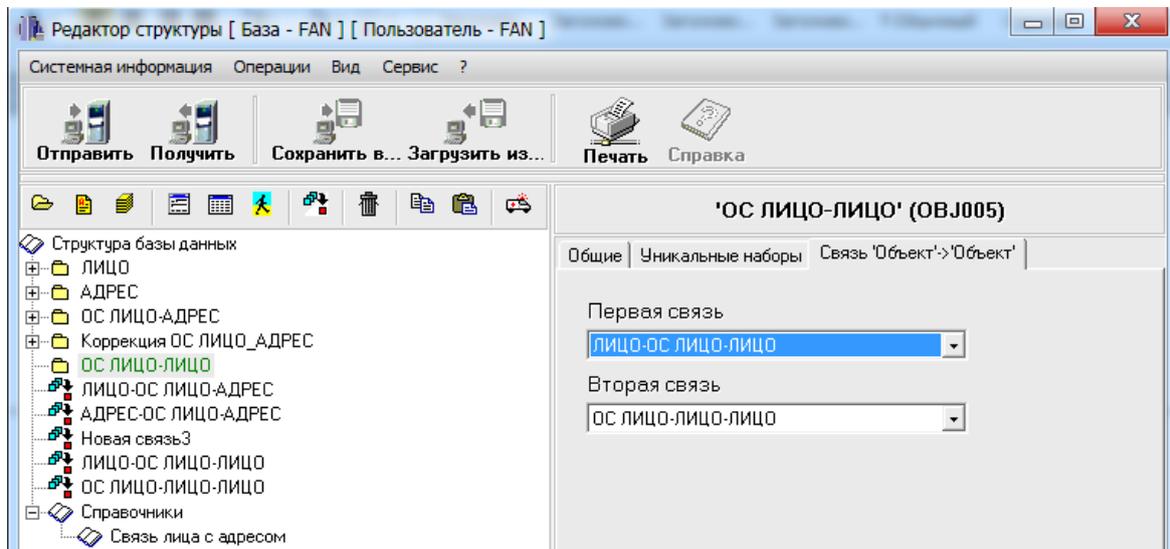


Рис. 4.18. Создание одинарной связи

4. Создадим объект КОРРЕКЦИЯ ОС ЛИЦО-ЛИЦО с атрибутами ДАТА КОРРЕКЦИИ и ТИП СВЯЗИ. Создадим для данного объекта форму и таблицу и свяжем его с объектом ОС ЛИЦО-ЛИЦО.

5. Создадим в объекте ОС ЛИЦО-ЛИЦО форму, в которую «перетащим» формы КОРРЕКЦИЯ ОС ЛИЦО-ЛИЦО, ЛИЦО и таблицу КОРРЕКЦИЯ ОС ЛИЦО-ЛИЦО.

6. Заходим в форму ЛИЦО (ВСЕ ФОРМЫ) объекта ЛИЦО, создаем новую закладку «Связи», на которую «перетаскиваем» объект ОС ЛИЦО-ЛИЦО (рис. 4.19).

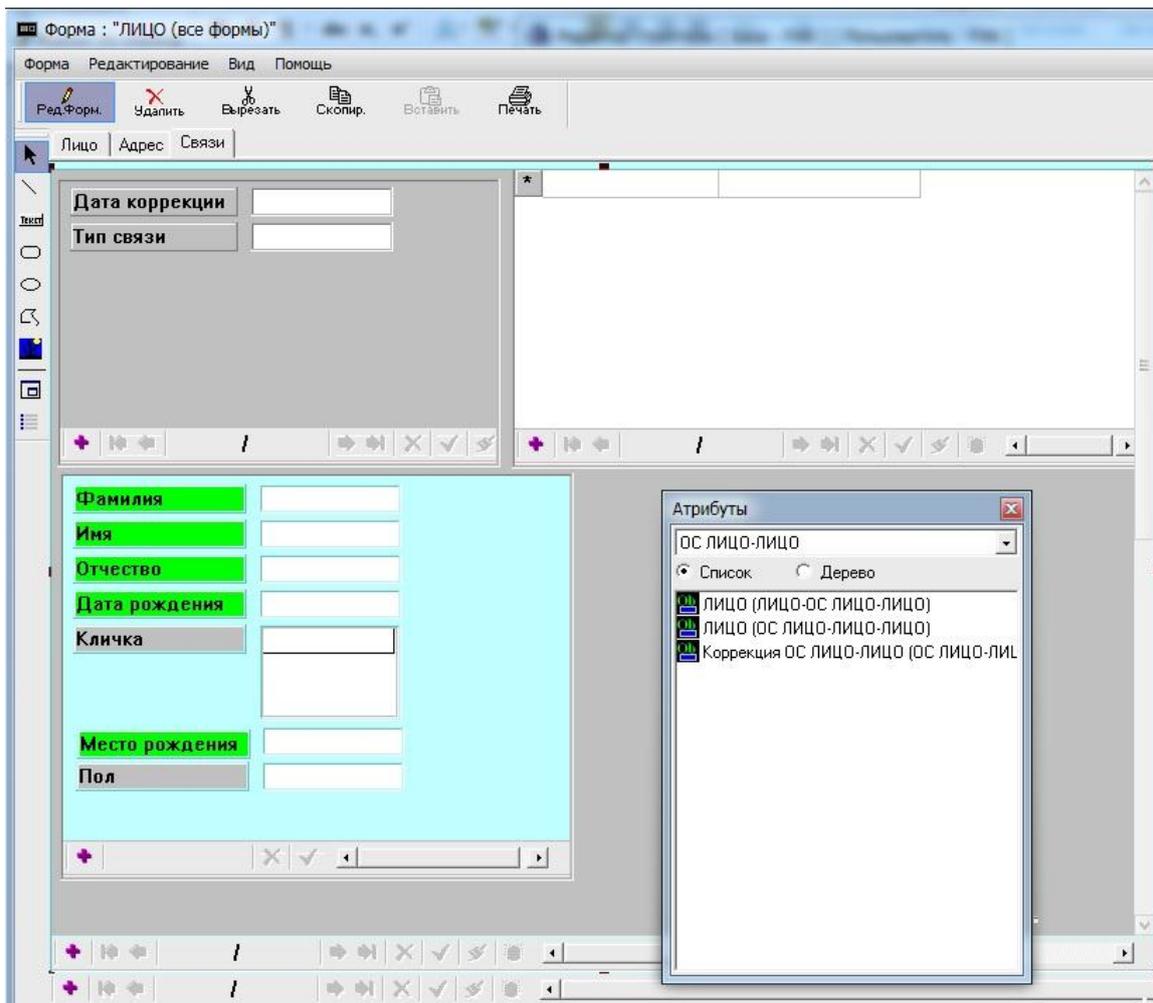


Рис. 4.19. Создания закладки «Связи» в форме объекта «Лицо»

Вопросы для самоконтроля

1. Какая программа из пакета информационного обеспечения администратора системы «Портрет» используется для создания базы данных в системе «Портрет-Поиск»?
2. Для чего необходимо создание объекта связи?
3. Опишите процедуру создания объекта связи в системе «Портрет-Поиск».
4. Опишите процедуру создания связи между двумя экземплярами одной сущности в системе «Портрет-Поиск».

Список литературы

1. Дейт К.Дж. Введение в системы баз данных / пер. с англ.; 8-е изд. М.: Вильямс, 2005. 1328 с.
2. Информационные системы и технологии в профессиональной деятельности: учеб. пособие. Хабаровск: Дальневосточный юрид. ин-т МВД России, 2012.
3. Кудрявцев К.Я. Создание баз данных: учеб. пособие. М.: НИЯУ МИФИ, 2010. 155 с.
4. Швецов В.И., Визгунов А.Н., Мееров И.Б. Базы данных: учеб. пособие. Н. Новгород: Изд-во ННГУ, 2004. 217 с.

Для заметок

Учебное издание

Панюшин Денис Борисович,
Щербич Сергей Викторович

Информационные системы: назначение, создание, использование
в деятельности подразделений оперативно-разыскной информации

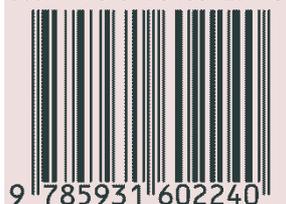
Учебное пособие

Корректурa и редактирование: *Е.В. Шабанова*
Техническое редактирование: *Е.К. Булатова*

Подписано в печать 24.08.2016. Формат 60x84/8
Усл. п. л. 4,53. Тираж 130 экз. Заказ № 91.

Редакционно-издательское отделение
Тюменского института повышения квалификации
сотрудников МВД России
625049, г. Тюмень, ул. Амурская, 75.

ISBN 978-5-93160-224-0



9 785931 602240