

Министерство образования и науки Украины

Национальный технический университет
«Харьковский политехнический институт»

**Методические указания к проведению лабораторных занятий по курсу
«Организация баз данных и знаний»**

Использование пакета ERwin для моделирования данных

**для студентов, обучающихся по направлению
6.050101 «Компьютерные науки и информационные технологии»**

Харьков 2015

ВВЕДЕНИЕ

Цель: Постановка задачи лабораторного практикума. Изучение и анализ предметной области. Выделение на основе анализа основных объектов базы данных.

Общее (упрощенное) описание предметной области

Некоторое предприятие приобретает продукцию у различных поставщиков. Поставщиками могут быть как юридические лица, так и физические лица – субъекты предпринимательской деятельности. Приобретение продукции осуществляется партиями и оформляется в виде договоров на поставку. Каждый договор на поставку продукции имеет уникальный номер и может быть заключен только с одним поставщиком. Основанием для поставки является какой-либо документ (предварительный заказ, счет-фактура и т.п.). В документах по каждому договору для каждого вида продукции указываются: наименование, размер поставленной партии и цена (в грн.). Поставленная продукция приходуется на склад с целью последующей реализации путем продаж, отпуска на реализацию и т.п.

Анализ предметной области позволил выделить и детализировать основные бизнес-процессы, связанные с поставками продукции. Предполагается, что в процессе работы предприятия придется хранить достаточно большие объемы данных, связанные с поставками продукции. Кроме того, информация о поставках продукции должна быть организована таким образом, чтобы персонал и руководство предприятия имели возможность осуществлять ее аналитическую обработку. В связи с этим для хранения и обработки информации, связанной с поставками продукции, необходимо создать базу данных. Анализ бизнес-процессов позволил выделить следующие информационные массивы, которые могут входить в состав такой базы данных.

1. Сведения о продукции

Включают в себя информацию о названии продукции, идентификаторе, который однозначно определяет каждую конкретную продукцию (код, артикул и т.п.), единицу измерения продукции (штука, ящик, кг и т.п.) и т.д. Продукция объединена в некоторые группы (товарные группы) – например, продукты питания, парфюмерия, бытовая химия и т.д. Предполагается, что каждая продукция может входить только в одну товарную группу.

2. Сведения о поставщиках продукции

Включают в себя информацию о субъектах предпринимательской деятельности, работающих на рынке и предлагающих продукцию, в приобретении которой заинтересовано рассматриваемое предприятие. Поставщиками могут быть как юридические лица, так и физические лица. К сведениям о поставщиках относятся такие данные, как название субъекта предпринимательской деятельности, индивидуальный налоговый номер, номер свидетельства плательщика НДС (для юридического лица); фамилия, имя, отчество, номер свидетельства о регистрации (для физического лица); адрес местонахождения, контактный телефон (для физического и юридического лица) и т.д.

3. Сведения о ценах на продукцию, предлагаемую различными поставщиками

Одна и та же продукция может предлагаться на рынке различными поставщиками, причем каждый поставщик может предлагать продукцию по разным ценам (розничной, мелкооптовой, оптовой и т.д.) в зависимости от объема закупки, условий договора и т.д.

4. Сведения о поставках продукции

Каждая поставка осуществляется на основании договора на поставку, который заключается между поставщиком и предприятием. Для каждой поставки известна следующая информация: поставщик, дата поставки, общая сумма поставки, данные о поставленной продукции. Данные о поставленной продукции включают в себя (по каждой продукции): наименование продукции, количество единиц, цену за единицу. Цена, по которой поставляется продукция, может отличаться от стандартного прайс-листа поставщика (для конкретного покупателя могут действовать специальные скидки, цена на отдельные виды продукции может назначаться индивидуально и т.д.)

На основе анализа перечисленных информационных массивов можно выделить следующие таблицы (сущности), которые будут входить в состав разрабатываемой базы данных. Для каждой таблицы (сущности) также приводятся поля (атрибуты), конкретизирующие хранимую в таблицах информацию.

1. Товарные группы

1.1. Код товарной группы

1.2. Название товарной группы

2. Единицы измерения продукции

2.1. Код единицы измерения

2.2. Название единицы измерения

3. Продукция

3.1. Код продукции

3.2. Название продукции

3.3. Товарная группа

3.4. Единица измерения

4. Виды цен на продукцию

4.1. Код вида цены

4.2. Название вида цены

5. Поставщики

5.1. Код поставщика

5.2. Название поставщика (для юридического лица)

5.3. Индивидуальный налоговый номер (для юридического лица)

5.4. Номер свидетельства плательщика НДС (для юридического лица)

5.5. Фамилия, имя, отчество (для физического лица)

5.6. Номер свидетельства о регистрации (для физического лица)

5.7. Адрес

5.8. Контактный телефон

6. Цены на рынке

6.1. Продукция

6.2. Поставщик

6.3. Вид цены

6.4. Значение цены

- 6.5. Условие предоставления цены
- 7. Договоры на поставку продукции
 - 7.1. Код договора
 - 7.2. Дата поставки
 - 7.3. Поставщик
 - 7.4. Комментарий (некоторая дополнительная информация о поставке)
- 8. Поставленная продукция
 - 8.1. Код договора
 - 8.2. Продукция
 - 8.3. Поставленное количество
 - 8.4. Цена за единицу

Приведенная выше информация может быть использована как исходная при разработке модели данных IDEF1X.

Часть 1

Цель работы: Построение моделей IDEF1X.

Выполнение работы.

1. Создание логической модели:
 - 1) запустить ERWin и создать новую модель, выбрав пункт New (File -> New);
 - 2) в окне «Create Model – Select Template» установить переключатель «New Model Type» в положение «Logical/Physical»;
 - 3) в разделе «Target Database» установить «Database» – Access и «Version» – 2000. После этого нажать кнопку «OK»;
 - 4) в результате на экране появится интерфейс пакета ERWin (рис. 1).
2. Создание сущностей и атрибутов.

Рассмотрим последовательность действий при создании сущности и ее атрибутов на примере сущности «Товарные группы»:

 - 1) щелчком мыши нажать кнопку «Entity» на панели инструментов;
 - 2) перенести курсор к тому месту экрана, где необходимо разместить сущность и щелкнуть левой кнопкой мыши. На экране появится изображение сущности с произвольным именем (рис.2);
 - 3) изменить имя сущности на требуемое, для чего щелкнуть по изображению сущности правой кнопкой мыши и выбрать пункт «Entity Properties ...». В появившемся окне «Entities» ввести новое название сущности – «Товарные_группы» (без кавычек) (рис. 3) и нажать кнопку «OK»;
 - 4) ввести атрибуты сущности, для чего щелкнуть по изображению сущности правой кнопкой мыши и выбрать пункт «Attributes...». В результате на экране появится окно «Attributes» (рис. 4). Щелкнуть мышью по кнопке «New» и в появившемся окне «New Attribute» ввести имя атрибута – «код_товарной_группы» (без кавычек). В разделе окна «Domain» установить тип «Number» (рис. 5) и нажать кнопку «OK». Ввести второй атрибут –

«название_товарной_группы» (без кавычек). Для него в «Domain» выбрать тип «String»;

- 5) установить первичный ключ для данной сущности. Для этого выбрать атрибут «код_товарной_группы» и щелкнуть по переключателю «Primary Key» (рис. 6);
- 6) щелкнуть в окне «Attributes» по кнопке «OK». Сущность «Товарные_группы» создана (рис. 7).

Остальные сущности, которые нужно создать на данном этапе, создаются аналогично. Данные о создаваемых сущностях и атрибутах приведены в таблице 1.

Таблица 1

Сущность	Атрибут	Ключ	Domain
Единицы_измерения	код_единицы_измерения	PK	Number
	название_единицы_измерения		String
Продукция	код_продукции	PK	Number
	название_продукции		String
Поставщики	код_поставщика	PK	Number
	адрес		String
	телефон		String
Виды_цен	код_вида_цены	PK	Number
	название_вида_цены		String
Договоры	код_договора	PK	Number
	дата_поставки		Datetime
	комментарий		String
Поставленная_продукция	количество		Number
	цена_за_единицу		Number
Юр_лица	название		String
	налоговый_номер		String
	номер_свид_НДС		String
Физ_лица	фамилия		String
	имя		String
	отчество		String
	номер_свидетельства		String

В результате создания сущностей и их размещения на экране, внешний вид логической модели должен соответствовать виду, приведенному на рис. 8.

3. Сохранить созданную модель, для чего щелкнуть мышью по кнопке «Save model» в панели инструментов и ввести имя файла – «delivery» (без кавычек).

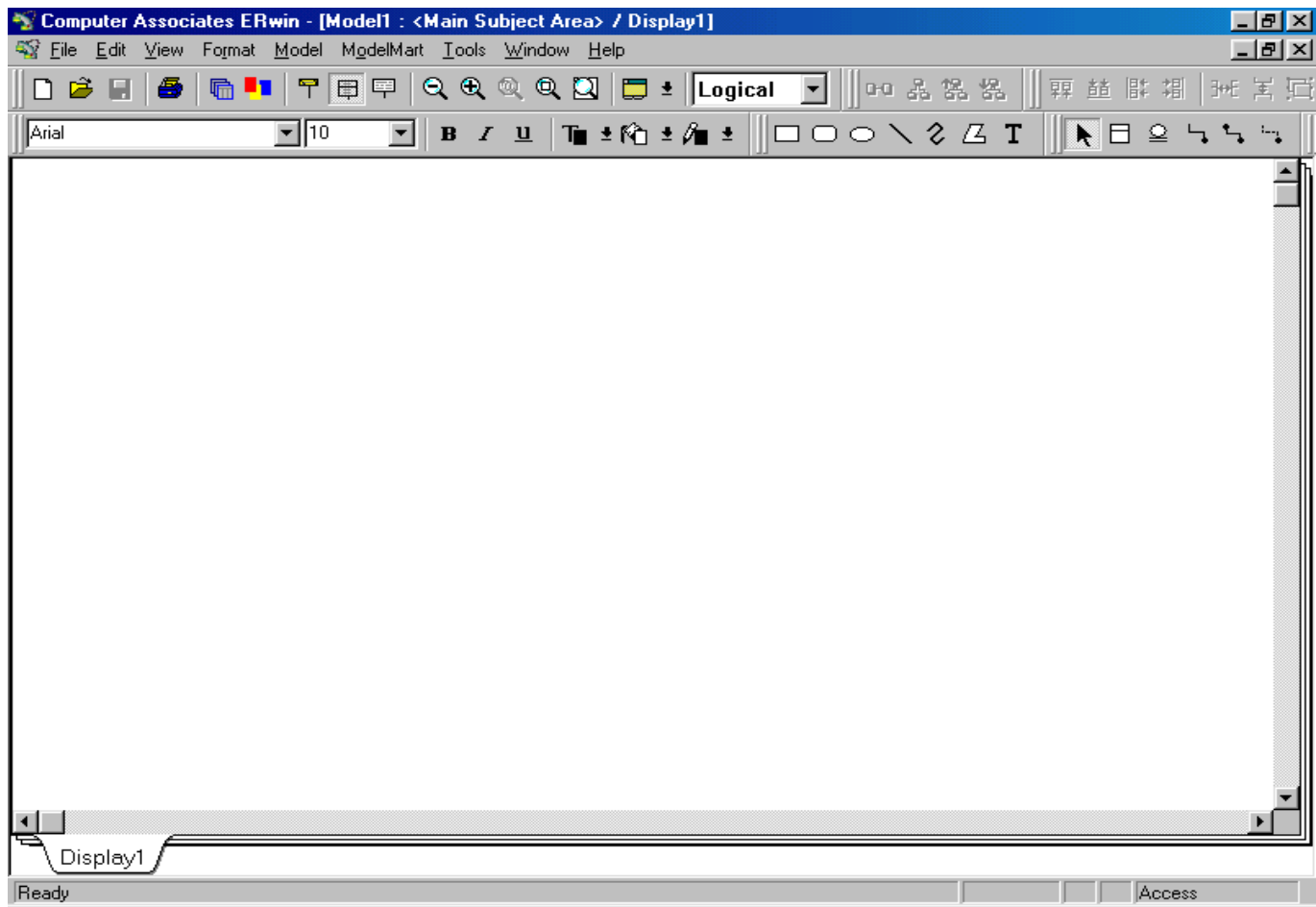


Рисунок 1

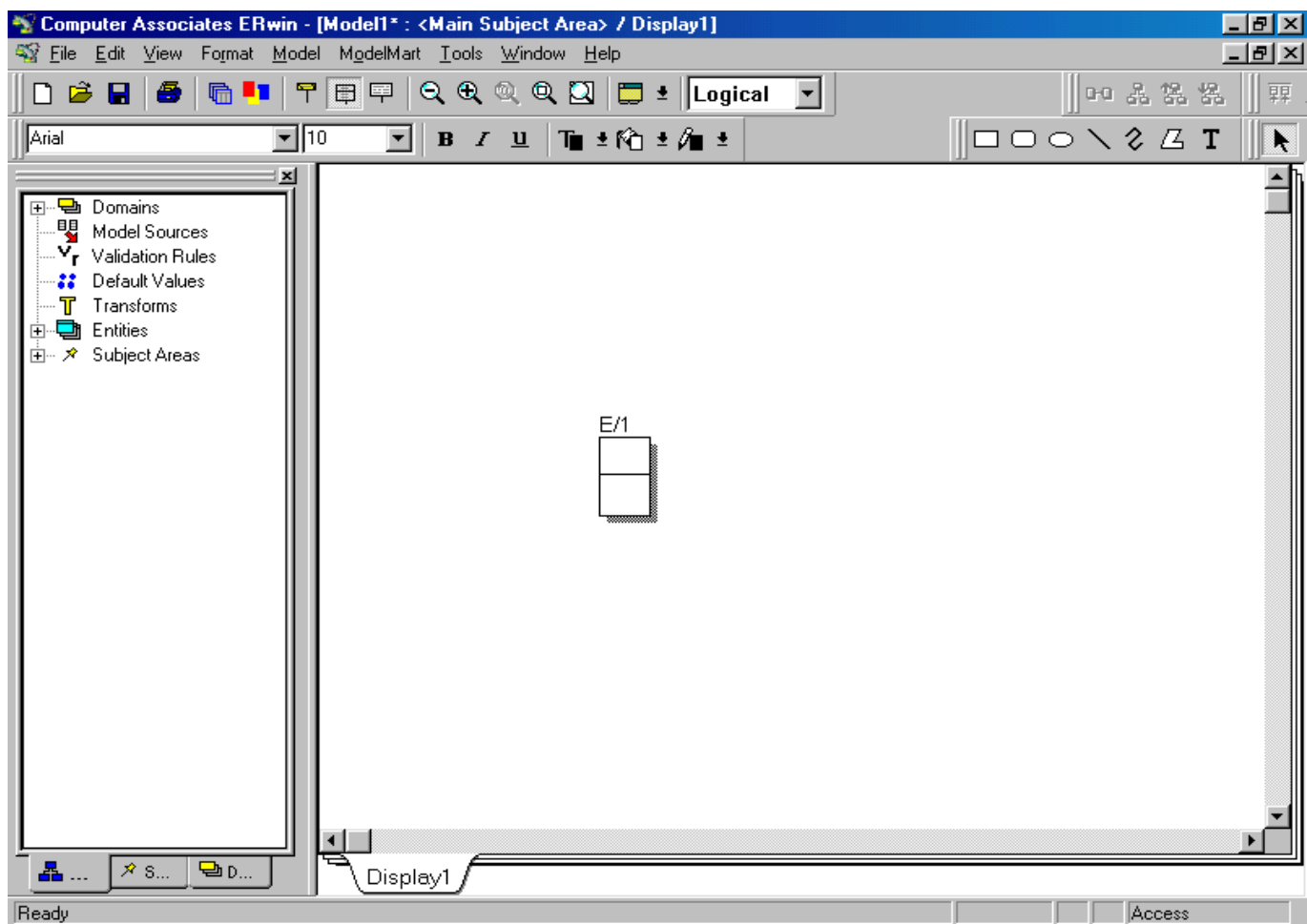
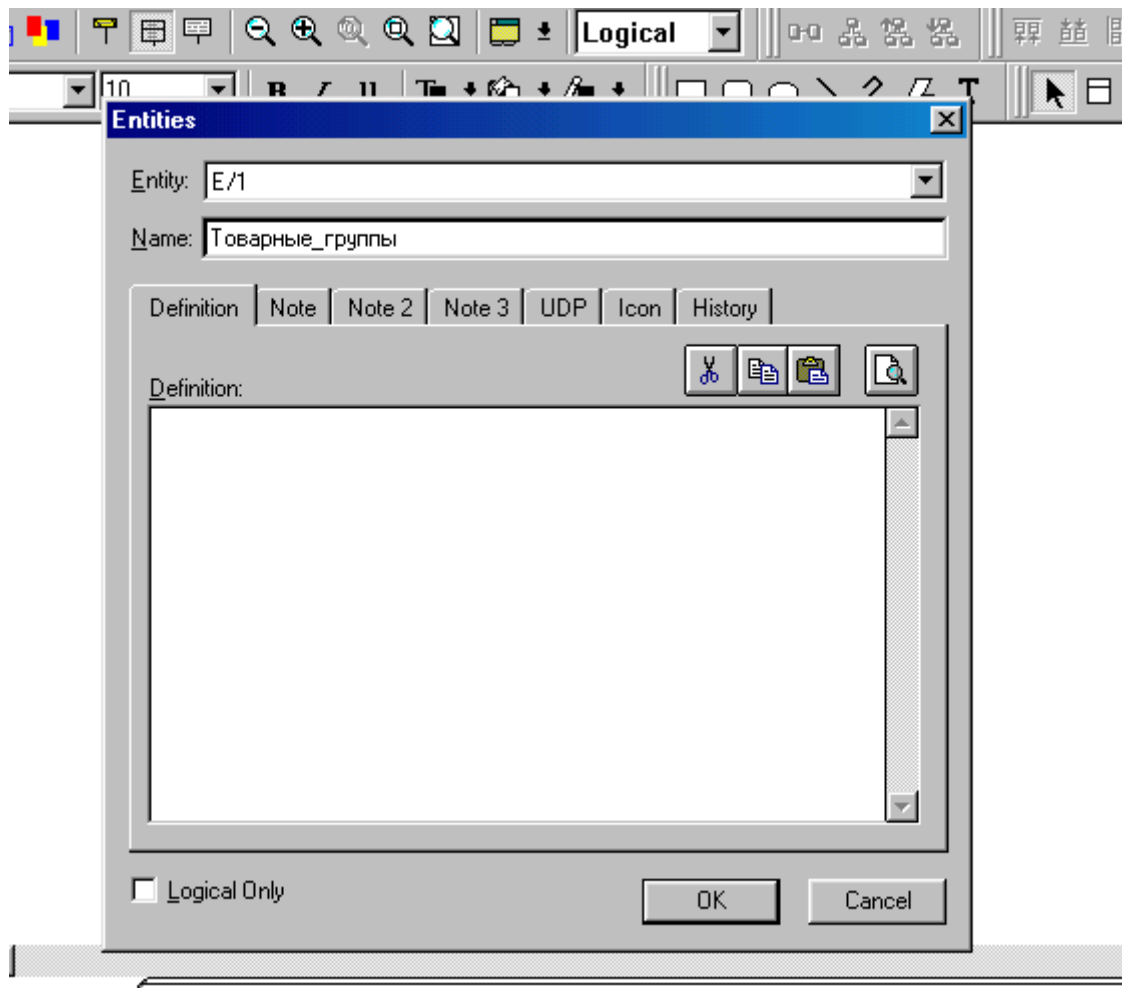


Рисунок 2



Display1

Рисунок 3

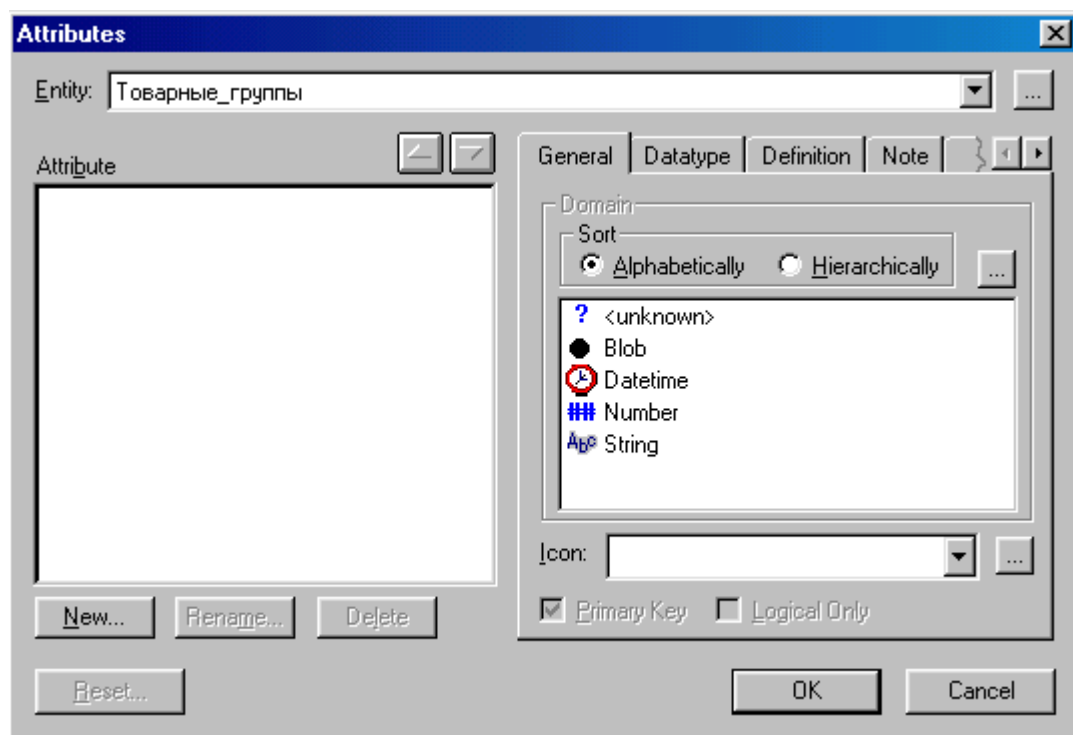


Рисунок 4

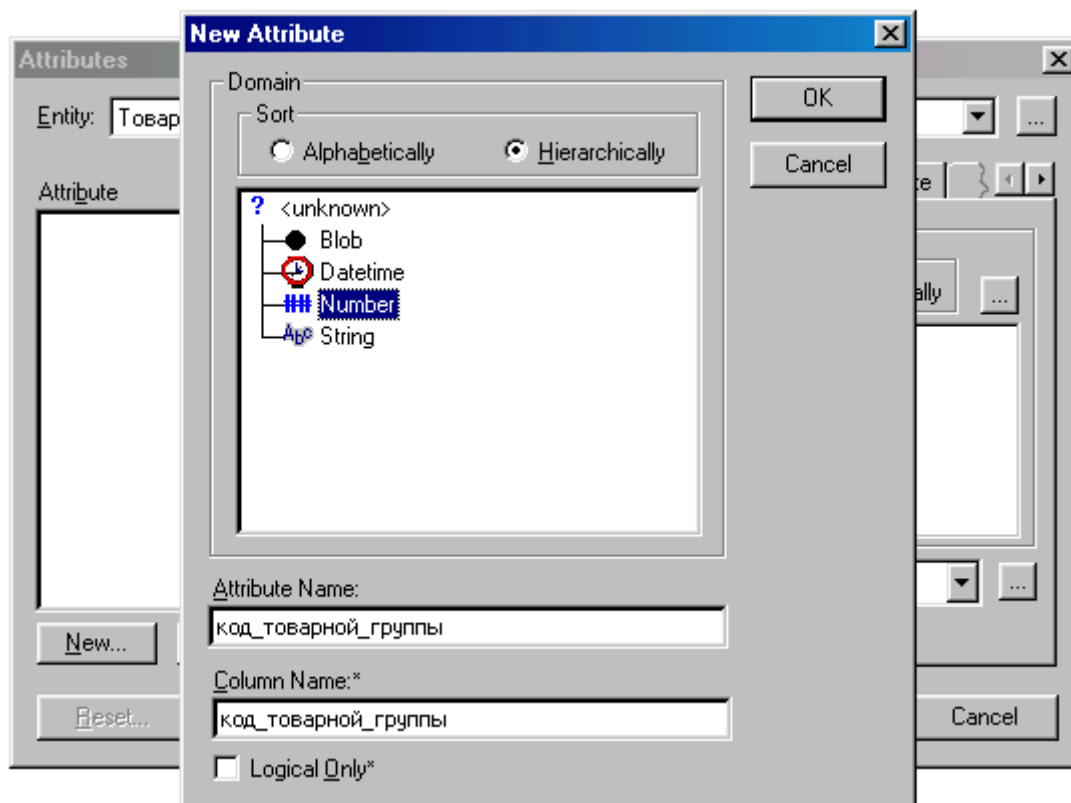


Рисунок 5

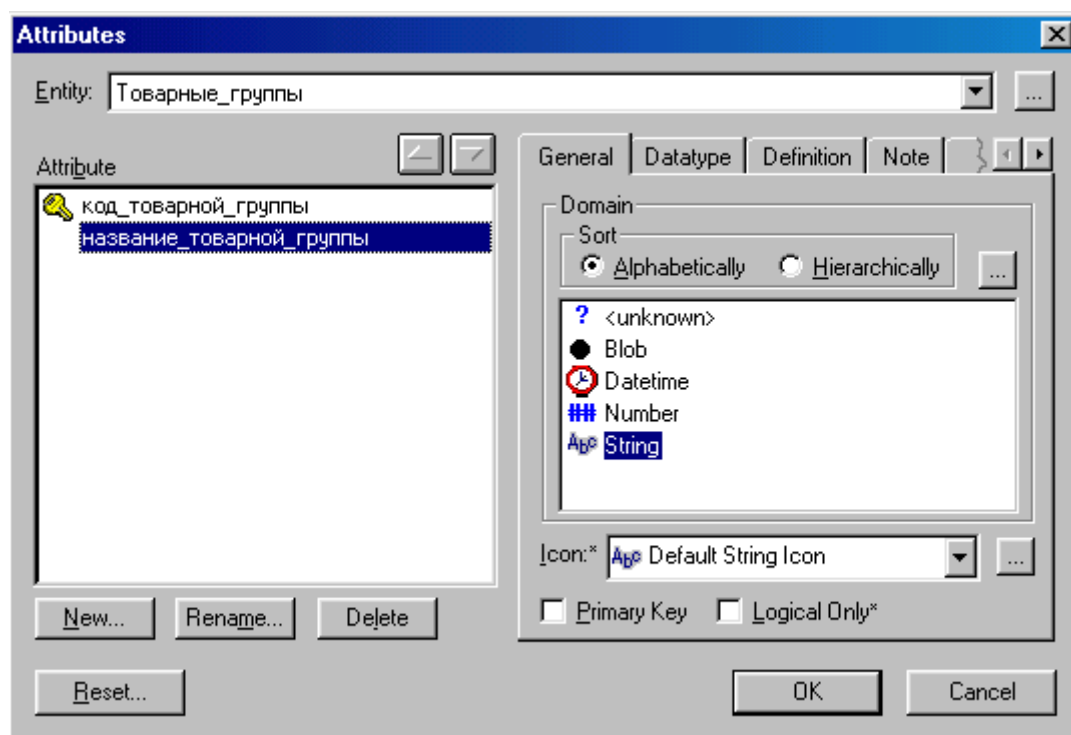


Рисунок 6

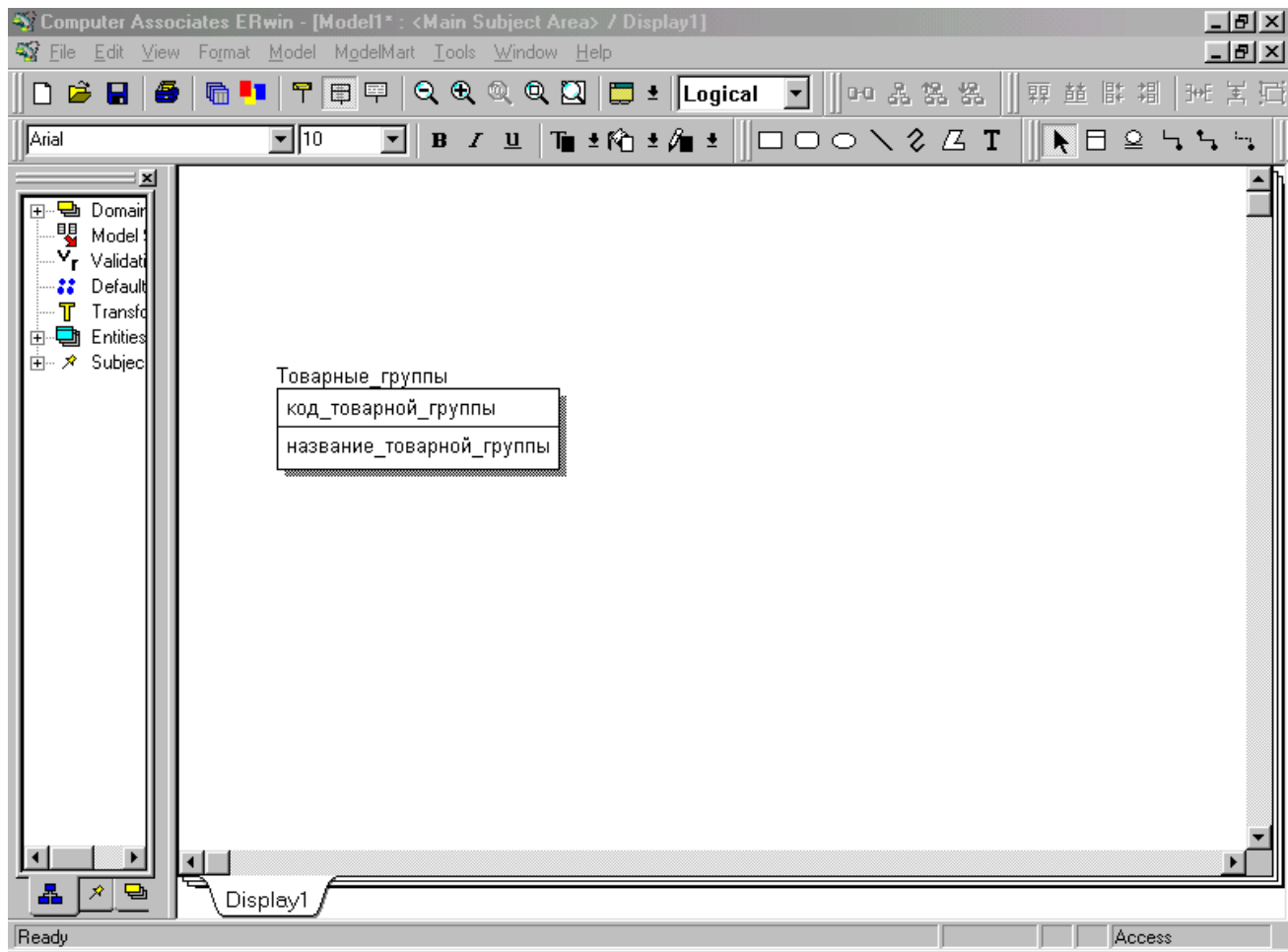


Рисунок 7

Поставщики	
код_поставщика	
адрес	
контактная_информация	

Договоры	
код_договора	
дата_поставки	
комментарий	

Юр_лица	
название	
налоговый_номер	
номер_свид_НДС	

Физ_лица	
фамилия	
имя	
отчество	
номер_свидетельства	

Поставленная_продукция	
количество	
цена_за_единицу	

Продукция	
код_продукции	
название_продукции	

Виды_цен	
код_вида_цены	
название_вида_цены	

Товарные_группы	
код_товарной_группы	
название_товарной_группы	

Единицы_измерения	
код_единицы_измерения	
название_единицы_измерения	

Рисунок 8

4. Создание связей

Рассмотрим последовательность действий при создании связи между сущностями на примере создания неидентифицирующей связи между сущностями «Продукция» и «Товарные группы»:

- 1) щелкнуть мышью по кнопке «Non-identifying relationship» в панели инструментов;
- 2) щелкнуть мышью сначала по сущности «Товарные группы», а затем по сущности «Продукция». В результате между сущностями появится неидентифицирующая связь (рис. 9). Атрибут, выполняющий функции первичного ключа (Primary Key) сущности «Товарные группы» мигрирует и будет выполнять функции внешнего ключа (Foreign Key) сущности «Продукция». Связь является неидентифицирующей, поскольку внешний ключ не входит в состав первичного ключа.

Примечание.

В том случае, если связь отображается на диаграмме не так, как требуется, щелкните по связи мышью и, не отпуская левую кнопку мыши, перетяните связь в требуемое положение.

Аналогично можно создать связи между созданными ранее сущностями. Перечень связей между сущностями и типы связей приведены в таблице 2.

Таблица 2

Родительская сущность	Дочерняя сущность	Тип связи	Кнопка
Товарные_группы	Продукция	неидентифицирующая	Non-identifying relationship
Единицы_измерения	Продукция	неидентифицирующая	Non-identifying relationship
Поставщики	Договоры	неидентифицирующая	Non-identifying relationship
Продукция	Поставленная_продукция	идентифицирующая	Identifying relationship
Договоры	Поставленная_продукция	идентифицирующая	Identifying relationship

В результате создания связей разрабатываемая диаграмма должна принять вид, аналогичный рис. 10.

5. Создание категориальных связей для дочерних сущностей в иерархии наследования

Иерархия наследования (или иерархия категорий) представляет собой особый тип объединения сущностей, которые разделяют общие характеристики. В рассматриваемом примере это относится к сущности «Поставщики». Поскольку поставщиками могут быть как юридические, так и физические лица, хранить информацию о поставщиках—физических лицах и поставщиках—юридических

лицах с помощью одной сущности нецелесообразно. В связи с этим ранее помимо сущности «Поставщики» были созданы сущности «Юр_лица» и «Физ_лица».

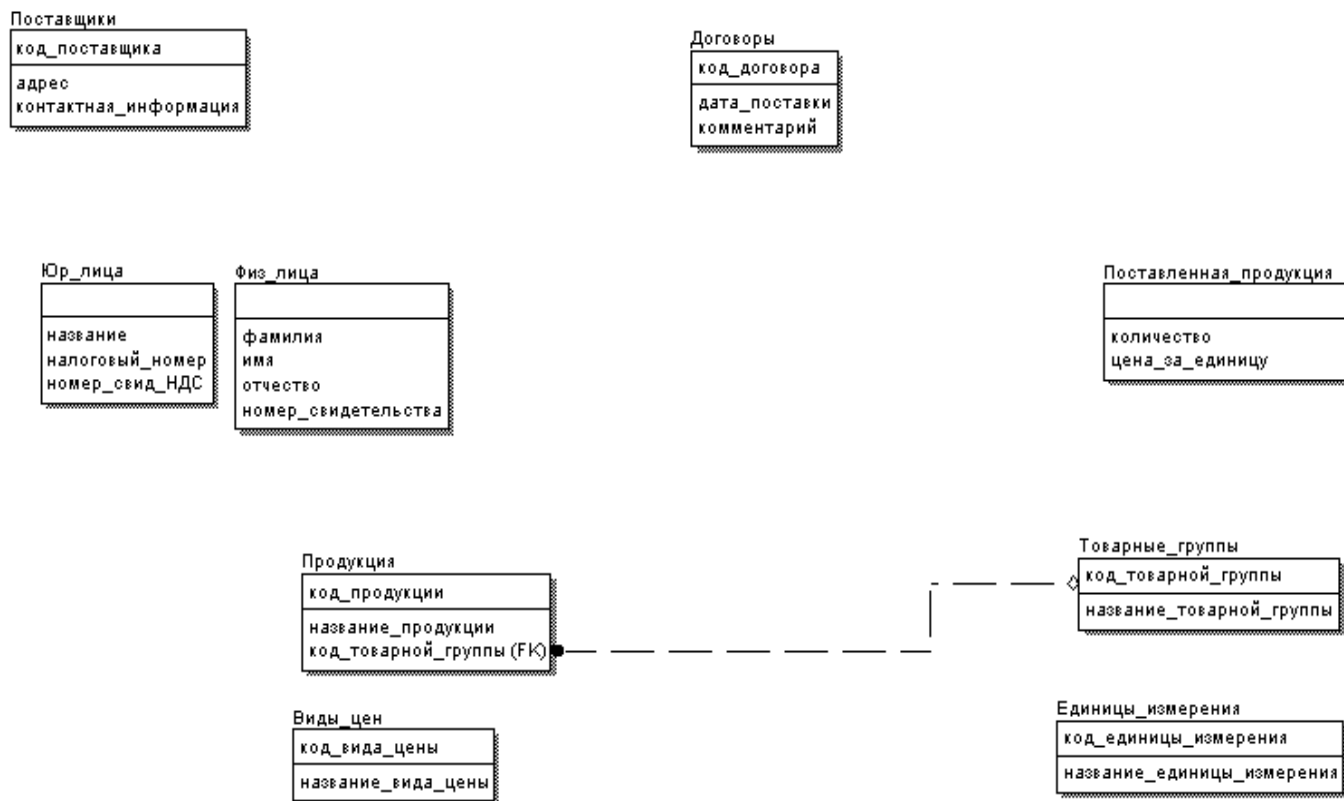


Рисунок 9

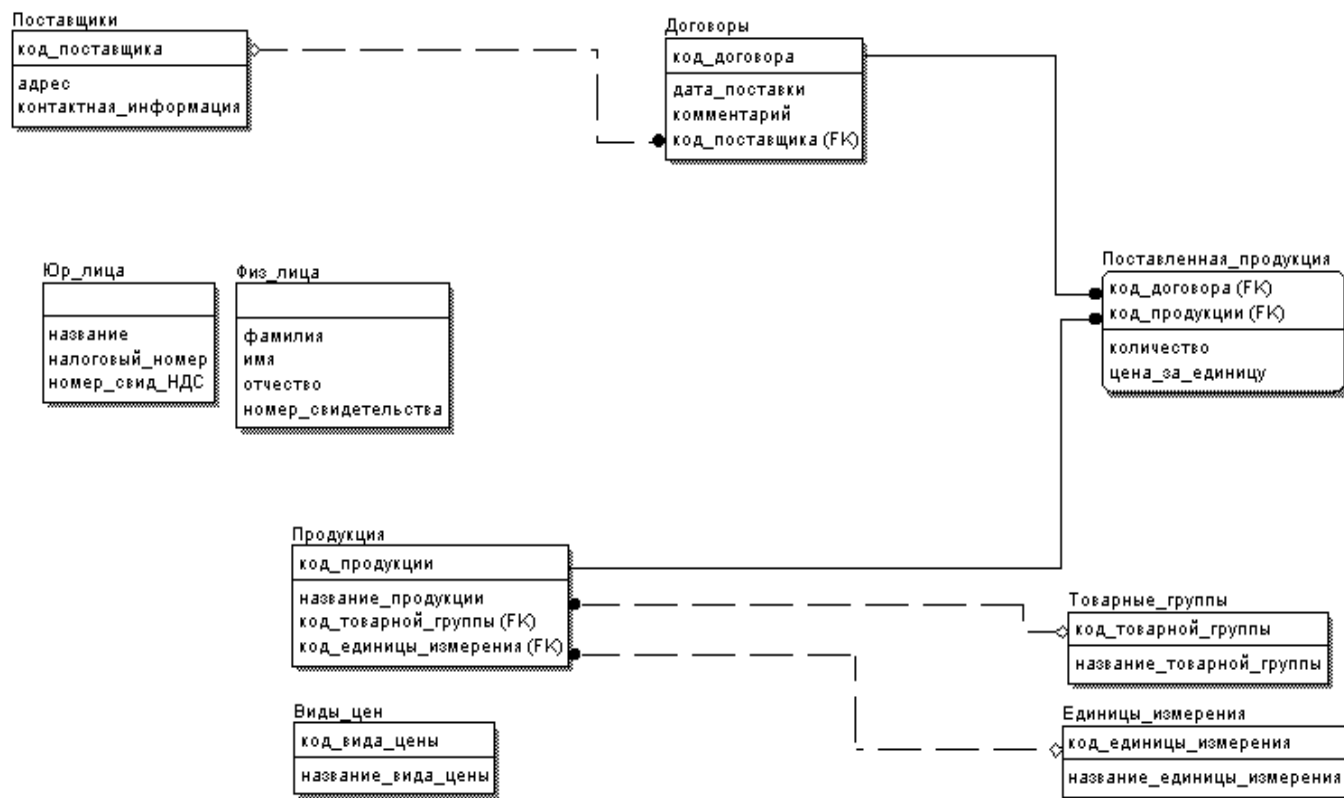


Рисунок 10

Для создания иерархия наследования, в которую войдут сущности «Поставщики», «Юр_лица» и «Физ_лица», необходимо:

- 1) щелкнуть мышью по кнопке «Complite sub-category» в панели инструментов;
- 2) щелкнуть мышью сначала по сущности «Поставщики», а затем по сущности «Юр_лица». В результате между сущностями появится категориальная связь (рис. 11). Атрибут, выполняющий функции первичного ключа (Primary Key) сущности «Поставщики» мигрирует и будет также выполнять функции первичного ключа сущности «Юр_лица»;
- 3) щелкнуть мышью по кнопке «Complite sub-category» в панели инструментов;
- 4) щелкнуть мышью сначала по категориальной связи (по символу категориальной связи), а затем по сущности «Физ_лица». В результате между сущностями появится категориальная связь (рис. 12). Атрибут, выполняющий функции первичного ключа (Primary Key) сущности «Поставщики» мигрирует и будет также выполнять функции первичного ключа сущности «Физ_лица».

6. Создание между сущностями связи типа «многие-ко-многим»

Связь типа «многие-ко-многим» будет использована для создания сущности, обеспечивающей хранение информации о состоянии рынка (ценовых предложениях поставщиков). Рассмотрим последовательность действий при создании связи типа «многие-ко-многим» между сущностями «Продукция», «Поставщики», «Типы цен»:

- 1) щелкнуть мышью по кнопке «Many-to-many relationship» в панели инструментов;
- 2) щелкнуть мышью сначала по сущности «Поставщики», а затем по сущности «Продукция». В результате между сущностями появится связь (рис. 13). В том случае, если связь отображается на диаграмме не так, как требуется, щелкните по связи мышью и, не отпуская левую кнопку мыши, перетяните связь в требуемое положение;
- 3) щелкнуть по созданной связи правой кнопкой мыши и в появившемся меню выбрать пункт «Create Association Entity», в окне «Many-to-Many Relationship Transform Wizard» нажать кнопку «Далее»;
- 4) в поле «Entity Name» вместо «Поставщик_Продукция» ввести «Цены_на_рынке» (без кавычек), затем нажать кнопку «Далее», затем – еще раз «Далее», затем «Готово». В результате появится новая сущность «Цены_на_рынке», обеспечивающая связь «многие-ко-многим» между сущностями «Поставщики» и «Продукция» (рис. 14);
- 5) наличие связи «многие-ко-многим» показывает, что каждый поставщик может предлагать на рынке различную продукцию, и, наоборот, каждая продукция может поставляться разными поставщиками. Однако наличие такой связи не дает возможности указать, по какой цене продукция предлагается поставщиками. В связи с этим для сущности «Цены_на_рынке» необходимо добавить еще один атрибут – «Цена». Для этого необходимо щелкнуть по сущности правой кнопкой мыши, выбрать в меню пункт «Attributes...» и добавить еще один атрибут – «Цена» (без кавычек). Для данного атрибута выбрать «Domain» – «Number»;

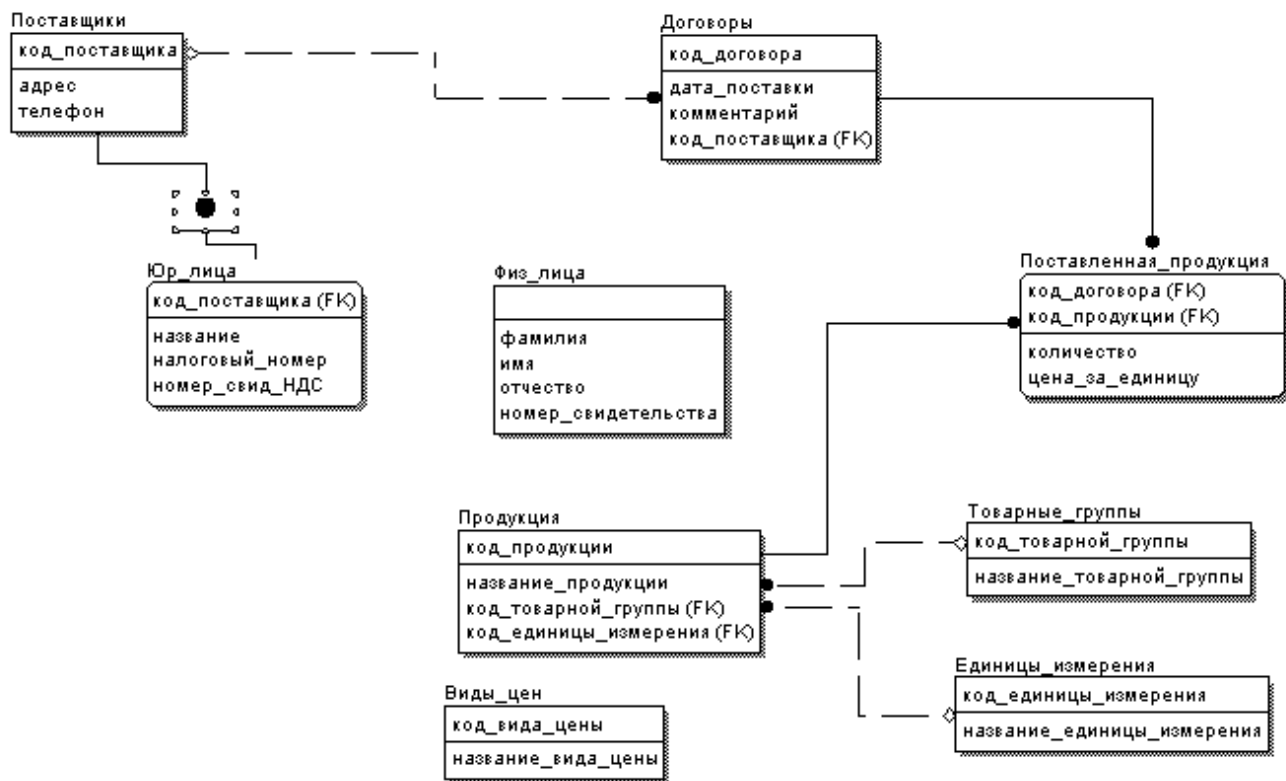


Рисунок 11

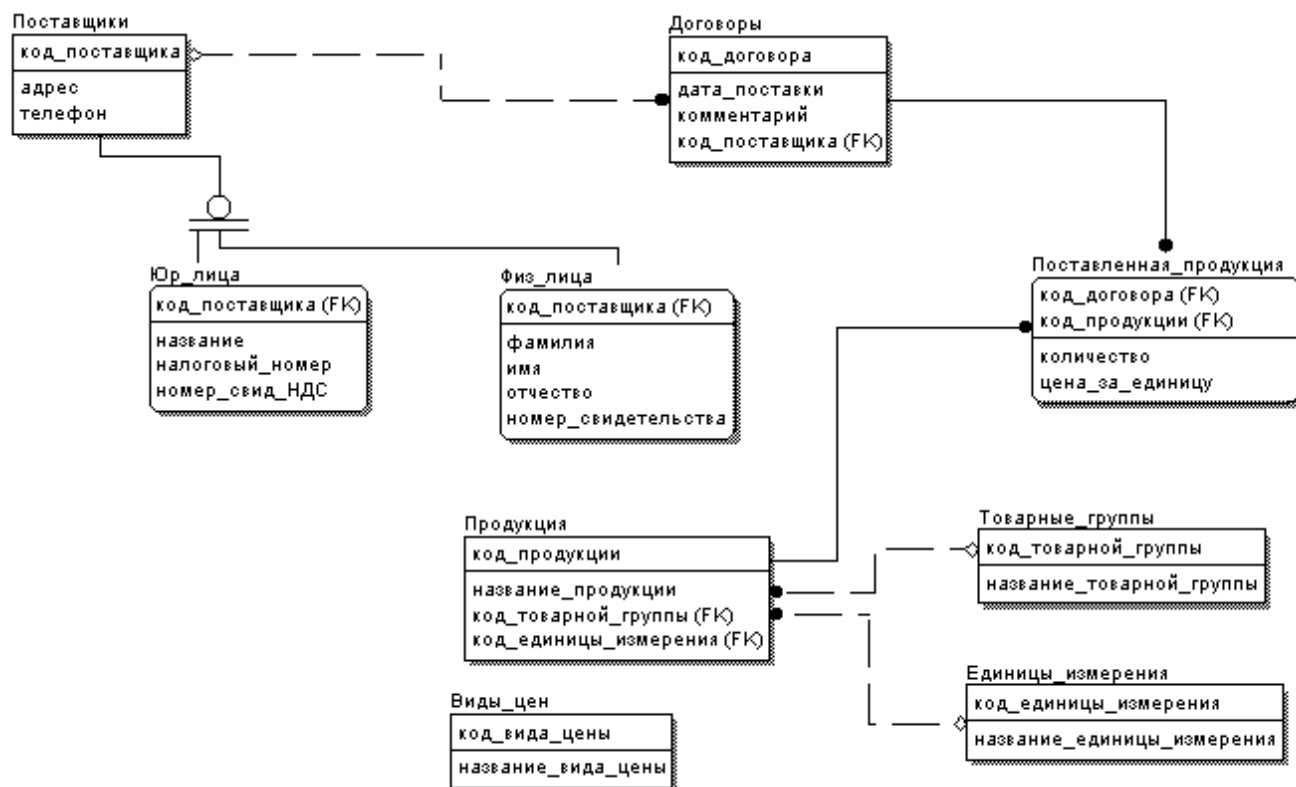


Рисунок 12

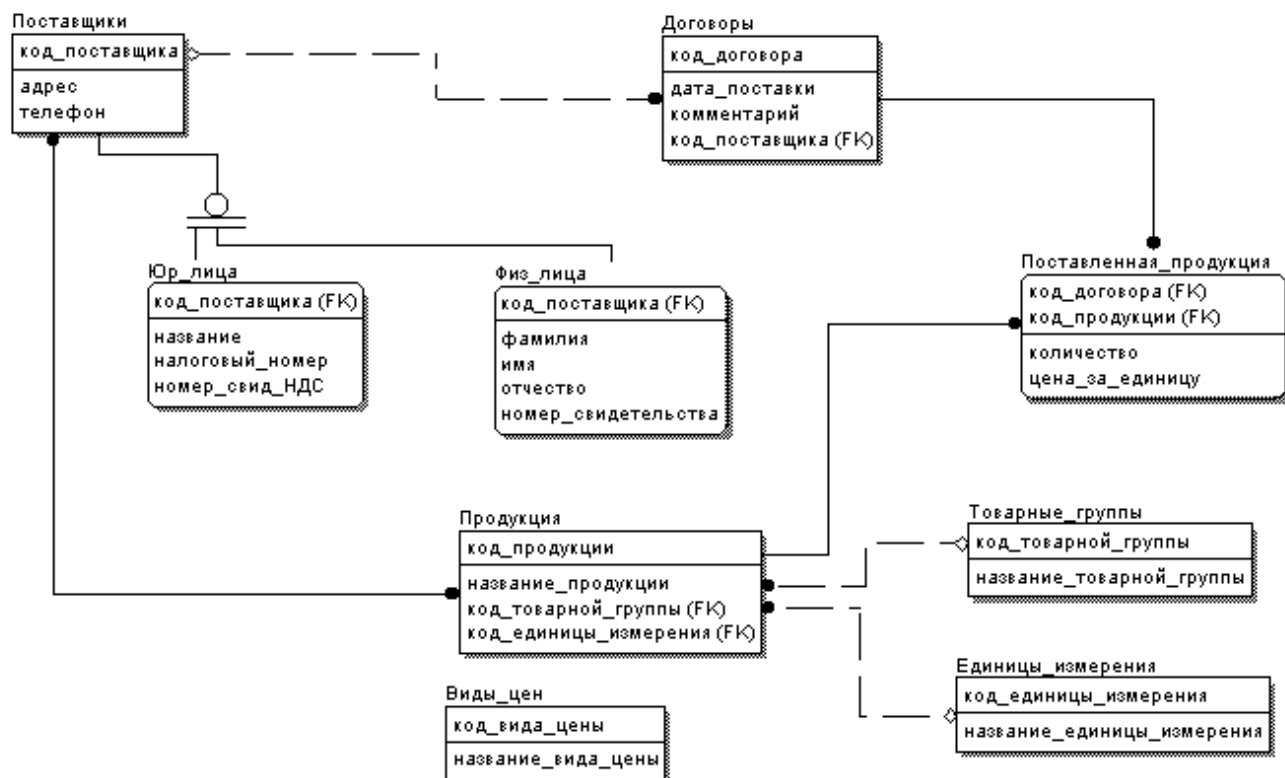


Рисунок 13

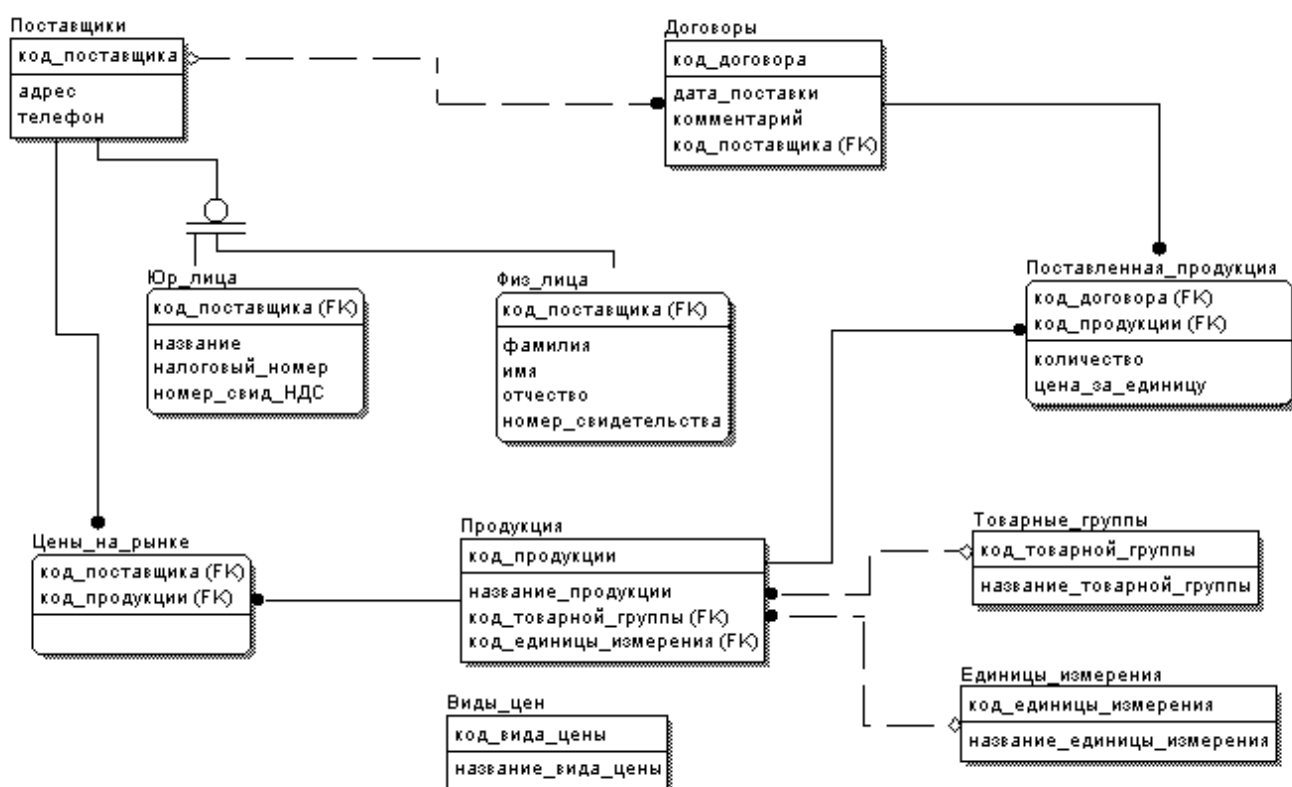


Рисунок 14

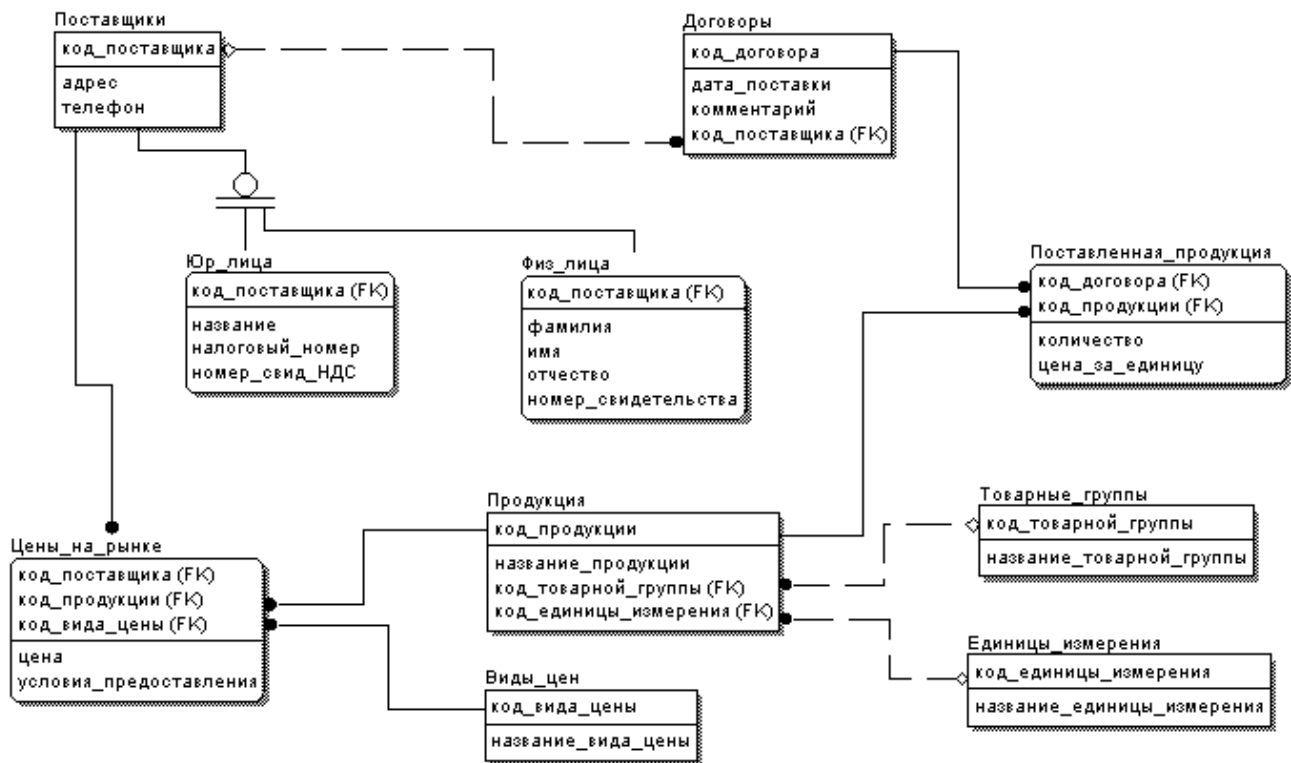


Рисунок 15

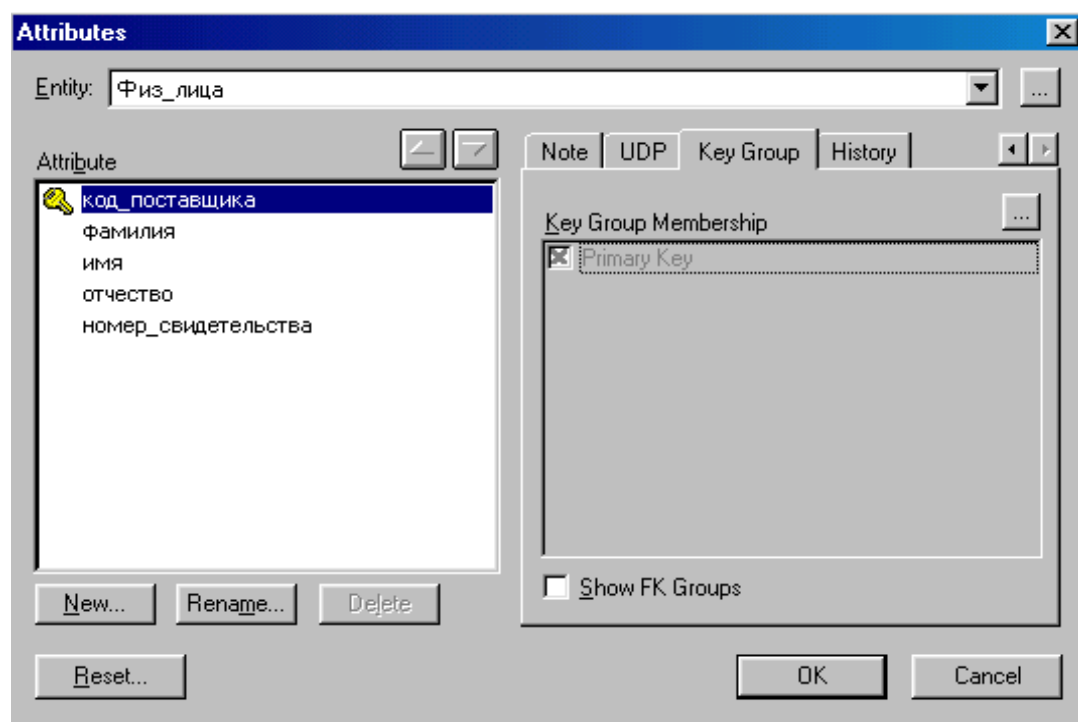


Рисунок 16

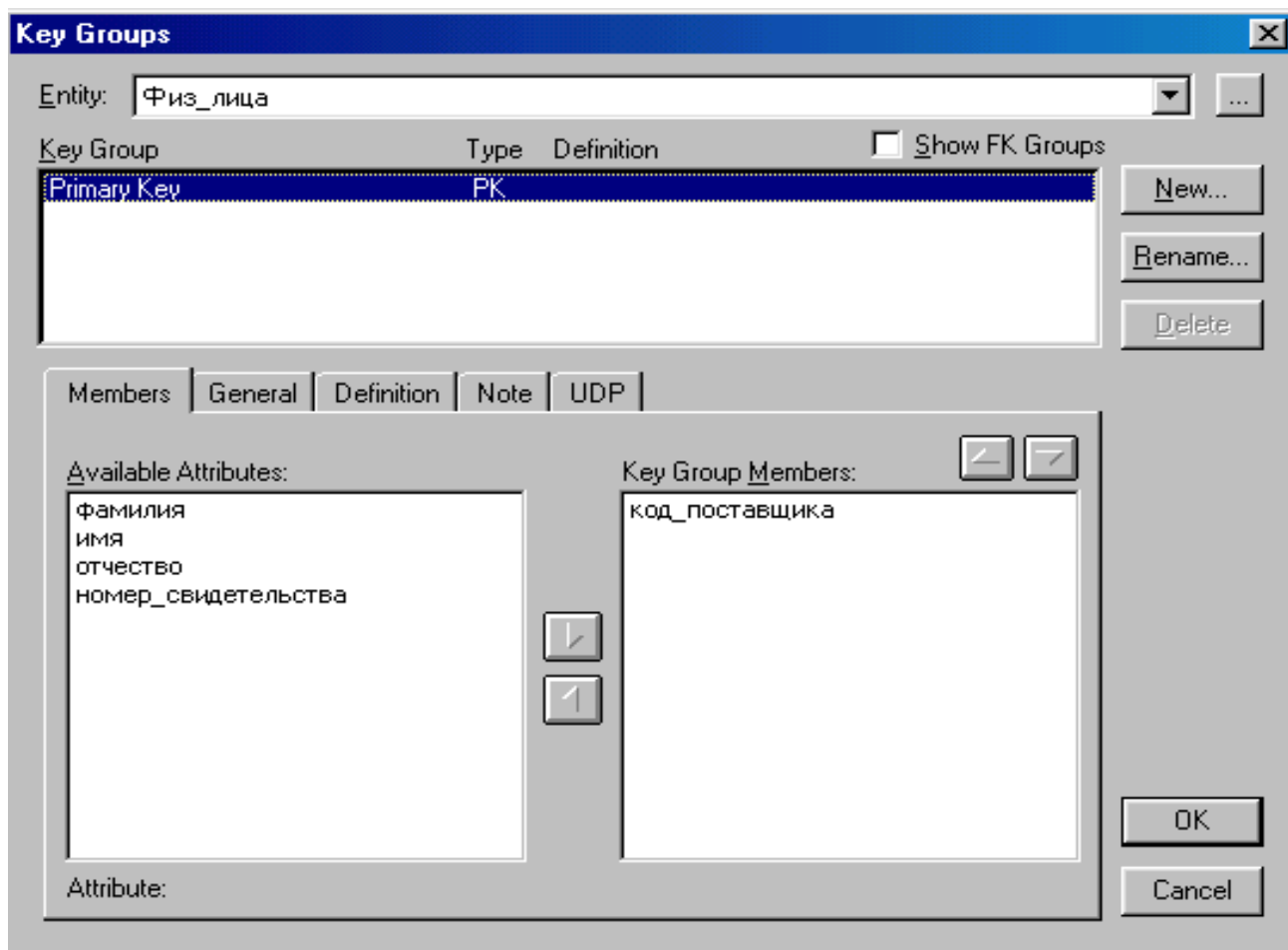


Рисунок 17

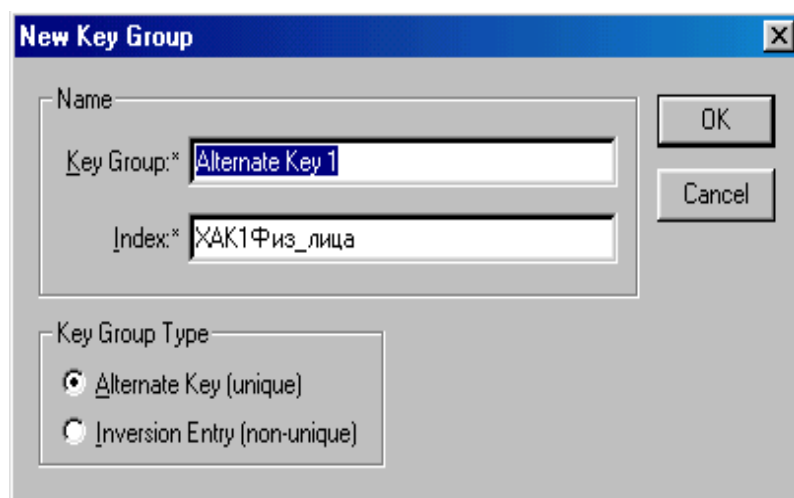


Рисунок 18

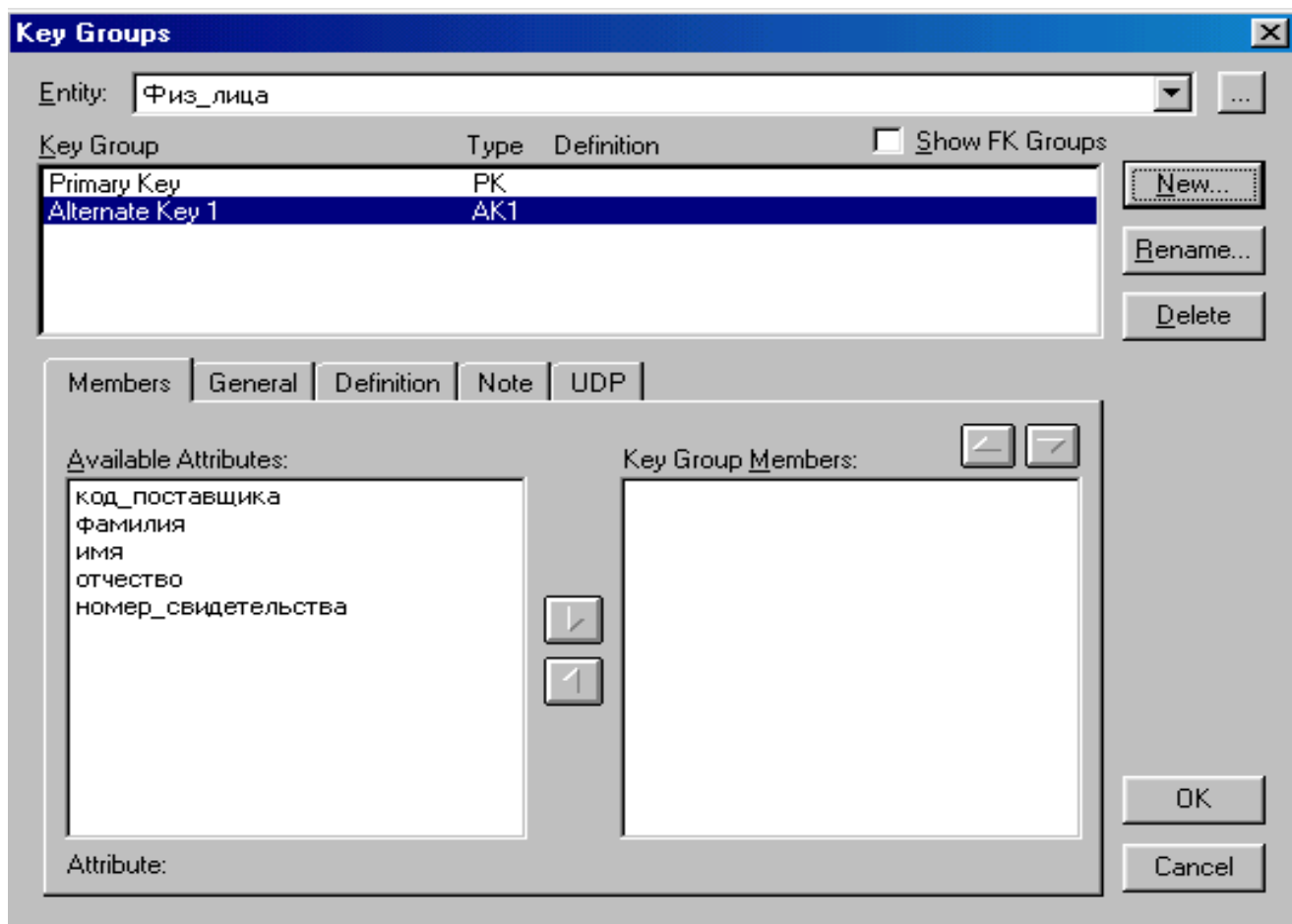


Рисунок 19

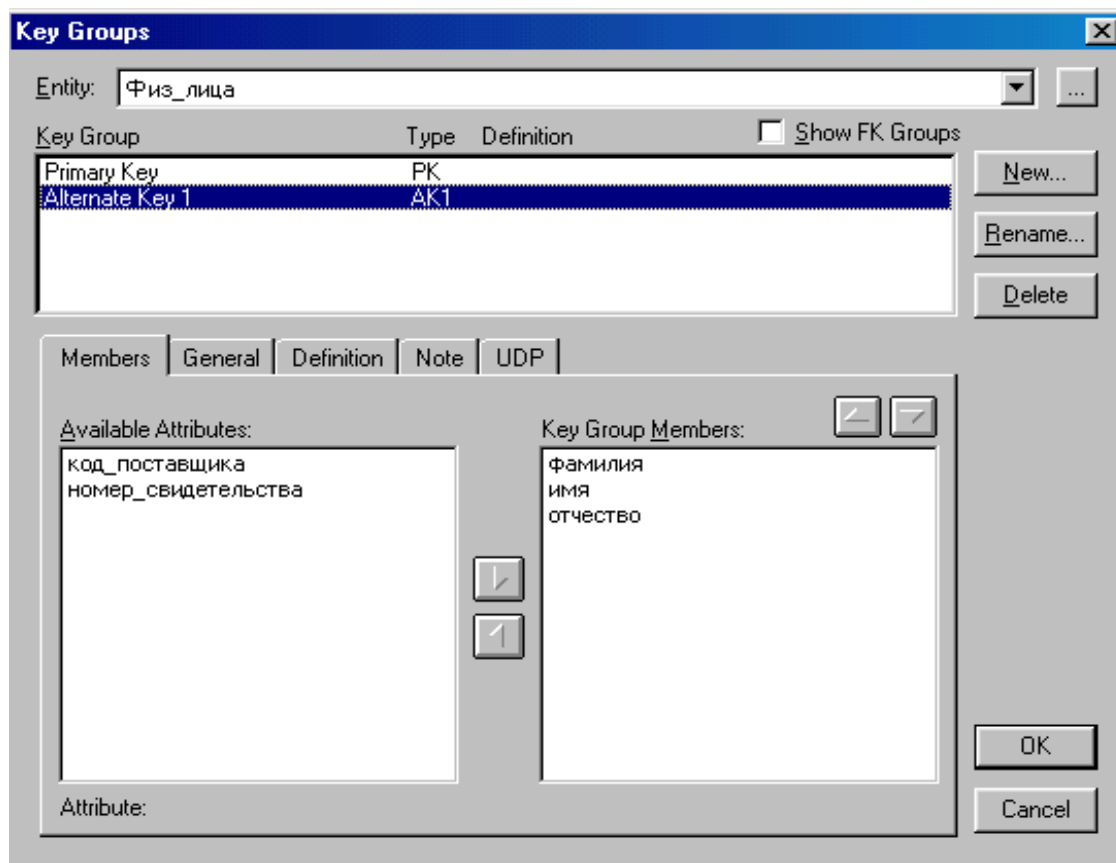


Рисунок 20

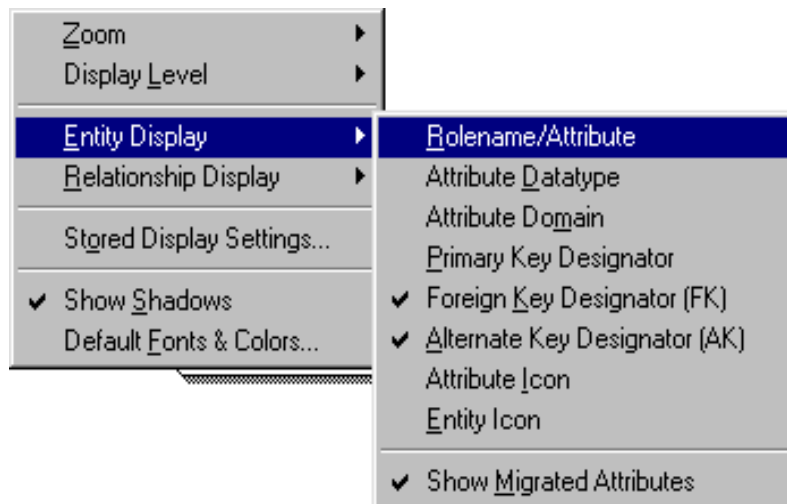


Рисунок 21

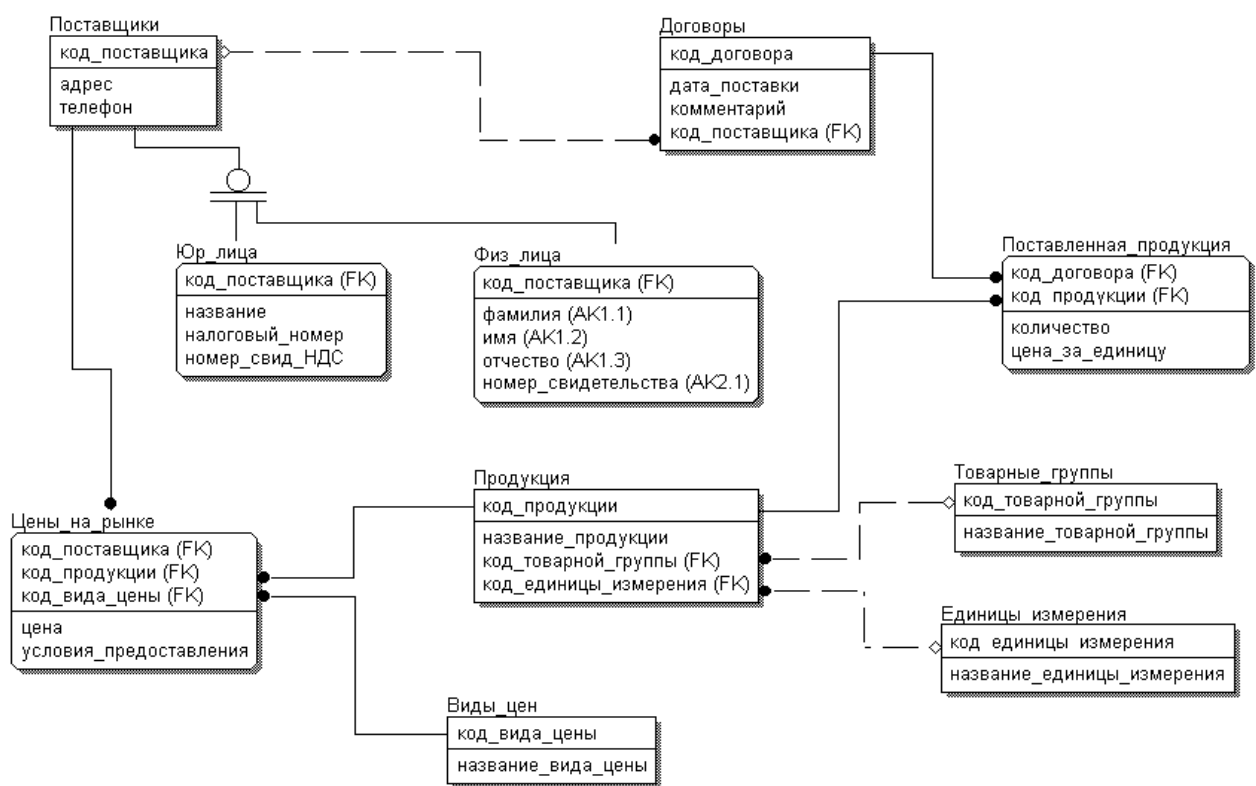


Рисунок 22

- 6) наличие в сущности «Цены_на_рынке» атрибута «Цена» дает возможность указывать, по какой цене предлагает какую-либо продукцию какой-либо поставщик. Однако при этом каждый поставщик может предлагать определенную продукцию только по одной цене. Для того, чтобы обеспечить возможность хранения данных о разных ценах на продукцию, предлагаемую одним поставщиком, установим связь между сущностями «Цены_на_рынке» и «Виды_цен». Для этого необходимо щелкнуть мышью по кнопке «Identifying relationship» в панели инструментов, затем щелкнуть мышью по сущности «Виды_цен», а затем – по сущности «Цены_на_рынке». В результате будет

установлена связь, первичный ключ сущности «Виды_цен» мигрирует и войдет в состав первичного ключа сущности «Цены_на_рынке» (рис. 15);

- 7) наличие в сущности «Цены_на_рынке» атрибута «Вид_цены» дает возможность указывать, по каким ценам (с указанием вида и значения цены) предлагает какую-либо продукцию какой-либо поставщик. Для того, чтобы обеспечить потребителей информацией об условиях продажи продукции по какому-либо виду цены, необходимо добавить еще один атрибут – «Условия_предоставления» Для данного атрибута выбрать «Domain» – «String» (рис. 15).

7. Создание альтернативных ключей.

Достаточно часто при построении модели данных возникает необходимость определения в сущности помимо первичного ключа еще одного или нескольких альтернативных ключей. Как правило, альтернативными становятся потенциальные ключи, не выбранные в качестве первичного. Наличие альтернативных ключей позволяет, в частности, контролировать уникальность значений атрибутов, не входящих в состав первичного ключа. Предположим, что для поставщиков–физических лиц возникла необходимость контролировать уникальность фамилии, имени, отчества поставщика, а также номера свидетельства о регистрации. Для этого необходимо создать два альтернативных ключа. Рассмотрим последовательность действий при создании таких альтернативных ключей:

- 1) щелкнуть правой кнопкой мыши по сущности «Физ_лица» и в появившемся меню выбрать пункт «Attributes...»;
- 2) в окне «Attributes» открыть вкладку «Key Group» (рис. 16) и нажать кнопку «...». В результате на экране появится окно «Key Groups» (рис. 17), в котором будут отображены данные о ключах данной сущности;
- 3) нажать кнопку «New». На экране появится окно «New Key Group» (рис. 18), с помощью которого определяются характеристики нового ключа. В данном случае никаких изменений делать не нужно. Нажатие кнопки «ОК» закроет это окно, в результате чего будет создан новый альтернативный ключ (рис. 19);
- 4) теперь необходимо сформировать список ключевых атрибутов. Для этого необходимо атрибуты «Фамилия», «Имя», «Отчество» переместить из списка «Available Attributes» в список «Key Group Members» (рис. 20);
- 5) теперь создадим второй альтернативный ключ. Последовательность действий аналогична предыдущей. В состав ключевых атрибутов нужно включить только атрибут «номер_свидетельства»;
- 6) окно «Key Groups» закрыть, нажав кнопку «ОК». На вкладке «Key Group» можно увидеть измененный список ключей данной сущности;
- 7) окне «Attributes» закрыть, нажав кнопку «ОК»;
- 8) в том случае, если в модели не отображаются данные о наличии в сущности альтернативных ключей, нужно щелкнуть правой кнопкой мыши по любому свободному месту модели (т.е. не по сущности и не по связи). В появившемся меню выбрать пункт «Entity Display», а в подменю – пункт «Alternate Key Designator (AK)» (рис. 21). В результате станет видна принадлежность атрибутов сущности к альтернативным ключам (рис. 22).

8. Корректировка свойств связей

В процессе разработки модели данных могут возникнуть ситуации, требующие корректировки свойств ранее созданных связей. Это, в частности, может быть связано с тем, что свойства связей, устанавливаемые по умолчанию, не соответствуют требованиям, предъявляемым к проектируемой базе данных. Рассмотрим пример корректировки свойств связи используя связь между сущностями «Продукция» и «Товарные_группы». Эта связь показывает, что любая продукция относится к определенной товарной группе. Однако особенностью этой связи является то, что какая-то продукция может не относиться ни к какой товарной группе. Это утверждение является следствием того, что значение внешнего ключа «Код_товарной_группы» в сущности «Продукция» может быть пустым (Null). Такое свойство связи устанавливается по умолчанию, но оно противоречит требованиям, сформулированным в результате анализа предметной области, и, следовательно, это свойство необходимо изменить. Рассмотрим последовательность действий при изменении свойств связи:

- 1) двойным щелчком мыши по связи открыть окно «Relationships», в котором отображаются свойства связи (рис. 23);
- 2) установить переключатель «Nulls» в положение «No Nulls»;
- 3) закрыть окно «Relationships», нажав кнопку «ОК»;
- 4) проанализировать, каким образом изменилось изображение связи в модели;
- 5) изменить аналогичные свойства всех остальных неидентифицирующих связей модели;
- 6) сохранить все изменения, сделанные в модели.

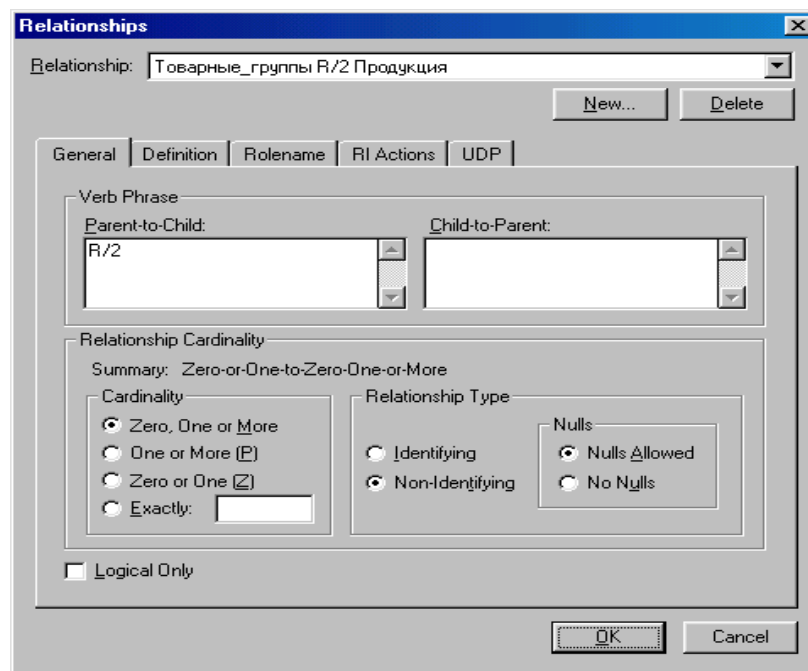


Рисунок 23

9. Корректировка категориальных связей в иерархии наследования

Для категориальных связей в процессе разработки модели данных также может потребоваться корректировка. Рассмотрим несколько примеров, связанных с корректировкой свойств категориальной связи или ее преобразованием.

Внимание! Изменения, которые будут сделаны в модели, сохранять не нужно!

Пример 1. Изменение типа иерархии категорий.

Иерархии категорий делятся на два типа – полные и неполные. Реализованная ранее категориальная связь является примером полной категории. Для того, чтобы изменить тип категории, нужно выполнить следующую последовательность действий:

- 1) щелкнуть правой кнопкой мыши по значку категориальной связи и в появившемся меню выбрать пункт «Subtype Relationship...», в котором отображаются свойства связи;
- 2) в окне «Subtype Relationship» установить переключатель «Subtype Type» в режим «Incomplite»;
- 3) нажать кнопку «ОК» и затем посмотреть, как изменится значок категориальной связи после изменения типа категории на неполную;
- 4) восстановить полный тип категории.

Пример 2. Замена иерархии наследования идентифицирующими связями.

Такая замена может быть обусловлена желанием упростить модель или какими-то другими соображениями проектировщика. Для того, чтобы заменить иерархию наследования идентифицирующими связями, нужно выполнить следующую последовательность действий:

- 1) щелкнуть мышью по значку категориальной связи и в панели инструментов щелкнуть по кнопке «Supertype-Subtype Identity»;
- 2) в окне «Supertype-Subtype Identity Wizard» нажать кнопку «Далее», затем опять нажать кнопку «Далее», затем нажать кнопку «Готово». Фрагмент модели, иллюстрирующий результат преобразования, приведен на рисунке 24. Как видно на рисунке, произошла замена категориальной связи на две идентифицирующие связи;
- 3) закрыть модель, не сохраняя сделанные изменения.

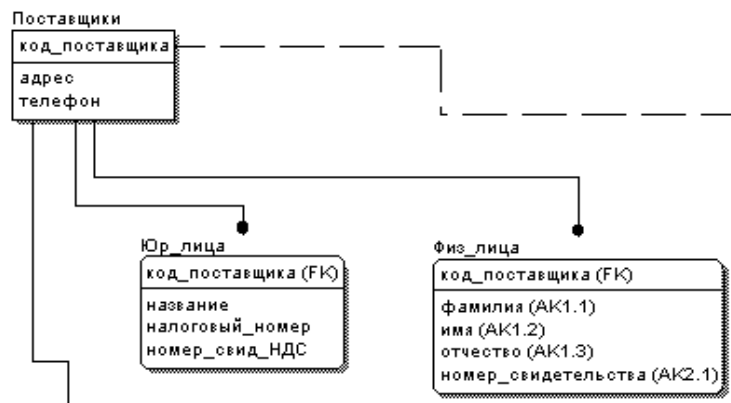


Рисунок 24

Пример 3. Миграция первичного ключа и неключевых атрибутов в иерархии наследования от потомков к предку.

Такая замена может быть обусловлена желанием упростить модель или какими-то другими соображениями проектировщика. Для того, чтобы выполнить миграцию первичного ключа и неключевых атрибутов в иерархии наследования от потомков к предку, нужно выполнить следующую последовательность действий:

- 1) открыть модель;
- 2) щелкнуть мышью по значку категориальной связи и в панели инструментов щелкнуть по кнопке «Supertype-Subtype Rollup»;
- 3) в окне «Supertype-Subtype Roll-Up Wizard» нажать кнопку «Далее», затем опять нажать кнопку «Далее», затем нажать кнопку «Готово». Фрагмент модели, иллюстрирующий результат преобразования, приведен на рисунке 25. Как видно на рисунке, в результате преобразования вместо трех сущностей «Поставщики», «Юр_лица» и «Физ_лица», в модели осталась только сущность «Поставщики», которая в состав атрибутов которой включены атрибуты сущностей «Юр_лица» и «Физ_лица». Структура такой сущности имеет очевидные недостатки с точки зрения требований нормализации;
- 4) закрыть модель, не сохраняя сделанные изменения.

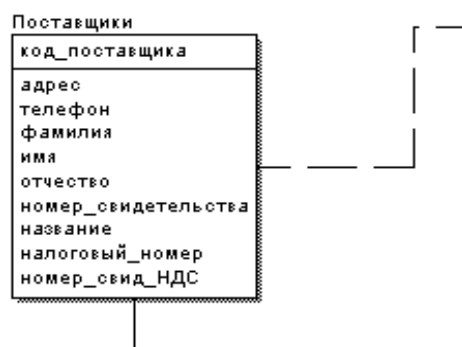


Рисунок 25

Пример 4. Миграция первичного ключа и неключевых атрибутов в иерархии наследования от предка к потомкам.

Такая замена может быть обусловлена желанием упростить модель или какими-то другими соображениями проектировщика. Для того, чтобы выполнить миграцию первичного ключа и неключевых атрибутов в иерархии наследования от предка к потомкам, нужно выполнить следующую последовательность действий:

- 1) открыть модель;
- 2) щелкнуть мышью по значку категориальной связи и в панели инструментов щелкнуть по кнопке «Supertype-Subtype Roll-down»;
- 3) в окне «Supertype-Subtype Roll-Down Wizard» нажать кнопку «Далее», затем опять нажать кнопку «Далее», затем нажать кнопку «Готово». Фрагмент модели, иллюстрирующий результат преобразования, приведен на рисунке 26. Как видно на рисунке, в результате преобразования вместо трех сущностей «Поставщики», «Юр_лица» и «Физ_лица», в модели остались только сущности «Юр_лица» и «Физ_лица», которая в состав атрибутов которых

включены атрибуты сущности «Поставщики». Структура таких сущностей имеет очевидные недостатки с точки зрения требований нормализации;

- 4) закрыть модель, не сохраняя сделанные изменения;
- 5) открыть модель снова.

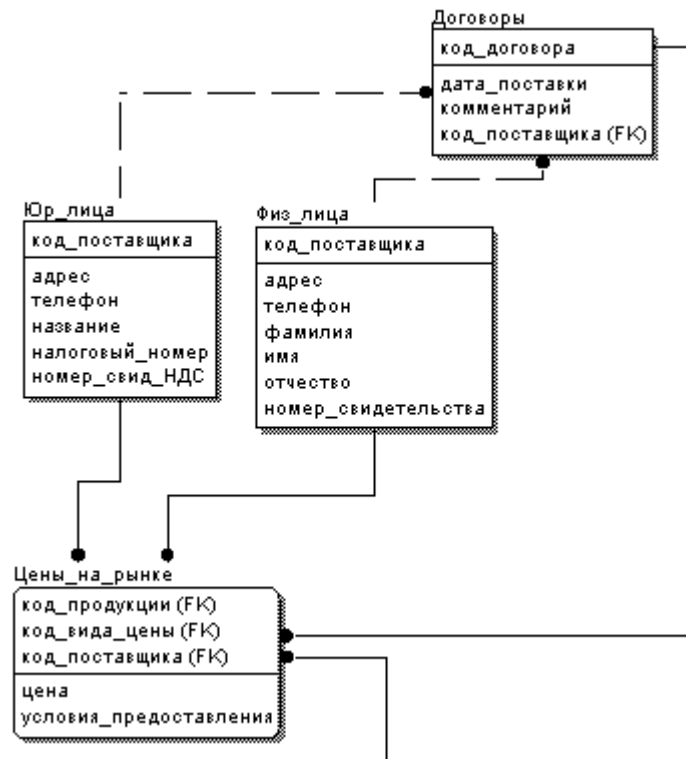


Рисунок 26

10. Переход к физической модели данных

Логическая модель данных может быть создана без определения СУБД, под управлением которой будет функционировать проектируемая база данных. Однако реальная база данных может быть создана с учетом особенностей конкретной СУБД. В данной работе СУБД была определена при создании модели. Создание физической модели данных, в которой учитываются особенности выбранной СУБД, происходит автоматически при построении логической модели. Для получения доступа к физической модели данных необходимо переключатель «Logical» перевести в положение «Physical». В результате на экране будет отображена физическая модель (рис. 27).

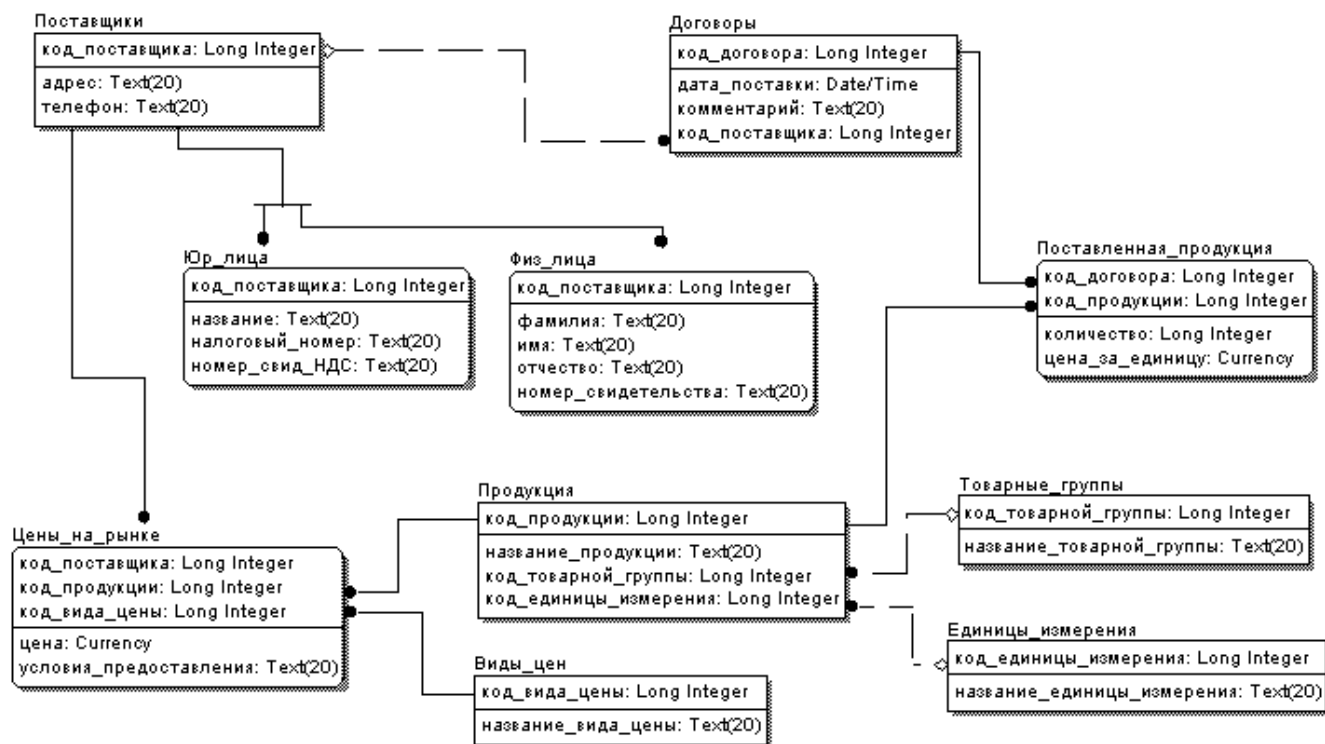


Рисунок 27

11. Сохранить созданную модель и закончить работу.

12. Требования к отчету

- 1) кратко описать основные этапы выполнения работы;
- 2) изобразить модели, разработанные в процессе выполнения работы (окончательный вид моделей);
- 3) описать назначение основных управляющих элементов интерфейса ERWin и особенности их использования при разработке модели данных.

Часть 2

Цель работы: Создание базы данных на основе модели IDEF1X.

Выполнение работы.

1. Открыть модель, созданную в результате выполнения 1 части работы и сохраненную в файле «delivery.er1».
2. Перейти к физической модели. Переход к физической модели требует изменения используемых при описании последовательности выполнения работы терминов. Так, в частности, далее вместо термина «сущность» будет использоваться термин «таблица», а вместо термина «атрибут» – «поле».
3. Изменить параметры полей в таблицах (атрибутов в сущностях)
Рассмотрим процедуру изменения параметров поля на примере поля «Цена_за_единицу» в таблице «Поставленная_продукция»:
 - 1) щелкнуть правой кнопкой мыши по таблице «Поставленная_продукция» и в появившемся меню выбрать пункт «Columns...»;
 - 2) в списке полей выбрать поле «Цена_за_единицу» и установить на вкладке «Access» тип данных «Currency» (рис. 28);
 - 3) нажать кнопку «ОК».Аналогично изменить тип данных на «Currency» для поля «Цена» таблицы «Цены_на_рынке».
4. Проверить установку целевой СУБД.
Для этого выбрать в меню пункт «Database» и в вертикальном меню выбрать пункт «Choose Database». В результате на экране появится окно, с помощью которого можно проверить и при необходимости изменить настройки, связанные с СУБД, в формате которой будет создаваться база данных на основании разрабатываемой модели (рис. 29)
5. Создание базы данных с помощью СУБД Access.
Для того, чтобы на основании модели данных была создана база данных, необходимо сначала создать новую базу данных. Такая база данных не должна содержать никаких объектов. Последовательность действий при создании базы данных может быть следующей:
 - 1) если есть необходимость (т.е. хочется чтобы эти файлы хранились отдельно), создать где-либо на диске отдельный каталог для размещения файлов базы данных. В данном случае в качестве такого каталога будем использовать D:\ER_LAB;
 - 2) запустить СУБД Access;
 - 3) выбрать режим создания новой базы данных и ввести ее имя – «поставки.mdb» (без кавычек) и место ее размещения (рис. 30)
 - 4) в результате будет создана новая база данных, не содержащая никаких объектов (рис. 31);
 - 5) закрыть окно базы данных и закрыть СУБД Access.

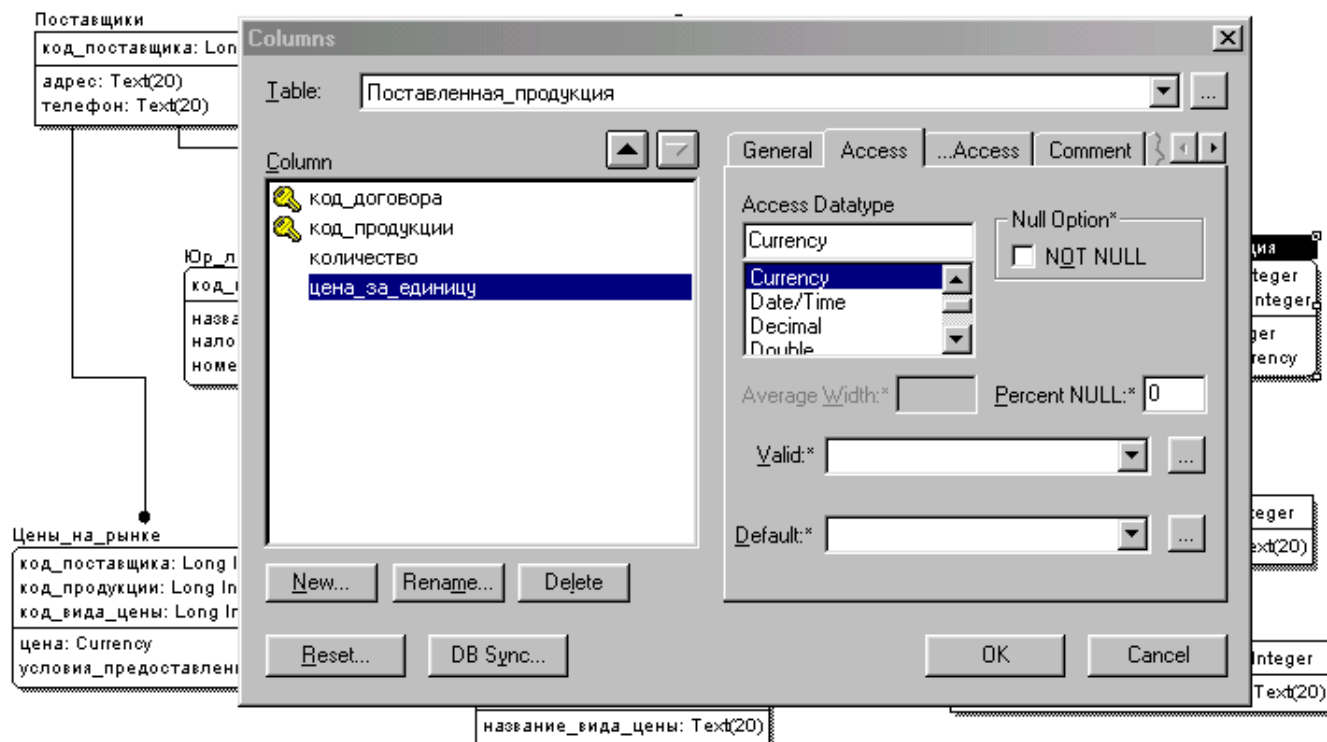


Рисунок 28

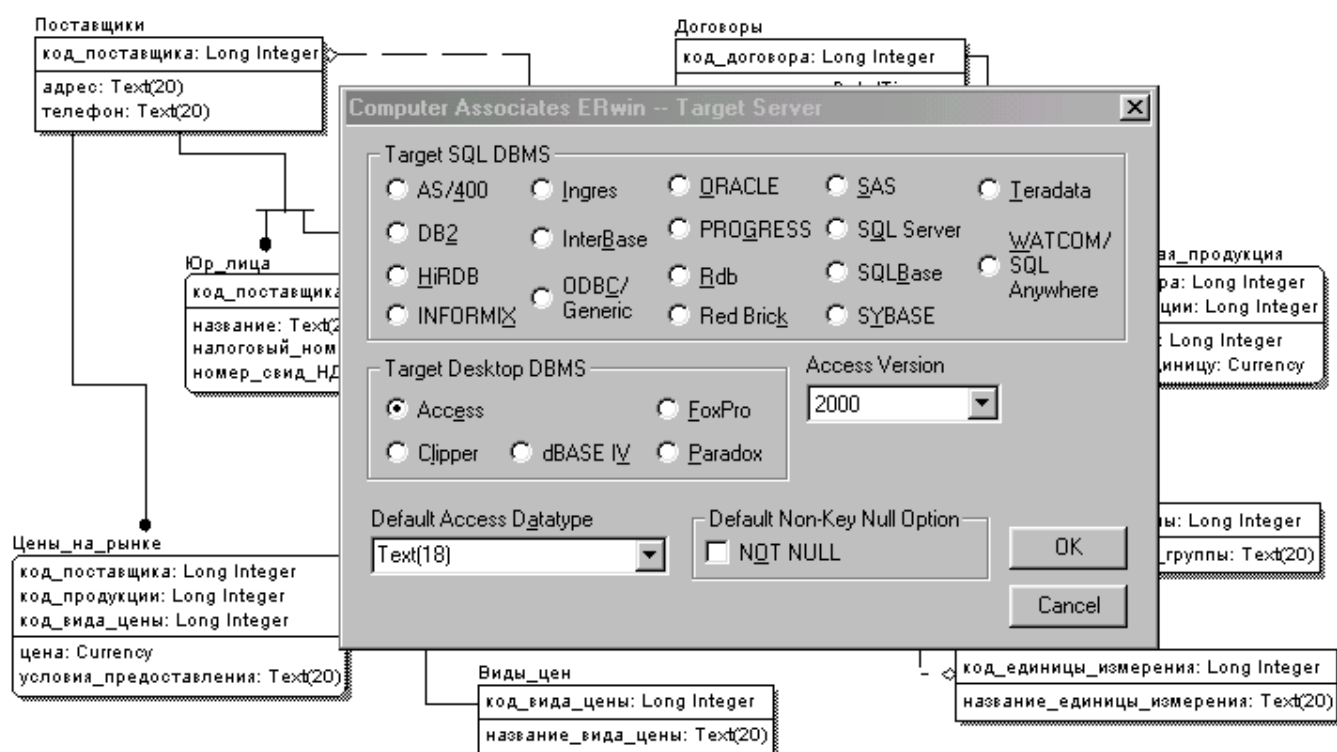


Рисунок 29

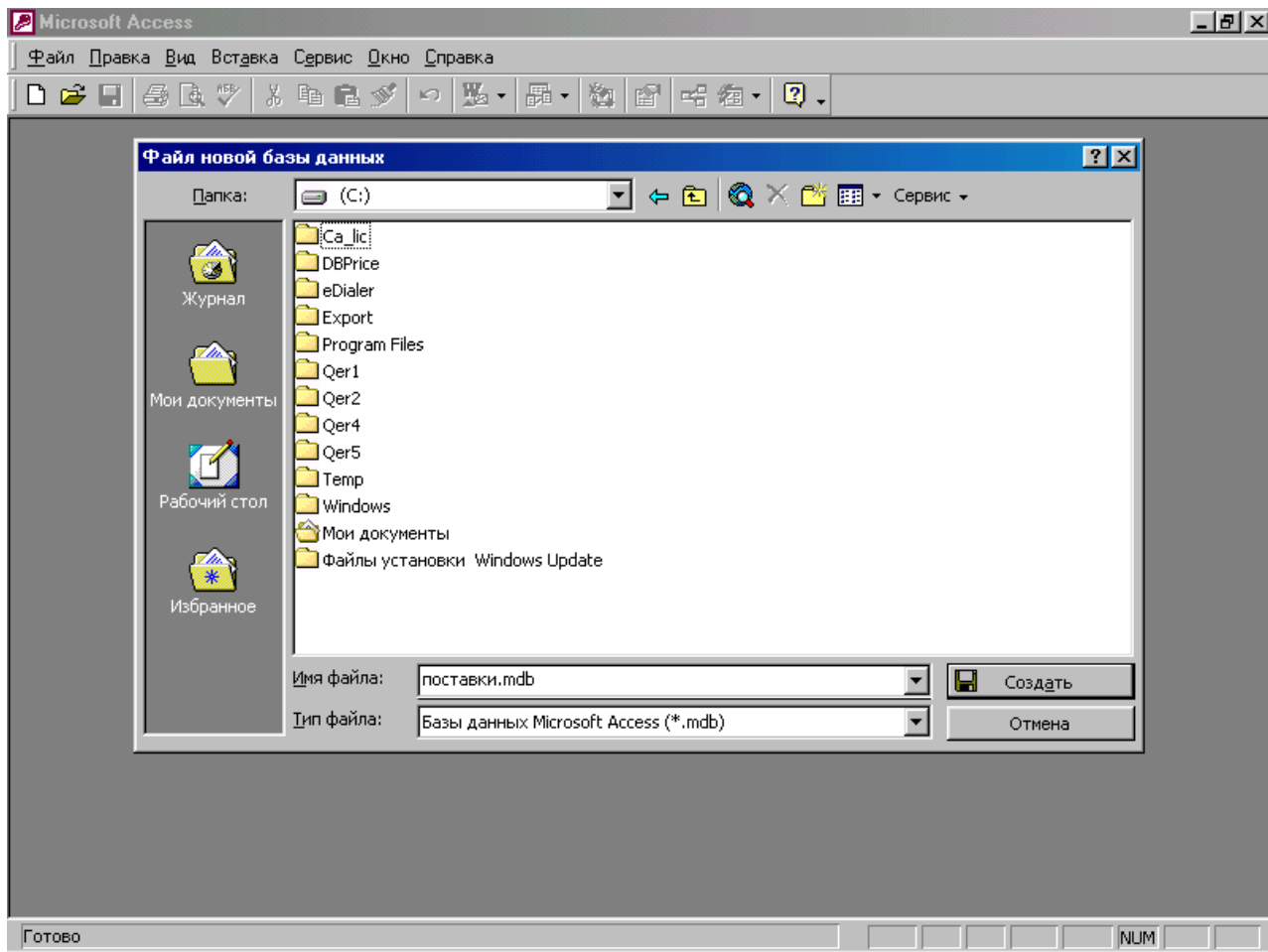


Рисунок 30

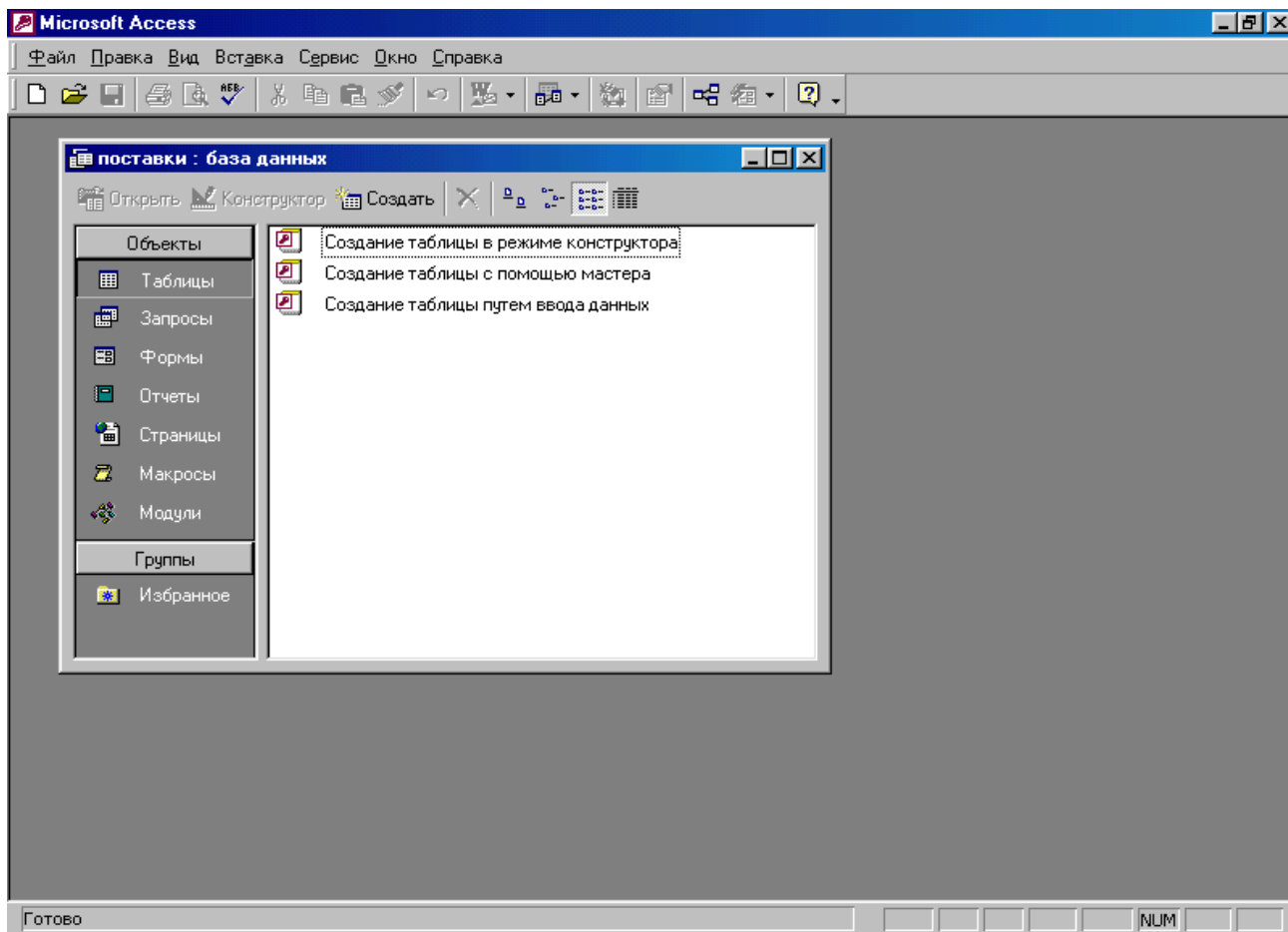


Рисунок 31

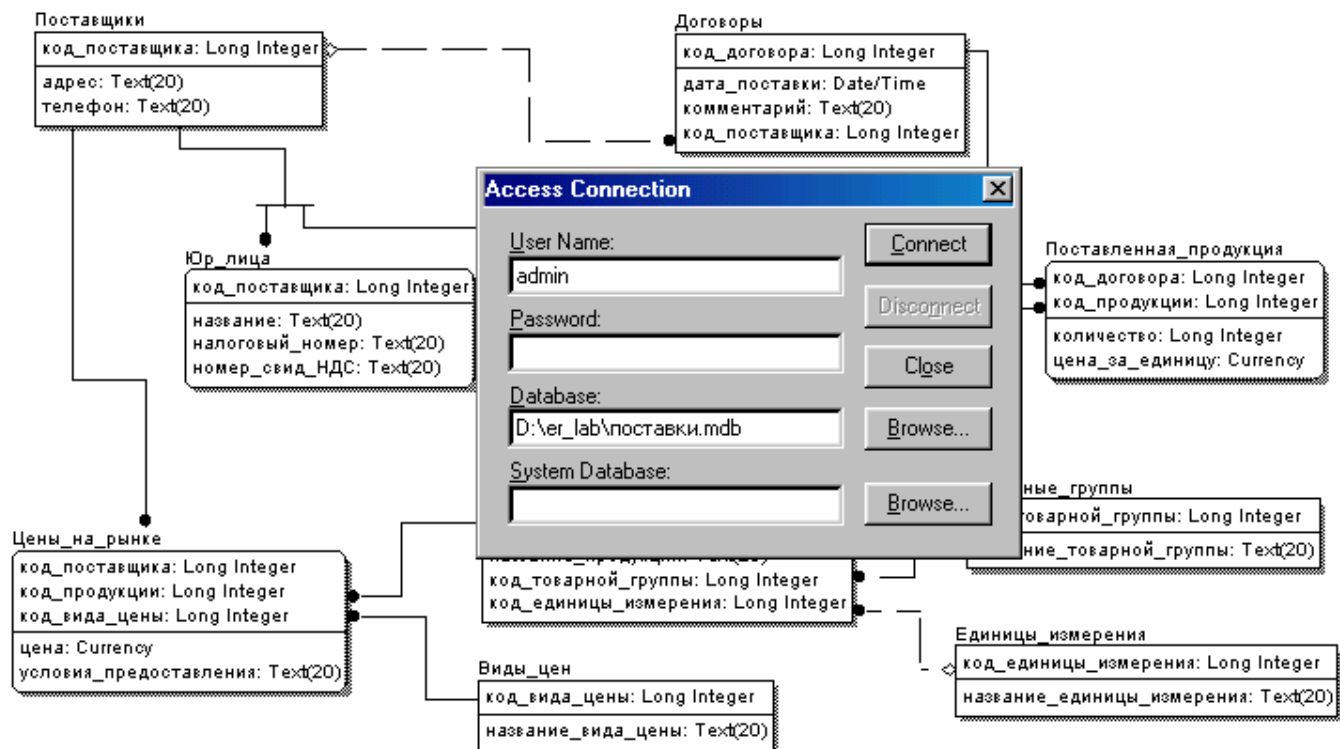


Рисунок 32

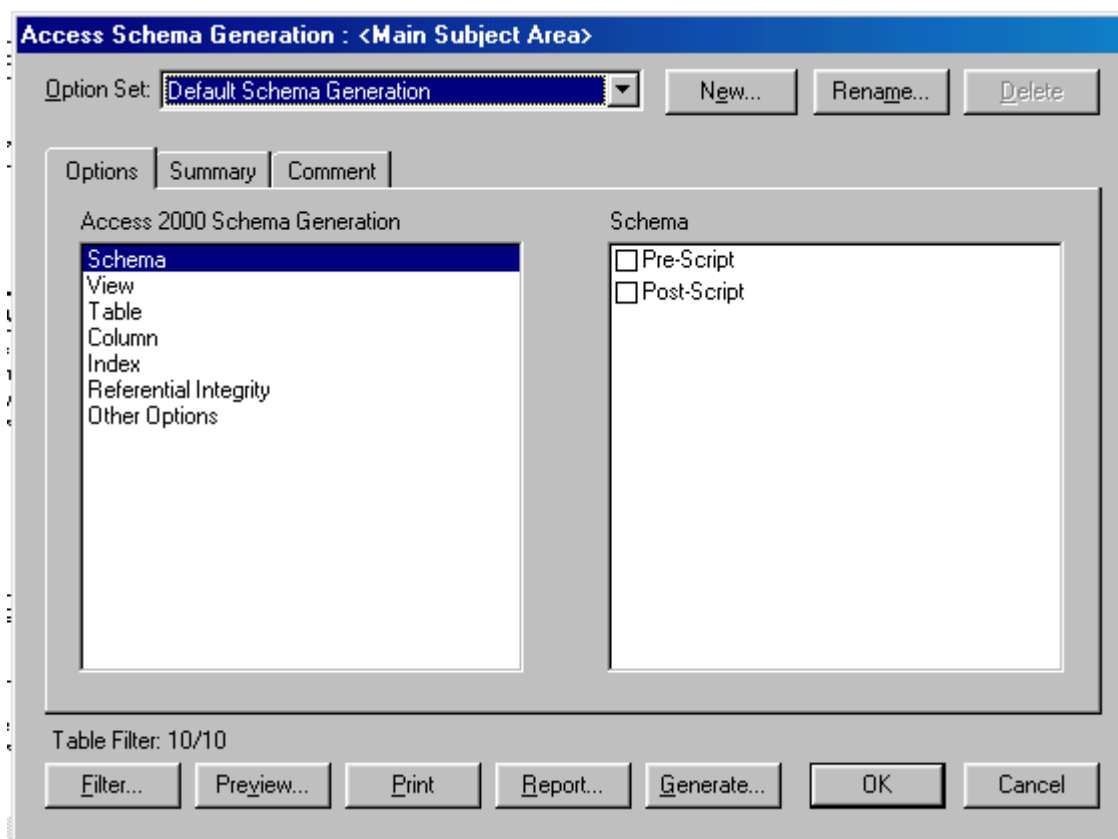


Рисунок 33

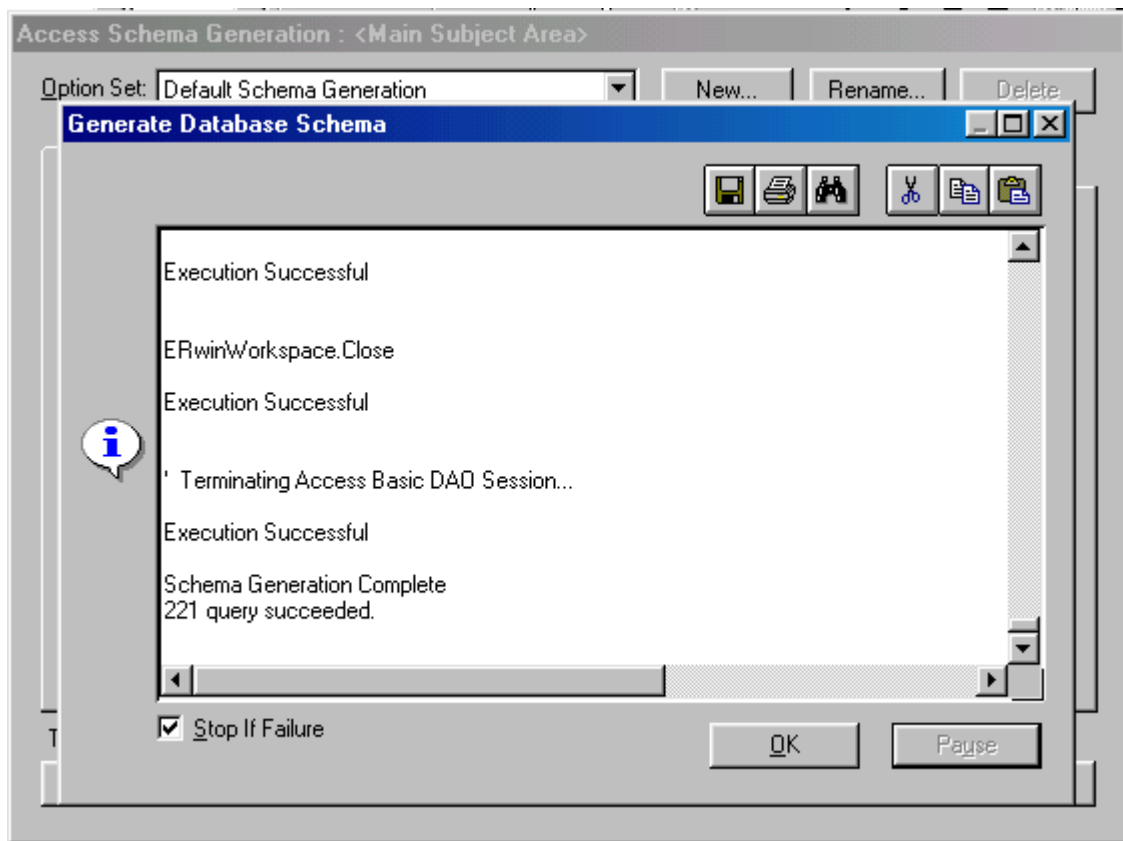


Рисунок 34

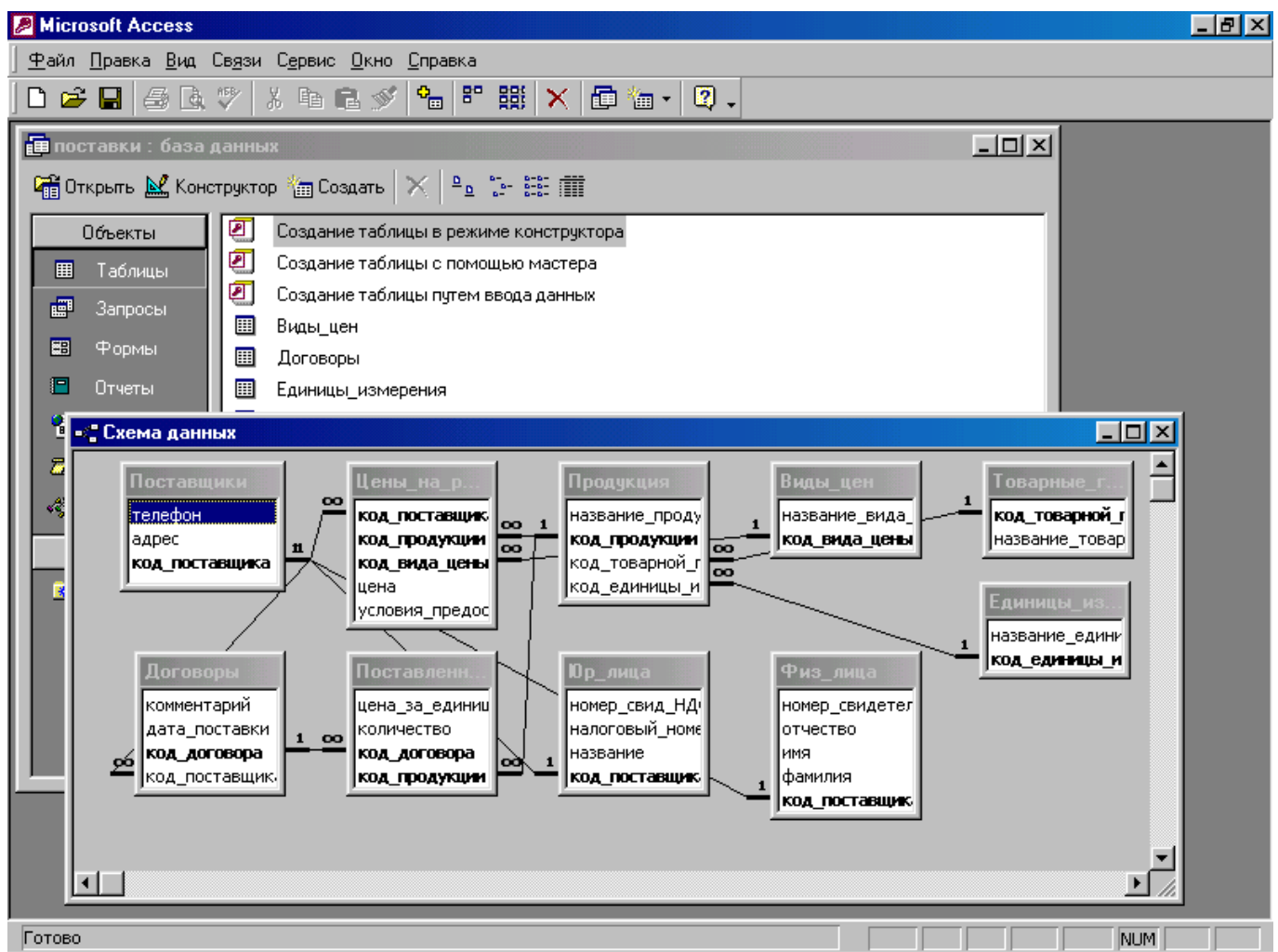


Рисунок 35

6. Подключение модели к созданной базе данных.

Последовательность действий при этом может быть следующей:

- 1) переключиться в режим работы с ERwin;
- 2) выбрать в меню пункт «Database» и в вертикальном меню выбрать пункт «Database Connection»;
- 3) в результате на экране появится окно «Access Connection». В поле «User Name» необходимо ввести «admin», в поле «Database» – «D:\er_lab\поставки.mdb». Все данные вводить без кавычек (рис. 32);
- 4) нажать кнопку «Connect».

7. Создание базы данных на основе разработанной модели.

Последовательность действий при этом может быть следующей:

- 1) выбрать в меню пункт «Tools» и в вертикальном меню выбрать пункт «Forward Engineer/Schema Generation...»;
- 2) в результате на экране появится окно «Access Schema Generation» (рис. 33);
- 3) нажать кнопку «Generate...»;
- 4) после завершения процесса нажать кнопку «OK» в окне «Generate Database Schema» (рис. 34), затем нажать кнопку «OK» в окне «Access Schema Generation»

8. Проверка результата создания базы данных.

Последовательность действий при этом может быть следующей:

- 1) запустить СУБД Access;
- 2) открыть базу данных «поставки.mdb» и визуально проверить наличие таблиц в базе данных и их соответствие таблицам, созданным при разработке модели;
- 3) открыть окно схемы данных (кнопка «Схема данных» в панели инструментов) и визуально проверить наличие связей между таблицами базы данных (рис. 35);
- 4) для таблицы «Физ_лица» проверить создание индексов, выполняющих функции альтернативных ключей. Для этого выполнить следующую последовательность действий:
 - в списке таблиц базы данных правой кнопкой мыши выбрать таблицу «Физ_лица» и в появившемся меню выбрать пункт «Конструктор». В результате на экране появится окно, отображающее структуру таблицы;
 - в главном меню СУБД Access выбрать пункт «Вид» и в вертикальном меню, соответствующем этому пункту, выбрать пункт «Индексы». В результате на экране появится окно, отображающее перечень индексов данной таблицы (рис. 36). Как видно, в этом перечне кроме индекса, выполняющего функции первичного ключа, присутствуют еще два индекса, выполняющие функции альтернативных ключей. Как и для первичного ключа, для альтернативных ключей выполняется требование уникальности;
 - закрыть окно, отображающее перечень индексов таблицы;
 - закрыть окно, отображающее структуру таблицы;
- 5) закрыть окно базы данных и закрыть СУБД Access.

Индексы: Физ_лица

Индекс	Имя поля	Порядок сортировки
PrimaryKey	код_поставщика	По возрастанию
ХАК1Физ_лица	фамилия	По возрастанию
	имя	По возрастанию
	отчество	По возрастанию
ХАК2Физ_лица	номер_свидетельства	По возрастанию

Свойства индекса

Ключевое поле	Да
Уникальный индекс	Да
Пропуск пустых полей	Нет

Имя данного индекса. Каждый индекс может содержать до 10 полей.

Рисунок 36

9. Сохранить созданную модель и закончить работу.

10. Требования к отчету

- 1) кратко описать основные этапы выполнения работы;
- 2) привести внешний вид созданной базы данных (например, схему данных), иллюстрирующий корректность создания базы данных;
- 3) в случае возникновения проблем в процессе создания базы данных на основе модели данных, описать возникшие проблемы и проанализировать причины их появления. Описать пути разрешения проблем.