

## ЛАБОРАТОРНАЯ РАБОТА

# СОЗДАНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ С ALLFUSION ERWIN DATA MODELER

### 1. Цель работы

Целью работы является изучение технологии построения на основе функциональной модели IDEF0 полной атрибутивной информационной модели с использованием пакета ERWin Data Modeler.

### 2. Задачи

Основными задачами лабораторной работы являются: закрепление и расширение теоретических знаний студентов, приобретение навыков информационного моделирования логического уровня, а также нормализации полученной модели.

### 3. Краткие теоретические сведения

#### 3.1. Понятие логической модели данных

Логическая модель позволяет понять суть проектируемой системы, отображая логические взаимосвязи между сущностями.

Различают 3 подуровня логического уровня модели данных, отличающиеся по глубине представления информации о данных:

- диаграмма сущность-связь (*Entity-Relationship Diagram (ERD)*);
- модель данных, основанная на ключах (*Key Based Model (KB)*);
- полная атрибутивная модель (*Fully Attributed Model (FA)*).

**Диаграмма «сущность-связь»** включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные сущности и связи между ними, которые удовлетворяют основным требованиям, предъявленным к ИС. Диаграмма сущность-связь может включать связи «многие ко многим» и не включать описание ключей. Как правило, ERD используется для презентаций и обсуждения структуры данных с экспертами предметной области.

**Модель данных, основанная на ключах**, - более подробное представление данных. Она включает описание всех сущностей и первичных ключей и предназначена для представления структуры данных и ключей, которые соответствуют предметной

области.

**Полная атрибутивная модель** – наиболее детальное представление структуры данных, включая все сущности, атрибуты и связи.

### 3.2. Основные элементы модели данных логического уровня

#### Сущности и атрибуты

**Сущность** представляет собой **множество экземпляров реальных или абстрактных объектов** (людей, событий, состояний, идей, предметов и т.п.), **обладающих общими атрибутами или характеристиками**. Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована. *Именованная сущность* осуществляется с помощью *существительного в единственном числе*. При этом имя сущности должно отражать тип или класс объекта, а не его конкретный экземпляр (например, АЭРОПОРТ, а не ВНУКОВО).

В нотации IDEF1X сущность изображается в виде прямоугольника, в зависимости от подуровня представления данных могут быть некоторые различия (рис. 3.1):

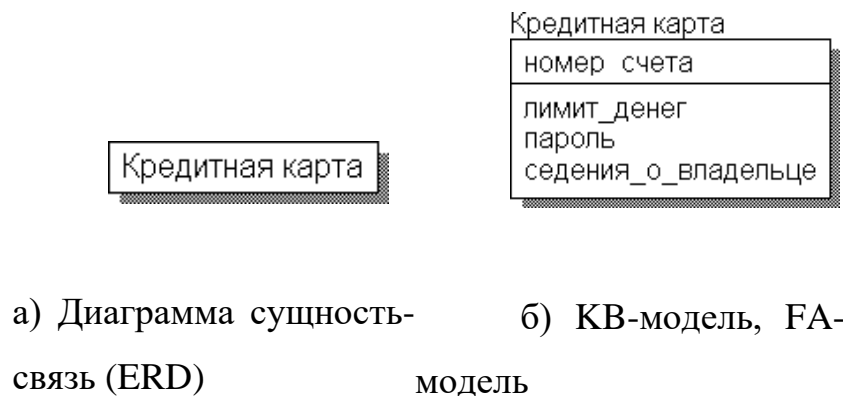


Рис. 3.1. Графическое представление сущностей в пакете ERWin Data Modeler

Примечание: если сущность является зависимой от другой сущности (сущностью-потомком), то она изображается прямоугольником со скругленными углами (см. рис. 3.3, сущность «Кредитная карта»). **Атрибуты (Attribute)** – характеристики сущностей. Экземпляр атрибута – это определенная характеристика экземпляра сущности. Например, допустим, что сущность «Работник» имеет атрибут «ФИО». В этом случае экземпляр сущности «Работник» (конкретный человек)

будет иметь экземпляр атрибута «ФИО».

*Атрибуты сущности бывают собственные или наследуемые. Собственные атрибуты являются уникальными в рамках модели. Наследуемые атрибуты передаются от сущности-родителя при определении идентифицирующей связи.*

**Первичные ключи (Primary Key).** Каждая сущность должна обладать *атрибутом или комбинацией атрибутов, чьи значения однозначно определяют каждый экземпляр сущности.* Эти атрибуты образуют *первичный ключ* сущности. При существовании нескольких возможных ключей один из них назначается первичным, остальные формируют **альтернативные ключи (Alternative Key)**, причем их число не ограничено. Отдельный атрибут может входить в качестве компоненты в более чем один альтернативный ключ. Атрибут первичного ключа может также служить частью альтернативного ключа.

Каждому альтернативному ключу присваивается уникальный номер и указатель «АК» (например, АК2.1 – означает, что этот атрибут входит во второй альтернативный ключ (2) и в нем является первым атрибутом (1). Первичный ключ обозначается иконкой.

Синтаксис первичного и альтернативных ключей показан на рис. 3.2.

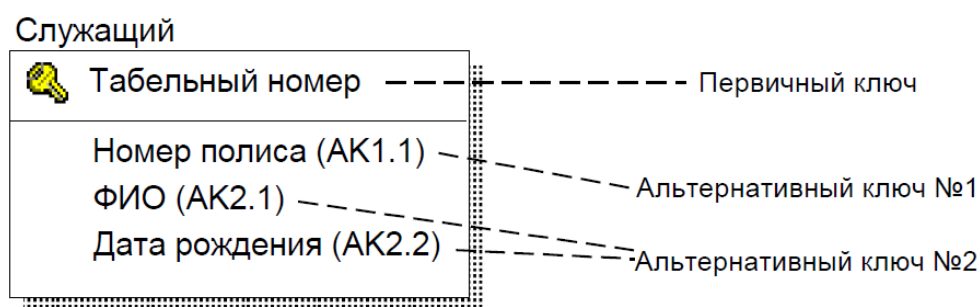


Рис. 3.2. Сущность с первичным и альтернативными ключами

**Внешние ключи (Foreign Key)**– если между двумя сущностями *имеется специфическое отношение* связи или *категоризации*, то *атрибуты, входящие в первичный ключ родительской или общей сущности, наследуются в качестве атрибутов сущностью-потомком или категориальной сущностью* соответственно. Эти атрибуты и называются внешними ключами. Наследуемый атрибут может использоваться в сущности в качестве части или целого первичного ключа,

альтернативного ключа или неключевого атрибута.

Внешний ключ изображается путем помещения внутрь блока сущности имен наследуемых атрибутов, после которых следуют буквы «FK» (они помещаются автоматически при создании связи). Пример наследования атрибутов представлен на рис. 3.3.

### **Связи (отношения – Relationship)**

**Связь является логическим соотношением между сущностями.** *Именованное отношения* осуществляется с помощью грамматического оборота *глагола* (ИМЕЕТ, ОПРЕДЕЛЯЕТ, МОЖЕТ ВЛАДЕТЬ и т.п.).

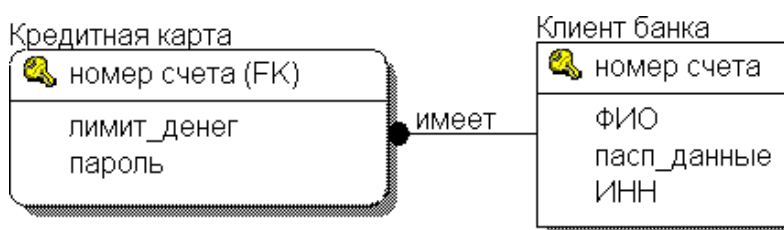


Рис. 3.3. Пример наследования первичного ключа

На логическом уровне можно установить идентифицирующую связь «один ко многим», связь «многие ко многим» и неидентифицирующую связь «один ко многим».

Для определения связей ERWin выбирается тип связи, затем мышью указывается родительская и дочерняя сущность. *Идентифицирующая связь изображается сплошной линией; не идентифицирующая - пунктирной линией.* Линии заканчиваются точкой со стороны дочерней сущности. При определении связи происходит миграция атрибутов первичного ключа родительской сущности в соответствующую область атрибутов дочерней сущности. Поэтому такие атрибуты не вводятся вручную.

**Мощность связи (Cardinality)** служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней.

### ***Отношения категоризации***

Сущность может быть разделена и представлена в виде 2 и более сущностей-категорий, каждая из которых имеет общие атрибуты и/или отношения, которые определяются однажды на верхнем уровне и наследуются на нижнем. Сущности-категории могут иметь и свои собственные атрибуты и/или отношения, а также, в свою очередь, могут быть декомпозированы своими сущностями-категориями

на следующем уровне. *Расщепляемая* на категории сущность получила название *общей сущности* (отметим, что на промежуточных уровнях декомпозиции одна и та же сущность может быть как общей сущностью, так и сущностью-категорией). *Узел, связывающий сущности-категории с общей*, называется *дискриминатором*. В зависимости от того, все ли сущности-категории включены в модель, отношение категоризации может быть полным или неполным. При отношении полной категоризации дискриминатор изображается в виде круга с двойным подчеркиванием, при неполной категоризации – с одинарным (рис. 3.4).

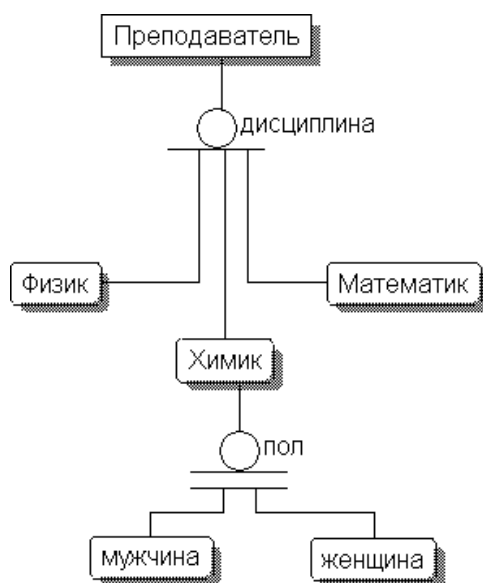


Рис. 3.4. Отношение полной и неполной категоризации в ERWin

### 3.3. Нормализация данных

Нормализация – это процесс проверки и реорганизации сущностей и атрибутов с целью удовлетворения требований к реляционной модели данных. Процесс нормализации сводится к последовательному приведению структур данных к нормальным формам – формализованным требованиям к организации данных.

Первая нормальная форма (1НФ). Сущность находится в первой нормальной форме тогда и только тогда, когда все атрибуты содержат атомарные значения. Среди атрибутов не должно встречаться повторяющихся групп, т.е. несколько значений для каждого экземпляра.

Вторая нормальная форма (2НФ). Сущность находится во второй нормальной форме, если она находится в первой нормальной форме и каждый не ключевой атрибут

полностью зависит от первичного ключа (не может быть зависимости от части ключа).

Третья нормальная форма (3 НФ). Сущность находится в третьей нормальной форме, если она находится во второй нормальной форме и никакой не ключевой атрибут не зависит от другого не ключевого атрибута (не должно быть зависимости между не ключевыми атрибутами).

## **Методика выполнения лабораторной работы**

### ***3.4. Составление пула – списка потенциальных сущностей***

Информационная модель может быть построена на основе функциональной или без нее. Использование функциональной модели в качестве основы для информационного моделирования позволяет создать структуру базы данных, полностью соответствующую функциям предприятия. Названия всех стрелок заносятся в пул – список потенциальных сущностей. Только в данном случае информационная модель будет адекватна выполняемым функциям. Функциональная модель рассматриваемого примера представлена в прил.

Список потенциальных сущностей может быть составлен вручную, но можно использовать глоссарий – Arrow Dictionary, или отчет по стрелкам – Arrow Report, который содержит только названия стрелок (рис. 4.1).

<b>Arrow Name</b>		
Варианты заданий		
График		
Графическая часть		
Задание		
Замечания, дополнения		
Курсовая работа		
Литература		
Методические указания		
Оценка за курсовую работу		
Положение	о	курсовом

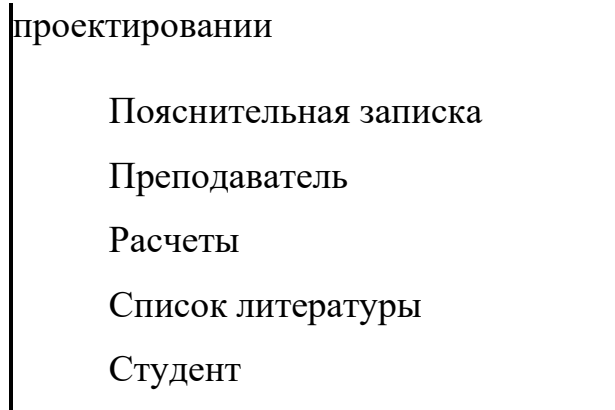


Рис. 4.1. Пул – список потенциальных сущностей

Теперь из этого списка необходимо выделить сущности, остальные стрелки станут атрибутами сущностей.

В качестве сущностей выделим следующие:

- 1) задание;
- 2) пояснительная записка;
- 3) курсовая работа;
- 4) положение о курсовом проектировании;
- 5) студент;
- 6) преподаватель;
- 7) график;
- 8) методические указания.

### ***3.5. Создание логической модели «сущность-связь»***

1. Запустите ERWin Data Modeler.
2. В появившемся диалоге выберите Create a new model (или File/New в меню в случае отсутствия диалогового окна).

3. В появившемся диалоговом окне (рис. 4.2) отметьте галочкой пункт Logical/Physical, это позволит создать модель как логического, так и физического уровня, а впоследствии сгенерировать схему БД. Нажмите кнопку ОК.

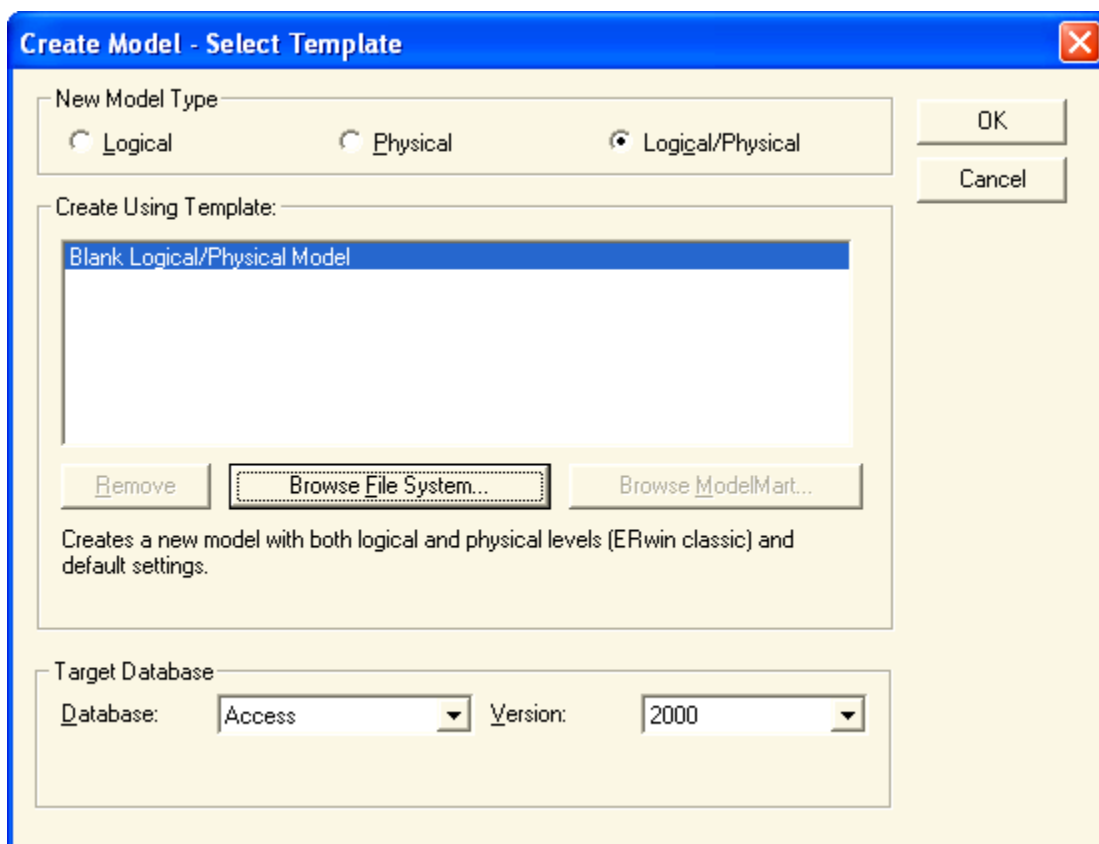


Рис. 4.2. Диалог выбора типа модели

4. Прежде всего, необходимо изменить настройки шрифта. Меню Format/Default Fonts&Colors..., закладка General. Установите шрифт Arial Cyr, отметьте галочкой пункт All Objects.

5. Для того чтобы создать сущность, необходимо нажать кнопку



. Кроме того, необходимо выбрать способ отображения сущности, нажав



кнопку . В этом случае, будет отображаться только имя



сущности, без атрибутов (что и требуется для создания диаграммы «Сущность-связь»). Создадим 6 сущностей, определенных в п. 4.1. Результат представлен на рис. 4.3.

6. Все сущности должны быть описаны. Для этого необходимо щелкнуть правой кнопкой мыши по сущности, выбрать меню Entity Properties и внести в раздел Definition определение (рис. 4.4).



Рис. 4.3. Сущности модели

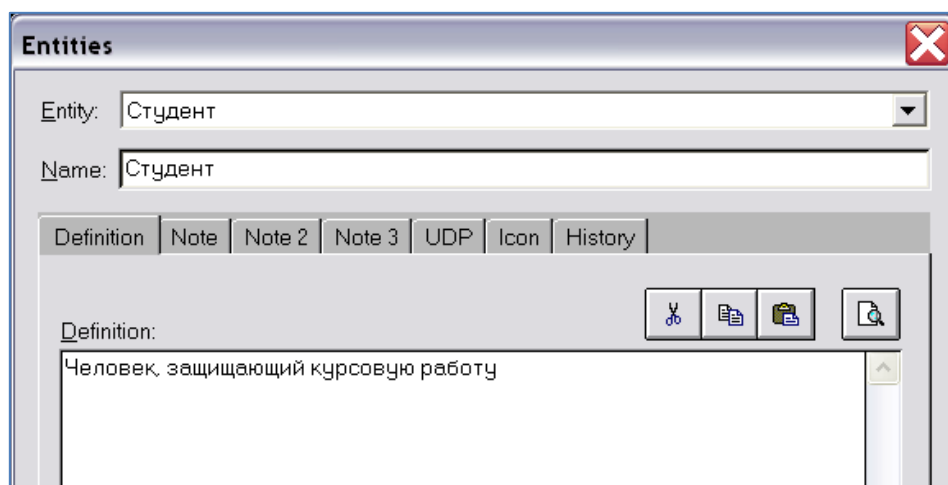
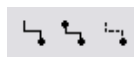


Рис. 4.4. Диалоговое окно Entity Properties

7. Для того чтобы установить связь, необходимо воспользоваться кнопками



в зависимости от вида устанавливаемой связи.

Сначала составим описание предметной области на естественном языке.

Любой студент должен выполнить одну или несколько курсовых работ.

Каждая курсовая работа должна выполняться одним студентом (в идеале).

Каждая курсовая работа выполняется в соответствии с методическими указаниями и положением о курсовом проектировании.

Курсовая работа сдается по графику.

Курсовая работа оформляется в виде пояснительной записки.

Преподаватель проводит консультации, проверяет и ставит оценку за курсовую работу.

Таким образом, формулируем имена связей: **СТУДЕНТ** выполняет **КУРСОВУЮ РАБОТУ**. **ПРЕПОДАВАТЕЛЬ** проверяет **КУРСОВУЮ РАБОТУ**.

**КУРСОВАЯ РАБОТА** выполняется в соответствии с **ЗАДАНИЕМ**.

**КУРСОВАЯ РАБОТА** оформляется в виде **ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ** определяют требования к **КУРСОВОЙ РАБОТЕ**.

**КУРСОВАЯ РАБОТА** организуется согласно **ПОЛОЖЕНИЮ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ**.

**КУРСОВАЯ РАБОТА** сдается по **ГРАФИКУ**.

Все связи являются связями «один ко многим». Во всех случаях сущность «Курсовая работа» является дочерней, за исключением связи с сущностью «Пояснительная записка». Определим типы связей и построим модель (рис. 4.5). Это начальный этап построения модели, в дальнейшем можно будет подкорректировать связи между сущностями.

8. Для того чтобы задать имя связи необходимо щелкнуть по ней правой кнопкой мыши, в появившемся меню выбрать Relationship Properties и задать Verb Phrase (рис. 4.6).

9. Чтобы сделать видимым имя связи, необходимо выбрать меню Format/Relationship Display и во всплывающем меню отметить галочкой Verb Phrase.

Также необходимо отметить галочкой Cardinality – для отображения мощности связи.

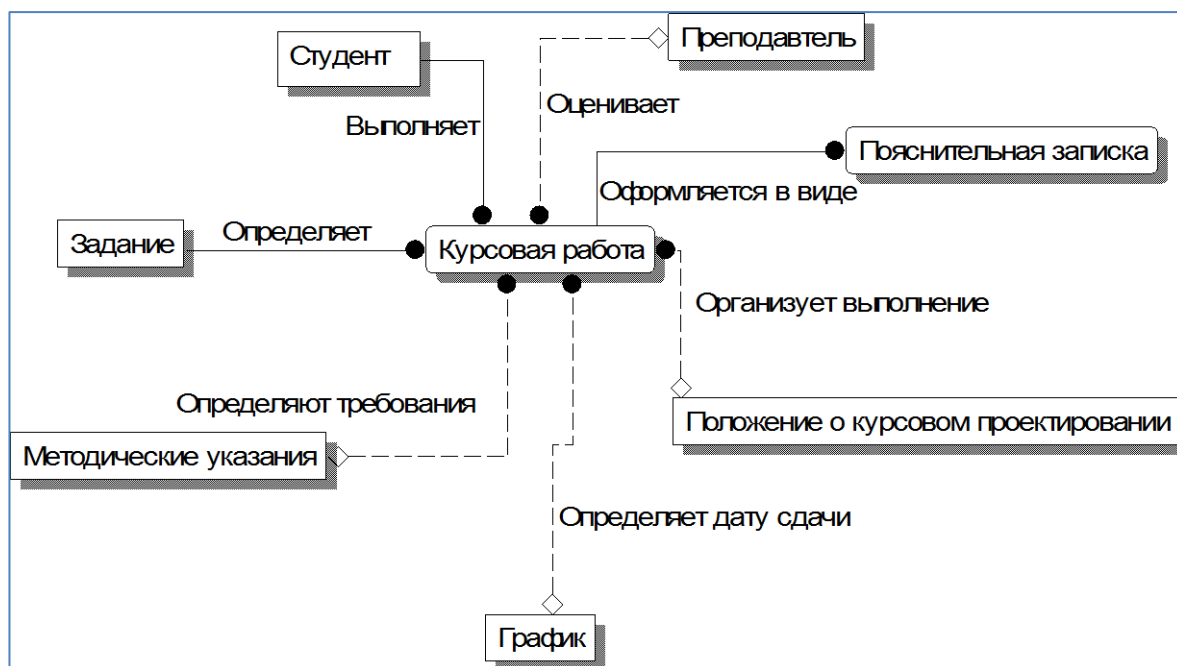


Рис. 4.5. Модель «сущность-связь»

**Relationships**

Relationship: Задание Определяет Курсовая работа

New... Delete

General | Definition | Rolename | RI Actions | UDP

Verb Phrase

Parent-to-Child: Определяет

Child-to-Parent:

Relationship Cardinality

Summary: One-to-Zero-One-or-More

Cardinality

☒ Zero, One or More

☐ One or More (P)

☐ Zero or One (Z)

☐ Exactly:

Relationship Type

☒ Identifying

☐ Non-Identifying

Nulls

☐ Nulls Allowed

☐ No Nulls


☐ Logical Only

OK Cancel

Рис. 4.6. Диалог Relationship Properties

### 3.6. Создание логической модели данных, основанной на ключах

1. Необходимо определить ключевые атрибуты для каждой сущности, обращая внимание на то, что дочерние сущности наследуют ключевые атрибуты от родительских (см. рис. 4.8).

2. Сменим режим отображения сущностей нажатием кнопки  на панели задач (при этом будут отображаться не только названия сущностей, но и атрибуты).

3. Чтобы добавить атрибуты, необходимо щелкнуть правой кнопкой мыши по сущности, выбрать пункт меню Attributes, в открывающемся диалоговом окне нажатием кнопки New добавить атрибут. Чтобы сделать его ключевым, необходимо поставить галочку напротив Primary Key (рис.4.7). Переименовать и удалить атрибут можно нажатием кнопок Rename или Delete соответственно. Кроме того, можно задать условный тип данных («блоб», дата, число или строка). Определение атрибутов задается на вкладке Definition.

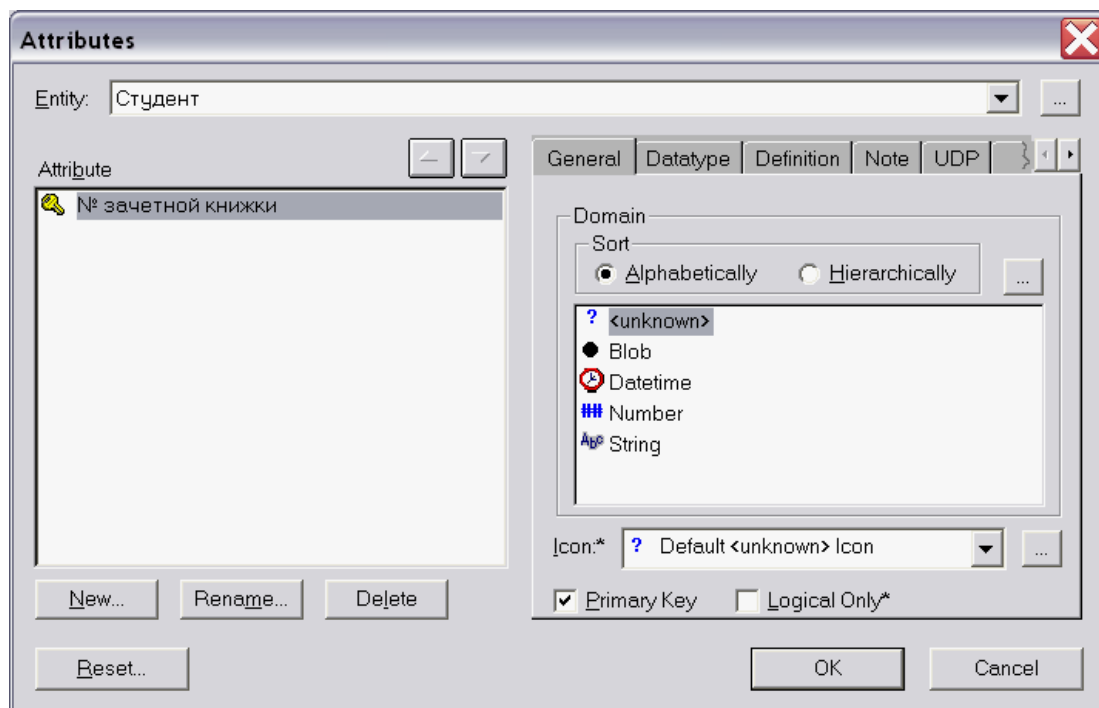


Рис. 4.7. Диалог Attributes

На данном этапе, когда видно, как наследуются ключи, можно подкорректировать типы связей с целью устранения избыточности. Получим, что сущность «Курсовая работа» однозначно идентифицируется составным ключом, содержащим код зачетной книжки и код дисциплины, специальность и курс.

Модель данных, основанная на ключах, представлена на рис. 4.8. Чтобы сделать видимым значок ключа рядом с атрибутом, необходимо выбрать пункт меню Format/Entity Display/Primary Key Designator.

### 3.7. Создание полной атрибутивной модели

Для того чтобы получить полную атрибутивную модель, необходимо дополнить сущности неключевыми атрибутами. Дополненная модель представлена на рисунке 4.9.

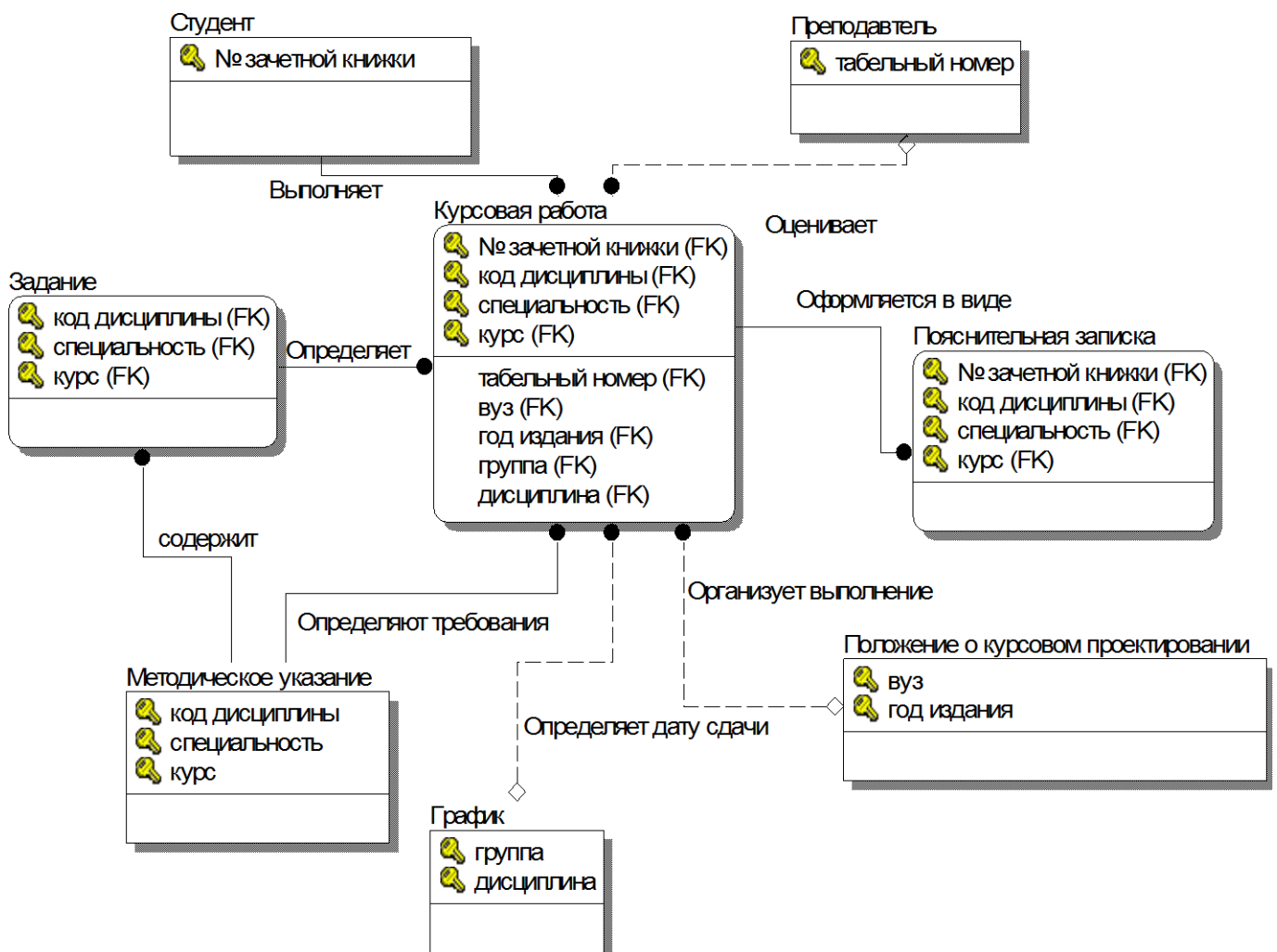


Рис. 4.8. Модель данных, основанная на ключах

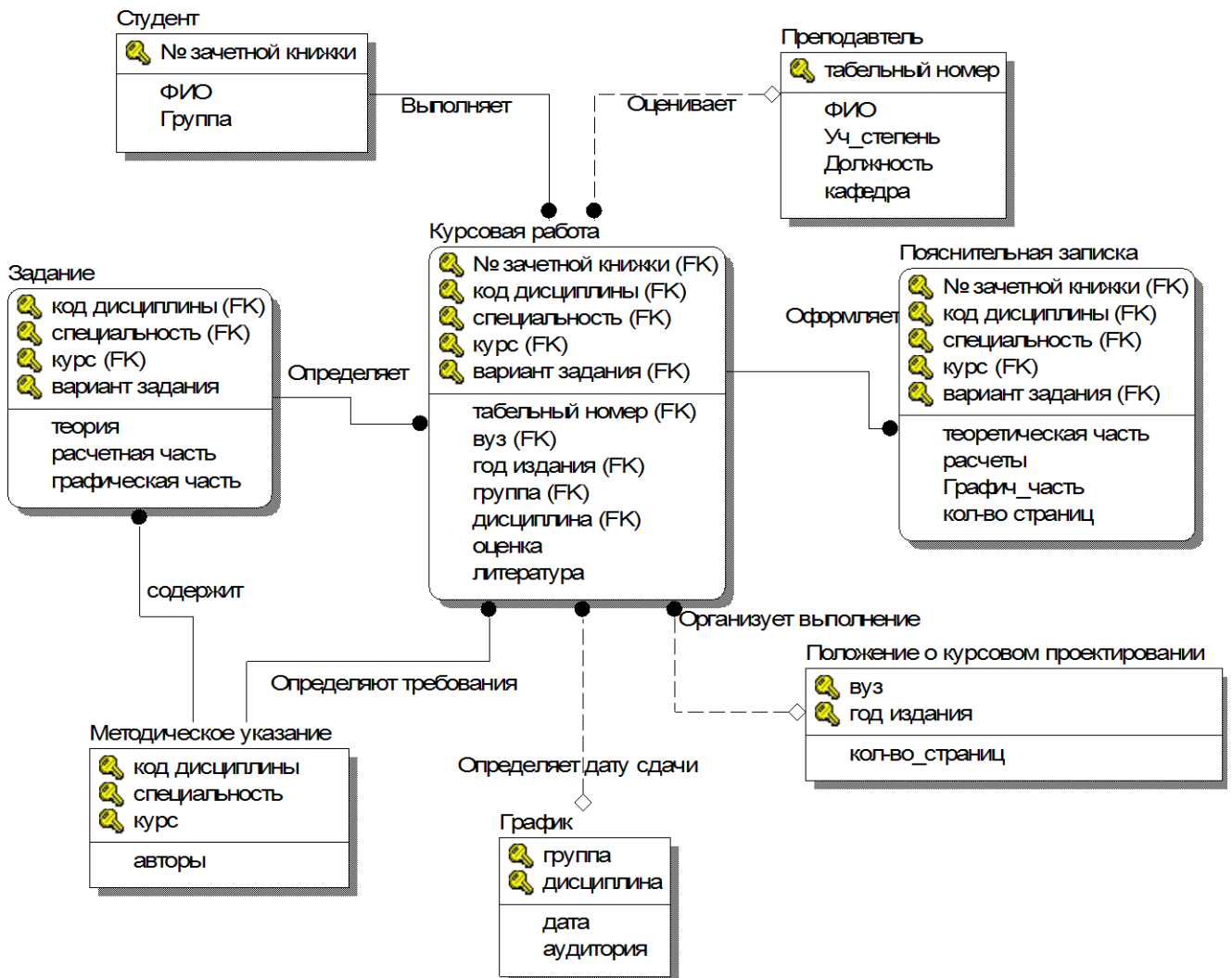


Рис. 4.9. Полная атрибутивная модель

### 3.8. Нормализация полной атрибутивной модели

Проверим, все ли атрибуты имеют атомарные значения, т.е. среди атрибутов не должно встречаться повторяющихся групп, нескольких значений для каждого экземпляра (например, номер телефона\_1, номер телефона\_2).

Видим, что атрибут «авторы» в сущности «Методическое указание» не удовлетворяет требованиям 1 НФ (у методических указаний может быть несколько авторов). Необходимо выделить сущность, которая будет содержать сведения об авторах методических указаний. Поскольку авторами всегда являются преподаватели вузов, новую сущность выделять не имеет смысла, свяжем сущности «Методическое указание» и «Преподаватель», предварительно удалив атрибут «авторы». Добавим атрибуту

«Табельный номер» имя роли «Преподаватель», чтобы облегчить понимание модели. Кроме того, в сущности «График» атрибут дисциплина также не отвечает

требованиям атомарности. Выделим отдельную сущность «Дисциплина» и свяжем ее с сущностями, которые зависят от нее. Атрибут «ФИО» является составным и не соответствует требованию атомарности, его необходимо разбить на 3. Остальные атрибуты соответствуют 1 НФ. Атрибутивная модель, приведенная к 1 НФ, представлена на рис. 4.10.

Приведем модель ко 2 НФ, проверим, все ли атрибуты зависят от составного ключа, а не его части. Проверка показала, что все неключевые атрибуты сущностей зависят от составного ключа полностью. Значит, модель удовлетворяет требованиям 2 НФ.

Проверим, есть ли транзитивная зависимость между не ключевыми атрибутами. Проверка показала, что взаимозависимости между не ключевыми атрибутами нет, но в сущности «График» есть зависимость между атрибутами ключа, т.к. «группа» зависит от «специальности». Выделим 2 сущности с соответствующими названиями и установим связи между ними. Модель, приведенная к 3НФ, представлена на рис. 4.11.

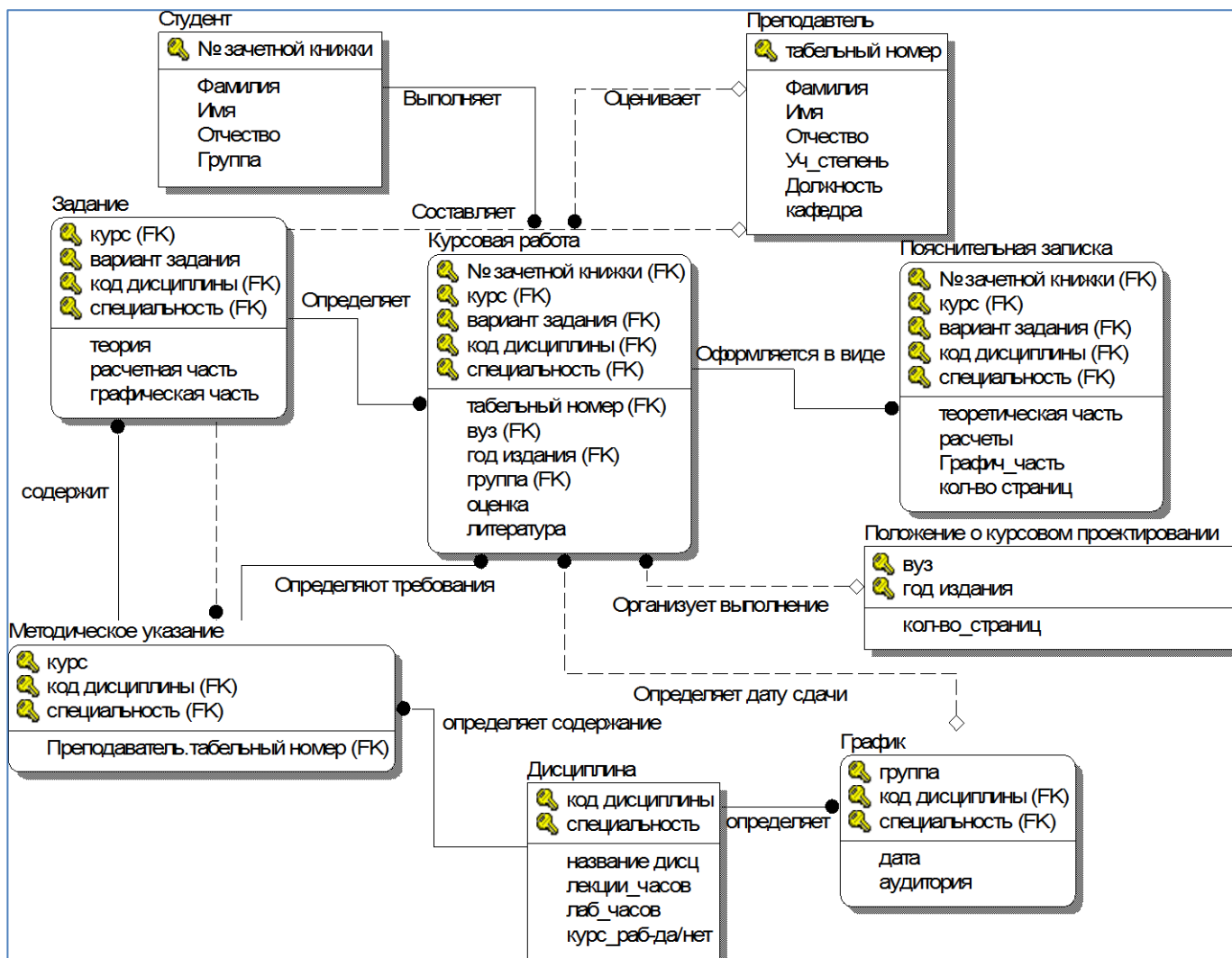


Рис. 4.10. Модель, приведенная к 1 НФ



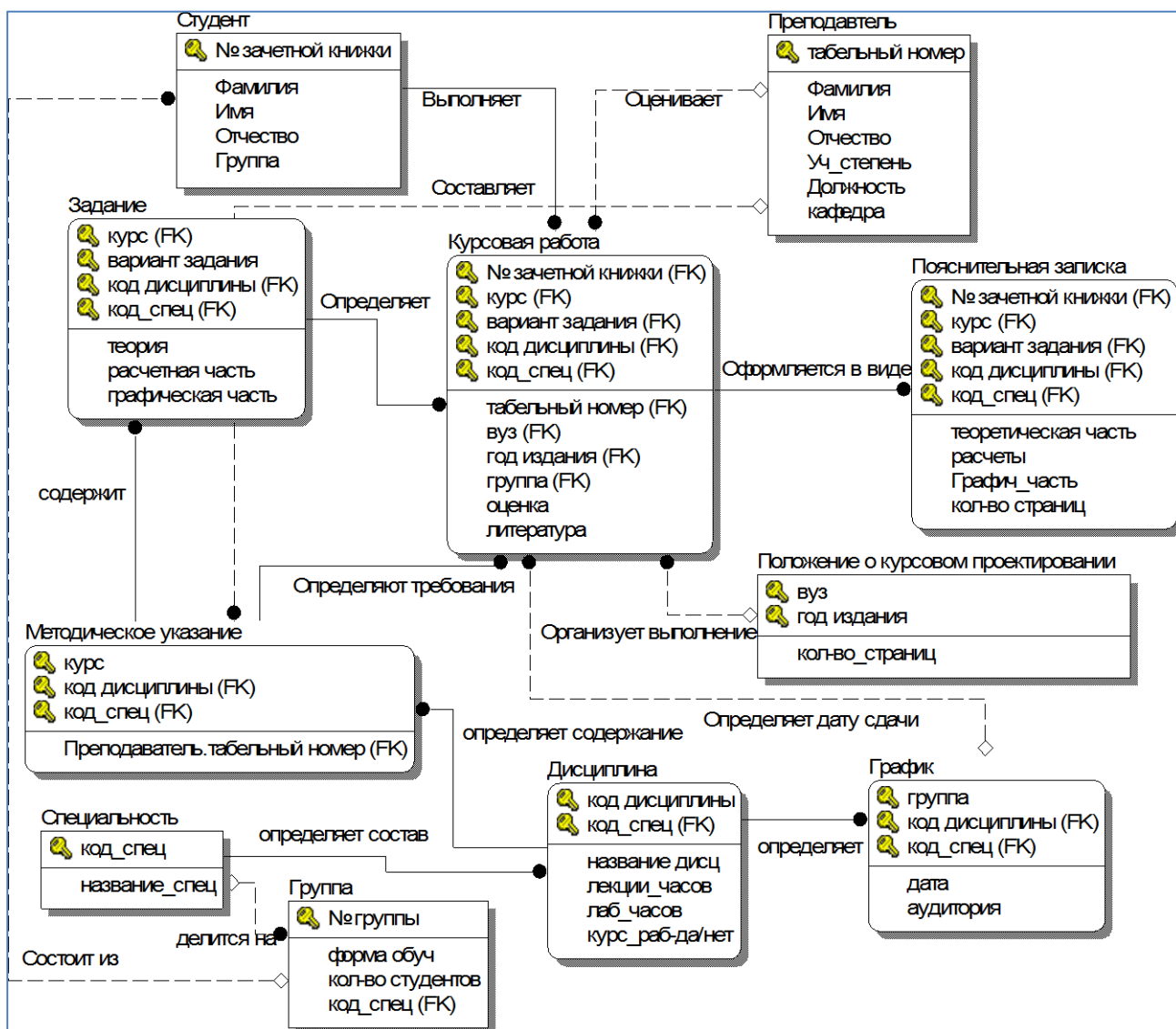


Рис. 4.11. Логическая модель, приведенная к 3НФ

Проставим мощности связей. Для этого щелкаем правой кнопкой мыши по связи, во всплывающем меню выбираем Relation ship Properties (рис. 4.6) и ставим один из 4-х типов мощности (на- пример, 1 студент выполняет 1 или много курсовых работ и т.п.). Нормализованная модель с расставленными мощностями представлена на рис. 4.12.

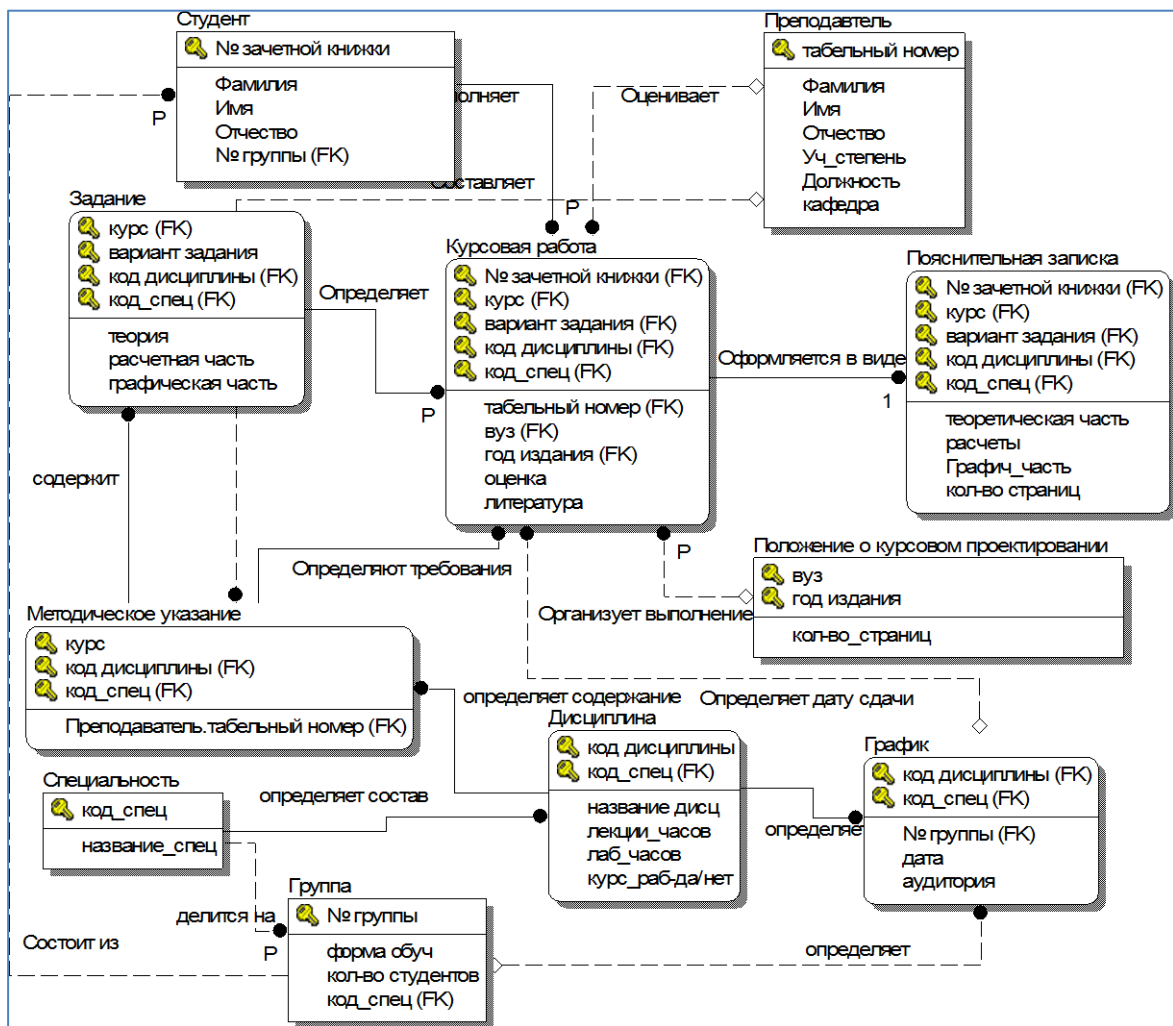


Рис. 4.12. Нормализованная модель

с проставленными мощностями связей

### Задание

В соответствии с вариантом задания построить полную атрибутивную модель данных логического уровня, привести ее к третьей нормальной форме.

#### 1. Контрольные вопросы

1. Для чего предназначена диаграмма «сущность-связь», дайте определение сущности. Назовите виды сущностей.
3. Чем отличается полная атрибутивная модель от диаграммы «сущность-связь»?
4. Какие типы отношений существуют и чем они отличаются?
5. Что такое мощность связи?
6. Чем отличаются отношения полной и неполной категоризации?
7. Что представляет собой нормализация? Сколько нормальных форм вы знаете?