

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1. Цель работы .....	5
2. Задачи .....	5
3. Краткие теоретические сведения.....	5
4. Рекомендации по выполнению лабораторных работ .....	7
5. Методика выполнения лабораторной работы.....	8
Упражнение 1. Построение логической информационной модели уровня «сущность-связь» .....	8
Упражнение 2. Разработка логической модели данных, основанной на ключах .....	15
Упражнение 3. Создание полной атрибутивной модели .....	16
Упражнение 4. Нормализация полной атрибутивной модели .....	17
Упражнение 5. Создание физической модели .....	19
6. Задание .....	23
7. Порядок выполнения работы .....	23
8. Критерии результативности работы .....	25
9. Требования к содержанию и оформлению отчета.....	25
10. Контрольные вопросы .....	25
СПИСОК ЛИТЕРАТУРЫ .....	26
Приложение А .....	27
Приложение Б .....	28

## ВВЕДЕНИЕ

Проектирование информационных систем – это сложнейшая задача, предполагающая использование различных методик и инструментов. Одной из важнейших проблем, стоящих перед проектировщиком, является определение структуры базы данных, соответствующей функциям автоматизируемого предприятия или подразделения.

Студенты специальностей, связанных с информационными системами и информационными технологиями, должны иметь навыки информационного моделирования и проектирования БД на основе созданных моделей.

Освоение технологии проектирования БД информационных систем начинается с изучения основных принципов системного анализа, методологии информационного моделирования в рамках дисциплины «Системное моделирование и CASE-технологии». Затем, после изучения дисциплины «Базы данных» полученные знания и навыки применяются для построения информационных моделей предметной области и проектирования баз данных в курсе «Проектирование информационных систем» («Проектирование АСОИУ»).

Для анализа структуры данных и информационного моделирования существует множество программных продуктов. Один из них – Microsoft Office Visio 2007, поддерживающий методологию IDEF1X (Integration DEFinition for Information Modeling). Он позволяет построить логическую модель данных, представляющую собой совокупность информационных объектов и связей между ними, а также физическую модель, непосредственно связанную с конкретной СУБД.

Полученные в результате выполнения лабораторного практикума навыки могут быть использованы студентами указанных специальностей в ходе дипломного проектирования, а также в практической деятельности по анализу и проектированию информационных систем.

При необходимости дополнительного изучения вопросов, связанных с информационным моделированием или теорией баз данных, следует обратиться к литературным источникам.

## 1. Цель работы

Целью работы является освоение технологии построения информационной модели логического и физического уровней в нотации IDEF1X с использованием пакета Microsoft Office Visio 2007.

## 2. Задачи

Основными задачами лабораторного практикума являются: приобретение студентами навыков построения информационной модели логического уровня, нормализации полученной модели, а также построения информационной модели физического уровня.

## 3. Краткие теоретические сведения

### 3.1. *Понятие информационной модели. Уровни информационной модели*

Методология IDEF1X – язык для семантического моделирования данных, основанный на концепции «сущность-связь».

Различают два уровня информационной модели: **логический** и **физический**.

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

Различают три 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)).

**Физическая модель** отражает физические свойства проектируемой базы данных (типы данных, размер полей, индексы). Параметры физической информационной модели зависят от выбранной системы управления базами данных (СУБД).

### 3.2. *Основные элементы информационной модели логического уровня*

#### 3.2.1. *Сущности и атрибуты*

**Сущность** – это множество **реальных** или **абстрактных объектов** (людей, предметов, документов 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, **Студент**, а не **Петров**) (рис. 3.1).



Рис. 3.1. Графическое представление сущности «Студент» в MS Office Visio

Любая сущность характеризуется набором атрибутов (**свойств**).

**Атрибут сущности** – характеристика сущности, то есть свойство реального объекта. Например, на рис. 3.1 атрибутами сущности «Студент» являются «ID студента», «Фамилия», «Имя», «Отчество», «Дата поступления» и «Номер билета».

В свою очередь, *атрибуты сущности* делятся на 2 вида: *собственные* и *наследуемые*. *Собственные* атрибуты являются уникальными в рамках модели. *Наследуемые* атрибуты передаются от сущности-родителя при установке связи с другими сущностями.

**Первичный ключ (Primary Key, PK).** Каждая сущность должна обладать *атрибутом* или *комбинацией атрибутов*, чьи значения *однозначно определяют* каждый экземпляр сущности. Эти атрибуты образуют *первичный ключ* сущности.

**Внешний ключ (Foreign Key, FK).** 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*.

### 3.2.2. Отношения в IDEF1X-модели

При построении информационной модели различают следующие типы отношений между сущностями: *идентифицирующее*, *не идентифицирующее*, *не специфическое* (*многие-ко-многим*) и *отношения категоризации*.

**Мощность отношения** служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней.

### ***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).

## **4. Рекомендации по выполнению лабораторных работ**

Лабораторная работа выполняется группой студентов (2-3 человека) в пакете Microsoft Office Visio 2007.

Данная работа может выполняться на основе результатов функционального моделирования предметной области.

Отчеты по лабораторным работам оформляется в виде документа в формате MS Office Word в соответствии с требованиями преподавателя.

## 5. Методика выполнения лабораторной работы

Методика выполнения лабораторной работы будет рассмотрена на примере процесса выполнения курсовой работы, функциональная модель которого представлена в *приложении А*.

### Упражнение 1. Построение логической информационной модели уровня «сущность-связь»

#### 5.1. Составление пула – списка потенциальных сущностей

Информационная модель может быть построена на основе функциональной модели (в нотации IDEF0). В этом случае названия всех интерфейсных дуг IDEF0-модели заносятся в *пул* – список потенциальных сущностей.

Список потенциальных сущностей для рассматриваемого примера будет представлен таблицей вида (рис. 5.1):

Варианты заданий
График
Графическая часть
Задание
Замечания, дополнения
Курсовая работа
Литература
Методические указания
Оценка за курсовую работу
Положение о курсовом проектировании
Пояснительная записка
Преподаватель
Расчеты
Список литературы
Студент

Рис. 5.1. Пул – список потенциальных сущностей

Теперь из этого списка необходимо выделить *сущности*, остальные интерфейсные дуги будут преобразованы в *атрибуты 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.

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


1. Запустите MS Office Visio 2007.

2. На закладке выбора шаблона выберите категорию *Программное обеспечение и базы данных* и в ней элемент *Схема модели базы данных*. Нажмите кнопку *Создать* в правой части экрана.

3. Установите необходимые параметры страницы (масштаб, ориентация страницы).

4. MS Office Visio 2007 поддерживает различные нотации моделей баз данных. Для того чтобы задать нотацию IDEF1X, необходимо выбрать пункты меню *База данных* → *Параметры* → *Документ*. В открывшемся окне на вкладке *Общие* установить переключатель в меню *Набор символов* на *IDEF1X*. Меню *Имена, видимые на схеме* позволяет указать, какие имена атрибутов сущности будут отображены на диаграмме (концептуальные, физические или оба варианта одновременно). В данном случае для логического представления информационной модели необходимо выбрать пункт *Концептуальные имена* (рис. 5.2).

В закладке *Отношение* окна *Параметры документа базы данных* в меню *Показывать* нужно отметить галочкой пункт *Мощность*, в меню *Отображение вида* выбрать пункт *Показывать вербальную фразу*, снять галочку в пункте *Обратный текст* (рис. 5.3). Данные настройки позволят отобразить имя и мощность связи в модели.

5. Для того чтобы создать сущность, необходимо перетащить элемент  на рабочее поле. Переход в режим редактирования сущности осуществляется двойным щелчком по сущности или по нажатию правой кнопки мыши и выбора пункта меню *Свойства базы данных*.

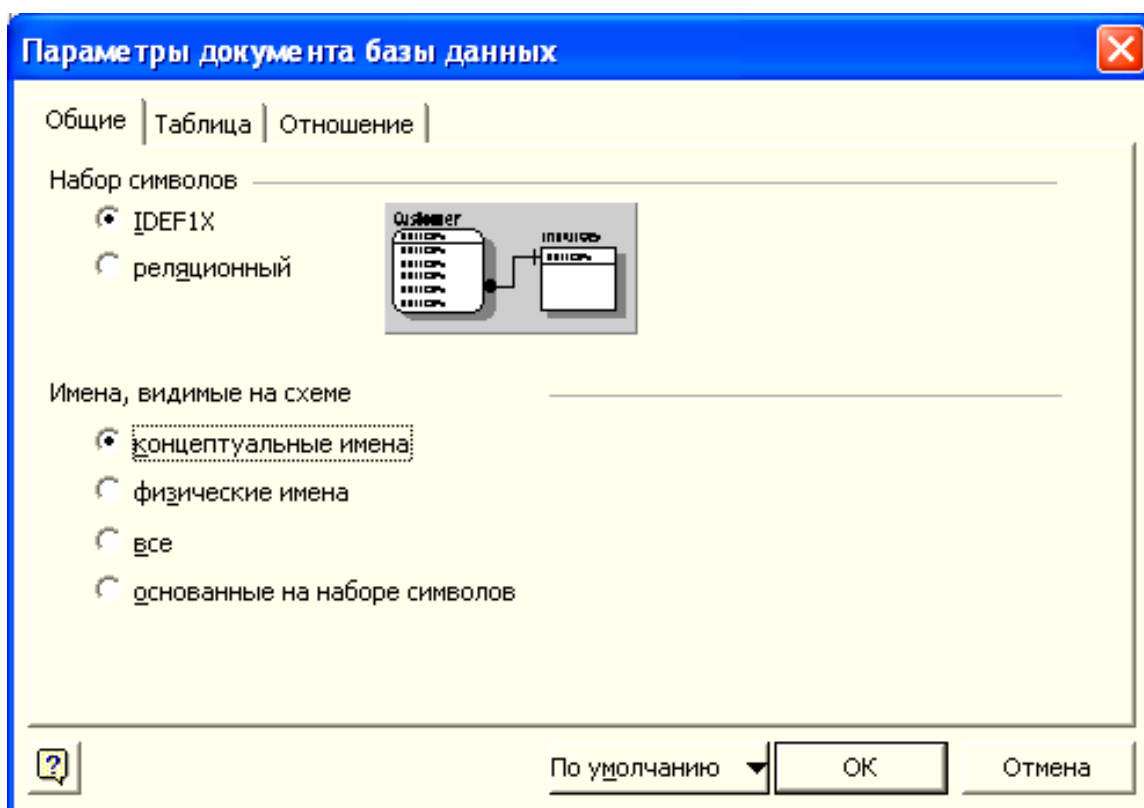


Рисунок 5.2 – Настройка параметров модели

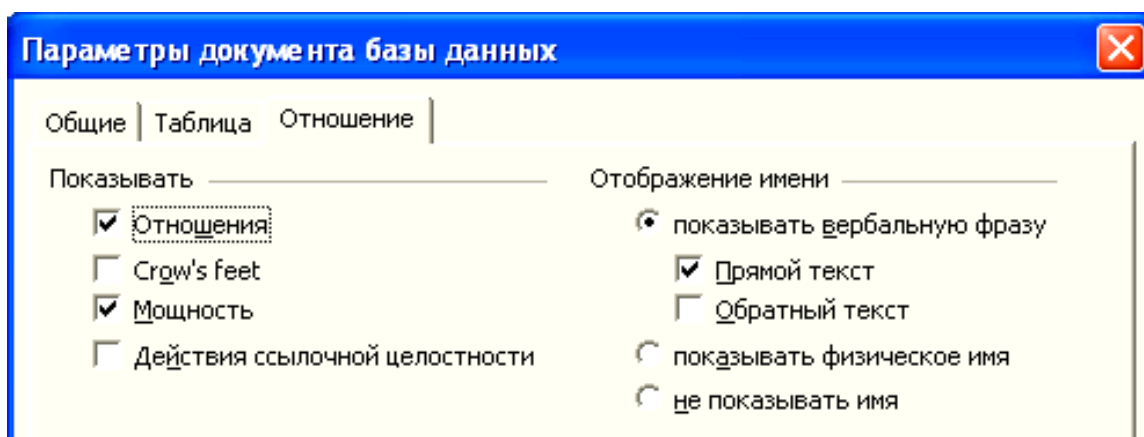


Рисунок 5.3 – Настройка вида отношений информационной модели

Чтобы задать имя сущности, в окне *Свойства базы данных* нужно выбрать категорию *Определение*, снять галочку в пункте *Синхронизация имен при вводе* (в противном случае, физическое и логическое имя сущности будут совпадать, что по практическим соображениям не всегда удобно) и задать концептуальное имя сущности. Руководствуясь данным алгоритмом, создадим 8 сущностей, определенных в пункте 5.1 (см. рис. 5.4).



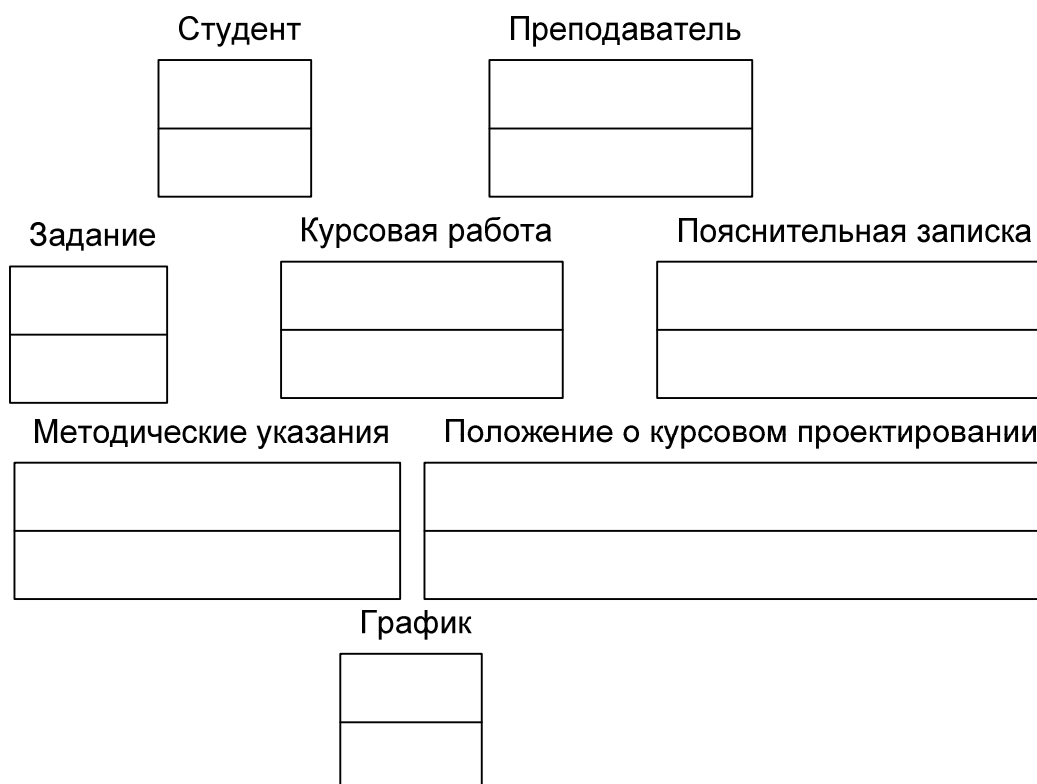


Рисунок 5.4 – Сущности информационной модели логического уровня

6. Далее необходимо установить связи между сущностями.

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, сформулируем 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**.

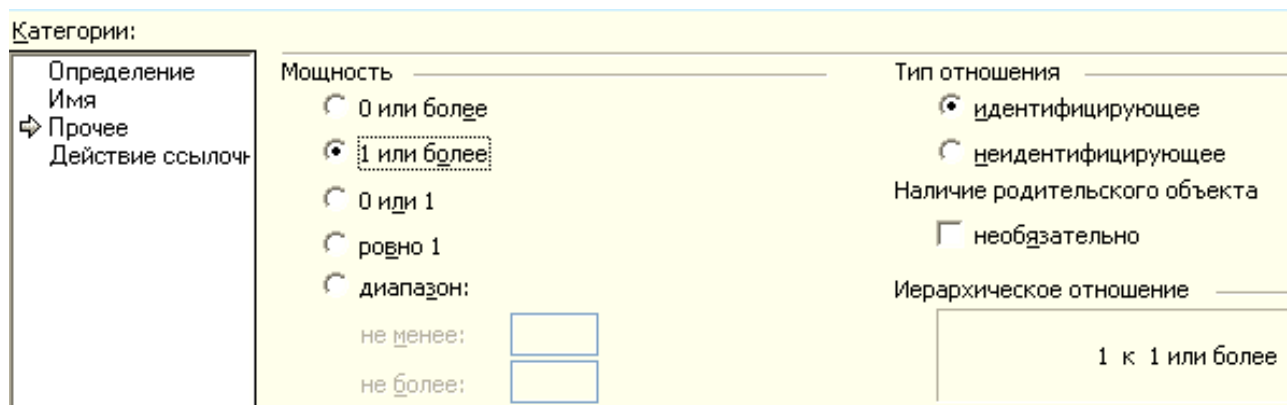
**КУРСОВАЯ РАБОТА** сдается по **ГРАФИКУ**.

Во всех случаях сущность *Курсовая работа* является дочерней, за исключением связи с сущностью *Пояснительная записка*. Определим типы связей и построим модель (см. рис. 5.7). В дальнейшем можно будет подкорректировать связи между сущностями.

Чтобы установить **связи** между сущностями, необходимо перетащить на рабочую область элемент , поднести один конец стрелки к родительской сущности, другой – к дочерней.

Примечание. При правильном связывании каждая сущность будет подсвечена красным цветом.

В MS Office Visio 2007 по умолчанию используется *не идентифицирующее* отношение. Чтобы изменить **тип связи**, необходимо двойным щелчком по связи открыть окно *Свойства базы данных* и в категории в категории *Прочее* указать тип отношения (идентифицирующее, не идентифицирующее). В этой же категории указывается мощность связи (см. рис. 5.5).



Категории:	Мощность	Тип отношения
Определение	<input type="radio"/> 0 или более	<input checked="" type="radio"/> идентифицирующее
Имя	<input checked="" type="radio"/> 1 или более	<input type="radio"/> неидентифицирующее
➔ Прочее	<input type="radio"/> 0 или 1	Наличие родительского объекта
Действие ссылок	<input type="radio"/> ровно 1	<input type="checkbox"/> обязательно
	<input type="radio"/> диапазон:	Иерархическое отношение
	не менее: <input type="text"/>	
	не более: <input type="text"/>	1 к 1 или более

Рисунок 5.5 – Определение типа связи и мощности

Примечание. Кроме того, при не идентифицирующем отношении нужно указать, является ли наличие родительской сущности обязательным (т.е. может ли существовать экземпляр

дочерней сущности, если не существует экземпляра родительской). Если наличие родительского объекта является необязательным, графически это отобразится в виде не закрашенного ромба со стороны родительской сущности.

Следующий шаг – в категории *Имя* в поле *Вербальная фраза* нужно указать имя отношения (рис. 5.6). Также можно указать имя связи в поле *Обратная фраза* для спецификации отношения потомок-родитель (в нашем случае обратная фраза отображаться не будет).

Примечание. Все изменения при закрытии окна свойств сохраняются автоматически.

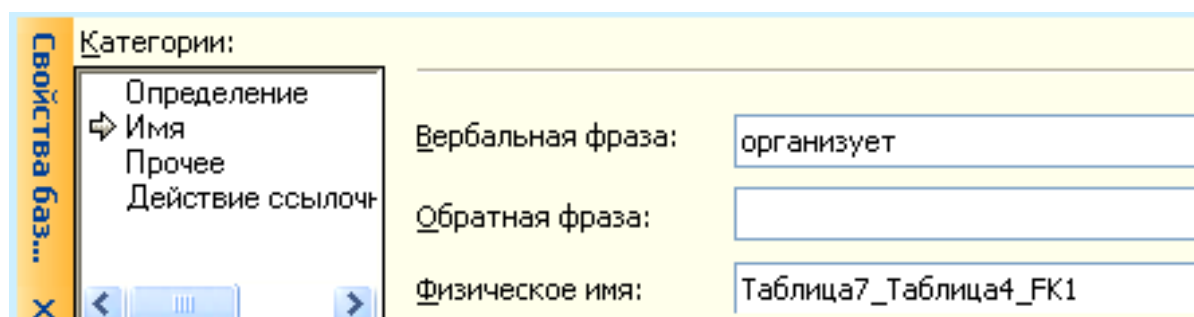


Рисунок 5.6 – Определение имени отношения

После определения имен, типов связей и задания мощностей получим информационную модель, представленную на рис. 5.7.

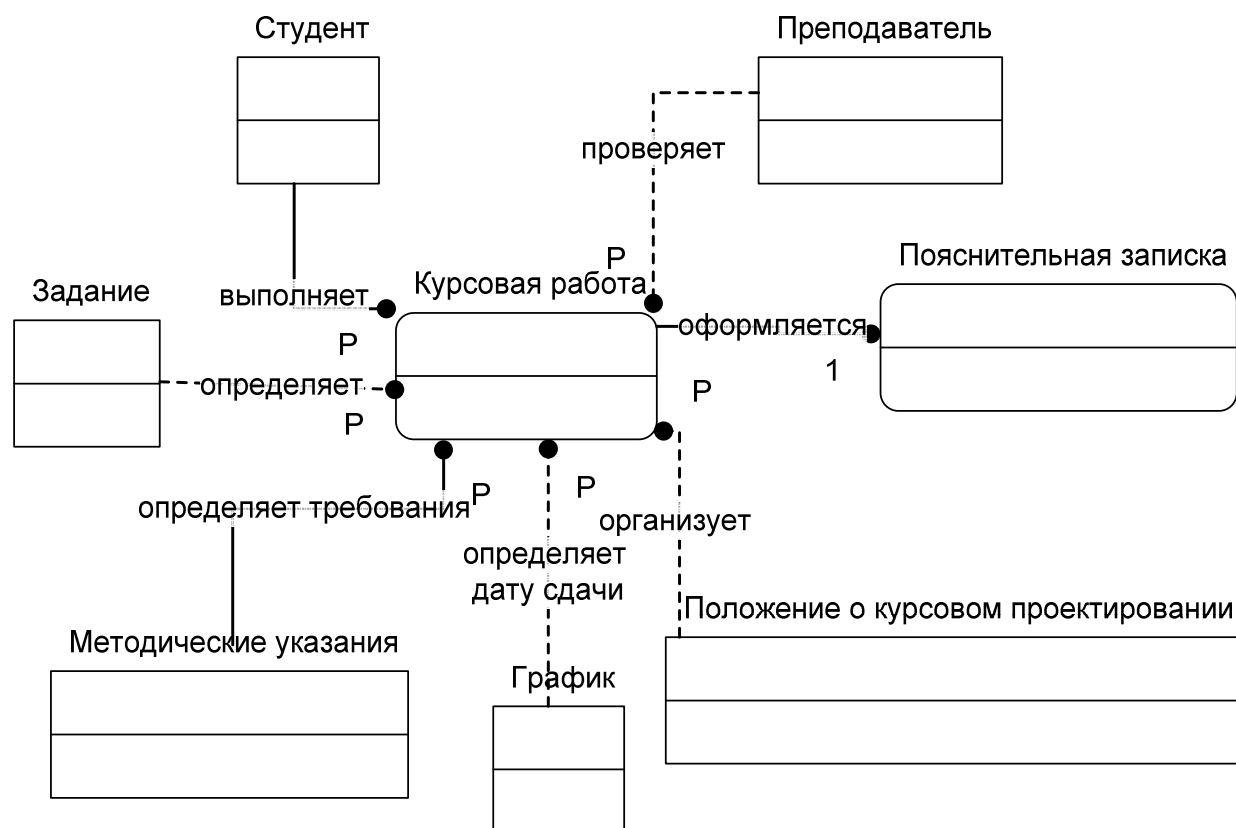


Рисунок 5.7 – Информационная модель уровня «сущность-связь»

## Упражнение 2. Разработка 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. 5.9).

Для этого двойным щелчком мыши по сущности откроем окно редактирования ее свойств, перейдем в категорию *Столбцы*, по нажатию кнопки *Добавить* введем имя поля (например, для сущности *Задание* ключевым атрибутом будет являться *Вариант задания*). Чтобы сделать атрибут ключевым, необходимо отметить галочкой пункт *РК* (рис. 5.8). Данное поле становится обязательным автоматически.

Физическое имя	Тип данных	Обязательное	РК	Заметки
Вариант задания	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Вариант
		<input type="checkbox"/>	<input type="checkbox"/>	

Показывать: ☐ Переносимый тип данных ☒ Физический тип данных (Microsoft Access)

Добавить  
Удалить  
Изменить...  
Вверх  
Вниз

Рисунок 5.8 – Определение ключевого атрибута

Аналогичным образом зададим ключевые атрибуты для всех сущностей информационной модели. Результат представлен на рис.5.9.

Как видно из рис. 5.9 по сравнению с информационной моделью уровня «сущность-связь», был изменен тип связи между сущностями *Методические указания* и *Курсовая работа*, поскольку ключевые атрибуты сущности *Методические указания* для сущности *Курсовая работа* будут являться избыточными (зная номер зачетной книжки, можно узнать специальность и курс, на котором учится студент).

2. Кроме того, отметим, что три сущности (*Задание*, *График*, *Методические указания*) содержат одинаковые атрибуты *Дисциплина*. Это является некорректным. Чтобы устранить данную ошибку, выделим одноименную сущность и свяжем ее идентифицирующими связями с вышеуказанными сущностями (рис. 5.10).

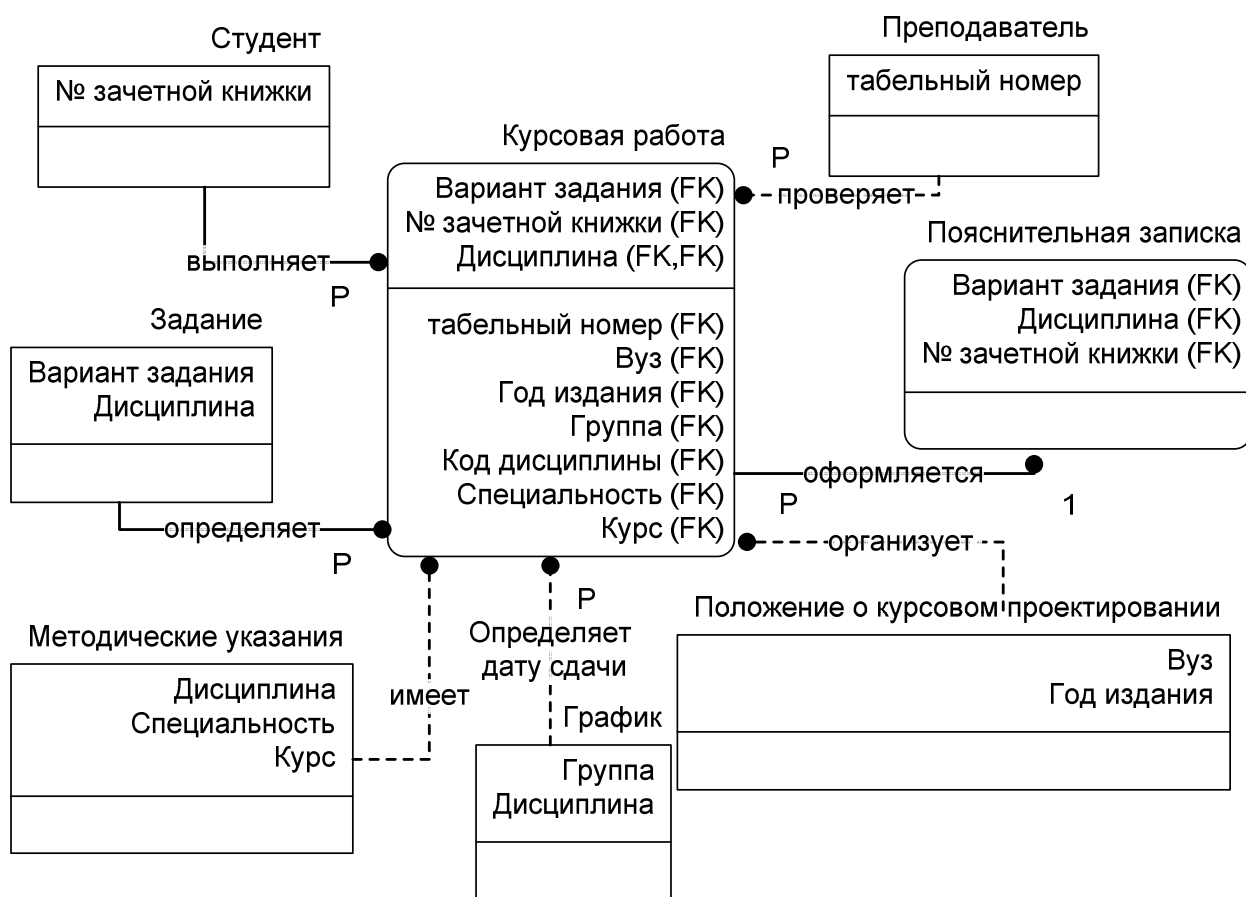


Рисунок 5.9 – Информационная модель с ключевыми атрибутами

### Упражнение 3. Создание полной атрибутивной модели

Для того чтобы получить полную атрибутивную модель, необходимо дополнить сущности не ключевыми атрибутами. Дополненная модель представлена на рис. 5.11.

Примечание. Если атрибут не является обязательным, нужно убедиться, что в окне *Свойства базы данных* в категории *Столбцы* в пункте *Обязательное* не стоит галочка. Не обязательные к заполнению атрибуты справа от имени имеют пометку (O).



Рисунок 5.10 – Скорректированная информационная модель, основанная на ключах

#### Упражнение 4. Нормализация полной атрибутивной модели

1. 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 *Методические указания* 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 *Методические указания* и *Преподаватель*, предварительно удалив атрибут *Авторы*. Остальные атрибуты соответствуют 1 НФ. Атрибутивная модель, приведенная к 1 НФ, представлена на рис. 5.12.

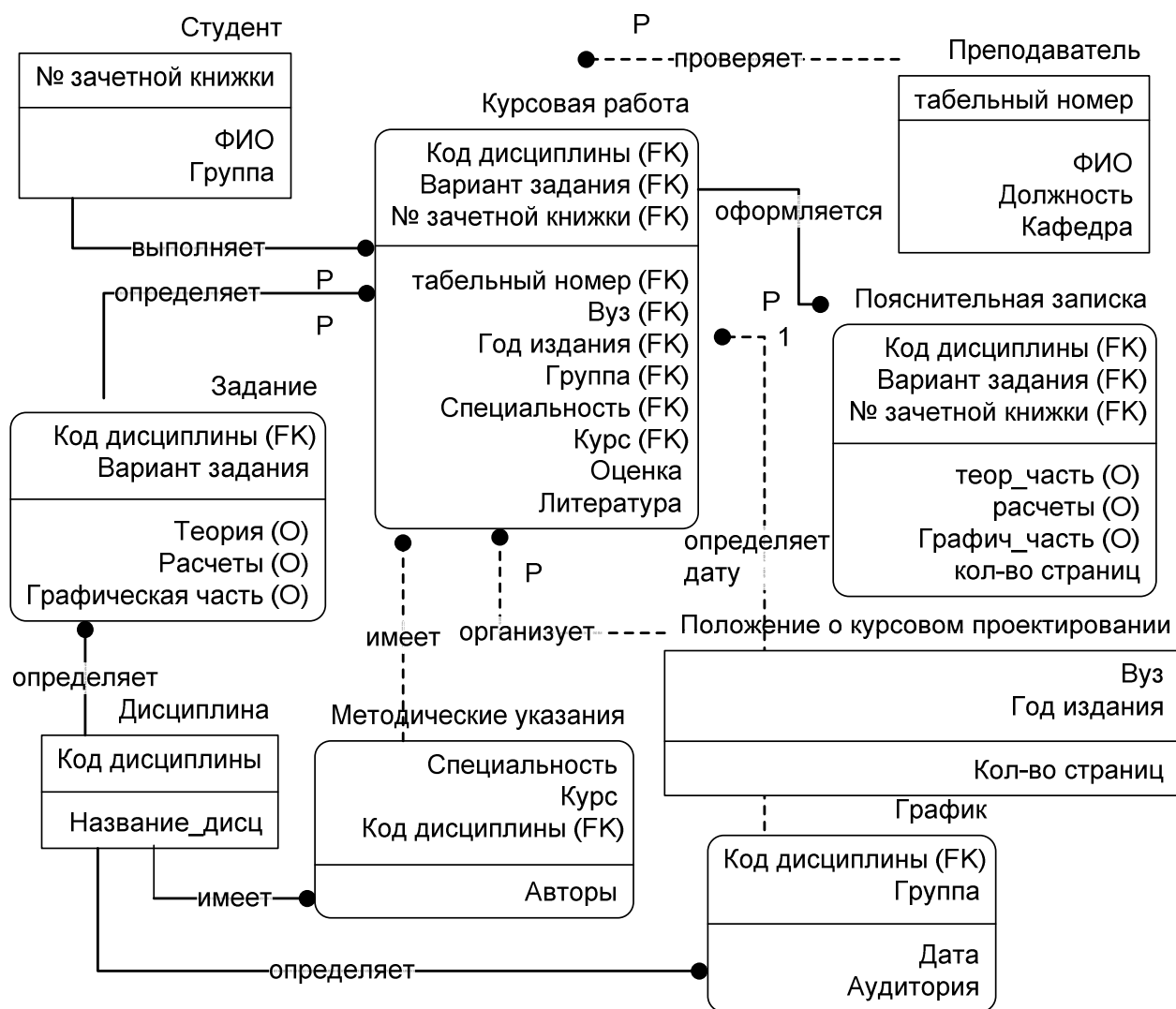


Рисунок 5.11 – Полная атрибутивная модель

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

3. Проверим, есть ли транзитивная зависимость между не ключевыми атрибутами. Проверка показала, что взаимозависимости между не ключевыми атрибутами нет. Таким образом, модель, представленная на рисунке 5.12, приведена к 3 НФ.

Примечание. К нормализации относились также действия, выполненные в п. 2 упражнения 2.



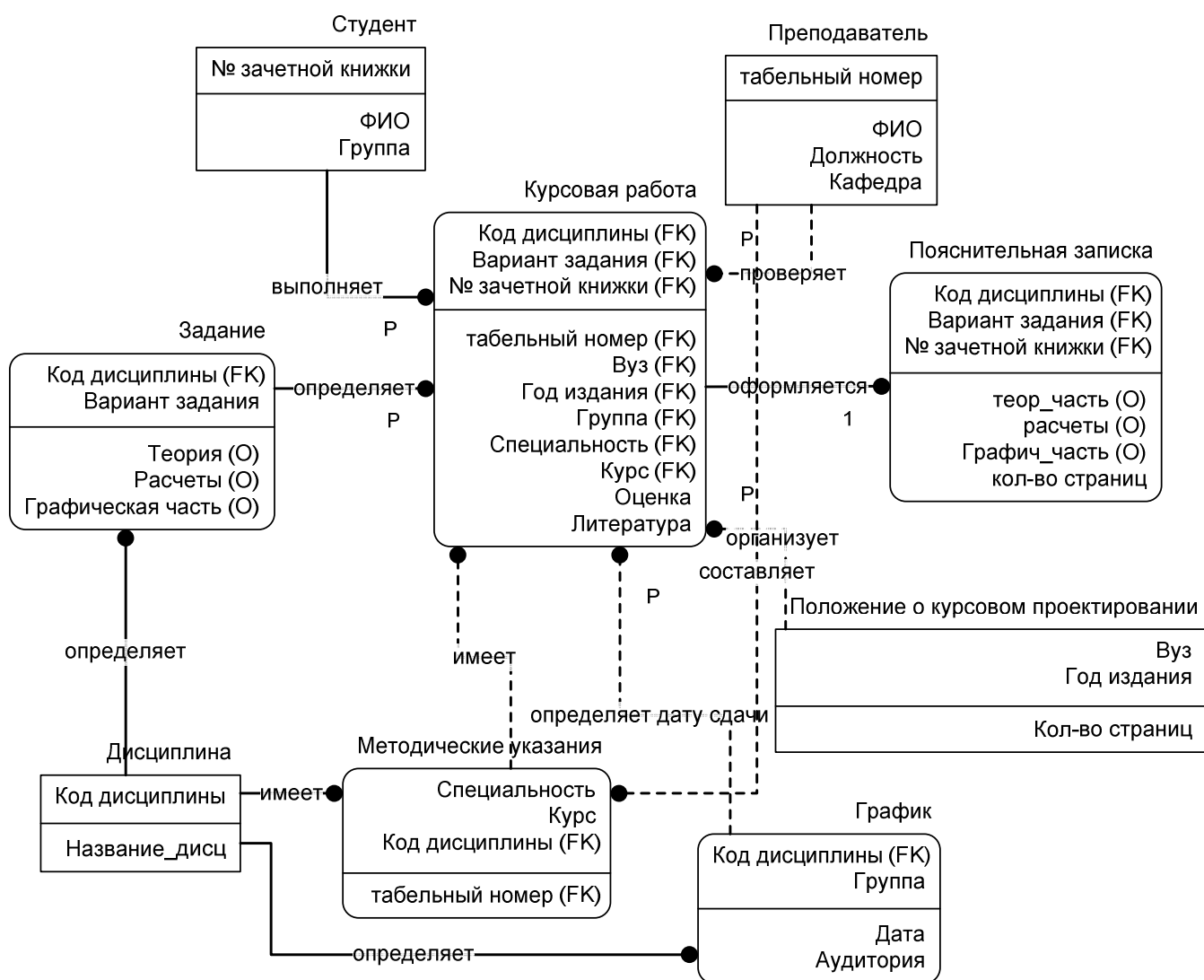


Рисунок 5.12 – Информационная модель, приведенная к 1 НФ

## Упражнение 5. Создание физической модели

1. Необходимо переключиться на физический уровень представления информационной модели. Для этого нужно выбрать пункты меню *База данных* → *Параметры* → *Документ*. В открывшемся окне на вкладке *Общие* установить переключатель в меню *Имена, видимые на схеме*. В данном случае для физического представления информационной модели необходимо выбрать пункт *Физические имена* (рис. 5.13).

2. В закладке *Таблица* окна *Параметры документа базы данных* в меню *Отображать* выбрать пункт *Вертикальные линии*, в меню *Типы данных* – *Показывать физические* и в меню *Порядок* – *Физический порядок* (рис. 5.14).

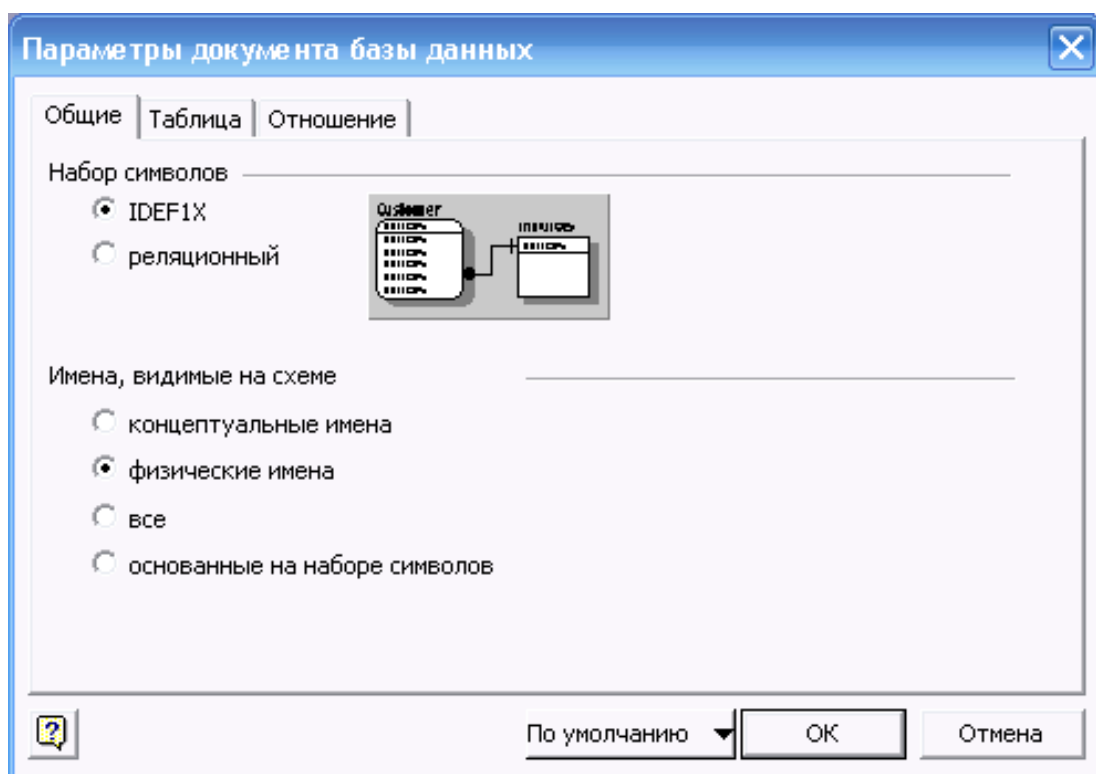


Рисунок 5.13 – Настройка параметров модели

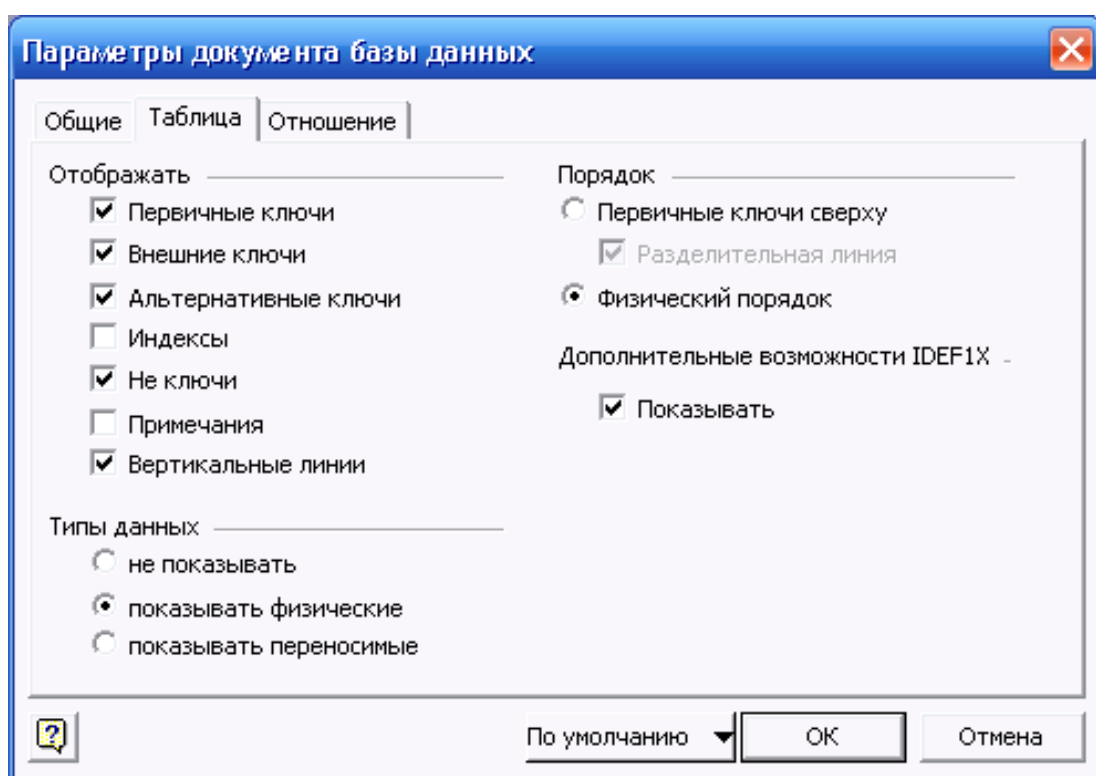


Рисунок 5.14 – Настройка параметров отображения сущности

3. В закладке *Отношение* окна *Параметры документа базы данных* в меню *Отображение вида* выбрать пункт *Показывать физическое имя* (рис. 5.15).

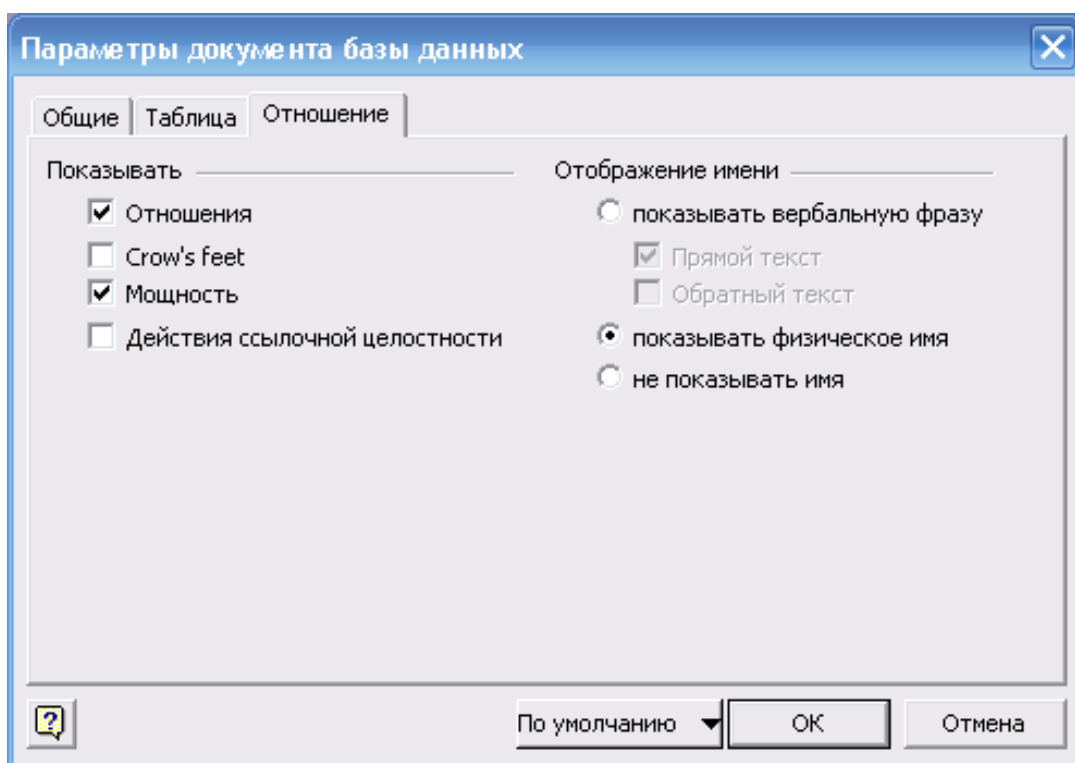


Рисунок 5.15 – Настройка вида отношений информационной модели

По окончании настройки документа информационная модель будет выглядеть, как представленная на рис. 5.16.

4. Для каждого атрибута (поля) необходимо определить тип данных.

**Примечание** (*Выбор между переносимыми и физическими типами данных*).

*Переносимые типы данных* — это обобщенные типы данных, соответствующие в разных системах баз данных простым, совместимым между собой физическим типам.

*Физические типы данных* — это типы данных, поддерживаемые целевой базой данных.

Щелкните сущность, содержащую атрибуты, для которых требуется установить типы данных.

В окне *Свойства базы данных* в списке *Категории* выберите вариант *Столбцы*.

Под списком столбцов установите переключатель в положение *Физический тип данных*.



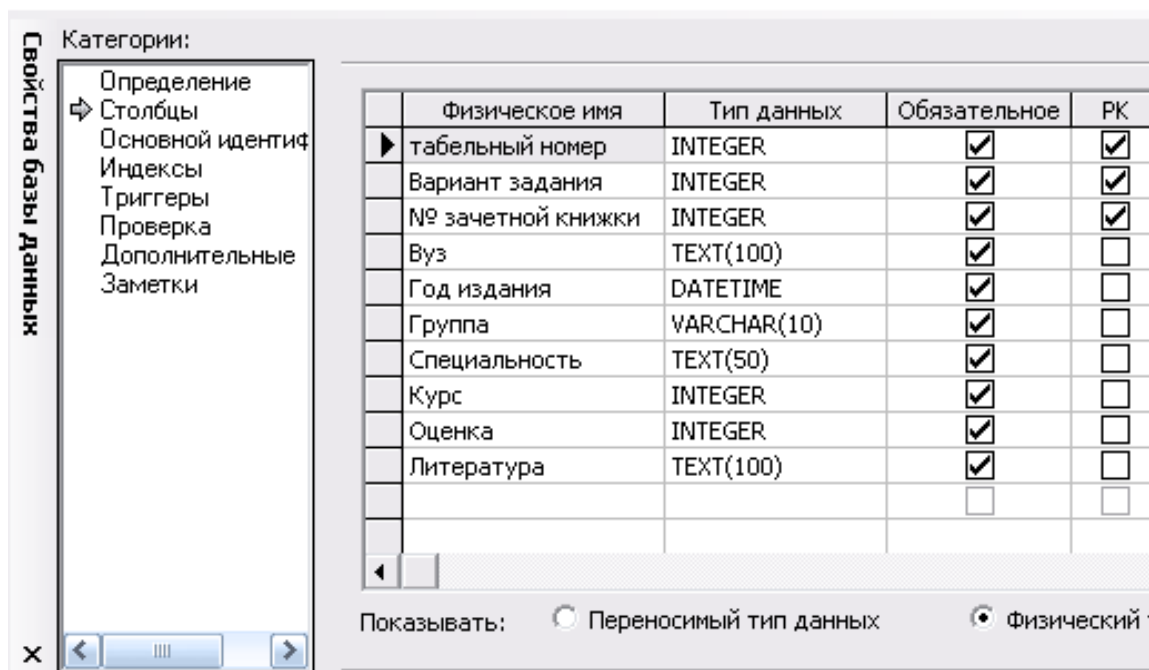


Рисунок 5.17 – Определение типа данных атрибутов сущности

После того, как будут выполнены все действия, физическая модель будет выглядеть, как показано на рис. 5.18.

Таким образом, проделав все вышеперечисленные действия, получим информационную модель физического уровня, на основе которой может быть сгенерирована схема БД (в нашем случае в MS Office Access).

## 6. Задание

В соответствии с вариантом задания, определенным преподавателем, последовательно выполнить следующие действия:

- 1) создать информационную модель логического уровня (выполнить упражнения 1 – 3). Минимальное количество сущностей – 4 (в зависимости от предметной области);
- 2) провести нормализацию полученной модели (упражнение 4);
- 3) на основе нормализованной логической модели построить информационную модель физического уровня (упражнение 5).

## 7. Порядок выполнения работы

Для выполнения работы необходимо:

- а) повторить правила техники безопасности при работе с вычислительной техникой;
- б) изучить соответствующий раздел лекционного курса, а также теоретическую часть настоящего методического указания;

- в) выполнить лабораторную работу согласно описанной в пункте 5 методике в соответствии с вариантом задания;
- г) в соответствии с требованиями, приведенными в разделе 8 практикума, оформить отчет по лабораторной работе;
- е) ответить на контрольные вопросы.

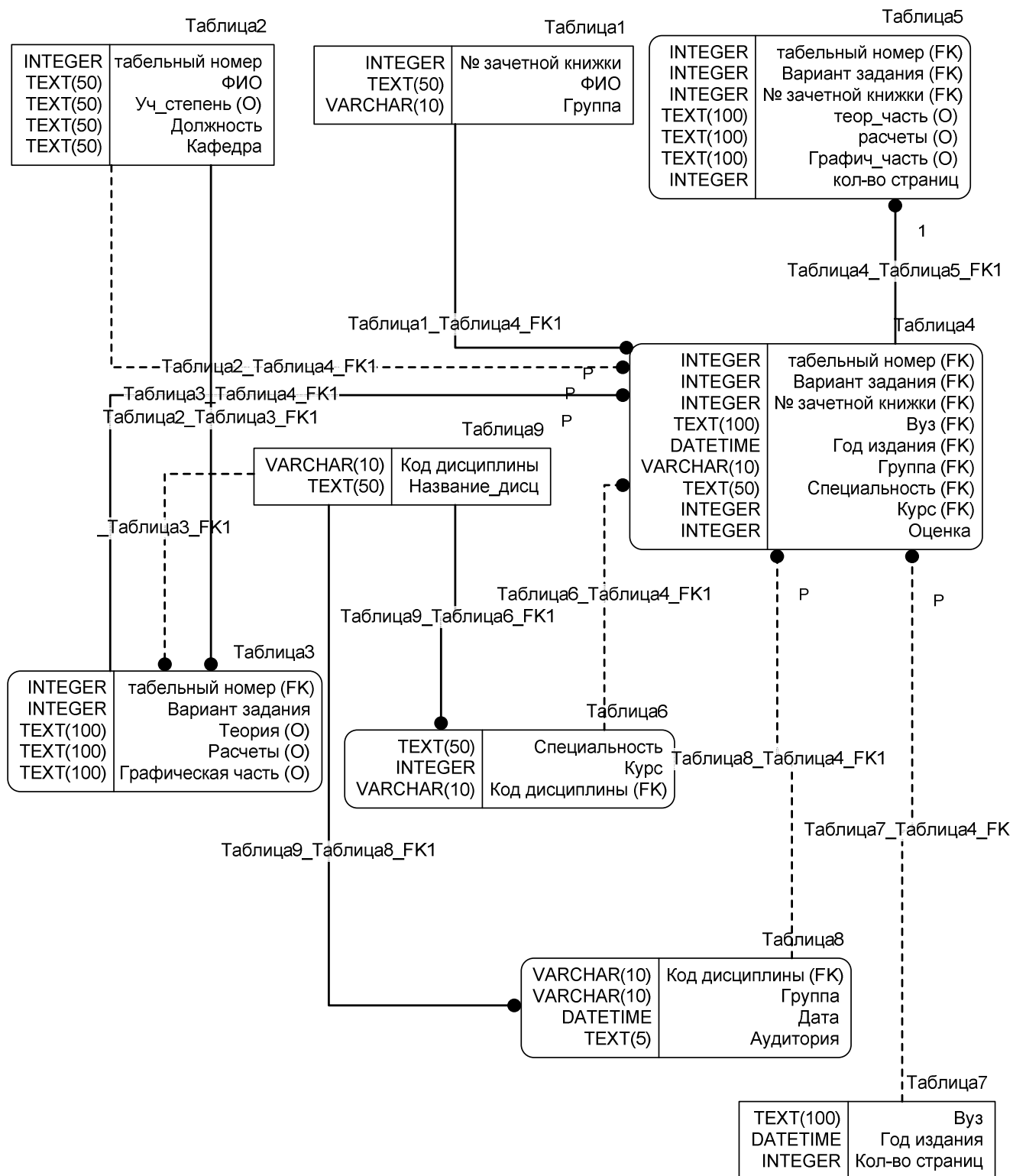


Рисунок 5.18 – Физическая модель базы данных

## **8. Критерии результативности работы**

Лабораторная работа считается выполненной в том случае, если:

1. Студент выполнил все задания в соответствии с представленной методикой;
2. Результаты выполнения работы, представленные в виде отчета, соответствуют предъявленным к ним требованиям;
3. Студент правильно ответил на все контрольные вопросы и выполнил контрольные задания.

## **9. Требования к содержанию и оформлению отчета**

Отчет по лабораторной работе должен быть оформлен в форме документа MS Office Word и содержать:

- 1) титульный лист;
- 2) название лабораторной работы, цель;
- 3) пул – список потенциальных сущностей;
- 3) нормализованную информационную модель логического уровня;
- 4) информационную модель физического уровня;
- 5) выводы по проделанной работе.

Отчет должен быть оформлен в соответствии с требованиями СТП УГАТУ. *Титульный лист* должен содержать следующие сведения: название и порядковый номер лабораторной работы, вариант задания, ФИО студентов, группу, ФИО преподавателя и т.п. По усмотрению преподавателя отчет может быть представлен в бумажном или электронном виде.

## **10. Контрольные вопросы**

1. Для чего предназначена диаграмма «сущность-связь»?
2. Чем отличается полная атрибутивная модель от диаграммы «сущность-связь»?
3. Какие виды отношений существуют и чем они отличаются?
4. Приведите пример идентифицирующего отношения.
5. Приведите пример отношения полной категоризации.
6. Чем отличаются отношения полной и неполной категоризации?
7. Что представляет собой нормализация?
8. В чем разница между логическим уровнем модели данных и физическим?

## СПИСОК ЛИТЕРАТУРЫ

1. Вендров А. М. Практикум по проектированию программного обеспечения экономических информационных систем. : – М.: Финансы и статистика, 2006.-190 с.

2. Вендров А. М. Проектирование программного обеспечения экономических информационных систем – М.: Финансы и статистика, 2006.-543 с.

3. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – М.: ДИАЛОГ-МИФИ, 2005 – 432с.



## Функциональная модель процесса «Выполнить курсовую работу»

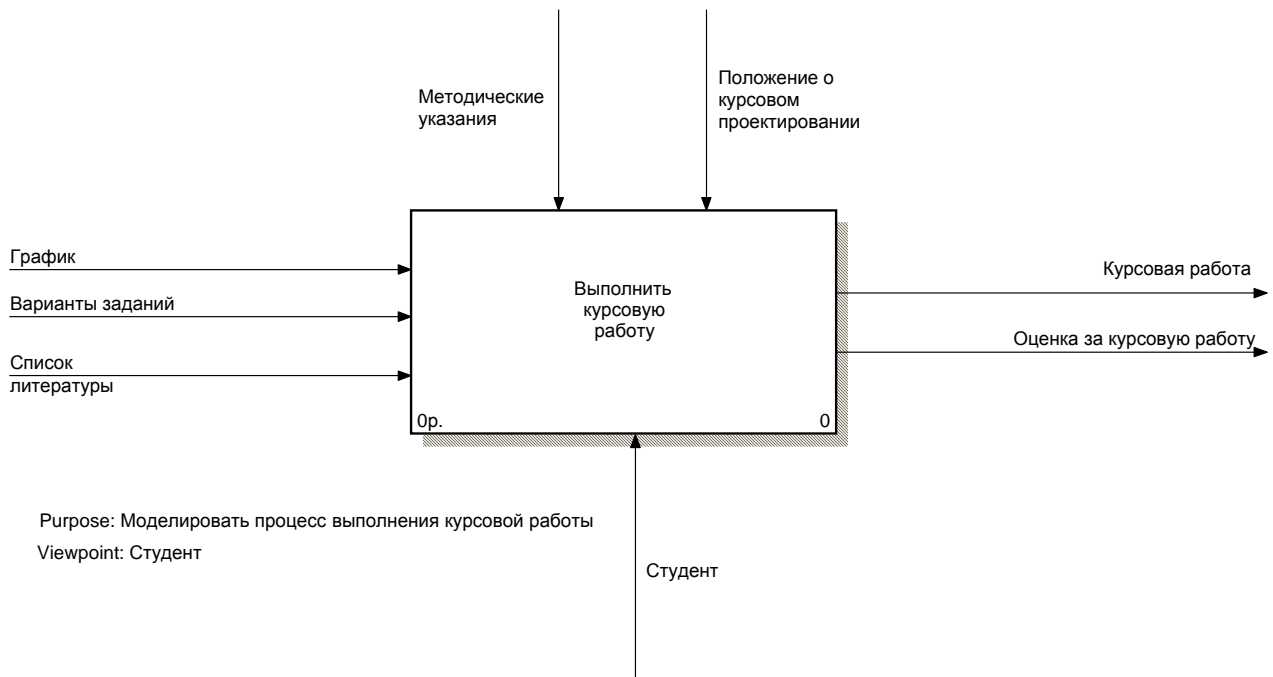


Рисунок А.1 – Контекстная диаграмма процесса выполнения курсовой работы

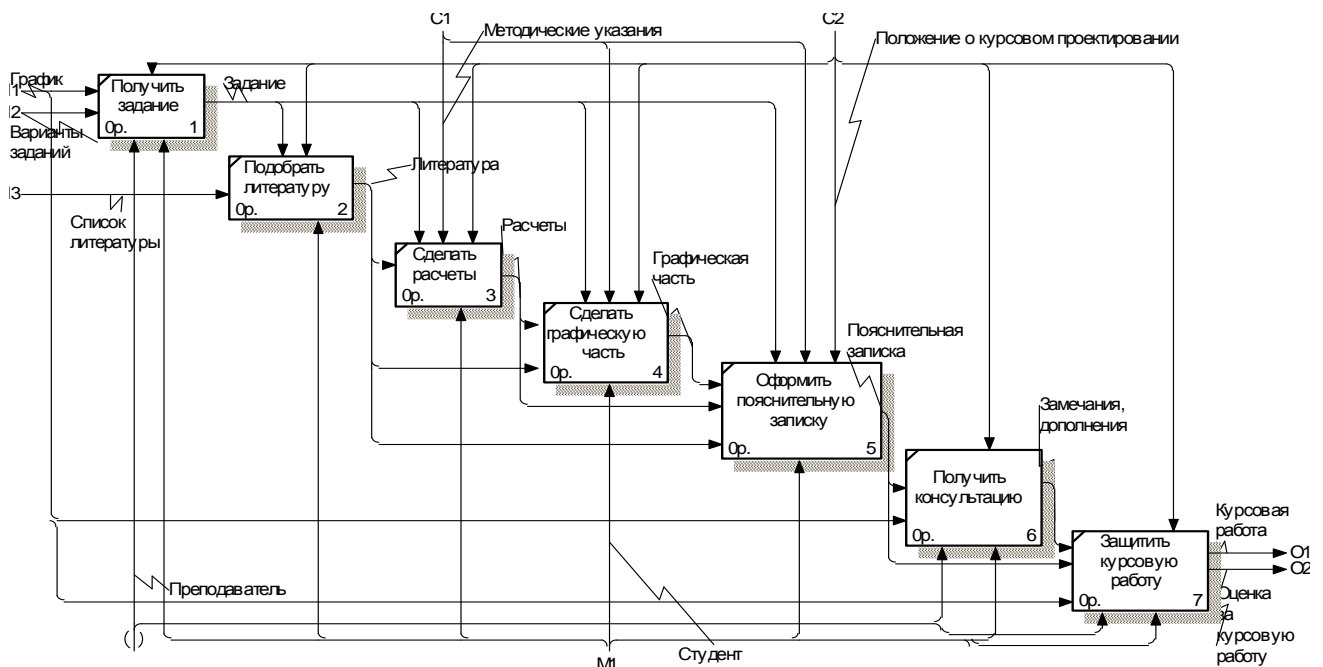


Рисунок А.2 – Диаграмма декомпозиция блока А0

## Приложение Б

### Некоторые типы данных Microsoft Office Access

#### 1. Символьные типы

Таблица Б1 - Перечень символьных типов

Тип данных	Назначение	Размер
CHAR	Строковый тип	до 32767 байт (по умолчанию 1 байт)
VARCHAR	То же, что и CHAR	
LONG VARCHAR	Символьный тип произвольной длины	Длина произвольная, ограничена максимальным размером файлов базы данных
TEXT	То же, что и LONG VARCHAR	

#### 2. Числовые типы

Таблица Б2 - Перечень числовых типов

Тип данных	Диапазон значений	Точность - число знаков после запятой	Размер
INTEGER	от -2 147 483 648 до +2 147 483 647	0	4 байта
SMALLINT	от -32 768 до +32 767	0	2 байта
REAL	от -3.4 е-38 до 3.4 е+38	до 6	4 байта
DOUBLE	от -1.797 е-308 до +1.797 е+308	до 15	8 байт
DECIMAL	числа, состоящие из N цифр с M цифрами в дробной части.	M	сколько требуется
NUMERIC	То же, что и DECIMAL		

### 3. Типы дата/время

Таблица Б3 - Форматы представления данных типа дата/время, определяемые по умолчанию

Тип данных	Формат, используемый по умолчанию
DATETIME	'YYYY-MM-DD HH : NN : ss.SSS'

- **YYYY** - четыре цифры, обозначающие год:
- **MM** - две цифры, обозначающие месяц:
- **DD** - две цифры, обозначающие день:
- **HH** - две цифры, обозначающие часы:
- **NN** - две цифры, обозначающие минуты:
- **ss** - две цифры, обозначающие секунды:
- **SSS** - три цифры, обозначающие доли секунд.

### 4. Двоичные типы

Двоичные типы предназначены для представления двоичных данных, включая изображения и другую информацию, не обрабатываемую собственными средствами СУБД.

Таблица Б4 - Двоичные типы

Тип данных	Назначение	Размер
BIT	Тип для представления значений 0 и 1	1 байт
BINARY	То же, что и CHAR, за исключением операций сравнения	до 32767 байт
LONG	Тип для представления двоичных данных произвольной длины	Длина произвольная (ограничена максимальным размером файлов базы данных)