Блок 2. Проектирование реляционных баз данных

Тема 7 Проектирование реляционных баз данных на основе принципов нормализации: первые шаги нормализации

Тема 8 Проектирование реляционных баз данных на основе принципов нормализации: дальнейшая нормализация

Тема 9 Проектирование реляционных баз данных с использованием семантических моделей: ER-диаграммы

Проектирование баз данных — процесс создания схемы базы данных и определения необходимых ограничений целостности.

Задача проектирования реляционной базы данных формулируется следующим образом: как в некоторой базе данных для заданного набора данных выбрать подходящую структуру?

При проектировании базы данных решаются **две основные проблемы**.

1. Каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не противоречило семантике предметной области и было, по возможности, лучшим (эффективным, удобным и т. д.)? Часто эту проблему называют **проблемой логического проектирования баз данных**.

2. Как обеспечить эффективность выполнения запросов к базе данных, т. е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создания каких дополнительных структур (например, индексов) потребовать и т. д.? Эту проблему обычно называют **проблемой физического проектирования баз данных**.

В случае реляционных баз данных трудно предложить какие-либо общие рекомендации по части физического проектирования, так как многое зависит от выбранной СУБД. Поэтому чаще всего ограничиваются вопросами логического проектирования реляционных баз данных, которые существенны при использовании любой реляционной СУБД.

Нормализация баз данных

Нормализация – это формальный метод анализа отношений на основе их первичного ключа и существующих функциональных зависимостей.

Процесс нормализации – это разбиение таблицы на две или более с целью ликвидации дублирования данных и потенциальной их противоречивости.

Цель нормализации – получение такого проекта БД, в котором каждый факт хранится в одном месте, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных из-за их избыточности.

Нормальные формы

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- первая нормальная форма (1NF) ;
- вторая нормальная форма (2NF) ;
- третья нормальная форма (3NF) ;
- нормальная форма Бойса-Кодда (BCNF) ;
- четвертая нормальная форма (4NF);
- пятая нормальная форма, или нормальная форма проекции-соединения (5NF или PJ/NF).

Основные **свойства** нормальных форм состоят в следующем:

- каждая следующая нормальная форма в некотором смысле лучше предыдущей нормальной формы;
- при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

В основе процесса проектирования лежит метод нормализации, т. е. декомпозиции отношения, находящегося в предыдущей нормальной форме, на два или более отношений, которые удовлетворяют требованиям следующей нормальной формы.

Первая нормальная форма - 1НФ

База данных находится **в первой нормальной форме**, если все ее таблицы являются отношениями, а столбцы таблиц удовлетворяют условию атомарности (значения в домене не являются ни списками, ни множествами и т.д.) и отсутствует избыточность данных.

1 НФ. ПРИМЕР

В задачах часто встречаются данные, имеющие более одного значения. Примером может служить заказ на изготовление каких-либо изделий, в который вошли названия изделий, их цены и заказанное количество:

Сущность «ЗАКАЗ» с атрибутами, полностью отражающими приведенную таблицу, хотя частично и находится в первой нормальной форме, т.е. нет неатомарных значений, но обладает двумя недостатками:

1. Если заказчику требуется только один или два вида изделий, в таблице окажутся пустые ячейки, что понижает коэффициент использования памяти. Для четвертого вида изделий место в заказе просто отсутствует, т.е. ему придется оформлять второй заказ. Значит, в таблице появится лишняя строка и новые пустые ячейки.
2. Поиск изделий с конкретным названием усложнен, так как его приходится искать в нескольких столбцах таблицы.

ЗАКАЗ

№	Заказчик	Изделие 1			Изделие 2			Изделие 3			Дата заказа
		Название	Цена	Кол-во	Название	Цена	Кол-во	Название	Цена	Кол-во	

Эти недостатки можно устранить одним из двух способов:

1. Разбить строку таблицы на несколько строк, в каждой из которых будет стоять только одна группа повторяющихся атрибутов (в рассмотренном примере это Название изделия, Цена, Количество).
2. Перевести повторяющиеся атрибуты в новую сущность, назначить ей первичный ключ (Код изделия) и связать с исходной сущностью ссылкой на первичный ключ последней (Номер заказа).

Итак, была сущность:

$$\overrightarrow{Заказ} = \langle Номер\ заказа\ ,\ Заказчик\ ,\ Дата\ заказа\ ,\ Назв.\ изделия1\ ,\ Цена1\ ,\ Кол\text{-}во1\ ,\ Назв.\ изделия\ 2\ ,\ Цена2\ ,\ Кол\text{-}во2\ ,\ Назв.\ изделия3\ ,\ Цена3\ ,\ Кол\text{-}во3\ \rangle$$

При использовании первого способа ее экземпляр будет иметь атрибуты:

$$\overrightarrow{Заказ} = \langle Номер\ заказа\ ,\ Заказчик\ ,\ Дата\ заказа\ ,\ Номер\ строки\ заказа\ ,\ Назв.\ изделия1\ ,\ Цена1\ ,\ Кол\text{-}во1\ \rangle$$

В результате этого преобразования увеличилось количество строк, зато повысилась плотность записи, и упростился поиск изделий. После выделения новой сущности (второй способ преобразования) избыточность информации устранена:

$$\overrightarrow{Заказ} = \langle Номер\ заказа\ ,\ Заказчик\ ,\ Дата\ заказа\ \rangle$$

$$\overrightarrow{Заказанное\ изделие} = \langle Номер\ заказа\ ,\ Код\ изделия\ ,\ Название\ изделия\ ,\ Цена\ ,\ Кол-во\ \rangle$$

Вторая нормальная форма - 2НФ

Вторая нормальная форма к требованию 1НФ добавляет еще одно: каждый неключевой атрибут должен функционально полно зависеть от первичного ключа (не должен зависеть от части составного ключа).

2 НФ. ПРИМЕР

Состав созданной нами в предыдущем примере сущности «ЗАКАЗАННОЕ ИЗДЕЛИЕ» служит характерным примером нарушения второй нормальной формы.

Первичный ключ этой сущности - сочетание атрибутов Номер заказа и Код изделия. От этого составного ключа зависит один атрибут - Количество заказанных изделий, остальные атрибуты - Название и Цена изделия зависят только от Кода изделия.

Чтобы привести сущность «ЗАКАЗАННОЕ ИЗДЕЛИЕ» ко второй нормальной форме, выделим из нее атрибуты, характеризующие изделие как таковое, создав еще одну сущность – «ИЗДЕЛИЕ» и будем ссылаться на нее из «ЗАКАЗАННОГО ИЗДЕЛИЯ» через Код изделия:

$$\overrightarrow{Заказанное\ изделие} = \langle Номер\ заказа\ ,\ Код\ изделия\ ,\ Кол-во\ \rangle$$

$$\overrightarrow{Изделие} = \langle Код\ изделия\ ,\ Название\ изделия\ ,\ Цена\ \rangle$$

Другой пример сущности, в которой часть неключевых атрибутов не зависит от первичного ключа:

$$\overrightarrow{Дисциплина} = \langle Код\ дисциплины\ ,\ Название\ дисциплины\ ,\ Код\ лектора\ ,\ Фамилия\ лектора\ ,\ Имя\ лектора\ ,\ Отчество\ лектора\ ,\ Содержание\ дисциплины\ \rangle$$

- При исключении дисциплины из планов обучения, вместе с нею пропадает и информация о лекторе.
- Если какой-либо лектор временно прекращает читать свою дисциплину, информация о нем также пропадает.

Правильным решением в этом случае будет выделение информации о лекторе в отдельную сущность со ссылкой на код дисциплины:

$$\overrightarrow{Дисциплина} = \langle Код\ дисциплины\ ,\ Название\ дисциплины\ ,\ Содержание\ дисциплины\ \rangle$$

$$\overrightarrow{Лектор} = \langle Код\ лектора\ ,\ Фамилия\ лектора\ ,\ Имя\ лектора\ ,\ Отчество\ лектора\ \rangle$$

$$\overrightarrow{Лектор\_Дисциплина} = \langle Код\ лектора\ ,\ Код\ дисциплины\ \rangle$$

Третья нормальная форма - 3НФ

Третья нормальная форма удовлетворяет определению 2НФ и ни один из ее неключевых атрибутов не связан функциональной зависимостью с любым другим неключевым атрибутом.

3 НФ. ПРИМЕР

В качестве примера рассмотрим сущность Сотрудник:

$$\overrightarrow{Сотрудник} = (Табельный\ номер, Фамилия, Имя, Отчество, Название\ должности, Оклад, Дата\ зачисления, Дата\ увольнения)$$

Здесь неключевые атрибуты Н*азвание должности* и *Оклад* находятся в транзитивной зависимости.

В чем опасность такой зависимости?

Во-первых, несколько человек могут работать в одной и той же должности. При изменении должностного оклада в этом случае нужно будет менять данные в каждой записи, содержащей эту должность.

Во-вторых, в организации могут оказаться уникальные должности, например, начальник, и тогда при увольнении этого сотрудника (удалении записи) потеряется информация об окладе для этой должности.

В рассмотренной ситуации нужно создать новую сущность «ДОЛЖНОСТЬ» с находящимися в транзитивной зависимости атрибутами - *Название должности* и *Оклад* и сделать ссылку от «СОТРУДНИКА» на «ДОЛЖНОСТЬ»:

$$\overrightarrow{Сотрудник} = \langle Табельный\ номер\ ,\ Фамилия\ ,\ Имя\ ,\ Отчество\ ,\ Код\ должности\ ,\ Дата\ зачисления\ ,\ Дата\ увольнения\ \rangle$$

$$\overrightarrow{Должность} = \langle Код\ должности\ ,\ Название\ должности\ ,\ Оклад\ \rangle$$

В реальных условиях сотрудник может переходить с одной должности на другую и может одновременно занимать несколько должностей в своей организации. В связи с этим нужно создать еще одну сущность, которая будет фиксировать должностные перемещения сотрудников, перенеся в нее атрибуты *Код должности, Дата зачисления* и *Дата увольнения,* и ссылаться на сущность «СОТРУДНИК» через *Табельный номер:*

$$\overrightarrow{Сотрудник} = \langle Табельный\ номер\ ,\ Фамилия\ ,\ Имя\ ,\ Отчество\ \rangle$$

$$\overrightarrow{Должность} = \langle Код\ должности\ ,\ Название\ должности\ ,\ Оклад\ \rangle$$

$$\overrightarrow{Должностные\ перемещения} = \langle Табельный\ номер\ ,\ Код\ должности\ ,\ Дата\ зачисления\ ,\ Дата\ увольнения\ \rangle$$

Вывод
по нормализации базы данных

Таким образом, таблица находится в **первой нормальной форме (1НФ)** тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таким образом, таблица находится во **второй нормальной форме (2НФ)**, если она удовлетворяет определению 1НФ и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом.

Таким образом, таблица находится в **третьей нормальной форме (3НФ),** если она удовлетворяет определению 2НФ и не одно из ее неключевых полей не зависит функционально от любого другого неключевого поля.

Этапы проектирования баз данных

Проектирование баз данных – это трудоемкий и ответственный процесс, который целесообразно разбить на стадии:

1. системный анализ;
2. концептуальное проектирование;
3. логическое проектирование;
4. физическое проектирование.

1. Системный анализ

Цель системного анализа предметной области как этапа проектирования – выделить предметную область как систему объектов и их взаимосвязей, определив при этом функционально-информационные требования к их последующему представлению в виде **системы взаимосвязанных данных**.

Существует два подхода к определению состава и структуры предметной области.

Функциональный подход предполагает, что проектирование начинается с анализа задач и, соответственно, функций, обеспечивающих реализацию информационных потребностей.

При **объектном** (предметом) подходе информационные потребности пользователей (задачи) жестко не фиксируются, а основное внимание сосредотачивается на выделении существенных объектов - предметов и связей, информация о которых может быть использована в прикладных задачах пользователей.

Главным результатом этапа системного анализа является определение **парадигмы информационной (инфологической) модели**, т.е. определение структурированности и динамичности информации, способа представления и характера использования информации.

2. Концептуальное проектирование

Следующей стадией проектирования системы баз данных является построение **семантической модели** предметной области, которая базируется на анализе свойств и природы объектов предметной области и информационных потребностей будущих пользователей разрабатываемой системы.

Эту стадию принято называть **концептуальным** или **инфологическим** проектированием системы, а ее результат – **концептуальной** или **инфологической** моделью предметной области.

Наиболее известным описанием объектов и связей между ними является **модель «сущность-связь»**, включающая:

- **систему атрибутов и средств описания предметной области**, например, логические связи между показателями или лингвистические свойства языка (семантику, синтаксис и т.д.);
- **ограничения целостности**, определяющие допустимость отдельных полей и взаимосвязей как на уровне семантики содержимого БД, так и ее физической структуры (отдельных файлов данных и взаимосвязей между ними);
- **описание информационных потребностей пользователей**, например, в виде типовых запросов, отражающих процедурные особенности обращения к данным.

3. Логическое проектирование

Задачей этой стадии проектирования системы базы данных является выбор подходящей СУБД и отображение в ее среду (структуру данных) спецификаций инфологической модели предметной области.

Другими словами, модель предметной области разрабатываемой системы должна быть представлена в терминах модели данных концептуального уровня выбранной конкретной СУБД. Эту стадию называют **логическим (или даталогическим)** проектированием базы данных, а ее результатом является **концептуальная схема базы данных (или *даталогичекая модель*)**, включающая определение всех информационных элементов и связей, в том числе задание типов, характеристик и имен.

4. Физическое проектирование

Стадия физического проектирования базы данных в общем случае включает:

- выбор способа организации базы данных;
- разработку спецификации внутренней схемы средствами модели данных ее внутреннего уровня;
- описание отображения концептуальной схемы во внутреннюю схему.

Модель «Сущность - связь»

Одной из наиболее популярных средств формализованного представления предметной области систем, ориентированных на обработку фактографической информации, является модель «сущность-связь».

Модель «Сущность-связь» - это неформальная модель предметной области, которая используется на этапе концептуального проектирования базы данных. Эта модель позволяет описывать объекты предметной области и их взаимоотношения. Основным достоинством модели является ее относительная простота и понятность за счет применения естественного языка, что позволяет использовать модель в качестве инструмента общения разработчика и будущего пользователя при сборе информации о базе данных предметной области.

Основное **назначение** модели «сущность-связь» - семантическое описание предметной области и представление информации для обоснования выбора видов моделей и структур данных, которые будут использованы в системе в дальнейшем.

Основные понятия модели «Сущность - связь»

Основным принципом, используемым при построении модели «сущность-связь», является использование трех конструктивных элементов для представления составляющих предметной области - **сущность, атрибут, связь.**

Сущность – это объект, который может быть идентифицирован неким способом, отличающим его от других объектов и о котором в системе будет накапливаться информация.

Набор сущностей - множество сущностей одного типа, обладающих одинаковыми свойствами.

Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств.

Атрибут - это поименованная характеристика сущности, которая принимает значения из некоторого допустимого множества.

Связь - это поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связи выступают в модели в качестве средства, с помощью которого представляются отношения между сущностями, имеющими место в предметной области.

Одна из участвующих в связи сущностей - **независимая**, называется родительской сущностью, другая - **зависимая**, называется дочерней или сущностью-потомком.

Типы связей

- Отображение 1:1 или связь «*один к одному*». С помощью отображения 1:1 определяют такой тип связи между сущностями A и B, когда каждому экземпляру сущности типа A соответствует один и только один экземпляр сущности B и, наоборот, каждому экземпляру сущности B соответствует один и только один экземпляр сущности A.
- Отображение 1:M или связь *«один ко многим».* С помощью этого отображения определяется тип связи между сущностями A и B, при котором одному экземпляру сущности A может соответствовать 0, 1 или несколько экземпляров сущности B, однако каждому экземпляру сущности B соответствует только один экземпляр сущности A.
- Отображение M:1 или связь «*многие к одному*» является обратным отображению 1:M.
- Отображение M:N или связь *«многие ко многим».* С помощью этого отображения определяется тип связи между сущностями A и B, при котором каждому экземпляру сущности A может соответствовать 0, 1 или несколько экземпляров сущности B. С одним экземпляром сущности A может быть связано либо несколько экземпляров сущности B, либо один, либо ни одного. И, наоборот, с одним экземпляром сущности B также может быть связано либо несколько экземпляров сущности A, либо один, либо ни одного, т.е. идентификация экземпляров сущностей неуникальна в обоих направлениях.

Этап концептуального проектирования начинается с моделирования предметной области. Проектировщик разбивает её на ряд **локальных областей**, каждая из которых включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи (подзадачи). Каждое **локальное представление** моделируется отдельно, затем они объединяются.

Выбор локального представления зависит от масштабов предметной области. Обычно она разбивается на локальные области таким образом, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6-7 сущностей. После того, как созданы локальные представления, выполняется их объединение.

На этапе объединения необходимо выявить и устранить все противоречия. Например, одинаковые названия семантически различных объектов или связей или несогласованные ограничения целостности на одни и те же атрибуты в разных приложениях. Устранение противоречий вызывает необходимость возврата к этапу моделирования локальных представлений с целью внесения в них соответствующих изменений.

По завершении объединения результаты проектирования являют собой **инфологическую модель предметной области**.

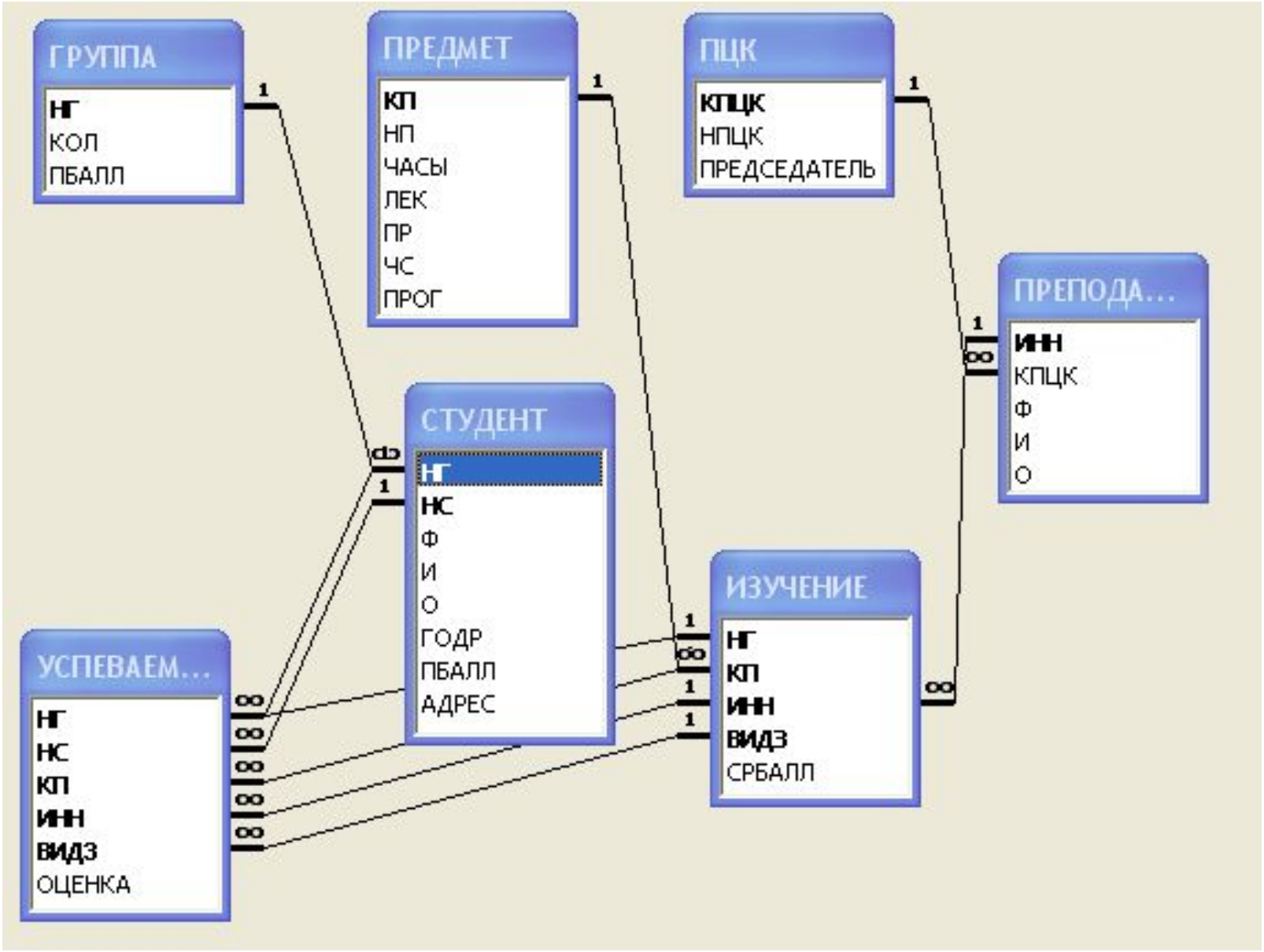


Рисунок 2.1 – Схема реляционной модели данных.

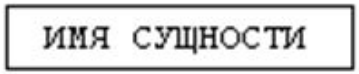
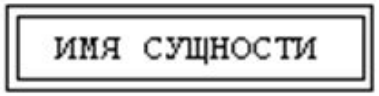
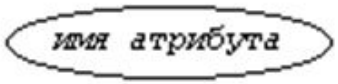
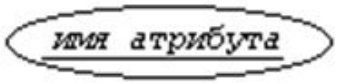
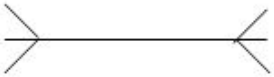
ER- диаграмма

Как отмечалось ранее, одна из основных целей семантического моделирования состоит в том, чтобы результаты анализа предметной области были отражены в достаточно простом, наглядном, но в то же время формализованном и достаточно информативном виде.

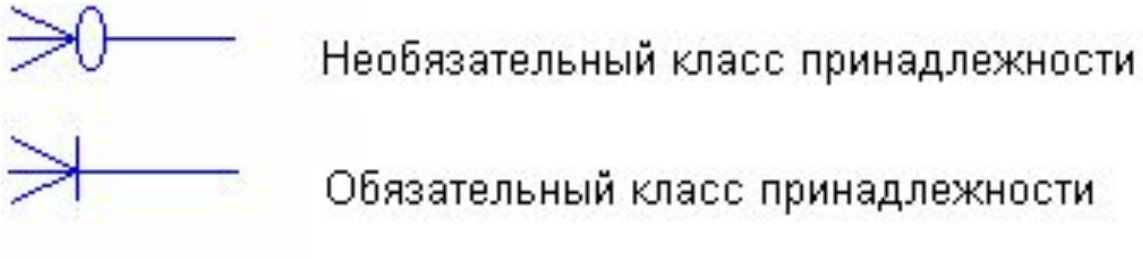
В этом случае моделирование предметной области базируется на использовании графических диаграмм, которые получили название ER-диаграммы. В ER-диаграмме сочетаются функциональный и информационный подходы, что позволяет представлять как совокупность выполняемых функций, так и отношения между элементами системы, задаваемые структурами данных.

Графическая форма позволяет отобразить в компактном виде (за счет наглядных условных обозначений) типологию и свойства сущностей и связей, а формализмы, положенные в основу ER-диаграмм, позволяют использовать на следующем шаге проектирования логической структуры базы данных строгий аппарат нормализации.

При построении ER-диаграммы используются следующие обозначения:

Обозначение	Значение
	Набор независимых сущностей
	Набор зависимых сущностей
	Атрибут
	Ключевой атрибут
	Набор связей
	Тип связи «один-к-одному»
	Тип связи «один-ко-многим»
	Тип связи «многие-ко-многим»

Важной характеристикой связи в диаграмме помимо ее типа является **класс принадлежности** входящих в нее сущностей.



Например, так как в каждом отделе обязательно должен быть руководитель (связь «руководит»), то каждой сущности «ОТДЕЛ» непременно должна соответствовать сущность «СОТРУДНИК». Однако не каждый сотрудник является руководителем отдела, следовательно, в данной связи не каждая сущность «СОТРУДНИК» имеет ассоциированную с ней сущность «ОТДЕЛ».

Таким образом, говорят, что сущность «СОТРУДНИК» имеет *обязательный класс принадлежности*, а сущность «ОТДЕЛ» имеет *необязательный класс принадлежности*. В дальнейшем обязательность бинарных связей будем обозначать следующим образом:

ПРИМЕР

Рассмотрим пример моделирования предметной области.

Предположим, что нашу задачу можно разделить на два локальных представления.

Выделим интересующие нас сущности и связи первого локального представления:

Прежде всего, предприятие состоит из отделов, в которых работают сотрудники, причем на предприятии не разрешено совмещение должностей. Оклад каждого сотрудника зависит от занимаемой им должности. В то же время, одну и ту же должность могут занимать одновременно несколько сотрудников. В результате этих рассуждений мы должны ввести наборы сущностей:

ОТДЕЛ (КОД_ОТДЕЛА, ИМЯ_ОТДЕЛА)

СОТРУДНИК (ТАБЕЛЬНЫЙ_НОМЕР, ИМЯ, КОД_ДОЛЖ, КОД_ОТД)

ДОЛЖНОСТЬ (КОД_ДОЛЖ, ИМЯ_ДОЛЖНОСТИ, ОКЛАД).

Тогда ER-диаграмма первого локального представления будет иметь вид, представленный на рисунке.

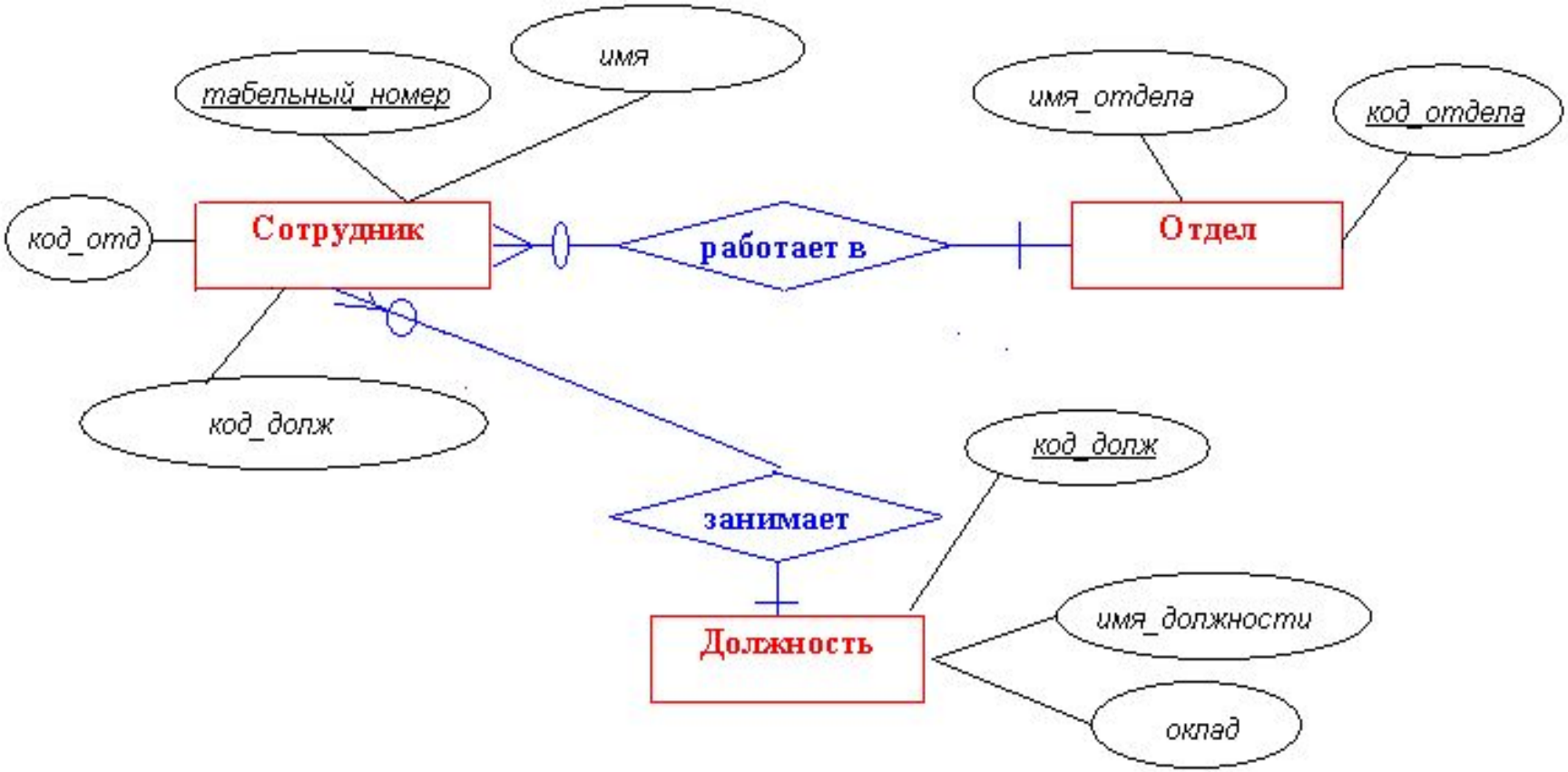


Рисунок 2.2 - ER-диаграмма первого локального представления.

Выделим интересующие нас сущности и связи второго локального представления:

Допустим, работники предприятия занимаются тем, что выполняют заказы, заключая договора, причем для выполнения заказа формируется одна или несколько рабочих групп.

Этому представлению соответствуют следующие сущности:

ЗАКАЗЧИК (КОД_ЗАК, ИМЯ_ЗАКАЗЧИКА, АДРЕС)

КОНТРАКТ (НОМЕР, СРОК_НАЧАЛА, СРОК_ОКОНЧАНИЯ, СУММА)

РАБОЧАЯ ГРУППА (КОД_ГР, НОМЕР_КНТР, ПРОЦЕНТ_ВОЗНАГРАЖДЕНИЯ)

Атрибут *«процент_вознаграждения»* отражает ту долю стоимости контракта, которая предназначена для оплаты труда членов соответствующей рабочей группы.

Рассмотрим теперь более внимательно информационный объект «ЗАКАЗЧИК». На практике очень часто возникает необходимость различать национальную принадлежность юридических лиц, с которыми предприятие вступает в договорные отношения. Это связано с тем, что для зарубежных фирм необходимо хранить, например, сведения о валюте, в которой осуществляются расчеты, языке, на котором подписан контракт и т.д. В свою очередь, для отечественных компаний необходимо иметь сведения о форме собственности (частная или государственная), поскольку от этого может зависеть порядок налогообложения средств, полученных за выполнение работ по контракту.

Таким образом, мы приходим к выводу, что необходимо ввести в рассмотрение еще два непересекающихся множества ЗАРУБЕЖНОЕ_ПРЕДПРИЯТИЕ (ВАЛЮТА, ЯЗЫК) и ОТЕЧЕСТВЕННОЕ_ПРЕДПРИЯТИЕ (ФОРМА_СОБСТВЕННОСТИ), объединение которых составляет полное множество ЗАКАЗЧИК.

Ассоциацию между этими объектами называют **отношением наследования** или **иерархической связью**, так как сущности ЗАРУБЕЖНОЕ_ПРЕДПРИЯТИЕ и ОТЕЧЕСТВЕННОЕ_ПРЕДПРИЯТИЕ наследуют атрибуты сущности ЗАКАЗЧИК (КОД_ЗАКАЗЧИКА, ИМЯ_ЗАКАЗЧИКА, АДРЕС). Для того, чтобы определить к какому подмножеству относится конкретная сущность из набора ЗАКАЗЧИК (и, соответственно, какой набор атрибутов она имеет) необходимо ввести атрибут *«национальная принадлежность»,* называемый ***дискриминантом***. Этот тип связи отображаетя на диаграмме, приведенной на рисунке 2.3.

Рисунок 2.3 – Отношение наследования.

Таким образом, ER-диаграмма второго локального представления будет иметь вид, представленный на рисунке 2.4.

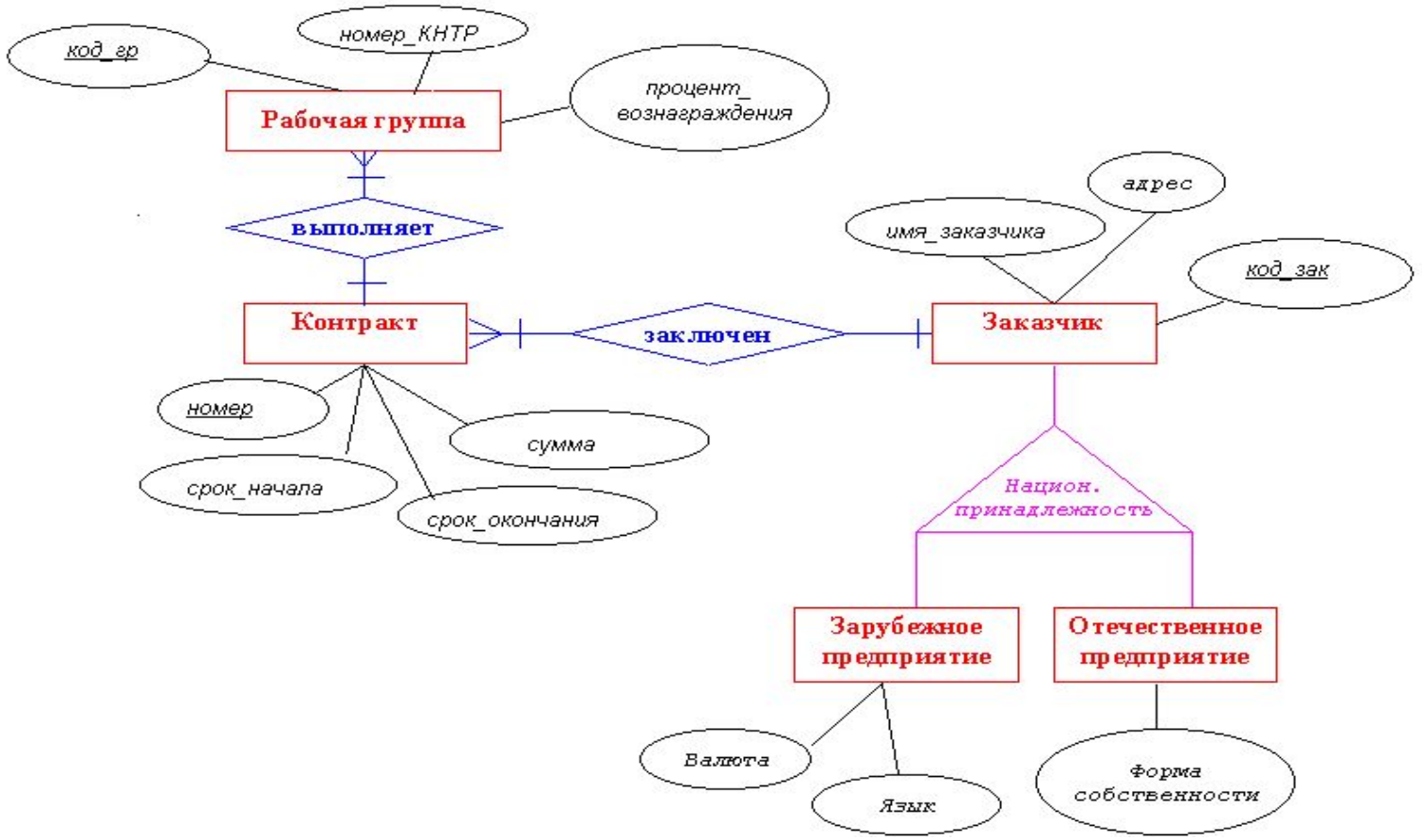


Рисунок 2.4 – ER - диаграмма второго локального представления.

Теперь можно объединить диаграммы двух локальных представлений (рисунок 2.5). Для этого надо определить, по каким сущностям эти два представления можно связать. Очевидно, что это сущности СОТРУДНИК и РАБОЧАЯ ГРУППА. Для организации связи в сущность СОТРУДНИК необходимо ввести атрибут КОД_РАБ_ГРУППЫ.

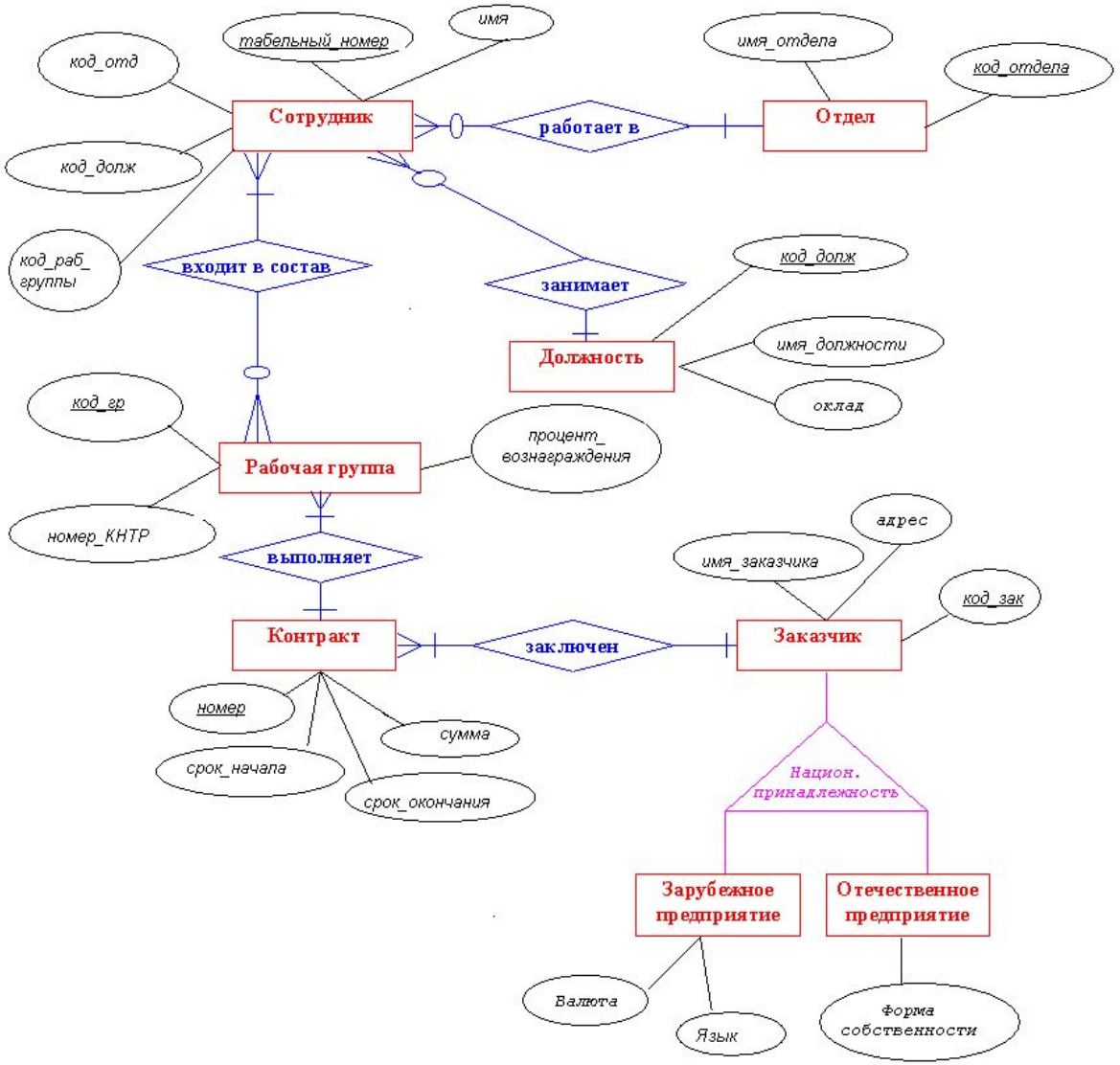


Рисунок 2.4 – Модель «сущность-связь»

Логическое проектирование БД

Основной задачей **логического проектирования** является разработка **даталогической схемы (модели),** ориентированной на выбранную систему управления базами данных.

Одним из основных критериев выбора СУБД является оценка того, насколько эффективно внутренняя модель данных, поддерживаемая системой, способна описать концептуальную схему предметной области.

Процесс логического проектирования состоит из следующих действий:

- отображение концептуальной схемы на логическую схему, получение даталогической модели данных, соответствующей внешнему уровню архитектуры автоматизируемой информационной системы;
- выбор ключей;
- нормализация отношений.

При отображении концептуальной модели данных предметной области на реляционную модель данных каждый прямоугольник ER-диаграммы - информационный объект преобразуется в таблицу (отношение).

Одни и те же данные могут группироваться в таблицы-отношения, различными способами, то есть, возможна организация различных наборов отношений взаимосвязанных информационных объектов предметной области. Группировка атрибутов в отношениях должна быть рациональной, предельно сокращающей дублирование данных и упрощающей процедуры их обработки и обновления.

Контрольные вопросы к блоку 2

1. Дайте определение понятиям нормализация, процесс нормализации.
2. Что такое нормальная форма?
3. Перечислите и определите все нормальные формы.
4. Что такое проектирование?
5. Перечислите основные этапы проектирования базы данных.
6. Перечислите задачи этапа системного анализа.
7. Что называется инфологической моделью?
8. Перечислите задачи этапа логического проектирования.
9. Перечислите задачи этапа физического проектирования.
10. Что называется моделью «Сущность-связь»?
11. Определите основное назначение модели «Сущность-связь»?
12. Перечислите виды атрибутов.
13. Определите типологию связей.
14. Определите соотношение понятий «сущность» и «связь».
15. Что такое ER-диаграмма?