

ТЕМА 2: Проектирование баз данных

ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ – процесс создания схемы базы данных и определения необходимых ограничений целостности

ОСНОВНАЯ ЦЕЛЬ ПРОЕКТИРОВАНИЯ БД –

это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте.

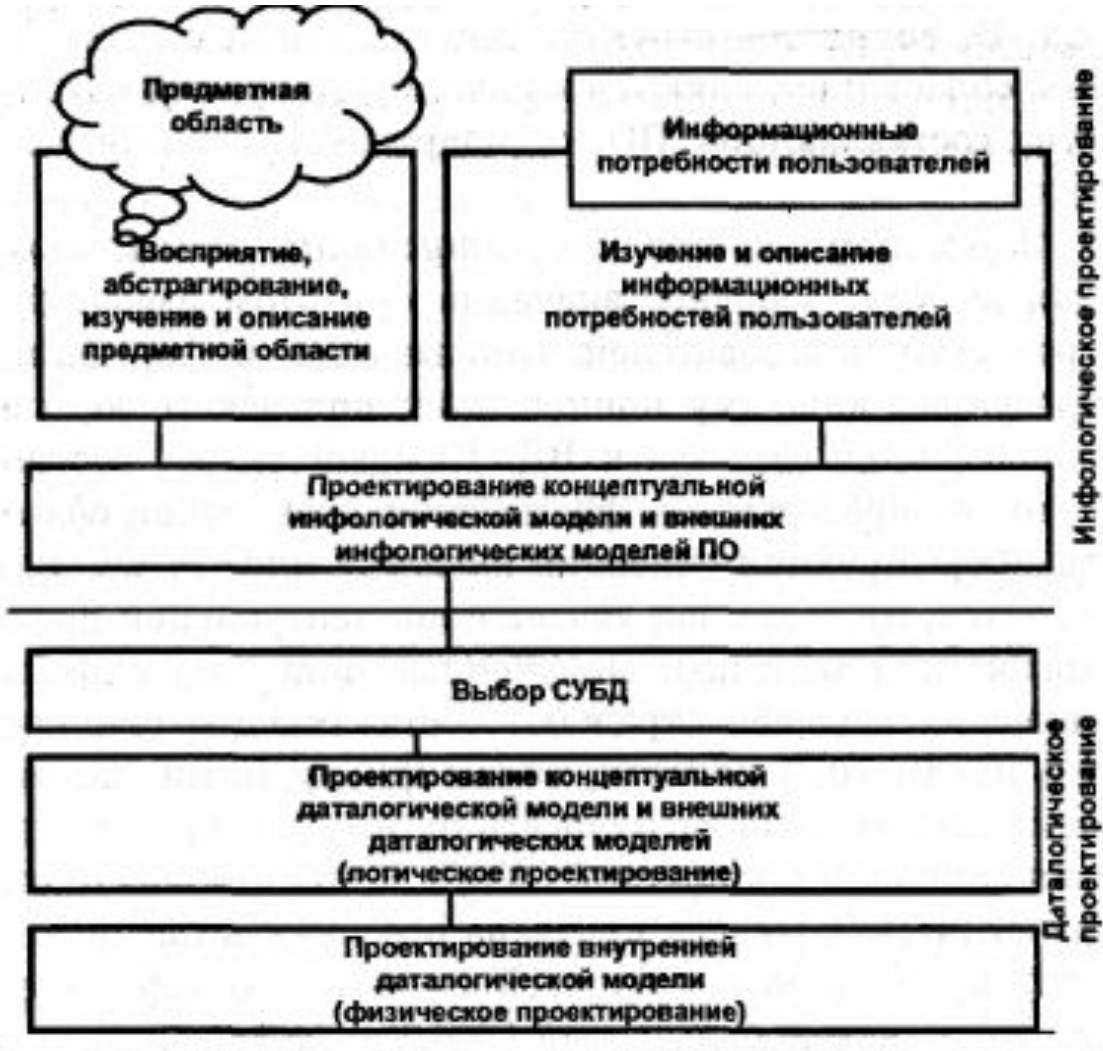
ПРИ ПРОЕКТИРОВАНИИ БАЗЫ ДАННЫХ РЕШАЮТСЯ ДВЕ ОСНОВНЫХ ПРОБЛЕМЫ.

- 1. Проблема логического проектирования**, заключающаяся в том, каким образом отобразить объекты области в модели данных, чтобы это отображение не противоречило области и было по возможности лучшим (эффективным, удобным и т.д.)?
- 2. Проблема физического проектирования**, заключающаяся в том, как обеспечить эффективность выполнения запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создание каких дополнительных структур (например, индексов)



ТЕМА 2: Проектирование баз данных

ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ



1. Инфологическое проектирование;
2. Даталогическое проектирование:
 - а) логическое проектирование;
 - б) физическое проектирование.



ТЕМА 2: Проектирование баз данных

ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

ЗАДАЧА ИНФОЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ – получение смысловых моделей, отражающих информационное содержание конкретной области. На этом этапе выполняется восприятие реальной действительности, изучение и описание предметной области.

ОПРЕДЕЛЯЮТСЯ ГРАНИЦЫ ОБЛАСТИ, ПРОИСХОДИТ АБСТРАГИРОВАНИЕ ОТ НЕСУЩЕСТВЕННЫХ ЧАСТЕЙ



ИЗУЧАЕТСЯ ПРЕДМЕТНАЯ ОБЛАСТЬ, НАКАПЛИВАЮТСЯ ЗНАНИЯ О НЕЙ (ЭТИ ЗНАНИЯ ПРЕДСТАВЛЯЮТСЯ В ВИДЕ МОДЕЛЕЙ: МАТЕМАТИЧЕСКИХ ФОРМУЛ, ДИАГРАММ)



ВЫПОЛНЯЕТСЯ СТРУКТУРИЗАЦИЯ ЗНАНИЙ О ПРЕДМЕТНОЙ ОБЛАСТИ



КОМПОНУЕТСЯ КОНЦЕПТУАЛЬНАЯ ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ, ОСНОВНОЕ ЗНАЧЕНИЕ ПРИ ЭТОМ ИМЕЮТ ПОТРЕБНОСТИ ПОЛЬЗОВАТЕЛЕЙ (ОПИСЫВАЕТСЯ ИНФОРМАЦИЯ, ТРЕБУЕМАЯ КАЖДОМУ КОНКРЕТНОМУ ПОЛЬЗОВАТЕЛЮ, Т.Е. ОПИСЫВАЮТСЯ ЗАПРОСЫ К БД)

ЗАДАЧА ЛОГИЧЕСКОГО ЭТАПА ПРОЕКТИРОВАНИЯ – организация данных, в форму, принятую в выбранной конкретной СУБД.

ЗАДАЧА ФИЗИЧЕСКОГО ЭТАПА ПРОЕКТИРОВАНИЯ – выбор рациональной структуры хранения данных и методов доступа к ним, исходя из арсенала методов и средств, который предоставляется разработчику системой управления базами данных.

ТЕМА 2: Проектирование баз данных

ER-диаграмма (entity-relationship diagram, диаграмма «сущность-связь») – это графическое представление инфологической модели.

ОСНОВНЫЕ ЭЛЕМЕНТЫ ER-МОДЕЛЕЙ:

- 1) сущности (объекты);
- 2) атрибуты сущностей;
- 3) связи между сущностями.

Атрибут сущности – это именованная характеристика, являющаяся некоторым свойством сущности. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными). Примерами атрибутов сущности «Сотрудник» могут быть такие атрибуты как «Табельный номер», «Фамилия», «Имя», «Отчество», «Должность», «Зарплата» и т.п. Атрибуты изображаются в пределах прямоугольника, определяющего сущность:

СУЩНОСТЬ – это класс однотипных объектов, информация о которых должна быть учтена в модели. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. Примерами сущностей могут быть такие классы объектов как «Поставщик», «Сотрудник», «Накладная». Каждая сущность в модели изображается в виде прямоугольника с наименованием.

Ключ сущности – это избыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Избыточность заключается в том, что удаление любого атрибута из ключа нарушается его уникальность.

Экземпляр сущности – это конкретный представитель данной сущности. Например, представителем сущности «Сотрудник» может быть «Сотрудник Иванов». Экземпляры сущностей должны быть различимы, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.



ТЕМА 2: Проектирование баз данных

Сущность может иметь несколько различных ключей. Ключевые атрибуты изображаются на диаграмме подчеркиванием:

Сотрудник
<u>Табельный номер</u>
Фамилия
Имя
Отчество
Должность
Зарплата

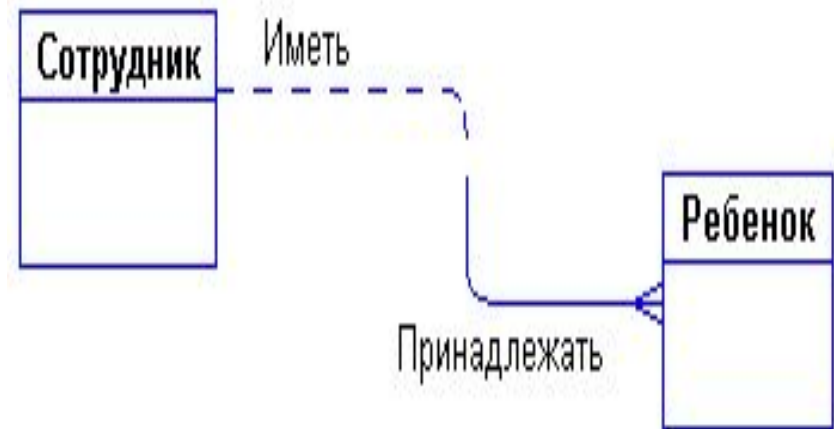
Связь - это некоторая ассоциация между двумя сущностями. Одна сущность может быть связана с другой сущностью или сама с собою.

Связи позволяют по одной сущности находить другие сущности, связанные с ней.

Например, связи между сущностями могут выражаться следующими фразами:

«СОТРУДНИК может иметь несколько ДЕТЕЙ»;

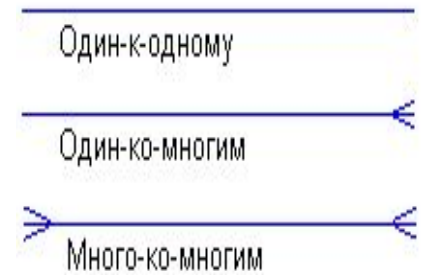
«каждый СОТРУДНИК обязан числиться ровно в одном ОТДЕЛЕ». Графически связь изображается линией, соединяющей две сущности:



ТЕМА 2: Проектирование баз данных



Каждая связь имеет два конца и одно или два наименования. Наименование обычно выражается в неопределенной глагольной форме: «иметь», «принадлежать» и т.п. Каждое из наименований относится к своему концу связи. Иногда наименования не пишутся ввиду их очевидности. Каждая связь может иметь один из следующих типов связи:



Связь типа **один-к-одному** означает, что один экземпляр первой сущности связан с одним экземпляром второй сущности.

Связь типа **один-ко-многим** означает, что один экземпляр первой сущности связан с несколькими экземплярами второй сущности. Это наиболее часто используемый тип связи. Сущность со стороны «один» называется **родительской**, сущность со стороны «много» — **дочерней**.

Связь типа **много-ко-многим** означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.

ТЕМА 2: Проектирование баз данных

КАЖДАЯ СВЯЗЬ МОЖЕТ ИМЕТЬ ОДНУ ИЗ ДВУХ МОДАЛЬНОСТЕЙ СВЯЗИ:

Модальность «**МОЖЕТ**» означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром.

Модальность «**ДОЛЖЕН**» означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.

Может

Должен

БАЗА ДАННЫХ СЧИТАЕТСЯ НОРМАЛИЗОВАННОЙ, ЕСЛИ ЕЕ ТАБЛИЦЫ ПРЕДСТАВЛЕНЫ КАК МИНИМУМ В ТРЕТЬЕЙ НОРМАЛЬНОЙ ФОРМЕ.


Первая нормальная форма 1НФ требует атомарности данных в таблицах, т.е. данные в таблицах должны быть представлены в виде минимально возможных и далее неделимых кусочков информации.

Вторая нормальная форма 2НФ требует, чтобы данные во всех неключевых столбцах полностью зависели от первичного ключа.

Третья нормальная форма, известная также как 3НФ, требует от нас, чтобы структура базы данных удовлетворяла требованиям 1НФ и 2НФ. Все неключевые столбцы таблицы зависели от первичного ключа, но были независимы друг от друга.

ТЕМА 2: Проектирование баз данных

ПРИВЕДЕНИЕ ТАБЛИЦЫ БД К ПЕРВОЙ НОРМАЛЬНОЙ ФОРМЕ:



Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R



Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

в одной ячейке содержится список из 3 элементов т.е. он не является атомарным

ПРИВЕДЕНИЕ ТАБЛИЦЫ БД КО ВТОРОЙ НОРМАЛЬНОЙ ФОРМЕ:

<u>Фирма</u>	<u>Модель</u>	Цена	Скидка
BMW	M5	5500000	5%
BMW	X5M	6000000	5%
BMW	M1	2500000	5%
Nissan	GT-R	5000000	10%



<u>Фирма</u>	<u>Модель</u>	Цена
BMW	M5	5500000
BMW	X5M	6000000
BMW	M1	2500000
Nissan	GT-R	5000000



<u>Фирма</u>	Скидка
BMW	5%
Nissan	10%

Таблица находится в первой нормальной форме, но не во второй. **ЦЕНА МАШИНЫ ЗАВИСИТ ОТ МОДЕЛИ И ФИРМЫ. СКИДКА ЗАВИСИТ ОТ ФИРМЫ, ТО ЕСТЬ ЗАВИСИМОСТЬ ОТ ПЕРВИЧНОГО КЛЮЧА НЕПОЛНАЯ.**

ТЕМА 2: Проектирование баз данных

ПРИВЕДЕНИЕ ТАБЛИЦЫ БД К ТРЕТЬЕЙ НОРМАЛЬНОЙ ФОРМЕ:



<u>Модель</u>	Магазин	Телефон
BMW	Риал-авто	87-33-98
Audi	Риал-авто	87-33-98
Nissan	Некст-Авто	94-54-12

<u>Магазин</u>	Телефон
Риал-авто	87-33-98
Риал-авто	87-33-98
Некст-Авто	94-54-12

<u>Модель</u>	Магазин
BMW	Риал-авто
Audi	Риал-авто
Nissan	Некст-Авто

В отношении атрибут «**Модель**» является первичным ключом. Личных телефонов у автомобилей нет, и телефон зависит исключительно от магазина.

Таким образом, в отношении существуют следующие функциональные зависимости:

МОДЕЛЬ → **МАГАЗИН**

МАГАЗИН → **ТЕЛЕФОН**

МОДЕЛЬ → **ТЕЛЕФОН**

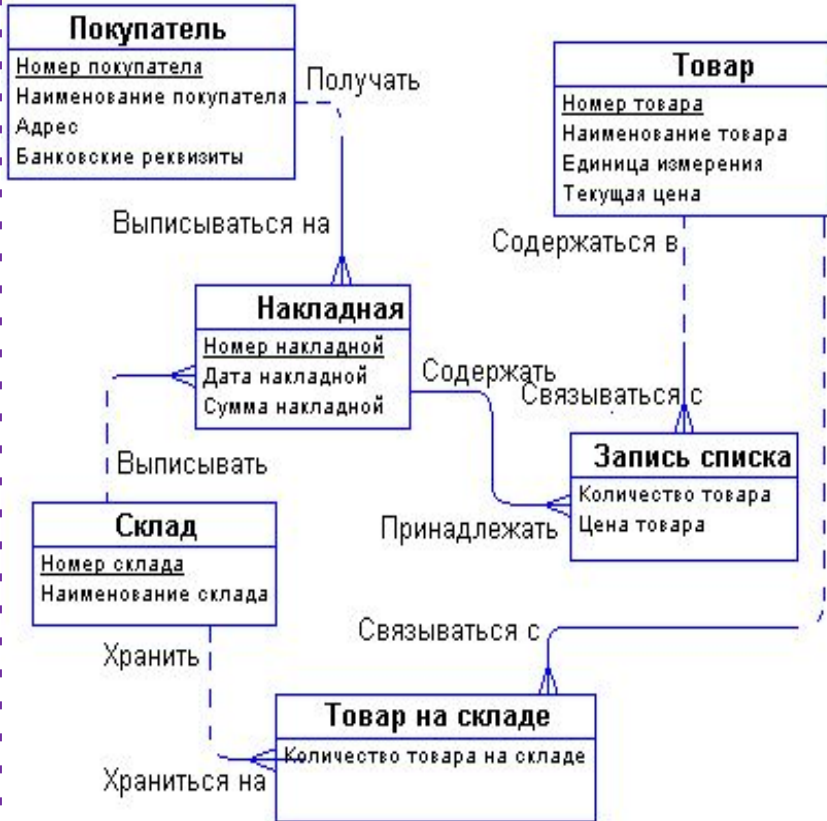
Зависимость **МОДЕЛЬ** → **ТЕЛЕФОН** является транзитивной, следовательно, отношение не находится в 3НФ.

В результате разделения исходного отношения получаются два отношения, находящиеся в 3НФ:



ТЕМА 2: Проектирование баз данных

ПРИМЕР ER-ДИАГРАММЫ



Разработанный пример ER-диаграммы является примером **концептуальной** диаграммы. Это означает, что диаграмма не учитывает особенности конкретной СУБД. По данной концептуальной диаграмме можно построить **физическую** диаграмму, которая уже будут учитываться такие особенности СУБД, как допустимые типы и наименования полей и таблиц, ограничения целостности и т.п.

